Universidad del Aconcagua

Facultad de ciencias sociales y administrativas

Ingeniería en Software

Rodolfo Adrián Becerra
2011
Tutor: Alejandro Vázquez

Sistemas Expertos para la realización de diagnóstico de trastornos neuromusculares con electromiografía

Mendoza, 14 de Abril de 2011

Índice

1. Resumen Técnico

La presente tesina desea investigar la utilización de sistemas expertos para la realización de sistema para diagnóstico de enfermedades neuromusculares con electrometría y demostrar la factibilidad de realizar un sistema de estar características usando reglas de producción, así como también las ventajas que esta tecnología nos brinda.

Para crear un sistema experto basado en reglas de producción es necesario un sistema de gestión de reglas de negocio (BRMS, por las siglas en inglés de business rule management system), que en este caso se propone usar Drools, el cual tiene un motor de reglas basado en inferencia de encadenamiento hacia adelante (forward chaining), más correctamente conocido como sistema de reglas de producción, usando una implementación avanzada del algoritmo Rete.

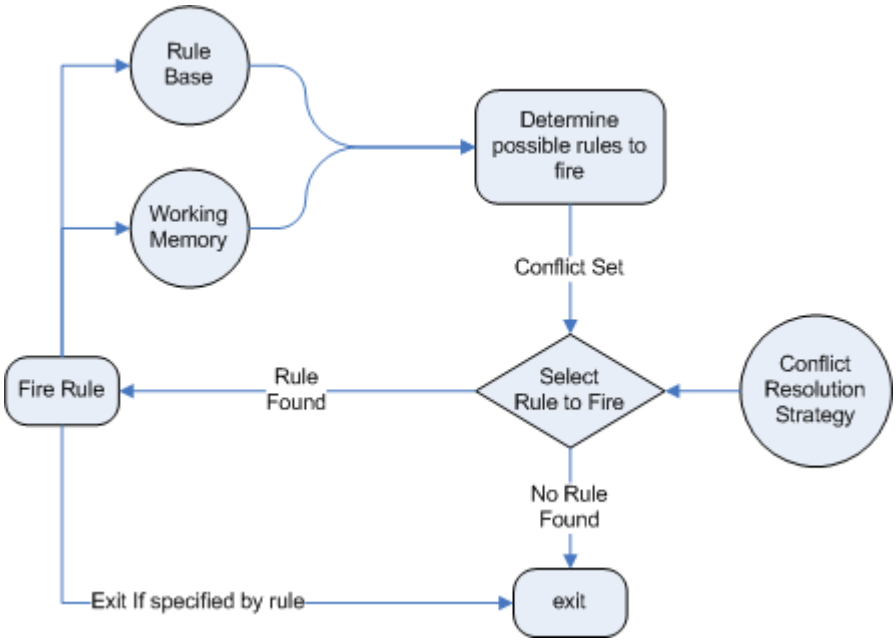
El algoritmo Rete es un algoritmo de reconocimiento de patrones eficiente para implementar un sistema de producción de reglas y es la base de diversas implementaciones más eficientes de sistemas expertos.

Drools es software libre distribuido según los términos de la licencia Apache.

El método de inferencia que tendría el sistema es el que se muestra en la imagen abajo.

Este tipo método de razonamiento trabaja con los hechos contenidos en la memoria de trabajo y las reglas almacenadas en la memoria de producción. A continuación se determina las posibles reglas que pueden ser lanzadas. A partir de aquí, se obtiene un conjunto de conflicto, que es básicamente, las reglas que pueden ser aplicadas a los hechos analizados. El paso siguiente es seleccionar que regla deberá ser lanzada, usando una estrategia de resolución de conflicto, la cual esta previamente configurada en el motor de inferencia. Si existe un regla para ser lanzada, la regla es lanzada, si el resultado de esa regla es un hecho nuevo este es insertado en la memoria de trabajo, pero

en caso de que dicha regla nos dé como resultado otra regla esta es insertada en la memoria de producción. En caso de que no exista una regla para ser lanzada el flujo de trabajo termina.



Las reglas de producción del sistema estarán almacenadas en un archivo con extensión .drl. Allí se almacenarán todas las reglas necesarias para identificar las diferentes enfermedades neuromusculares.

El sistema debe trabajar con una base de datos. En este trabajo se ha optado usar PosgreSql por diferentes razones explicadas en el punto 3.4.7 Base de Datos.

El desarrollo del sistema se puede realizar en diferentes IDEs, los más usados son NetBeans y Eclipse. Se recomienda usar Eclipse ya que el mismo contiene un plugin para la utilización de Drools.

El sistema será realizado utilizando Java por lo que será multiplataforma pudiendo ser usado en el Sistema Operativo que uno desee.

Como complemento a los síntomas que presenta el paciente, se tomarán valores del examen electromiográfico que se serán insertados por el médico especialista para generar el diagnóstico.

Los valores que se tomarán del examen electromiográfico, para luego compararlos con valores normales de pacientes sanos son los parámetros del PUM (Potencial del Umbral Motora), los cuales son amplitud, duración, estabilidad, morfología y el patrón de máximo esfuerzo.

2. Introducción

Desde la aparición de las computadoras hasta nuestros días, la gente ha invertido grandes esfuerzos por tratar de dar una cierta capacidad de decisión a estas máquinas, incluso un cierto grado de inteligencia.

Un Sistema Experto en sí no tiene verdadera Inteligencia Artificial, más bien, es un sistema basado en el conocimiento que, mediante el buen diseño de su base de información y un adecuado motor de inferencias para manipular dichos datos proporciona una manera de determinar resoluciones finales dados ciertos criterios.

Un sistema experto de cómputo es el encargado de tomar decisiones altamente especializadas con base en los conocimientos de expertos sobre un área en particular, por lo que los datos son almacenados de forma estructurada para su recuperación. Además de la capacidad de ofrecer soluciones sobre algún problema, incluye la explicación del porque se llegó a determinadas medidas.

La medicina es un área en donde se requiere de mucho entrenamiento para ser un especialista, además, cuando existe una amplia diversidad de enfermedades y trastornos, los síntomas pueden ser confusos cuando se busca determinar rápidamente un diagnóstico oportuno, que puede significar la sobrevivencia o la muerte del paciente.

En este sentido, el sistema experto sustituye al especialista en un área dominada plenamente por el médico. La parte importante son los recursos que se refieren al conocimiento almacenado adquirido, ya sea con la ayuda de un especialista o bien, a través del sistema que integra un módulo de aprendizaje, donde se construye su propio conocimiento.

2.1 Aéreas disciplinarias y campo de operación

Los sistemas expertos pueden ser aplicados a una gran variedad de áreas, entre

ellas: medicina, análisis de estados financieros, planificación financiera, Industria, robótica, reconocimiento de patrones, etc.

Los campos de operación más destacados son los siguientes:

- Control de procesos, supervisión.
- Diseño.
- Diagnóstico.
- Planificación.
- Asesoramiento.
- Formación.
- Capacitación.
- Monitoreo.
- Interpretación
- Predicción.

2.2 Definición del Problema

El examen de electromiografía se utiliza para el pronóstico y/o diagnóstico de una patología que afecta al Sistema Nervioso. La realización del diagnóstico de un trastorno neuromuscular envuelve la habilidad del médico para identificar un defecto específico en la función neuromuscular. Algunas veces, el médico puede deducir cual es el defecto funcional así como la enfermedad asociada con él por medio de un examen físico, mandando a hacer una prueba sanguínea u observando la anatomía de nervios y músculos. Pero otras veces; el médico tendrá que evaluar de forma más específica y directa la función de los nervios y los músculos, así como las conexiones entre ellos, por medio de la electromiografía (EMG). El médico es el responsable de realizar luego de la electromiografía un diagnóstico, lo cual conlleva pérdida de tiempo, posibilidades de errores humanos, lo que a veces da resultados no confiables y divergencias de opiniones dependiendo de cada médico. Este proyecto propone aprovechar las características

enmarcadas en un Sistema Experto para evaluar el examen de la conducción nerviosa (NCV) del electromiógrafo, asociado al conocimiento del experto para obtener un pronóstico y/o un diagnóstico definitivo. De esta manera, el sistema permitirá ahorrar tiempo, dando soporte a la toma de decisiones y obteniendo resultados completamente confiables ya que están basados en el conocimiento del experto.

2.3 Inserción institucional y ámbito de cobertura

Los Sistemas Expertos son aplicables allí donde es necesario:

- Resolver problemas para los que no existe un modelo matemático adecuado o su solución es muy compleja, como en Medicina, Ingeniería, Exploración, Diseño, Análisis, etc.
- Preservar el conocimiento de expertos y hacerlo accesible a más personas. Esto es, diseminar conocimientos escasos y costosos de adquirir.
- Explicar al usuario el proceso de razonamiento seguido para llegar a los resultados. Formalizar y clarificar conocimientos.
- Evitar fallos en labores rutinarias complejas
- Ampliar de forma más rápida los conocimientos de los especialistas.
- Diagnosticar los fallos con mayor rapidez y conseguir tareas de planificación más completas y consistentes.
- Manejar grandes volúmenes de datos con técnicas no convencionales.

2.4 Justificación del proyecto

Los sistemas expertos son de mucha utilidad en la vida real, y apoyan en gran manera a los sistemas de soporte a la decisión, ya que nos permiten realizar decisiones basadas en la experiencia humana de algún especialista en determinada área, como por

ejemplo la medicina, esto es con el fin de retener el conocimiento y conllevándonos a una toma de decisiones más apegada a la realidad y con más información.

Con esto podemos establecer que uno de los bienes mejor valuados es el conocimiento humano, y con esto la capacidad de tomar decisiones y de aportar un punto de opinión. En la actualidad, con la ayuda de las personas especializadas podemos crear un sistema que simule la evaluación de los mismos a través de estudio de cierta situación. A esto se le conoce como sistema experto.

En los últimos años, la progresiva incorporación de los avances tecnológicos (aplicación de la informática) en la práctica de la medicina han hecho que esta ciencia médica del futuro sufra un cambio radical.

Los Sistemas Expertos nos permiten tomar mejores decisiones, que lógicamente, se traducen en resultados positivos. Está claro que el desarrollo de estos sistemas ha ido incrementándose a través del tiempo, y por lo tanto ha podido ayudar a muchísima gente, específicamente en el ámbito de la medicina, por lo que se considera de gran importancia abordar este proyecto de diagnóstico de trastornos neuromusculares.

El presente trabajo pretende contribuir en las investigaciones de Sistemas expertos aplicados a la Medicina.

2.5 Hipótesis

A través de un Sistema Experto de diagnóstico de trastornos neuromusculares se reduce considerablemente, a razón de solo unos minutos, el tiempo destinado al diagnóstico de una patología neuromuscular, obteniendo resultados totalmente confiables, evitando fallos humanos y brindando la posibilidad de consultar como se ha llegado a ese resultado.

2.6 Objetivos

2.6.1 General

Determinar la viabilidad de desarrollo de un Sistema Experto para la realización de diagnóstico de trastornos neuromusculares con electromiografía.

2.6.2 Específicos

- Realizar un cronograma que estime el tiempo necesario para realizar el proyecto, así como también los recursos asignados a cada tarea.
- Analizar que arquitectura debería tener el sistema experto para la implantación de un sistema de diagnóstico para enfermedades neuromusculares con electromiografía.
- Definir los costos de la implementación de las soluciones.
- Determinar las tecnologías a integrar para el desarrollo de la solución.

2.7 Antecedentes

Los sistemas expertos surgen, entonces, de la inteligencia artificial a mediados de los años sesenta. En ese período se creía que bastaban unas pocas leyes de razonamiento junto con potentes computadoras para producir resultados brillantes.

Un intento en ese sentido fue el llevado a cabo por los investigadores Alan Newell y Herbert Simon que desarrollaron un programa denominado GPS (General Problem Solver ó solucionador general de problemas). Podía trabajar con criptoaritmética, con las torres de Hanoi y con otros problemas similares. Lo que no podía hacer el GPS era resolver problemas del mundo real, tales como un diagnóstico médico.

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

A partir de 1965, un equipo dirigido por Edward Feigenbaum, comenzó a desarrollar sistemas expertos utilizando bases de conocimiento definidas minuciosamente.

En 1967 se construye DENDRAL, que se considera como el primer sistema experto. Se utilizaba para identificar estructuras químicas moleculares a partir de su análisis espectrográfico.

Entre 1970 y 1980 se desarrolló MYCIN para consulta y diagnóstico de infecciones de la sangre. Este sistema introdujo nuevas características: utilización de conocimiento impreciso para razonar y posibilidad de explicar el proceso de razonamiento. Lo más importante es que funcionaba de manera correcta, dando conclusiones análogas a las que un ser humano daría tras largos años de experiencia. En MYCIN aparecen claramente diferenciados motor de inferencia y base de conocimientos. Al separar esas dos partes, se puede considerar el motor de inferencias aisladamente. Esto da como resultado un sistema vacío o Shell.

Así surgió EMYCIN (MYCIN Esencial) con el que se construyó SACON, utilizado para estructuras de ingeniería, PUFF para estudiarla función pulmonar y GUIDON para elegir tratamientos terapéuticos.

En esa poca se desarrollaron también: HERSAY, que intentaba identificar la palabra hablada, y PROSPECTOR, utilizado para hallar yacimientos de minerales. De este último derivó el shell KAS (Knowledge Acquisition System).

A partir de 1980 se ponen de moda los sistemas expertos, numerosas empresas de alta tecnología investigan en esta área de la inteligencia artificial, desarrollando sistemas expertos para su comercialización. Se llega a la conclusión de que el éxito de un

sistema experto depende casi exclusivamente de la calidad de su base de conocimiento.

El inconveniente es que codificar la pericia de un experto humano puede resultar difícil, largo y laborioso.

Un ejemplo de sistema experto moderno es CASHVALUE, que evalúa proyectos de inversión y VATIA, que asesora acerca del impuesto sobre el valor a añadido o I.V.A.

2.8 Aplicaciones en la medicina

Para realizar un diagnóstico se requiere información sobre los síntomas del paciente, condición general, historial clínico y resultados del laboratorio. Estos datos se obtienen a partir de una serie de preguntas, cada una de las cuales es determinada a partir de la respuesta anterior del paciente utilizando diversas reglas o a través de la experiencia (almacenada en la memoria del ser humano experto o bien, del sistema experto). Al principio las preguntas son generadas para reducir el número de enfermedades posibles planteando una hipótesis, y al final se realizan preguntas para soportar el diagnóstico.

Una de las formas comunes de llegar a un diagnóstico es mediante el interrogatorio al paciente, en este sentido, los sistemas expertos son los más aptos para esta tarea. Cuando el interrogatorio al paciente se realiza de forma correcta se podrá elegir el tratamiento adecuado para su problema.

Otro punto a favor de los sistemas expertos es que al tener almacenado el conocimiento en medios electrónicos, nunca se deteriorará, por el contrario, con el módulo de aprendizaje se logran ingresar nuevas reglas para tratar nuevas enfermedades o trastornos, lo que asegura también que al realizar la prueba en pacientes con los mismos síntomas se diagnostique de la misma forma. A pesar de la precisión de los sistemas expertos, una parte que hace falta para poder respaldar los resultados o para poder llegar a ellas más rápido es la exploración física.

Las formas de razonamiento diagnóstico tienen similitud con los razonamientos de los sistemas expertos:

- **Probabilísticas**. Se basan en la frecuencia de ocurrencia de las enfermedades y consideran variables como sexo, edad, peso, frecuencia y la probabilidad asociada entre síntomas-enfermedad.
- **Causales**. Encuentran relaciones fisiopatológicas y las relacionan con los efectos que causan, que pueden ser datos clínicos o antecedentes, así como el humor del paciente, por citar algunos.
- **Determinísticos**. Son mucho más directos, ya que identificando cada síntoma, se asocia con una regla que lleva directamente hacia el diagnóstico. Se pueden analizar, por ejemplo, la presencia de cefaleas, fiebre, alteraciones de la conciencia y rigidez de nuca pueden significar meningoencefalitis.

2.9 Limitaciones

El sistema no contemplará la entrega del electromiógrafo, el mismo debe ser adquirido independientemente del sistema.

El sistema no contemplará el análisis de señal del electromiograma, el sistema se limita solo a recibir, como valores de entrada, los datos que el electromiógrafo da como resultado. Luego esos resultados serán analizados de acuerdo a valores normales cargados en las reglas del sistema.

En el proyecto no se implementará el sistema propuesto, el objetivo del mismo es realizar un análisis de cómo aplicar un sistema experto para diagnóstico de enfermedades electromusculares, demostrando su factibilidad y que con un sistema de estas características es posible reducir tiempos en las consulta de los pacientes, agilizando el proceso de diagnóstico de dichas enfermedades.

3 Marco teórico

3.1 Inteligencia Artificial.

Algunas de las definiciones de Inteligencia artificial son:

- La inteligencia artificial es el estudio de cómo hacer que los ordenadores hagan cosas que, en estos momentos, hace mejor el hombre. [Elaim Reich]
- Bajo Inteligencia entiendo la capacidad de un ser vivo o una maquina de ordenar informaciones, extensas observaciones, experiencias, descubrir interrelaciones para abstraer de esta forma cosas y poderlas ligar entre sí. [Alexander Sporl, 1971]
- La automatización de actividades que vinculamos con procesos de pensamiento humano, actividades tales como toma de decisiones, resolución de problemas, aprendizaje,...[Bellman, 1978]
- El estudio de las facultades mentales mediante el uso de modelos computacionales. [Charniak y McDermott, 1985]
- La rama de la ciencia de la computación que se ocupa de la automatización de la conducta inteligente. [Luger y Stubblefield, 1993]

3.2 Sistemas Expertos

3.2.1 Concepto

Los sistemas expertos son programas que capturan el conocimiento de un experto e imitan sus procesos de razonamiento cuando resuelven los problemas en un determinado dominio.

Los sistemas expertos son un subconjunto especial dentro de los sistemas basados en el conocimiento, que incorporan en la base de conocimiento (KDD) del sistema el conocimiento de un experto.

Una definición formal de los sistemas expertos, aceptada por muchos autores, es la aprobada por el Grupo Especialista en Sistemas Expertos de la Sociedad Británica de Ordenadores, que los define de la forma siguiente:

"Un sistema experto es visto como la incorporación en un ordenador de un componente basado en el conocimiento, que se obtiene a partir de la pericia (conocimiento técnico) de un experto, de tal forma que el sistema pueda ofrecer asesoramiento inteligente o tomar una decisión inteligente sobre una función del proceso. Una característica adicional deseable, que muchos considerarían fundamental, es la capacidad del sistema, si se le solicita, de justificar su propia línea de razonamiento de un modo directamente inteligible para el interrogador..."

Un Sistema Experto es un sistema capaz de realizar una tarea que generalmente se considera que es difícil y que requiere cierto grado de experiencia humana.

Muchos de los sistemas expertos que han sido desarrollados en los últimos quince años han sido implantados como sistemas basados en reglas de producción. Una de las razones es que ciertos tipos de conocimiento experto pueden ser codificados muy fielmente como conjunto de reglas. [Frost, 1989]

Un Sistema experto es un programa de software que apoya la toma de decisiones tomando la información y construyendo el conocimiento de un experto humano. Y la base de conocimiento es la que contiene las ideas o conceptos de un campo especifico, con un cierto grado de especialización. Normalmente estas bases representan el conocimiento en forma de reglas que por lo regular son de forma si – entonces, permitiéndole contemplar el conocimiento heurístico incluyendo la intuición, el discernimiento, las inferencias y todos estos datos, no olvidemos, los adquiere de un experto en el área. Por lo tanto los

Sistemas Expertos como los Agentes Inteligentes, al tener la base de conocimiento puede razonar, hacer interfaces y determinar criterios.[Shadbolt, 1999]

Tanto en los Sistemas Expertos como en los Agentes Inteligentes hay un actor muy importante, el ingeniero del conocimiento, ya que es el que traduce el conocimiento del experto en conocimientos y reglas con base en los hechos para crear una base de conocimiento. Hay muchas herramientas y métodos que utiliza el ingeniero del conocimiento para obtener el conocimiento entre los cuales se encuentra la entrevista, la encuesta y la observación, pero el método más sencillo es el propuesto en el artículo de los investigadores Shadbolt y Milton, donde describen de una manera sencilla los pasos para obtener información del experto.

1. Conducir una entrevista con el experto para sondear el conocimiento que se va a adquirir, conocer la terminología, el propósito del conocimiento y el sistema.

Se muestra en el anexo 2.

2. Crear un diagrama conceptual derivado de los resultados de la entrevista y utilizarlo para generar preguntas que cumplan con los propósitos del sistema.

3. Realizar otra entrevista, semi-estructurada, con el experto utilizando las preguntas del paso 2.

4. Generar los conceptos, reglas, atributos, valores, relacionados que van surgiendo de las entrevistas.

5. Representar estos elementos de la manera más apropiada (Texto, diagramas, ilustraciones, hipertexto, anécdotas, etc.)

6. Representar los resultados al experto y permitirle realizar cambios en el conocimiento ya capturado.

7. Consultar con otros expertos y permitirle realizar cambios en el conocimiento ya capturado.

El sistema Experto y los Agentes Inteligentes, además de la base de

conocimientos deben contar con una interfaz humana, con la cual el usuario interactúa con el sistema, cuentan también con la máquina de inferencia, que une las entradas del usuario a la base de conocimientos, aplica principios lógicos y produce la ayuda experta solicitada. La máquina de inferencia busca información y relaciones en la base de conocimientos, y proporcionar respuestas, pronósticos y sugerencias en la misma forma en que lo haría un experto humano. Puede utilizar el encadenamiento mixto en el que se crean primero las conclusiones y se trabaja hacia atrás hasta los hechos de soporte. Si los hechos no apoyan la conclusión, se elige y prueba otra. O el encadenamiento hacia delante que se inicia con los hechos y trabaja hacia delante hasta las conclusiones.

Los Sistemas Expertos proponen grandes ventajas como: el ofrecer asesoría experta cuando no hay expertos humanos cerca; conservar el conocimiento de los expertos después de que estos abandonan una organización; combinar el conocimiento de varios expertos; lograr que el conocimiento esté disponible para mas personas, mejorar la productividad y el desempeño de quienes toman decisiones; reducir el número de errores humanos y finalmente es importante saber que un solo sistema experto puede ampliar las capacidades de toma de decisiones con el uso de Agentes Inteligentes.

Los sistemas expertos también pueden ser definidos en función de sus cualidades funcionales. En este sentido, Hayes-Roth considera como las funcionalidades más importantes las siguientes:

Pueden resolver problemas muy difíciles tan bien o mejor que los expertos humanos.

- Razonan heurísticamente, usando lo que los expertos consideran que son reglas empíricas efectivas, e interactúan con los humanos de forma adecuada, incluyendo el lenguaje natural.
- Manipulan y razonan sobre descripciones simbólicas.
- Pueden funcionar con datos que contienen errores, usando reglas de enjuiciamiento inciertas.

- Pueden contemplar múltiples hipótesis en competición simultáneamente.
- Pueden explicar por qué están formulando una pregunta.
- Pueden explicar su proceso de razonamiento y justificar sus conclusiones.

En síntesis, un Sistema Experto puede almacenar el conocimiento de expertos para un campo de especialidad determinada, y muy estrechamente delimitada, y solucionar un problema mediante la deducción lógica.

Los Sistemas Expertos basan su rendimiento en la cantidad y calidad del conocimiento de un dominio específico y no tanto en las técnicas de solución de problemas.

3.2.2 Clasificación de los Sistemas Expertos

La clasificación de los Sistemas Expertos está totalmente relacionada con el tipo de problema que se intenta solucionar.

Se acostumbra distinguir diez tipos de problemas que se han encarado con sistemas expertos: interpretación, predicción, diagnóstico, diseño, planea miento, monitoreo, tratamiento, reparación, capacitación y control.

Los sistemas de interpretación infieren la descripción de una situación a partir de datos observables. Esta categoría comprende sistemas de comprensión de lengua hablada, análisis de señales, interpretación de señales, análisis de estructuras químicas, y otros.

Los sistemas de predicción infieren consecuencias de situaciones dadas. Esta categoría incluye pronósticos meteorológicos, predicciones demográficas, estimaciones de cosechas, pronósticos militares.

Los sistemas de diagnóstico infieren funcionamientos incorrectos a partir de los datos.

Esta categoría incluye diagnósticos médicos, veterinarios, electrónicos, mecánicos y de software. Usualmente, relacionan síntomas (irregularidades en el

comportamiento) con sus posibles causas. Existen al menos dos formas muy distintas de encararlos: a través de asociaciones empíricas (heurísticas), o a través de la simulación de las posibles fallas en el diseño o, la implementación o los componentes para generar funcionamientos incorrectos consistentes con las observaciones.

Los sistemas de diseño o desarrollan configuraciones de objetos que satisfacen las restricciones del problema. Estos problemas incluyen diseño o de circuitos y de construcciones, así como confección de presupuestos. Muchos sistemas de diseño o intentan minimizar una función objetivo que mide costos, u otras características indeseables.

Los sistemas de planeamiento diseñan acciones. Están orientados a programación automática, robótica, preparación de proyectos, experimentos, y acciones militares.

Utilizan modelos de la conducta de los agentes, para inferir el efecto de las actividades planeadas.

Los sistemas de monitoreo comparan las observaciones con características cruciales para el éxito de los planes que se están ejecutando. Reconocen dos tipos de problemas: la violación de una condición que pone en riesgo el plan, o un efecto potencial del plan que contradice las restricciones del problema. Existen sistemas de monitoreo asistidos por computadora para plantas de energía nuclear, tránsito aéreo, sistemas médicos, y otros, pero los sistemas expertos de este tipo recién están saliendo de los laboratorios.

Los sistemas de tratamiento prescriben remedio para los funcionamientos incorrectos.

Estos sistemas utilizan capacidades de planeamiento, diseño y predicción para crear recomendaciones para corregir un problema diagnosticado.

Los sistemas de reparación desarrollan y ejecutan planes para administrar un remedio para un problema diagnosticado. Utilizan capacidades de tratamiento, planeamiento y ejecución. Los sistemas expertos recién comienzan a ingresar en este

campo.

Los sistemas de capacitación diagnostican y tratan la conducta de los estudiantes.

Estos sistemas comienzan construyendo una descripción hipotética del conocimiento del alumno para interpretar su comportamiento. Después diagnostican las debilidades de éste conocimiento, e identifican un remedio apropiado. Finalmente, planifican y ejecutan una interacción con el alumno para resolver el problema.

Los sistemas expertos de control dirigen adaptativamente todo el comportamiento de un sistema. Para ello, interpretan la situación actual, predicen la futura, diagnostican las causas de problemas anticipados, formulan planes para resolverlos, y monitorean su ejecución para garantizar el éxito. Corresponden a este tipo los problemas de tránsito aéreo, administración de empresas, dirección de batallas, y otros. Este ser sin duda un campo de un gran desarrollo para los sistemas expertos, porque este tipo de problemas difícilmente encuentra solución con metodologías más convencionales.

3.2.3 Por qué utilizar sistemas expertos

- Con la ayuda de un Sistema Experto, personas con poca experiencia pueden resolver problemas que requieren un "conocimiento formal especializado".
- Los Sistemas Expertos pueden obtener conclusiones y resolver problemas de forma más rápida que los expertos humanos.
- Los Sistemas Expertos razonan pero en base a un conocimiento adquirido y no tienen sitio para la subjetividad.
- Se ha comprobado que los Sistemas Expertos tienen al menos, la misma competencia que un especialista humano.
- El uso de Sistemas Expertos es especialmente recomendado en las siguientes situaciones:

- Cuando los expertos humanos en una determinada materia son escasos.
- En situaciones complejas, donde la subjetividad humana puede llevar a conclusiones erróneas.
- Cuando es muy elevado el volumen de datos que ha de considerarse para obtener una conclusión.

3.3 Electromiografía

Consiste en el registro de la actividad eléctrica del músculo, y se realiza mediante la inserción de un electrodo con forma de aguja en el mismo, con el fin de registrar su actividad eléctrica. El electromiograma no se lleva a cabo de una forma estándar, sino que se diseña en cada caso en función de la historia clínica y la exploración neurológica.

La electromiografía tiene un papel esencial en la evaluación de los pacientes con trastornos del sistema neuromuscular, que incluye el sistema nervioso periférico (nervios que salen de la médula espinal para dirigirse a sus órganos correspondientes, la unión del músculo y del nervio, y los denominados "músculos esqueléticos"), aunque también puede ser útil en el estudio de los pacientes con trastornos del sistema nervioso central (cerebro y médula espinal).

3.3.1 Objetivo del examen

El objetivo principal del examen es la localización de una lesión neuromuscular en casos de duda. Antes de iniciar el electromiograma se debe efectuar una correcta valoración clínica para poder planificar la exploración adecuada para cada paciente, con objeto de acortar el tiempo de exploración y minimizar la posible incomodidad al paciente (aunque, en general, es una prueba que se tolera bastante bien).

Una vez realizadas la historia clínica y la exploración neurológica, el

electromiografista establece una hipótesis sobre la localización de la lesión. A continuación, y mediante la electromiografía, confirma el lugar de la lesión.

Otros objetivos del examen electromiográfico son evaluar el grado de afectación y el curso evolutivo del trastorno. Es importante señalar que el examen electromiográfico no permite establecer diagnósticos específicos, sino que es la correlación de los datos clínicos junto con los aportados por otras pruebas y estudios (incluyendo el electromiograma) la que establece el diagnóstico definitivo.

En cuanto a las indicaciones propiamente dichas, son:

- Diferenciación entre debilidad del sistema nervioso central y periférico. En la mayoría de los casos los síntomas son suficientes para establecer la diferencia. Sin embargo, en casos dudosos, el electromiograma puede ser crucial para confirmar la impresión clínica.
- Diferenciación entre debilidad de origen muscular o nerviosa. Es importante remarcar que en ocasiones dicha diferenciación puede resultar difícil, por la coexistencia de alteraciones de ambos tipos, o por dificultades en la interpretación.
- Determinación del grado de afectación de los nervios. Esto es importante de cara a la toma de decisiones respecto al tratamiento.
- Caracterizar los trastornos de la unión neuromuscular (es la zona donde se unen el nervio y el músculo) y diferenciar así procesos que afectan a este nivel, como la "miastenia gravis").
- Diferenciación entre calambre y contractura.

3.3.2 Preparación para el examen

Generalmente, no se requiere preparación especial. Evite el uso de cualquier crema o loción el día del examen.

La temperatura corporal puede afectar los resultados de este examen. Si hace

mucho frío afuera, espere en un cuarto tibio por un rato antes de que se lleve a cabo el examen.

3.3.3 Forma en que se realiza el examen

Una aguja con un electrodo se introduce por la piel, en el músculo. La actividad eléctrica detectada por el electrodo se despliega en un osciloscopio (y puede ser oída a través de una corneta). Debido a que los músculos están aislados y por lo general son unidades grandes, cada electrodo suministra sólo una imagen promedio de la actividad del músculo seleccionado. Puede ser necesario colocar varios electrodos en diferentes sitios para obtener una evaluación precisa.

Después de colocar los electrodos, se le puede pedir al paciente que contraiga el músculo (por ejemplo, doblando el brazo). La presencia, tamaño y forma de la onda producida por el osciloscopio (el potencial de acción) suministra información sobre la capacidad del músculo para responder al estímulo

nervioso. Cada fibra muscular que se contrae producirá un potencial de acción, y su tamaño afectará el ritmo (la frecuencia con que ocurre el potencial de acción) y el tamaño (amplitud) del potencial de acción.

Generalmente, se lleva a cabo un examen de la velocidad de conducción nerviosa junto con una electromiografía.

3.3.4 Riesgos del examen electromiográfico

Es importante conocer la existencia de ciertos riesgos que conlleva el examen. Uno de los puntos más importantes es el control de la infección: el contagio puede producirse al paciente por parte del material que se utiliza, o bien al personal por parte del paciente. Hoy día esto es muy raro debido a la adopción de medidas preventivas obligatorias en los hospitales.

Otro de los puntos a tener en cuenta se refiere a los pacientes con alteraciones de la coagulación, o los que siguen tratamiento que la modifican. En estos casos, debe regularse individualmente, confrontando los beneficios del estudio con la posibilidad de que se produzca una hemorragia intramuscular debido a la inserción del electrodo-aguja.

Finalmente nos referimos a aquellos pacientes a los que hay que realizar un estudio electromiográfico en los músculos cervicales, torácicos o abdominales, en los cuales, debido a la proximidad del pulmón y del abdomen, la aguja podría penetrar dichas estructuras, con el riesgo de dañarlas. Sin embargo, el entrenamiento del personal que realiza estas pruebas hace esto prácticamente imposible.

3.3.5 Valores normales

Normalmente hay muy poca actividad eléctrica en un músculo en reposo. Introducir las agujas puede causar alguna actividad eléctrica, pero una vez que los músculos se calman, se debe detectar muy poca actividad de este tipo.

Cuando se flexiona un músculo, la actividad comienza a aparecer. A medida que

se contrae más el músculo, la actividad eléctrica se incrementa y se puede observar un patrón. Este patrón le ayuda al médico a determinar si el músculo está respondiendo como se debe.

3.4 Velocidad de conducción nerviosa (VCN)

La velocidad de conducción nerviosa es una prueba de la velocidad de las señales eléctricas a través de un nervio.

3.4.1 Preparación para el examen

Es necesario mantener la temperatura corporal normal, dado que la temperatura baja retarda la conducción nerviosa.

Coméntele al médico si usted tiene un desfibrilador cardíaco o un marcapasos, ya que puede ser necesario tomar precauciones.

3.4.2 Forma en que se realiza el examen

Se colocan parches, llamados electrodos de superficie, semejantes a los usados para un ECG, sobre la piel por encima de los nervios en diversos lugares. Cada parche emite un impulso eléctrico muy leve que estimula el nervio.

La actividad eléctrica resultante del nervio es registrada por los otros electrodos. La distancia entre los electrodos y el tiempo que le toma a los impulsos eléctricos viajar entre los electrodos se utiliza para determinar la velocidad de las señales nerviosas.

Con frecuencia, una electromiografía (registro a partir de agujas colocadas dentro de los músculos) se realiza al mismo tiempo que este examen

3.4.3 Razones por la que se realiza el examen

Este examen se utiliza para diagnosticar daño o destrucción del nervio.

Ocasionalmente, el examen se puede utilizar para evaluar trastornos de nervios o músculos, incluyendo miopatía, síndrome de Lambert-Eaton o miastenia grave.

3.4.4 Valores normales

La velocidad de conducción nerviosa se relaciona con el diámetro del nervio y con su grado normal de mielinización, que es la presencia de vaina de mielina en el axón. Los bebés recién nacidos tienen valores que equivalen aproximadamente a la mitad del valor de los adultos, y los valores para los adultos se alcanzan normalmente a la edad de 3 a 4 años.

Nota: los rangos de los valores normales pueden variar ligeramente entre diferentes laboratorios. Se deberá hablar con un médico acerca del significado de los resultados específicos de un examen.

3.5 Drools

Es un sistema de gestión de reglas de negocio (BRMS, por las siglas en inglés de business rule management system) con un motor de reglas basado en inferencia de encadenamiento hacia adelante (forward chaining), más correctamente conocido como sistema de reglas de producción, usando una implementación avanzada del algoritmo Rete.

Es software libre distribuido según los términos de la licencia Apache.

Drools soporta el estándar JSR-94 para su motor de reglas de negocio y framework de empresa para construcción, mantenimiento y refuerzo de políticas de empresa en una organización, aplicación o servicio. Drools usa JCR (JackRabbit) para gestionar el repositorio de reglas, y el estándar JAAS para la autorización y autenticación

También existe un plug-in de Eclipse para facilitar el desarrollo con esta herramienta.

Para especificar las reglas Drools utiliza el lenguaje de reglas de drools (DRL)

para especificar las condiciones, acciones y funciones de las mismas, las cuales se puede expresar con distintos lenguajes, como Java y MVEL. Entonces las reglas son guardadas en archivos de texto con la extensión drl.

Entre las opciones para definir las reglas aparte del DRL se pueden especificar lenguajes específicos de dominio (dsl), los cuales se asocian a un drl, y también existe la opción de especificar las reglas en una planilla de cálculo, como Excel.

3.6 PostgreSQL

PostgreSQL Server es un servidor de base de datos relacional de código abierto. Es desarrollado y mantenido por una comunidad de desarrolladores voluntarios. Al ser un sistema de código abierto bajo la licencia GPL 3 permite utilizarlo en forma gratuita para fines comerciales, inclusive permite modificar su código fuente.

En los últimos años PostgreSQL ha ido ganando terreno y hoy es considerado uno de los mejores motores de base de datos, inclusive siendo comparado con soluciones pagas tipo ORACLE o IBM Informix.

4. Desarrollo

4.1 Cronograma del sistema

El cronograma estipulado para el desarrollo del sistema experto se encuentra en el archivo Cronograma del sistema.mpp colocado como anexo.

En el proyecto actual no se desarrollan todas las fases del cronograma ya que el propósito no es obtener el producto funcionando e implementado, sino hacer el análisis para demostrar la factibilidad de realizar un sistema de diagnóstico electromuscular a través de un sistema experto.

Desde la Fase 2 hasta la Fase 4 es un proceso cíclico, por lo que se han estimado 10 iteraciones para completar el proyecto, en cada iteración, el tiempo de cada tarea va variando, mientras más nos acercamos a la ultima iteración dejamos de adquirir conocimientos (Fase 2) y comenzamos a darle mas prioridad al diseño y prueba del sistema.

La Fase 5, de documentación, se realiza durante todo el desarrollo del proyecto, es por eso que aparecen 285 días, pero lo cierto es que se le dedica sólo unos minutos algunos días y otros varias horas, dependiendo de las tareas que se vayan realizando y deban documentarse.

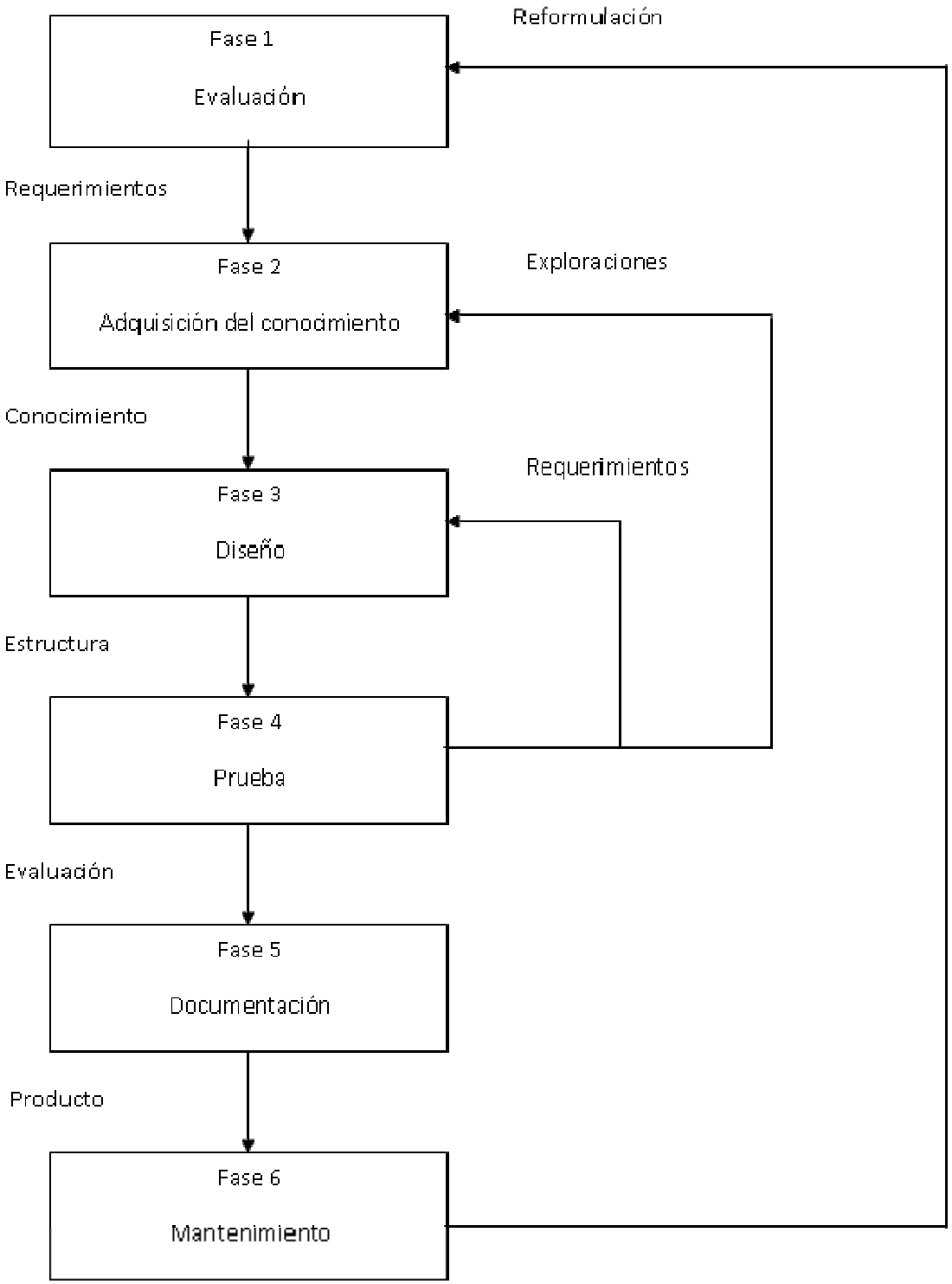
La finalización del proyecto se estima en 313 días, es decir, aproximadamente unos 16 meses.

4.2 Metodología para el desarrollo del sistema experto.

Al igual que para desarrollar un sistema de información convencional existen varias metodologías de desarrollo como la Ingeniería de la Información, tendencias estructuradas y orientadas a objetos, así existen varias metodologías para desarrollar un

sistema experto. Como ya sabemos el área de sistemas expertos es relativamente joven por lo cual no se dispone de una única metodología sino que cada autor propone una de acuerdo a su forma de desarrollo. Sin embargo existen algunas que han tenido éxito más que otras lo cual ha llevado a su mayor difusión.

Para este caso propongo trabajar con la Metodología de Ingeniería del Conocimiento de John Durkin, de la cual se muestra una breve descripción a continuación:


Reformulación
Fase 1
Evaluación
Requerimientos
Fase 2
Adquisición del conocimiento
Exploraciones
Conocimiento
Requerimientos
Fase 3
Diseño
Estructura
Fase 4
Prueba
Evaluación
Fase 5
Documentación
Producto
Fase 6
Mantenimiento

FASE 1: EVALUACIÓN

1.1 Motivación para el Esfuerzo.

1.2 Identificar problemas candidatos.

1.3 Estudio de viabilidad.

1.4 Análisis de Costo/Beneficio.

1.5 Seleccionar el mejor proyecto.

1.6 Escribir el proyecto propuesto.

FASE 2: ADQUISICIÓN DEL CONOCIMIENTO

2.1 Recolección del conocimiento.

2.2 Interpretación.

2.3 Análisis.

2.4 Diseño de métodos para recolectar conocimiento adicional.

FASE 3: DISEÑO

3.1 Seleccionar Técnica de Representación del Conocimiento.

3.2 Seleccionar Técnica de Control.

3.3 Seleccionar Software de Desarrollo de Sistema Experto.

3.4 Desarrollo de Prototipo.

3.5 Desarrollo de Interfase.

3.6 Desarrollo del Producto.

FASE 4: PRUEBAS

4.1 Validación del Sistema.

4.2 Evaluación de la Prueba/Evaluación.

FASE 5: DOCUMENTACIÓN

5.1 Relación de temas que deben ser documentados.

5.2 Organización de la documentación.

5.3 Documentación Impresa.

5.4 Documentación en hipertexto.

5.5 Reporte Final

FASE 6: MANTENIMIENTO

6.1 Modificaciones probables del sistema.

6.2 Responsables de mantenimiento.

6.3 Interfaces de documentación del mantenimiento.

FASE 1: EVALUACION

1.1 Determinar Motivación para el Esfuerzo

Consiste en determinar ¿Por qué está la organización motivada para seguir Sistemas Expertos? Algunas organizaciones están mirando resolver un problema particular mientras que otras están interesadas en encontrar que puede hacer la tecnología por ellos.

De acuerdo a lo antes mencionado existen dos posiciones que puede asumir una organización al incursionar en la tecnología de Sistemas Expertos:

Conducida por el Problema: ocurre cuando la organización trata de resolver un problema que ya se ha identificado.

Conducida por la Solución: en algunos casos una organización es motivada para explorar una tecnología nueva por un interés general o curiosidad.

1.2 Identificar problemas candidatos

Esta tarea solo ocurre cuando la organización es conducida por la solución. Este paso es hecho antes que la viabilidad formal y estudios costo/beneficio y es llamado PRE-DETERMINACION.

Formando la Lista

Cuando se forma la lista de problemas candidatos se debería buscar la ayuda de individuos dentro de la organización. Un buen lugar para observar dentro de la organización es el nivel medio. Estos individuos tienen una visión global de operaciones

y conocimiento acerca de problemas de cada día. Su visión es valiosa porque se descubre áreas donde la aplicación de un sistema experto tiene el potencial para proporcionar valor real a la organización.

Demostración de la Tecnología

Si la organización está explorando la aplicación de Sistemas Expertos, entonces se debería ver al proyecto como una demostración de la tecnología. Por lo tanto, un problema pequeño y relativamente simple es más preferible que un complejo. Por pequeño, quiere decir que el alcance del problema no cubre un gran número de problemas complejos. Por simple, quiere decir que el problema parece a primera vista de ser solucionable. Como guía para solucionar el problema considerar lo que otros han hecho en el pasado.

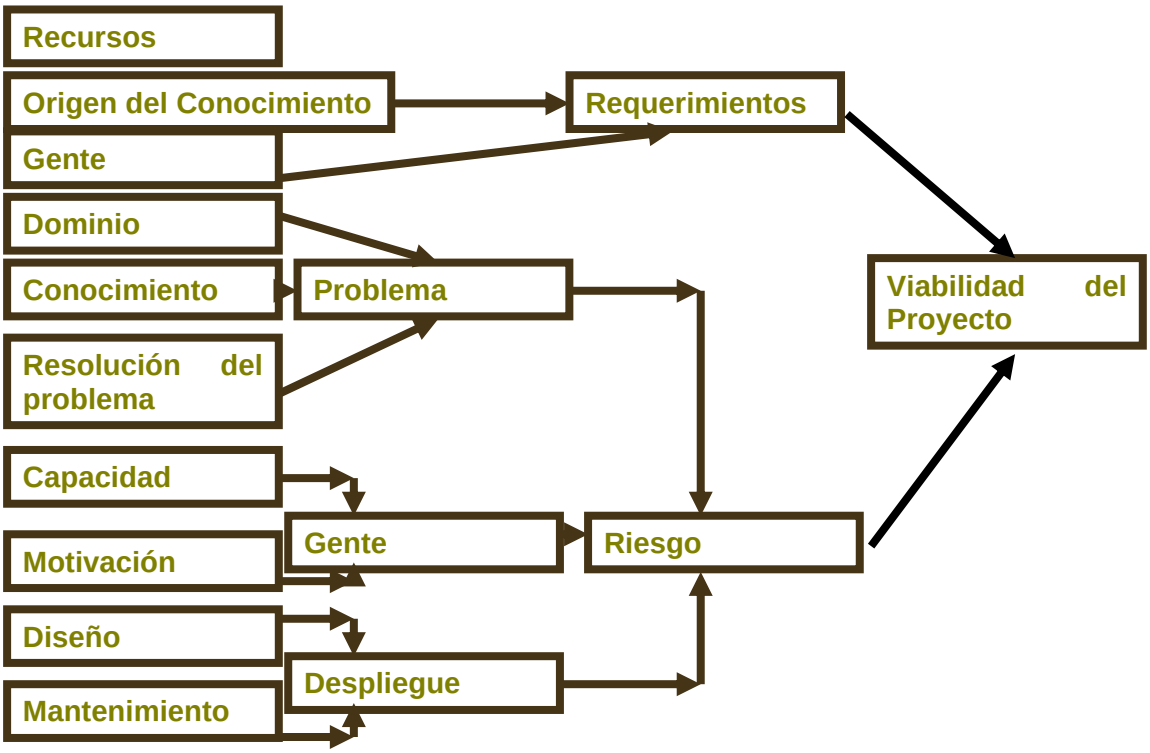
Sugerencias para escoger un buen problema

Para las organizaciones buscando explorar la tecnología considere los siguientes puntos:

- Toma de decisión humana.
- Conocimiento heurístico.
- Pequeño.
- Simple.
- Éxito probable.
- Algún valor.

1.3 Estudio de Viabilidad

Gráficamente el esquema de viabilidad se representa así:



En esta tarea lo primordial es tratar de determinar si el proyecto tendrá éxito. Se consideran dos puntos a evaluar:

Primero: Una lista de ítems que debería reunir el proyecto es verificado. Estos ítems incluyen los recursos propios, un recurso de conocimiento y personal del proyecto.

La siguiente lista de requerimientos debería ser verificada primero cuando se considera un problema para una aplicación de Sistema Experto:

- Disponibilidad de conocimiento para la solución del problema(experto)
- Disponibilidad de un Ingeniero del Conocimiento.
- La solución del problema puede ser validada.
- Disponibilidad de fondos.
- Disponibilidad de software de desarrollo de sistema.
- Disponibilidad de facilidades de computador.

Segundo: Considerar asuntos que son importantes para el éxito del proyecto, pero los cuales son subjetivos de naturaleza y requieren algún juicio para determinar. Ellos incluyen características del problema, características de la gente involucrada del proyecto y asuntos de despliegue. Aún cuando un proyecto reúne los requerimientos verificados hay otros asuntos que pueden prevenir el completo éxito del proyecto. Un proyecto de sistema experto puede fallar por razones que caen dentro de las tres categorías: problema, gente y despliegue.

Viabilidad del Problema

Los asuntos de viabilidad incluyen características de dominio, conocimiento y tareas de solución del problema.

Comprende:

- Conocimiento experto necesitado.
- Los pasos de solución son definibles.
- Conocimiento simbólico usado.
- Heurísticas usadas.
- El problema es solucionable.
- Existen problemas exitosos.
- El problema es bien enfocado.
- El problema es estable
- Conocimiento incompleto o incierto utilizado.
- Solución más una recomendación.

Asuntos de viabilidad de la gente

La capacidad y la motivación de la gente involucrada en el proyecto son asuntos importantes para considerar cuando se juzga la viabilidad del proyecto .Los principales actores de un proyecto de sistema experto son: experto de dominio, ingeniero de

conocimiento, usuario final, y administración .Determinar su impacto en la viabilidad del proyecto es un desafió debido a las complejidades de naturaleza humana. Se necesita considerar sus deseos, miedos, y emociones para juzgar si ellos efectivamente contribuirán el proyecto. Las características principales que deben tener cada persona involucrada en un proyecto de sistema experto son:

Experto

- Puede comunicar el conocimiento.
- El experto puede dedicar tiempo.
- El experto es cooperativo, no hostil o escéptico del proyecto.

Ingeniero de conocimiento

- El ingeniero de conocimiento tiene buenas habilidades de comunicación.
- El ingeniero del conocimiento puede relacionar el problema al software.
- El ingeniero de conocimiento tiene destrezas de programación de sistema experto.
- El ingeniero del conocimiento puede dedicar el tiempo.

Usuario final

- El usuario final puede dedicar tiempo.
- El usuario final es receptivo al cambio.
- El usuario final es cooperativo.

Gerencia

- La gerencia apoya al proyecto.
- La gerencia es receptiva al cambio.
- La gerencia no es escéptica.
- La gerencia tiene expectativas razonables.

- La gerencia entiende objetivos.

Asuntos de viabilidad de Despliegue

Se debe considerar:

- El sistema puede ser introducido fácilmente.
- El sistema puede ser mantenido.
- El sistema puede ser integrado con recursos existentes.
- Entrenamiento disponible.
- El sistema no tiene una ruta crítica.

Determinación de viabilidad

Un esfuerzo por determinar la viabilidad de un sistema experto fue proporcionado por Beckman (1991) el cual formo una lista de temas para considerar, luego asignó a cada uno un número que reflejaba su importancia relativa. Esta lista de verificación de peso es luego comparada a algún problema candidato, y si el problema encuentra un tema, recibe los puntos predescritos del tema. La suma de todos los puntos es luego usada para atribuir un porcentaje de la viabilidad del proyecto. Un ejemplo de esta forma de determinar la viabilidad de muestra a continuación:

Asuntos de viabilidad del problema

Asuntos de viabilidad del problema		
Peso	Asunto	Puntaje

1	Conocimiento experto necesitado	
2	Los pasos de solución de problema son definibles	
1	Conocimiento simbólico usado	
1	Heurísticas usadas	
2	El problema es solucionable	
2	Existen sistemas exitosos	
2	El problema es bien enfocado	
1	El problema es razonablemente complejo	
1	El problema es estable	
1	Conocimiento incompleto o incierto utilizado	
1	No determinístico	
1	Solución más una recomendación	
16	Puntos Totales	Puntaje Total
	Viabilidad = Puntaje total/Puntos totales	

Una deficiencia con este método es que muchos temas son subjetivos y son difíciles para responder de un modo sí o no. Considere por ejemplo el tema del ámbito del problema. Para un problema dado nosotros podríamos ser capaces de comentar sobre este tema, pero puede ser demasiado limitante para ser restringido a una respuesta de sí o no. Esta limitación puede también guiar a responder errores que produce una falsa figura de determinación de viabilidad.

Un diferente tipo de estrategia fue desarrollado que corrige este problema. Como la técnica anterior, empieza por formar una lista de temas importantes para considerar. Cada tema es luego asignado un peso (entre 0 y 10) que refleja la importancia de cada tema durante la evaluación de un proyecto dado, los números (entre 0 y 10) son atribuidos a cada tema que refleja el grado de creencia en el tema. Este valor es luego multiplicado por el valor del tema para establecer un puntaje por el tema. Todos los puntajes son luego añadidos y divididos por la suma de los pesos del tema. Este número es limitados entre 0 y 10, y proporciona una estimación de determinación de viabilidad

del proyecto.

Los valores de “peso” son resultados de la experiencia de consulta de Durkin sobre los esfuerzos de determinación de proyectos anteriores.

A continuación aplicaré este método para la determinación de la viabilidad para el sistema propuesto:

Formulario de determinación de viabilidad del problema

ASUNTOS DE VIABILIDAD DEL PROBLEMA			
PUNTAJE =	PESO *	VALOR	ASUNTO
56	7	8	Conocimiento experto necesitado
81	9	9	Los pasos de solución de problema son definibles
49	7	7	Conocimiento simbólico usado
64	8	8	Heurísticas usadas
100	10	10	El problema es solucionable
56	8	7	Existen sistemas exitosos
72	9	8	El problema es bien enfocado
48	6	8	El problema es razonablemente complejo
56	7	8	El problema es estable
63	9	7	Conocimiento incompleto o incierto utilizado
35	5	7	No determinístico
54	6	9	Solución más de una recomendación
734	91	96	
PUNTAJE TOTAL	PESO TOTAL	VIABILIDAD DEL PROBLEMA =	PUNTAJE TOTAL / PESO TOTAL
			8.06 = 734/91

Formulario de determinación de viabilidad de personal.

ASUNTOS DE VIABILIDAD DE PERSONAL			
PUNTAJE =	PESO *	VALOR	ASUNTO
			EXPERTO DE DOMINIO

63 63 49	7 9 7	9 7 7	El experto puede comunicar el conocimiento El experto puede dedicar tiempo El experto es cooperativo
175	23	23	
PUNTAJE TOTAL	PESO TOTAL	VIABILIDAD DEL EXPERTO = PUNTAJE TOTAL / PESO TOTAL	
		7.6 = 175/23	
			INGENIERO DEL CONOCIMIENTO
72 64 72 81	8 8 9 9	9 8 8 9	Buenas habilidades de comunicación Puede relacionar el problema al software Tiene destrezas de programación de sistema experto Puede dedicar tiempo
289	34	34	
PUNTAJE TOTAL	PESO TOTAL	VIABILIDAD DEL INGENIERO DEL CONOCIMIENTO = PUNTAJE TOTAL / PESO TOTAL	
		8.5 = 289/34	
			USUARIO FINAL
48 63 56	6 7 7	8 9 8	El usuario final puede dedicar tiempo El usuario final es receptivo al cambio El usuario final es cooperativo
167	20	25	
PUNTAJE TOTAL	PESO TOTAL	VIABILIDAD DEL USUARIO FINAL = PUNTAJE TOTAL / PESO TOTAL	
		167/20 = 8.35	

			GERENCIA
81	9	9	La gerencia apoya al proyecto
56	7	8	La gerencia es receptiva al cambio
56	7	8	La gerencia no es escéptica
42	6	7	La gerencia tiene expectativas razonables
72	8	9	La gerencia entiende objetivos
307	37	41	
PUNTAJE TOTAL	PESO TOTAL	VIABILIDAD LA GERENCIA =	PUNTAJE TOTAL / PESO TOTAL
			8.29 = 307/37

Formulario de determinación de viabilidad de despliegue

ASUNTOS DE VIABILIDAD DEL DESPLIEGUE			
PUNTAJE =	PESO *	VALOR	ASUNTO
56	7	8	El sistema puede ser introducido fácilmente
54	9	6	El sistema puede ser mantenido
42	7	6	El sistema no tiene una ruta critica
72	9	8	El sistema puede ser integrado con recursos existentes
56	7	8	Entrenamiento disponible
280	39		
PUNTAJE TOTAL	PESO TOTAL	VIABILIDAD DEL DESPLIEGUE =	PUNTAJE TOTAL / PESO TOTAL
			7.17 = 280/39

Luego de realizar el estudio de viabilidad para cada categoría resulto en los siguientes puntajes:

CATEGORIA	PUNTAJE TOTAL	PESO TOTAL
Problema	734	91
Gente	900	114
Despliegue	280	39
	1914	244

VIABILIDAD DEL PROYECTO = 1914/244 = 7.84

Se puede usar estas formas para establecer los valores de viabilidad para los proyectos candidatos, y escoger esos con valores más altos para considerarlos después. Para los proyectos con bajos valores globales, este método también proporciona una visión dentro de que área es deficiente, como asuntos de problema, asuntos de gente, etc.

1.4 Análisis Costo/ Beneficio

El próximo paso es determinar el esperado pago-justificación para el proyecto.

Para la mayoría de los proyectos este es usualmente es medido en un análisis costo beneficio. La organización desea evidencia tangible que muestre que la inversión de tiempo y dinero es justificado. Bajo las mejores condiciones este puede ser una tarea difícil. Cuando el proyecto involucra una tecnología nueva como sistemas expertos, la tarea encuentra incertidumbre adicional.

Costo del Proyecto

Los costos principales del proyecto son establecidos por los gastos de trabajo y software. La cuenta de gastos de trabajo para el tiempo gastado en el proyecto son por el ingeniero del conocimiento, el experto de dominio y el usuario final.

La opción del software de desarrollo del sistema experto está basada en la naturaleza del problema y las facilidades del computador de la organización.

Los asuntos de beneficio

El beneficio de desarrollar un sistema experto puede ser medido en una de las cuatro maneras: productividad mejorada, costos bajos, calidad mejorada o un asunto muy tangible pero importante—imagen mejorada.

1. Productividad mejorada
    - Mejores Decisiones
    - Decisiones más rápidas
    - Propaga especialización
2. Costos más bajos
    - Reduce costos de trabajo
    - Mejora uso de material
3. Calidad mejorada
    - Producto Superior
    - Servicios superiores
    - Proporciona entrenamiento
4. Imagen mejorada
    - Innovador

1.5: Seleccionar el mejor proyecto

Para cada problema inicialmente seleccionado para el esfuerzo de determinación, ahora se tiene la información sobre su viabilidad y su conveniencia. La próxima tarea es seleccionar uno para seguir un proyecto de sistema experto.

El cuadro que usted ahora tiene de cada posible proyecto es ambos cualitativo y cuantitativo. El estudio de viabilidad proporcionó un número que refleja la estimación del proyecto de viabilidad global. Este número es principalmente el valor de comparar varios proyectos. El estudio de costo/beneficio también proporcionó números. El costo del proyecto es usualmente fácil de estimar, y en algunos casos, se puede aproximar los

ahorros esperados o ganancias de la organización. Se debería también tener un sentido del impacto que el proyecto puede tener en establecer sistemas expertos dentro de la organización.

Conociendo la motivación de la organización es de ayuda sobre esta tarea. Si ellos son conducidos por el problema, se debería mostrar que el proyecto es viable y que los beneficios esperados excedieron al costo del proyecto. Aun cuando la organización está explorando la tecnología (conducida por la solución) – aparentemente una situación cómoda. —usted debería aún proporcionar alguna justificación para el esfuerzo. Estas organizaciones son usualmente más tolerantes de los beneficios de corto plazo, pero ellos esperan que el proyecto engendre beneficios a largo plazo.

1.6 Escribir el proyecto propuesto

Siguiendo la selección de un buen problema, se puede necesitar escribir un proyecto propuesto que documente los esfuerzos esperados. Esta propuesta debería documentar que es para ser hecho, porque el proyecto es importante, y como se ejecutará el esfuerzo. En la discusión de cada uno de esto puntos, el propuesto debería ser breve y al punto.

FASE 2: ADQUISICION DEL CONOCIMIENTO

Siguiendo las fases, la siguiente tarea es la adquisición del conocimiento. Esta tarea es el desafío más difícil en el desarrollo de un sistema experto.

Procesos de Adquisición de Conocimiento

La adquisición del conocimiento es inherentemente un proceso cíclico. Sigue las tareas de recolección del conocimiento, su interpretación y análisis, y el diseño de métodos para recolectar conocimiento adicional.

La recolección es la tarea de adquirir conocimiento del experto. Este esfuerzo requiere entrenamiento en técnicas de entrevistas. Además requiere buenas habilidades de

comunicación interpersonal y la habilidad para obtener la cooperación del experto.

La interpretación de la información recolectada envuelve la identificación de piezas clave de conocimiento, como conceptos, reglas, estrategias, etc.

El análisis envuelve el estudio de las piezas clave del conocimiento destapado durante la tarea de interpretación. Este esfuerzo proporciona la visión de formar las teorías en la organización del conocimiento y estrategias de solución de problemas.

El diseño es la tarea de preparación para el siguiente encuentro con el experto. Siguiendo la realización de las tareas anteriores, se forma una nueva comprensión del problema. Este esfuerzo puede haber expuesto nuevos conceptos que necesitan exploración extensa. Las técnicas de extracción del conocimiento son entonces escogidas para obtener esta información durante la próxima reunión.

Problemas con adquisición de Conocimiento

Existen muchos problemas con la adquisición del conocimiento que hacen de esta una tarea difícil. Muchos de estos problemas pueden remontarse a la dificultad en extraer conocimiento del experto.

Sin premeditación de conocimiento.- A través de la experiencia resolviendo un problema, un experto a menudo compila el conocimiento de la solución del problema en una forma compacta, la cual permite una solución eficiente del problema. Si el experto es preguntado para describir su método de solución de problemas, él a menudo hará saltos mentales sobre problemas importantes.

Incapacidad para verbalizar el conocimiento.- Muchas tareas son difíciles de verbalizar debido a que ellas fueron aprendidas mirando a otros individuos ejecutando estas tareas. Los esfuerzos de la labor manual representan este tipo de tareas.

Proveer conocimiento irrelevante.- Muchas sesiones de extracción pueden ser sostenidas con el experto durante el proyecto. Después de un tiempo, la cantidad de

información recolectada puede estar agobiando. Para hacer la materia mucho peor, mucha de esta información puede ser irrelevante para el proyecto. La tarea es filtrar a través de toda esta información y escoger sólo los problemas importantes.

Proveer conocimiento incompleto.- Un experto a menudo puede proporcionar una descripción incompleta de sus procesos mentales. Si el problema es una simple omisión, la situación puede ser fácilmente corregida. Sin embargo, si ocurre porque el experto es inconsciente del conocimiento usado, (problema de compilación) el desafío puede ser mayor.

Proveer conocimiento incorrecto.- Un experto puede proporcionar conocimiento incorrecto porque él no está informado o debido a un simple error durante la introspección. En cualquier caso, esto lleva a un cuerpo incompleto del conocimiento en sistemas expertos.

Proveer conocimiento inconsistente.- El conocimiento proporcionado por el experto puede ser inconsistente con cualquier declaración. Este problema ocurre frecuentemente cuando el experto proporciona una explicación de sus estrategias de resolución de problemas.

Esfuerzo del equipo cooperativo

El éxito del proceso de extracción del conocimiento dependerá grandemente de formar un equipo de individuos que son hábiles y cooperativos. Cada miembro del equipo es responsable de tareas que solapan tareas de otros. Una interacción considerable puede esperarse y esto es importante para nutrir el a veces frágil espíritu de cooperación.

Técnica de entrevista

La técnica de obtención del conocimiento más común utilizada hoy en día en el

diseño de sistemas expertos es el método de la entrevista. Esta técnica envuelve una interacción directa entre el experto y el ingeniero del conocimiento, donde las preguntas son dadas para destapar el conocimiento. Para hacer este productivo esfuerzo, la entrevista debe ser efectivamente manejada.

El manejo de la entrevista propiamente requiere que varios puntos sean dirigidos. Algunos de los básicos relacionan a los artículos como preparar la agenda, horario de la sesión, y preparar una lista de materiales. Otros problemas son más intangibles, pero importantes para el esfuerzo. Saber cómo empezar, conducir y terminar efectivamente la entrevista son consideraciones importantes para adquirir la información deseada y para mantener la cooperación de los miembros del equipo. También es importante saber cómo hacer las preguntas de una manera que proporcionará la información deseada.

Existen diferentes técnicas de entrevistas para ganar tipos ciertos de conocimiento y para evitar algunos problemas típicos asociados con la extracción del conocimiento.

Análisis de Conocimiento

Siguiendo la entrevista, la información recolectada necesita ser analizada. Los objetivos de este esfuerzo son determinar qué fue aprendido y que problemas adicionales debe seguirse.

Normalmente una trascripción es primero hecha de una grabación de la sesión. Esta trascripción es luego revisada para identificar las piezas clave del conocimiento, conceptos, reglas, etc. Estas piezas de conocimiento son luego analizadas para formar teorías en su organización y cómo ellas relacionan a lo que ya es conocido sobre el problema. También se agregan estas piezas de conocimiento a la documentación del proyecto de una manera discutida después en este capítulo.

Un alcance que puede ayudar a analizar el conocimiento recolectado es grabar la información recolectada gráficamente. Las representaciones gráficas en la forma de

mapas de concepto, redes de inferencia, diagramas de flujo y árboles de decisión pueden ser de valor particular.

FASE 3: DISEÑO

Esta tarea comienza con la selección de la técnica de representación del conocimiento y la estrategia de control. Es seguida con la selección de una herramienta de software que reúne mejor las necesidades del problema. Un sistema prototipo pequeño es luego construido para validar el proyecto y para proporcionar una guía para el trabajo futuro. El sistema es entonces extensamente desarrollado y refinado para encontrar los objetivos del proyecto. Este proceso es estructurado de acuerdo a las siguientes tareas:

Tarea 1: Seleccionar Técnica de Representación del Conocimiento

Tarea 2: Seleccionar Técnica de Control

Tarea 3: Seleccionar Software de Desarrollo de Sistema Experto

Tarea 4: Desarrollo de Prototipo

Tarea 5: Desarrollo de Interface

Tarea 6: Desarrollo del Producto

TAREA 1: Seleccionar Técnica de Representación del Conocimiento

Se debe escoger una técnica de representación del conocimiento que mejor muestre la manera en que el experto modela el conocimiento del problema mentalmente. Sin embargo, para razones prácticas, se debe además considerar los recursos y capacidades de la organización.

Un método basado en frames es apropiado si el experto describe el problema referenciando los objetos importantes y sus relaciones, particularmente si el estado de un objeto afecta a otro objeto. Esta situación es encontrada en problemas tipo simulación o algunas donde las relaciones causales son importantes.

Otra señal que un método basado en frame puede ser bien escogido es que el experto considere varios objetos similares cuando resuelve el problema. Un sistema

basado en frame puede razonar sobre objetos similares usando solo unas pocas reglas del modelo de emparejamiento que trabajan a través una clase de objetos. Esto proporciona un método eficaz al codificar los objetos y las reglas.

Un método basado en reglas es conveniente si el experto discute el problema principalmente usando declaraciones tipo IF/THEN.

El método de la inducción es de valor si existen ejemplos pasados del problema. La inducción también es apropiada si no existe ningún experto real en el problema, pero una historia de información del problema esta disponible que puede usarse para derivar los procedimientos de toma de decisión automáticamente.

TAREA 2: Seleccionar Técnicas de Control

El encadenamiento hacia adelante es apropiado si el experto primero recolecta información sobre el problema y luego ve qué puede ser concluido.

El encadenamiento hacia atrás es una buena opción si el experto primero considera alguna conclusión o meta, luego intenta demostrarlo buscando la información de apoyo.

En este caso, el experto está principalmente interesado en demostrar alguna hipótesis o recomendación. También, si el número de metas es mucho menor que la cantidad de posible data, entonces considera un alcance de encadenamiento hacia atrás.

Paradigmas de Resolución de Problemas

Otra manera para que usted pueda ganar la visión en escoger ambos, la técnica de representación de conocimiento y la estrategia de inferencia es revisar lo que otras han hecho en el pasado en esfuerzos similares.

Siguiendo estas mismas ideas, los diseñadores del sistema experto escogen a menudo representación del conocimiento y técnicas de control sobre la base del problema que resuelve el paradigma. Estas opciones confían en los éxitos del pasado.

Se han hecho los esfuerzos pasados para relacionar cada paradigma a varios

características que pueden ser deseables en el diseño del sistema experto (Gevarter 1987, Martin y Ley 1988) .Lo siguiente muestra una aproximación del análisis hecho a proyectos de sistemas expertos realizados anteriormente en un esfuerzo al elaborar cada proyecto que resuelve el problema, inspección a la representación de conocimiento y las técnicas de control que se emplearon. El resultado de este esfuerzo se muestra en el siguiente esquema:

Tipo de problema versus inferencia y Representación de Conocimiento

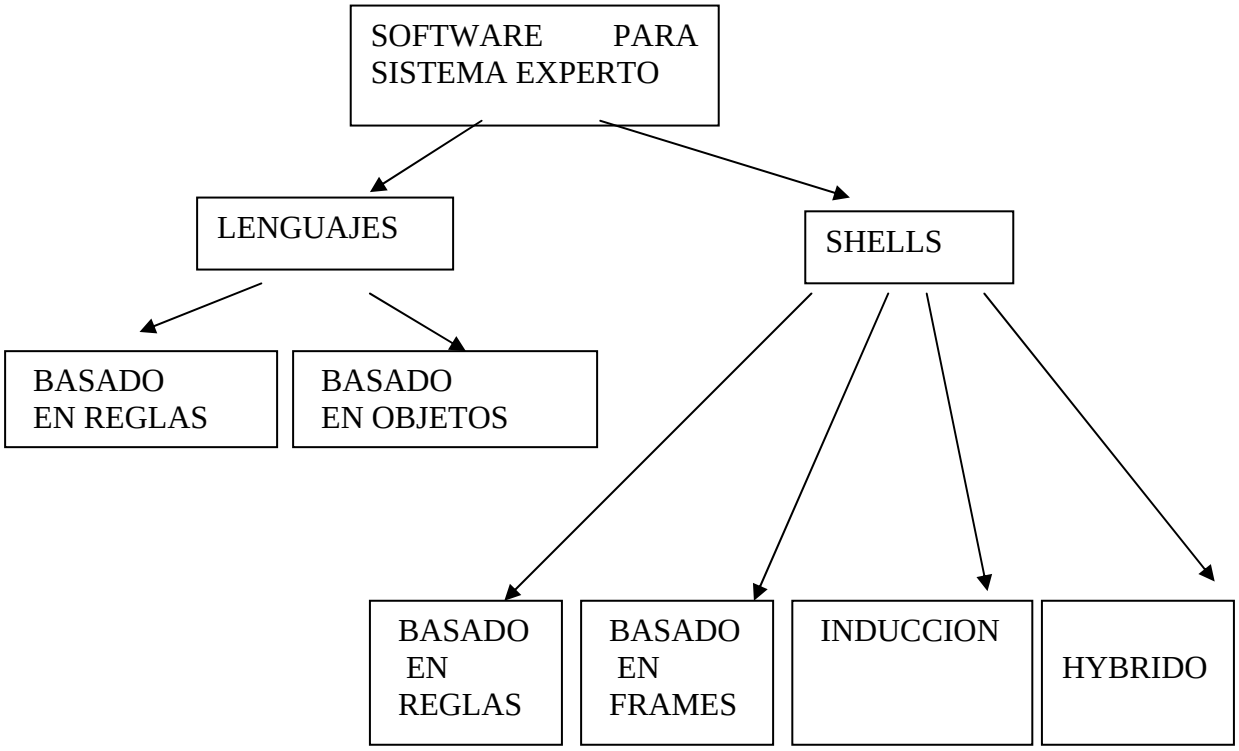
TIPO DE PROBLEMA VERSUS INFERENCIA Y REPRESENTACIÓN DE CONOCIMIENTO					
TIPO DE PROBLEMA	INFERENCIA		REPRESENTACIÓN DEL CONOCIMIENTO		
	HACIA ATRÁS	HACIA ADELANTE	REGLAS	FRAMES	INDUCCIÓN
CONTROL	BAJO	ALTO	ALTO	AVG.	BAJO
DISEÑO	BAJO	ALTO	ALTO	BAJO	BAJO
DIAGNÓSTICO	ALTO	BAJO	ALTO	MEDIO	MEDIO
INSTRUCCIÓN	ALTO	MEDIO	ALTO	MEDIO	BAJO
INTERPRETACIÓN	MEDIO	ALTO	ALTO	BAJO	ALTO

SEGUIMIENTO	BAJO	ALTO	ALTO	MEDIO	BAJO
PLANIFICACIÓN	BAJO	ALTO	ALTO	MEDIO	BAJO
PREDICCION	MEDIO	ALTO	ALTO	BAJO	ALTO
PRESCRIPCIÓN	MEDIO	MEDIO	ALTO	BAJO	BAJO
SELECCIÓN	ALTO	BAJO	ALTO	BAJO	MEDIO.

TAREA 3: Seleccionar Software para el desarrollo del sistema experto

Hay una gran variedad de herramientas de software disponibles para el desarrollo de un sistema experto. Ellos van desde los lenguajes de programación básicos hasta los de desarrollo de alto nivel (SHELLS).

Categorías de software



Importancia de características de software

Las características a considerar en la elección de una herramienta de desarrollo son:

General

- Costo
- El Hardware de la computadora
- Licencia
- Apoyo en la capacitación

Desarrollador de Interfaz

- Codificando el conocimiento.
- Razonamiento inexacto.

- Establecer reglas
- Acceso externo al programa:
- Utilidades de depuración.

Interfaz de usuario

- Preguntas
- Explicaciones
- Gráficos
- Hipertexto

TAREA 4: Desarrollo del Prototipo

Seleccionado el software de acuerdo a los requerimientos del proyecto, el desarrollo del sistema se empieza. La mayoría de proyectos de sistemas expertos empiezan el desarrollo construyendo un prototipo de sistema pequeño. Un prototipo es un modelo del sistema final. Su estructura básica, que representa y procesa el conocimiento del problema, es igual al esperado en el sistema final. Aunque el prototipo es sólo una pequeña versión del sistema final limita la habilidad propiamente si el diseñó envía los propósitos siguientes al servidor.

- Validación de aproximación del sistema experto.
- Confirma opción de técnica de representación de conocimiento y estrategias de control.
- Proporciona una vía de adquisición de conocimiento.

Definir una estrategia global

Para iniciar el diseño del prototipo es necesario definir una estrategia global. Esta búsqueda es una serie de tareas de nivel alto que el sistema necesitará realizar.

Definir Estructura de Conocimiento

Durante el desarrollo del prototipo se debe crear un esquema de trabajo que se acomode los cambios futuros.

Dar validez al Proyecto

En la primera fase, se hacen los esfuerzos para probar la base de conocimiento completa para la lógica y consistencia. La naturaleza exhaustiva de esta prueba sólo es posible al inicio del proyecto cuando la base de conocimiento es pequeña. Esta comprobación destapa las deficiencias en el conocimiento y búsqueda de razonamiento, y valida la opción de la técnica de representación de conocimiento y de desarrollo de software.

La segunda fase es probar es más de una demostración el sistema. Su propósito es quitar algún posible escepticismo por el proyecto que podría sostenerse por los individuos dentro de la organización. Aunque el sistema habrá limitado la capacidad en su formulario del prototipo, una demostración exitosa en algún problema pequeño nutrirá el apoyo por el proyecto.

En el futuro el prototipo madurará al punto dónde puede atacar los problemas reales que formen el dominio. En esta fase de probar, se compara los resultados del sistema con aquellos del experto.

¿Lanzar el Prototipo?

Una dificultad típica que puede descubrirse al seguir la comprobación del prototipo es que la opción original de la herramienta de desarrollo de software era pobre. Por ejemplo, puede encontrarse que la técnica de representación de conocimiento o el método de la inferencia es impropia.

Vía para la Adquisición de Conocimiento

Es también fructífero usar el sistema del prototipo como una vía para adquirir el

conocimiento.

Por su naturaleza, un prototipo del sistema es sólo una rendición pequeña del sistema final. Los límites de su conocimiento en el problema son rápidamente puestos en claro durante la prueba, dónde los fracasos son las reglas. Con la cooperación del experto, un estudio después de este fracaso abre la puerta al conocimiento adicional. El experto puede determinar por qué el resultado dado por el sistema está equivocado, y puede proporcionar la visión en qué conocimiento está extrañando en el sistema que le impidió alcanzar el resultado correcto. De esta forma, el prototipo actúa como otra herramienta que el ingeniero de conocimiento puede usar para sondear el conocimiento adicional.

TAREA 5: Desarrollo de la Interfaz

Deben definirse las características técnicas de la interfaz al principio del proyecto con la cooperación del usuario. El desarrollo de la interfaz debe empezar con el desarrollo del prototipo del sistema experto.

Las claves para un diseño eficaz de la interfaz son:

- Consistencia
- Claridad
- Control
- Colores de la pantalla

TAREA 6: Desarrollo del Producto

Durante el desarrollo del prototipo, se sostienen las sesiones de extracción de conocimiento y se corren las pruebas. Con cada refinamiento, la capacidad del sistema se mejora. En un modo evolutivo, el prototipo del sistema empieza a asumir la forma del sistema final. No hay un punto fijo dónde esta transición ocurre; el prototipo evoluciona gradualmente hasta que el sistema sea completado.

Refinamiento del conocimiento

Una característica básica de un sistema experto es que gana su forma de poder de conocimiento. Esta tarea implica ensanchar y profundizar el conocimiento.

El conocimiento es hecho más ancho agregando nuevos conceptos. En los sistemas basados en reglas cuando se agregan las reglas se agrega a este nuevo conocimiento. En los sistemas basados en frames, el nuevo concepto se agrega generando un nuevo frame de la clase.

Ahondando el conocimiento involucra información adicional que apoya el conocimiento existente. En los sistemas basados en reglas, este tipo de desarrollo se ha realizado agregando reglas que apoyan las reglas existentes. En los sistemas basados en frames, se agregan los nuevos rasgos al marco existente.

Refinamiento del Control

Una versión temprana de un sistema experto normalmente incluye las estrategias de control simples. Una opción de encadenamiento dirigido hacia atrás o adelante podría hacerse, junto con un juego pequeño de metas. Ésta es una manera buena de empezar el diseño, desde que al principio se quiere determinar si se está entrando la dirección correcta. Con los beneficios del proyecto, se verá maneras buenas de introducir las estrategias de control más complejas.

Una área dónde pueden esperarse refinamientos en el control del sistema está en la agenda de la meta. La agenda de la meta proporciona una lista de metas que el sistema sigue en alguna sucesión del juego. Durante el proyecto, se puede encontrar una necesidad para agregar las metas a la agenda o refinar existentes en tareas más finas.

Usted también puede descubrir que la sucesión estricta de una agenda de la meta también está reprimiendo la aplicación. En este evento, se puede querer hacer las metas sensibles al contexto de la sesión. Esto puede lograrse a través del uso de meta-regla. Una meta-regla puede escribirse y establecer las nuevas metas o cargar otras bases de

conocimiento sobre la base de la información descubierta.

Aunque se puede empezar con una sola opción de encadenamiento dirigido hacia atrás o adelante, se puede encontrar una necesidad de cambiar entre ellos. Esta situación normalmente ocurre si el problema involucra varias tareas algunas de los cuales pueden manejarse bien por una de las técnicas de la inferencia. Cuando esto ocurre, se debe estructurar varias bases de conocimiento, cada uno con su propia técnica de inferencia.

El Refinamiento de la interfaz

Algunos de los puntos típicos que el usuario final puede proporcionar como guía son:

- La facilidad de uso.
- Las direcciones de la pantalla.
- Las preguntas.
- Las clarificaciones.
- Los resultados.
- Las técnicas interactivas (el ratón, el lightpen, etc.)

El Razonamiento inexacto.

Algunos sistemas expertos necesitan usar una técnica de razonamiento inexacta. Sin embargo, en las fases tempranas del proyecto, se verifica el conocimiento obtenido del experto en un sentido exacto. Es decir, deben codificarse hechos, reglas, o frames en el sistema de una manera exacta. El resultado del razonamiento del sistema puede verificarse más fácilmente si un acercamiento lógico se toma en la codificación del conocimiento. Siguiendo este paso de la comprobación, pueden usarse los métodos del razonamiento inexacto para refinar la performance del sistema.

FASE 4: PRUEBA

Conforme prosigue el proyecto el sistema experto necesitará ser probado y evaluado periódicamente para asegurar que su performance está convergiendo hacia las metas establecidas. Deben tomarse las decisiones en que se probará, cómo y cuándo las pruebas se dirigirán, y quién será involucrado en las pruebas. Es importante que estas decisiones se tomen temprano, en un momento cuando las metas del proyecto originales se establecen.

El proceso de la evaluación se preocupa más por la aprobación del sistema y aceptación del usuario.

Validación del sistema

Un sistema experto modela la decisión de un experto humano. Si se diseñó correctamente, el sistema deriva los mismos resultados que el experto y razona de una manera similar al experto. Por consiguiente, el esfuerzo de aprobación debe dirigirse a lo siguiente:

- Valide los resultados del sistema.
- Valide que proceso razona el sistema.
- Validar los Resultados.

Durante la prueba, la información del problema se da al sistema experto y la recomendación del sistema se compara con resultados cedidos por un individuo llamado el "evaluador."

Hay tres consideraciones mayores al diseñar una prueba para validar los resultados de un sistema experto:

- La selección del criterio de la prueba.
- La selección de los casos de la prueba.
- La selección del evaluador.

Seleccionar el Criterio de Prueba

Cada proyecto tiene alguna meta para lograr. Para juzgar si el proyecto ha encontrado su meta con éxito, el criterio normalmente se establece cuando el proyecto se evalúa.

Si la organización está usando la tecnología para dirigirse a un problema específico (conducida por el problema), establecer un criterio de prueba entonces es normalmente directo. Es decir, el sistema debe demostrar que logra algún valor medible en tales factores como: economías del costo, mejora de productividad, la mejora de calidad del producto etc. Son problemas muy tangibles, pero ellos son a menudo difíciles de medir hasta que el sistema se haya especializado en el campo.

Un acercamiento diferente confía en comparar la relativa performance del sistema con aquella del experto en el campo.

Seleccionando los Casos de la Prueba

Al trabajar en una aplicación con demandas, es importante que se pruebe el sistema primero para los problemas típicos antes de probar los más difíciles.

Selección de Evaluadores

Si el sistema experto será usado por otros expertos se recomienda que estos sean parte del equipo de “evaluadores”y que no estén asociados al proyecto.

Si el sistema será usado por los no expertos, entonces ellos deben ser parte del equipo de la evaluación. Ellos pueden proporcionar comentarios adelante si el sistema proporciona resultados buenos, los resultados más rápidos, etc.

- Además debe considerar los siguientes puntos:
- Evite el Prejuicio Potencial
- Valide el Razonamiento
- Aprendiendo de los Errores

- La Aceptación del usuario: Dado por:
  - Facilidad de uso.
  - Claridad de las preguntas.
  - Claridad de las explicaciones.
  - Presentación de resultados.
  - Utilidades del sistema.
  - Encuesta al usuario.
- Evolución de la Prueba / Evaluación

  Paso 1: La Comprobación preliminar

  Paso 2: Examinando la demostración

  Paso 3: Probando Validación Informal

  Paso 4: Prueba de refinamiento

  Paso 5: Prueba Formal

  Paso 6: Comprobación del campo

FASE 5: DOCUMENTACIÓN

Como un proyecto de sistema experto maduro, la cantidad de conocimiento recolectado del experto crece. Después de un tiempo, debe encontrar la cantidad de información abrumadora. Para manejar esta situación, tendrá que decidir temprano sobre algún método para documentar efectivamente esta información.

Si está propiamente diseñado, esto también servirá para las siguientes tareas de mantener el sistema y escribir el reporte final del proyecto.

¿Qué necesita ser documentado?

Durante un proyecto de sistema experto, la información que se necesita para retener y grabar en la documentación sirve para tres propósitos primarios:

- Referencias para desarrollar el sistema experto.
- Referencias para redactar el informe final.

- Referencias para mantener el sistema experto.

Durante el esfuerzo de desarrollo, se necesitará volver a menudo a esta documentación para grabar la nueva información o estudiar previamente la información descubierta. Desde que muchos proyectos requieren un reporte final de proyecto, la información grabada en la documentación sirve como una fuente valiosa para este esfuerzo. Siguiendo el despliegue del sistema experto, el sistema necesitará ser mantenido. Para acomodar cada uno de estos esfuerzos, debe documentar lo siguiente:

- Conocimiento
- Gráficos de conocimiento
- Código fuente
- Pruebas
- Transcripciones
- Glosario de términos específicos del dominio
- Reportes.

¿Cómo organizar la Documentación?

Además de contener la información listada en la sección anterior, la documentación debe ser organizada para facilitar el desarrollo del sistema, la escritura de los reportes y el mantenimiento del sistema. Para lograr esto, la documentación debe reunir las siguientes especificaciones:

- Fácil entrada de nuevo conocimiento
- Fácil acceso y modificación del antiguo conocimiento.
- Fácil acceso para la información relacionada.
- Fácil repetición del material para redactar el reporte.

Reporte Final

Para muchos proyectos de sistema expertos necesita escribir un reporte final. Hay variaciones de que será presentado en este reporte que depende de la organización para quien el trabajo fue hecho. El contenido del reporte final del proyecto debe incluir lo siguiente:

- Página del título
- Tabla de contenidos.
- Resumen ejecutivo
- Visión global del proyecto
- Descripción del programa
- Resultados de las pruebas
- Resumen
- Referencias
- Bibliografías
- Apéndices.

FASE 6: MANTENIMIENTO

Muchos sistemas expertos contienen conocimiento que está evolucionando con el tiempo. La organización que usa el sistema puede adquirir nuevos productos y equipos, o cambiar procedimientos para trabajar con los recursos existentes. Este cambio declara modificaciones apropiadas requeridas al sistema.

Conforme es usado el sistema experto, las deficiencias pueden también ser descubiertas. Los usuarios pueden encontrar dificultad para usar el sistema, o pueden descubrir omisiones. Mantener cualquier tipo de software puede ser costoso.

Dada la probabilidad de que necesita cambios el sistema y sus costos asociados, necesita ser establecido un programa de mantenimiento efectivo para cada proyecto de sistema experto. Los usuarios necesitan un camino para reportar problemas que ellos

encuentran, y los individuos con habilidades de ingeniero de conocimiento deben estar disponibles para hacer los cambios. Debe además haber una manera para manejar el esfuerzo del mantenimiento para asegurar que la tarea se logre efectivamente.

Los mayores temas a considerar al reunir un programa de mantenimiento de sistema experto son:

- Documentación
- Pensar en el mantenimiento durante el diseño
- Estructura Modular
- Separar el conocimiento de la información
- Meta Reglas
- Problemas del Software
- Habilidades de programación
- Portabilidad del sistema
- Utilidades de modificación
- Acuerdo de mantenimiento
- ¿Quién mantiene el sistema?
- Cambios del documento.

4.3 Equipo de Desarrollo

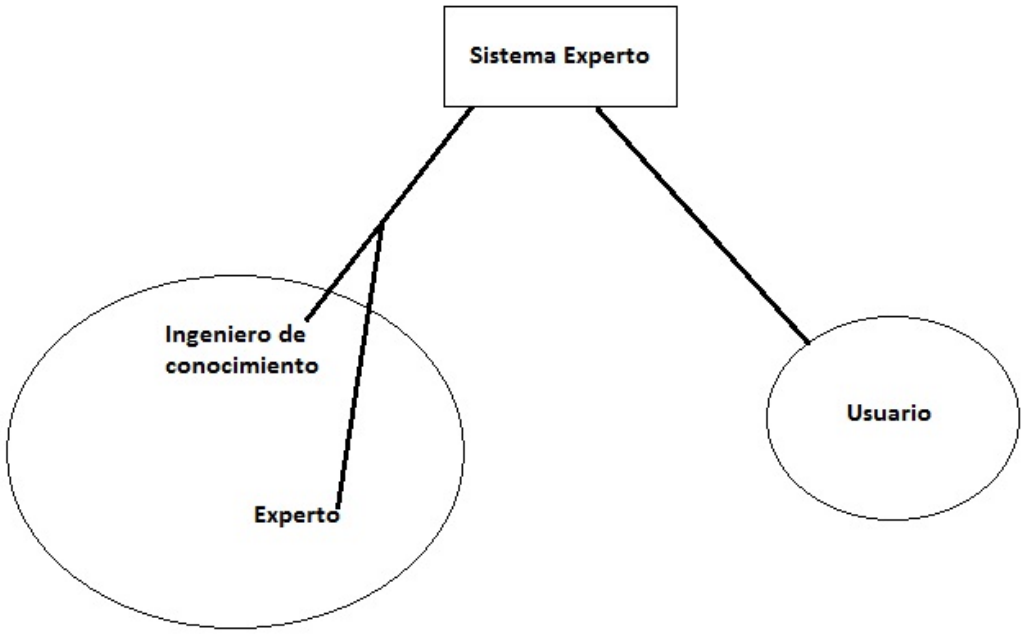
Las personas que componen un grupo o un equipo, como en todos los ámbitos deben cumplir unas características y cada uno de ellos dentro del equipo desarrolla un papel distinto.

A continuación detallaré cada componente del equipo dentro del desarrollo y cuál es la función de cada uno:

- **El experto**: La función del experto es la de poner sus conocimientos especializados a disposición del Sistema Experto.

- **El ingeniero del conocimiento**: plantea las preguntas al experto, estructura sus conocimientos y los implementa en la base de conocimiento;
    - En conjunto con el ingeniero del conocimiento trabajará un desarrollador java Senior y un desarrollador Java Junior.
- **El usuario:** aporta sus deseos y sus ideas, determinado especialmente el escenario en el que debe aplicarse el Sistema Experto.

El esquema de representación del equipo de desarrollo es el siguiente:



En el desarrollo del Sistema Experto, el ingeniero del conocimiento y el experto trabajan muy unidos. El primer paso consiste en elaborar los problemas que deben ser resueltos por el sistema. Precisamente en la primera fase de un proyecto es de vital importancia determinar correctamente el ámbito estrechamente delimitado de trabajo. Aquí se incluye ya el usuario posterior, o un representante del grupo de usuarios. Para la aceptación, y para el éxito del sistema, es de vital y suma importancia tener en cuenta los deseos y las ideas del usuario.

Una vez delimitado el dominio, nos pondremos a "engrosar" nuestro sistema con

los conocimientos del experto. El experto debe comprobar constantemente si su conocimiento ha sido transmitido de la forma más conveniente. El ingeniero del conocimiento es responsable de una implementación correcta, pero no de la exactitud del conocimiento. La responsabilidad de esta exactitud recae en el experto.

En el desarrollo del Sistema Experto, el ingeniero del conocimiento y el experto trabajan muy unidos. El primer paso consiste en elaborar los problemas que deben ser resueltos por el sistema. Precisamente en la primera fase de un proyecto es de vital importancia determinar correctamente el ámbito estrechamente delimitado de trabajo. Aquí se incluye ya el usuario posterior, o un representante del grupo de usuarios.

Para la aceptación, y e consecuencia para el éxito, es de vital y suma importancia tener en cuenta los deseos y las ideas del usuario.

Una vez delimitado el dominio, nos pondremos a "engrosar" nuestro sistema con los conocimientos del experto. El experto debe comprobar constantemente si su conocimiento ha sido transmitido de la forma más conveniente. El ingeniero del conocimiento es responsable de una implementación correcta, pero no de la exactitud del conocimiento. La responsabilidad de esta exactitud recae en el experto.

A ser posible, el experto deberá tener comprensión para los problemas que depara el procesamiento de datos. Ello facilitará mucho el trabajo. Además, no debe ignorarse nunca al usuario durante el desarrollo, para que al final se disponga de un sistema que le sea de máxima utilidad.

La estricta separación entre usuario, experto e ingeniero del conocimiento no deberá estar siempre presente. Pueden surgir situaciones en las que el experto puede ser también el usuario. Este es el caso, cuando exista un tema muy complejo cuyas relaciones e interacciones deben ser determinadas una y otra vez con un gran consumo de tiempo. De esta forma el experto puede ahorrarse trabajos repetitivos.

La separación entre experto e ingeniero del conocimiento permanece, por regla general inalterada.

4.4 Valores del exámen electromigráfico.

Para el diagnóstico será necesario que el paciente indiqué cuales son los síntomas (hechos) que posee y además como complemento a esta información también se tomará en cuenta ciertos valores que arroja el electromiógrafo (que también son hechos para el sistema) para diagnosticar una enfermedad. En base a estos síntomas y valores el sistema comenzará un análisis junto a su memoria de producción, lugar donde se encuentran las reglas. Cuando cierta cantidad de hechos satisfagan a una regla, esta regla será guarda en la agenda para luego ser ejecutada.

Los valores que se ingresarán, además de los síntomas y datos del paciente son los parametros PUM (Potencial de unidad motora) brindados por el electromiógrafo.

En patología neuromuscular se parte siempre de un concepto fisiológico fundamental: el de unidad motora (UM) (Lidell y Sherrington,1925). Una UM es el conjunto formado por una motoneurona alfa del asta anterior de la médula (o del troncoencéfalo), su axón y las fibras musculares por él inervadas. El número de fibras musculares de una UM (también llamado razón de inervación) varía entre 25 o menos en los músculos extraoculares (que requieren un control muy fino) hasta 2000 en los músculos de fuerza como los gemelos. Un potencial de unidad motora (PUM) es el resultado de la sumación temporoespacial de los potenciales de acción de las fibras musculares pertenecientes a una unidad motora.

La mayoría de las enfermedades neuromusculares se deben a la alteración de algún componente de la unidad motora.

Dentro de la actividad voluntaria, los potenciales de unidad motriz (PUM) son el objeto principal de estudio. Consisten en la suma de distintos potenciales de acción de grupos de fibras musculares que se están contrayendo casi sincronizadamente. Pueden ser monofásicos, bifásicos o trifásicos y, en ocasiones, polifásicos con cinco o más fases. Su duración está comprendida entre 2 y 15 ms y su amplitud entre 100 pV y 2 mV, aunque estas magnitudes dependen mucho del tipo de electrodos empleado y del músculo

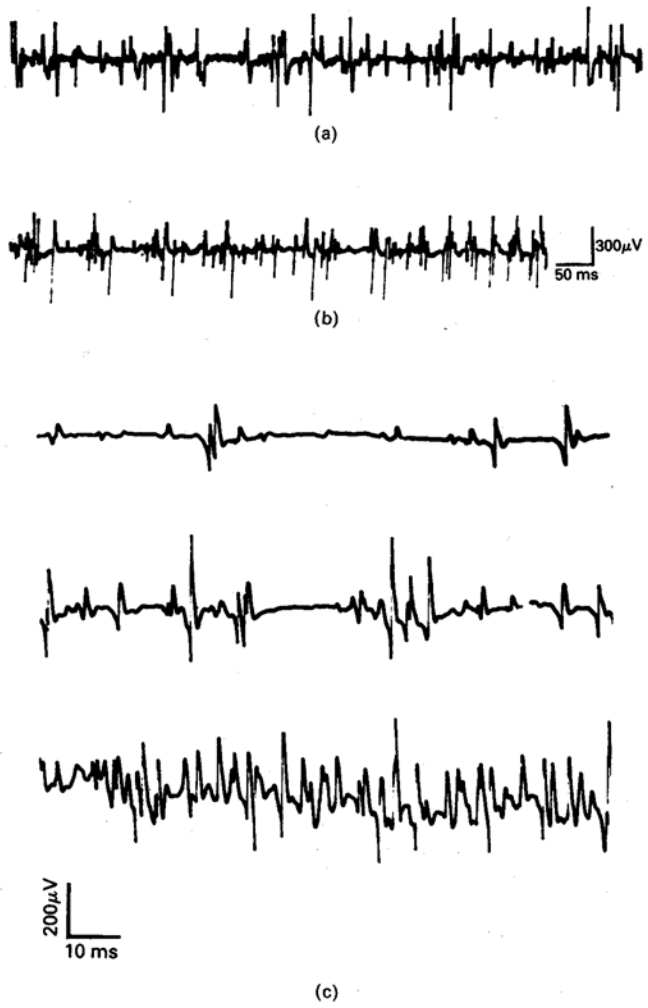
considerado (número de fibras de la UM) (figura 5.a).

La forma y las dimensiones de los PUM pueden modificarse en gran medida en sujetos enfermos: por ejemplo, en algunas nefropatías periféricas la duración de los PUM aumenta, así como su número de fases (figura 5b).

La forma y las dimensiones de los PUM pueden modificarse en gran medida en sujetos enfermos: por ejemplo, en algunas nefropatías periféricas la duración de los PUM aumenta, así como su número de fases.

El registro de los PUM se suele realizar contrayendo débilmente el músculo en observación. Si la contracción se hace mucho más intensa, se obtiene lo que se conoce por patrón de interferencia: los PUM se superponen siendo difícil distinguir sus características individuales.

El aspecto del registro se muestra en la figura 5.c. Los PUM y el patrón de interferencia constituyen los registros principales de la actividad voluntaria



A continuación se estudia la actividad electromiográfica durante la activación voluntaria del músculo para valorar, las características de reclutamiento de los PUM, la configuración de los PUM y el patrón de máximo esfuerzo.

Reclutamiento. Con una contracción de intensidad mínima (umbral de activación) la frecuencia de batido de un PUM es normalmente de 5 a 10 Hz. La frecuencia de reclutamiento es la frecuencia de batido de una unidad motora cuando la siguiente empieza a ser reclutada.

La Configuración de los PUM es de gran importancia cara al diagnóstico. Suelen distinguirse varios parámetros:

Amplitud. Se mide pico a pico y es un parámetro de gran utilidad diagnóstica cuando es claramente patológica.

Duración. La duración de los PUM es uno de los parámetros de más importancia diagnóstica por su correlación con el número de fibras de la UM (véase más adelante). Es mayor en los músculos de los miembros y aumenta con la edad.

Estabilidad. Se analiza mejor atenuando bajas frecuencias del PUM mediante los filtros pasa alta. Es muy útil para evaluar rápidamente la transmisión neuromuscular y la reinervación.

Morfología. Los PUM tienen habitualmente una morfología bifásica, más raramente tri o tetra fásica. Cuando tienen más de 4 fases se denominan polifásicos. Se valora también la presencia de satélites (potenciales tardíos)

Patrón de máximo esfuerzo. Se correlaciona con el número de UM que se activan. Clásicamente se distinguen 5 grados de distintos: normal, deficitario, muy deficitario, simple, ausencia de actividad voluntaria.

En general, los músculos a examinar se seleccionan según la sintomatología que el paciente presente. Si ésta es focal, como en las radiculopatías, deben explorarse, además de los músculos clínicamente afectados, algunos músculos supra e infrayacentes para poder hacer una valoración topográfica. En los procesos generalizados se recomienda explorar músculos proximales y distales pertenecientes a extremidades superiores e inferiores, así como músculos cefálicos y paravertebrales.

En cuanto a la metodología, los *filtros* deben situarse entre 20 y 5 Khz para la actividad espontánea y entre 2 Hz y 10 kHz para el estudio de los PUM, a menos que el programa utilizado indique otros parámetros.

Aparte de la edad deben tenerse en cuenta otros factores que pueden modificar los parámetros de los PUM. El frío tiende a aumentar la duración de los PUM y debe controlarse en los músculos superficiales. El sexo femenino tiende a tener PUMs de duración más breve.

En las miopatías, por ejemplo, los PUM son de baja amplitud como corresponde

a la pérdida de fibras activas de la UM. Adquieren además carácter polifásico por su irregularidad espacial. Sin embargo, al no existir pérdida en el número de UM funcionantes, los patrones de contracción demuestran abundante actividad eléctrica, es decir, numerosos PUM, aunque de pequeño tamaño. (BAPP: breves, abundantes, pequeños, polifásicos).

4.5 Algoritmo de Rete

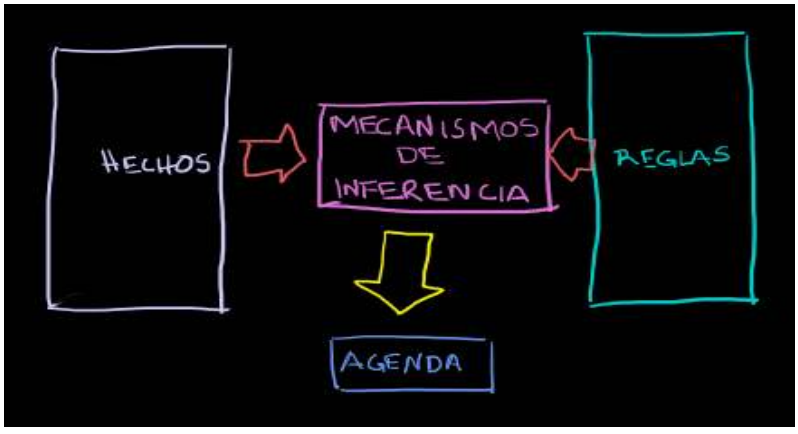
Drools usa un algoritmo muy eficaz para comparar los hechos con los patrones de reglas y determinar cuáles de ellas han satisfecho sus condiciones.

A este algoritmo se le llama algoritmo de comparación de patrones Rete.

Para escribir reglas eficientes no es necesario comprender el algoritmo de Rete, pero la comprensión del algoritmo que este algoritmo facilita que se entienda por qué es más eficiente escribir reglas de una manera específica.

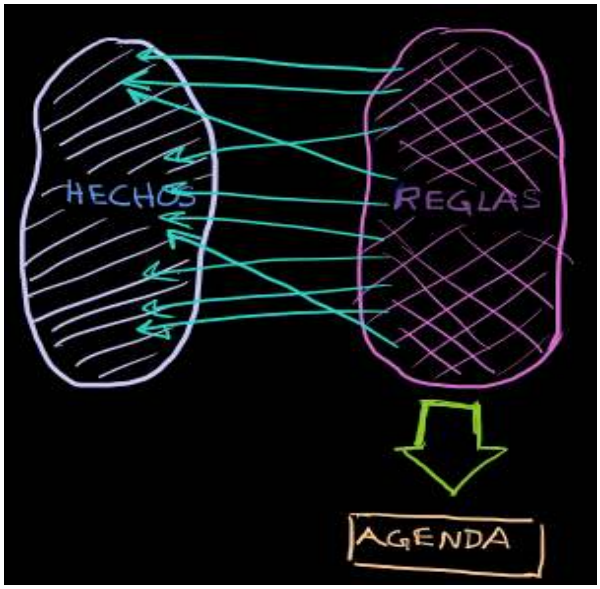
Por una parte tenemos un conjunto de reglas y un conjunto de hechos, también hay un mecanismo de inferencia el cual relaciona ambos conjuntos para ver que reglas tiene hechos que se han cumplidos y de esta manera aquellas reglas que contemplen los hechos cumplidos se guardarán en la agenda para luego ser disparadas.

En la siguiente figura se muestra un problema atendido mediante el algoritmo Rete.



Si sólo tiene que ocurrir una vez el proceso de comparación, entonces la solución del problema es sencilla, el mecanismo de inferencia puede examinar cada regla y luego buscar el grupo de hechos para determinar si se han satisfecho los patrones de la regla.

En este caso es posible colocar la regla en la agenda, como se muestra a continuación



Sin embargo, en los lenguajes basados en reglas, el proceso de comparación tiene lugar varias veces, en condiciones normales la lista de hechos se modificará durante cada ciclo de ejecución, y se pueden agregar o eliminar hechos.

Estos cambios pueden provocar que algunos patrones no satisfechos previamente se satisfagan, o viceversa

El problema de la comparación se vuelve un proceso continuo.

4.5.1 Redundancia temporal

Si se hace que el mecanismo de inferencia verifique cada regla para dirigir la búsqueda de hechos después de cada ciclo de ejecución, se tendrá una técnica simple y directa para resolver este problema, pero este método puede ser muy lento

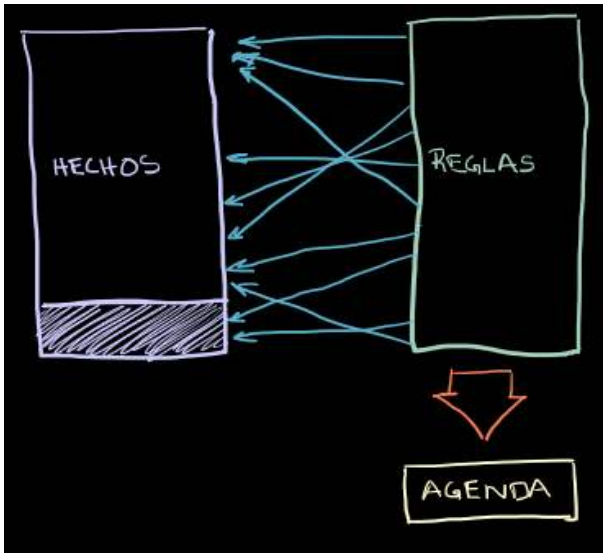
La mayor parte de los sistemas basados en reglas exhiben una propiedad llamada redundancia temporal.

Por lo general, las acciones de una regla sólo cambiarán unos cuantos hechos de la lista, es decir, los hechos cambian lentamente con el tiempo.

Cada ciclo de ejecución puede verificar un pequeño porcentaje de los agregados o cambiados y, por tanto, sólo un pequeño porcentaje de reglas suele verse afectado por estos cambios.

De esta manera, hacer que las reglas dirijan la búsqueda de los hechos necesarios requiere de muchos menos cálculos innecesarios, ya que es probable que la mayor parte de reglas en el ciclo actual no hayan cambiado.

La ineficiencia de este método se muestra en la siguiente figura, en la que el área sombreada representa los cambios que se hicieron a la lista de hechos.



4.5.2 Funcionamiento

Es posible evitar el re cálculo innecesario si se recuerda lo que ya se ha

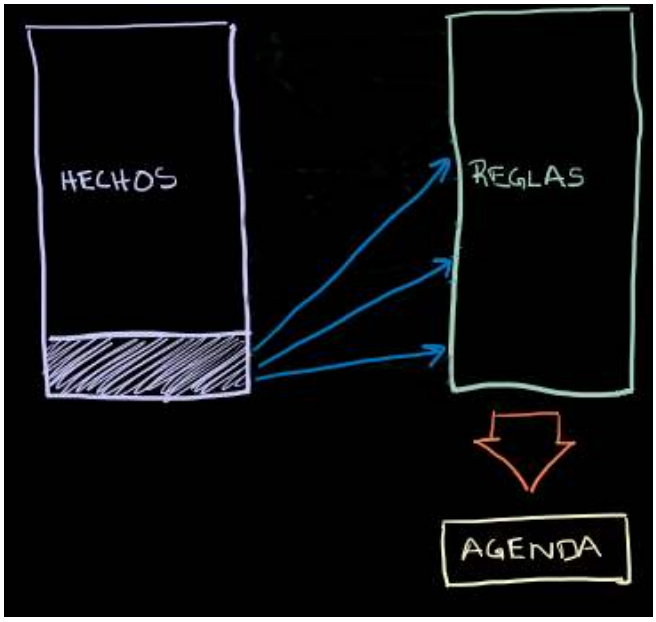
comparado de un ciclo a otro y luego calcular sólo los cambios necesarios para los hechos recién agregados o eliminados.

En esta otra modalidad las reglas permanecen estáticas y los hechos cambian, de modo que los hechos deben encontrar las reglas y no al revés.

El algoritmo de comparación de patrones Rete está diseñado para aprovechar la redundancia temporal exhibida por sistemas expertos basados en reglas, lo que hace al guardar el estado del proceso de comparación de un ciclo a otro y luego calculando de nuevo los cambios en este estado sólo para los cambios que suceden en la lista de hechos.

Es decir, si un conjunto de patrones encuentra dos de los tres hechos necesarios en un ciclo, no se requiere de una verificación durante el siguiente ciclo para los dos hechos que ya se encontraron, sólo el tercer hecho es de interés.

De esta manera tenemos la siguiente estructura:



En lugar las que sean las reglas las que hagan la búsqueda de los hechos, se guardan en memoria aquellos hechos que las reglas tienen identificado que ya se han cumplido y solo aquellos hechos que han cambiado van a buscar si existen o si pueden

confirmar alguna de las reglas que existen.

El estado del proceso de comparación sólo se actualiza a medida que se agregan y se eliminan hechos, si el número de hechos agregados y eliminados es pequeño en comparación con el número total de hechos y patrones, el proceso de comparación avanzará con rapidez.

Coincidencia parcial

Si sólo se procesan las actualizaciones a la lista de hechos, cada regla debe recordar lo que ya se ha comparado,es decir, si un nuevo hecho ha coincidido con el tercer patrón de una regla, la información sobre las comparaciones con los dos primeros patrones debe estar disponible para finalizar el proceso de comparación.

Al tipo de información de estado que indica los hechos que tienen patrones coincidentes en una regla se le llama «coincidencia parcial»

Una coincidencia parcial para una regla es cualquier conjunto de hechos que satisface los patrones de una regla.

Una coincidencia parcial de todos los patrones de una regla también sería una activación.

Coincidencia de patrón

Al otro tipo de información de estado guardada se le llama coincidencia de patrón, que se presenta cuando un hecho ha satisfecho un patrón individual de cualquier regla sin considerar las variables de otros patrones que puedan restringir el proceso de coincidencia.

De esta manera el Algoritmo de comparación Rete es mucho más eficiente porque hace una comparación específica solo de los cambios y tiene una memoria de aquellos elementos que ya se han cumplido en las reglas

Desventajas

La principal desventaja es que utiliza mucha memoria, en tanto que la comparación simple de todos los hechos con todos los patrones no necesita memoria; el

solo guardado del estado del sistema usando coincidencias de patrón y coincidencias parciales puede consumir cantidades considerables de memoria.

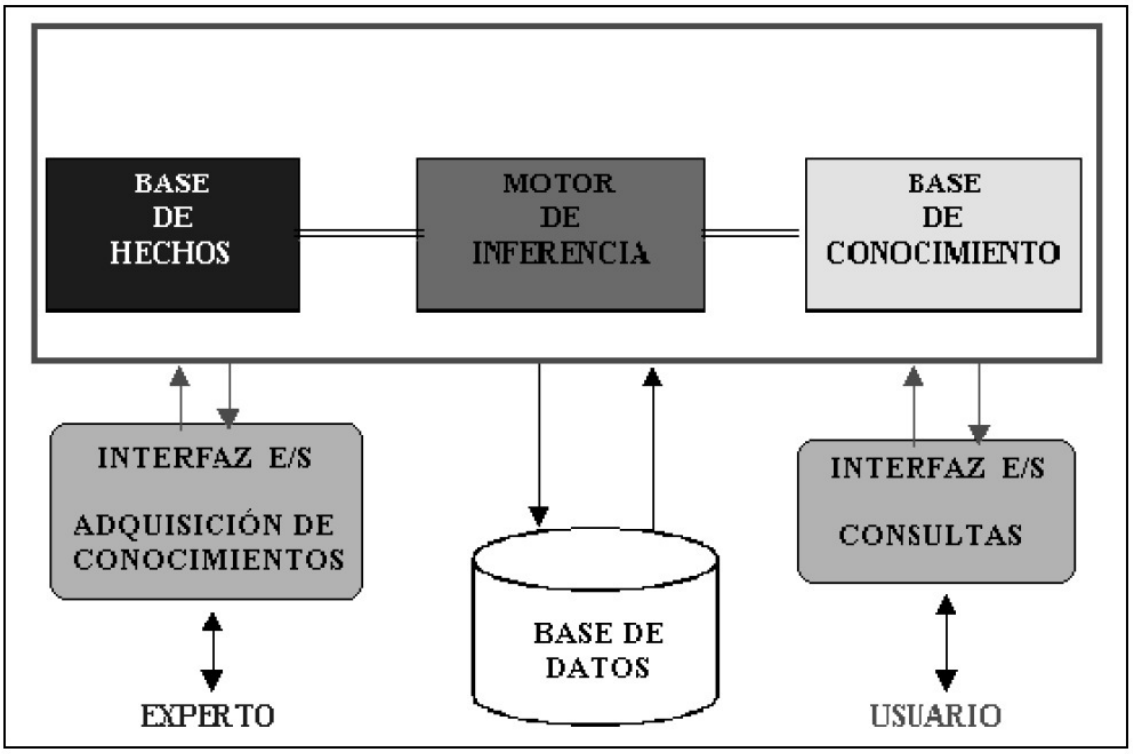
4.6 Arquitectura del Sistema Experto propuesto.

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

Componentes de que tendrá el sistema:

- La Base de Conocimientos o Base de Reglas
- Base de Hechos o Memoria de Trabajo
- El Mecanismo de Inferencia
- Módulo de comunicación o de entrada / salida con el usuario habitual: consultas
- Otros módulo s (opcionales):
  - El Componente o Módulo Explicativo
  - La Interface con el Usuario Experto o Componente de Adquisición
  - Base de Datos

En el gráfico se puede observar la relación de los elementos:



A continuación se detallará cada uno de los elementos:

4.6.1 La base de conocimientos

La base de conocimientos del sistema experto contendrá el conocimiento de los hechos y de las experiencias de los expertos. Esto significa, que contendrá también las reglas y los procedimientos del dominio que son importantes para la solución del problema.

Los conocimientos son del tipo: La neuropatía alcohólica es un trastorno neuromuscular debido al consumo excesivo y prolongado del alcohol. La representación de este conocimiento puede realizarse orientándola, por ejemplo, según objetos. Los objetos de una base de conocimientos pueden ser entonces: enfermedad, trastorno neuromuscular, neuropatía alcohólica. Estos objetos están relacionados de tal forma que “trastorno muscular” tiene todas las cualidades que “trastorno”, y además todas las

cualidades específicas de trastorno neuromuscular.

Todas las cualidades de un trastorno, por ejemplo: causa, síntoma, tratamiento, pronóstico están descritas en el objeto “trastorno”. A través de la relación formulada, “trastorno neuromuscular” hereda estas cualidades, de forma que sólo hará falta describir sus cualidades particulares.

Este tipo de programación se define como programación orientada a objetos y se utiliza con frecuencia en el desarrollo de los Sistemas Expertos.

Cómo se lleva a cabo la clasificación en grupos de las características y de los procedimientos alrededor de un objeto con las técnicas de programación, y cómo deben ser las relaciones entre los objetos pueden variar mucho de aplicación a aplicación.

Junto a estos objetos, la base de conocimientos dispone de reglas. Estas reglas se representan en forma de:

Si *premisas* Entonces *Conclusión y /o Acción*

En la zona de las premisas se solicitan vinculaciones lógicas referentes a las cualidades de los objetos.

En la zona de la conclusión se añaden nuevos hechos y cualidades a la base de conocimientos y/o se ejecutan acciones. Esto se define a menudo como programación orientada a reglas.

En la creación de la base de conocimientos se plantean las siguientes preguntas básicas:

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

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

4.6.2 Base de Hechos

Podemos decir que constituye la Memoria de Trabajo (Working Memory) del sistema experto.

Es el lugar donde se almacenarán los datos de entrada y conclusiones intermedias que se van generando durante el proceso de razonamiento. También llamada Base de Afirmaciones.

Contiene las metas actuales, las hipótesis actuales. También los datos de los estados, del contexto, de los hechos.

Representa, además, el conjunto de reglas pendientes con sus méritos relativos.

Estos hechos representan la estructura dinámica del conocimiento ya que su número puede verse incrementado a medida que se van relacionando las reglas.

La memoria de trabajo es una base de datos global de símbolos representando hechos y aserciones acerca del problema. Los datos son instancias de objetos, que pueden representar objetos físicos u objetos conceptuales relacionados al dominio de aplicación del problema.

Podemos decir que, en un momento dado, el contenido de la memoria de trabajo está indicando el estado de solución del problema.

En general, los objetos de la Memoria de Trabajo son de la forma (objeto atributo valor), como por ejemplo (Raúl edad 38), y son usados para “conducir” la ejecución de las reglas.

La memoria de trabajo contiene elementos, que pueden ser desde simples secuencias de caracteres hasta objetos de estructura compleja.

Las formas de representar estos datos, la implementación, varían de lenguaje a lenguaje en este proyecto se utilizará Java por lo que la representación será en objetos.

4.6.3 El mecanismo de inferencia

El motor de inferencia es como un módulo de software que usa los datos (hechos o evidencia) y el conocimiento (el conjunto de reglas almacenado en la base de conocimiento) para obtener nuevas conclusiones o hechos.

Por ejemplo, si la premisa de una regla es cierta, entonces la conclusión de la regla debe ser también cierta. Los datos iniciales se incrementan incorporando las nuevas conclusiones.

Por ello, tanto los hechos iniciales o datos de partida como las conclusiones derivadas de ellos forman parte de los hechos o datos de que se dispone en un instante dado. Para obtener conclusiones, los expertos utilizan diferentes tipos de reglas y estrategias de inferencia y control

. El método de inferencia que se utilizará en este proyecto es el encadenamiento hacia adelante (el cual se explicará más adelante) ya que se implementará con el sistema de gestión de reglas de negocios DROOLS.

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

Ejemplo, sea una Regla: Si p y q entonces r

Se dan los hechos: p y q. Ellos son justo aquellos hechos que se mencionan en la cláusula "si" de la regla, es decir, las condiciones para la aplicabilidad de la regla.

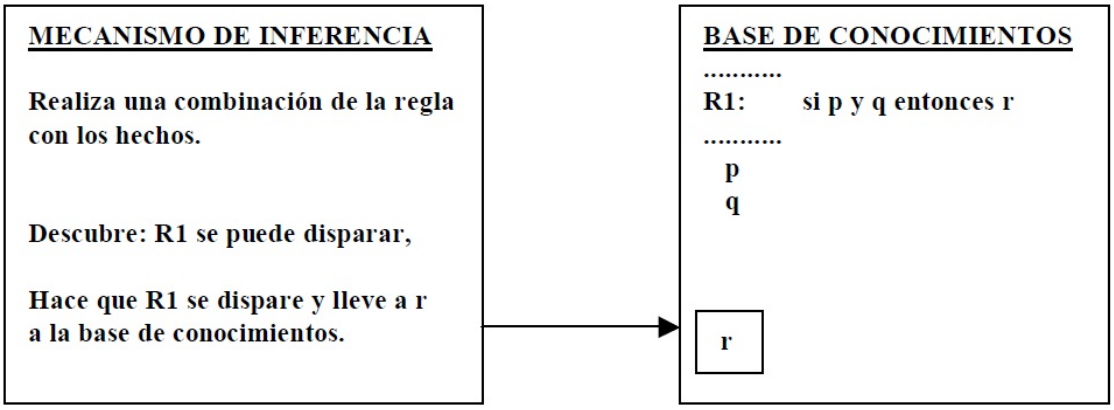
Aplicar la regla es: deducir de los hechos p y q el hecho r.

En un Sistema Experto existirá un hecho sólo cuando esté contenido en la base de conocimientos.

Los hechos que constan en la cláusula "si" se llaman premisas, y el contenido en la cláusula entonces" se llama conclusión. Cuando se aplica una regla sobre algunos hechos cualesquiera se dice que se dispara.

El disparo de una regla provoca la inserción del nuevo hecho en la base de

conocimientos:



4.6.3.1 Reglas de Inferencia

En lo relativo a las reglas de inferencia, básicamente el motor puede usar:

Modus Ponens: es quizás la regla de inferencia más comúnmente utilizada. Se utiliza para obtener conclusiones simples. En ella, se examina la premisa de la regla, y si es cierta, la conclusión pasa a formar parte del conocimiento. Como ilustración, supongamos que tenemos la regla, “Si A es cierto, entonces B es cierto” y que sabemos además que A es cierto. La regla Modus Ponens concluye que B es cierto. Esta regla de inferencia, que parece trivial, debido a su familiaridad, es la base de un gran número de sistemas expertos.

Modus Tollens: se utiliza también para obtener conclusiones simples. En este caso se examina la conclusión y si es falsa, se concluye que la premisa también es falsa. Por ejemplo, supongamos de nuevo que se tiene la regla, “Si A es cierto, entonces B es cierto” pero se sabe que B es falso. Entonces, utilizando la regla Modus Ponens no se puede obtener ninguna conclusión, pero la regla Modus Tollens concluye que A es falso.

Las funciones del mecanismo de inferencia son:

- Determinación de las acciones que tendrán lugar, el orden en que lo harán y cómo

lo harán entre las diferentes partes del Sistema Experto.

- Determinar cómo y cuándo se procesarán las reglas, y dado el caso también la elección de qué reglas deben procesarse.
- Control del diálogo con el usuario.

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

Ante problemas o clases de problemas distintos se estructuran, como es lógico, diferentes mecanismos de inferencia. El mecanismo de inferencia debe de estar "adaptado" al problema a solucionar. Una imposición de dinero exige, bajo ciertas circunstancias, una estrategia distinta de procesamiento del conocimiento para diagnóstico de enfermedades.

4.6.3.2 Estrategias de Inferencia

Existen algunas estrategias básicas de evaluación de las reglas en un sistema de producción llamadas:

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

La estrategia de encadenamiento hacia adelante es una de las estrategias de inferencia más utilizadas para obtener conclusiones compuestas. Esta estrategia puede utilizarse cuando las premisas de ciertas reglas coinciden con las conclusiones de otras. Cuando se encadenan las reglas, los hechos pueden utilizarse para dar lugar a nuevos hechos. Esto se repite sucesivamente hasta que no pueden obtenerse más conclusiones. El tiempo que consume este proceso hasta su terminación depende, por una parte, de los hechos conocidos, y, por otra, de las reglas que se activan. Este algoritmo puede ser implementado de muchas formas. Una de ellas comienza con las reglas cuyas premisas tienen valores conocidos. Estas reglas deben concluir y sus conclusiones dan lugar a

nuevos hechos. Estos nuevos hechos se añaden al conjunto de hechos conocidos, y el proceso continúa hasta que no pueden obtenerse nuevos hechos.

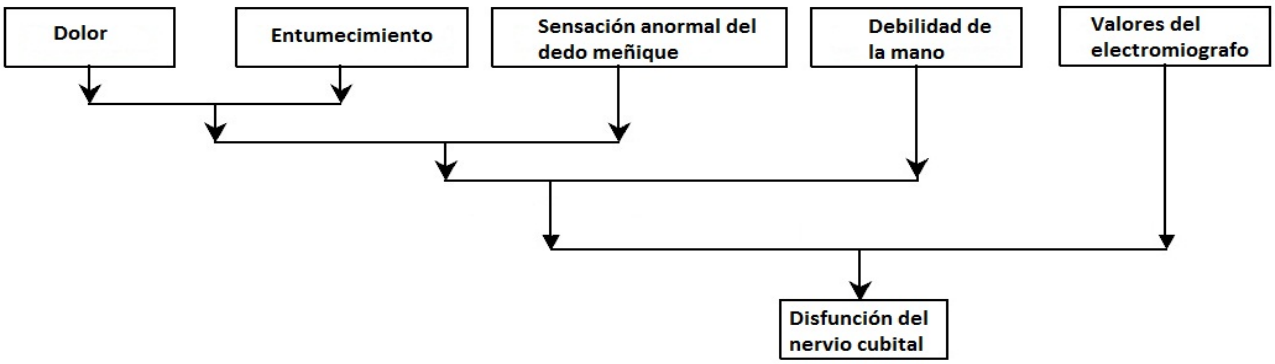
Es la inferencia que hay entre uno y otro dato, es decir a partir de un dato existente y una regla vamos a inferir o deducir nuevos datos adicionales que tienen todas las posibilidades sugeridas.

La base de datos se divide en muchas categorías hasta llegar a un conjunto de datos que satisfacen ciertas condiciones, o nuevos productos. Y estas nuevas hipótesis deben de ser posible satisfacer la verdad.

Al encadenamiento hacia delante se le llama algunas veces “conducido por datos” porque el motor de inferencia utiliza la información que el usuario le proporciona para moverse a través de una red de ANDs y ORs lógicos hasta que se encuentra un punto terminal, que es el objeto. Si el motor de inferencia no puede encontrar el objeto usando la información existente, entonces pide más. Los atributos que definen al objeto crean un camino que conduce al mismo objeto: la única forma de alcanzar dicho objeto es la de satisfacer todas las reglas. Por tanto, un motor de inferencia de Encadenamiento hacia delante comienza con alguna información y luego intenta encontrar algún objeto que encaje con dicha información.

Un sistema de encadenamiento hacia delante esencialmente construye un árbol desde las hojas hasta la raíz.

Se muestra lo anterior en el ejemplo siguiente, en el que se llegaría al objeto ‘’Disfunción del Nervio Cubital’’ cuando se le dan al motor los valores de los atributos correspondientes:



A la síntesis del análisis se le llama encadenamiento hacia atrás, y es el mecanismo opuesto al anterior.

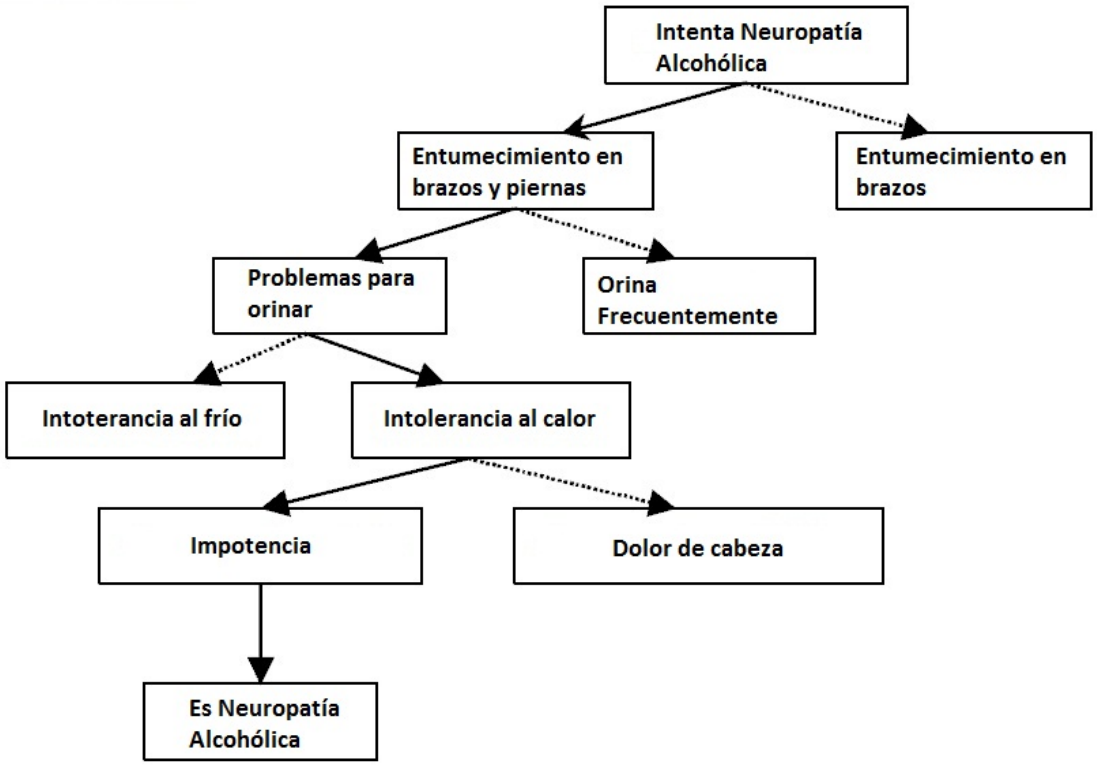
Un motor de inferencia de encadenamiento hacia atrás comienza con la hipótesis (un objeto) y pide información para confirmarlo o negarlo.

Al encadenamiento hacia atrás se le llama algunas veces “conducido por objetos” porque el sistema experto empieza con un objeto e intenta verificarlo.

El mecanismo consiste en tomar cada nuevo objeto “válido” generado y nuevamente proceder a verificarlo a partir de las nuevas alternativas.

Si no se obtiene ninguna conclusión con la información existente, entonces el algoritmo fuerza a preguntar al usuario en busca de nueva información sobre los elementos que son relevantes para obtener información sobre la hipótesis objetivo.

La estrategia de encadenamiento hacia atrás, “poda” un árbol, como puede verse en el ejemplo siguiente:



Como señalé anteriormente, el método de inferencia que utilizará el sistema es el de encadenamiento hacia adelante.

4.6.4 La interface de usuario

La interfaz puede analizarse como formada por diferentes componentes.

Nosotros la veremos, básicamente, como la forma en la que el sistema se presenta ante el usuario.

En el diseño de la misma, como en los módulos anteriores, surgen dudas y preguntas como por ejemplo:

- ¿Cómo debe responder el usuario a las preguntas planteadas?
- ¿Cómo saldrán las respuestas del sistema a las preguntas que se le planteen?
- ¿Qué informaciones se representarán de forma gráfica?

Los requisitos o características de la interface que presentaremos al usuario voy a resumirlas en cuatro, que a mi opinión son las más importantes y las más a tener en cuenta al desarrollar el sistema:

El aprendizaje del manejo debe ser rápido: El usuario no debe dedicar mucho tiempo al manejo del sistema, debe ser intuitivo, fácil en su manejo. No debemos olvidar que nuestro sistema simula al comportamiento de un experto. Debe ser cómodo y tener un manejo relativamente sencillo.

- Debe evitarse en lo posible la entrada de datos errónea.
- Los resultados deben presentarse en una forma clara para el usuario:
  Se insiste en que los resultados deben ser claros y concisos.
- Las preguntas y explicaciones deben ser comprensibles.

Con estas cuatro reglas crearemos nuestra interface con grandes posibilidades de éxito.

4.6.5 El componente explicativo

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

Siempre es deseable que durante el trabajo de desarrollo del sistema se conozca el grado de progreso en el procesamiento del problema. Como he dicho en anterioridad nos pueden surgir unas preguntas como las siguientes:

- ¿Qué preguntas se plantean y por qué?
- ¿Cómo ha llegado el sistema a soluciones intermedias?
- ¿Qué cualidades tienen los distintos objetos?

A pesar de insistir sobre la importancia del componente explicativo es muy difícil y hasta ahora no se han conseguido cumplir todos los requisitos de un buen componente explicativo. Muchos representan el progreso de la consulta al sistema de forma gráfica.

Además los componentes explicativos intentan justificar su función rastreando hacia atrás el camino de la solución. Aunque encontrar la forma de representar finalmente en un texto lo suficientemente inteligible las relaciones encontradas depara las mayores dificultades. Los componentes explicativos pueden ser suficientes para el ingeniero del conocimiento, ya que está muy familiarizado con el entorno del procesamiento de datos, y a veces bastan también para el experto; pero para el usuario, que a menudo desconoce las sutilezas del procesamiento de datos, los componentes explicativos existentes son todavía poco satisfactorios.

4.6.6 El componente de adquisición

Un buen componente de adquisición ayudar considerablemente la labor del Ingeniero del Conocimiento. Este puede concentrarse principalmente en la estructuración del conocimiento sin tener que dedicar tanto tiempo en la actividad de programación.

Como hice en el campo de la interface, daremos unas reglas o requisitos para la realización de nuestro componente de adquisición.

Requisitos o características del componente de adquisición:

El conocimiento, es decir, las reglas, los hechos, las relaciones entre los hechos, etc., debe poder introducirse de la forma más sencilla posible.

- Posibilidades de representación clara de todas las informaciones contenidas en una base de conocimientos.
- Comprobación automática de la sintaxis.
- Posibilidad constante de acceso al lenguaje de programación.

Como se pone en práctica cada uno de los requisitos depender del lenguaje de programación elegido y del hardware que tengamos. El experto deber estar algo familiarizado con el componente de adquisición para poder realizar modificaciones por sí sólo.

El componente de adquisición puede presentarse de diferentes maneras:

- Interfaz de Adquisición interactiva con el usuario (experto o desarrollador generalmente): puede incluir mecanismos para facilitar su adquisición y depuramiento interactivo y para automatizar la adquisición (aprendizaje). Corresponde a la Interfaz de usuario.
- Interfaz de Adquisición automática, no es interactiva con el usuario, realiza aprendizaje a partir de reglas y procedimientos establecidos.
- Los posibles orígenes de conocimientos son:
    - Los nuevos conjuntos de consultas y respuestas, consideradas como válidas por los expertos.
    - La búsqueda en bases de datos convencionales o el producto de otros sistemas software.
    - Una combinación de las anteriores.

4.6.7 Base de Datos

Para el almacenamiento de los datos se utilizará PostgreSQL. Las razones por las que se decide utilizar este motor de base de datos relacional son:

- Licencia gratuita para uso comercial y modificación. (GPL v3)
- Facilidad de backups.
- Posibilidad de manejar grandes cantidades de datos para reportes históricos.

- Posibilidad de replicación/clustering si la carga del sistema aumenta.
- Estabilidad.
- Documentación muy extendida y un grupo de desarrolladores dispuestos a ayudar en la solución de problemas.

En la base de datos se almacenaran todos los datos que el sistema requiera.

4.7 Técnicas de Representación del conocimiento

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

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

Estos métodos formales y matemáticos no son, sin embargo, imprescindiblemente métodos auxiliares apropiados para comunicarse con expertos de los más diversos sectores especializados. Tampoco ofrecen estos métodos 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.

Los procedimientos clave son:

- **Reglas de producción**: estas se basan en la lógica de predicados, una descripción del saber en forma de reglas “si..... entonces.....”
- **Redes semánticas**: una representación gráfica del saber sobre objetos y sus relaciones.

- **Frames**: es una estructura de datos que sirve para representar una situación estereotipada. Añadido a cada Frame hay varios tipos de información. Parte de esta información hace referencia a cómo utilizar el frame; otra se refiere a lo que uno puede esperar que suceda en segundo lugar. Y otra a su vez indica qué hacer si tales esperanzas no son confirmadas
- **Cálculo de predicados**: deducción lógica de resultados; mediante el cumplimiento de determinadas condiciones puede extraerse una deducción lógica. La solución puede tener el valor “verdadero” o “falso”.

Para la realización del sistema se propone usar reglas de producción para la representación del conocimiento, ya que es una de las formas de representación más utilizadas para el desarrollo de un sistema experto, contando con herramientas gratuitas que nos permiten editar, crear y manipular reglas de producción.

4.8 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 una unidad de conocimiento (chunk). Es el componente más pequeño del que consta el sistema en su totalidad.

Se ha descubierto que los expertos están mejor predispuestos a formular sus conocimientos con ayuda de reglas “si....entonces.....”

Esta es al parecer la razón de que en la actualidad la mayoría de los Sistemas Expertos y los de mayor éxito se basen en reglas de producción.

En la realización de Sistemas Expertos con reglas de producción se ha hecho pronto patente la necesidad de procesar conocimientos vagos. Para la valoración de los resultados se utilizan ahora factores de certeza (certainly factors), que son factores

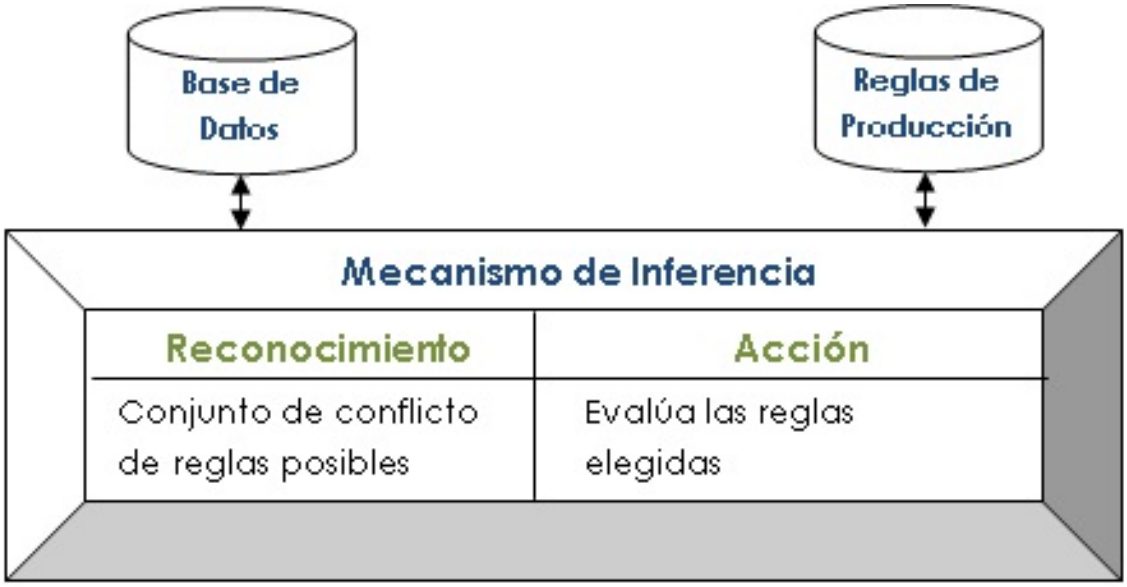
arbitrarios de valoración que suelen encontrarse casi siempre dentro de los límites –1 a +1. Aquí el –1 podría ser “seguro que no”, el –0.5 “probablemente no”, el 0 es “desconocido”, el +0.5 “probablemente sí” y el +1 “seguro que sí”. El ámbito de valores dentro del espacio de solución elegido es continuo.

Ejemplo (extraído de MYCIN, SE médico para la determinación de enfermedades infecciosas bacterianas):

IF **(1) la infección es debida a bacterias primarias, y**
(2) la localización del cultivo es una de las muestras estériles, y
(3) la entrada probable es el tracto gastrointestinal,
THEN **Es bastante probable (,7) que la identidad de los organismos sea bacteria.**

Los sistemas que se redactan con reglas de producción reciben el nombre de Sistemas de Producción.

En la figura puede observarse la arquitectura que tendrá el sistema de producción:



4.9 Ejemplo práctico de aplicación

Un médico puede utilizar un sistema experto para efectuar diagnósticos de manera precisa y rápida. Por ejemplo, mediante un método determinístico para el diagnóstico y a través de inferencias con reglas, como lo muestra el diagrama 2, el médico encuentra el tratamiento correcto para cada enfermedad.

En un sistema experto real se tienen muchas reglas analizadas y diseñadas por los médicos especialistas.

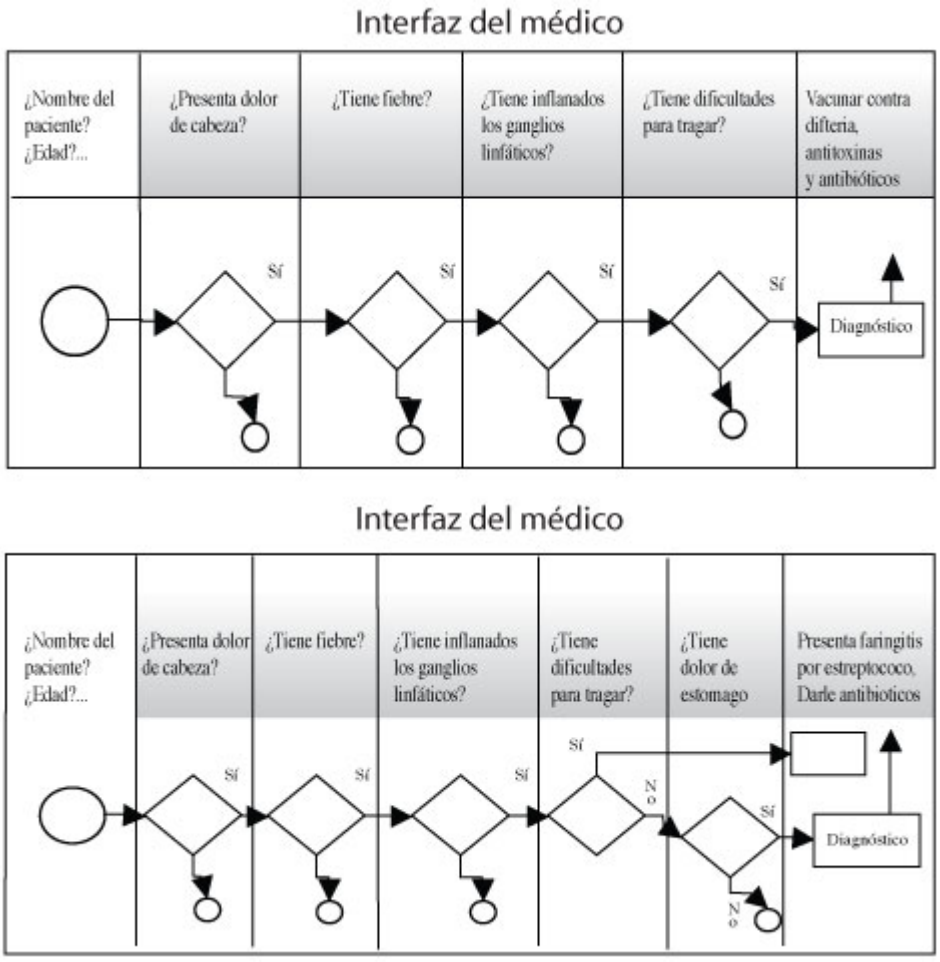
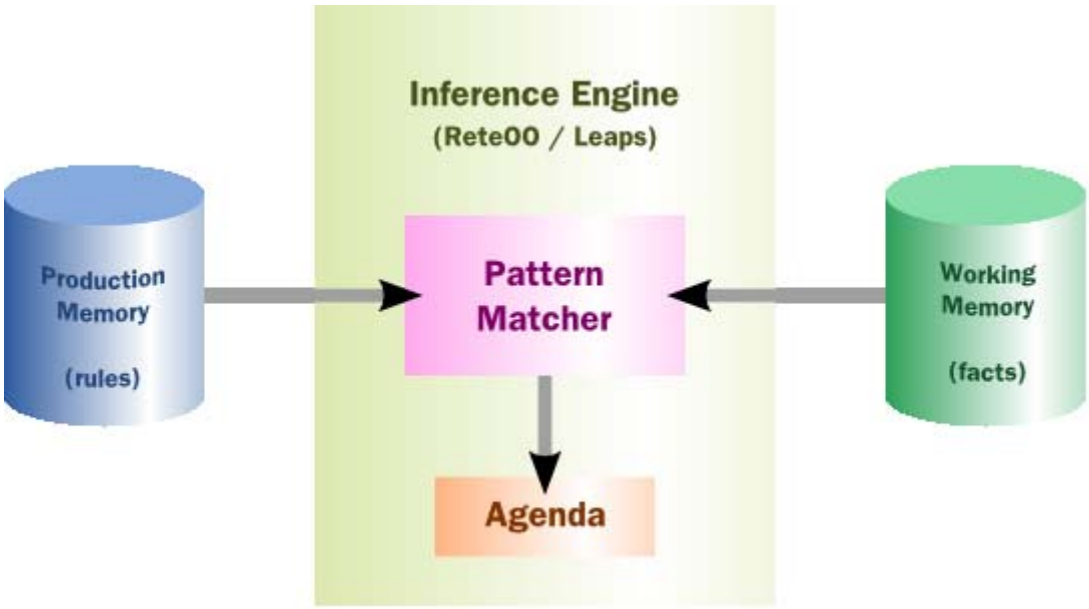


Fig1. Ejemplo sencillo de sistema experto usando el método determinístico.

4.10 Funcionamiento Drools.

El principal elemento es el **motor de inferencias** el cual evalúa reglas de negocio dados unos datos (hechos) para inferir conclusiones que se reflejan en las acciones a tomar. El proceso de evaluación de hechos vs reglas se denomina “Pattern Matching”. El siguiente diagrama representa el sistema de reglas:



- **Memoria de Producción:** representa el espacio donde son almacenadas las reglas.
- **Memoria de Trabajo (Working Memory):** representa el espacio donde la evaluación de reglas toma lugar. Allí se realizan aserciones sobre los hechos de forma que son modificados o retirados. Puede verse como una sesión sobre la cual vamos a lanzar la evaluación de reglas sobre los objetos java (hechos).
- **Activación:** Objeto que se crea cuando los hechos satisfacen las condiciones de una regla.
- **Agenda:** Componente que administra la ejecución ordenada de activaciones creadas a partir de la misma aserción sobre mismos hechos mediante una estrategia de solución de conflictos.
- **Hechos:** Datos que se utilizan para evaluar las reglas. Pueden verse como clases java (beans) sobre los cuales se van a aplicar reglas. Cualquier atributo que vaya a ser consultado o modificado por una regla debe tener su respectivo método getter y setter. Deben ser insertados al working memory antes de lanzar (fire) las reglas.

Luego de insertados los hechos al working memory, se llama al método fireAllRules() que se encarga de revisar las condiciones sobre los objetos y de ejecutar las acciones correspondientes. Por otro lado la creación del Working Memory, la inserción de los Facts y el llamado al método de fireAllRules si se hacen desde la aplicación java.

4.10.1 Manipulación de los hechos

Los hechos declarados son simples objetos Java que pueden ser modificados a través de sus métodos y propiedades públicos.

Los hechos son declarados dentro de la memoria de trabajo con el método assertObject. Ejemplo:

```
WorkingMemory memory = ruleBase.newWorkingMemory();
FactHandle hechoDeclarado1 = memory.assertObject( hechoUno );
FactHandle hechoDeclarado2 = memory.assertObject( hechoDos );
FactHandle hechoDeclarado3 = memory.assertObject( hechoTres );
```

Hechos declarados en la memoria de trabajo pueden ser desmentidos y removidos de la misma usando el metodo retractObject(...). Ejemplo:

```
WorkingMemory memory =ruleBase.newWorkingMemory();
FactHandle hechoDeclarado = memory.assertObject( hechoUno );
memory.retractObject(hechoDeclarado);
```

Los hechos declarados dentro de la memoria de trabajo pueden ser modificados con el método modifyObject. Ejemplo:

```
WorkingMemory memory = ruleBase.newWorkingMemory();
FactHandleTrastornoDeclarado = memory.assertObject( new trastorno( "Neuropatia
Alcohólica" ) );
memory.modifyObject( trastornoDeclarada,new Trastorno( "Mononeuritis Múltiple" ) );
```

Los hechos también pueden ser removidos mientras las reglas son disparadas, utilizando el método retractObject dentro de la consecuencia de la regla.

```
<java:consequence>
        drools.retractObject( trastorno );
</java:consequence>
```

Los hechos también pueden ser modificados mientras las reglas son disparadas, utilizando el método modifyObject dentro de la consecuencia de la regla.

<java:consequence>
trastorno.setSintoma(new Sintoma("Entumecimiento"));
drools.modifyObject(trastorno);
</java:consequence>

4.10.2 Archivo de reglas DRL

Utiliza dos módulos semánticos el Módulo Semántico Base y el Módulo semántico Java.

```
<rule-set name="Reglas trastornos neuromusculares"
    xmlns="http://drools.org/rules"
    xmlns:java="http://drools.org/semantics/java"
    xmlns:xs="http://www.w3.org/2001/XMLSchema-instance"
    xs:schemaLocation="http://drools.org/rules rules.xsd
    http://drools.org/semantics/java java.xsd">

    <rule name="Trastorno Neuropatia Alcoholica">
        <parameter identifier="personaTrastorno">
        <class>org.drools.examples.simple.personaTrastorno</class>
        </parameter>
        <java:condition> personaTrastorno.tieneEntumecimientoPiernas ==
true</java:condition>
        <java:consequence>
            System.out.println( "La persona presenta entumecimiento de
piernas" );
        </java:consequence>
    </rule>

</rule-set>
```

En este caso nuestro rule-set tiene un elemento de regla simple llamado “`Reglas Trastornos Neuromusculares`”. Esta regla busca en la memoria de trabajo instancias de el objeto `org.drools.examples.simple.PersonaTrastorno`. Si encuentra un instancia del objeto `PersonaTrastorno` y además a esa instancia presenta entumecimiento, la regla lanzará una consequence, que imprime “La persona presenta entumecimiento de piernas” en la línea de comandos. Todos los elementos de la regla son

compuestos por uno o más parámetros, una o más condiciones y consecuencias. La forma de estas condiciones y consecuencia son diferentes para cada módulo semántico.

La consecuencia mostrada arriba se lanzará cada vez que una instancia `PersonaTrastorno` que presente entumecimiento pase por este rule set

4.10.3 API en tiempo de ejecución.

Drools contiene una API que puede ser ejecutada de la siguiente manera:

```
//Se cargan las reglas desde el classpath
//en este ejemplo se espera que existe un archive llamado
//TrastornosNeuromusulares.java.drl
//una vez obtenido el classpath señalamos el archivo drl
RuleBase ruleBase = RuleBaseLoader.loadFromStream(
this.getClass().getResourceAsStream(
"/rules/TrastornosNeuromusulares.java.drl" ) );

WorkingMemory workingMemory = ruleBase.newWorkingMemory( );
//pone "personaTrastorno" en la working memory

workingMemory.assertObject( personaTrastorno );
workingMemory.fireAllRules( );
```

La clase RuleBaseLoader nos permite la carga de las reglas de diferentes formas, ya sea desde un stream o un reader. También puede cargarse desde un archivo, un classpath o desde un string.

Las reglas pueden ser implementadas del modo que uno quiera, ya sea en una base de datos o un sistema de archivos son las formas más comunes.

Los atributos xmlns le dicen al Framework del Módulo Semántico que módulo semántico usar cuando parsea el archivo DRL. Cualquier número de módulo semántico adicional puede ser agregado, por lo que se puede escribir su propio lenguaje de reglas.

Dentro del elemento *rule-set* se puede agregar cuantas reglas se necesiten, y cada regla debe tener un atributo *name,* al menos un parámetro, al menos una condición y exactamente una condición.

4.10.4 Módulo Semántico Java

El Módulo Semántico Java permite embeber código Java directamente en el archivo DRL. Esto puede ser algo muy poderoso pero también muy peligroso. Poderoso porque podemos usar el modelo de herencia de los objetos de java. Pero también peligroso porque también uno puede ser tentado a desviarse del modelo de programación declarativa. Esto es muy útil para empezar a trabajar con archivos DRL y motores de reglas, pero se recomienda aislar patrones del código java y encapsularlos en un DSL.

El Módulo semántico Java utiliza una librería llamada “Janino” para parsear el código Java embebido y compilarlo a bytecode. Las condiciones y consecuencias son compiladas durante la construcción del rulebase.

xml namespace: Antes de comenzar a escribir reglas utilizando la semántica de Java se debe declarar el xml namespace.

```
<rule-set name="Trastornos Neuromusculares RuleSet"
        xmlns="http://drools.org/rules" xmlns:java="http://drools.org/semantics/java">
        …..
</rule-set>
```

Condition*:* Se utiliza para determinar si una determinada condición se cumple. Una regla necesita al menos una condición, y cada una debe ser evaluada para poder disparar una consecuencia.

```
<condition> personaTrastorno.tieneEntumecimientoPiernas() == true </condition>
```

Consequence: Permite utilizar el lenguaje Java semántico para manipular la información en la memoria de trabajo.

```
<java:consequence>
        drools.assertObject(personaTrastorno);
        personaTrastorno.presentaNeuropatiaAlcoholica(true);
        System.out.println(personaTrastorno.getNombre() + " presenta Neuropatía
Alcochólica. ");
</java:consequence>
```

4.10.5 Módulo semántico Base

El módulo de base semántico provee un lenguaje base para construir conjuntos de reglas en XML.

Es imposible aplicar solo el módulo semántico base, se debe utilizar además uno del los tres módulos de lenguaje que se incluyen en Drools (Java semantic module, Groovy Semantic Module, Python Semantic Module) o escribir tu propio lenguaje específico de dominio.

rule-set: cada archivo DRL debe tener exactamente un elemento *rule-set*, el cual debe tener un nombre único en la base de reglas.

```
<rule-set name="Trastornos Neuromusuclares“
    xmlns="http://drools.org/rules"
    xmlns:java="http://drools.org/semantics/java">
   …...
</rule-set>
```

rule: Cada *rule-set* debe contener al menos un elemento rule. Cada elemento *rule* debe poseer un nombre único dentro del *rule-set*. Opcionalmente puede contener los atributos:

salience: Numérico. Default =0

no-loop: Booleano. Default=yes

```
<rule name="Hello World" salience="10" no-loop="true">
   ... ...
</rule>
```

Parameter*:* Cada elemento rule debe contener al menos un elemento parameter el cual debe tener un nombre único para su atributo identifier dentro del ámbito de rule. Requiere un ObjectType como una clase:

```
<parameter identifier="goodbye">
        <class>java.lang.String</class>
 </parameter>
```

Class: Es un ObjectType para usarse dentro de un elemento parameter. Contiene el nombre de una clase propia o una clase disponible de un paquete previamente importado.

<class>java.util.HashMap</class>
<class>HashMap</class>

5 Integración de la solución

5.1 Análisis de costos

El costo para construir un sistema experto depende del personal, recursos y el tiempo dedicado a su construcción. Además del hardware y el software necesario para ejecutar una herramienta de sistema experto, también puede haber un costo considerable en la capacitación. Si el personal tiene poca o ninguna experiencia en una herramienta, puede resultar costoso instruirlo.

5.1.1 Costo de desarrollo

Para el desarrollo del sistema será necesario contar con el siguiente personal:

Cantidad	Requisito	Requerimientos	Puesto	Costo
1	Ingeniero en Software	Experiencia 3 años como lider de proyecto Experiencia en desarrollo de sistemas expertos Experiencia mínima 3 años en JAVA	Lider de Proyecto	$ 6.000
1	Programadores JAVA Senior	Experiencia minima 3 años en JAVA	Desarrollador Senior	$ 5.000
1	Programador JAVA Junior	Experiencia minima 1 año en JAVA	Desarrollador Junior	$ 3.000

Para el desarrollo del proyecto se trabajarán 44 horas semanales, aproximadamente 22 días al mes. Observando el cronograma estima que el proyecto durará aproximadamente 13 meses (285 días) de desarrollo, por lo tanto el costo total de personal es:

Costo por unidad	Meses	Total
$ 6.000,00	13	$ 78.000,00
$ 5.000,00	13	$ 65.000,00
$ 3.000,00	13	$ 39.000,00
		$ 182.000,00

5.1.2 Costo implementación

Para la implementación del sistema se recomienda tener una pc con las siguientes características como mínimo:

Procesador: Intel Atom D410 1,66 ghz

Video: Integrado Intel

Memoria Ram: 2Gb DDR2 800

Disco Rígido: 320 Gb SATA II

Sistema Operativo: Linux UBUNTU 10.4.

El costo aproximado de una pc de estas características es de $2400.

Otro costo a tener en cuenta es el costo de capacitación. El costo de capacitación será de $40 la hora. La capacitación durará 4 semanas, se realizará 3 veces por semana con un total de 3 horas por día.

Semanas	Horas	Precio por hora	Total
4	12	$ 40,00	$ 480,00

Para llevar adelante la implantación del sistema es necesario un electromiógrafo que tenga la capacidad de volcar los datos en una computadora. De esta manera, los datos recibidos serán procesados para luego ser ingresados como datos de entrada al sistema propuesto. Como anteriormente se mencionó, el proyecto no tiene como finalidad proporcionar este equipamiento. Teniendo en cuenta que el sistema tiene como usuario final un electromiografista, éste ya debería contar con uno. Por lo tanto, los costos de equipamiento no se contemplarán en el análisis de costo a desarrollar.

5.2 Análisis del impacto

La implementación del proyecto tendrá un impacto directo sobre la efectividad a la hora de diagnosticar una enfermedad neuromuscular, el médico y el paciente, influyendo directamente sobre la calidad en el servicio de los hospitales, centro médicos, clínicas, etc.

5.2.1 Situación Actual

El médico toma los datos del paciente: nombre completo y apellido. El paciente le comenta al médico que síntomas, dolores o molestias ha tenido. El médico busca en ficheros el historial clínico del paciente si tuviese. El médico le realiza las preguntas de rutina.

Una vez con la historia clínica y la exploración neurológica, el electromiografista establece una hipótesis sobre la localización de la lesión. A continuación, y mediante la electromiografía, confirma el lugar de la lesión.

Actualmente, el tiempo necesario para un examen es variable, aunque debe calcularse una hora por paciente, incluyendo la elaboración del informe, la búsqueda del historial clínico del paciente y las preguntas de rutina. El examen se realiza por un electromiografista, que es un médico especialista en neurofisiología, es decir, experto en la anatomía, fisiología y patología del sistema neuromuscular.

5.2.2 Mejora Esperada.

Con el sistema actual se pretende acortar el tiempo invertido por paciente a la hora de realizar una consulta por trastornos neuromusculares con electromiografía, como también brindar apoyo la hora diagnosticar un trastorno neuromuscular.

El sistema almacenará los datos de cada paciente, nombre, apellido, dni, dirección, teléfono, datos subjetivos proporcionados por el paciente (síntomas), datos objetivos obtenidos de la exploración física y de las exploraciones complementarias

obtenidos luego de la elecromiografía, diagnóstico, pronóstico y tratamiento. Con estos datos se creará el historial clínico del paciente. Esto le permitirá ahorrar tiempo al médico en la búsqueda del historial clínico del paciente, ya que ingresando por ejemplo su dni el sistema brindará la información completa del historial clínico del paciente o le dará la posibilidad de crear uno si no existiese uno.

Luego de ingresar los síntomas en el sistema y en conjunto con los resultados del electromiógrafo, éste generará un informe sobre los posibles trastornos que puede tener el paciente así como también los posibles músculos afectados, brindándole al médico apoyo para las decisiones a tomar a la hora de generar un diagnóstico definitivo, y como consecuencia un diagnóstico totalmente confiable.

El médico tendrá la posibilidad de ver cómo el sistema llegó a ese resultado, corroborar los valores arrojados por el electromiógrafo, valores normales, guardar los resultados y el diagnóstico generado por el sistema asignándolo al historial clínico del paciente.

De esta manera se realizará un sistema que no sólo genere un posible diagnóstico sino que también integrará otras funciones que el médico debe realizar cada vez que atiende a un nuevo paciente ahorrando una cantidad de tiempo considerable.

5.2.3 Resumen de la solución

El sistema planteado integra varias tecnologías entre ellas Java para la interfaz del sistema y Drools como el motor de reglas. El IDE en el que se puede desarrollar el sistema puede ser Netbeans o Eclipse ambos gratuitos, pero recomiendo usar el segundo ya que existe un plug-in de Drools para Eclipse, el cual nos permite trabajar cómodamente editando, creando y depurando reglas de negocio. La forma más sencilla de instalar el plug-in es a través del gestor de actualizaciones de Eclipse. La base de datos en la que se almacenarán los datos del paciente, el diagnóstico y otra información es PostgreSQL.

6 Conclusiones

Los sistemas expertos son de mucha utilidad en la vida real, y apoyan en gran manera a los sistemas de soporte de decisión, ya que permiten tomar decisiones basadas en la experiencia humana de algún especialista en determinada área. Esto es con el fin de retener el conocimiento y de esa manera convertirlo en un activo importante para cualquier organización. Este tipo de sistemas permite ayudar a tomar decisiones correctas aún a aquellas personas que tienen todos los conocimientos pero tal vez no la suficiente experiencia, como por ejemplo, un médico que está comenzando a incursionar en el ámbito laboral.

El período de realización de un sistema experto es largo, no por el desarrollo de la aplicación, sino por el proceso de adquisición de conocimientos, ya que el conocimiento a adquirir es un conocimiento especializado, con el cual el ingeniero del conocimiento no se encuentra familiarizado.

Muchas veces, los trastornos neuromusculares son confundidos con otros, llevando al paciente a comenzar un tratamiento a destiempo. Este Sistema Experto ayudará en el diagnóstico rápido de trastornos neuromusculares, para que comience un tratamiento de manera inmediata, y evitar efectos severos.

A lo largo del documento se ha logrado cumplir con el objetivo general que es determinar la viabilidad de desarrollo de un Sistema Experto para la realización de diagnóstico de trastornos neuromusculares con electromiografía.

Adicionalmente se ha dado respuesta a cada uno de los objetivos específicos planteados en el punto 1.6.2. Dentro de los mismos se incluía:

- Realizar un cronograma que estime el tiempo necesario para realizar el proyecto, así como también los recursos asignados a cada tarea. Esto fue realizado en el punto *3.1 Cronograma del sistema.*
- Analizar que arquitectura debería tener el sistema experto para la implantación de

un sistema de diagnóstico para enfermedades neuromusculares con electromiografía. Esto fue resuelto en el punto *3.3 Arquitectura del Sistema Experto propuesto.*

- Definir los costos de la implementación de las soluciones. Esto fue resuelto en el punto *4.1 Análisis de costos*
- Determinar las tecnologías a integrar para el desarrollo de la solución. Este punto fue resuelto durante el desarrollo del proyecto. El sistema se puede realizar con diferentes tecnologías, aquí se propuso utilizar el lenguaje de programación Java, Drools como motor de reglas, Eclipse como IDE y PostgreSQL como base de datos. Todas estas características son multiplataforma, es decir que el sistema funcionará bajo diferentes sistemas operativos, como Windows o Linux. Otra ventaja es que las herramientas propuestas son todas de código abierto y libre de licencias, como son Linux, PostreSQL, Java y Drools.

7 Anexos

“Cronograma del sistema.mpp”: archivo de Microsoft Project el cual incluye el cronograma de sistema experto, tareas involucradas en cada fase y recursos asignados a cada tarea.

8 Bibliografía

8.1 Libros / Redacciones

a) Cohen, D. “SISTEMAS DE INFORMACIÓN PARA LA TOMA DE DECISIONES”. McGrawHill.
b) Durkin, J. “EXPERT SYSTEMS: DESIGN AND DEVELOPMENT”. New York. Maxwell Macmilan. 1994
c) Sánchez, J. “SISTEMAS EXPERTOS: UNA METOLOGIA DE PROGRAMACION”.Prentice Hall. 1991
d) Giarratano Joseph. “SISTEMAS EXPERTOS: PRINCIPIOS Y PROGRAMACION”. International Thomson Editores. Tercera Edición. 2001.

8.2 Sitios de Internet

a) www.postgresql.org: Página oficial del motor de base de datos, contiene descargas, documentación y foro para consultas mantenido por una amplia comunidad de usuarios.
b) http://www.jboss.org/drools: Página oficial del motor reglas propuesto, contiene descargas, documentación y foro para consultas mantenido por una amplia comunidad de usuarios.
c) http://www.informaticaintegral.net/
d) http://www.enterate.unam.mx/
e) http://www.saludalia.com/docs/Salud/web_saludalia/temas_de_salud/doc/neurologia/doc/doc_electromiografia.htm
f) http://www.nlm.nih.gov/medlineplus/spanish/ency/article/003927.htm

9 Glosario

Linux: Sistema operativo de código abierto y libre de uso para fines comerciales. Bajo la licencia GPL.

Java: es un lenguaje de programación orientado a objetos, desarrollado por Sun Microsystems a principios de los años 90. El lenguaje en sí mismo toma mucha de su sintaxis de C y C++, pero tiene un modelo de objetos más simple y elimina herramientas de bajo nivel, que suelen inducir a muchos errores, como la manipulación directa de punteros o memoria.

IDE: (en inglés integrated development environment) es un programa informático compuesto por un conjunto de herramientas de programación. Es un entorno de programación que ha sido empaquetado como un programa de aplicación, es decir, consiste en un editor de código, un compilador, un depurador y un constructor de interfaz gráfica (GUI).

Plug-in (complemento): Un complemento es una aplicación que se relaciona con otra para aportarle una función nueva y generalmente muy especifica. Esta aplicación adicional es ejecutada por la aplicación principal e interactúan por medio de la API. También se lo conoce como plug-in (del inglés "enchufable"), add-on (agregado), complemento, conector o extensión.