



PONTIFICIA UNIVERSIDAD CATÓLICA MADRE Y MAESTRA – PUCMM
FACULTAD DE CIENCIAS DE LAS INGENIERÍAS
Departamento de Ingeniería en Sistemas y Computación – ISC

Asignatura:

ST-ISC-317-001 Programación Lógica

Trabajo de Curso:

Sistema Experto, Diagnóstico de Enfermedades

Autores:

2012-0467 Manuel Tolentino

2013-0940 Shantall Girón

2013-1134 Darlenys Gómez

Tutores:

Prof. Antonio Rivero y Prof. Miguel Moronta

Entregado en Fecha:

Martes, 25 de Julio de 2017



RESUMEN / ABSTRACTO DEL PROBLEMA

Se debe realizar un sistema que realice diversas preguntas al usuario, y en base a sus respuestas, ofrezca recomendaciones u opciones al mismo. Sistemas de este tipo sirven como asistentes para diagnóstico de enfermedades, asistentes de menú de comida en base a preferencias, asistentes para gimnasio, entre otros. En este caso el sistema está orientado al diagnóstico de enfermedades. Se deben contemplar al menos 10 consultas de parte del usuario, dependiendo de cada situación, con un sistema de inferencias (reglas) y una base de conocimientos (hechos) de por lo menos 150 líneas de código.

The system must be created to ask the user various questions and, based on their answers, offers recommendations or options to the user. Systems of this type work as disease diagnosis assistants, food menu assistants based on preferences, gym assistants, among others. In this case the system is oriented to the diagnosis of diseases. At least 10 user queries must be considered, depending on each situation, with a system of inferences (rules) and a knowledge base (facts) of at least 150 lines of code.



PONTIFICIA UNIVERSIDAD CATÓLICA MADRE Y MAESTRA – PUCMM
FACULTAD DE CIENCIAS DE LAS INGENIERÍAS
Departamento de Ingeniería en Sistemas y Computación – ISC
Asignatura: ST-ISC-317-001 Programación Lógica

TABLA DE CONTENIDO

| | |
|---------------------------------------|----|
| Resumen / Abstracto del Problema..... | 1 |
| Análisis..... | 1 |
| Cuerpo del Informe..... | 3 |
| Código Fuente | 10 |
| Condiciones de Error | 11 |
| Conclusiones..... | 12 |
| Recomendaciones | 13 |
| Referencias Bibliográficas | 14 |

ANÁLISIS

Breve descripción del proyecto

Los Sistemas Expertos, son sistemas informáticos que simulan el proceso de aprendizaje, memorización, razonamiento, comunicación y acción en consecuencia de un experto humano en cualquier rama de la ciencia.

Estas características le permiten almacenar datos y conocimientos, sacar conclusiones lógicas, tomar decisiones, aprender de la experiencia y los datos existentes, comunicarse con expertos humanos, explicar el porqué de las decisiones tomadas y realizar acciones como consecuencia de todo lo anterior.

Técnicamente un sistema experto, contiene una base de conocimientos que incluye la experiencia acumulada de expertos y un conjunto de reglas para aplicar ésta base de conocimientos en una situación particular que se le indica al programa. Cada vez el sistema se mejora con adiciones a la base de conocimientos o al conjunto de reglas.

En esta ocasión se pretende realizar un sistema experto orientado al ámbito medicinal, específicamente al diagnóstico de enfermedades. Dicho sistema debe contar con una interfaz de usuario, la cual debe crear una especie de diálogo (sistema-usuario). El usuario va a introducir las posibles respuestas a las preguntas. A partir de dichas respuestas, el sistema deberá dar una posible solución/enfermedad al usuario utilizando las reglas de inferencia que contiene en su base de conocimiento.

Objetivos del proyecto

- Interacción con el usuario.
- Capturar respuestas del usuario y utilizarlas para dar resultados óptimos.
- Tener una base de conocimiento compleja con datos verídicos.
- Creación de reglas que cumplan con técnicas de inferencias (IF/THEN) para deducir una solución al problema planteado.
- Interfaz gráfica en otro lenguaje/plataforma.
- Comprobar entre las posibles opciones contenidas en la base de conocimiento para dar un buen resultado, tomando en cuenta de que las preguntas no sean redundantes e innecesarias.

División del problema y mecanismos utilizados

Debido al principal propósito del sistema, el cual es interactuar con el usuario, obtener respuestas claras (yes or no questions) y dar un resultado óptimo, es utilizada la técnica de inferencia que usa las reglas "IF THEN" para ayudar a romper repetidamente la meta principal en sub-objetivos más pequeños, y así poder ser más fáciles de probar.

Una vez dividido el objetivo principal en sub-problemas es más fácil de operar, este método se implementa a través de la creación de reglas para cada enfermedad, contando cada una con una serie de síntomas. De esta manera, resulta ser más fácil ir verificando los síntomas a través de las respuestas que va dando el usuario.

Al recibir una respuesta "No" en un síntoma X, el sistema automáticamente es capaz de ignorar todas las posibles enfermedades que contengan dicho síntoma.

Luego de realizar un recorrido por todas las reglas y haber finalizado con todas las posibles preguntas al usuario, el sistema será capaz de dar como resultado al usuario la enfermedad dependiendo de los síntomas obtenidos a través de su interacción.

Componentes del sistema

-Base de conocimiento: consiste en conocimiento sobre el dominio del problema en forma de bases de datos estáticas y dinámicas.

-Conocimiento estático: consiste en reglas y hechos que se cumplen como parte del sistema y no cambian durante la ejecución del sistema.

-Conocimiento dinámico: consiste en hechos relacionados con una consulta particular del sistema. Al comienzo de la consulta, la base de conocimientos es dinámica, a menudo llamada memoria de trabajo, está vacía. A medida que avanza la consulta, la base de conocimientos dinámicos crece y se utiliza junto con el conocimiento estático en la toma de decisiones. La memoria de trabajo se borra al final de la consulta del sistema.

-Interfaz de usuario: encargada de interactuar con el usuario para obtener respuestas que ayudan al sistema a tomar decisiones.

CUERPO DEL INFORME

-Capítulo 1 – Descripción del Negocio

Un sistema experto es un sistema de información basado en el conocimiento que usa su 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.

Un sistema experto debe contar con una interfaz, encargada de interactuar con el usuario y recolectar información necesaria para ser evaluada en las reglas contenidas en la base de conocimiento de dicho sistema.

Debe ser capaz de evaluar diferentes alternativas basadas en la información recolectada y dar un resultado óptimo.

Aunque existen muchos casos y condiciones impredecibles en el diario vivir, en el ámbito que sea orientado debe estar estructurado para ser capaz de llegar a resultados esperados.

Si existe un caso que el sistema no puede manejar, el usuario debe ser notificado de alguna manera u otra.

Se debe crear una conexión entre la interfaz y la parte lógica del sistema, de manera que resulte amigable y sencilla para el usuario sin importar que tan complejas o extensas sean las reglas que conforman el sistema.

- Capítulo 2 – Solución Propuesta

Para la solución del problema se crearon reglas para determinar las posibles enfermedades, además reglas para verificar los síntomas que X enfermedad contiene. El sistema cuenta con el mecanismo práctico para ir guardando las respuestas del usuario en cada iteración, ayudando a ir deduciendo cual es la posible enfermedad o simplemente ir descartando enfermedades.

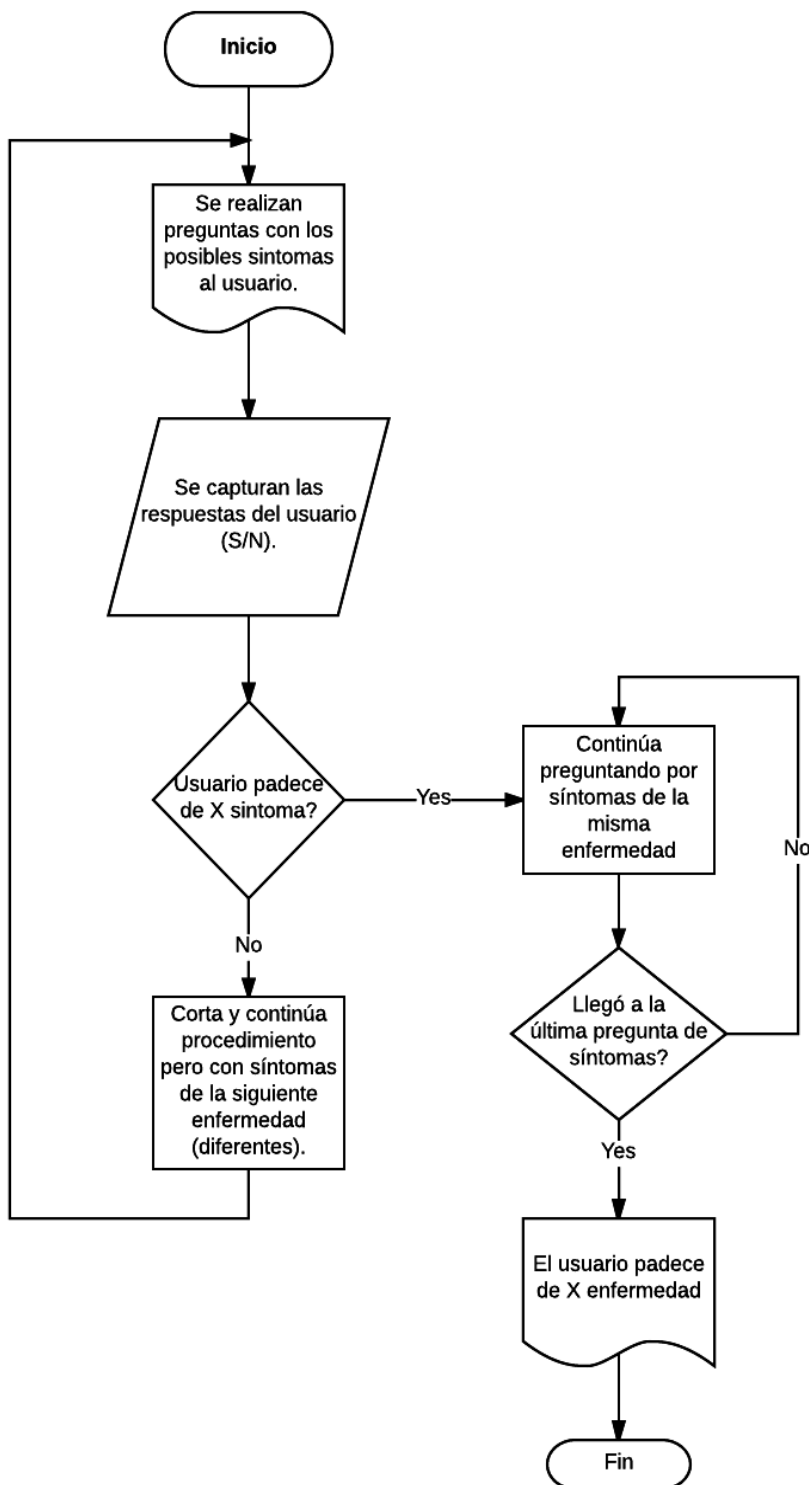
El sistema inicia con el predicado de pregunta, quien es el encargado de realizar el diálogo con el usuario para obtener información sobre varios síntomas.

Los síntomas son considerados como positivos o negativos (Si/No), estos se van verificando en la base de conocimiento estática y la "hipótesis (enfermedad_sintoma)" se satisface en función de los datos almacenados sobre los síntomas.

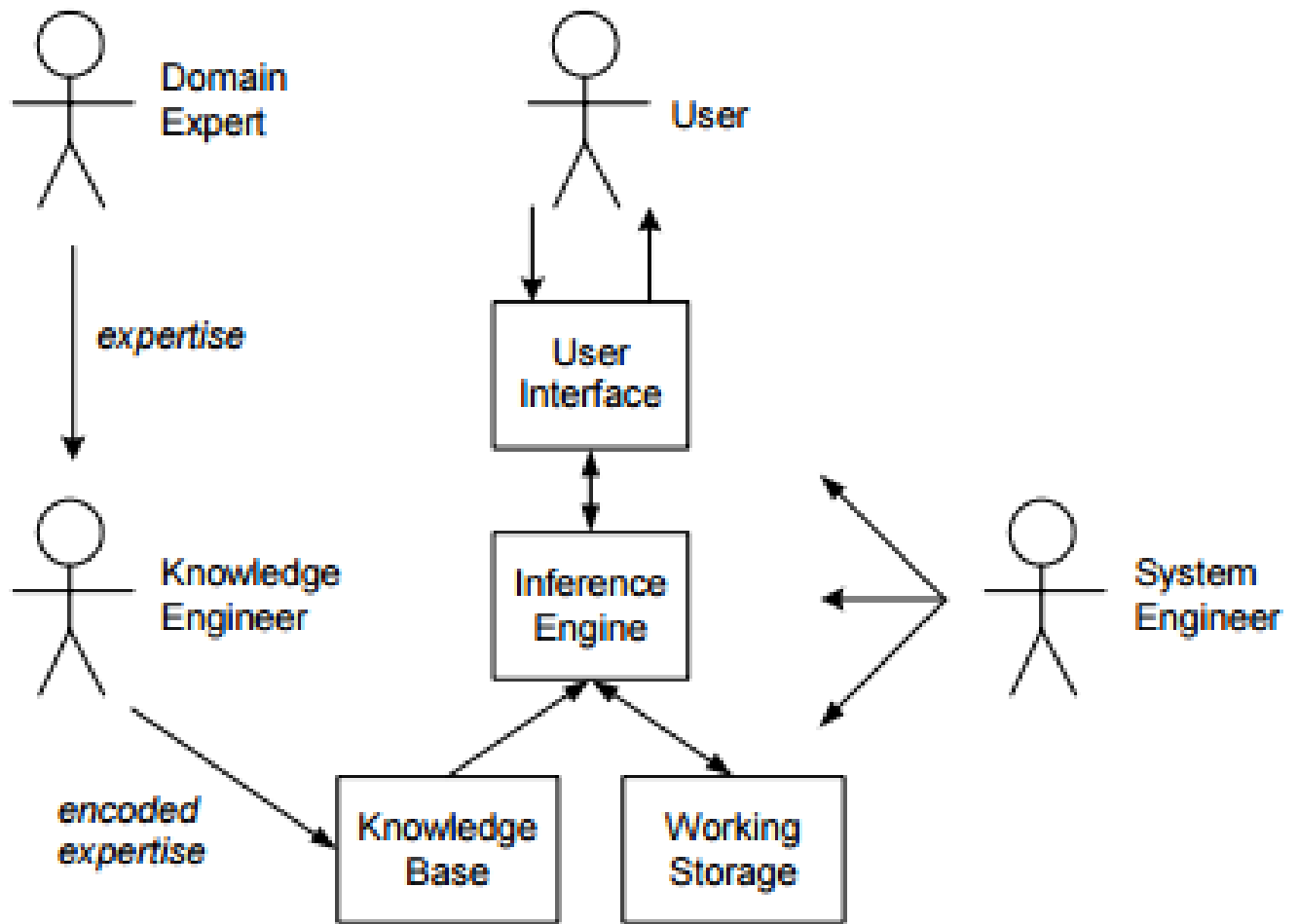
Si se cumple el objetivo de la hipótesis, entonces se muestra la enfermedad, de lo contrario se muestra 'no se puede diagnosticar la enfermedad'.

Puede suceder el caso de que el usuario tenga síntomas de varias enfermedades, el sistema será capaz de dar como resultados todas las posibles enfermedades que coinciden con dichos síntomas.

Diagrama de flujo



Caso de uso del sistema



Capturas de pantalla

Sistema Experto - Diagnostico de enfermedades

Responda las siguientes preguntas respecto a los sintomas que padece:

Usted tiene estornudos ? s/n

SI NO

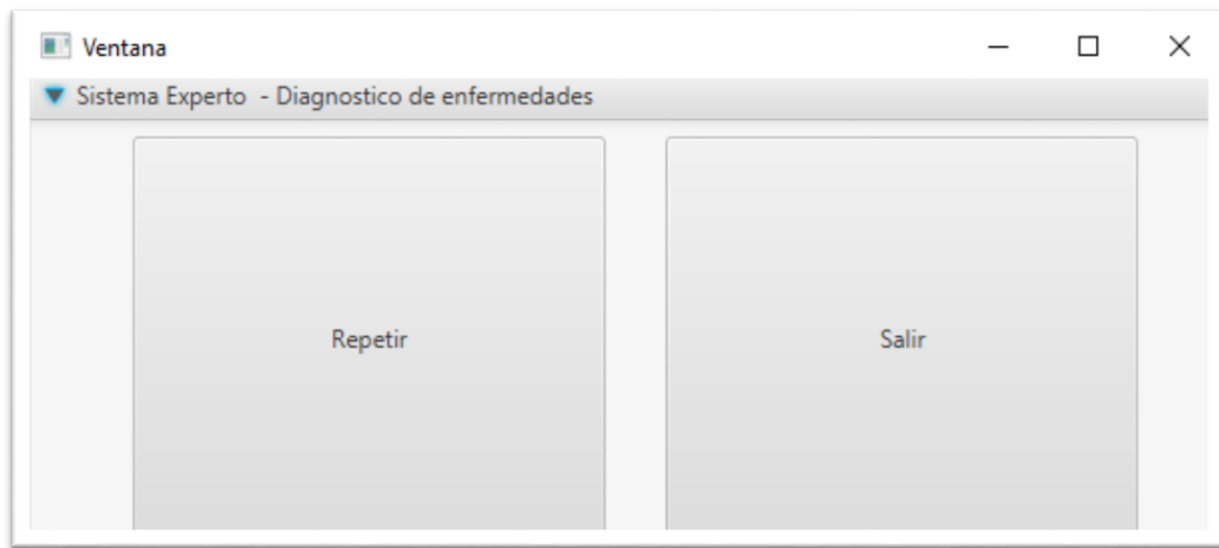
Cancelar

Ventana

Sistema Experto - Diagnostico de enfermedades

| Enfermedades | Descripciones |
|--------------|---|
| alergia | Es un conjunto de alteraciones de carácter respiratorio, nervioso o eruptivo que se producen en el sistema inmunológico por una extrema |
| | |
| | |
| | |
| | |
| | |
| | |
| | |
| | |
| | |

SALIR



Aspectos relevantes del código, instrucciones, etc.

Para empezar cabe resaltar que se buscaron distintas técnicas para lograr integrar la interfaz con prolog, desde lenguajes como JAVA, PHP y C#; siendo la solución satisfactoria JAVA combinado con PHP. Describiendo mejor como se integraron estos lenguajes con prolog, se utilizó PHP para crear un servicio el cual se comunica con prolog ejecutando comandos como si fueran nativos desde la consola. En cuanto a JAVA realizamos get a una url la cual nos devuelve por parámetros lo que le pedimos.

Luego de que ya tuvimos la conexión con los datos de prolog se utilizó javafx para desarrollar las ventanas y crear cierta lógica a la cual el usuario debía entender, esta consistió en la se le realizaba una pregunta al usuario y este debía responder si o no por medio de dos botones.

Cabe destacar que se buscó poder crear nuevas enfermedades mediante un assert a la base de datos dinámica en prolog, para luego obtenerlas realizando un listing de esas que insertamos. Sin embargo, no existe la suficiente documentación que explique e incluso ejecutando el comando directo a la consola funcionaba, pero nunca fue satisfactorio desde nuestro servicio PHP.

CÓDIGO FUENTE

Código fuente anexado junto con este documento.

CONDICIONES DE ERROR

Hay ocasiones en la que una persona puede tener una enfermedad y no contar con todos los síntomas. En nuestro sistema, no es tomado en cuenta este tipo de caso, es decir, para una persona tener una enfermedad debe padecer de todos los síntomas declarados en la base de conocimiento.

CONCLUSIONES

A partir de la realización de este proyecto consideramos que se han logrado los objetivos planteados, permitiendo al usuario interactuar con una interfaz, que, sin saberlo, interactúan con reglas y hechos contruidos con información verídica definidos en el lenguaje de Prolog.

Para alcanzar dicho objetivo se hizo uso de los lenguajes PHP y Java, esto después de intentar la implementación solo en Java y posteriormente en C#. Los cambios surgieron debido a la poca documentación encontrada sobre vincular, manejar y adaptar nuestra lógica y otros lenguajes con Prolog.

Para nuestro equipo resultó ser un proyecto sumamente interesante, ya que se pudo ver un uso más real donde Prolog puede implementar y satisfacer un problema real, lo que nos llevó a verlo desde otro punto de vista al que lo habíamos visto anteriormente.

RECOMENDACIONES

Un aspecto sumamente importante que se puede mejorar en el presente proyecto sería que a partir de diagnosticar una enfermedad, al usuario se le informe las posibles causas de dicha enfermedad, tratamientos que puede seguir, etc.

En el ámbito del usuario administrador, sería bueno que se puedan introducir enfermedades y sus síntomas a través de la interfaz.

El sistema actual se basa en reglas de inferencia, utilizando modus ponens (IF/THEN), el cual fue de mucha utilidad para la lógica del mismo. Pero como existen casos en los cuales un paciente no presenta todos los síntomas de X enfermedad, sería bien que se implementara el mecanismo de asistirle y ser capaz de encontrar una enfermedad, que, aunque no contenga todos los síntomas especificados, sea la posibilidad o deducción más cercana.

REFERENCIAS BIBLIOGRÁFICAS

<http://www.expertsystem.com/>

https://www.tutorialspoint.com/artificial_intelligence/artificial_intelligence_expert_systems.htm

<https://www.journals.elsevier.com/expert-systems-with-applications>

<http://elvex.ugr.es/decsai/iaio/slides/A2%20Expert%20Systems.pdf>

<http://www.amzi.com/ExpertSystemsInProlog/xsipfrtop.htm>

https://en.wikibooks.org/wiki/Expert_Systems/Prolog

<https://www.metalevel.at/prolog/expertsystems>