

UNIVERSIDAD DE GUAYAQUIL

**FACULTAD DE CIENCIAS MATEMATICAS Y FISICAS
CARRERA DE INGENIERIA EN SISTEMAS COMPUTACIONALES**

IMPLEMENTACIÓN DE UN SISTEMA DE BASE DE DATOS

INTELIGENTE INTEGRANDO BASE DE

CONOCIMIENTOS EN PROLOG

Y BASE DE DATOS MYSQL

TESIS DE GRADO

Previa a la obtención del Título de:

INGENIERO EN SISTEMAS COMPUTACIONALES

AUTOR: ALEX EZEQUIEL PORTILLA PANCHANA

TUTOR: ING. MARÍA ANGÉLICA SANTA CRUZ

GUAYAQUIL – ECUADOR

2010

CERTIFICADO DE ACEPTACIÓN DEL TUTOR

En mi calidad de Tutor del Primer Curso de Fin de Carrera, nombrado por el Departamento de Graduación y la Dirección de la Carrera de Ingeniería en Sistemas Computacionales de la Universidad de Guayaquil,

CERTIFICO:

Que he analizado el Proyecto de Grado presentado por el/la egresado(a) ALEX EZEQUIEL PORTILLA PANCHANA, como requisito previo para optar por el título de Ingeniero cuyo problema es:

IMPLEMENTACION DE UN SISTEMA DE BASE DE DATOS INTELIGENTE INTEGRANDO BASE DE CONOCIMIENTOS EN PROLOG Y BASE DE DATOS MYSQL

Considero aprobado el trabajo en su totalidad.

Presentado por:

Portilla Panchana Alex Ezequiel

CI: 0916725898

María Angélica Santacruz M.

Tutor

Guayaquil, marzo de 2011

Guayaquil, marzo de 2010

APROBACIÓN DEL TUTOR

En mi calidad de Tutor del trabajo de investigación, “IMPLEMENTACION DE UN SISTEMA DE BASE DE DATOS INTELIGENTE INTEGRANDO BASE DE CONOCIMIENTOS EN PROLOG Y BASE DE DATOS MYSQL” elaborado por el Sr. **ALEX EZEQUIEL PORTILLA PANCHANA**, egresado de la Carrera de Ingeniería en Sistemas Computacionales, Facultad de Ciencias Matemáticas y Físicas de la Universidad de Guayaquil, previo a la obtención del Título de Ingeniero en Sistemas, me permito declarar que luego de haber orientado, estudiado y revisado, la Apruebo en todas sus partes.

Atentamente

Ing. María Angélica Santacruz M.

TUTOR

DEDICATORIA

Este trabajo, quiero dedicarlo a mis padres Carlos y Teresita de Jesús, por todo el cariño demostrado a lo largo de mi vida, quienes me dieron su apoyo incondicional para que yo pudiera alcanzar esta meta. A mi esposa Jessica mi hija Katherine y Elvira, por haber estado a mi lado en los momentos que los necesité.

AGRADECIMIENTO

Quiero agradecer a Dios por todas sus bendiciones, por ser mi guía, ya que sin él no podría lograr nada. A los docentes, que compartieron sus conocimientos a lo largo de mi carrera universitaria. También a todos mis compañeros y amigos que colaboraron y confiaron en mí, para el desarrollo de éste proyecto.

TRIBUNAL DE GRADO

Ing. Fernando Abad Montero
DECANO DE LA FACULTAD
CIENCIAS MATEMATICAS Y FISICAS

Ing. Juan Chanabá Alcócer
DIRECTOR

Ing. Xavier Loaiza Herrera
MIEMBRO DEL TRIBUNAL

Ing. Jorge Medina Avelino
MIEMBRO DEL TRIBUNAL

Ing. María Angélica Santacruz Maridueña
MIEMBRO DEL TRIBUNAL (TUTOR)

AB. José Júpiter Wiles
SECRETARIO

UNIVERSIDAD DE GUAYAQUIL

FACULTAD DE CIENCIAS MATEMATICAS Y FISICAS
CARRERA DE INGENIERIA EN SISTEMAS COMPUTACIONALES

IMPLEMENTACIÓN DE UN SISTEMA DE BASE DE DATOS***INTELIGENTE INTEGRANDO BASE DE******CONOCIMIENTOS EN PROLOG******Y BASE DE DATOS MYSQL***

Autor: Alex Portilla Panchana
Tutor: María Angélica Santacruz

RESUMEN

De acuerdo a las investigaciones podemos indicar que la implementación de este proyecto, ayudara al personal médico de nuestro país a detectar y evaluar los síntomas de los pacientes de pediatría que sufren de trastornos gastrointestinales; ayudándonos a detectar esos casos en los niños y con toda esta información poder reducir los índices de mortalidad, empleando recomendaciones oportunas basados en experiencia del experto el cual esta registrado en la base de conocimiento del sistema.

La presente propuesta consiste en implementar de un sistema de base de datos inteligente integrando Prolog y MySQL a través de una interfaz gráfica desarrollada en Java. El uso de estas herramientas nos permitirá combinar la programación convencional procedural y la programación lógica.

Para el desarrollo del prototipo usaremos un lenguaje de programación lógica Prolog con una base de datos MySQL, utilizando una rama de la inteligencia artificial que se encargará de almacenar, relacionar, inferir y mostrar una solución en base a conocimientos recopilados de estos expertos de los diferentes casos registrados anteriormente, los cuales son almacenados en una base de datos MySQL, la misma que nos brindara la integridad en los datos que se necesitan, sin contar con la ventaja en el costo por tratarse de una herramienta open source.

ABSTRACT

In agreement to the investigations we can indicate that the implementation of this project, it should help to the medical personnel of our country to detect and evaluate the symptoms of the patients of pediatrics who suffer from gastrointestinal disorders; helping ourselves to detecting these cases in the children and with all this information to be able to reduce the death rates, using opportune treatments based on the cases already registered in our system.

Our offer consists of helping of a system of intelligent database integrating Prolog and MySQL. The use of these tools they will allow us to combine the conventional programming procedural and the logical programming in the clinical systems, so much in hospitals, medical doctor's offices and clinics of health.

For the development of the prototype we will use a logical language of Prolog's programming with a database MySQL, using a branch of the artificial intelligence that it will take charge storing, relating, inferring and showing a solution on the basis of knowledge compiled of these experts of the different cases registered previously, which are stored in a database MySQL, the same one that was offering to us the referential integrity in the information that are needed, without possessing the advantage in the cost for treating itself about a tool open source.

UNIVERSIDAD DE GUAYAQUIL

**FACULTAD DE CIENCIAS MATEMATICAS Y FISICAS
CARRERA DE INGENIERIA EN SISTEMAS COMPUTACIONALES**

IMPLEMENTACIÓN DE UN SISTEMA DE BASE DE DATOS

INTELIGENTE INTEGRANDO BASE DE

CONOCIMIENTOS EN PROLOG

Y BASE DE DATOS MYSQL

Proyecto de trabajo de grado que se presenta como requisito para optar por el título de INGENIERO en Sistemas Computacionales

Autor: Alex Portilla Panchana

C.I. 0916725898

Tutor: María Angélica Santacruz

Guayaquil, marzo de 2011

ÍNDICE GENERAL

	Pág.
CARÁTULA	i
CERTIFICADO DE ACEPTACIÓN DEL TUTOR	ii
CARTA DE ACEPTACIÓN DEL TUTOR	iii
DEDICATORIA	iv
AGRADECIMIENTO	v
TRIBUNAL DE GRADO	vi
RESUMEN	vii
ÍNDICE GENERAL	x
ÍNDICE DE CUADROS	xiii
ÍNDICE DE GRÁFICOS	xv
INTRODUCCIÓN	1
CAPÍTULO 1.- EL PROBLEMA	
PLANTEAMIENTO DEL PROBLEMA	3
Ubicación del Problema en un contexto	3
Situación Conflicto Nudo Críticos	4
Causas del problema, Consecuencias	4
Delimitación del Problema	5
Formulación del Problema	6
Evaluación del Problema	7
Objetivos de la Investigación	8
Objetivos Generales	8
Objetivos Específicos	9
Justificación e Importancia de la Investigación	9

	Pág.
CAPÍTULO II.- MARCO TEÓRICO	
Antecedentes del estudio	13
Fundamentación Teórica	14
¿Qué son bases de datos inteligentes?	14
Las características de la propuesta	16
Utilidad para el país, sector, la universidad	17
Beneficiarios	18
Sistema experto	18
Componentes de un sistema experto	21
Lenguajes de Programación de Sistemas Expertos	22
Ejemplos de Sistemas Expertos	23
Limitaciones de los sistemas expertos	23
Representación del conocimiento en un sistema experto	24
Formas de representación de los conocimiento	25
Ingeniería del conocimiento	25
Adquisición de conocimiento	26
Inteligencia artificial	29
Historia de la Inteligencia Artificial	31
El Futuro de la IA	33
Técnicas y campos de la Inteligencia Artificial	34
Prolog	35
Objetivos para el cual fue creado	37
Predicados	38
Historia de Prolog	39
Estructura de Prolog	40
Hechos	40
Reglas	41
Variables	41
predicado consult	43

Predicados dinámicos	43
La persistencia	43
Base de Datos Relacionales	45
Características	47
Componentes de una base de datos	48
Tipos de usuario de una Base de Datos	49
SQL	49
Lenguaje de Definición de datos (DDL)	51
Sentencia select	51
Procedimientos almacenados	51
Productos	51
Las principales características de MySQL	51
Ventajas de las Bases de Datos	51
Desventajas de las Bases de Datos	52
Las principales características de MySQL	53
Administración de bases de datos	53
Sistema de gestión de base de datos (SGBD)	55
Pediatría	56
Gastroenterología	59
Enfermedades gastrointestinales	60
Componentes del sistema GastroMedical	62
Base de conocimientos	62
Base de hechos	63
El motor de inferencia	66
Reglas	66
Interfaz gráfica	67
Fundamentación legal	69
Hipótesis preguntas a contestarse	71
Variables de la investigación	77
Definiciones conceptuales	78

	Pág.
Reglas	78
Modus Ponens	78
Inferencia por encadenamiento de reglas	79
Bases de Datos Relacionales	81
JAR	82
 CAPÍTULO III.- METODOLOGÍA	
DISEÑO DE LA INVESTIGACIÓN	84
Modalidad de la Investigación	84
Población y Muestra	85
Tamaño de la Muestra	86
Matriz de operacionalización de las variables	87
Instrumento de recolección de datos	87
Instrumentos de la investigación	88
Entrevista	88
Procedimientos de la investigación	89
 CAPÍTULO IV.- MARCO ADMINISTRATIVO	
Cronograma	92
Presupuesto	93
Manual del Prototipo	94
Manual de Usuario de la aplicación	151
 ANEXOS	 163
 CAPÍTULO V.- CONCLUSIONES Y RECOMENDACIONES	
Conclusiones	183
Recomendaciones	187
REFERENCIAS BIBLIOGRÁFICAS	189

ÍNDICE DE CUADROS

	Pág.
Cuadro # 1	52
Ventajas de las bases de datos	
Cuadro # 2	53
Desventajas de las bases de datos	
Cuadro # 3	73
Costos sin sistema GastroMedical	
Cuadro # 4	73
Costos con el sistema GastroMedical	
Cuadro # 5	74
Beneficios tangibles	
Cuadro # 6	74
Beneficios intangibles	
Cuadro # 7	75
Preguntas y Respuestas	
Cuadro # 8	76
Tabla comparativa	
Cuadro # 9	85
Población	
Cuadro # 10	87
Matriz de operacionalización de variables	
Cuadro # 11	93
Presupuesto	

ÍNDICE DE GRÁFICOS

	Pág.
Gráfico # 1	26
Etapas de un Sistema Experto	
Gráfico # 2	28
Campos de la Inteligencia Artificial	
Gráfico # 3	42
Esquema del funcionamiento de la máquina Prolog	
Gráfico # 4	46
Tablas relacionadas mediante identificadores	
Gráfico # 5	48
Etapas de un Sistema Experto	
Gráfico # 6	49
Componentes de una base de datos	
Gráfico # 7	54
DBMS Subsistemas	
Gráfico # 8	78
Sistema de Inferencia	
Gráfico # 9	80
Encadenamiento de Reglas	
Gráfico # 10	94
Integración	
Gráfico # 11	96
Arquitectura GastroMedical	

Gráfico # 12	97
Casos de uso	
Gráfico # 13	98
Instalación MySQL	
Gráfico # 14	99
Instalación Query Browser	
Gráfico # 15	101
Diagrama Entidad Relación Roles/Usuarios	
Gráfico # 16	104
Diagrama Entidad Relación del Sistema	
Gráfico # 17	110
Instalación SWI-Prolog	
Gráfico # 18	119
Instalación SWI-Prolog	
Gráfico # 19	120
Instalación Java	
Gráfico # 20	121
Variables de Entorno	
Gráfico # 21	122
Propiedades servidor	
Gráfico # 22	123
Propiedades servidor	
Gráfico # 23	123
usuario puerto	

Gráfico # 24	124
Propiedades Administración	
Gráfico # 25	124
start	
Gráfico # 26	125
Paquetes	
Gráfico # 27	125
Paquete principal	
Gráfico # 28	135
Paquetes Proyecto tesis	
Gráfico # 29	142
Biblioteca Jar	
Gráfico # 30	143
añadir biblioteca	
Gráfico # 31	144
select	
Gráfico # 32	146
Empezar consulta	
Gráfico # 33	151
Ingreso al sistema	
Gráfico # 34	152
Datos Personales	
Gráfico # 35	153
Empezar consulta	

Gráfico # 36	153
Siguiente pregunta	
Gráfico # 37	154
GastroMedical concluye	
Gráfico # 38	155
Diagnóstico	
Gráfico # 39	155
Recomendaciones	
Gráfico # 40	156
Ver recomendaciones	
Gráfico # 41	156
Menú archivo	
Gráfico # 42	157
Menú Buscar	
Gráfico # 43	158
Submenú Acerca de	
Gráfico # 44	158
Presentación y versión	
Gráfico # 45	159
Presentar Búsqueda	
Gráfico # 46	160
Mostrar Datos Personales	
Gráfico # 47	160
Salir	

	Pág.
Gráfico # 48	161
Nueva Enfermedad	
Gráfico # 49	161
Guardar enfermedad	
Gráfico # 50	162
Cerrar el sistema	

INTRODUCCIÓN

Los sistemas de bases de datos inteligentes se derivan de la integración de la tecnología de bases de datos con técnicas desarrolladas en el campo de los sistemas expertos y la inteligencia artificial, que permite hacer deducciones a través de inferencias. Se basa principalmente en reglas y hechos que son almacenados en la base de datos MySQL adicionando a la estructura del motor relacional un motor de inferencia que habilita conjuntos de acciones disparadas por reglas y consultas ampliadas por estas inferencias.

La composición de la Base de Datos Inteligente consta de una base de datos relacional, una base de conocimientos, un motor de inferencia, y medios de interacción hombre-máquina de tal forma que el conjunto es visto como un único sistema inteligente.

El cual tiene como objetivo:

- Conseguir que la Base de Datos inteligente tenga capacidad de aprendizaje
- Diseñar el prototipo de Base de Datos Inteligente para que muestre su hipótesis y que justifique la línea de razonamiento al realizar una consulta medica.
- Tener la capacidad de expresar consultas por medio de reglas lógicas.

La implementación de una BDI (Base de Datos Inteligente), precisa del desarrollo de un sistema que posea la capacidad de gestionar conocimiento a través del procesamiento de datos simples almacenados en tablas controlados por un SGDB

(Sistema Gestor de Bases de Datos); incorporando la capacidad de utilizar la experiencia de un experto particular.

Se realizará la implementación de un prototipo de Base de Datos Inteligentes al que llamaremos “GastroMedical” para apoyar a los médicos Pediatría a determinar un diagnóstico en pacientes de 6 a 11 años con problemas Gastrointestinales, con una interfaz desarrolla a de manera tal que el usuario pueda disponer de ella cuando sus necesidades específicas de diagnóstico requieran información, el objetivo del prototipo es acreditarle validez al conjunto teórico investigado.

Se inicia con una descripción de las palabras claves del proyecto su definición y características. Seguidamente presentamos las diferentes patologías, sus síntomas en pacientes pediatras y las recomendaciones de la enfermedad. Posteriormente las características y arquitectura de la Base de Datos Inteligentes y su relación con Prolog, MySql y Java herramienta de software en la que se desarrollará el proyecto.

Luego, se hace una descripción del desarrollo e implementación del Sistema GastroMedical en la medicina detallando la metodología y los inconvenientes encontrados durante el proceso.

Por último, se presentan las conclusiones y recomendaciones para próximos adelantos.

CAPÍTULO I

EL PROBLEMA

PLANTEAMIENTO DEL PROBLEMA

UBICACIÓN DEL PROBLEMA EN UN CONTEXTO

Las bases de datos son accedidas generalmente por un conjunto de aplicaciones periféricas que incluyen tecnologías y herramientas de muy diverso tipo con los que interactúa y que sirven de interfaz entre los usuarios y la base de datos.

El presente trabajo presenta una nueva aplicación de los sistemas inteligente en el área de la medicina, integrando una base de datos relacional con una base de conocimiento. Su objetivo principal es desarrollar un prototipo de BDI basado en un sistema pediatra en enfermedades comunes de gastroenterología. Este prototipo presenta una alternativa de solución en caso de una emergencia médica de algún menor, específicamente hablando, la cual se basa en conocimientos básicos de la medicina y en enfermedades gastroenterológicas de los niños, así como también los diagnósticos de los síntomas, y sus posibles soluciones.

El prototipo de Sistema de Integración de Base de Datos y Base de Conocimiento que se pretende implementar siguiendo el planteamiento de esta propuesta, se usará como herramienta para el apoyo de las primeras consultas médica, la cual asistirá a los pediatra en brindar un mejor diagnostico clínico.

SITUACIÓN CONFLICTO NUDOS CRÍTICOS

Integrar conocimiento con bases de datos relacional fue sugerido por el desarrollo en el campo de la Inteligencia Artificial conocida como 'IA' en la programación de comportamientos inteligentes.

La investigación en IA incluye estudios sobre la representación de reglas lógicas que operen sobre los datos, esto es sistemas que descansan sobre una base de conocimientos para razonar sobre el conjunto de datos.

Históricamente, la evolución de los SGBD nos ha proporcionado métodos para analizar datos y encontrar correlaciones y dependencias entre ellos.

CAUSAS Y CONSECUENCIAS DEL PROBLEMA

El análisis de datos ha cambiado recientemente y ha adquirido una mayor importancia, debido principalmente a tres factores:

a) Incremento de la potencia de los ordenadores.

b) Incremento del ritmo de adquisición de datos. El crecimiento de la cantidad de datos almacenados se ve favorecido no sólo por el abaratamiento de los discos y sistemas de almacenamiento masivo, sino también por la automatización y técnicas de recogida de datos.

c) Han surgido nuevos métodos, principalmente de aprendizaje y representación de conocimiento, desarrollados por la comunidad de inteligencia artificial.

En esta Tesis se describe cómo el rendimiento de ciertos programas de Prolog se puede mejorar el almacenamiento de grandes listas de hechos en una base de datos MySql y no como hechos Prolog.

Actualmente existen numerosos casos por mala atención médica en dispensarios o centros ambulatorios públicos en especial en zonas rurales de la ciudad. Se propone un prototipo de sistema de base de datos inteligente aplicado a la medicina que emplee esta tecnología propuesta para realizar asistencia durante las primeras consultas médica en esos centros, el mismo que una vez investigado e implementado permitirá diagnosticar una enfermedad en el dominio y de esta forma reducir el índice de enfermedades pediatra en gastroenterología al tener un diagnóstico oportuno.

DELIMITACIÓN DEL PROBLEMA

El prototipo de base de datos inteligente se encargara de diagnosticar enfermedades gastrointestinales, con los diferentes síntomas que vaya reconociendo, estos síntomas serán nuestros hechos, de los cuales, con reglas lógicas, asociando diferentes tipos de síntomas podamos concluir en una enfermedad en especial, esto justifica el uso de la

programación lógica en swi-Prolog, esto implica que todo el conocimiento tanto del especialista (experto) como de las bibliografías tendrán que estar modelados.

No se debe generalizar o concluir que el prototipo GastroMedical hace una completa simulación del Experto Humano, y no está dicho en ningún momento que el prototipo reemplazará completamente al médico, sino que servirá de ayuda en el proceso de diagnóstico y evaluación de los síntomas en las primeras consultas.

El interés de los Sistemas de Bases de Datos Inteligente tiende a incrementarse conforme se amplía su campo de aplicación (Gestión, Sistemas Expertos, Medicina). Recomendados cuando expertos humanos son escasos, cuando es muy elevado el volumen de datos que ha de considerarse para obtener una conclusión

FORMULACIÓN DEL PROBLEMA

¿Es conveniente emplear un lenguaje de programación lógica para crear la base de datos inteligente aplicada en la medicina en el campo de pediatría gastrointestinal?

Entre los objetivos de la creación de la base de datos inteligente está la evolución de los lenguajes de programación lógica Prolog, como también el bajo “costo y facilidad” de programación que este puede ofrecer, ya que es un programa que usa la lógica para resolver los problemas. Por otra parte, el lenguaje está orientado para trabajar en aplicaciones de inteligencia artificial.

Otro punto importante, las bases de datos relacionales pueden proporcionar a prolog con gran cantidad de datos (Hechos). Para desarrolladores de bases de datos, es

bueno poder usar el motor de inferencia brindado por prolog. La combinación de Prolog, MySql con interfaz Java, puede convertirse en un sistema inteligente de Base de Datos.

Prolog nos permitirá modificar los datos de nuestra base de conocimientos durante la ejecución de la aplicación porque cuenta con predicados dinámicos el cual nos permite añadir o eliminar cláusulas en nuestra base de conocimientos durante la ejecución. Se pueden añadir y eliminar hechos o reglas de cualquier predicado que sea dinámico. Esto permitirá que el aplicativo obtenga nuevo conocimiento de un experto que en nuestro estudio es un médico pediatra.

EVALUACIÓN DEL PROBLEMA

Los aspectos generales de evaluación son:

Delimitado: El sistema GastroMedical recolectara todos los signos y síntomas que afectan al niño. El sistema pronosticara, el tipo de enfermedad, complicaciones, las recomendaciones a tomar para la enfermedad.

Claro: GastroMedical diagnosticar una patología en el área medica pediatra gastroenteróloga con técnica de preguntas y respuesta, la misma que de forma transparente integra una basa relacional con otra de conocimiento. El usuario del aplicativo se le mostrará como si fuese un solo sistema.

Factible: El aplicativo al ser desarrollado con herramientas Open Source nos permite reducir los costos que representaría la compra de programas licenciados. Las Herramientas empleadas cuenta con buena documentación de parte de sus fabricantes. También se contó con conocimientos de medicina, además de referencias bibliográficas de pediatría, para lo cual contaremos con un Medico que se encargara de brindar su conocimiento para el sistema.

Relevante: Este prototipo aporta una nueva forma de programar, y de beneficio social al disponer de un aplicativo que nos ayude a pronosticar la enfermedad cuando el experto medico es escaso.

Original: Como se dice en todo este documento la BDI en nuestro medio no está aun desarrollado, este trabajo deja las bases para la creación de nuevas tecnología de bases inteligentes.

Contextual: Se desarrolla con software de código abierto.

OBJETIVOS

OBJETIVO GENERAL:

Diseñar un prototipo que permita la Integración entre una base de datos relacional MySQL y un lenguaje de programación lógica de Prolog, utilizando un prototipo con una interfaz de usuario desarrollada en un lenguaje de Programación orientado a objeto Java para crear una Base de Datos Inteligente aplicada en el campo de la

Pediatría en enfermedades gastrointestinales para proveer sistema de ayuda a las consultas medicas de los usuarios devolviendo un diagnostico médico.

OBJETIVOS ESPECÍFICOS:

- Analizar y documentar los requerimientos de hardware y software que se van a necesitar para desarrollar el prototipo, que la BDI identifique la mejor elección dentro de una lista de posibilidades de una manera efectiva e inteligente.
- Conseguir que la Base de Datos inteligente tenga capacidad de aprendizaje para optimizar los resultados de las búsquedas.
- Diseñar el prototipo de Base de Dato Inteligente para que muestre soluciones alternativas y que justifique la línea de razonamiento al realizar una elección presente.
- Realizar las pruebas necesarias para detectar los posibles errores.
- Desarrollar una interfaz de usuario (prototipo) sencilla que permita introducir datos para realizar las búsquedas y generar conocimiento.

JUSTIFICACION E IMPORTANCIA

La razón que me llevo a plantear este tema de proyecto es porque voy a emplear herramientas de desarrollo que permita combinar dos formas diferentes de programar como lo es la programación convencional procedural y la programación lógica, sería de gran utilidad en la ingeniería de sistemas y la práctica clínica, sirviendo como

instrumento para la realización de futuros en este campo. Ya que en la actualidad, no han habido en nuestro medio desarrollos de sistemas que aporten una posible solución de este tipo de programación propuesta.

La documentación producto de la investigación total del proyecto se convierte en un gran pilar para promover el desarrollo de herramientas enmarcadas en el área de la Inteligencia Artificial y Base de Datos, ya que puede ser fácilmente consultado tanto por alumnos de la Carrera de Sistemas, o por cualquier otra persona interesada en el tema.

Los aportes del presente trabajo de investigación son los siguientes:

- En la parte de la Informática el aporte es el desarrollo de un Sistema utilizando la metodología Buchanan y lógica de predicados, mediante los cuales el motor de inferencia llega a una conclusión mediante reglas de inferencia como ser el Modus Ponens.
- Con el presente proyecto se pretende aportar una herramienta en el campo de la medicina que permita evaluar a los pacientes pediatra con problemas gastrointestinal y así brindar la asistencia oportuna para prevenir posibles empeoramiento de la salud.
- Se brinda una solución informática al problema presentado.
- Se beneficiaran los pacientes, los usuarios del sistema y a los desarrolladores de esta forma de programar que se interese por el tema.
- Reducir la cantidad de mortalidad de niños por enfermedades

gastrointestinales.

- Aumentar el conocimiento de las enfermedades de los niños en sus respectivos padres, para su respectivo tratamiento en casa.

En sectores populosos de cualquier ciudad es común encontrarse con enfermedades gastrointestinales que afectan a los niños, y no se cuenta con el conocimiento necesario para poder atender de manera adecuada dicho tipo de patología o infecciones que se puedan presentar en los menores, ya que por esos lugares el nivel económico es bajo en especial a aquellos lugares que no cuentan con recursos para un servicio médico; con lo que se genera el problema de tener que recurrir a un pediatra, lo que representa inconvenientes a las familias pobres. En algunos casos los padres suelen confundir este tipo de síntomas con algo simple y pasajero dolor de barriga y no toman las debidas medidas para la solución del mismo, que en el peor de los casos puede causar la muerte del niño.

Refiriéndonos más concisamente, toda esta problemática está fundamentada y centrada en el área de la Medicina Humana, dentro de la cual tratamos las enfermedades gastrointestinales.

Entonces la misión, será dar una solución informática empleando sistema de base de datos inteligentes, para poder facilitar los diagnósticos de las enfermedades que puedan sufrir dichos menores, esto implica una gran responsabilidad y precisión ya que es importante conocer exactamente la enfermedad que se pueda detectar. En este caso usaremos un Lenguaje de Programación lógica con una base de datos, utilizando una rama de la inteligencia artificial, que se encargará de almacenar, relacionar,

inferir y mostrar una solución en base a conocimientos recopilados de estos expertos.

El proyecto pretende realizar la implementación de un prototipo de Base de Datos Inteligentes al que llamaremos “GastroMedical” para apoyar a los especialistas médicos en Pediatría a determinar un diagnóstico de control en pacientes niños de 6 a 11 años con problemas gastrointestinales.

La programación declarativa permite describir la solución de un problema, y esto se logra a través de mecanismos de inferencia que, a través de la abstracción de conceptos, permiten evaluar distintas opciones para obtener una conclusión a partir de una premisa.

Se implementará una Base de Datos Inteligente para obtener los datos que serán asertados como hechos en una base de conocimiento prolog de esta forma lograremos una arquitectura de sistema inteligentes que ofrece soluciones para poder brindar un diagnóstico en el momento que sea requerido.

Las base de datos MySql nos brindara la integridad referencial en los datos que se necesitan y prolog nos dará la inferencia lógica para obtener un diagnostico medico que sirva para evaluar las consultas a través de síntomas.

CAPÍTULO II

MARCO TEÓRICO

ANTECEDENTES DEL ESTUDIO

En este proyecto se pretende integrar un conjunto de herramientas para la construcción de aplicaciones de Bases de Datos que operen a un nivel de deducción y conocimiento frente a las técnicas tradicionales de programación, para lograr un prototipo de software medico-pediatra que nos permita diagnosticar pacientes con síntomas gastrointestinales.

Dentro del campo universitario se realizaron trabajos de Tesis sobre “BASES DE DATOS INTELIGENTES” presentado por Sergio Antonio Becerra Zepeda de la Universidad de Colima – México, Facultad de Ingeniería Mecánica y Eléctrica. Este trabajo no tuvo aplicación en la medicina.

También tenemos un Sistema experto pediatra en enfermedades gastrointestinales realizado en el curso de programación lógica por: Henry Goicochea Miranda, Romer Vargas Otiniano y Barca Rafael de la Universidad Nacional de Trujillo en el cual desarrollaron dicho sistema netamente en swi prolog y XPCE que es una librería para crear interfaces, No integraron Bases de Datos ni interfaz de usuario.

En el trabajo de BDI propuesto se propone una integración de una Base Relacional MySql con una Base de Conocimientos desarrollada en un lenguaje de Quinta

Generación Prolog, además de un lenguaje orientado a objetos Java que me permite crear la interfaz de usuarios para la interacción hombre-máquina además es quien se encarga de armar esta integración en tiempo de ejecución a través de JDBC y una interfaz paquetería jpl.jar en el cual se ha emplean algunos de sus métodos. Las consultas y datos del paciente quedan registrados dentro de una base de Datos dando como resultado un Sistema de Integración de base de datos con conocimiento llamado GastroMedical.

FUNDAMENTACIÓN TEÓRICA

¿QUÉ SON BASES DE DATOS INTELIGENTES?

Para Agnes M. Rodríguez (1998), los sistemas de base de datos inteligente es:

“Los sistemas de bases de datos inteligentes se derivan de la integración de la tecnología de bases de datos con técnicas desarrolladas en el campo de la inteligencia artificial. Las bases de datos inteligentes se describen como una base de datos que contiene el conocimiento sobre el contenido de la información. Es un sistema de manejo de bases de datos el cual ejecuta la validación y el proceso”

En la definición que da Sergio Becerra (1999) acerca de este término base de datos inteligentes es:

“En las bases de datos inteligentes se concentra la evolución e integración de las bases de datos relacionales activas y las técnicas avanzadas para la búsqueda,

recuperación y procesamiento de la información en el área de los sistemas expertos y la inteligencia artificial”.

Técnicamente en una Base de Datos Inteligentes (BDI) converge la arquitectura de los Sistema Experto (SE) y sistemas basados en conocimiento por lo que para en su desarrollo se incluye como aspecto fundamental la programación del conocimiento. Algunas bases de datos inteligentes pueden incluir reglas que pueden ser aplicadas a la información entrante o saliente.

El trabajo presente se implementa un conjunto de herramientas para el diseño de aplicaciones con Bases de Datos que operen a un nivel de abstracción versus las técnicas tradicionales.

Para los efectos de nuestro estudio, la arquitectura de los sistemas expertos, constituye el eje sobre el que se ha de construir el modulo de gestión de base de conocimientos, esto es, la gestión de procesos deductivos de la base de reglas que opera sobre la base de datos.

Existen varias arquitecturas para el manejo de un sistema basado en el conocimiento:

- Añadir algunas facciones de bases de datos a un sistema experto.
- Añadir alguna inteligencia para un sistema de manejo de bases de datos
- Incorporar un sistema de manejo de bases de datos al alcance de un sistema experto.

Comúnmente un sistema experto consiste de dos componentes: una base de datos de hechos y reglas, conocido como la base de conocimiento, y un motor de inferencia,

que es un programa que puede aplicar esas reglas y hechos para buscar una solución "experta" a una pregunta.

Los sistemas de información y bases de datos inteligentes están desarrollándose dinámicamente en el campo de las ciencias de computadoras. Actualmente, se están reportando avances en los usos de los sistemas de bases de datos inteligentes los cuales se utilizan para la solución de problemas gerenciales, industria, ingeniería, administración, educación y en el caso que se está proponiendo en la medicina.

En este trabajo empiezan a vislumbrarse aplicaciones médicas pediatra como en el caso de consultas, diagnóstico y recomendaciones a pacientes con problemas gastrointestinales. Es importante aclarar que se entiende por consulta inteligente aquella que se realiza por medio de lógica de predicados es decir con procesos no procedurales.

La composición de la Base de Datos Inteligente consta de: una base de datos relacional, una base de conocimientos, un motor de inferencia, y medios de interacción hombre-máquina. Veamos algunos conceptos que nos ayudaran a comprender mejor el trabajo de investigación realizado en esta Tesis.

LAS CARACTERÍSTICAS DE LA PROPUESTA

Una Base de Datos Inteligente (B.D.I.) debe contar al menos con las siguientes características:

Tener la capacidad de expresar consultas por medio de reglas lógicas.

Tener la capacidad de definir reglas con las cuales deducir o inferir información adicional a partir de los hechos almacenados en la base de datos

Como característica fundamental de una Base de Datos Inteligente es la posibilidad de inferir información a partir de los datos almacenados.

GASTROMEDICAL se caracteriza por utilizar un modelo de producción basado en reglas con almacenamiento en tablas de la base de datos.

UTILIDAD PARA EL PAÍS, SECTOR, LA UNIVERSIDAD

La implementación de una Base de Datos Inteligente (BDI) que permita hacer esta diferenciación sería de gran utilidad en la toma de decisiones, en la práctica clínica, y quedan sentadas las bases para aplicaciones futuras de esta tecnología y arquitectura.

Finalmente, la documentación producto de la investigación total del proyecto GastroMedical, se convierte en un gran pilar para promover el desarrollo de herramientas enmarcadas en el desarrollo de la inteligencia artificial combinada con base de datos relacional para asegurar la integridad referencial de sus datos. Puede ser fácilmente consultado tanto por alumnos de la Carrera de Ingeniería en Sistemas, como por cualquier otra persona interesada en el tema.

Socialmente con el presente proyecto el aporte que se logra es importante ya que con este prototipo “GastroMedical” se llegara a poder diagnosticar a los niños si tienen problemas gastrointestinales, y así reducir el índice de estas enfermedades.

En Informática el aporte es la implementación de un prototipo de base de datos inteligente utilizando la metodología Buchanan (4) y lógica de predicados, mediante

los cuales el motor de inferencia llega a una conclusión mediante reglas de inferencia. La medicina y la ingeniería de sistemas son dos de las áreas del conocimiento humano que más desarrollo han presentado durante las últimas décadas.

BENEFICIARIOS

Los Sistemas de Base de Datos Inteligentes ofrecen una posible solución en un tiempo prudentemente corto, además, que la codificación del conocimiento y del razonamiento para realizar un diagnóstico efectivo del sistema puede producir beneficios adicionales.

En este trabajo se muestra una aproximación entre ingeniería de conocimiento y medicina de urgencias, trazando las líneas básicas para implementar un software completamente funcional que permita agilizar el proceso de consulta médica descrito en las líneas anteriores, trayendo beneficios no solo a las entidades médicas que, sino a la implementación de nuevas formas de programar.

Una vez cimentadas estas bases, queda abierto el camino para que el lector de este trabajo que esté interesado en este tema pueda continuar desarrollando el aplicativo, expandiéndolo hacia otras áreas de la medicina, o en otros campos o haciéndole mejoras que permitan obtener una mayor flexibilidad y robustez.

SISTEMA EXPERTO

Para Fernández Ríos Manuel (1999), que es citado por Alvarez Sequera, un sistema experto es:

“Es un tipo de software que imita el comportamiento de un experto humano en la solución de un problema, pueden almacenar conocimientos de expertos para un campo determinado, muy delimitado y solucionar un problema mediante deducción lógica de conclusiones”.

Un sistema experto (ES, siglas del término Expert System) es un sistema de información basado en el conocimiento de un área de aplicación compleja y específica a fin de actuar como un consultor experto para los usuarios finales. Los sistemas expertos proporcionan respuestas sobre un área problemática muy específica al hacer inferencias semejantes a las humanas sobre los conocimientos obtenidos en una base de conocimientos especializados. Además deben tener la capacidad de explicar al usuario su proceso de razonamiento y conclusiones. Así los Sistemas Expertos proporcionan apoyo a la toma de decisiones a los usuarios finales en la forma de consejo de un consultor experto acerca de un área problemática específica.

“Los sistemas expertos suelen estar diseñados de manera distinta a los programas convencionales, porque los problemas no tiene generalmente una solución algorítmica y dependen de inferencias para obtener una solución razonable. Como el sistema experto depende de la inferencia, debe ser capaz de explicar su razonamiento para que éste pueda verificarse" (Sistemas Expertos, principios y programación, capítulo 1, 2001:6).

En resumen un SE, es una clase de programa que es capaz de manejar problemas que normalmente necesitan de la intervención humana para su solución. Los sistemas

expertos son desarrollados con la ayuda de un Experto de Campo que son los que tienen la información de los procesos mentales, que le permiten solucionar los distintos problemas y un Ingeniero de Conocimiento que es quien formaliza el conocimiento del experto de campo, es decir, que le dan una forma simbólica y manipulable para el sistema.

Para resolver problemas se usan métodos heurísticos que determinan que parte de la experiencia de los expertos de campo son aplicables, es decir, que son procedimientos que se aprenden con la experiencia en la resolución de problemas en un dominio particular. Estas heurísticas deben ser descubiertas por ingeniero de conocimientos y programadas en el sistema expertos.

“Se denomina heurística a la capacidad de un sistema para realizar de forma inmediata innovaciones positivas para sus fines. La capacidad heurística es un rasgo característico de los humanos, desde cuyo punto de vista puede describirse como el arte y la ciencia del descubrimiento y de la invención o de resolver problemas mediante la creatividad y el pensamiento lateral o pensamiento divergente....” (¹).

¿Un Sistema Experto (SE) utiliza técnicas de la Inteligencia Artificial?

Los Sistemas Expertos son uno de los puntos que componen las investigaciones en el campo de la IA y para nuestra propuesta. Los SE son una expresión de los sistemas

¹ **Heurística**

http://www.taringa.net/posts/info/2379078/_Que-es-la-Heuristica_.html

basados en el conocimiento. Con la aplicación de técnicas de Inteligencia Artificial finaliza la transición del procesamiento de datos al procesamiento de conocimientos. Los sistemas expertos se aplican por norma general en problemas que implican un procedimiento basado en el conocimiento.

Características de los sistemas expertos:

1. Solucionan problemas aplicando su experiencia de una forma eficaz, haciendo deducciones a partir de datos incompletos o inciertos.
2. Explican y justifican lo que están haciendo..
3. Reestructuran y reorganizan el conocimiento.
4. Interpretan al mismo tiempo el espíritu y la letra de las reglas.
5. Determinan cuando un problema está en el dominio de su experiencia.

Los sistemas expertos conocidos modelan a lo sumo las tres primeras características.

Componentes de un sistema experto:

Base de conocimientos: Contiene el conocimiento de los hechos y las experiencias de los expertos en un dominio determinado.

Mecanismo de inferencia: Puede simular la estrategia de solución de un experto.

Componente explicativo: Explica al usuario la estrategia de solución encontrada y el por qué de las decisiones tomadas.

Interfase de usuario: Sirve para que este pueda realizar una consulta en un lenguaje lo más natural posible.

Componente de adquisición: Ofrece ayuda a la estructuración e implementación del conocimiento en la base de conocimientos.

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

Lenguajes de Programación de Sistemas Expertos

Mencionaremos dos lenguajes más importantes en el desarrollo de Sistemas Expertos:

LISP.- El nombre LISP es la abreviatura de List-Processing, ya que el LISP fue desarrollado para el procesamiento de listas. La lista es la estructura más importante de LISP. El lenguaje LISP fue diseñado ya a finales de los años 50 por McCarthy. A lo largo de los últimos años se han desarrollado muchos dialectos, por ejemplo MACLISP, COMMONLISP, INTERLISP, ZETALISP, donde el COMMONLISP se está imponiendo cada vez más como estándar.

PROLOG.- Es la abreviatura de PROgramación LOGica, con lo que hacemos mención a la procedencia del lenguaje: Es una realización de lógica de predicados, como lenguaje de programación. En la actualidad, el PROLOG se aplica como lenguaje de desarrollo en aplicaciones de Inteligencia Artificial en diferentes proyectos de Europa. En los Estados Unidos, el LISP está más extendido que el PROLOG. Pero para la mayoría de los terminales de trabajo de Inteligencia Artificial se ofrece también el PROLOG.

La lógica se representa en forma de predicados. Estos predicados aparecen en tres formas distintas: como hechos, como reglas y como preguntas. La lógica formulada como hechos y reglas se define como base de conocimientos. A esta base de conocimientos se le pueden formular preguntas. Los mecanismos importantes del PROLOG son: recursividad, instanciación, verificación, unificación, backtracking.

Ejemplos de Sistemas Expertos

MYCIN.- Sistema Experto para diagnósticos médicos, es un sistema experto para realizar diagnósticos; su función es la de aconsejar a los médicos en la investigación y determinación de diagnósticos en el campo de las enfermedades infecciosas en la sangre.

XCON.- Sistema Experto para configuración de Ordenadores. Según los deseos individuales del cliente se configuran redes de ordenadores VAX. Ya que el abanico de productos que se ofrecen en el mercado es muy amplio, la configuración completa y correcta de un sistema de estas características es un problema de gran complejidad.

Limitaciones de los sistemas expertos

- No siempre se encuentra disponible el conocimiento que se va a captar.
- Resulta difícil extraer la experiencia de los seres humanos.
- El enfoque de cada experto ante una situación tal vez es diferente, aunque correcto.
- Es difícil, incluso para un experto altamente capacitado, valorar con exactitud, las situaciones para bajo presiones de tiempo.

- Los usuarios de sistemas expertos tiene límites, cognoscitivos naturas, por lo que tal vez no aprovechen al máximo los beneficios del sistemas.
- Los sistemas expertos trabajan solo con áreas temáticas, definidas perfectamente, por el diagnostico de una falla, en una maquina.
- La mayoría de los expertos no tiene medios independientes de verificar si sus conclusiones son razonables o correctas.
- El vocabulario, o jerga, que utiliza los expertos para expresar hechos y relaciones, con frecuencia es limitado y no comprendido por otros.
- En la construcción de los sistemas expertos, a menudo se requiere la ayuda de ingenieros de conocimientos, los cuales escasean, además de que sus servicios resultan caros.
- La falta de confianza de los usuarios finales, tal vez sea una barrera para el uso de los sistemas expertos.
- La trasferencia de conocimiento se sujeta a prejuicios de percepción y de criterio.
- La responsabilidad por una mala asesoría proporcionada por un sistema experto resulta difícil de evaluar. (Tecnologías de de información para la administración; CESCA 1999: 569).

Representación del conocimiento en un sistema experto

Para diseñar y crear un sistema experto es necesario encontrar un área de interés, que en la presente propuesta esta se centra en el área de especialista pediatra en gastroenterología donde se seleccionan a los expertos o fuentes de información

especializada, que son los especialistas capaces de resolver los problemas en dicha área. Ahora casi siempre estos especialistas, son expertos en un dominio específico y es sobre este dominio, donde poseen su mayor experiencia.

Una vez seleccionado al experto o a los expertos o fuentes de información y estos estén de acuerdo en dar sus conocimientos, comienza a jugar su papel el "Ingeniero de Conocimientos", que es el encargado de extraerle los conocimientos al experto y darle una representación adecuada, ya sea en forma de reglas u otro tipo de representación, conformando así la base de conocimientos del sistema experto.

Formas de representación de los conocimientos:

- Reglas de producción
- Redes semánticas
- Marcos (Frames)

La forma de representación más usada es por reglas de producción, también llamadas reglas de inferencias. Casi todos los sistemas expertos están basados en este tipo de representación.

INGENIERIA DEL CONOCIMIENTO

En la definición que da D. Michie. Knowledge Engineering. Cybernetics, 2. 197-200. 1973 es: Es el proceso en el que se adquiere el conocimiento a partir de un especialista o de otra fuente y se codifica en el Sistema. y es llevada a cabo por un ingeniero de conocimiento.

El Ingeniero del Conocimiento.- El Ingeniero de Conocimiento o receptores de conocimiento, es o son las personas encargadas de modelar la habilidad en el dominio plantean las preguntas al experto y los que conducen a entrevistas con los expertos. estructura sus conocimientos y los implementa en la base de conocimientos.

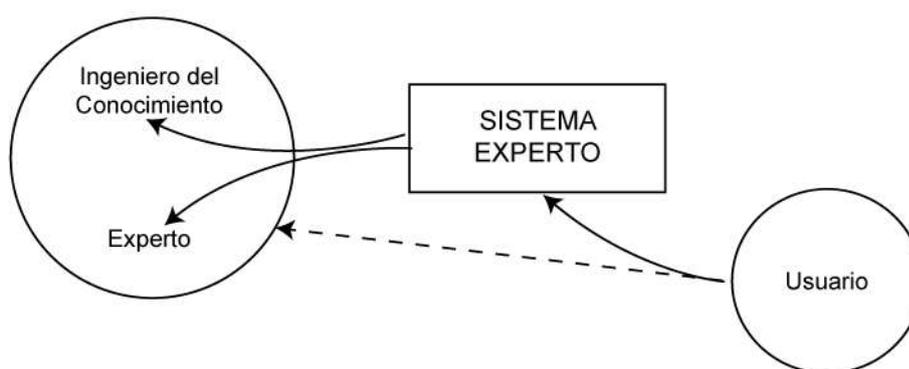


Grafico # 1 Etapas de un Sistema Experto

ADQUISICIÓN DE CONOCIMIENTO

Buchanan et al. (1983) define la adquisición de conocimiento de esta Forma:

“Es la transferencia y transformación de experiencia que puede usarse para resolver un problema”

Esta transferencia es llevada a cabo por una serie de entrevistas entre el ingeniero de conocimiento, que normalmente es un especialista en sistema, y el experto en el dominio de problemas que es capaz de articular su experiencia. Esta estimado que esta forma de hacerlo produce muy poca cantidad de conocimiento codificado en un intervalo de tiempo.

Esta poca eficiencia ha llevado a los investigadores a buscar otras formas de hacerlo

Estos problemas han llevado a los investigadores a tratar de automatizar el proceso de adquisición de conocimiento. Para hacer esto, lo primero que hay que entender es como se puede representar el conocimiento.

Metodología de Buchanan. [Buchanan y Shortliffe.,1984]

La metodología de Buchanan tiene como pilar básico la adquisición de conocimiento (de distintas fuentes, como ser libros, expertos, otros).

El Ingeniero de Conocimiento procede a través de una serie de etapas para producir un Sistema Experto de las cuales se destacan seis importantes etapas:

1. **Identificación.-** Etapa donde se identifican los participantes y roles, los recursos y las fuentes de conocimiento. Además se establecen: facilidades computacionales, presupuestos, objetivos o metas.
2. **Conceptualización.-** Se analizan los conceptos vertidos por el experto del campo. Los mismos serán tomados en cuenta con sumo interés, pues el experto humano o de campo es quién conoce en detalle los fundamentos particulares del tema a investigar.
3. **Formalización.-** Se identifican los conceptos relevantes e importantes. El objetivo es de formalizar el diagrama de información conceptual y los elementos de sub problemas en una especificación parcial para comenzar a construir un prototipo de la Base de Conocimiento.
4. **Implementación.-** Se formaliza el conocimiento obtenido del Experto y se elige la organización, el lenguaje y el ambiente de programación.

5. **Testeo.-** Se observa el comportamiento del prototipo, el funcionamiento de la Base de Conocimiento y la estructura de las inferencias.
6. **Revisión del prototipo.** Se refina el prototipo, o si fuera el caso, se rediseña y se reformulan los conceptos.

Los sistemas basados en el conocimiento definen un área muy concreta dentro de la inteligencia artificial, conocida como Ingeniería del conocimiento. Los sistemas desarrollados en este campo siguen un patrón heurístico similar al razonamiento humano.

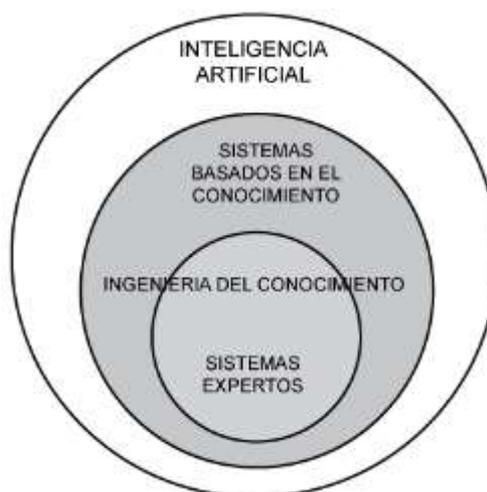


Grafico # 2 Campos de la Inteligencia Artificial

La parte mas compleja es armar nuestra base de conocimiento de la propuesta planteada en este Prototipo. Se pudo contar con al colaboración de dos especialista medico (ver anexos entrevistas). y de referencias medicas y listas oficiales de la Organización Mundial de la Salud (OMS), así como los Diagnóstico de Roma. Se llegó a concluir en los síntomas mas relevantes para

determinar cada una de las enfermedades mas importantes del Sistema GastroMedical donde se programó el conocimiento obtenido.

Inconvenientes:

- La adquisición del conocimiento es difícil y cara
- La reutilización del conocimiento en contextos diferentes no es simple.
- Obstáculos para el aprendizaje y la adaptación.

Por eso es importante combinarlos con otras tecnología como la Base de datos relacional y una interfaz Java para darle robustez a un sistema.

INTELIGENCIA ARTIFICIAL ‘IA’

Con el fin de discutir el concepto de sistemas de base de datos inteligente, primero es útil examinar brevemente qué se entiende por "inteligente".

La Inteligencia Artificial también conocida como la Quinta Generación, es una ciencia que trata de la comprensión de la inteligencia y del diseño de máquinas inteligentes, es decir, el estudio y la simulación de las actividades intelectuales del hombre razonamiento, percepción, aprendizaje, creación. Se denomina inteligencia artificial (IA) a la rama de las Ciencias de la Computación dedicada al desarrollo de agentes racionales no vivos.

“ Él termino “inteligencia artificial” fue acuñado formalmente en 1956 durante la conferencia de Dartmouth, más para entonces ya se había estado trabajando en ello durante cinco años en los cuales se había propuesto muchas definiciones distintas que en ningún caso habían logrado ser aceptadas

totalmente por la comunidad investigadora. La IA es una de las disciplinas más nuevas que junto con la genética moderna es el campo en que la mayoría de científicos más les gustaría trabajar.” (2).

Se define la inteligencia artificial como aquella inteligencia exhibida por artefactos creados por humanos (es decir, artificial). A menudo se aplica hipotéticamente a los computadores. El nombre también se usa para referirse al campo de la investigación científica que intenta acercarse a la creación de tales sistemas.

Debido a que la inteligencia artificial tuvo muchos padres no hay un consenso para definir ese concepto, pero podemos decir que la inteligencia artificial se encarga de modelar la inteligencia humana en sistemas computacionales.

Marvin Minsky (1968) Expresa **"El problema de la Inteligencia Artificial es la naturaleza de la misma inteligencia, un tema que nadie comprende muy bien. ¿Porqué No? Quizás en parte porque nadie ha tenido la oportunidad de estudiar otros tipos de inteligencia distintos de la humana"**

Minsky de forma concisa define la inteligencia artificial como **"La ciencia de hacer máquinas de hacer cosas que requieren de inteligencia si se hace por los hombres"**, mientras que Eysenck (1990) nos da una forma de avanzar en el dominio de bases de datos inteligente con la afirmación:

² **inteligencia artificial**

http://www.inteligenciaartificial.cl/ciencia/software/ia/inteligencia_artificial.htm

"La IA está preocupada por el intento de desarrollar programas informáticos complejos que será capaz de realizar tareas cognitivas difíciles."

Las tareas que una base de datos inteligente debe abordar son potencialmente muy difícil para una mente humana, para hacer frente a las tareas de este tipo implican la búsqueda y conseguir la información significativa a través de un conjunto grande de datos. Sería casi imposible para la mente humana inducir o deducir cualquier nuevo dato significativo de los repositorios de datos con la eficiencia y la velocidad de la máquina en una base de datos. Escritores recientes en el campo como Bertino, Catania y Zarri (2001) hacen de este matrimonio entre las dos tecnologías explícita, **"Base de datos de los sistemas inteligentes (IDBS) se derivan de la integración de la base de datos (DB), la tecnología con las técnicas desarrolladas en el campo de la inteligencia artificial (AI)"**. También señalan las debilidades inherentes de las tecnologías utilizadas de forma aislada, bases de datos tradicionales carentes de valor semántico y la incapacidad de los métodos de inteligencia artificial para hacer frente a grandes conjuntos de datos.

Historia de la Inteligencia Artificial

Los juegos matemáticos antiguos, como el de la Torres de Hanoi (aprox 3000ac), demuestran el interés por la búsqueda de un bucle resolutor, una IA capaz de ganar en los mínimos movimientos posibles.

En 1903 Lee De Forest inventa el triodo (tambien llamados bulbo o valvula de vacio). Podria decirse que la primera gran maquina inteligente diseñada por el hombre fue el

computador ENIAC, compuesto por 18.000 valvulas de vacio, teniendo en cuenta que el concepto de "inteligencia".

En 1937, el matemático inglés Alan Mathison Turing (1912-1953) publicó un artículo de bastante repercusión sobre los "Números Calculables", que puede considerarse el origen oficial de la Informática Teórica.

En este artículo, introdujo la Máquina de Turing, una entidad matemática abstracta que formalizó el concepto de algoritmo y resultó ser la precursora de las computadoras digitales. Con ayuda de su máquina, Turing pudo demostrar que existen problemas irresolubles, de los que ningún ordenador será capaz de obtener su solución, por lo que a Alan Turing se le considera el padre de la teoría de la computabilidad.

También se le considera el padre de la Inteligencia Artificial, por su famosa Prueba de Turing, que permitiría comprobar si un programa de ordenador puede ser tan inteligente como un ser humano.

En 1951 William Shockley inventa el transistor de unión. El invento del transistor hizo posible una nueva generación de computadoras mucho más rápidas y pequeñas.

En 1956, se acuñó el término "inteligencia artificial" en Dartmouth durante una conferencia convocada por McCarthy, a la cual asistieron, entre otros, Minsky, Newell y Simon. En esta conferencia se hicieron previsiones triunfalistas a diez años que jamás se cumplieron, lo que provocó el abandono casi total de las investigaciones durante quince años.

En 1980 la historia se repitió con el desafío japonés de la quinta generación, que dio lugar al auge de los sistemas expertos, pero que no alcanzó muchos de sus objetivos, por lo que este campo ha sufrido una nueva detención en los años noventa.

En 1987 Martin Fischles y Oscar Firschein describieron los atributos de un agente inteligente. Al intentar describir con un mayor ámbito (no solo la comunicación) los atributos de un agente inteligente, la IA se ha extendido a muchas áreas que han creado ramas de investigación enormes y diferenciadas.

Podemos entonces decir que la IA incluye características humanas tales como el aprendizaje, la adaptación, el razonamiento, la autocorrección, el mejoramiento implícito, y la percepción modelar del mundo. Así, podemos hablar ya no sólo de un objetivo, sino de muchos dependiendo del punto de vista o utilidad que pueda encontrarse a la IA.

Muchos de los investigadores sobre IA sostienen que "la inteligencia es un programa capaz de ser ejecutado independientemente de la máquina que lo ejecute, computador o cerebro".

El Futuro de la IA:

El cerebro humano tiene 100.000 millones de neuronas. Un programa de ordenador puede simular unas 10.000 neuronas. Si a la capacidad de proceso de un ordenador la sumamos la de otros 9.999.999 ordenadores, tenemos la capacidad de proceso de 10.000.000 ordenadores.

Multiplicamos 10.000.000 ordenadores por 10.000 neuronas cada uno y da = 100.000 millones de neuronas simuladas. Un cerebro humano será simulado en el futuro gracias a internet y cualquiera podría programarlo.

Una vez que la inteligencia artificial tenga una inteligencia igual o superior a la del hombre, obligatoriamente surgirá un cambio político y social, en el que la IA tiene todas las de ganar si se da cuenta que no necesita a los humanos para colonizar el universo. Suena a ciencia ficción pero actualmente orbitando están los satélites de comunicaciones con súper procesadores.

Técnicas y campos de la Inteligencia Artificial:

- Aprendizaje Automático (Machine Learning)
- Ingeniería del conocimiento (Knowledge Engineering)
- Lógica difusa (Fuzzy Logic)
- Redes neuronales artificiales (Artificial Neural Networks)
- Sistemas reactivos (Reactive Systems)
- Sistemas multi-agente (Multi-Agent Systems)
- Sistemas basados en reglas (Rule-Based Systems)
- Razonamiento basado en casos (Case-Based Reasoning)
- Sistemas expertos (Expert Systems)
- Sistemas basados en conocimientos.
- Bases de datos inteligentes (Intelligent Databases)
- Redes Bayesianas (Bayesian Networks)

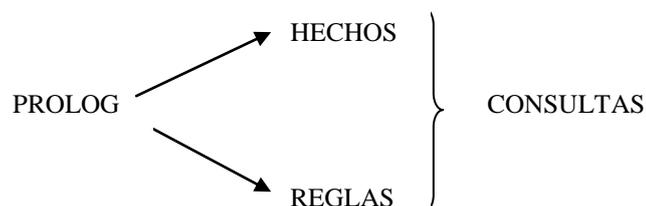
- Vida artificial (Artificial Life). La VA no es un campo de la IA, sino que la IA es un campo de la VA.
 - Computación evolutiva (Evolutionary Computation)
 - Estrategias evolutivas
 - Algoritmos genéticos (Genetic Algorithms)
- Técnicas de Representación de Conocimiento
 - Redes semánticas (Semantic Networks)
 - Frames
- Visión artificial
- Audición artificial
- Lingüística computacional
- Procesamiento del lenguaje natural (Natural Language Processing)
- Minería de datos (Data Mining)

PROLOG

Definición de Colmerauer Alain y Roussel Philippe (1992) sobre prolog:

“El Prolog (o PROLOG), proveniente del francés PROgramation et LOGique,] es un lenguaje de programación lógico e interpretado, bastante conocido en el medio de investigación en Inteligencia Artificial”.

Prolog es un lenguaje declarativo basado en Reglas y Hechos de lógica, cuya información es retribuido en forma de consultas. Originado en Europa a principios de los 70's por Alain Colmerauer.



Es un lenguaje declarativo e interpretado, en este tipo de lenguajes se representan los conocimientos sobre un determinado dominio y sus relaciones. A partir de ese conocimiento, se deducen las respuestas a las cuestiones planteadas, es decir se obtiene una inferencia.

El dominio lo constituye un conjunto de objetos. El conocimiento se formaliza mediante un conjunto de relaciones que describen de forma simultánea las propiedades y sus interacciones. Se declara el conocimiento disponible acerca de:

- Objetos:
 - propiedades
 - relaciones
- Reglas: que determinan interacciones lógicas, tales como: si ocurre A, entonces B; sino C.

Eddy Tuckler (1999) nos define prolog de la siguiente manera:

“Prolog es un lenguaje de programación hecho para representar y utilizar el conocimiento que se tiene sobre un determinado dominio. Más exactamente, el dominio es un conjunto de objetos y el conocimiento se representa por un conjunto de relaciones que describen las propiedades de los objetos y sus

interrelaciones. Un conjunto de reglas que describa estas propiedades y estas relaciones es un programa Prolog.

Prolog es un lenguaje de programación que es usado para resolver problemas que envuelven objetos y las relaciones entre ellos”.

Prolog por su naturaleza muestra una habilidad para describir gramáticas, en particular gramáticas libres de contexto.

Prolog representa el lenguaje principal en la categoría de Programación Lógica. A diferencia de otros lenguajes, Prolog no es un lenguaje de programación para usos generales, sino que está orientado a resolver problemas usando el cálculo de predicados.

Objetivos para el cual fue creado:

Entre los objetivos de la creación de Prolog está la evolución de los lenguajes, como también el bajo “costo y facilidad” de programación que este puede ofrecer, ya que como su nombre lo indica PROLOG (PROgramming in LOGic), es un programa que usa la lógica para resolver los problemas. Por otra parte, el lenguaje está orientado para trabajar en aplicaciones de inteligencia artificial.

PROLOG está orientado a la resolución de problemas mediante el cálculo de predicados, basado en:

- o Preguntas a la base de datos.
- o Pruebas matemáticas.

El programa PROLOG especifica cómo debe ser la solución, en vez de dar el algoritmo para su resolución. La solución se obtiene mediante búsqueda aplicando la lógica de predicados.

Predicados.- La lógica se representa en forma de predicados, está basada en la idea de las sentencias realmente expresan relaciones entre objetos, así como también cualidades y atributos de tales objetos. Los objetos pueden ser personas, objetos físicos, o conceptos. Tales cualidades, relaciones o atributos, se denominan predicados. Los objetos se conocen como argumentos o términos del predicado. Cada predicado está definido unívocamente por su nombre y su aridad.

Se utilizan para expresar propiedades de los objetos o relaciones entre objetos, así como también cualidades y atributos de tales objetos. Estos predicados aparecen en tres formas distintas: como hechos, como reglas y como preguntas. La lógica formulada como hechos y reglas se define como base de conocimientos. A esta base de conocimientos se le pueden formular preguntas.

Un predicado es la relación directa con una expresión. Cada predicado usado en una cláusula de Prolog debe ser declarado, basado en la declaración de los tipos de dominios para cada uno de los nombres de los objetos. Ejemplo:

paciente(nombre, edad, peso, presion_sanguínea).

colera(diarrea, tiene varias deposiciones diarias).

"Un predicado especifica la relación existente entre los argumentos del mismo.

El número de argumentos a los que se aplica dicho predicado se denomina aridad. Con un predicado podemos representar algo que sucede en el mundo

real (hecho), o una regla (regla de inferencia), que nos permite deducir hechos que suceden en ese dominio mediante la aplicación de la misma. La sintaxis de Prolog para especificar un predicado es la siguiente:

nombre_predicado(arg1,arg2,... , argN). cuando el predicado represente un hecho.

cabeza_predicado([[arg1], ... [argN]]):-cuerpo_predicado([[arg1], [arg2],..., [argN]]). (En el caso de que el predicado represente una regla de inferencia).

Las reglas de inferencia se pueden definir como la especificación de una relación entre predicados y argumentos que permiten plasmar el hecho de que si la parte derecha (cuerpo) del predicado se cumple, se cumple la parte izquierda (cabeza). El eje lo proporciona el símbolo ":-" que representa para nosotros la implicación lógica ” (³).

Prolog es adecuado para el análisis, y codificación de la lógica que se encuentran en analizar árboles. De hecho, Prolog viene con un analizador incorporado.

Historia de Prolog

Se trata de un lenguaje de programación de quinta generación ideado a principios de los años 70 en la Universidad de Aix-Marseille (Marsella, Francia) por los profesores Alain Colmerauer y Philippe Roussel. Nació de un proyecto que no tenía como objetivo la implementación de un lenguaje de programación, sino el procesamiento de

³ **predicado especifica...**

http://www.uhu.es/nieves.pavon/pprogramacion/temario/tema1/tema1.html#_Toc495042874

lenguajes naturales. Alain Colmerauer y Robert Pasero trabajaban en la parte del procesado del lenguaje natural y Jean Trudel y Philippe Roussel en la parte de deducción e inferencia del sistema. Interesado por el método de resolución SL, Trudel persuadió a Robert Kowalski para que se uniera al proyecto, dando lugar a una versión preliminar del lenguaje Prolog a finales de 1971 y apareciendo la versión definitiva en 1972. Esta primera versión de Prolog fue programada en ALGOL W.

Inicialmente se trataba de un lenguaje totalmente interpretado hasta que, en 1983, David H.D. Warren desarrolló un compilador capaz de traducir Prolog en un conjunto de instrucciones de una máquina abstracta denominada Warren Abstract Machine, o abreviadamente, WAM. Desde entonces Prolog es un lenguaje semi-interpretado.

Estructura de Prolog

Un programa Prolog está formado por una secuencia de enunciados (cláusulas): hechos, reglas y variables, operadores y comandos. Lo anterior se detalla a continuación

Hechos.- Expresan relaciones entre objetos. Supongamos que queremos expresar el hecho de que "un paciente tiene fiebre". Este hecho, consta de dos objetos, "paciente" y "fiebre", y de una relación llamada "tiene".

La forma de representarlo en prolog es: tiene(paciente, fiebre).

Algunas características de los hechos son:

- Los nombres de objetos y relaciones deben comenzar con una letra minúscula.
- Primero se escribe la relación, y luego los objetos separados por comas y encerrados entre paréntesis.

- Al final de un hecho debe ir el carácter "." (punto).

El orden de los objetos dentro de la relación es arbitrario, pero debemos ser coherentes a lo largo de la base de hechos.

Reglas.- Las reglas se utilizan en Prolog para significar que un hecho depende de uno o más hechos. Es la representación de las implicaciones lógicas del tipo $p \rightarrow q$ (p implica q).

Algunas características son:

- Una regla consiste en una cabeza y un cuerpo, unidos por el signo " :- ".
- La cabeza esta formada por un único hecho.
- El cuerpo puede ser uno o más hechos (conjunción de hechos), separados por una coma (","), que actúa como el "y" lógico.
- Las reglas finalizan con un punto (".").

Variables.- Representan objetos que el mismo PROLOG determina. Una variable puede estar instanciada o no instanciada. Esta instanciada cuando existe un objeto representado por una variable. De este modo, cuando preguntamos "Un paciente tiene X ?,"

Prolog busca en los hechos síntomas que tiene un paciente y respondería:

X = fiebre. Instanciando la variable X con el objeto fiebre.

Los nombres de variables comienzan siempre por una letra mayúscula.

Como ya nombramos las cláusulas haremos el siguiente razonamiento lógico:

tiempo(lluvioso) --> suelo(mojado)

suelo(mojado)

Que el suelo esté mojado, es una condición suficiente de que el tiempo sea lluvioso, pero no necesaria. Por lo tanto, a partir de ese hecho, no podemos deducir que esté lloviendo (pueden haber regado las calles). La representación correcta en

Prolog, sería:

suelo(mojado) :- tiempo(lluvioso).

suelo(mojado).

Cabe señalar que la regla esta "al revés". Esto es así por el mecanismo de deducción hacia atrás que emplea Prolog.

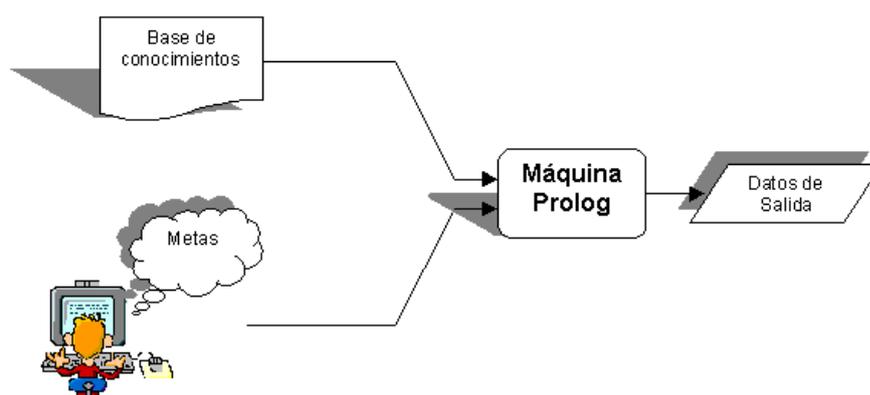


Grafico # 3 Esquema del funcionamiento de la máquina Prolog

Operadores y Comandos

Son predicados predefinidos en Prolog para las operaciones matemáticas básicas.

Su sintaxis depende de la posición que ocupen, pudiendo ser infijos o prefijos.

Por ejemplo el operador suma ("+"), podemos encontrarlo en forma prefija "+(2,5)" o bien infija, '2 + 5'.

También dispone de predicados de igualdad y desigualdad:

$X = Y$ igual; $X \neq Y$ distinto; $X < Y$ menor; $X > Y$ mayor; $X = < Y$ menor o igual;

$X > = Y$ mayor o igual.

consult. El predicado consult esta pensado para leer y compilar un programa Prolog o bien para las situaciones en las que se precise añadir las cláusulas existentes en un determinado fichero a las que ya están almacenadas y compiladas en la base de datos.

Su sintaxis puede ser una de las siguientes:

`consult(fichero); consult(fichero.pl); consult(c:\tesis\NetBeans\ fichero.pl).`

corte. El operador corte, representado por el símbolo "!" nos da un cierto control sobre el mecanismo de deducción del Prolog. Su función es la de controlar el proceso de reevaluación, limitándolo a los hechos que nos interesen. Supongamos la siguiente regla:

`regla :- hecho1, hecho2, !, hecho3, hecho4, hecho5.`

Prolog efectúa reevaluaciones entre los hechos 1, 2 sin ningún problema, hasta que se satisface el hecho2. En ese momento se alcanza el hecho3, pudiendo haber a continuación reevaluaciones de los hechos 3, 4 y 5. Sin embargo, si el hecho3 fracasa, no se intentará de ninguna forma reevaluar el hecho2.

Predicados dinámicos. Nos permite añadir o eliminar cláusulas en nuestra base de conocimientos durante la ejecución.

- Se pueden añadir y eliminar hechos o reglas de cualquier predicado que sea dinámico, aunque lo habitual es añadir o eliminar hechos solamente.

- Nos permitirán modificar los datos de nuestra base de conocimientos durante la ejecución de nuestros programas.

Existen distintos tipos de predicados dinámicos y cada uno de ellos nos permitirá modificar de una manera u otra nuestros programas en tiempo de ejecución:

1. **dynamic** Nos permite crear predicados dinámicos.

?- :-dynamic predicado/aridad

2. **assert** Nos permite introducir hechos de un predicado dinámico.

?- assert(predicado(Variable/s)).

3. **listing** Nos lista todo el contenido de la base de conocimientos actualmente.

Aunque nombre de forma distinta a las variables.

4. **retract** Elimina hechos existentes de un predicado de la base de conocimientos.

?- retract(predicado(Variable/s)).

5. **retractall** – Elimina todos los hechos existentes de un predicado de la base de conocimientos.

?- retractall(predicado(Variable/s)).

6. **abolish** Elimina todo absolutamente, tanto hechos como predicados dinámicos.

?- abolish predicado/aridad.

La persistencia: Es una característica de ciertos predicados dinámicos que encapsulan estado. El efecto principal de la declaración de un predicado persistente es

que los cambios dinámicos realizados se mantienen desde una ejecución a la siguiente.

Persistencia de datos: los datos deben mantenerse después de que la aplicación que los creó haya finalizado. El usuario no tiene que hacer copia explícitamente.

BASE DE DATOS RELACIONALES

Las bases de datos relacionales son las que han tenido más uso comercial. Están orientadas a registros y su concepto fundamental es la relación que es representada por una tabla. Una tabla es un conjunto de tupla de atributos o campos que serán las columnas y los datos almacenados que serán las sucesivas filas. La manera de acceder a cualquiera de ellos se realiza mediante un campo; el cual, será el campo clave y el valor de cada fila en esta columna será diferente.

En informática, o concretamente en el contexto de una base de datos relacional, un registro (también llamado fila o tupla) representa un objeto único de datos implícitamente estructurados en una tabla.

La información está centralizada, actualizada y es más sencillo realizar actualizaciones y copias de seguridad. La información puede estar en forma de texto, números, ficheros, XML, etc.

“Una base de datos es un “almacén” que nos permite guardar grandes cantidades de información de forma organizada para que luego podamos

encontrar y utilizar fácilmente. A continuación te presentamos una guía que te explicará el concepto y características de las bases de datos” (4).

El término de bases de datos fue escuchado por primera vez en 1963, en un simposio celebrado en California, USA. Una base de datos se puede definir como un conjunto de información relacionada que se encuentra agrupada ó estructurada.

Desde el punto de vista informático, la base de datos es un sistema formado por un conjunto de datos almacenados en discos que permiten el acceso directo a ellos y un conjunto de programas que manipulen ese conjunto de datos.

Cada base de datos se compone de una o más tablas que guarda un conjunto de datos. Cada tabla tiene una o más columnas y filas. Las columnas guardan una parte de la información sobre cada elemento que queramos guardar en la tabla, cada fila de la tabla conforma un registro. A las tablas se las denominaba “relaciones”, de ahí el nombre de base de datos relacional.

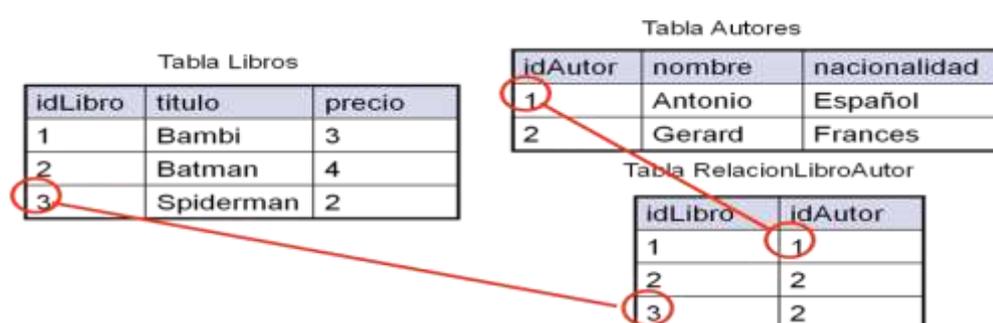


Grafico # 4 La información se relaciona mediante identificadores (id)

⁴ Una base de datos es un “almacén”...

<http://www.maestrosdelweb.com/principiantes/%C2%BFque-son-las-bases-de-datos/>

Características:

- Entre las principales características de los sistemas de base de datos podemos mencionar:
- Independencia lógica y física de los datos.
- Redundancia mínima.
- Acceso concurrente por parte de múltiples usuarios.
- Integridad de los datos.
- Consultas complejas optimizadas.
- Seguridad de acceso y auditoría.
- Respaldo y recuperación.
- Acceso a través de lenguajes de programación estándar.
- Simplicidad
- Generalidad
- Facilidad de uso para el usuario final
- Períodos cortos de aprendizaje
- Las consultas de información se especifican de forma sencilla
- Posibilidad de manejar una gran cantidad de información

¿Por qué utilizar una base de datos?

“ Una base de datos proporciona a los usuarios el acceso a datos, que pueden visualizar, ingresar o actualizar, en concordancia con los derechos de acceso que

se les hayan otorgado. Se convierte más útil a medida que la cantidad de datos almacenados crece” (5).

Una base de datos puede ser local, es decir que puede utilizarla sólo un usuario en un equipo, o puede ser distribuida, es decir que la información se almacena en equipos remotos y se puede acceder a ella a través de una red.

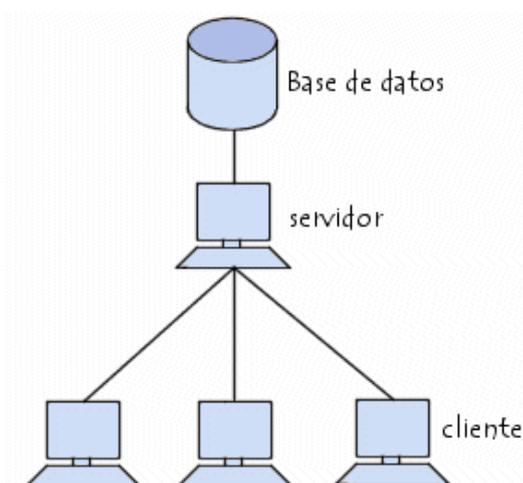


Grafico # 5 Base de Datos

La principal ventaja de utilizar bases de datos es que múltiples usuarios pueden acceder a ellas al mismo tiempo.

Componentes de una base de datos

- **Hardware:** constituido por dispositivos de almacenamiento como disco, tambores, cintas.
- **Software:** que es el DBMS o sistemas administración de la Base de Dato.

⁵ **Por qué utilizar una base de datos”...**
<http://es.kioskea.net/contents/bdd/bddintro.php3>

- **Datos:** los cuales están almacenados de acuerdo a la estructura externa y van a ser procesados para convertirse en información.

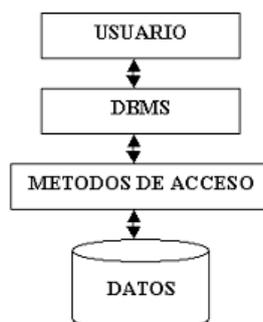


Grafico # 6 Componentes de una base de datos

Tipos de usuario de una Base de Datos

- **Usuario final:** Es la persona que utiliza los datos y solo ve los datos convertidos en información.
- **Desarrollador de aplicaciones:** es la persona que desarrolla los sistemas que interactúan con la Base de Datos. Además de convertir también modelar, diseñar sistemas B/D.
- **DBA (Data Base Administrator):** Es la persona que asegura integridad, consistencia, redundancia, seguridad. Este es el Administrador de la Base de Datos quien se encarga de realizar el mantenimiento diario o periódico de los datos (DBMS). El DBA trabaja con el DBMS (Herramientas que nos ayudan con la integridad de la seguridad de acceso).

SQL (Standard Query Language): Lenguaje de consulta estándar que permite:

Consulta de los datos seleccionando con diferentes criterios y realizando operaciones, inserción, actualización y borrado de la información, creación, alteración y borrado de las tablas y sus campos, Gestión de usuarios y sus privilegios de acceso.

Sus datos cumplen reglas de integridad:

- Las filas no tienen una posición concreta
- Un campo no puede ser una lista de valores
- No existen filas con todos los campos iguales (las haría indistinguibles y podría provocar problemas)
- Existe al menos una columna que identifica a la fila (id), denominada clave primaria (Primary Key)
- Se permiten valores NULL diferentes del 0 o de la cadena de caracteres vacía (para expresar dato no disponible).

Cumplen las siguientes leyes básicas:

- Una tabla sólo contiene un número fijo de campos.
- El nombre de los campos de una tabla es distinto.
- Cada registro de la tabla es único.
- El orden de los registros y de los campos no está determinados.
- Para cada campo existe un conjunto posible de valores.

Comandos.- Los comandos SQL se dividen en categorías:

Lenguaje de Manipulación de datos (DML): Obtiene, Inserta, Borra y actualiza datos select, insert, delete, update.

Lenguaje de Definición de datos (DDL): Crea, borra y cambia tablas, usuarios, vistas, índices... create table, drop table, alter table.

Sentencia select: También conocido como statement o query (consulta). Permite recuperar la información de una o varias tablas, especifica uno o más campos, una o más tablas y un criterio de selección.

La base de datos devuelve los campos indicados de aquellas filas que cumplan el criterio de selección.

Procedimientos Almacenados.- Un procedimiento almacenado es un grupo de sentencias SQL que pueden ser llamadas por su nombre. Admiten parámetros y se almacenan y se ejecutan en la base de datos.

Algunos productos tienen un lenguaje imperativo (PL/SQL de Oracle) y otros permiten implementarlos en Java.

Productos: Existen muchos productos de bases de datos, comerciales y software libre

MySQL (Software Libre) – <http://www.mysql.org>

Derby (Software Libre) - <http://db.apache.org/derby>

Oracle (Comercial) - <http://www.oracle.com>

MS SQL Server (Comercial) - <http://www.microsoft.com/sql>

Para nuestra propuesta se va emplear como base de Datos a MySQL 5.5.

Las principales características de MySQL

- Interioridades y portabilidad
- Probado con un amplio rango de compiladores diferentes
- Funciona en diferentes plataformas

- Usa GNU Automake, Autoconf, y Libtool para portabilidad.
- APIs disponibles para C, C++, Eiffel, Java, Perl, PHP, Python, Ruby, y Tcl.
- Uso completo de multi-threaded mediante threads del kernel.
- Proporciona sistemas de almacenamiento transaccionales y no transaccionales.
- Usa tablas en disco B-tree (MyISAM) muy rápidas con compresión de índice.
- Relativamente sencillo de añadir otro sistema de almacenamiento. Esto es útil si desea añadir una interfaz SQL para una base de datos propia.
- Un sistema de reserva de memoria muy rápido basado en threads.
- Joins muy rápidos usando un multi-join de un paso optimizado.
- Tablas hash en memoria, que son usadas como tablas temporales.
- Las funciones SQL están implementadas usando una librería altamente optimizada y deben ser tan rápidas como sea posible.

Cuadro # 1

Ventajas de las Bases de Datos ⁽⁶⁾

<i>Cuadro Resumen de las Ventajas de las Bases de Datos</i>	
Referidas	Ventajas
Los Datos	<ul style="list-style-type: none"> • Independencia de estos respecto de los tratamientos y viceversa. • Mejor disponibilidad de los mismos. • Mayor eficiencia en la recogida, codificación y entrada.

⁶ **ventajas / desventajas de las Bases de Datos...**
<http://html.rincondelvago.com/sistemas-de-bases-de-datos.html>

Los Resultados	<ul style="list-style-type: none"> • Mayor coherencia. • Mayor valor informativo. • Mejor y más normalizada documentación de la información.
Los Usuarios	<ul style="list-style-type: none"> • Acceso más rápido y sencillo de los usuarios finales. • Más facilidades para compartir los datos por el conjunto de los usuarios. • Mayor flexibilidad para atender a demandas cambiantes.

Cuadro # 2

Desventajas de las Bases de Datos

<i>Cuadro Resumen de las desventajas de las Bases de Datos</i>	
Relativas a	Desventajas
La implantación	<ul style="list-style-type: none"> • Costosa en equipos(lógico y físico). • Ausencia de estándares. • Larga y difícil puesta en marcha. • Rentabilidad a mediano plazo.
Los usuarios	<ul style="list-style-type: none"> • Personal especializado. • Desfase entre teoría y práctica.

Administración de bases de datos

Rápidamente surgió la necesidad de contar con un sistema de administración para controlar tanto los datos como los usuarios. La administración de bases de datos se

realiza con un sistema llamado DBMS (Database Management System [Sistema de Administración de Bases de Datos]). El DBMS es un conjunto de servicios (aplicaciones de software) para administrar bases de datos, que permite:

- un fácil acceso a los datos
- el acceso a la información por parte de múltiples usuarios
- la manipulación de los datos encontrados en la base de datos (insertar, eliminar, editar)

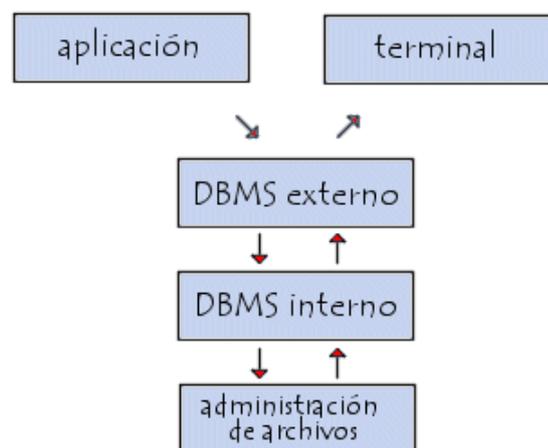


Grafico # 7 DBMS Subsistemas

El DBMS puede dividirse en tres subsistemas:

- El sistema de administración de archivos: para almacenar información en un medio físico.
- El DBMS interno: para ubicar la información en orden.
- El DBMS externo: representa la interfaz del usuario.

SISTEMA DE GESTIÓN DE BASE DE DATOS (SGBD)

“ Los sistemas de gestión de bases de datos en inglés Database Management System, abreviado DBMS son un tipo de software muy específico, dedicado a servir de interfaz entre la base de datos, el usuario y las aplicaciones que la utilizan, un DBMS (Sistema manejador de base de datos) otorgan la posibilidad de acceso, modificación, control y seguridad de la información almacenada en la base de datos ” (7) .

“SGBD: Conjunto coordinado de programas, procedimientos, lenguajes, etc. Que suministra tanto a usuarios no informáticos como a los analistas, programadores o al administrador, los medios necesarios para describir, recuperar y manipular los datos, manteniendo su integridad, confidencialidad y seguridad” (8) .

Funciones del SGBD:

De descripción o definición: Especificar los datos que la integran, estructura y relaciones entre ellos, reglas de integridad semántica, controles de acceso, así como las características físicas y lógicas. Esta función la realiza el Lenguaje de Definición de Datos, propio del SGBD.

De manipulación: Permite a los usuarios buscar, eliminar o modificar los datos de la base, de acuerdo a las normas de seguridad, lo que se realiza mediante el Lenguaje de

⁷ Sistema de Gestión de Base de Datos...

<http://www.maestrosdelweb.com/principiantes/%C2%BFque-son-las-bases-de-datos/>

⁸ SGBD: Conjunto coordinado...

<http://html.rincondelvago.com/sistemas-de-bases-de-datos.html>

Manipulación de Datos, mediante un Lenguaje Huésped, admitido por un Lenguaje Anfitrión; o bien por un Lenguaje Auto contenido.

De utilización: Reúne todas las interfaces que necesitan los diferentes tipos de usuarios para comunicarse con la base y proporciona un conjunto de procedimientos para el administrador.

Clasificación de los SGBD:

Esta clasificación está basada en el modelo de datos en que está basado el SGBD. Los modelos de datos más habituales son:

- Relacional (SGBDR): representa a la base de datos como una colección de tablas. Estas bases de datos suelen utilizar SQL como lenguaje de consultas de alto nivel.
- Orientado a objetos: define a la base de datos en términos de objetos, sus propiedades y sus operaciones. Todos los objetos que tienen la misma estructura y comportamiento pertenecen a una clase y las clases se organizan en jerarquías.
- Objeto-relacional o relacional extendido: son los sistemas relacionales con características de los orientado a objetos.
- Jerárquico: representa los datos como estructuras jerárquicas de árbol.

PEDIATRÍA

“La pediatría es la rama de la medicina que se especializa en la salud y las enfermedades de los niños. Se trata de una especialidad médica que se centra en

los pacientes desde el momento del nacimiento hasta la adolescencia, sin que exista un límite preciso que determine el final de su validez.

En la antigüedad la pediatría no existía como una disciplina médica, sino que la salud de los niños quedaba bajo responsabilidad de sus madres y comadronas. A partir del Renacimiento comenzó a considerarse a las enfermedades de los niños como una especialidad médica y, recién en el siglo XIX, la pediatría se convirtió en una rama científica ” (9).

“La palabra “pediatría” y sus afines significa “sanador de los niños”, sino que deriva de dos palabras griegas: (“El País”= niño) y (“iatros” = médico o curandero).

Pediatría es una especialidad médica relativamente nueva, el desarrollo sólo en la mitad del siglo 19. Abraham Jacobi (1830-1919) es conocido como el padre de la pediatría, debido a sus muchas contribuciones al campo.

Pediatría difiere de la medicina de adultos en muchos aspectos. Las diferencias son el tamaño del cuerpo obvio paralelo con los cambios de maduración. El cuerpo más pequeño de un bebé o recién nacido es sustancialmente diferente fisiológicamente a la de un adulto. Los defectos congénitos, la variación genética y los problemas del desarrollo son de mayor preocupación para los pediatras que a menudo son los médicos de adultos.

⁹ La pediatría
<http://definicion.de/pediatria/>

El tratamiento de un niño no es como tratar a un adulto en miniatura. Una diferencia importante entre la pediatría y la medicina de adultos es que los niños son menores de edad y, en la mayoría de las jurisdicciones, no puede tomar decisiones por sí mismos.

Las cuestiones de la tutela, la privacidad, la responsabilidad jurídica y el consentimiento informado siempre deben tenerse en cuenta en todos los procedimientos pediátricos. En cierto sentido, los pediatras a menudo tienen que tratar a los padres y, a veces, la familia, y no sólo el niño.

Los adolescentes están en su clase jurídica propia, que tenga el derecho a sus propias decisiones de atención de salud en determinadas circunstancias ” (10).

Existen diversas especialidades dentro de la pediatría, como la gastroenterología que es el caso de nuestro estudio mediante el sistema ‘GastroMedical’, la gastroenterología que es una rama de la medicina que se ocupa del estómago y de los intestinos y de sus enfermedades. Por extensión, se aplica también a la rama médica que se ocupa de todo el aparato digestivo y de sus enfermedades, de acuerdo a la Organización Mundial de la Salud (OMS), la salud es el estado de completo bienestar físico, mental y social.

¹⁰ **La palabra ‘pediatría’ ...**

<http://www.news-medical.net/health/Pediatrics-What-is-Pediatrics-%28Spanish%29.aspx>

En la presente propuesta se puede contar con la ayuda de expertos humanos, en este caso especialistas pediatras que puedan aportar con conocimiento que no cuente el sistema.

GASTROENTEROLOGIA

“Especialidad médica que estudia el sistema digestivo humano y las enfermedades que le afectan. Los especialistas en ese campo, llamados gastroenterólogos, estudian y tratan trastornos del esófago, el estómago, el intestino delgado, el intestino grueso y el colon lo que, en conjunto, se conoce como tracto gastrointestinal, así como del hígado, la vesícula biliar y otros órganos implicados en la digestión” ⁽¹¹⁾.

“Es una subespecialidad de la medicina interna. Generalmente el gastroenterólogo es también endoscopista, o sea realiza endoscopia digestiva (esofagogastroduodenoscopia, colangiopancreatografía retrógrada endoscópica, colonoscopia, rectosigmoidoscopia y rectoscopia).

Hay ramas de la gastroenterología que se focalizan dedican a un organo digestivo específico, como ejemplos:

- **Hepatología: Enfermedades del hígado y las vías biliares. Recientemente hay tienen un papel importante en el manejo de pacientes que van a recibir un trasplante de hígado.**

¹¹ **"Gastroenterología." Microsoft® Student 2009 [DVD]. Microsoft Corporation**

- **Endoscopía terapéutica:** Además de hacer diagnóstico de imágenes de las enfermedades digestivas, ofrece tratamiento a las mismas. Algunos ejemplos son: detener un sangrado de una úlcera péptica o de unas várices esofágicas, extraer cálculos que obstruyan la vía biliar, reseca tumores tempranos y pólipos, colocación de gastrostomía percutánea endoscópica (PEG), etc.
- **Proctología:** Enfermedades del recto (hemorroides, fisuras, tumores, abscesos y fístulas) ” (12).

ENFERMEDADES GASTROINTESTINALES: Las enfermedades gastrointestinales ocupan una de las primeras causas de consulta médica en niños. Son ocasionadas por varios motivos que pueden ser desde orgánicos y psicológicos, pero principalmente son causadas por bacterias, virus o parásitos que penetran al organismo por medio de alimentos y agua contaminada principalmente con materia fecal. Entre los principales microorganismos que las ocasionan están: la Salmonella, la Escherichia coli, la Shigella, las Giardias, las temibles amibas, Ascaris lumbricoides, etc. Las principales manifestaciones son:

- Fiebre
- Dolor estomacal o abdominal (cólicos)
- Náuseas
- Vómito

¹² " **Es una subespecialidad**
<http://www.dgsm.unam.mx/web/gastro.html>

- Diarrea
- Estreñimiento

Una de las consecuencias y complicaciones más graves cuando hay diarrea y vómito, es la deshidratación.

Los órganos que son afectados con mayor frecuencia son: el esófago, el estómago, el duodeno, el ano, el recto, el páncreas y los intestinos, el delgado y el grueso.

Entre los estudios para identificar exactamente el tipo de problema, están los de sangre, materia fecal, endoscopías, radiografías y ecografías, además de la exploración física y la historia clínica.

Entre las enfermedades más comunes, están: la Colitis, el Reflujo Gastroesofágico, el Colon irritable, la Hepatitis C, la Salmonelosis, Amibiasis, Gastritis, Úlceras, Cálculos biliares, Agruras o pirosis y otras más.

Todas estas enfermedades se previenen con:

- Higiene y preparación adecuada de alimentos.
- Evitar comer en la calle o en lugares poco higiénicos, cerca de animales o de personas cuya higiene evidentemente es mala.
- Lavar las manos antes de comer y después de ir al baño.
- Hervir el agua y dejarla reposar durante media hora o tomar agua embotellada.
- Lavar bien frutas y verduras y desinfectarlas con unas gotitas de cloro antes de enjuagarlas bien.

- Revisar bien las fechas de caducidad y calidad de los empaques de los alimentos que consumimos.
- Comer tres veces al día, sin exagerar e incluyendo alimentos de los tres grupos, frutas y verduras que contengan fibra y tomar dos litros de agua diariamente) ” (13).

COMPONENTES DEL SISTEMA GASTROMEDICAL

Como todo sistema inteligente, GastroMedical cuenta con un conjunto de elementos que a continuación describimos:

BASE DE CONOCIMIENTOS: Contiene el conocimiento sobre el dominio, consiste en las reglas de solución de problemas presentadas en forma de un Árbol de Conocimiento, procedimientos y datos intrínsecos al área del problema. La base, contiene el conocimiento del sistema con respecto a un tema específico para el que se diseña el sistema.

Este conocimiento se obtuvo en visitas a página de internet como la Organización Mundial de la Salud, (14) de Enciclopedia Encarta, y entrevista con el experto médico de las cuales se determinaron las principales y más comunes enfermedades gastroenterológicas aplicando el método Buchanam, sus características distintivas, sus rasgos comunes y las diferentes formas en que la enfermedad suele presentarse. El

¹³ **Enfermedades gastrointestinales...**

<http://www.esmas.com/salud/enfermedades/infecciosas/434933.html>

¹⁴ **OMS**

<http://www.who.int/topics/es/>

<http://www.who.int/mediacentre/factsheets/fs107/es/index.html>

conocimiento se representa típicamente como un conjunto de reglas y hechos formando una Base de conocimientos.

Una base de conocimientos es creada por ingenieros de conocimiento, quienes trasladan el conocimiento de expertos humanos a predicados, reglas y estrategias. Luego de un largo trabajo de mesa se obtuvieron las siguientes patologías ⁽¹⁵⁾ de interés para nuestro prototipo GastroMedical:

1. Apendicitis
2. Ascariasis
3. Cólera
4. Diarrea
5. Gastroenteritis
6. Síndrome de Intestino Irritable o Colitis Nerviosa
7. Estreñimiento
8. Gastritis
9. Intoxicación Alimentaria

BASE DE HECHOS (MEMORIA DE TRABAJO): Contiene los hechos sobre el problema que se han descubierto durante la consulta.

Durante la consulta con el sistema GastroMedical, el usuario responde a las preguntas que se encuentran como una aserción en la base de hechos. El sistema empareja esta

¹⁵ patología s. f. Parte de la medicina que estudia la naturaleza de las enfermedades, especialmente de los cambios estructurales y funcionales de los tejidos y órganos que las causan.

<http://es.thefreedictionary.com/patolog%C3%ADas>

información con el conocimiento disponible en la base de conocimientos para deducir nuevos hechos.

Nuestra Base de Hechos está almacenada en la Base de Datos y asertada en SWI-Prolog y comprende el listado de alarmas que provienen del módulo de Captura de Datos (Preguntas), realiza ciclos hasta obtener respuesta al análisis de causas, genera una respuesta durante el proceso de inferencia y conclusiones intermedias generadas durante la ejecución del programa.

Ejemplos de la Base de Hechos (tipo pregunta) del sistema GastroMedical.

SGDB: MySQL Query Browser

Base: Tesis

Tabla: esPregunta

('alimentoDudosaProceden', 'puede haber comido alimento de dudosa procedencia');

('alimentoMalCocido',' puede haber comido alimentos mal cocinados');

('calambresMusculares',' tiene calambres musculares');

('dolorAbdominal',' tiene dolor abdominal');

('evacuacionAbundante',' tiene deposición (diarrea) abundante');

('evacuacionGrisaseo',' tiene deposiciones pardo/grisaseas como agua de arroz');

('evacuacionIntestinal',' tiene deposiciones amarillentas');

('evacuacionLiquida',' tiene deposiciones líquidas');

('sedExcesiva',' tiene mucha sed');

('vomitoProfundoNauseas',' tiene vómito profundo con nauseas');

Es posible consultar los datos objetivo que serán transformados al formato jpl como hechos, dichos datos, serán el producto de varias consultas y se redireccionan hacia un archivo pl a partir del que se edificará la base de datos en intenso, sobre los que se lleva a cabo el proceso de inferencia.

Dicha base de hechos, se crea y destruye según sea preciso en las operaciones generales del prototipo GastroMedical, las operaciones sobre ésta las efectúa un archivo pl compilado en Prolog. Las bases en intenso se manipulan conforme a un conjunto determinado de reglas que constituyen el núcleo del motor de inferencia. Todos los ejemplos en que interviene el motor de Prolog tienen la misma base, sus aspectos principales:

- a) **La Base Persistente.** Perdura en toda la ejecución del programa, no se pierde.

Compuesta por la tabla: esPregunta

```
table esPregunta(
    sintoma varchar(40),
    descripcion varchar(100) );
```

- b) **La base en intenso.** Compuesta por la estructura:

```
esPregunta(carac,'sintomas_segun_sus_caracteristicas').
```

- c) **El objetivo de la inferencia.**

Extraer las preguntas entre los hechos para compararla con la reglas e inferir si es correcto devuelve una afirmación o un false en caso de que no encontrarse en la base de hechos.

- d) **El repositorio de reglas.**

El repositorio de reglas está escrito en Lenguaje SWI-Prolog, compilado en el archivo `pediatra.pl`.

EL MOTOR DE INFERENCIA: Modela el proceso de razonamiento humano trabajando con la información contenida en la base de conocimientos y la base de hechos para deducir nuevos hechos, es un tipo de programa que esta entre el usuario y una base de conocimiento, en un sistema. Opera sobre una base de datos en intenso en función de un repositorio de regla; todas las acciones del motor responden a eventos predefinidos que se disparan desde la aplicación Java, en este proceso - como se ha descrito anteriormente - interviene una secuencia ordenada de pasos, desde la selección de los datos objetivo hasta los productos del proceso de inferencia. Así, el flujo y procesamiento de datos está regulado por el programa de aplicación, éste genera la base en intenso dentro del código prolog en tiempo de ejecución, inicializa el motor de inferencia y a través de él consulta la base citada.

REGLAS: Entiéndase por regla a una proposición lógica que relaciona dos o más proposiciones lógicas: la premisa y la conclusión. Una regla es una estructura condicional que relaciona lógicamente la información contenida en la parte del antecedente con otra información contenida en la parte del consecuente.

Uno de los modelos comunes involucra los métodos de encadenamiento de reglas para formar una línea de razonamiento. Backward Chaining es un algoritmo de búsqueda a lo largo de las reglas para inferir conclusiones a partir de los hechos y las reglas. Todos los algoritmos van disparando reglas a medida que se cumplen las

condiciones. Orientado por el Objetivo o Encadenamiento hacia atrás Si la conclusión es conocida pero la ruta de la conclusión no es conocida (condiciones), entonces el método que se utiliza es el encadenamiento hacia atrás.

INTERFAZ GRÁFICA: Está desarrollada en Java Net Beans IDE 6.5 y permite al usuario interactuar con el aplicativo.

El Sistema GastroMedical está inspirado inicialmente en las técnicas desarrolladas en Inteligencia Artificial en el marco de los sistemas “Pregunta - Respuesta”, adaptándolas a las limitaciones específicas de las Bases de Datos.

La medicina basada en la evidencia, ⁽¹⁶⁾ o en pruebas es la manera de abordar los problemas clínicos, utilizando los resultados de la investigación científica. Formular de manera precisa una pregunta a partir del problema clínico del paciente. Consiste en convertir las interrogantes que surgen durante el encuentro clínico, y que percibimos como necesidades de información, en una pregunta, simple y claramente definida.

El aplicativo como se dijo se implementará con el lenguaje JAVA para la programación, y MySQL en el montaje de la base de datos de información. La conexión entre JAVA y MySQL se hará por medio de JDBC. Para la base de conocimiento de gastroenterología se utilizará PROLOG, y para el enlace entre PROLOG y JAVA se utilizará la paquetería de software libre JPL.JAR.

¹⁶ **Medicina basada en la evidencia**

<http://www.infodoctor.org/rafabravo/torpes.htm>

<http://www.infodoctor.org/rafabravo/mbe2.htm>

Con el trabajo de desarrollo de este aplicativo se demuestra que es posible seguir extendiendo el uso de los sistemas inteligente a áreas específicas de la medicina, en este caso a la gastroenterología, a través de herramientas de software gratuitas, las cuales pueden ser otorgadas por sus fabricantes sin restricción alguna.

Para esto, se hará uso del DBMS MySQL 5.0 al igual que SWI-Prolog 5.6, como se explica en la fundamentación legal de este capítulo al ser software libre permite ser utilizado sin necesidad de recurrir a licencias comerciales o académicas, solo se requiere el recurso humano, el cual consiste de un ingeniero de sistemas y un médico especializado, apoyados de ser necesario en bibliografía recomendada.

La versatilidad de JAVA, su portabilidad y la gran existencia de recursos tecnológicos creados para este lenguaje de programación, fueron factores clave en su elección para implementar este prototipo.

NetBeans IDE 6.5, permite agilizar el uso del paquete Swing para la creación de toda la interfaz de usuario del Sistema GastroMedical, obteniendo de esta manera una ganancia en tiempo y efectividad de la programación en Java.

El DBMS MySQL 5.0 se ha convertido en una de las más populares bases de datos de fuente abierta debido a su consistente y rápido desempeño, su alta confiabilidad y su fácil uso. Es usada en más de 11 millones de instalaciones que van desde grandes corporaciones hasta aplicaciones embebidas especializadas a través de todo el mundo. MySQL corre en mas de 20 plataformas que incluyen Linux, Windows, OS/X, AIX, entre otras.

En este trabajo se usará un sistema prototipo que se ejecuta sobre un intérprete de Prolog medico para demostrar la integración de estas dos bases y obtener una BDI.

SWI Prolog, un sistema de acceso libre que se encuentra disponible en la página:

<http://www.swi-prolog.org/download/stable>

El DBMS MySQL 5.0 se encuentra disponible en la página:

<http://dev.mysql.com/downloads/>

En estas páginas se puede encontrar el manual de prácticas que incluye instrucciones sobre la instalación del sistema.

FUNDAMENTACIÓN LEGAL

Términos Legales Ecuatorianos.- El artículo 349 de la Constitución de la República del Ecuador, norma que es garantía del Estado, entre otras, mejorar una formación continúa y mejoramiento pedagógico y académico del personal docente.

Art. 349.- El sistema de educación superior tiene como finalidad la formación académica y profesional con visión científica y humanística, la investigación científica y tecnológica; la innovación, promoción, desarrollo y difusión de los saberes y las culturas, la construcción de soluciones para los problemas del país, en relación con los objetivos del régimen de desarrollo.

Art. 387.- El Estado destinará los recursos necesarios para la investigación científica, el desarrollo tecnológico, la innovación, la formación científica, la recuperación y desarrollo de saberes ancestrales y la difusión del

conocimiento. Un porcentaje de estos recursos se destinara a financiar proyectos mediante fondos concursales.

Reglamento Codificado de Régimen Académico del Sistema Nacional de Educación Superior

Art. 27. 2 Para la obtención del grado académico de Licenciado o del Título Profesional universitario o politécnico, el estudiante debe realizar y defender un proyecto de investigación conducente a una propuesta para resolver un problema o situación práctica, con características de viabilidad, rentabilidad y originalidad en los aspectos de acciones, condiciones de aplicación, recursos, tiempos y resultados esperados.

Con fecha 10 de abril del 2008, el Presidente de la República de Ecuador, Rafael Correa, mediante decreto No. 1014 establece como política para las entidades de la Administración Pública Central, el uso de software libre en sus sistemas y equipamientos informáticos. Es interés del Gobierno de Ecuador alcanzar soberanía y autonomía tecnológica, así como un ahorro de recursos públicos. El Software Libre le permite al Estado mayor seguridad informática, libre acceso a datos y programas, ahorro en costos de licencias y es un generador de empleo para profesionales ecuatorianos.

DECRETO 1014

Art. 1.- Establecer como política pública para las entidades de la Administración Pública Central la utilización de Software Libre en sus sistemas y equipamientos informáticos.

Art. 3.- Las Entidades de la Administración Pública Central previa a la instalación del software libre en sus equipos, deberán verificar la existencia de capacidad técnica que brinde el soporte necesario para el uso de este tipo de software.

Art. 5.- Tanto para software libre como software propietario, siempre y cuando se satisfagan los requerimientos, se debe preferir las soluciones en este orden:

Nacionales que permitan autonomía y soberanía tecnológica

Regionales con componente nacional

Regionales con proveedores nacionales

Internacionales con componente nacional

Internacionales con proveedores nacionales

Internacionales

HIPÓTESIS PREGUNTAS A CONTESTARSE

Preguntas a contestarse, para proyecto factible.

¿Es conveniente emplear una base de Datos relacional en lógica de programación prolog?

Se demuestra cómo el rendimiento del programa Prolog se puede mejorar con el almacenamiento de grandes listas de hechos en una base de datos MySQL y no como hechos tipo Prolog. La mejora se produce porque la Base de Datos relacional MySQL están fuertemente optimizada para la búsqueda en grandes tablas.

En la presente propuesta 'GastroMedical' no usa directamente una base de hechos desarrollada en Prolog ya que esta no me asegura la integridad referencial y va permitir la redundancia de hechos (datos). Estos hechos originalmente se deben tener almacenados en un archivo pl pero para el caso de la BDI este es reemplazado por una base de datos llamada 'tesis' creada en MySQL.

Cuando la consulta medica es realizada por usuario del sistema, dicha búsqueda va ser realizada directamente a la base de Datos relacional teniendo como resultado una base en intenso la misma que es asertada por el sistema en tiempo de ejecución. Es decir la base de hecho prolog (archivo.pl) es reemplazada por la Base de Datos Relacional 'tesis' que es el tema de la investigación propuesta. y esa consulta (que en prolog seria el predicado consult) es reemplazado por la sentencia select de MySQL y la búsqueda del hecho se lo hará por indice el mismo que sería la propia llave primaria. Prolog carece de campos índices en su base de hechos. Ver Manual del Prototipo.

¿La aplicación de una BDI en la medicina tendría mejoras a favor del usuario de este sistema?

El sistema GastroMedical está diseñado para resolver y deducir síntomas de pacientes en su mayoría de escasos recursos que no disponen a la mano de conocimientos médicos y en nuestro medio los síntomas son mal interpretados, auto recetándose y así empeoran la salud del niño incluso pudiendo provocar la muerte del menor. Por ejemplo si el niño presenta serios síntomas de apendicitis y no se diagnostica la patología a tiempo este problema puede empeorar causando la perforación de este órgano obteniendo resultados inesperados.

Evaluación Costo-Beneficio

COSTOS

Cuadro # 3
Costos sin 'Sistema GastroMedical'

Costos	Costos en Hospital Público	Costos en Consultorio Privado	Costos en Clínica
Consulta	s/ 0.00	s/ 20.00	s/ 30.00
Movilidad	s/ 5.00	s/ 10.00	s/ 10.00
Total	s/ 5.00	s/ 30.00	s/ 40.00

Cuadro # 4

Costos con el 'Sistema GastroMedical'

Sistema GastroMedical	s/ 0.00
Movilidad	*

Nota: * La movilidad en particular esta influenciado según la distancia donde se encuentre el paciente. Los valores difieren de acuerdo al sector.

Beneficios esperados:

Cuadro # 5

Beneficios Tangibles
Reducción de Enfermedad y Mortalidad

Cuadro # 6

Beneficios Intangibles
Los niños reciben su diagnóstico en el momento oportuno
Los Padres aprenden de las enfermedades
Cuenta con la ayuda en cualquier momento

Limitaciones

GastroMedical nos permite deducir y diagnosticar una cantidad de enfermedades de los niños las mismas que pueden estar en estado avanzado por lo que se recomendaría la hospitalización e intervención humana.

Epidemiología

“La apendicitis aguda afecta al 6% de la población general, Alcanza su mayor incidencia entre los 6 y 12 años. La edad promedio de aparición es de 10 años, en niñas y niños por igual. Un tercio de las apendicitis se presentan como perforadas. Luego de 48 horas el 80% puede perforarse. En los menores de 3 años, la perforación usualmente ha ocurrido al momento del diagnóstico. La incidencia a esa edad, sin embargo, no va más allá del 2%. En niños preescolares y escolares, la incidencia de la perforación (diagnóstico tardío) varía entre un 30% a un 60%.

En el Ecuador a fines de 1995, la apendicitis aguda fue un problema de salud pública importante, ubicándose en el octavo lugar entre las diez primeras causas de morbilidad, con una tasa de 9.5 por cada 10.000 habitantes ” (17).

¿La BDI GastroMedical se compararía a un Sistema Experto y como tal reemplazaría al médico especialista?

No se debe generalizar o concluir que se está desarrollando un sistema experto como tal, se está logrando la integración de formas diferentes de programar en donde se emplea las ventajas de cada herramienta para obtener una BDI que en nuestro diseño de prototipo se emplearía al campo medico haciendo una simulación para devolver un diagnostico como lo haría el especialista médico, y no está dicho en ningún momento que el prototipo reemplazará completamente a éste Experto humano, sino más bien servirá de ayuda en el proceso de diagnóstico durante las primeras consultas del paciente y evaluación de los síntomas que presente. Dejando así asentadas las base para futuras aplicaciones de este prototipo de BDI.

Ejemplo de cómo se diagnostica en GastroMedical (Ver Manual de Usuario). Los datos personales del paciente son ingresados previamente en el sistema.

Cuadro # 7 Preguntas y Respuestas

GastroMedical	¿ Kathy come alimentos con alto contenido en grasa y bajo contenido en fibra (comidas rápidas, colas) ?
Usuario	Si
GastroMedical	¿ Kathy toma poca agua o líquidos durante el día?

¹⁷ **Epidemiologia**

<http://www.buenastareas.com/ensayos/Apendicitis-En-Ni%C3%B1os/552361.html>

Usuario	Si
GastroMedical	¿ Kathy tiene evacuaciones dolorosas con pesadez abdominal ?
Usuario	Si
GastroMedical	¿ Kathy tiene evacuaciones duras (heces duras) ?
Usuario	Si
GastroMedical	¿ tiene evacuaciones con menos frecuencia o no defeca durante varios días ?
Usuario	Si
GastroMedical	¿ Kathy tiene evacuaciones secas (heces secas) ?
Usuario	Si
GastroMedical	¿ hace poco ejercicio ?
Usuario	Si
GastroMedical	El paciente Kathy presenta síntomas de estreñimiento

¿Es necesario y realizable un sistema de Diagnóstico Médico con el uso de software libre en sistemas y equipamientos informáticos?

Como describimos en la Fundamentación Legal de este capítulo, el desarrollo con tecnología Open Source en países como el Ecuador, se puede y se está desarrollando software de calidad y de bajo costo.

Cuadro # 8

Tabla Comparativa.				
Herramienta	Instalación	Soporte	Pros	Contras
Java	Sencilla.	Muy amplio. Certificaciones a un precio accesible.	Excelente implementación de Java. Integración con MySQL, PostgreSQL. Seguro, Orientado por Objetos , Robusto, Interactivo, Independiente de arquitectura de	La de poseer la tecnología de la máquina virtual, si se hace referencia a la velocidad..

			hardware Interpretado y rápido	
MySQL	Sencilla.	El último recurso de soporte para la comunidad son los foros en http://forums.mysql.com .	MySQL lidera el mercado de las bases de datos de código abierto y, según todo indica, el futuro de MySQL con Sun está en la comunidad Open Source.	Un gran porcentaje de utilidades de Mysql no están documentadas
Prolog	Sencilla.	Disponibles en internet	Ideal para inteligencia artificial y demostracion de teoremas	mal para aquellos que no dominen recursividad. Dificil de aprender

Ofrecer soporte para una empresa de aplicaciones open source es diferente a las demás en muchas formas, muchas de ellas positivas. Por ejemplo, si un negocio usa una aplicación antigua (comercial de código cerrado) y el vendedor de la aplicación deja de dar soporte, el negocio está perdido. Esto no sucede sin embargo con una aplicación open source porque siempre habrá alguien que esté dando soporte y porque las aplicaciones open source, según su filosofía, siempre están en continuo crecimiento y por tanto, nunca dejarán de tener soporte.

VARIABLES DE LA INVESTIGACIÓN

BASES DE DATOS INTELIGENTES

BASES DE CONOCIMIENTOS

INTELIGENCIA ARTIFICIAL

PROLOG

MYSQL

DEFINICIONES CONCEPTUALES

REGLAS: Las reglas de programación lógica se parecen un poco a las vistas relacionales de una base de datos. Especifican relaciones virtuales que no están almacenadas realmente, pero que se pueden formar a partir de los hechos aplicando mecanismos de inferencia basados en las especificaciones de las reglas. La principal diferencia entre las reglas y las vistas es que en las primeras puede haber recursión y por tanto pueden producir vistas que no es posible definir en términos de las vistas relacionales estándar. Reglas expertas mezcladas con datos pueden ser programadas en el mismo sistema sin utilizar ninguna herramienta externa al aplicativo.



Grafico # 8 Sistema de Inferencia

MODUS PONENS: El 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, supóngase que se tiene la regla.

? SI A es cierto, ENTONCES ? B es cierto"

y que se sabe además que "A es cierto".

Se concluye que "B es cierto".

INFERENCIA POR ENCADENAMIENTO DE REGLAS: Esta estrategia de inferencia 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 (Base de Conocimiento), y el proceso continúa hasta que no pueden obtenerse nuevos hechos.

“Encadenamiento hacia atrás.- En este tipo de razonamiento, la idea principal es empezar de los posibles resultados (o hipótesis) de un determinado objetivo o meta, es decir, partiendo de una hipótesis, se buscan evidencias que apoyen (o contradigan) esa hipótesis y que termina con la aceptación o rechazo de la misma. El encadenamiento hacia atrás o razonamiento hacia atrás resulta ser efectivo en aplicaciones en la cual pueden haber muchas respuestas posibles de algo y cuando el número de posibles salidas no es tan numeroso. En este caso,

dada una conclusión el sistema experto trata de determinar qué condiciones se necesitan para obtener esa conclusión.

Observa el siguiente ejemplo .” (18).

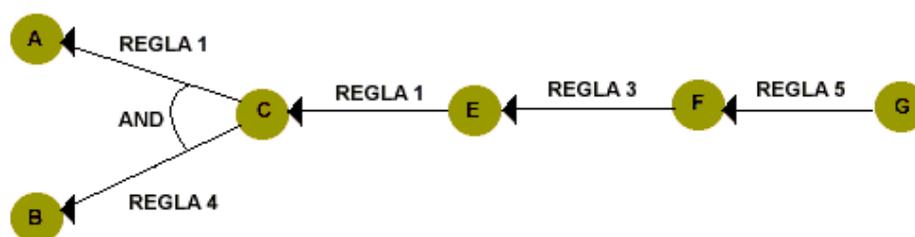
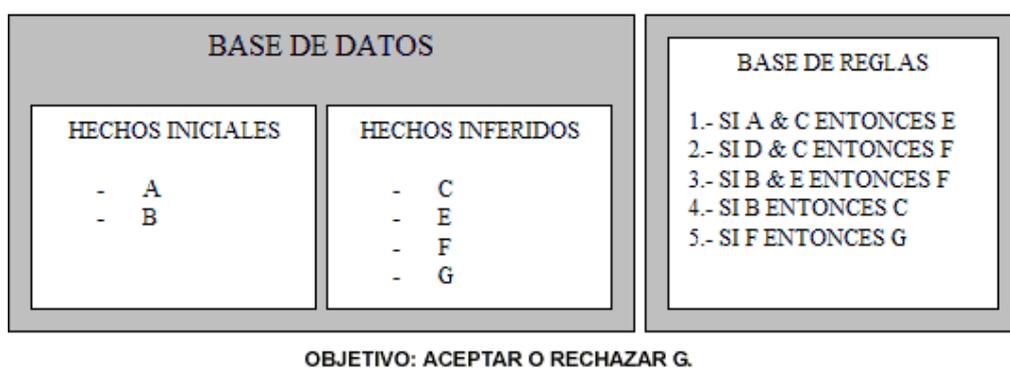


Grafico # 9 Encadenamiento de Reglas

La Figura muestra un ejemplo de encadenamiento hacia atrás. En esta ocasión se inicia a partir de una hipótesis, la hipótesis para este ejemplo es G y el objetivo de encadenar hacia atrás es ver si se acepta o se rechaza G. En el encadenamiento hacia atrás se inicia buscando una regla que incluya al objetivo G en su conclusión ENTONCES. Puesto que la única regla que califica con esta característica es la regla 5, entonces se inicia con ésta. El proceso se explica a continuación:

¹⁸ **Encadenamiento hacia atrás**
<http://geslava.netfirms.com/logifun/capitLF2.htm>

Primero, el motor de inferencia busca si G se encuentra en la base de datos. Como G no está en la base de datos es necesario encadenar hacia atrás y lo único que se tiene son los hechos iniciales A y B .

Después de la regla 5, se procede a buscar alguna regla cuya conclusión sea F. Según la base de reglas, hay dos reglas con esta característica, la regla 2 y 3. Tomando la regla 2 dice que si D y C entonces F pero D no es conclusión de otra regla ni es un hecho inicial y en este caso hay dos opciones: moverse a otra regla o que el sistema le solicite más información al usuario para de esta forma concluir G. Para efectos de este ejemplo se desecha la regla 2 y se continua con otra regla, este proceso se llama retroceso (backtracking).

Considerando la regla 3 es necesario examinar que B y E sean conclusiones. Para el caso de B, se dice que es una conclusión aceptable o que existe debido a que es un hecho inicial. La regla 1 dice si A y C entonces E, A es un conclusión aceptable puesto que al igual que B es un hecho inicial; sólo falta por ver que C sea una conclusión y la regla que lo establece es la 4.

Como B es premisa de la regla 4, ver en seguida que sea conclusión, como B existe entonces C también. Si A y C entonces E, esto hace que F se acepte.

GastroMedical utiliza la máquina abstracta de Prolog y la máquina virtual de Java en el mismo proceso, por lo tanto La máquina virtual de Java y la máquina abstracta de Prolog se ejecutan en procesos totalmente separados por lo que cada uno dispone de

su propia memoria; puede ser útil para utilizar la aplicación en varias máquinas y para reutilizar el mismo código Java para varias aplicaciones Prolog.

BASES DE DATOS RELACIONALES: Las bases de datos relacionales son las que han tenido mas uso comercial. Están orientadas a registros y su concepto fundamental es la relación que es representada por una tabla. Una tabla es un conjunto de tupla de atributos o campos que serán las columnas y los datos almacenados que serán las sucesivas filas. La manera de acceder a cualquiera de ellos se realiza mediante un campo; el cual, será el campo clave y el valor de cada fila en esta columna será diferente.

Las bases de datos relacionales se caracterizan por:

- Simplicidad.
- Generalidad.
- Facilidad de uso para el usuario final.
- Períodos cortos de aprendizaje.
- Las consultas de información se especifican de forma sencilla.
- Posibilidad de manejar una gran cantidad de información.

Cumplen las siguientes leyes básicas:

- Una tabla sólo contiene un número fijo de campos.
- El nombre de los campos de una tabla es distinto.
- Cada registro de la tabla es único.
- El orden de los registros y de los campos no está determinados.
- Para cada campo existe un conjunto posible de valores.

El modelo relacional de la Arquitectura propuesta de GastroMedical [**Ver Anexo 1**]

JAR: (14) Un archivo JAR (Java Archive) es un tipo de archivo que permite ejecutar aplicaciones escritas en lenguaje Java. Las siglas están deliberadamente escogidas para que coincidan con la palabra inglesa "jar" (tarro). Existen tres operaciones básicas con este tipo de archivos: ver contenido, comprimir y descomprimir. Los archivos JAR utilizan una extensión de igual nombre.

La persistencia: Es una característica de ciertos predicados dinámicos que encapsulan estado. El efecto principal de la declaración de un predicado persistente es que los cambios dinámicos realizados se mantienen desde una ejecución a la siguiente.

Aserción es una proposición o enunciado en los que se afirma o da por cierto algo, ejemplo: se afirma, simplemente, que "S es P"; es decir, que un determinado predicado corresponde a un sujeto.

“La IA es una ciencia que trata de la comprensión de la inteligencia y del diseño de máquinas inteligentes, es decir, el estudio y la simulación de las actividades intelectuales del hombre.” (19).

¹⁹ **La IA es...**

http://html.rincondelvago.com/inteligencia-artificial_9.html

CAPÍTULO III

METODOLOGÍA

Investigación Aplicada

DISEÑO DE LA INVESTIGACIÓN

MODALIDAD DE LA INVESTIGACIÓN

En esta modalidad es en parte siguiendo la línea investigativa ya que el prototipo se fundamenta en parte en la investigación bibliográfica. La investigación bibliográfica constituye una excelente introducción a todos los otros tipos de investigación, puesto que ésta proporciona el conocimiento de las investigaciones ya existentes teorías, hipótesis, resultados y métodos usados acerca del tema o problema que yo he tenido que resolver en esta implementación. La propuesta, que se ofrece es dar solución de ingeniería de conocimiento a través de una Base de Datos Inteligente frente a los problemas que enfrentan la medicina pediatra en el área de gastroenterología, implementar una aplicación a base de ingeniería de software y del conocimientos que no genere un costo elevado tanto en infraestructura como en el software que se va a utilizar.

El tipo de investigación que se utiliza es descriptiva y explicativa debido a que se proporcionara manuales que indican la configuración y la implementación del

proyecto antes planteado. También es de comprobación ya que comprobamos que no es necesario contar con licencias de software para desarrollar software de calidad empleando el uso de software libre como se menciona en el capítulo II. El Internet permite la aplicación de los métodos de la investigación.

POBLACIÓN Y MUESTRA

Población:

Se entiende por población al conjunto de elementos con características comunes, que pueden formar parte de un universo.

La población que hemos elegido para nuestro análisis son los profesionales médicos de algunas instituciones, profesionales en medicina cuya experiencia la comparte a través del internet y pacientes a ser consultados.

Muestra:

Es la parte representativa de una población.

Conjunto formado por los medico local de esta ciudad y varios paciente evaluados.

Cuadro # 9 Población

POBLACIÓN	NUMERO
Médicos	4
Pacientes	3
TOTAL	7

Se aplica la fórmula aquí citada :

$$n = \frac{PQN}{\frac{(N-1)E^2 + PQ}{K^2}}$$

n: Tamaño de muestra PQ: Varianza =0.25
 N: Población
 E: Margen de error
 K: Constante de corrección del error =2

EL TAMAÑO DE LA MUESTRA

$$n = \frac{P \cdot Q \cdot N}{(N - 1)E^2 / K^2 + P \cdot Q}$$

P = Probabilidad de éxito (0.50)
 Q = Probabilidad de fracaso (0.50)
 N = Tamaño de la población (7)
 E = error de estimación (6%)
 K = # de desviac. Típicas "Z" (1: 68%, **2: 95,5%**, 3: 99.7%)
 n = Tamaño de la muestra

$$n = \frac{0.50 \times 0.50 \times 7}{(7 - 1)0.06^2 / 2^2 + 0.50 \times 0.50}$$

$$n = \frac{1.75}{(6)(0.0036) / 4 + 0.25}$$

$$n = \frac{1.75}{(6)(0.0009) + 0.25}$$

$$n = \frac{1.75}{0.0054 + 0.25}$$

$$n = \frac{1.75}{0.2554}$$

$$n = 6.85$$

Cálculo de la fracción muestral:

$$f = \frac{n}{N} = \frac{6.85}{7} = 0.978$$

P = Probabilidad de éxito (0.50)
 Q = Probabilidad de fracaso (0.50)
 N= Tamaño de la población (7)
 E= error de estimación (6%)
 K= # de desviac. Típicas "Z" (1: 68%, **2: 95,5%**, 3: 99.7%)
 n = Tamaño de la muestra

Cuadro # 10

MATRIZ DE OPERACIONALIZACIÓN DE VARIABLES

Variables	Indicadores	Técnicas y/o Instrumentos
VARIABLES INDEPENDIENTES		
Inteligencia artificial	Métodos de resolución, sintaxis, agentes	Internet, Foros, Bibliografía
Bases de conocimiento	Almacenamiento, inferencia, axiomas	Internet, Foros, fiabilidad de los datos
MySQL	Datos, consultas, almacenamiento	Internet, Foros
Prolog	Recursión, aprendizaje	Internet, Foros
VARIABLE DEPENDIENTES		
Bases de datos Inteligentes	Respuesta del prototipo en la arquitectura propuesta	Consultas a expertos, Internet Referencias bibliográficas

Elaboración: Alex Portilla

INSTRUMENTOS DE RECOLECCIÓN DE DATOS

Para la recolección de datos para el presente proyecto se ha tomado en consideración revisar documentales haciendo una lectura científica y análisis de contenido para poder llegar a una mejor solución al problema antes planteado. Además como

instrumento de campo se realizaran encuestas a especialistas o personas que conozcan del tema antes propuesto.

En el proyecto se utilizara la encuesta como técnica de campo para recolectar información que nos despejará las dudas que teníamos al inicio del proyecto, ayudándonos a mejorar los objetivos específicos y poder elaborar una propuesta que cumpla con las expectativas de nuestra población.

INSTRUMENTOS DE LA INVESTIGACIÓN

La técnica usada es la encuesta, para la cual utilizaremos un guión de la entrevista, cuestionarios y el Internet.

La encuesta va a estar dirigidos a profesionales en este caso ingenieros en sistemas que trabajen en el área de sistemas como administradores de bases de datos o en el área de desarrollo o tecnología.

ENTREVISTA (Ver anexo I)

Conclusiones:

- Las personas encuestadas no han utilizado una base de datos con conocimientos para una implementación médica y que sea con herramientas tipo Open Source.

- Los profesionales médicos encuestados no tienen un sistema de apoyo a las primeras consultas.
- La mayoría de los encuestados les gustaría usar un software con una interfaz sencilla que sirva de apoyo a las primeras consultas de los pacientes

PROCEDIMIENTOS DE LA INVESTIGACIÓN

El Problema

PLANTEAMIENTO DEL PROBLEMA

Ubicación del Problema en un contexto

Situación Conflicto Nudo Críticos

Causas del problema, Consecuencias

Delimitación del Problema

Formulación del Problema

Evaluación del Problema

Objetivos de la Investigación

Justificación e Importancia de la Investigación

Marco Teórico

Antecedentes del estudio

Fundamentación Teórica

¿Qué son bases de datos inteligentes

Las características de la propuesta

Utilidad para el país, sector, la universidad

Beneficiarios

Sistema experto

Ingeniería del conocimiento

Inteligencia Artificial

Prolog

Base de Datos Relacionales

SGDB

Pediatría

Gastroenterología

Enfermedades gastrointestinales

Componentes del sistema GastroMedical

Base de conocimientos

Base de hechos

El motor de inferencia

Reglas

Interfaz gráfica

Fundamentación legal

Hipótesis preguntas a contestarse

Variables de la investigación

Definiciones conceptuales

Inferencia por encadenamiento de reglas

Bases de datos relacionales

Metodología

Diseño de la investigación

Modalidad de la Investigación

Población y Muestra

Tamaño de la Muestra

Matriz de operacionalización de las variables

Instrumento de Recolección de Datos

Instrumentos de la investigación

Entrevista

Procedimientos de la investigación

CAPÍTULO IV

MARCO ADMINISTRATIVO

CRONOGRAMA

ID	Task Name	Start	Finish	Duration	% Complete	Resource Names
1	Presentacion del diseño de Tesis	04/05/10	04/09/10	5d	0	INICIO
2	ANALISIS	04/12/10	05/21/10	30d?	0	
3	Levantamiento de información / Definición del Problema	04/12/10	04/23/10	10d	0	ANALISIS
4	Recolección de los requisitos del sistema	04/26/10	04/26/10	1d?	0	Objetivos y Alcances
5	Requisitos de Hardware, software	04/26/10	04/26/10	1d?	0	
6	Definición de Objetivos y Alcances	04/27/10	05/03/10	5d	0	Requisitos del Sistema
7	Analisis de procesos	05/04/10	05/21/10	14d	0	
8	Motor de inferencia	05/04/10	05/10/10	5d	0	
9	Base de Conocimiento	05/11/10	05/17/10	5d	0	
10	Base de Datos	05/18/10	05/21/10	4d	0	
11	DISEÑO	05/14/10	06/08/10	18d?	0	
12	Esquema del proyecto	05/24/10	06/08/10	12d	0	DISEÑO
13	Entrega de los objetivos en Dpto. graduacion	05/14/10	05/14/10	1d?	0	
14	Diagramas de secuencia y Flujo	05/17/10	05/20/10	4d	0	Diagrama
15	Arquitectura	05/21/10	05/24/10	2d	0	Instalaciones
16	DESARROLLO DEL PROTOTIPO	05/25/10	09/07/10	76d?	0	DESARROLLO
17	Instalación y configuración de la BD, Prolog, Java	05/25/10	05/26/10	2d	0	Instalaciones
18	Diseño del Modelo Relacional de la Base de Datos	05/27/10	05/27/10	1d?	0	modelo E/R
19	Generación de script de Creación de Tablas	05/28/10	06/16/10	14d	0	PROTOTIPO
20	Implementación de la Base de Conocimiento en Prolog	06/17/10	07/06/10	14d	0	Base conocimiento
21	Desarrollo de la interfaz de usuario	07/07/10	08/31/10	40d	0	PROTOTIPO
22	Documentación	09/01/10	09/07/10	5d	0	Documentacion
23	PRUEBAS GENERALES	09/08/10	10/05/10	20d	0	
24	Prueba de la Base de Datos	09/08/10	09/09/10	2d	0	PRUEBAS
25	Prueba de la Base de Conocimiento	09/10/10	09/16/10	5d	0	PRUEBAS
26	Prueba de usabilidad de la interfaz de usuario	09/17/10	09/21/10	3d	0	PRUEBAS
27	Prueba y corrección de errores	09/22/10	10/05/10	10d	0	PRUEBAS
28	IMPLEMENTACION	10/06/10	10/21/10	12d	0	
29	Terminación del Prototipo y Entrega	10/06/10	10/12/10	5d	0	TERMINACION
30	Revisión de Documentación	10/13/10	10/21/10	7d	0	REVISION

Cuadro # 11
PRESUPUESTO

Detalle de egresos del proyecto

EGRESOS	DÓLARES
Costo Primer Curso Fin de Carrera	\$ 900,00
Reuniones de trabajo e investigación	
Transporte	30,00
Internet	75,00
Refrigerios	35,00
Consultas/citas médicas	40,00
Suministros de oficina y computación	
8 resmas de hojas	23,50
Impresiones	100,00
Encuadernado, , anillado de tesis de grado	50,50
7 Dvd's	5,50
Fotocopias	30,00
Libros y documentos	100,00
Transporte (autobús y taxis)	50,00
Flash Memory	20,50
Equipo de Computación	650,00
AMD Duron, Procesador 1.3 Ghz	
Memoria Ram 480 MB, Disco duro 80 GB	
TOTAL EGRESOS	\$ 2.110,00
TOTAL INGRESOS ALEX PORTILLA P.	\$ 2.110,00

MANUAL DEL PROTOTIPO

El siguiente proyecto está desarrollado en un computador con las siguientes características:

Main Board Clon	Windows XP
CPU	AMD 1.3 GHZ
Memoria	512 MB
HDD	80 GB
Memoria	512 MB
HD	80 GB expandible

ETAPAS B.D.I.

Consta de una base de datos relacional MySQL 5.0, una base de conocimientos, un motor de inferencia diseñado en SWI-Prolog, y medios de interacción hombre-máquina Java.

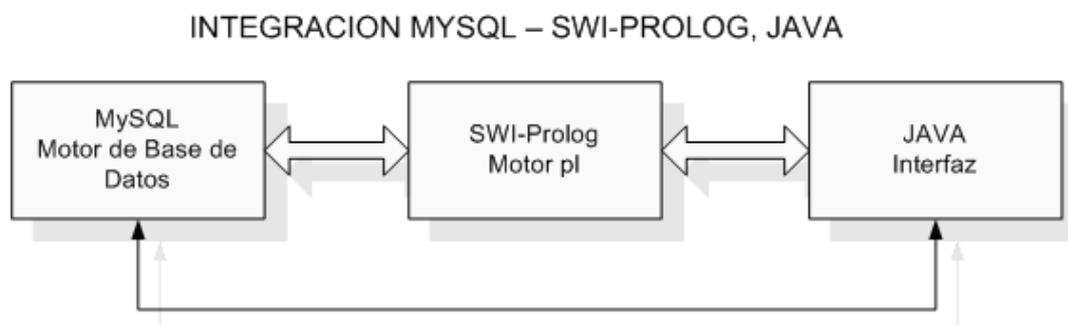


Grafico # 10 Integración

Como se ha explicado en el Capítulo II del marco teórico todo el repositorio de información como los datos personales del paciente, la historia clínica, los conocimientos obtenidos desde el subsistema de adquisición de conocimientos será almacenada en nuestra base de datos relacional MySQL 5.0.

Swi-Prolog como lenguaje de programación lógico y elaboración de prototipos nos dará el motor de inferencia para obtener nuestra base de conocimiento en intenso., formada por las reglas de inferencia, a través de una interfaz creada en Java, el usuario solo vera al sistema como un solo programa.

ARQUITECTURA DEL PROYECTO

A continuación se muestra la Arquitectura del prototipo GastroMedical.

- Consta de las siguientes partes:
- Subsistema de adquisición de conocimiento
- Motor de inferencia
- Repositorio de Base de Datos Relacional
- Repositorio de Base de Conocimiento
- Subsistema de explicación
- Interfaz de usuario
- Aprendizaje.

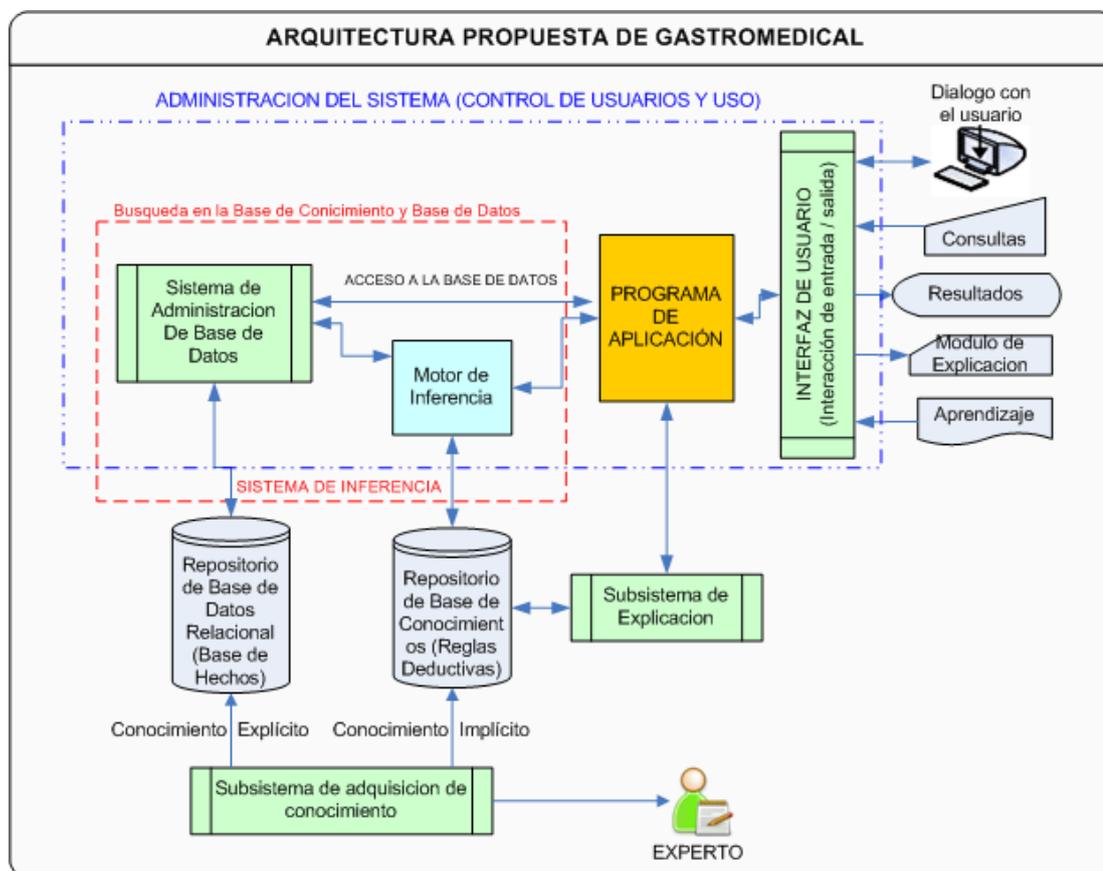


Grafico # 11 Arquitectura GastroMedical

La base de Hechos es un Repositorio de datos obtenidos del subsistema de adquisición del conocimiento. Originalmente la base de hechos es diseñado en prolog se almacena en un archivo de extensión pl el cual es accedido con el predicado consult, pero para nuestro propuesta como se explicó anteriormente no vamos a usar dicho archivos sino que este archivo pl es reemplazado por las tablas que componen la base de datos llamada tesis creada en MySQL 5.0. y como se detalla en el Capitulo II el uso y las ventajas de una Base de Datos Relacional como repositorio de los Hechos de prolog. Estos subsistemas y fases de trabajo de la arquitectura propuesta

se integran desde el Ide NetBeans a través de la paquetería JPL para formar el prototipo de Base de Datos Inteligente.

DIAGRAMAS DE CASOS DE USO

Para comprender los requerimientos del sistema fue necesario realizar un análisis sobre los escenarios en los que se construye el Sistema GastroMedical y también las acciones que conllevan a procesos y/o actividades así como también quien los realiza: Estos escenarios generaron casos de uso para el prototipo.

Actores del Sistema

Los usuarios que consultan al sistema GastroMedical por medio de su interfaz gráfica, otro actor es el especialista médico y son presentados en el siguiente gráfico.

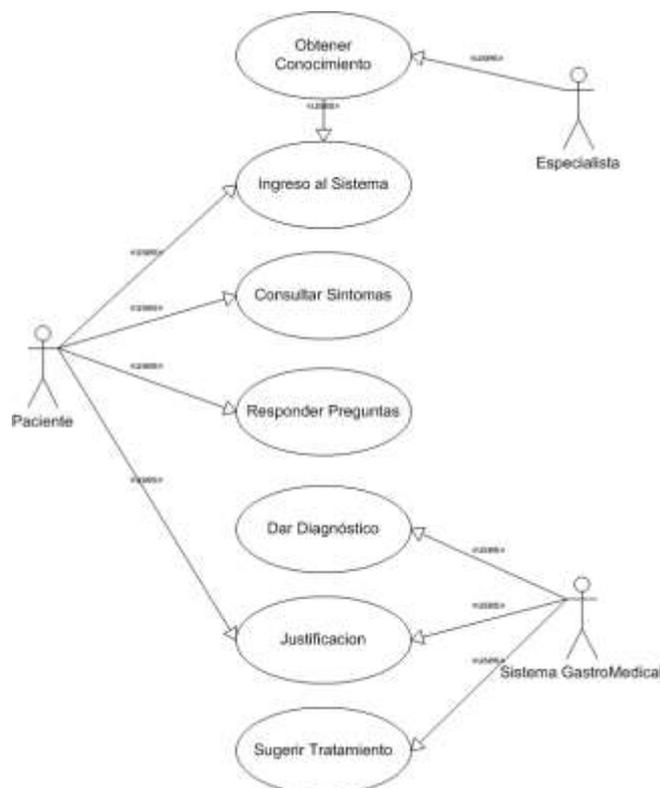


Grafico # 12 Casos de uso

MYSQL INSTALACIÓN

A continuación se resumen los pasos de instalación de MySQL

Instalación del Servidor de Base de Datos

Seguir las opciones por defecto

Asignar contraseña al root, por ejemplo “password”

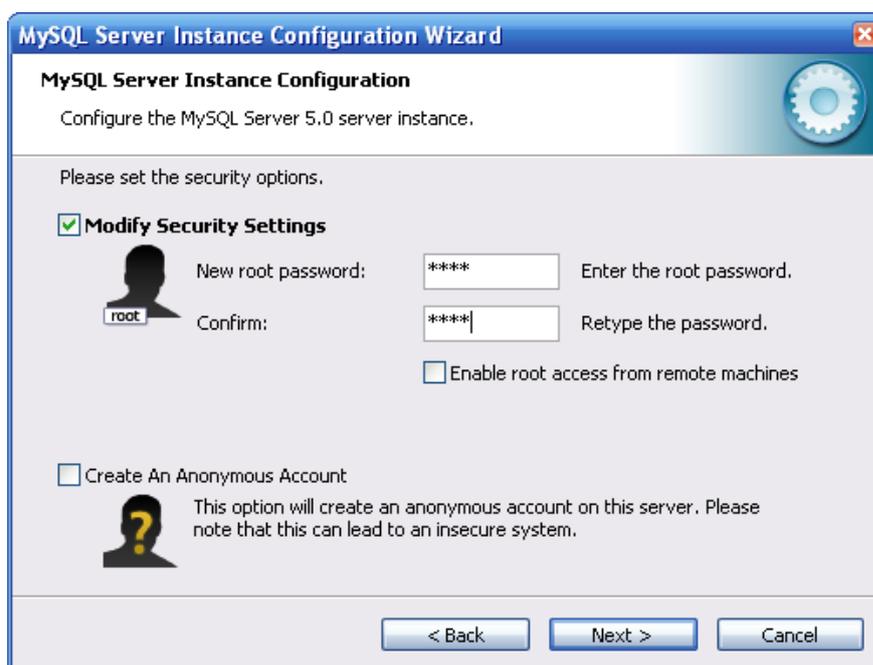


Grafico # 13 Instalación MySQL

Instalación de las herramientas interactivas

Seguir las opciones por defecto

Iniciar MySQL Query Browser

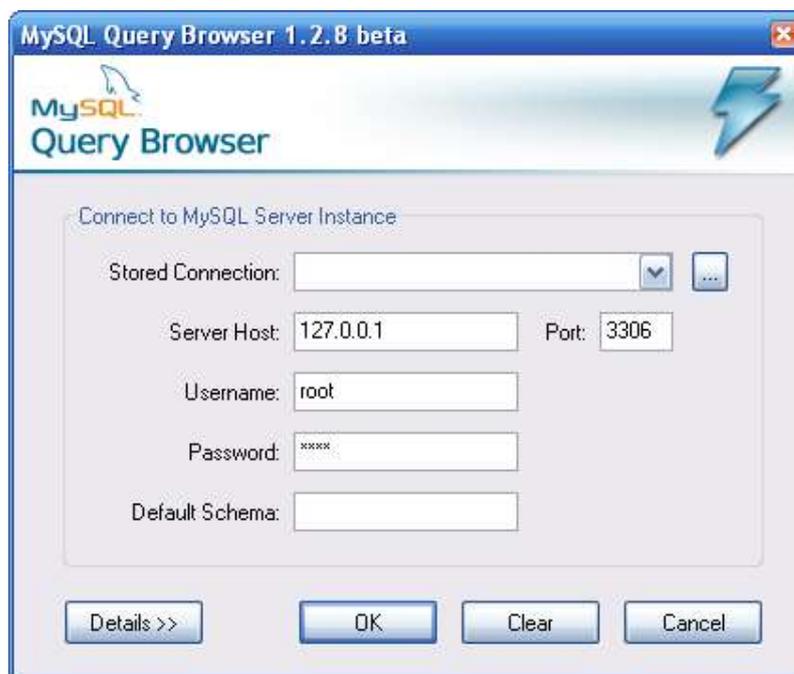


Grafico # 14 Instalación Query Browser

Instalación del driver JDBC

Descomprimos el fichero mysql-connector-java-5.0.6.zip

Añadimos la librería mysql-connector-java-5.0.6-bin.jar

Si compilamos desde línea de comandos, añadimos el fichero a la variable de sistema CLASSPATH.

MODELO ENTIDAD RELACION DE GASTROMEDICAL

A continuación se describe los scripts de creación de las tablas para formar el modelo Entidad / Relación de todo el proyecto.

Primero se crea la base llamada tesis desde la línea de comando de MySQL

```
create database tesis;
```

use tesis; -- me permite indicar a la base que voy usar esta base de nombre 'Tesis'.

Modelo Entidad Relación Roles de usuarios.- Me permite configurar a nivel de Base que usuario va tener todos los permisos del sistema como administrador de GastoMedical. Se describe los scripts de creación:

```
CREATE TABLE ESTADOS(  
    id_estado char(1) not null,  
    descripcion varchar(15) not null,  
    PRIMARY KEY (id_estado)  
);  
  
CREATE TABLE USUARIOS(  
    id_usuario varchar(15) not null,  
    contrasenia varchar(100) not null,  
    Nombres varchar(30) not null,  
    Apellidos varchar(30) not null,  
    estado char(1) not null,  
    primary key (id_usuario),  
    foreign key (estado) references estados(id_estado)  
);  
  
CREATE TABLE ROLES(  
    id_rol integer(1) not null,  
    descripcion varchar(15) not null,  
    primary key (id_rol)  
);  
  
CREATE TABLE PERMISOS(  
    id_permiso integer(2) not null,
```

consultas char(1) not null comment 'S= Si puede realizar consultas. N no tiene permiso e consulta',
 insercion char(1) not null comment 'S= Si puede insertar Nueva Enfermedad. N no tiene permiso de insercion',
 primary key (id_permiso)
);

```
CREATE TABLE ROLES_USUARIOS(
  id_usuario varchar(15) not null,
  id_rol integer(1) not null,
  id_permiso integer(2) not null,
  estado char(1) not null,
  foreign key (id_usuario) references usuarios(id_usuario),
  foreign key (id_rol) references roles(id_rol),
  foreign key (id_permiso) references permisos(id_permiso),
  primary key (id_usuario) );
```

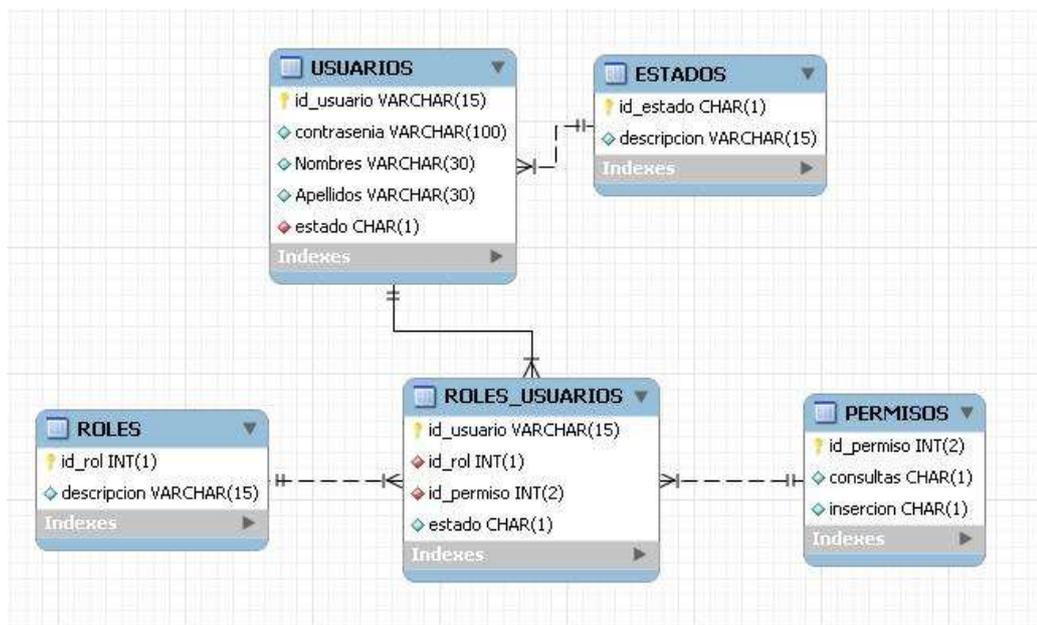


Grafico # 15 Diagrama Entidad Relación Roles/Usuarios

Scripts de creación del repositorio de información para registro de usuarios, hechos y reglas.

```
CREATE TABLE ESPREGUNTA(
  sintoma varchar(35) not null,
  descripcion varchar(260) not null,
  primary key (sintoma,descripcion)
);
```

La Tabla: ESPREGUNTA es el repositorio de todos los hechos que conforman la base de conocimiento que al ser assertada en prolog a través de Java, se convertirá en predicados afirmativos tipo prolog es decir siempre van a ser considerados como verdaderos.

```
CREATE TABLE TIPOENFERMEDAD(
  esenfermedad varchar(55),
  carac varchar(35),
  foreign key (carac) references espregunta(sintoma),
  primary key (esenfermedad,carac)
);
```

La Tabla: TIPOENFERMEDAD es la interpretación de una regla en prolog. Prolog va manejar Reglas persistente propias de su código nativo y Reglas en intenso depositados en la base cuando se llama al módulo de aprendizaje.

```
CREATE TABLE DIAGNOSTICO(
  id_enfermedad varchar(55),
  descripcion varchar(2000),
  foreign key (id_enfermedad) references tipoenfermedad(esenfermedad),
```

```
primary key (id_enfermedad)  
);
```

```
CREATE TABLE TRATAMIENTO(  
    id_enfermedad varchar(55),  
    descripcion varchar(2000) not null,  
    tratamiento varchar(3000) not null,  
    foreign key (id_enfermedad) references diagnostico(id_enfermedad),  
    primary key (id_enfermedad)  
);
```

```
CREATE TABLE DATOS_PACIENTE(  
    nombres varchar(30) not null,  
    apellidos varchar(30) not null,  
    edad integer(2) not null,  
    fechanacimiento date not null,  
    cedula varchar(10),  
    telefono integer not null,  
    ciudad varchar(30) not null,  
    direccion varchar(50) not null,  
    fechaconsulta datetime not null,  
    id_hclinica integer not null,  
    primary key (id_hclinica,nombres,apellidos,fechanacimiento)  
);
```

```
CREATE TABLE HISTORIAL_PACIENTE(  
    id_hclinica integer not null,  
    fechaconsulta datetime not null,  
    enfermedaddiagnosticada varchar(55),  
    foreign key (id_hclinica) references datos_paciente(id_hclinica),
```

```

foreign key (enfermedad) references
diagnostico(id_enfermedad),
primary key (id_hclinica,fechaconsulta)
);

```

```

CREATE TABLE NUEVA_ENFERMEDAD(
enfermedad varchar(55) not null,
dr_responsable varchar(70) not null,
fecha datetime not null,
regla_elpaciente_tiene varchar(1000) not null,
regla_evaluarcarac varchar(1500) not null,
foreign key (enfermedad) references diagnostico(id_enfermedad),
primary key (enfermedad)
);

```

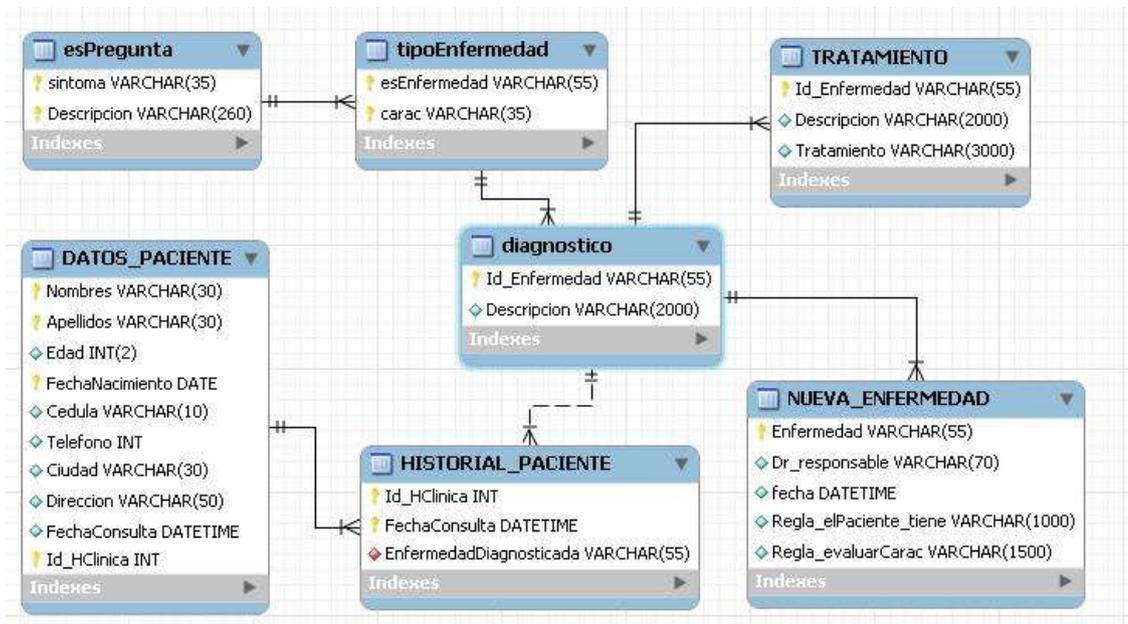


Grafico # 16 Diagrama Entidad Relación del Sistema

A continuación se crean las funciones y procedimientos de la base_

Procedimiento: PR_INSERT_PACIENTE.- Este proceso me permite insertar los datos personales de los pacientes consultados. Es llamado desde el aplicativo en Java.

```
DELIMITER $$
```

```
DROP PROCEDURE IF EXISTS `tesis`.`PR_INSERT_PACIENTE` $$
```

```
CREATE DEFINER=`root`@`localhost` PROCEDURE `PR_INSERT_PACIENTE` (
```

```

    IN Pv_Nombres varchar(30),
    IN Pv_Apellidos varchar(30),
    IN Pn_Edad integer(2),
    IN Pd_Fecha_Nacimiento date,
    IN Pv_Cedula varchar(10),
    IN Pn_Telefono integer,
    IN Pv_Ciudad varchar(30),
    IN Pv_Direccion varchar(50)
    /*OUT pv_error varchar(30)*/
)

```

```
BEGIN
```

```
    INSERT INTO datos_paciente(
```

```
        Nombres,
```

```
        Apellidos,
```

```
        Edad,
```

```
        FechaNacimiento,
```

```
        Cedula,
```

```
        Telefono,
```

```
        Ciudad,
```

```
        Direccion,
```

```
        FechaConsulta,
```

```
        Id_HClinica
    )

```

```

)
VALUES (
    Pv_Nombres,
    Pv_Apellidos,
    Pn_Edad,
    /*DATE_FORMAT(Pv_Fecha_Nacimiento,'%d/%m/%Y'),*/
    Pd_Fecha_Nacimiento,
    Pv_Cedula,
    Pn_Telefono,
    Pv_Ciudad,
    Pv_Direccion,
    now(),
    /*Pn_Id_HClinica*/
    F_SECUENCIA()
);
END $$
DELIMITER ;

```

Procedimiento: PR_UPDATE_PACIENTE.- Este proceso me permite actualizar los datos personales de los pacientes que ya se encuentran registrados en consultas anteriores al sistema. Es llamado desde el aplicativo en Java.

```
DELIMITER $$
```

```

DROP PROCEDURE IF EXISTS `tesis`.`PR_UPDATE_PACIENTE` $$
CREATE PROCEDURE `tesis`.`PR_UPDATE_PACIENTE` (
    in Pv_Nombres varchar(30),
    in Pv_Apellidos varchar(30),
    in Pv_Cedula varchar(10),

```

```

        in Pn_Telefono integer,
        in Pv_Ciudad varchar(30),
        in Pv_Direccion varchar(50)
    )

```

```

BEGIN
    update datos_paciente
    set Cedula=Pv_Cedula,
        Telefono=Pn_Telefono,
        Ciudad=Pv_Ciudad,
        Direccion=Pv_Direccion
    where Nombres=Pv_Nombres
    and Apellidos=Pv_Apellidos;
END $$

```

```

DELIMITER ;

```

Procedimiento: PR_INSERTA_HISTORIAL .- Este proceso me permite registrar la el resumen de historia clínica del paciente una vez que haya terminado la consulta medica. Es llamado desde el aplicativo en Java.

```

DELIMITER $$

```

```

DROP PROCEDURE IF EXISTS `tesis`.`PR_INSERTA_HISTORIAL` $$
CREATE          DEFINER=`root`@`localhost`          PROCEDURE
`PR_INSERTA_HISTORIAL`(
                IN Pn_Id_HClinica integer,
                IN Pv_EnfermedadDiagnosticada varchar(60)
            )

```

```

BEGIN
INSERT INTO HISTORIAL_PACIENTE(
  Id_HClinica,
    FechaConsulta,
    EnfermedadDiagnosticada
  )
VALUES (
  Pn_Id_HClinica,
  now(),
  Pv_EnfermedadDiagnosticada
  );

END $$

DELIMITER ;

```

Función: F_SECUENCIA .- Esta función me permite generar un secuencial en la base llamada 'Tesis' para la historia clínica del paciente. Es llamado desde el aplicativo en Java.

```

DELIMITER $$

DROP FUNCTION IF EXISTS `tesis`.`F_SECUENCIA` $$
CREATE DEFINER=`root`@`localhost` FUNCTION `F_SECUENCIA`()
RETURNS int(11)
BEGIN
DECLARE
  Ln_secuencia integer;
  SET Ln_secuencia=0;
  select IFNULL(max(Id_HClinica),10000)+1

```

```

    into Ln_secuencia
    from datos_paciente;
    return Ln_secuencia;

END $$

```

```
DELIMITER ;
```

Función: F_BUSCA_PACIENTE .- Esta función me permite buscar un paciente que ya haya tenido un historial clínico en el sistema GastroMedical. Es llamado desde el aplicativo en Java a través de un menú de búsqueda.

```
DELIMITER $$
```

```

DROP FUNCTION IF EXISTS `tesis`.`F_BUSCA_PACIENTE` $$
CREATE DEFINER=`root`@`localhost` FUNCTION `F_BUSCA_PACIENTE`(
    Pv_Nombres varchar(30),
    Pv_Apellidos varchar(30)
) RETURNS Int
BEGIN
DECLARE
    Ln_hc Int;
    SET Ln_hc=0;

    SELECT Id_HClinica
    into Ln_hc
    FROM datos_paciente
    WHERE Nombres=Pv_Nombres
    and Apellidos=Pv_Apellidos

```

```
;  
  
return Ln_hc;  
  
END $$  
  
DELIMITER ;
```

INSTALACION DE SWI-PROLOG 5.6

La plataforma base para este trabajo es el sistema operativo Windows XP, no obstante es posible trabajar en otras plataformas como Unix y Linux.

Se utilizará el SWI-Prolog (disponible en www.swi-prolog.org/download.html). La versión a la fecha es SWI-Prolog 5.6.64 por Jan Wielemakerse de donde puede descargarse para plataformas Unix/Linux y Windows (95/98/ME y NT/2000/XP).

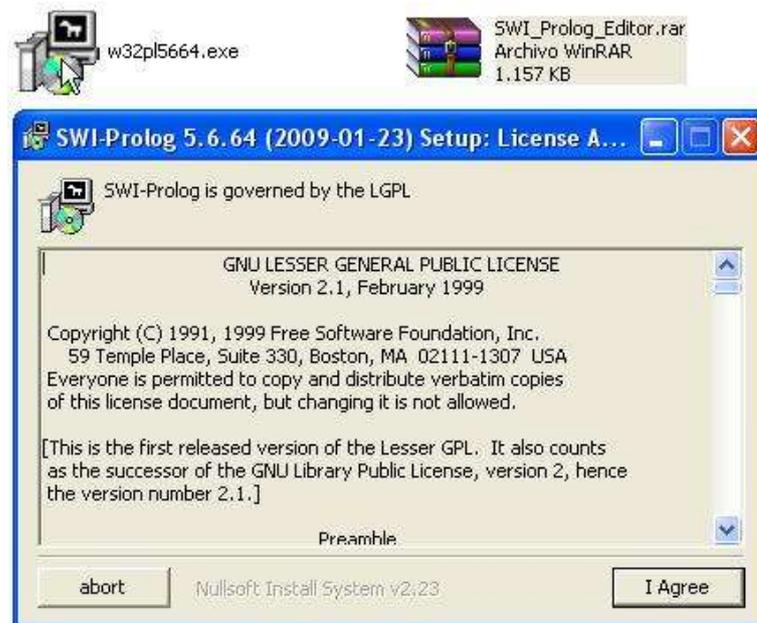


Grafico # 17 Instalación SWI-Prolog

Para Windows el instalador es un archivo ejecutable que al invocarlo despliega varias pantallas solicitando configuración personalizada para la instalación. En la instalación debe marcarse el cuadro correspondiente a JPL Java<->Prolog, para que se incluyan los archivos jar y las librerías dll que permiten realizar la interface.

Configuración con NetBeans 6.5

SWI proporciona un archivo jpl.jar que es un conjunto de clases Java y funciones C que proveen una interfase entre Java y Prolog. JPL usa una interfase nativa de Java (JNI) para conectarse al motor Prolog Foreign Language Interface (FLI). Este archivo se encuentra en nuestro caso en C:\Archivos de programa\pl\lib. En el proyecto Java de Netbeans se debe adicionar el archivo jpl.jar haciendo clic derecho en libraries y seleccionando a continuación Add/Jar. Luego se selecciona el archivo jpl.jar que se encuentra en el mencionado directorio. Las librerías dll que permiten que SWI se comunique con los programas escritos en Java se encuentran en el directorio C:\Archivos de programa\pl\bin. Al hacer click derecho sobre el proyecto Java en Netbeans se tiene una pantalla en la que se debe elegir Run. En el recuadro Workspace del Netbeans se debe colocar la dirección donde se encuentran los dll's; es decir, C:\Archivos de programa\pl\bin.

Archivo de La Base de Conocimiento diseñado desde Swi- Prolog

Prolog es ideal para crear diseño de prototipos de programas para sistemas expertos empleando inferencia con el motor propio de prolog. El propósito de propuesta presentado 'Implementación de una base de Datos Inteligente integrando una base de

conocimiento creada y diseñada en prolog con una base de Datos Relacional MySql a través de una interfaz de un lenguaje de programación lógica Java. Para nuestro estudio se empleo el IDE NetBeans 6.5 por las ventajas que me presenta para crear este tipo de proyecto.

Como se ha dicho, no se pretende crear un Sistema experto como tal, pero si la integración de dos formas diferentes de programar donde se deben aprovechar las ventajas de estas herramientas de desarrollo.

```
/* Archivo pediatra.pl */
```

```
/* -- dynamic me permite crear predicados dinamicos que son afectados por el assert -  
-- */
```

```
:- dynamic tiene/3, enfermedadEs/2,esPregunta/2, elPacienteTiene/2, intoxicacion/1.
```

```
/* ----- HECHOS DINAMICOS ----- */
```

```
/*--- son ejecutados desde el aplicativo para crear la base en intenso de todos los  
hechos que conforman la base de Hechos. Estos hechos son extraidos desde la Base  
MySQL ---*/
```

```
/* ----- RESPUESTAS ----- */
```

```
esRespuesta(si).
```

```
esRespuesta(no).
```

```
/* ----- Formulación de las Preguntas y Validación de La Respuesta ----- */
```

```
/*-- ARBOL DE REGLAS DE INFERENCIA --*/
```

```

preguntar(P,Carac, R):- esPregunta(Carac, P),
                        /*write(Paciente),
                        write(' ¿'), write(P), write('? '),
                        read(R), */
                        esRespuesta(R),!.

/* ----- Guardamos la información de la respuesta a las pregunta -----*/
evaluarCaracteristicaTiene(P, Carac, R):- tiene(P, Carac, R),!.
evaluarCaracteristicaTiene(P, Carac, R):- preguntar(P,Carac,R),!.
esContagioCol(X):- evaluarCaracteristicaTiene(X, alimentoDudosaProcedencia,
R),R=si,!.
esContagioCol(X):- evaluarCaracteristicaTiene(X, alimentoMalCocido, R),R=si,!.
esEvacuacionCol(X):-evaluarCaracteristicaTiene(X, evacuacionAbundante,
R1),R1=si,
                        evaluarCaracteristicaTiene(X, evacuacionGrisaseo, R2),R2=si,
                        evaluarCaracteristicaTiene(X, evacuacionLiquida, R),R=si,
                        !.
esDificultadExpulsarHeces(X):- esDificultadParaEvacuar(X),
                        evaluarCaracteristicaTiene(X, evacuacionSeca, R2), R2=si,!.
esDificultadParaEvacuar(X):-evaluarCaracteristicaTiene(X, evacuacionDura, R),
R=si,
                        evaluarCaracteristicaTiene(X, evacuacionDolorosa, R1), R1=si,!.

```

esProblemaAbdominal(X):- evaluarCaracteristicaTiene(X, cambioAlimentacion, R),
R=si,

 evaluarCaracteristicaTiene(X, consumoAgua, R1), R1=si,

 evaluarCaracteristicaTiene(X, pocoEjercicio, R2), R2=si,!.
.

esTipoDeEvacuacion(X):- evaluarCaracteristicaTiene(X, diarreaNormal, R), R=si,!.
.

esTipoDeEvacuacion(X):- evaluarCaracteristicaTiene(X, evacuacionPoca, R), R=si,!.
.

/* ----- TIPO DE ENFERMEDADES ----- */

esApendicitis(X):-evaluarCaracteristicaTiene(X, abdomenDuro, R0), R0=si,

 evaluarCaracteristicaTiene(X, apetito, R1), R1=si,

 evaluarCaracteristicaTiene(X, cojera, R2), R2=si,

 evaluarCaracteristicaTiene(X, diarreaNormal, R3), R3=si,

 evaluarCaracteristicaTiene(X, dolorBajoVentre, R4), R4=si,

 evaluarCaracteristicaTiene(X, dolorIntensifica, R5), R5=si,

 evaluarCaracteristicaTiene(X, dolorTos, R6), R6=si,

 evaluarCaracteristicaTiene(X, fiebre38_5, R7), R7=si,

 evaluarCaracteristicaTiene(X, hinchazonAbdominal, R8), R8=si,

 evaluarCaracteristicaTiene(X, nauseas, R9), R9=si,

 evaluarCaracteristicaTiene(X, vomito, R10),R10=si,

 evaluarCaracteristicaTiene(X, zonaAdolorida, R11), R11=si,
.

!

esAscari(X):- evaluarCaracteristicaTiene(X, apetito, R0),R0=si,

 evaluarCaracteristicaTiene(X, dolorEstomago, R2),R2=si,

evaluarCaracteristicaTiene(X, esTos, R3),R3=si,
 evaluarCaracteristicaTiene(X, fiebre, R4),R4=si,
 evaluarCaracteristicaTiene(X, hinchazonAbdominal, R5),R5=si,
 evaluarCaracteristicaTiene(X, obstruccionIntestinal, R6),R6=si,
 evaluarCaracteristicaTiene(X, perdidaPeso, R7),R7=si,
 evaluarCaracteristicaTiene(X, presenciaLombrices, R8),R8=si,
 evaluarCaracteristicaTiene(X, vomito, R9),R9=si,
 !.

esColera(X):- esContagioCol(X),

evaluarCaracteristicaTiene(X, calambresMusculares, R2), R2=si,
 evaluarCaracteristicaTiene(X, dolorAbdominal, R3), R3=si,
 esEvacuacionCol(X),
 evaluarCaracteristicaTiene(X, sedExcesiva, R4), R4=si,
 evaluarCaracteristicaTiene(X, vomitoProfundoNauseas, R5), R5=si,
 !.

esColitis(X):-evaluarCaracteristicaTiene(X, aliviaEvacuacion, R0), R0=si,

evaluarCaracteristicaTiene(X, apetito, R1), R1=si,
 esTipoDeEvacuacion(X),/* diarrea y estreñimiento(EvacuacionPoca) */
 evaluarCaracteristicaTiene(X, dolorAbdominal, R2), R2=si,
 evaluarCaracteristicaTiene(X, dolorConstante, R3), R3=si,
 evaluarCaracteristicaTiene(X, estres, R4), R4=si,
 evaluarCaracteristicaTiene(X, evacuacionDolorosa, R5), R5=si,

evaluarCaracteristicaTiene(X, evacuacionIncompleta, R6), R6=si,
 evaluarCaracteristicaTiene(X, fatiga, R7), R7=si,
 evaluarCaracteristicaTiene(X, llenura, R8), R8=si,
 evaluarCaracteristicaTiene(X, perdidaPeso, R9), R9=si,
 evaluarCaracteristicaTiene(X, pocoEjercicio, R10), R10=si,
 evaluarCaracteristicaTiene(X, retencionGases, R11), R11=si,
 evaluarCaracteristicaTiene(X, ruidosAbdomen, R12), R12=si,
 !.

esDiarrea(X):- evaluarCaracteristicaTiene(X, dolorAbdominal, R2), R2=si,
 evaluarCaracteristicaTiene(X, evacuacionLiquida, R3), R3=si,
 evaluarCaracteristicaTiene(X, flatulencias, R6), R6=si,
 evaluarCaracteristicaTiene(X, orina, R8), R8=si,
 evaluarCaracteristicaTiene(X, perdidaPeso, R9), R9=si,
 evaluarCaracteristicaTiene(X, sedExcesiva, R10), R10=si,
 /* evaluarCaracteristicaTiene(X, vomitaFrecuencia, R11), R11=si,*/
 !.

esEstrenimiento(X):- esDificultadExpulsarHeces(X),
 evaluarCaracteristicaTiene(X, evacuacionPoca, R4), R4=si,
 esProblemaAbdominal(X),
 !.

esGastritis(X):-evaluarCaracteristicaTiene(X, acidezEstomacal, R0),R0=si,
 evaluarCaracteristicaTiene(X, apetito, R1),R1=si,

esTipoDeEvacuacion(X),
 evaluarCaracteristicaTiene(X, dispepsia, R2),R2=si,
 evaluarCaracteristicaTiene(X, embalonamientoEstomago, R3),R3=si,
 evaluarCaracteristicaTiene(X, estres, R4),R4=si,
 evaluarCaracteristicaTiene(X, flatulencias, R5),R5=si,
 evaluarCaracteristicaTiene(X, llenura, R6),R6=si,
 evaluarCaracteristicaTiene(X, nauseas, R7),R7=si,
 evaluarCaracteristicaTiene(X, reflujo, R8),R8=si,
 evaluarCaracteristicaTiene(X, vomito, R9),R9=si, !.

esGastroEntero(X):- evaluarCaracteristicaTiene(X, colico, R0), R0=si,
 evaluarCaracteristicaTiene(X, diarreaNormal, R1),R1=si,
 evaluarCaracteristicaTiene(X, dolorAbdominalRetortijon, R2), R2=si,
 evaluarCaracteristicaTiene(X, fiebre, R3),R3=si,
 evaluarCaracteristicaTiene(X, nauseas, R4),R4=si,
 evaluarCaracteristicaTiene(X, vomitoAmarillento, R5),R5=si,
 !.

elPacienteTiene(X,E):-enfermedadEs(X,E),!.

/* ----- EVALUACION DE ENFERMEDADES ----- */

elPacienteTiene(X,apendicitis):-

esApendicitis(X),assert(enfermedadEs(X,apendicitis)),!.

elPacienteTiene(X, ascariasis):- esAscari(X),assert(enfermedadEs(X,ascari)),!.

elPacienteTiene(X, colera):- esColera(X), assert(enfermedadEs(X,colera)),!.

```

elPacienteTiene(X, colitis):- esColitis(X),assert(enfermedadEs(X,colitis)),!.
elPacienteTiene(X, diarrea):- esDiarrea(X),assert(enfermedadEs(X,diarrea)),!.
elPacienteTiene(X,estreñimiento):-
esEstrenimiento(X),assert(enfermedadEs(X,estrenimiento)),!.
elPacienteTiene(X, gastritis):- esGastritis(X),assert(enfermedadEs(X,gastritis)),!.
elPacienteTiene(X,gastroenteritis):-
esGastroEntero(X),assert(enfermedadEs(X,gastro)),!.
%elPacienteTiene(X,intoxicacion):-
esIntoxicacion(X),assert(enfermedadEs(X,intoxicacion)),!.
elPacienteTiene(_X, 'no Determinado'):-!.
%el retractall borrar lo almacenado en la memoria de la variable tiene y
enfermedadEs
empezar(A,Tipo):-%retractall(enfermedadEs(_,_)),
write('SISTEMA GASTROMEDICAL PEDIATRA\n'),
elPacienteTiene(A,Tipo),
nl,
write('GASTROMEDICAL concluye que '),
nl,
write(A), nl,
write(' tiene '), nl,
write(Tipo),
%retractall(tiene(_,_,_)), !.

```

CONNECTOR ODBC

Para conectarnos a una base de datos de MySQL, recuerde que únicamente se puede hacer vía ODBC por lo cual deberá tener instalado el controlador DBCdeMySQL. En este manual partiremos suponiendo que Connector ODBC se encuentra instalado ya que la instalación del ODBC Es sumamente sencilla.

Una vez que tenemos nuestra base de datos ‘tesis’, precedemos a realizar la conexión para Windows a través de nuestro Origen de datos ODBC, de la siguiente manera:

- Inicio ->Panel de control ->Rendimiento y mantenimiento->Herramientas Administrativas-> Orígenes de datos (ODBC)-> DSN



Grafico # 18 ODBC

- Una vez aquí procedemos a agregar el origen de nuestra conexión con su respectivo Driver.
- Seleccionamos nuestro conector que previamente lo habíamos instalado.
- Procedemos a llenar los campos correspondientes.

JAVA 1.6u18 IDE NET BEANS 6.5

La instalación y configuración de estas dos herramientas son sumamente sencilla que no necesita profundizar en este tema pero si la forma correcta de configuración de las variables y los ODBC.



Grafico # 19 Instalación Java

Para esto modificaremos algunas Variables del Sistema de la siguiente manera.

Clic alterno (derecho) a “Mi PC”. Pestaña “Opciones avanzadas”.

Clic “Variables de entorno”.

Modificaremos y crearemos algunas variables del sistema

Clic en “Nueva”.

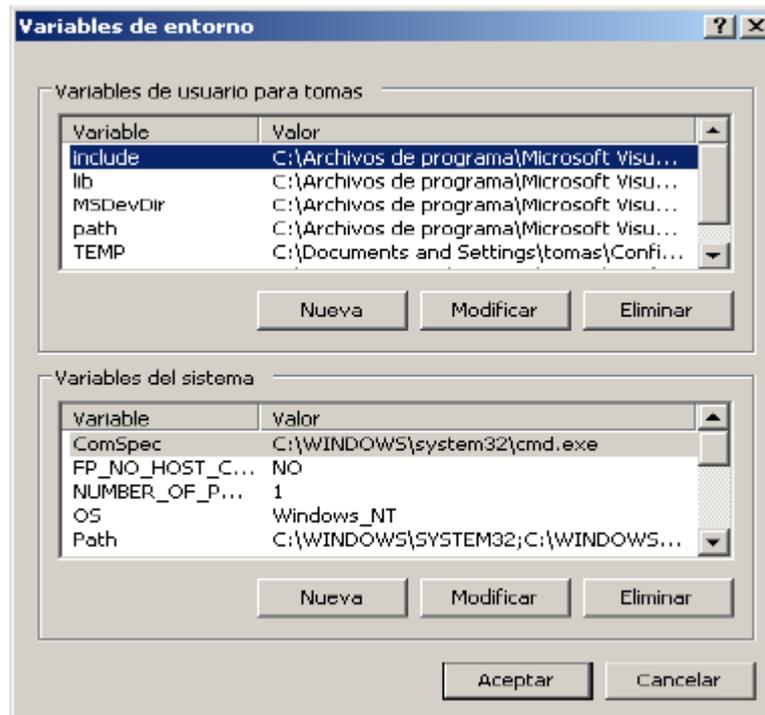


Grafico # 20 Variables de Entorno

a) Nombre de variable: HOME

Valor de variable: %HOMEDRIVER%%HOMEPATH% > Aceptar

b) Nombre de variable: SWI_HOME

Valor de variable: C:\Archivos de programa\pl

c) Nombre de variable: CLASSPATH

Valor de variable: %SWI_HOME%\lib\jpl.jar

d) Nombre de variable: JAVA_HOME

C:\Archivos de programa\Java\jdk1.6.0_18

Ahora seteamos la variable path del sistema con el valor capturado en la variable java_home

e) Nombre de variable:PATH

Valor de variable:

```
%JAVA_HOME%\bin;%JAVA_HOME%\lib\tools.jar;%JAVA_HOME%\jre\lib\rt.jar;
%SWI_HOME%\bin;%SWI_HOME%\lib\jpl.jar;%SWI_HOME%\lib%SWI_HOME%\library;
```

CONEXIÓN DE UNA BASE DE DATOS MYSQL 5.0 CON NETBEANS 6.5

En esta configuración se muestra los pasos para configurar una conexión a una base de datos MySQL 5.0 desde el IDE NetBeans IDE 6.5. se muestran los pasos para la configuración del driver de MySql para su manejo desde el IDE NetBeans de Java. Una vez realizada la conexión, puedes realizar sentencias DML desde java. El IDE NetBeans 6.5 tiene incorporado en su librería su propio ODBC para MySQL.

Configuración de Propiedades del servidor de MySQL

El IDE NetBeans 6.5/6.7 viene con soporte para MySQL RDBMS. Antes de que puedas acceder a la base de datos de MySQL Server en NetBeans IDE, debes configurar las propiedades del servidor MySQL. En la pestaña Prestaciones, expande el nodo Bases de datos, en ella desplegar el menu Databases.

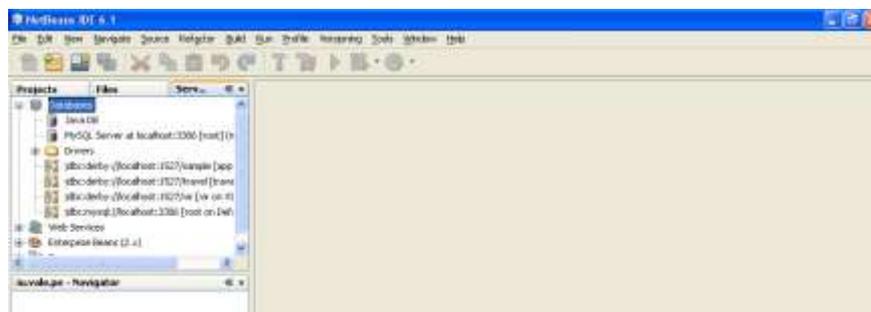


Grafico # 21 Propiedades servidor

Hacemos clic derecho sobre MySQL Server y entramos a las propiedades, en este caso, se muestra el ejemplo sobre un servidor local.

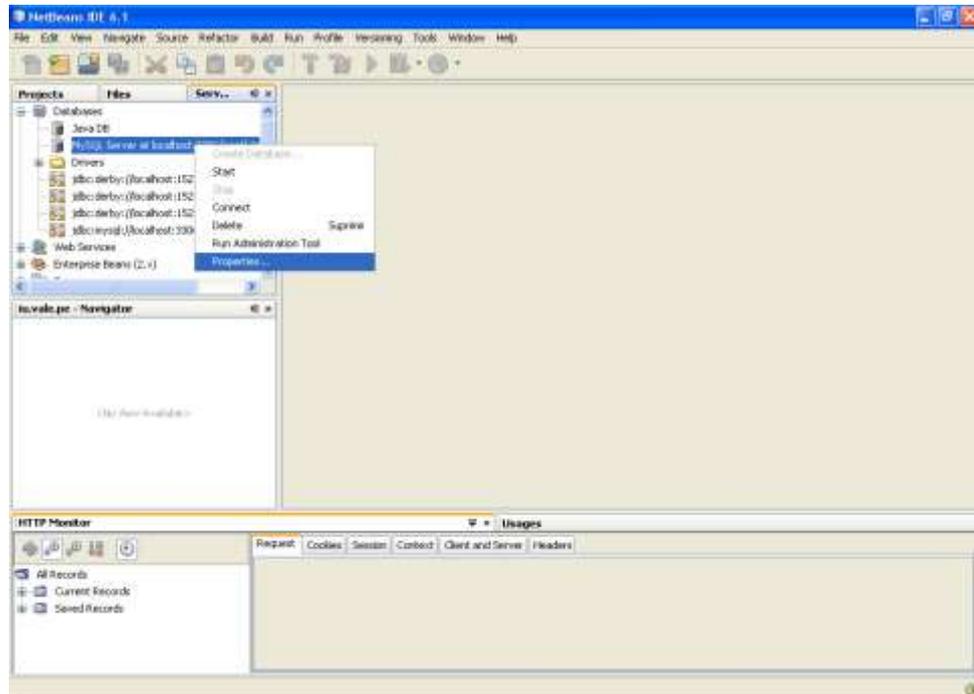


Grafico # 22 Propiedades servidor

En la interfaz de propiedades básicas de MySQL Server editamos el nombre de usuario y el password, dejamos el puerto por defecto: 3306.

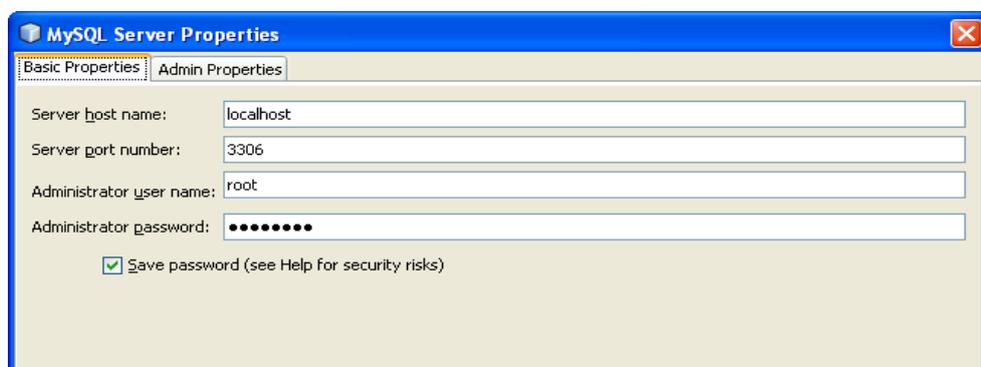


Grafico # 23 usuario puerto

Luego vamos a las propiedades de administración, y brindamos los path (rutas) solicitados. En este caso, se cuenta con MySQL Server desde AppServer. Luego damos click en Ok.

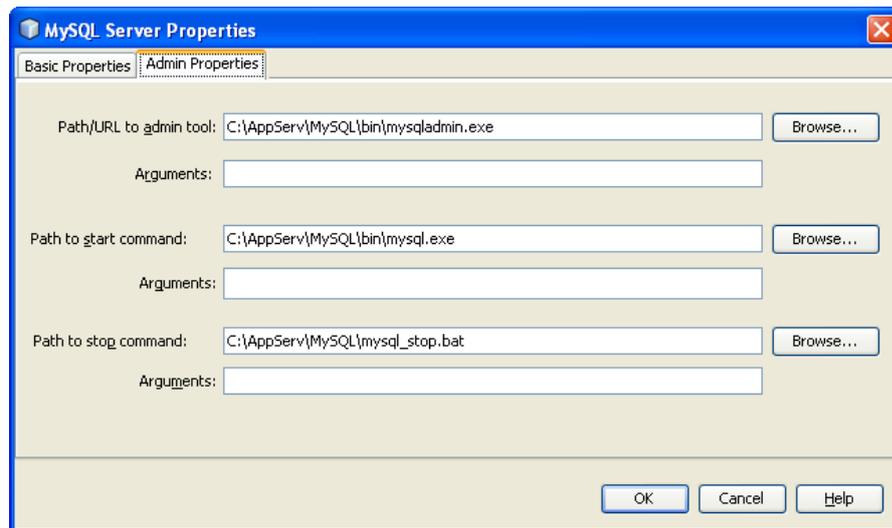


Grafico # 24 Propiedades Administración

Luego estamos listos para inicializar el manejador MySQL, para eso mediante hacemos Start como se muestra en la figura 22.

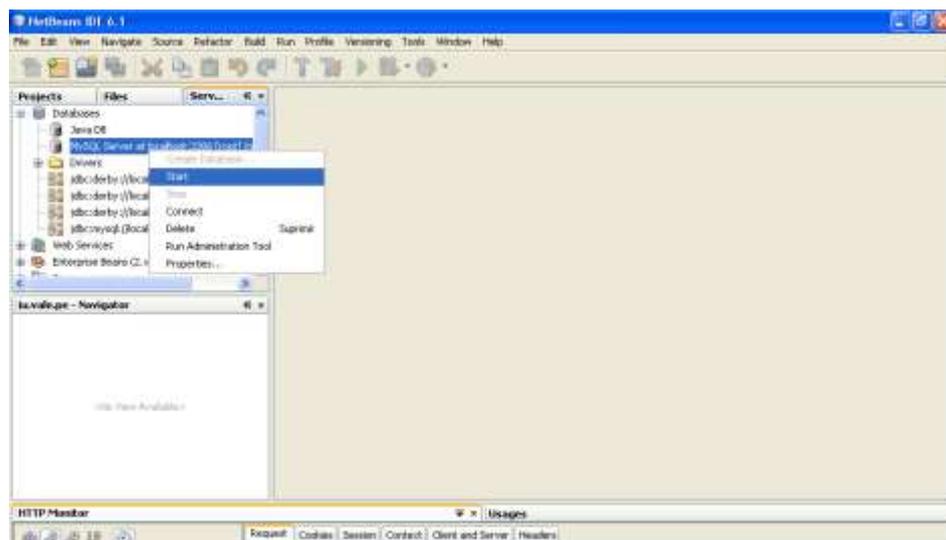


Grafico # 25 start

Ahora podemos verificar que podemos usar MySQL Server desde NetBeans desplegando el contenido de MySQL Server at localhost, mostrándose todas las bases de datos con las que contamos actualmente.

Paquetes y clases de GastroMedical creados desde Java



Grafico # 26 Paquetes

Tenemos el paquete principal que son los fuentes del sistema y las bibliotecas.



Grafico # 27 Paquete principal

En el Sistema GastroMedical tenemos la Clase Conexión que es la que nos permite la conexión a la Base. A continuación describimos la clase conexion.java dentro del paquete Conexión.

```
package Conexion;

import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.SQLException;

public class Conexion {

    private static Connection con;

    //Connection con = null;

    public static Connection getConexion() {

        if (con == null) {

            inicializaConexion();

        }

        return (getCon());

    }

    public static Connection getCon() {

        return con;

    }

    static String bd = "tesis";
```

```
static String user = "root";

static String password = "fedoracore";

static String url = "jdbc:mysql://localhost/"+bd;

private static void inicializaConexion() {

    try {

        Class.forName("com.mysql.jdbc.Connection");

        con = (Connection) DriverManager.getConnection(url, user, password);

        con.setAutoCommit(false); // no haga commit automatico

        if (con != null) {

            System.out.println("Conexion a base de datos "+url+" ... Exitosa Ok Alex "+bd);

        }

    }

    catch(SQLException ex) {

        System.out.println("Hubo un problema al intentar conectarse con la base "+url);

    }

    catch(ClassNotFoundException ex) {

        System.out.println(ex);

    }

    catch (Exception ex) {ex.printStackTrace();}

}

public static void cerrarConexion() {

    try {
```

```

        if (con != null)
            con.close();
    } catch (SQLException e) {
        e.printStackTrace();
    }
}

public static void main(String args[]) {
    Conexion.getConexion();
}
}

/* ----- Fin de la Clase -----*/

```

La siguiente clase: **LoginFrame.java** es la que permite el ingreso al sistema la misma que consulta a la Base 'tesis' a través de la clase conexión.java.

El ingreso desde esta ventana es a través de nombre de usuario y clave serian los dos parámetros a ser consultados y recibe la base. El campo clave (password) esta encriptado dentro de la tabla usuarios de la base.

Esta clase realiza un import a la paquetería swing de java para generar la interfaz y poder presentar la ventanita de ingreso.

La clase esta incluida en el cd de esta Tesis.

```

/* ----- LoginFrame.java -----*/

package proyectoTesis;

import javax.swing.JOptionPane;

```

```
import java.sql.Connection;

import java.sql.ResultSet;

import java.sql.SQLException;

import Conexion.Conexion;

import com.mysql.jdbc.Statement;

import java.sql.PreparedStatement;

public class LoginFrame extends javax.swing.JFrame {

    public LoginFrame() {

        initComponents();

    }

    private void initComponents() {

        /* ... */

    }

    private void jPasswordFieldActionPerformed(java.awt.event.ActionEvent evt) {

        jButtonEntrar.doClick();

    }

    private void jTextFieldUserActionPerformed(java.awt.event.ActionEvent evt) {

        jButtonEntrar.doClick();// esto me permite q al realizar un Enter active el button
Entrar

    }

    private void jButtonSalirActionPerformed(java.awt.event.ActionEvent evt) {

        System.exit(0);

    }

}
```

```

}

private void jButtonEntrarActionPerformed(java.awt.event.ActionEvent evt) {

    String Jusuario = jTextFieldUser.getText();

    String Jclave = String.valueOf(jPasswordField.getPassword());

    boolean consult=false;

    boolean insercion=false;

    Connection con = Conexion.getConnection();

    try {

        int cont=0;

        String query="select count(*) from usuarios where id_usuario=? and
contrasenia=SHA1(?) and estado='A'";

        //String query="select count(*) from usuarios where usuario=
"+Jusuario+" and contrasenia=SHA1('"+Jclave+"') ";

        PreparedStatement s = con.prepareStatement(query);

        s.setString(1,Jusuario);

        s.setString(2,Jclave);

        ResultSet rs=s.executeQuery();

        System.out.println("rs: "+rs);

        if(rs.next()){

            cont=rs.getInt(1);

        }

        if(cont!=0){

```

```
JOptionPane.showMessageDialog(this, "Bienvenidos a GastroMedical",
"Mensaje", JOptionPane.INFORMATION_MESSAGE);

//new Fr_Menu().setVisible(true);

try {

    Statement select = (Statement) con.createStatement();

    String sql = "SELECT p.consultas, p.insercion "+
                "FROM ROLES_USUARIOS r, permisos p "+
                "where r.id_permiso=p.id_permiso "+
                "and r.id_usuario='"+Jusuario+"' "+
                "and r.estado='A'";

    ResultSet rs_permisos;

    rs_permisos = select.executeQuery( sql );

    if(rs_permisos.next() ) {

        if(rs_permisos.getString("p.consultas").equals("S"))

            consult=true;

        if(rs_permisos.getString("p.insercion").equals("S"))

            insercion=true;

    rs_permisos.close();

    }

    } catch ( Exception e ) {

        e.printStackTrace();

    }

}
```

```

        Fr_Menu fr= new Fr_Menu();

        fr.Fr_MenuU(Jusuario,consult,insercion);

        fr.setLocationRelativeTo(null);

        fr.setVisible(true);

        this.dispose();
    }

    else{

        JOptionPane.showMessageDialog(this,"Ingrese un usuario y clave
correctos para Ingresar a GastroMedical");

        jTextFieldUser.setText("");

        jPasswordField.setText("");

        jTextFieldUser.grabFocus();
    }

    s.close();

    rs.close();

    } catch (SQLException ex) {

        System.out.println("Hubo un problema al intentar obtener lo datos de
MySql "+ex);

    }

}

private void jPasswordFieldKeyPressed(java.awt.event.KeyEvent evt) {

    //jTextFieldUser.setText("Cuando sel presiona caulquier tecla sale esto");

```

```

}

public static void main(String args[]) {

    java.awt.EventQueue.invokeLater(new Runnable() {

        public void run() {

            //new LoginFrame().setVisible(true);}

            LoginFrame loginFrame = new LoginFrame

            loginFrame.setLocationRelativeTo(null);

            loginFrame.setVisible(true); //APP Hacemos visible

        }

    });

}

```

A continuación se presenta la case principal del Sistema GastroMedical y es la base de toda la interfaz del prototipo se muestra los métodos principales de esta clase la misma que se encuentra en el código fuente adjunto en el cd del tema propuesto.

```

/*----- Fr_Menu.java -----*/

package proyectoTesis;

import java.util.Date;

import jpl.*;

import java.sql.SQLException;

import Enfermedad_Nueva.*;

import Validar_Cedula.*;

import Busquedas.*;

```

```
import java.sql.CallableStatement;

public class Fr_Menu extends javax.swing.JFrame {

    String nombres;

    int hc=0000;

    String id_paciente="";

    static String usuario="";

    static String nombre="";

    static String Carac="";

    static String salida001;

    static int cont=0;

    static String Enfermedad="";

    String r="";

    boolean lb_empezar=false;

    boolean lb_salir=false;

    cadena_split cad_split= new cadena_split();

    private static Connection con;

    ResultSet rs_Enfermedad;

    ResultSet res_caracteristica;

    public void Fr_MenuU(String usuario_login,boolean consult,boolean insercion)

    {...}

    public Fr_Menu() {...}

    public void Nueva_Enfermedad(){...}
```

```
public void inferencia_prolog() {...}  
  
public void tipoEnfermedad(String Enfermedad)  
  
public void esPregunta() {...}  
  
public void empezar() {...}  
  
public void insert_HistorialPaciente() {...}  
  
public void clear() { ...}  
  
public static void main(String args[]){... } }
```

Existen mas clases contenidas dentro del **‘ProyectoTesis’** e incluidas en el cd del proyecto. En la siguiente conexión se vera la importancia de estas clases.



Grafico # 28 Paquetes

INTEGRACIÓN JAVA PROLOG (JPL.JAR)

Las clases contenidas en jpl.jar suministran los métodos necesarios para comunicarse con Prolog. En nuestro caso tenemos un programa en Prolog ‘pediatra.pl’, para invocarlo desde un entorno visual en Java, primero debemos construir una consulta la cual tiene la siguiente forma:

Cuando el usuario realiza clic sobre el botón Empezar se dispara este método de la clase Fr_menu.java:

```
private void jButtonEmpezarActionPerformed(java.awt.event.ActionEvent evt) {
    jButtonEmpezar.setEnabled(false);
    jButtonSgte.setEnabled(true);
    jButtonPorque.setEnabled(true);
    jRadioButtonNo.setEnabled(true);
    jRadioButtonSi.setEnabled(true);
    Family f =new Family();
    f.abolish_facts();
    f.load_prolog();// carga a pediatra.pl desde la clase java Family
    Nueva_Enfermedad();// busca si existe una enfermedad nueva hace conexion
    f.assert_facts(); // asser en prolog la base de hechos esPregunta
    inferencia_prolog();
    //f.abolish_facts(); // borra la assercion
}
```

Se instancia una variable de la clase Family.java la misma que se detalla a continuación:

```
package proyectoTesis;

import jpl.*;

import com.mysql.jdbc.Connection;

import com.mysql.jdbc.Statement;

import java.sql.ResultSet;

import java.sql.PreparedStatement;

import Conexion.Conexion;

public class Family{

    //variables globales

    static String nombre;

    static String Carac="";

    static String salida001;

    static int cont=0;

    static String Enfermedad="";

    private static Connection con;

    // funcion para cargar prolog consult

    static void load_prolog() {// es invocado desde el buttom Empezar

        JPL.init();

        String t1 = "consult('pediatra.pl)";
```

```

Query q1 = new Query(t1);

        System.out.println( t1 + " " + (q1.hasSolution() ? "succeeded" :
"failed") );

    if(q1.hasSolution()){

        System.out.println("pediatra.pl... Alex la consulta prolog es exitosa");

    }else{

        System.out.println("pediatra.pl... Alex failed La consulta prolog");

    } }

static void assert_facts() {

    try {

        Class.forName("com.mysql.jdbc.Connection");

        con = (Connection) Conexion.getConexion();

        if (con != null) {

            }

            Statement select = (Statement) con.createStatement();

            String sql = "select * from esPregunta";

            ResultSet res = select.executeQuery( sql );

            while( res.next() ) {

                Term child = new Atom( res.getString( "sintoma" ) );

                Term parent = new Atom( res.getString( "descripcion" ) );

                Term t[] = { child, parent};

                Term pair = new Compound( "esPregunta",t); // hecho: esPreguta(carac,descripcion).

```

```

        Query assert_query = new Query( "assert", pair );
        assert_query.oneSolution();

        } // termina el ResultSet fin del while

    select.close();

    } catch ( Exception e ) { e.printStackTrace();}

}

static void abolish_facts() {

    Term pred1 = new Atom( "tiene" ); // limpiamos la base de hechos: tiene

    Term num1 = new Integer( 3 );

    Term t1[] = {pred1, num1};

        Term rem1 = new Compound( "/",t1);

        Query query = new Query( "abolish", rem1 );

        query.oneSolution();

    Term pred2 = new Atom( "enfermedadEs" ); // limpiamos la base de hechos

    Term num2 = new Integer( 2 );

    Term t2[] = {pred2, num2};

        Term rem2 = new Compound( "/",t2);

        Query query2 = new Query( "abolish", rem2 );

        query2.oneSolution();

    }

}

```

LA INTEGRACIÓN DE PROLOG Y LA BASE DE DATOS MYSQL A TRAVES DE NETBEANS 6.5

A continuación se presentan las técnicas de interacción entre Prolog, Java y Bases de datos. Herramientas necesarias:

SWI-Prolog: www.swi-prolog.org

JPL viene en la librería propia del programa prolog:

C:\Archivos de programa\pl\lib

Java JDK lo podemos descargar: <http://java.sun.com/javase/downloads>

<http://www.oracle.com/technetwork/java/javase/downloads/index.html>

MySQL lo podemos descargar: www.mysql.org

<http://dev.mysql.com/>

http://profs.info.uaic.ro/~alaiba/mw/index.php?title=Integrarea_Prolog_%C5%9Fi_R_DBMS_pentru_aplica%C5%A3ii_web

Uso de JPL.- Puede insertar datos desde la base tesis de la tabla esPregunta que se transforma en predicados tipo prolog los mismos que son considerados los hechos de nuestra base de conocimiento en el programa Prolog, estos datos provienen de una base de datos relacional MySQL.

De esta manera podemos combinar las ventajas de Prolog con acceso a los recursos de datos grandes, los programadores podría centrarse en la aplicación de procesamiento de la lógica y no en la creación o administración de base de conocimientos. ¿Debería también tenerse en cuenta que esta no es la única forma de utilizar un RDBMS con Prolog.

Otra alternativa es la interfaz SWI Prolog ODBC. Algunos compiladores de este tipo de programación hacen uso de la funcionalidad del lenguaje lógico para bases de datos y logran la implementación de consultas recursivas sobre bases de datos relacionales, sin embargo, presentan problemas de compatibilidad al momento de reutilizar el código diseñado en aplicaciones desarrolladas con otro tipo de compiladores, ya que las librerías dinámicas generadas por los mismos, incluyen operaciones básicas para almacenamiento y recuperación de información de bases de datos pero no tienen funciones que permitan construir e identificar predicados tipo Prolog en forma directa de la base de datos, encontrándonos con restricciones para diseñar y desarrollar objetos de sistemas de software que utilizan módulos lógicos independientes. Para resolver este problema, se ha desarrollado una conexión desde la interfaz de java que traduce predicados tipo Prolog a una base de datos relacional, y los interpreta a través de consultas diseñada con SQL, y una interfaz gráfica para introducir y recuperar información sin necesidad de que el usuario sea experto en programación lógica.

Agregar la librería jpl a la librería de Net Beans

3.- Ahora agregamos jlp.jar a nuestro proyecto "Libraries". En swi-prolog el jpl.jar esta en la siguiente dirección: "C:\Archivos de programa\pl\lib".



Grafico # 29 Biblioteca Jar

Pasos desde el Ide NEtBeans 6.5

- a) Agregar librería jpl al Library Manager:

Menú Herramientas > Bibliotecas

- b) Clic en el botón Biblioteca Nueva > Nombre de la Biblioteca: jpl

- c) Agregar archivo jar/carpeta > C:\Archivos de programa\pl\lib\jpl y Aceptar

- d) Agregar librería jpl al proyecto

En la paleta Proyectos, clic derecho en el nombre del proyecto y seleccionar Propiedades.

En Categories seleccionar: Bibliotecas

En la solapa Compilar > Añadir Biblioteca

Seleccionar jpl y Añadir Biblioteca.

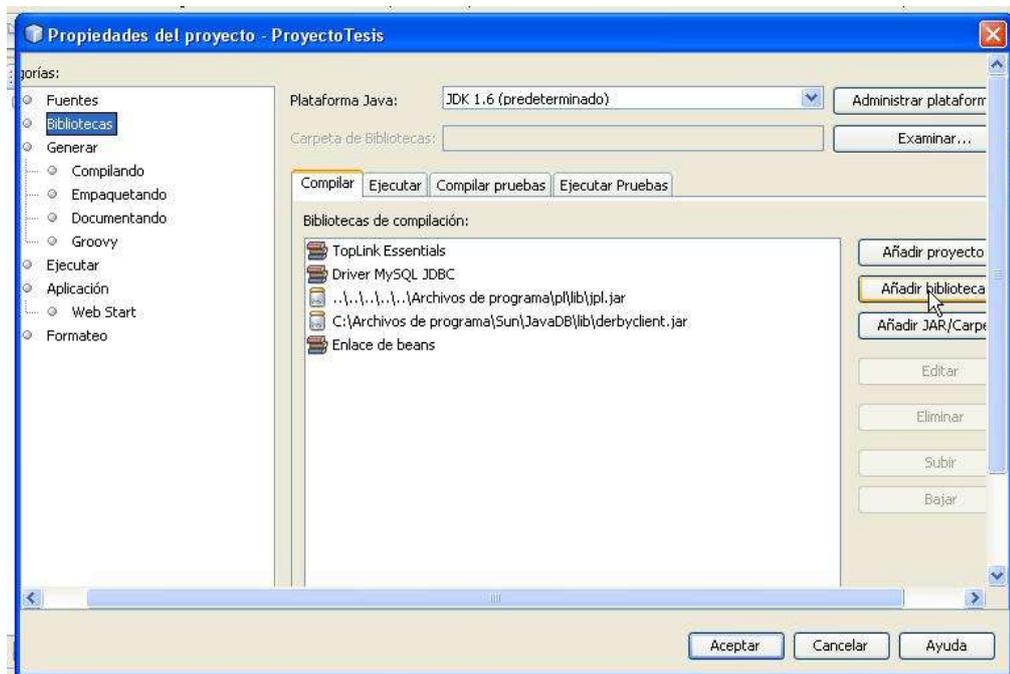


Grafico # 30 añadir biblioteca

Se puede utilizar Prolog en aplicaciones Java utilizando la interfaz de JPL que la podemos encontrar en:

[~ Wwww.jaist.ac.jp/ fuji/work/docs/html4/jpro_p2.html](http://www.jaist.ac.jp/fuji/work/docs/html4/jpro_p2.html)

Llamar a Prolog de los programas de Java utilizando JPL

En el código del archivo paediatra.pl podemos observar que no existen hechos en el programa prolog los cuales intencionalmente van ser obtenidos desde la base de datos tesis de mySQL los hechos desde el principio en el archivo de pediatria.pl, no es obligatorio. La relación es por lo tanto un buen candidato para codificar y almacenar los Hechos (datos que se encuentran en la tabla esPregunta) en la base de Datos Relacional .

Script de inserción:

Tabla: esPregunta

insert into esPregunta values('abdomenDuro','tiene dolor abdominal agudo en el lado inferior derecho');

insert into esPregunta values('apetito','tiene falta de apetito');

insert into esPregunta values('cojera',' cojea al caminar');

insert into esPregunta values('diarreaNormal','tiene diarrea normal, la común');

insert into esPregunta values('dolorBajoVientre', 'siente dolor en la zona cercana al ombligo');

insert into esPregunta values('dolorIntensifica','el dolor se intensifica a medida que pasa el tiempo');

insert into esPregunta values('zonaAdolorida','al tocar la zona adolorida, se contrae de manera espontánea');

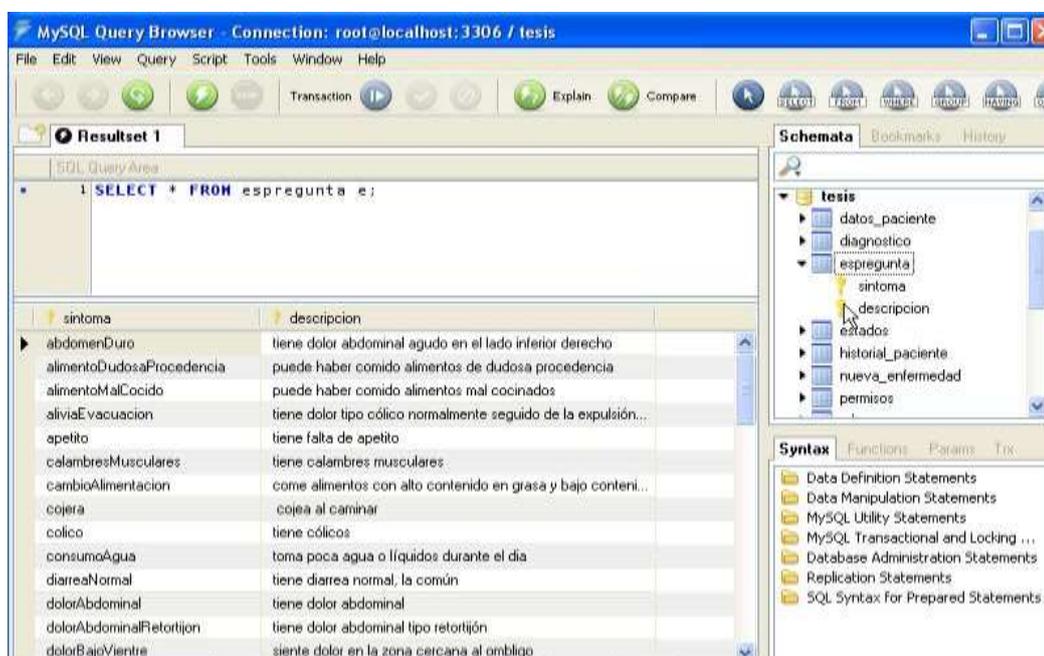


Grafico # 31 select

En el archivo original de prolog seria así con su respectiva base de hechos:

```
/* ----- HECHOS ----- */

% APENDICITIS

esPregunta(abdomenDuro,' tiene dolor abdominal agudo en el lado inferior derecho ').

esPregunta(apetito,' tiene falta de apetito ').

esPregunta(cojera,' cojea al caminar ').

esPregunta(diarreaNormal,' tiene diarrea normal, la comun ').

esPregunta(dolorBajoVientre, ' siente dolor en la zona cercana al ombligo ').

esPregunta(dolorIntensifica,' el dolor se intensifica a medida que pasa el tiempo ').

esPregunta(zonaAdolorida,' al tocar la zona adolorida, se contrae de manera
espontánea ').
```

Hemos quitado las líneas de hechos para bajar de peso al programa pediatra.pl

```
:- dynamic tiene/3, enfermedadEs/2,esPregunta/2, elPacienteTiene/2, intoxicacion/1.
```

```
/* ----- HECHOS DINAMICOS ----- */
```

```
%% no existen la base de hechos en prolog.
```

Especificación de los hechos dinámicos: pediatra.pl:

En el archivo pediatra.pl, se eliminan sólo los hechos y se mantiene las reglas. Para permitir a prólog manejar dinámicamente los hechos, tenemos que declarar el predicado esPregunta/2, (aridad 2) que sean dinámicos. Por lo tanto debe ser insertada la primera línea de la siguiente manera:

pediatra.pl con la declaración de dinámica

```
:- dynamic esPregunta/2.      % el /2 (es la aridad) el hecho tiene dos variables.
```

A continuación, los hechos se pueden especificar de forma dinámica en Prolog para ello hemos creado la clase family.java que se encarga de realizar esta integración de las dos bases y ejecuta los predicados dinámicos gracias a la paquetería jpl de prolog. La integración empieza cuando el usuario termina de ingresar sus datos y empieza la consulta de sus síntomas desde el botón Empezar.

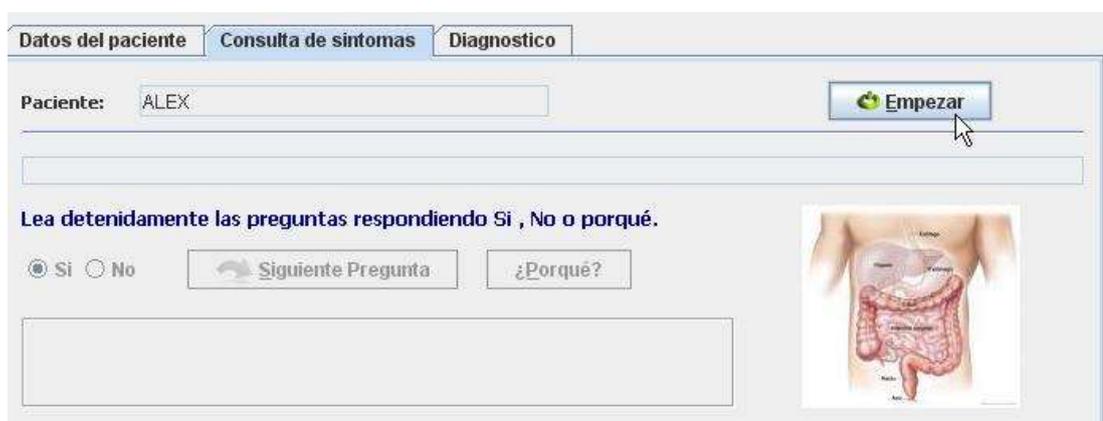


Grafico # 32 Empezar consulta

Este instancia la clase Family.java:

```
Family f =new Family();
f.abolish_facts();
f.load_prolog();// carga a pediatra.pl desde la clase java Family
Nueva_Enfermedad();// busca si existe una enfermedad nueva
f.assert_facts(); // asser (borra) en prolog la base de hechos esPregunta
inferencia_prolog(); ejecuta el motor de inferencia de prolog.
```

Veamos cada uno de estos métodos de la clase family.java. El método abolish_facts() se encarga de borrar los hechos dinámicos que existiesen en prolog después de realizada una consulta. Emplea variables propias del paquete jpl: Term y Compound.

```

static void abolish_facts() {
    Term pred1 = new Atom( "tiene" );// limpiamos la base de hechos: tiene
    Term num1 = new Integer( 3 );
    Term t1[] = {pred1, num1 };
    Term rem1 = new Compound( "/",t1);
    Query query = new Query( "abolish", rem1 );
    query.oneSolution();
    Term pred2 = new Atom( "enfermedadEs" );// limpiamos la base de hechos
    Term num2 = new Integer( 2 );
    Term t2[] = {pred2, num2};
    Term rem2 = new Compound( "/",t2);
    Query query2 = new Query( "abolish", rem2 );
    query2.oneSolution();
}

```

Los hechos se recogerán a partir de la base de datos tesis usando Java y JDBC se colocan luego en el programa Prolog. La idea básica es combinar el programa Java para llamar prólog, por ejemplo, pediatra.pl. y la otra para tener acceso a RDB a través del JDBC.

Para utilizar este programa en una aplicación Java, se debe importar el paquete jpl.jar (la interfaz java para prolog) al proyecto e inicializar especificando la ruta completa al archivo Prolog como se explicó anteriormente.

/* Importamos 'java.util.Hashtable' para recibir el resultado de prolog. El último, "java.sql.*", que nos permita acceder a la RDB (Relational Data Base).

*/

import java.util.Hashtable;

import jpl.Term;

import jpl.Query;

import jpl.JPL;

import jpl.Compound;

import jpl.Util;

import com.mysql.jdbc.Connection;

import com.mysql.jdbc.Statement;

import java.sql.ResultSet;

/* Flujo de control:

En el nivel superior se define el flujo de control.

El método, load_prolog(), de la clase Family.java primero carga el programa prolog, pediastra.pl'.

static void load_prolog() { // es invocado desde el buttom Empezar

 JPL.init();

 String t1 = "consult('pediatra.pl')";

 Query q1 = new Query(t1);

 System.out.println(t1 + " " + (q1.hasSolution() ? "succeeded" : "failed"));

 if(q1.hasSolution()){

```

        System.out.println("pediatra.pl... Alex la consulta prolog es exitosa");
    }else{
        System.out.println("pediatra.pl... Alex failed La consulta prolog");
    }
}

select.close();

con.close();

} catch ( Exception e ) {
    e.printStackTrace();
}
}

```

Otro método, "assert_facts()", establece la conexión con RDB y recupera los datos. El método se afirma o aserta en prolog cada hecho, cada vez que recupera una línea de datos convirtiendo la tabla esPregunta en predicados tipo Hechos de prolog a través de un consulta select a la base.

```

static void assert_facts() {
    try {
        Class.forName("com.mysql.jdbc.Connection");
        con = (Connection) Conexion.getConexion();
        if (con != null) {
            System.out.println("Conexión a base de datos Ok ");
        }
    }
}

```

```

Statement select = (Statement) con.createStatement();

String sql = "select * from esPregunta";

ResultSet res = select.executeQuery( sql );

while( res.next() ) {

    Term child = new Atom( res.getString( "sintoma" ) );

    Term parent = new Atom( res.getString( "descripcion" )

);

    Term t[] = { child, parent};

    Term pair = new Compound( "esPregunta",t);

    Query assert_query = new Query( "assert", pair );

    assert_query.oneSolution();

} // termina el ResultSet fin del while

select.close();

//con.close();

} catch ( Exception e ) {e.printStackTrace();}

}

```

La API está definido por tres interfaces : Prolog , Query y Term . Este método es aplicado por Fred Dushin, el autor del JPL.

MANUAL DE USUARIO DE LA APLICACIÓN

Introducción.- El Manual de usuario tiene como objetivo permitir que la aplicación del Sistema GastroMedical pueda brindarles a los usuarios una interacción amigable y fácil de usar.

Por lo tanto para que este aplicativo sea entendible hemos diseñado ventanas amigables y de fácil entendimiento orientados a usuario no expertos en sistemas.

1.- Ingreso al sistema GastroMedical

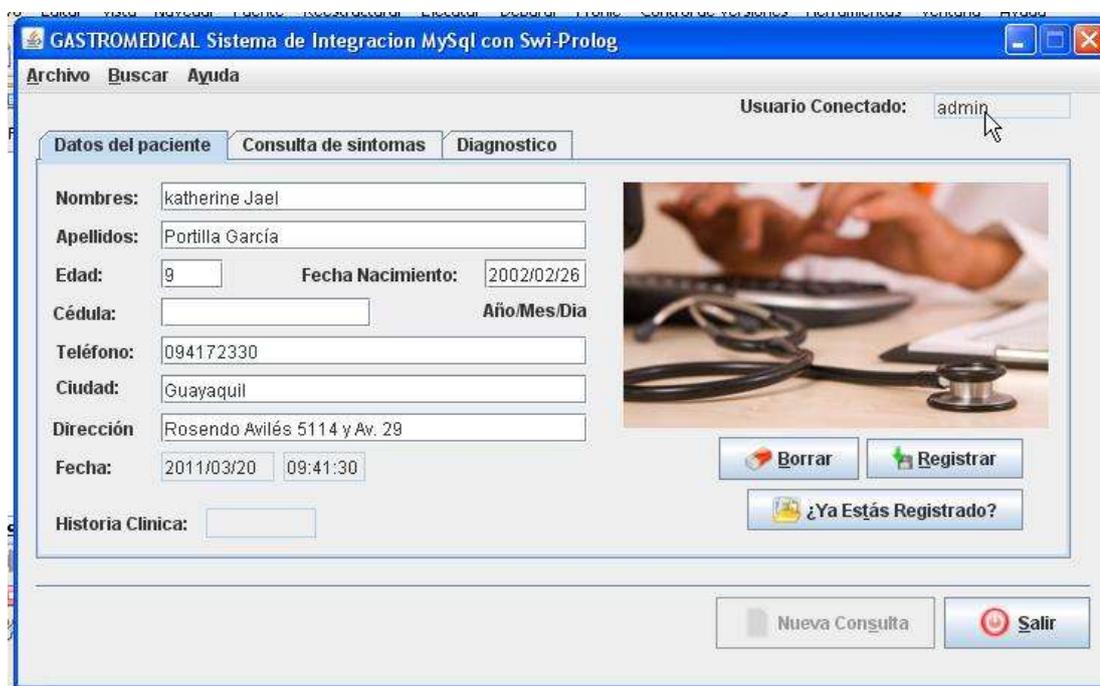


Grafico # 33 Ingreso al sistema

Se procede a ingresar el nombre de usuario y su respectiva contraseña.

Existen usuario tipo administrador y usuario solo para consultas. El usuario administrador es quien tiene todos los privilegios de sistema tales como: consultar, evaluar e ingresar nuevas patologías, el perfil del usuario administrador debe ser el medico especialista y es quine se encargara de alimentar a la base de datos inteligente.

2.- Registro del Paciente



The screenshot shows a web application window titled "GASTROMEDICAL Sistema de Integracion MySql con Swi-Prolog". The window has a menu bar with "Archivo", "Buscar", and "Ayuda". In the top right corner, it says "Usuario Conectado: admin". Below the menu bar are three tabs: "Datos del paciente" (selected), "Consulta de sintomas", and "Diagnostico". The "Datos del paciente" tab contains several input fields: "Nombres" (katherine Jael), "Apellidos" (Portilla García), "Edad" (9), "Fecha Nacimiento" (2002/02/26), "Cédula" (empty), "Teléfono" (094172330), "Ciudad" (Guayaquil), "Dirección" (Rosendo Avilés 5114 y Av. 29), and "Fecha" (2011/03/20 09:41:30). There is also a "Historia Clínica" field. To the right of the form is a small image of a stethoscope. Below the form are buttons for "Borrar", "Registrar", and "¿Ya Estás Registrado?". At the bottom of the window are buttons for "Nueva Consulta" and "Salir".

Grafico # 34 Datos Personales

Se procede a recoger los datos personales del paciente para luego ingresarlos a la base de datos de GastroMedical.

3.- Realizar la consulta

La consulta inicia al realizar clic en el botón empezar. Luego el sistema me va realizando preguntas que deben ser contestado seleccionando las opciones Si o No.



Grafico # 35 Empezar consulta

Cuando un pregunta es contestada se formulará la siguiente al realizar clic en el botón Siguiente Pregunta.

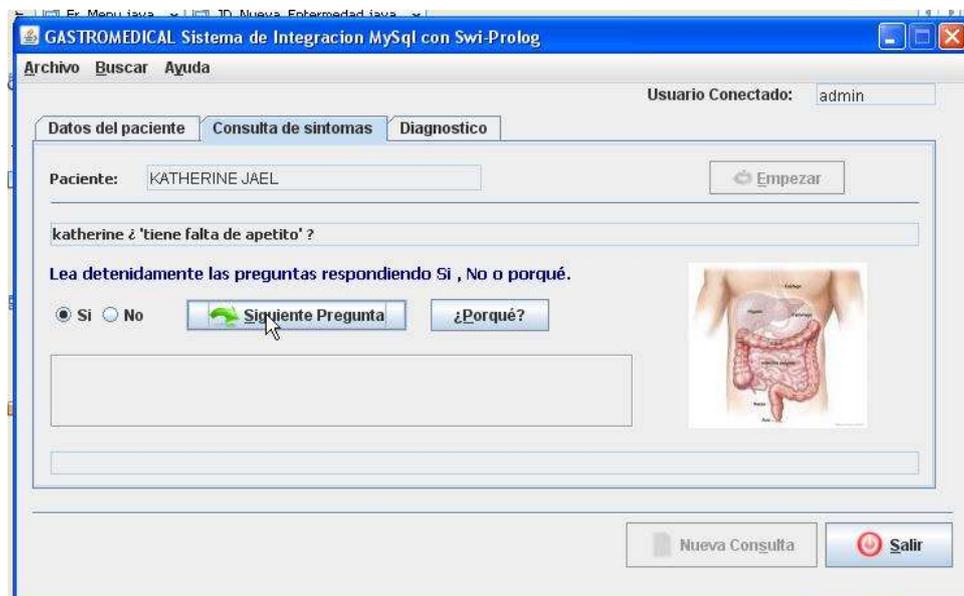


Grafico # 36 Siguiente pregunta

4.- Diagnóstico

Cada pregunta realizada el Sistema buscara si esta de acuerdo a la información proporcionada y este a su vez devolverá una respuestas a los síntomas prestados, es decir un diagnóstico.

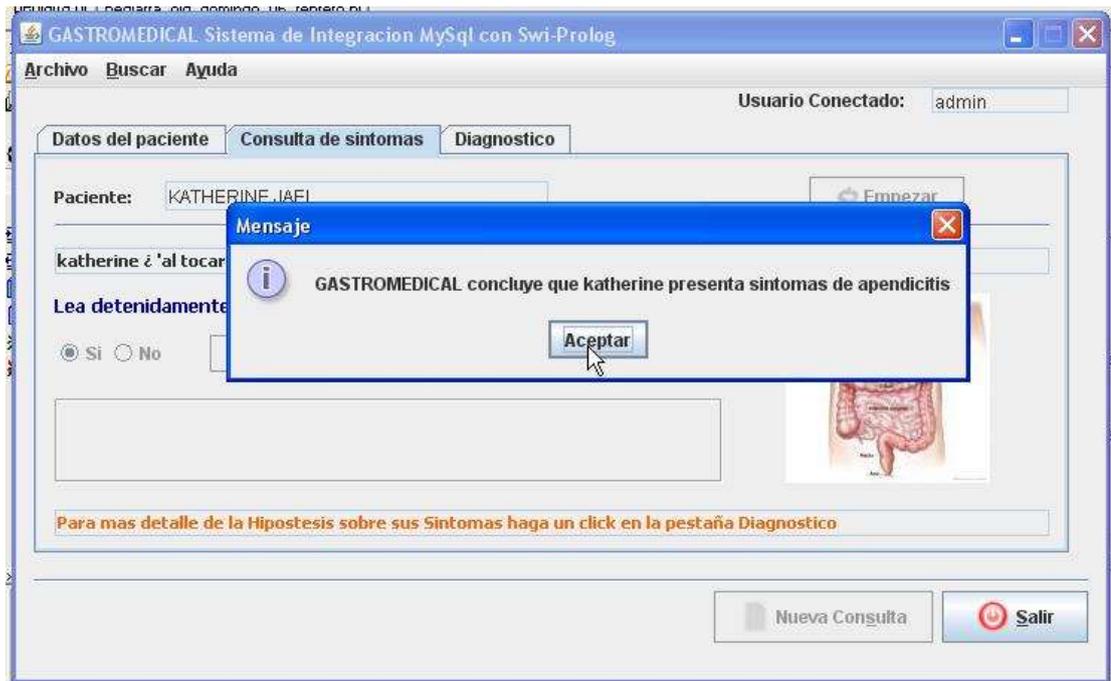


Grafico # 37 GastroMedical concluye

3.- Justificar Diagnóstico

Al obtener el diagnóstico dado os dirigimos a la pestaña Diagnóstico que es donde se justifica el porque de esa conclusión.

Realice un clic en el botón Mostrar del la pestaña Diagnóstico Grafico # 34..

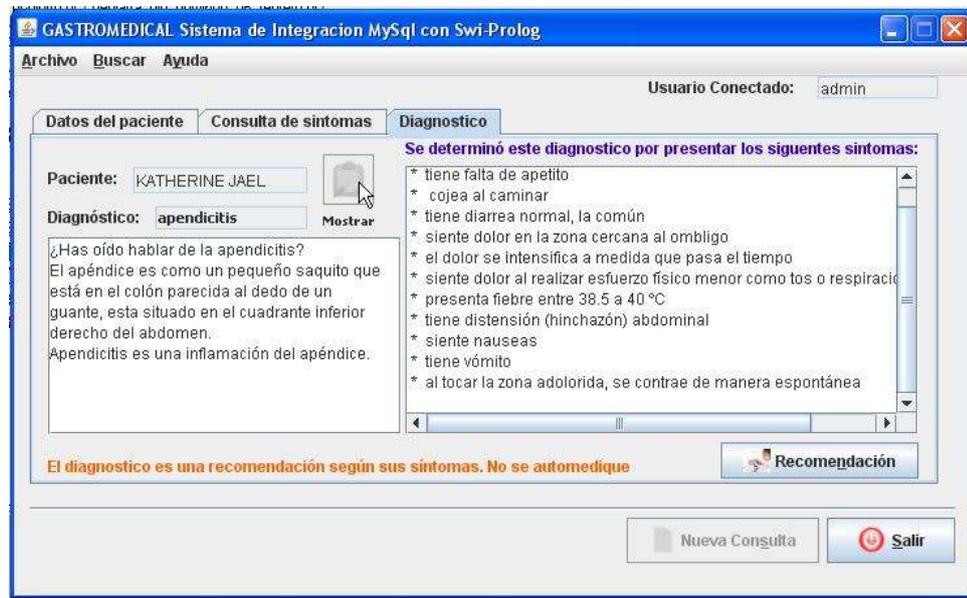


Grafico # 38 Diagnóstico

4.- Recomendaciones o sugerencias

Una vez obtenido la justificación del diagnóstico procedamos a ver las sugerencias de la enfermedad presentado, en caso de ser de gravedad nos indicará que de inmediato acudamos a un centro de salud especializado en esa patología.

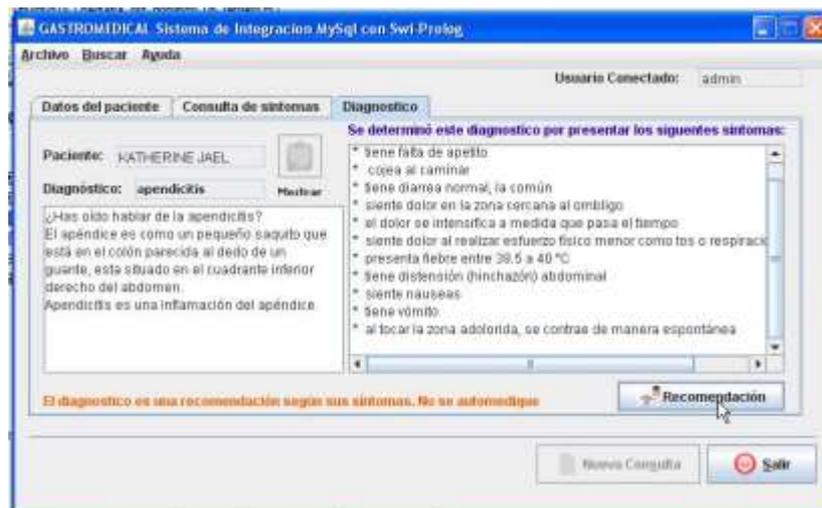


Grafico # 39 Recomendaciones

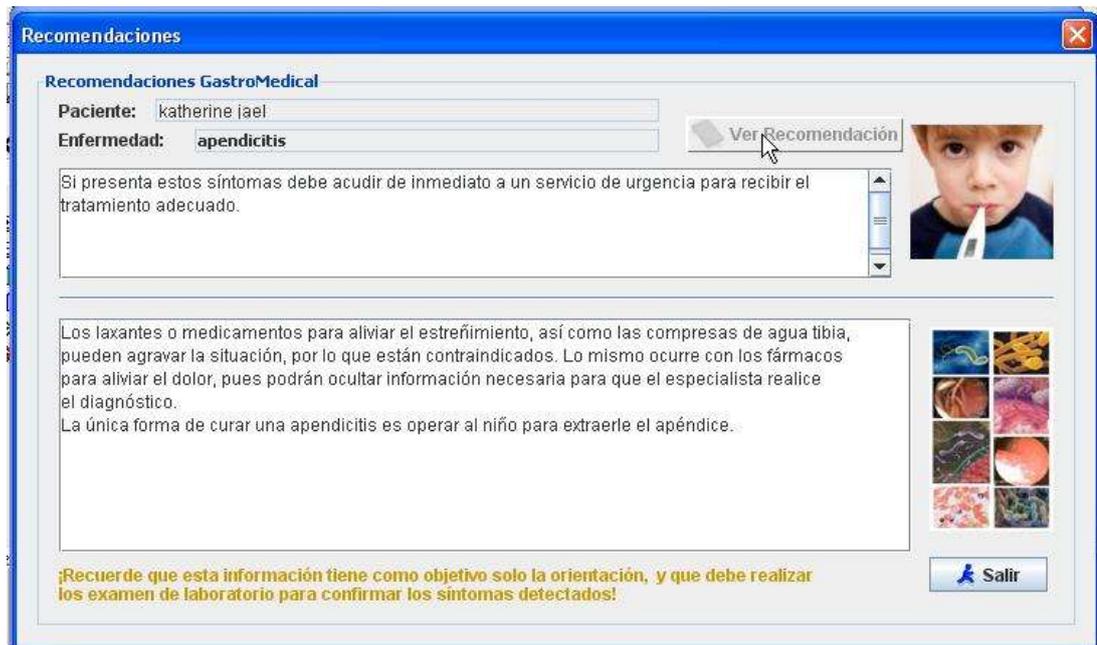


Grafico # 40 Ver recomendaciones

USOS DEL MENÚ DEL SISTEMA GASTROMEDICAL

1.- Menú Archivo

El menú archivo consta de de dos submenú el cual puede estar habilitado o deshabilitado, dependiendo si el usuario es administrador o de consultas.



Grafico # 41 Menú archivo

2.- Menú Buscar

Este menú me permite realizar búsquedas dentro de la base de Datos de GastroMedical sea esta por el número de historia clínica o por nombre, apellidos o una búsqueda general.

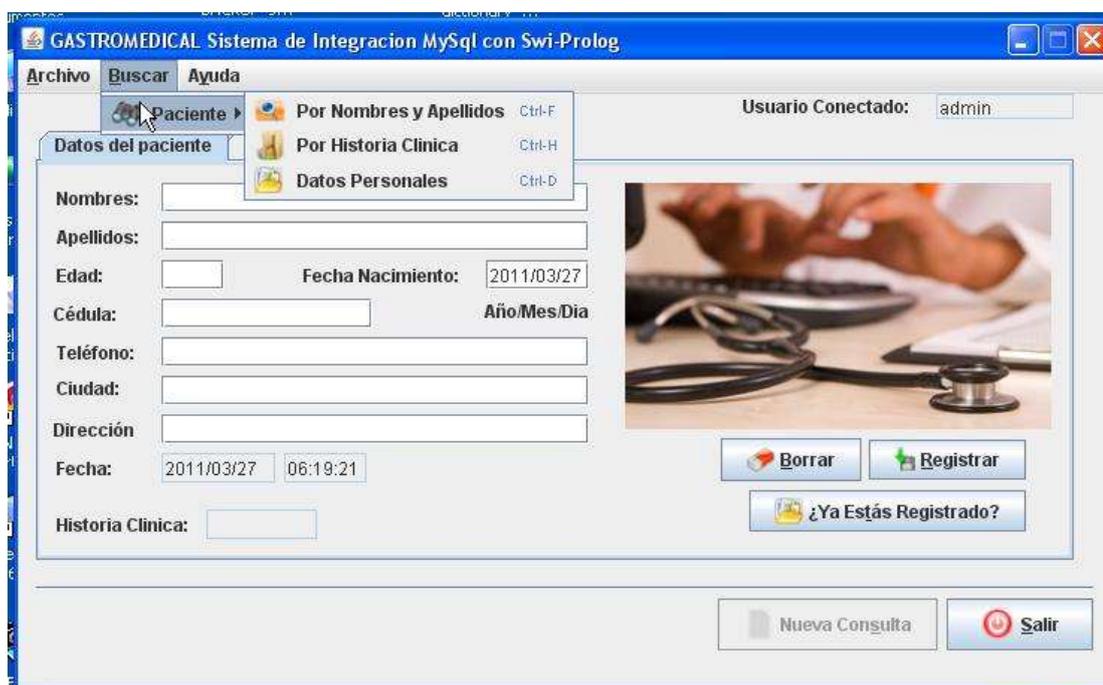


Grafico # 42 Menú Buscar

3.- Menú Ayuda

Este menú e permite tener un idea d quien fue el desarrollador del Sistema y la versión del mismo.

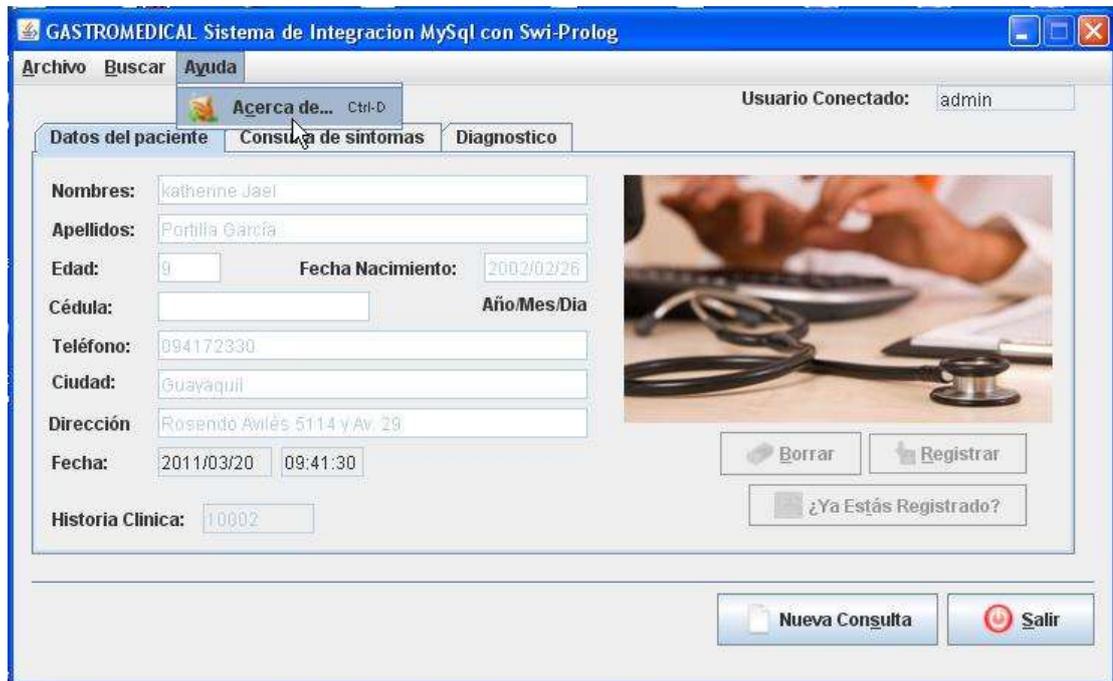


Grafico # 43 Submenú Acerca de

3.- Menú Ayuda o Acerca de...

Este menú e permite tener un idea de quien fue el desarrollador del Sistema y la versión del mismo.

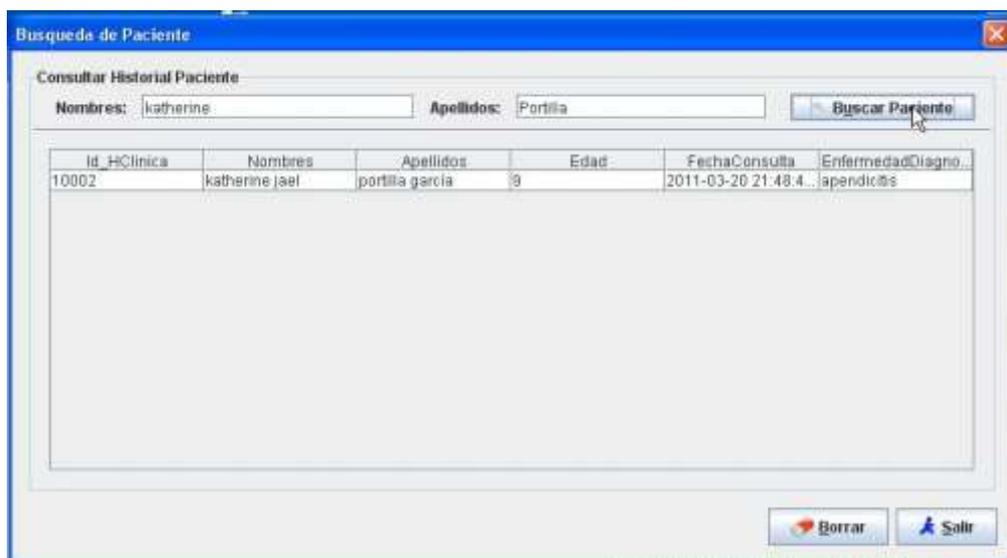


Grafico # 44 Presentación y versión

BUSQUEDAS

1.- Búsquedas por nombres y apellidos

El menú búsqueda por nombre y apellido nos consulta a la base de datos devolviéndonos los datos personales de las consulta que el paciente haya realizado en determinadas fechas.



Id_HClínica	Nombres	Apellidos	Edad	FechaConsulta	EnfermedadDiagno
10002	katherine jael	portilla garcia	9	2011-03-20 21:48:4	apendicitis

Grafico # 45 Presentar Búsqueda

2.- Búsquedas Datos Personales del Paciente

Este Menú de búsqueda nos permite mostrar los datos personales del registro de un paciente y así continuar en la atención por parte del sistema para recibir su consulta.

Datos Personales

Datos Personales del Paciente

Nombres:

Apellidos:

Historia Clínica:

Edad: **Años** **Fecha Nacimiento:** **Año/Mes/Dia**

Cédula:

Teléfono: **Fecha Registro:**

Ciudad:

Dirección:

Grafico # 46 Mostrar Datos Personales

3.- Botón Salir

El botón Salir o el botón de cerrar (la X) tiene un alerta que me pide la confirmación para poder cerrar la aplicación.

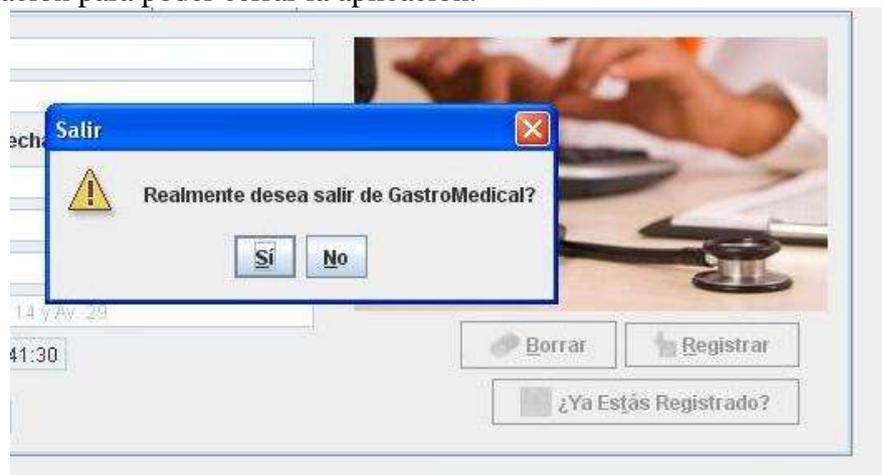


Grafico # 47 Salir

INGRESO DE NUEVAS ENFERMEDADES

Para ingresar una nueva patología es necesario ser un usuario administrador.

The screenshot shows the 'GASTROMEDICAL Sistema de Integración MySql con Swi-Prolog' application. The user is logged in as 'admin'. The 'Ingresar Nueva Patología' menu item is highlighted. The form contains the following fields:

- Nombres:
- Apellidos:
- Edad: Fecha Nacimiento: 2001/03/20
- Cédula: Año/Mes/Día
- Teléfono:
- Ciudad:
- Dirección:
- Fecha: 2011/03/20 10:20:21

Buttons: 'Borrar', 'Registrar', and '¿Ya Estás Registrado?'. A stethoscope image is visible on the right side of the form.

Grafico # 48 Nueva Enfermedad

Se llenan los datos correctamente y se procede a guardar la nueva enfermedad.

The screenshot shows the 'GastroMedical Nueva Enfermedad' application. The title is 'INGRESO DE LA NUEVA ENFERMEDAD'. The form contains the following fields and sections:

- Dr. Responsable: Suarez Pita
- Fecha: 20/03/2011 - 20/03/2011
- Nombre de la Enfermedad: H1N1
- Ingresar Síntomas**

Síntoma	Nombre	Descripción	Acción
sintoma 1	fiebre	presenta fiebre de 39 grados	Guardar
sintoma 2	tos	tiene tos con frecuencia	Guardar
sintoma 3	insomnio	no puede dormir en la noche	Guardar
sintoma 4			Guardar
sintoma 5			Guardar
sintoma 6			Guardar
sintoma 7			Guardar
sintoma 8			Guardar
sintoma 9			Guardar
- Ingreso sobre la enfermedad**
 - Descripción de la Enfermedad
 - Prevención
 - Tratamiento

el tratamiento es a traves de via intravenosa

Buttons: 'Nuevo', 'Guardar', and 'Salir'.

Grafico # 49 Guardar enfermedad

Salida del Sistema GastroMedical

Las salida del sistema puede ser de diferentes maneras, por ejemplo con el botón cerrar, el botón salir, el menú salir o el comando Ctrl+Q. En cualquiera de estas acciones el sistema nos va pedir la confirmación del mismo.

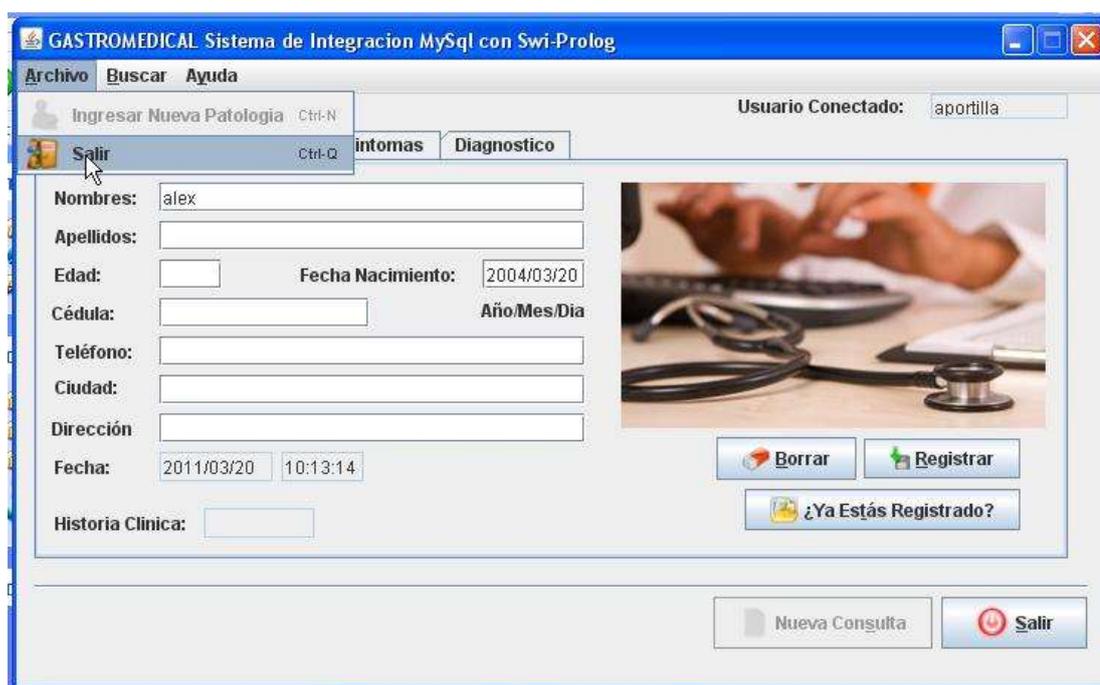


Grafico # 50 Cerrar el sistema

ANEXO 1

ENTREVISTA AL ESPECIALISTA

Guayaquil, Octubre de 2010

UNIVERSIDAD DE GUAYAQUIL
 FACULTAD DE CIENCIAS MATEMATICAS Y FISICAS
 CARRERA DE INGENIERIA EN SISTEMAS COMPUTACIONALES

IMPLEMENTACION DE UN SISTEMA DE BASE DE DATOS INTELIGENTE INTEGRANDO BASE DE CONOCIMIENTOS EN PROLOG Y BASE DE DATOS MYSQL

El proyecto pretende realizar la implementación de un prototipo de Base de Datos Inteligentes llamado "GastroMedical" aplicada en el área médica en pediatría gastrointestinal para proveer sistema de ayuda a las consultas de los paciente devolviendo un diagnostico médico como lo haría un experto.

ENTREVISTA

Dr. (a) B. Kenia Gavilanes Pesantes.....

¿Cuáles son las patologías más frecuentes en su consulta?

<u>Bronquitis</u>	<u>Alergias</u>
<u>Neumonía</u>	<u>dermatitis atópica</u>
<u>Enf. diarreica</u>	<u>dermatitis del pañal</u>
<u>vómitos</u>	<u>Parasitosis</u>
<u>amigdalitis</u>	<u>Dolor Abdominal</u>
<u>Rinitis alérgica</u>	<u>Inf. vías urinarias</u>

¿De las siguientes enfermedades gastrointestinal cuáles son más comunes?

Apendicitis	<input type="checkbox"/>
Ascariasis	<input checked="" type="checkbox"/>
Cólera	<input type="checkbox"/>

Diarrea	<input checked="" type="checkbox"/>
Gastritis	<input type="checkbox"/>
Gastroenteritis	<input checked="" type="checkbox"/>
Intoxicación Alimentaria	<input type="checkbox"/>
Síndrome de Intestino Irritable (Colitis Nerviosa)	<input type="checkbox"/>
Estreñimiento	<input checked="" type="checkbox"/>
Otras: (menciónelas)	<input checked="" type="checkbox"/>
<u>Inyección x H. pylori</u>	<input type="checkbox"/>
<u>Micosis Intestinal</u>	<input type="checkbox"/>
_____	<input type="checkbox"/>
_____	<input type="checkbox"/>

¿Cómo está la gastroenterología pediátrica ecuatoriana en comparación con el resto de los países?

Tenemos el mismo de diagnósticos y tratamientos similares pues la actualización es constante.

¿Dispone usted de un sistema de Base de Conocimientos y de Datos que le sirva como soporte al momento de evaluar consultas y brindar un diagnóstico médico?

SI NO

¿Le gustaría ayudar con su experiencia a elaborar la base de conocimiento del proyecto de Base de Datos Médica Inteligente "GastroMedical" para que muestre soluciones alternativas y que justifique la línea de razonamiento al realizar una consulta presente?

SI NO

¿Le gustaría un sistema de ayuda médica con una interfaz sencilla fácil de entender que permita introducir datos para realizar las búsquedas y generar conocimiento?

SI NO

¿Sabía usted que los sistemas expertos incrementan la probabilidad, frecuencia, y consistencia de la toma de buenas decisiones, ayudan a distribuir la experiencia humana?

SI NO

El motivo de la entrevista es para obtener conocimiento de un experto pediatra en enfermedades gastrointestinal y poder elaborar una base de conocimiento experta en el área e implementar una Base de Datos que tenga la inteligencia de inferir con ese conocimiento y devolver un diagnóstico como lo haría un experto médico.

En el siguiente listado se ha elaborado los síntomas que presenta cada enfermedad, las cuales se obtuvo con información a través de internet, la OMS y Enciclopedia Encarta, nos podría ayudar con su experiencia afirmando o contradiciendo los enunciados y en el caso necesario agregar una nueva patología común de la enfermedad tratada.

APENDICE:

Síntomas frecuentes que presenta esta enfermedad (en el caso de presentar fiebre indique su valor en grados).

- tiene falta de apetito SI NO
- tiene dolor abdominal agudo en el lado inferior derecho SI NO
- cojea al caminar SI NO
- tiene diarrea(normal, la común) SI NO
- siente dolor en la zona cercana al ombligo SI NO
- el dolor se intensifica a medida que pasa el tiempo SI NO
- siente dolor al realizar esfuerzo físico menor como tos o respiración profunda SI NO
- tiene fiebre baja(menos de 38.5 °C) *38.5 - 39 - 40 °C* SI NO
- tiene hinchazón abdominal *Relajado* SI NO
- siente nauseas SI NO
- vomita con sangre SI NO

vomita con facilidad (sin dolor) SI NO

al tocar la zona adolorida, se contrae de manera espontánea SI NO

De existir síntomas que usted considere importante favor mencionarlo

muchos pacientes sólo refieren dolor sin nada más que lo acompaña.....

¿Cuál es el procedimiento de diagnóstico? Una vez adquirida esta enfermedad?

El diagnóstico es clínico pero se puede complementar con Rx Abdomen, BHC, PCR, Eco Abdomen, elemental de orina.....

ASCARIASIS:

Síntomas frecuentes que presenta esta enfermedad (en el caso de presentar fiebre indique su valor en grados)

tiene falta de apetito SI NO

tiene dificultad para respirar SI NO

tiene dolor de estómago SI NO

tiene tos SI NO

tiene fiebre(la común) SI NO

tiene distensión abdominal (hinchazón abdominal) SI NO

tiene obstrucción intestinal sólo en casos de ovillos de Ascaris SI NO

esta perdiendo peso SI NO

tiene presencia de lombrices en las heces SI NO

tiene vómito SI NO

De existir síntomas que usted considere importante favor mencionarlo

Hay palidez, puede presentar diarreas, prurito Anal, suena intranquila y cefalea.....

¿Cuál es el procedimiento de diagnóstico? Una vez adquirida esta enfermedad?

.. Coproparasitaria seriado ..

COLERA:

Síntomas frecuentes que presenta esta enfermedad (en el caso de presentar fiebre indique su valor en grados)

- puede haber comido alimentos de dudosa procedencia SI NO
- puede haber comido alimentos mal cocinados SI NO
- tiene calambres musculares SI NO
- tiene dolor abdominal SI NO
- tiene deposición (diarrea) abundante SI NO
- tiene deposiciones pardo/grisáceas blanquesinas como agua de arroz SI NO
- tiene deposiciones amarillentas SI NO
- tiene deposiciones líquidas SI NO
- tiene mucha sed SI NO
- tiene vómito profundo con náuseas SI NO

De existir síntomas que usted considere importante favor mencionarlo

.. fiebre muy elevada ; sarambines ; deshidratación rápida ..

¿Cuál es el procedimiento de diagnóstico? Una vez adquirida esta enfermedad?

.. Cultivo de heces para aislar la bacteria y citología de moco fecal para determinar presencia de focolo infeccioso ..

DIARREA:

Síntomas frecuentes que presenta esta enfermedad (en el caso de presentar fiebre indique su valor en grados)

- tiene la boca seca y pegajosa SI NO
- tiene dificultad para comer y para beber los líquidos..... SI NO
- tiene dolor abdominal SI NO
- tiene 3 o más deposiciones líquidas o semilíquidas en menos de 12 H SI NO
- presenta sangre en las heces *A. veces* SI NO
- tiene fiebre alta SI NO
- tiene gases SI NO
- siente náuseas SI NO
- orina con menos frecuencia *Relativo # de deposiciones* SI NO
- esta perdiendo peso SI NO
- tiene mucha sed SI NO
- tiene vómito profundo con náuseas SI NO

De existir síntomas que usted considere importante favor mencionarlo

por lo general cuadros diarreicos sin vómitos y autolimitados, si hablamos de diarrea solamente.

¿Cuál es el procedimiento de diagnóstico? Una vez adquirida esta enfermedad?

citología de feca en micro fecal, CPS, Nickerson para hongos, H. pilory en heces

GASTROENTERITIS:

Síntomas frecuentes que presenta esta enfermedad (en el caso de presentar fiebre indique su valor en grados)

- tiene fiebre(la común)..... SI NO

- tiene evacuaciones acuosa con pus, sangre o moco SI NO
- tiene vómito amarillento SI NO
- tiene diarrea(normal, la comun) SI NO
- siente nauseas SI NO
- tiene colicos SI NO
- tiene dolor abdominal tipo retortijón SI NO

De existir síntomas que usted considere importante favor mencionarlo

La falta de apetito + palidez + pérdida de peso.

¿Cuál es el procedimiento de diagnóstico? Una vez adquirida esta enfermedad?

*Cultivo de heces, Nickeron en heces,
Citología fecal, Rotavirus, Adenovirus,
Coproparasitario*

ESTREÑIMIENTO:

Síntomas frecuentes que presenta esta enfermedad (en el caso de presentar fiebre indique su valor en grados)

- tiene falta de apetito SI NO
- come alimentos con alto contenido en grasa y bajo contenido en fibra (comidas rápidas, colas) SI NO
- toma poca agua o líquidos durante el día SI NO
- tiene evacuaciones dolorosas con pesadez abdominal SI NO
- tiene evacuaciones duras (heces duras) SI NO
- no defeca durante varios días SI NO
- tiene evacuaciones secas (heces secas) SI NO
- hace poco ejercicio SI NO

De existir síntomas que usted considere importante favor mencionarlo

deposiciones con sangre

¿Cuál es el procedimiento de diagnóstico? Una vez adquirida esta enfermedad?

Capro parasitario, Rx Abdomen 2 posiciones,
Colon por Enemas

SINDROME DEL INTESTINO IRRITABLE O COLITIS NERVIOSA:

Síntomas frecuentes que presenta esta enfermedad (en el caso de presentar fiebre indique su valor en grados)

- tiene dolor tipo cólico normalmente seguido de la expulsión de gases o heces que lo alivian SI NO
- tiene falta de apetito SI NO
- tiene diarrea(normal, la comun) SI NO
- tiene dolor abdominal SI NO
- tiene dolor constante que se agrava después de las comidas SI NO
- se siente estresado, presenta estados de ansiedad emocional o depresión SI NO
- existe la sensación subjetiva de que no ha habido una evacuación intestinal completa relativo SI NO
- tiene presencia de moco en las heces SI NO
- tiene evacuaciones dolorosas con pesadez abdominal SI NO
- se siente fatigado, cansado o debil SI NO
- tiene fiebre(la común) Relativo SI NO
- esta perdiendo peso SI NO
- hace poco ejercicio SI NO
- retiene los gases(no puede liberar sus gases) SI NO

tiene ruidos en la parte del abdomen SI NO
 Siente llenura o inflamación (distensión) abdominal SI NO

De existir síntomas que usted considere importante favor mencionarlo

distensión abdominal

¿Cuál es el procedimiento de diagnóstico? Una vez adquirida esta enfermedad?

Colon por enema, Coproparazitología
 ReCo sigmoidoscopia

GASTRITIS:

Mencionar los síntomas frecuentes que presenta esta enfermedad (en el caso de presentar fiebre indique su valor en grados)

dolor abdominal SI NO
 distensión abdominal SI NO
 gases (dispepsias) SI NO
 Inapetencia SI NO
 sensación de plenitud SI NO
 diarreas alternadas con SI NO
 estreñimiento SI NO
 náuseas SI NO
 vómitos SI NO
 SI NO
 SI NO
 SI NO
 SI NO

¿Cuál es el procedimiento de diagnóstico? Una vez adquirida esta enfermedad?

Coproparatario, Nicheion en heces, H. pi
lony en heces, Endoscopía

Dra. Blanca Gavilana
PEDIATRA
Reg. S. 11765
SANOSER S.A

Dra Blanca Gavilana

Entrevistado

Dra. Blanca Gavilana P
C.I.: 0913516340
Consultorio: Clínica Santamaría
Telf.: 2404650
098419295

Entrevistador

Alex Portilla Panchana
Egresado de Ingeniería en Sistemas
Universidad de Guayaquil

Dra. Blanca Gavilana
PEDIATRA
Reg. S. 11765
SANOSER S.A

ANEXOS 2

Enfermedades:

- Apendicitis
- Ascariasis
- Cólera
- Diarrea
- Gastroenteritis
- Síndrome de Intestino Irritable (Colitis Nerviosa)
- Estreñimiento
- Intoxicación Alimentaria
- Gastritis

En el siguiente listado se ha elaborado los síntomas que presenta cada enfermedad, las cuales se obtuvo con información a través de internet, la OMS y Enciclopedia Encarta, entrevistas con dos especialistas médicos.

APENDICE:

El apéndice es como un pequeño saquito que está en el colón parecida al dedo de un guante, esta situado en el cuadrante inferior derecho del abdomen. Apendicitis es una inflamación del apéndice.

Síntomas comunes:

tiene dolor agudo en el lado inferior derecho del abdomen

cojea al caminar

tiene diarrea(normal, la común)

siente dolor en la zona cercana al ombligo

el dolor se intensifica a medida que pasa el tiempo

siente dolor al realizar esfuerzo físico menor como tos o respiración profunda

tiene fiebre mayor a 38,5 °C a 40°C

tiene falta de apetito

tiene hinchazón abdominal

siente nauseas

tiene vómito

al tocar la zona adolorida, se contrae de manera espontánea

ASCARIASIS:

La Ascariasis es la mas común de las infecciones causada por el parasito intestinal *Ascaris lumbricoides*, se encuentra asociada con una higiene personal deficiente, condiciones sanitarias precarias o lugares en los que se utilizan heces humanas como fertilizante. La infección se produce por contacto con el suelo o cuando se ingieren alimentos o bebidas contaminadas con huevos del parasito. Una vez que han penetrado en el organismo, llegan al intestino delgado, donde liberan las larvas.

Síntomas comunes:

tiene dolor de estómago

tiene tos

tiene falta de apetito

tiene distensión abdominal (hinchazón abdominal)

tiene obstrucción intestinal

esta perdiendo peso

tiene presencia de lombrices en las heces

presenta fiebre (la común)

tiene vómito

COLERA:

El cólera es una infección intestinal aguda, grave, que se caracteriza por la aparición de evacuaciones diarreicas abundantes, con vomito y deshidratación que puede llevar al paciente a acidosis y colapso circulatorio en el termino de 24 horas y en los casos no tratados puede ocasionar la muerte.

El organismo responsable de la enfermedad es el *Vibrio Cholerae*. La única forma de contagio es a través del agua y los alimentos contaminados por heces (en las que se encuentra la bacteria) de enfermos de cólera.

Síntomas comunes:

puede haber comido alimentos de dudosa procedencia

puede haber comido alimentos mal cocinados

tiene calambres musculares

tiene dolor abdominal

tiene deposición (diarrea) abundante

tiene deposiciones pardo/grisáceas blanquecinas como agua de arroz

tiene deposiciones líquidas

tiene mucha sed

tiene vómito profundo con nauseas

SINDROME DEL INTESTINO IRRITABLE O COLITIS NERVIOSA

Si sufre con frecuencia de dolor abdominal y sus evacuaciones son irregulares, con tendencia al estreñimiento, a la diarrea (ambos) y tiene además sensación de llenura, e inflamación o distensión que aumentan el volumen de su abdomen, obligándolo a aflojar su ropa, usted podría tener Colon irritable, también denominado "Síndrome del Intestino Irritable". Es frecuente que el dolor se presente en las mañanas, cuando inicia la actividad del colon o intestino grueso. El estrés, la ansiedad emocional o la depresión suelen empeorar los síntomas.

Estos síntomas pueden presentarse de forma continua o tener exacerbaciones y remisiones al pasar las semanas, meses o años, habiendo quienes incluso se acostumbran a ellos.

Síntomas comunes:

tiene dolor tipo cólico normalmente seguido de la expulsión de gases o heces que lo alivian

tiene diarrea(normal, la común)

tiene dolor abdominal

tiene dolor estomacal constante que se agrava después de las comidas

tiene evacuaciones dolorosas con pesadez abdominal

existe la sensación subjetiva de que no ha habido una evacuación intestinal completa

tiene evacuaciones con menos frecuencia o no defeca durante varios días'

se siente fatigado, cansado o debil

tiene falta de apetito

Siente llenura o inflamación (distensión) abdominal

esta perdiendo peso

hace poco ejercicio

retiene los gases(no puede liberar sus gases)

tiene ruidos en la parte del abdomen

se siente estresado, presenta estados de ansiedad emocional o depresión

DIARREA:

La OMS define la Diarrea como la producción de 3 o más deposiciones líquidas o semilíquidas en 12 horas. Puede presentarse de forma aguda o crónica.

Aguda de corta duración, de 48-72 horas, normalmente relacionada con las Infecciones bacterianas o víricas.

Crónica implica la persistencia de síntomas por encima de las 2-3 semanas, normalmente relacionada con trastornos funcionales.

También puede ser producida por la toma de sustancias toxicas que lesionan el intestino, por situaciones de tensión emocional o por estados nerviosos alterados.

Síntomas comunes:

tiene dolor abdominal

tiene 3 o más deposiciones líquidas o semilíquidas en menos de 12 H

tiene gases

orina con menos frecuencia

esta perdiendo peso

tiene mucha sed

ESTREÑIMIENTO:

El estreñimiento se define como; una disminución en la frecuencia de las defecaciones, en comparación con el patrón normal del niño (algunos médicos definen el estreñimiento como menos de tres defecaciones por semana).

Síntomas comunes:

come alimentos con alto contenido en grasa y bajo contenido en fibra (comidas rápidas, colas

toma poca agua o líquidos durante el día

tiene evacuaciones dolorosas con pesadez abdominal

tiene evacuaciones duras (heces duras)

tiene evacuaciones con menos frecuencia o no defeca durante varios días

tiene evacuaciones secas (heces secas)

tiene falta de apetito

hace poco ejercicio

GASTROENTERITIS:

La inflamación de la mucosa del estómago se denomina gastritis, mientras que la de los intestinos se conoce como enteritis. Cuando hay inflamación en ambos órganos

se produce una gastroenteritis, causadas por virus (rotavirus, virus Norwalk), bacterias o protozoos. La gastroenteritis es un trastorno inflamatorio que se manifiesta principalmente con un síndrome diarreico.

Síntomas comunes:

tiene diarrea(normal, la común)

tiene dolor abdominal tipo retortijón

tiene cólicos

tiene fiebre(la común)

siente nauseas

tiene vómito amarillento

/* Patologías pendientes a agregar a la Base de Gastromedical */

INTOXICACION ALIMENTARIA:

La intoxicación alimentaria se produce a consecuencia de ingerir alimentos contaminados con bacterias comunes como E. coli o estafilococo.

Éstas intoxicaciones se pueden presentar en una sola persona o en un conjunto de ellas como brote, debido a que comieron lo mismo, sobretodo si son concentraciones de personas o escuelas, etc., donde se da un manejo de alimentos sin refrigerar durante largos períodos de tiempo o su preparación no es higiénica.

Las personas que tienen mayor riesgo de intoxicación son los niños y las personas ancianas.

Síntomas comunes:

tiene cólicos

tiene evacuaciones acuosa con pus, sangre o moco

tiene dolor de cabeza

tiene fiebre y escalofríos

Después de 2 a 6 horas de haber ingerido alimentos presentó molestias estomacales

siente nauseas

tiene vomito

Debido a que la intoxicación alimentaria es una enfermedad “auto limitante”, es decir; tiene un ciclo que comienza y termina en un tiempo programado, usted generalmente se recuperará en un par de días aunque en algunos casos, como en la infección por *Campylobacter*, puede ser de hasta una semana.

Tratamiento:

El objetivo principal deberá ser evitar la deshidratación y conseguir el alivio de los síntomas.

No consuma alimentos sólidos hasta que la diarrea haya pasado y evite los productos lácteos que pueden empeorarla (estado temporal de intolerancia a la lactosa).

Consuma líquidos ricos en electrolitos como el pedialite, gatorade, y si le gusta lo natural: agua de coco.

GASTRITIS:

La Organización Mundial de Salud (OMS) define a la gastritis crónica como inflamación de la mucosa gástrica que puede ser perjudicial para la salud.

La gastritis consiste en la inflamación o hinchazón de la mucosa gástrica (cubierta que tapiza las paredes del estomago), puede ser crónica o aguda, es un padecimiento provocado principalmente por el abuso de analgésicos, bebidas alcohólicas, tabaquismo y por una infección estomacal (por la bacteria *Helicobacter pylori*), sin embargo, también puede ser incitada por la ansiedad y el estrés.

Síntomas comunes:

tiene acidez o ardor en el estomago

tiene falta de apetito

tiene dispepsia (digestión difícil o dolorosa)

siente su estomago como una bola

se siente estresado, presenta estados de ansiedad emocional o depresión

tiene flatulencias (gases en el estómago)

siente llenura o inflamación (distensión) abdominal

tiene reflujo (Erupto tipo vomito)

tiene vómito

La gastritis es curable si es detectada a tiempo, no obstante es conveniente e importante la atención medica; aquí encontraras algunas recomendaciones que pueden disminuir el riesgo a contraerla o reducir el dolor.

1. Tomar agua (ocho vasos al día) en vez de bebidas gaseosas, café, o jugos cítricos.
2. No saltarse los horarios de comida.
3. Evitar tomar analgésicos en exceso pues son irritantes para el estomago.

4. No comer alimentos grasosos, condimentados, ni picantes.
5. Tratar de controlar el estrés y la ansiedad.
6. Consumir frutas de preferencia con cascara como la manzana, duraznos etc.
y reducir los frutos cítricos.
7. Consumir con regularidad verduras y leguminosas.
8. Aumentar el consumo de fibra.
9. Reducir el consumo del tabaco, preferentemente eliminarlo.
10. Consumir con medida bebidas alcohólicas.
11. Evitar ingerir comida o bebidas demasiado calientes

CAPÍTULO V

CONCLUSIONES Y RECOMENDACIONES

CONCLUSIONES

De acuerdo a lo planteado en los objetivos se logró demostrar mediante esta investigación todo lo propuesto.

Se diseñó una base de conocimiento según la metodología de Buchanam la cual consiste en adquirir conocimiento de distintas fuentes; especialista medico, bibliografía de internet y Enciclopedia Encarta 2009.

Mediante reglas de inferencia, descritas en el Capítulo 2, el proceso de diseño y elaboración del prototipo fue implementado, determinando las tareas criticas a solucionar y en las cuales debería enfocarse la funcionalidad del prototipo.

Se comprobó el empleo de un lenguaje de quinta generación SWI-Prolog; ya que este lenguaje de programación equilibra las propiedades del modelo abstracto de la Programación Lógica y por otro lado consigue que la implementación sea eficiente.

La integración de tecnologías java y prolog, y en especial de base de datos relacional deja un camino para el desarrollo de sistemas inteligentes. Con técnicas propias de la Inteligencia Artificial y los Sistemas Expertos para la creación de sistemas médicos

que permitan diagnosticar en la primeras consultas a través de síntomas que presente el paciente.

Pueden ser utilizadas para analizar el conocimiento y para aplicar el razonamiento deductivo sobre dicho conocimiento. Es flexible, ya que permite representar, en forma sencilla, los hechos sin importar para qué son utilizados.

No olvidar que la integración entre bases de datos relacional y sistemas de conocimientos es esencial en una BDI, por lo que estas herramientas, además de hacer más fácil su integración, permitirán un mantenimiento más eficiente de la BDI.

Es conveniente indicar, que los usuarios de este tipo sistemas no son expertos, por lo que el sistema debe ser una ayuda y no un problema mas.

El objetivo principal del proyecto no ha sido construir un Sistema Experto que simulase exactamente el diagnóstico de un pediatra, sino complementar a los profesionales, expertos en la materia, sirviéndoles de ayuda. Así como ser punto de referencia para aquellos usuarios que no sean expertos y puede utilizar la aplicación como utilidad práctica para su profesión.

Se ha presentado la aplicación a diversos usuarios sin conocimientos informáticos para comprobar la sencillez de la interfaz y los resultados obtenidos fueron satisfactorios siendo, según ellos, una aplicación sencilla y fácil de manejar.

Se realizó la implantación de la aplicación usando Java 1.6 NetBeans IDE 6.5 que presentaba las características idóneas necesarias para la construcción de ésta y

utilizando MySQL 5 para la consulta de la Base de Datos y almacenamiento de información. Todo realizado en un software de código abierto.

Java que permite la portabilidad de la aplicación independientemente de la plataforma que se esté desarrollando. Además se diseñó el prototipo GastroMedical para que pueda ser extendido en otras áreas de aplicación. Con todo ello, se puede concluir que se cumplen los objetivos iniciales del proyecto.

Los conflictos después de ser sobrellevados, permitieron dar una serie de conclusiones sobre las cuales se pueden edificar las bases para futuras aplicaciones que integren sistemas de datos inteligente y programación convencional. Para crear una óptima base de datos relacional no basta sólo con conocer la teoría básica del modelo entidad relación. Es necesario conocer con profundidad el área para la cual se está diseñando la base de datos, y de esta forma crear un modelo entidad relación.

Para programar en Java (y en cualquier otro lenguaje de programación general) es conveniente examinar la amplia gama de herramientas de software existentes diseñadas para este lenguaje antes de escoger cualquiera de ellas, con el fin de encontrar la que aporte más en la consecución de las metas programadas.

Una sección previo de las herramientas disponibles es fundamental para permitir minimizar el tiempo al desarrollo de esta aplicativo. La interconexión de Java con MySQL demandó una investigación adicional para poder configurar la conectividad a través del Connector JDBC.

Tanto MySQL como Java son herramientas que tienen respaldo tecnológico de empresas de software y lo cual permitió encontrar en la documentación establecida la información necesaria para realizar las correcciones necesarias en la implementación de esta interconexión.

Para diseñar un aplicativo que involucre dos o más tecnologías se deben escoger herramientas que posean suficiente documentación y recursos adicionales, los cuales puedan servir de material de consulta o apoyo en caso de presentarse dudas o inconvenientes en el manejo o adaptación de dichas herramientas.

La interconexión de Java con MySQL, para comunicar Java con swi-Prolog planteó un problema especial, puesto que no es mucha la documentación que se puede encontrar sobre este tema. Esta interconexión, implementa la interfaz (paquete) `jpl.jar`, requirió una profundización en Java sobre la clase abstracta `jpl`.

Adicionalmente, el tener que ejecutar predicados Prolog desde el Ide NetBeans Java y recibir información desde dicho software, demandó verificaciones de que la información obtenidas por el aplicativo fueran las esperadas, que en caso de error se pudiera deducir que el problema estaba en la interconexión y no en el aplicativo. Superadas las dificultades y comprobada la integración, se pudo demostrar la funcionalidad que permite la interfaz `jpl` para integrar Prolog en cualquier entorno Java.

Con el trabajo de desarrollo de este aplicativo se demuestra que es posible seguir extendiendo el uso de las BDI a áreas específicas de la medicina, en este caso la

gastroenterología, a través de herramientas de software libre, las cuales pueden ser otorgadas por sus fabricantes sin restricción alguna, como lo describe la Fundamentación Legal del Capítulo 2. Solo se requiere el recurso humano, el cual consiste de un ingeniero en sistemas y un experto médico, apoyados de ser necesario en bibliografía recomendada.

RECOMENDACIONES

Demostrada la confiabilidad del prototipo GastroMedical, se recomienda la utilización del mismo en diagnósticos pediátricos en el área de gastroenterología para poder obtener beneficio para el paciente de 6 a 11 años, dentro de las primeras consultas. Se aconseja alimentar la base de conocimientos para llegar a diagnósticos específicos en función al conocimiento cada vez más amplio del especialista médico.

Las bases de datos inteligentes son un modelo que en nuestro medio es joven que aun le falta investigación, por esta razón pierde la simplicidad y la facilidad de aprendizaje del usuario final de las bases de datos relacionales; ya que no hay un lenguaje estándar a la hora de realizar reglas.

Este proyecto es el primero que se implementa en la Carrera de Ingeniería en Sistemas Computacionales, Facultad de Ciencias Físicas y Matemáticas de la Universidad de Guayaquil, utilizando la tecnología de la interface jpl para la conexión entre Java y el lenguaje de programación lógica Prolog quienes integran a la base de datos MySQL.

Estas herramientas de software al ser gratuitas, las investigaciones pueden seguir este camino para desarrollar nuevas aplicaciones de sistemas de Base de Datos Inteligentes aplicados en otras áreas.

Este trabajo deja como resultado un prototipo consistente cuya estructura es totalmente funcional.

Los Sistemas de datos inteligentes no deben considerarse como soluciones aisladas respecto a otros desarrollos de software. La aplicación del software convencional debe realizarse allí donde tenga sentido hacerlo.

Las herramientas con que se implementó el aplicativo (Prolog, Java, Mysql) no son exigentes en cuanto a capacidad de máquina para su funcionamiento. Sin embargo, se sugiere disponer de un computador de capacidad media en previsión de futuras ampliaciones del aplicativo o de la adición de nuevos módulos.

Aunque el aplicativo tiene una interfaz gráfica amigable e intuitiva, es recomendable que el usuario lea con detenimiento el manual del usuario para comprender en su totalidad la filosofía del funcionamiento del software y lograr obtener el mayor beneficio a través de una correcta utilización.

BIBLIOGRAFÍA

LIBRO

Nelson Tratado de Pediatría

Volumen I

Decimo Sexta Edición

Editorial: McGraw - Hill Interamericana

DIRECCIONES WEB

BDI Sergio Antonio Becerra Zepeda

<http://www.buenastareas.com/ensayos/Bases-De-Datos-Inteligentes/806838.html>

<http://www.scribd.com/doc/29606202/Trabajo-de-Base-de-Datos-Inteligentes>

Adquisición del conocimiento

<http://www.tesis.ufm.edu.gt/pdf/3367.pdf>

OMS

<http://www.who.int/topics/es/>

<http://www.who.int/mediacentre/factsheets/fs107/es/index.html>

Enfermedades gastrointestinales.html

<http://www.esmas.com/salud/enfermedades/infeciosas/434933.html>

"Apendicitis." Microsoft® Student 2009 [DVD]. Microsoft Corporation, 2008

"Cólera." Microsoft® Student 2009 [DVD]. Microsoft Corporation, 2008.

"Ascariasis." Microsoft® Student 2009 [DVD]. Microsoft Corporation, 2008.

Gastroenteritis." Microsoft® Student 2009 [DVD]. Microsoft Corporation, 2008

"Gastritis." Microsoft® Student 2009 [DVD]. Microsoft Corporation, 2008

Dr. Gary Gaugler/Photo Researchers, Inc. Microsoft ® Encarta ® 2009. ©

Colon irritable." Microsoft® Student 2009 [DVD]. Microsoft Corporation, 2008.

Microsoft ® Encarta ® 2009. © 1993-2008 Microsoft Corporation.

Intoxicación alimentaria." Microsoft® Student 2009 [DVD]. Microsoft Enciclopedia Encarta

Dra. Lorena Calderón de C. Gastroenteróloga – Nutricionista Asesora Medica LLS
Kazura JW. Nematode infections. In: Goldman L, Ausiello D, eds. Cecil Medicine.
23rd ed. Philadelphia, Pa: Saunders Elsevier; 2007: chap 378.

www.pediatraldia.cl

Gastroenteorologia - Criterios diagnósticos de Roma II

<http://www.iqb.es/digestivo/diagnosis/romatoc.htm>

Medicina basada en la evidencia

<http://www.infodoctor.org/rafabravo/torpes.htm>

<http://www.infodoctor.org/rafabravo/mbe2.htm>

Definición de pediatría

<http://definicion.de/pediatria/>

Gastro.html

<http://www.dgsm.unam.mx/web/gastro.html>

Patologías

<http://www.buenastareas.com/ensayos/Apendicitis-En-Ni%C3%B1os/552361.html>

Que-es-la-Heuristica

http://www.taringa.net/posts/info/2379078/_Que-es-la-Heuristica_.html

Qué son las base de datos

<http://www.sg.inter.edu/acc/prMIS206/rodriguez/pag-03.htm>

MySQL 5.0 Reference Manual.pdf

<http://dev.mysql.com/doc/>

Sistema de Gestión de Base de Datos (SGBD)

<http://www.maestrosdelweb.com/principiantes/%C2%BFque-son-las-bases-de-datos/>

<http://es.kioskea.net/contents/bdd/bddintro.php3>

sistema_gestor_de_base_de_datos_sgbd.php.htm

http://www.error500.net/garbagecollector/archives/categorias/bases_de_datos/sistema_gestor_de_base_de_datos_sgbd.php

Sistema experto pediatra en enfermedades gastrointestinales

<http://eventos.spc.org.pe/jpc2007/MyReview/FILES/p22.pdf>

Conexion Java con Swi-Prolog

<http://soytami.blogspot.com/2009/07/conexion-java-con-swi-prolog.html>

<http://jc-info.blogspot.com/2008/12/cmo-programar-en-swi-prolog.html>

Conectar Java con Prolog usando JPL

<http://jc-info.blogspot.com/2009/09/conectar-java-con-prolog-usando-jpl.html>

JPL - Conectar Java con Prolog de David Mota

<http://dmottab.blogspot.com/2010/03/jpl-conectar-java-con-prolog.html>

Free-Software-vs-Open-Source.html

<http://www.atinachile.cl/content/view/2481/Free-Software-vs-Open-Source.html>

ley-organica-de-educacion-superior.pdf

<http://blogs.utpl.edu.ec/secretariageneral/files/2010/09/ley-organica-de-educacion-superior-con-veto-parcial.pdf>

Ley de informática en Ecuador

<http://www.informatica-juridica.com/legislacion/ecuador.asp>

http://www.derechoecuador.com/index.php?option=com_content&task=view&id=3091&Itemid=426

<http://www.informatica.gov.ec/descargas/sids11014.pdf>

<http://www.informatica.gov.ec/>