

Dr. Log

Mariano Muñoz Masís 2016121607, Karla Rivera Sánchez 2016100425, Luis Daniel Prieto Sibaja 2016072504

Área Académica de Ingeniería en Computadores

Lenguajes, Compiladores e Interpretes

Instituto Tecnológico de Costa Rica

Abstract—Expert systems (ES) are computer applications that involve non-algorithmic experience to solve a certain type of problem. For example, expert systems are used for diagnosis at the service of humans and machines. There are ES that play chess, that make financial decisions, that configure computers, that supervise real time systems, that decide insurance policies, and carry out other tasks that require human experience. The objective of this task is to develop an ES for a medical office, the interface must be completely natural using the Spanish language and the ES must behave like a human doctor.

I. MANUAL DE USUARIO

El presente programa tiene dos partes una desarrollada en el lenguaje ProLog y otra desarrollada en Java, la base de conocimiento (parte lógica del programa) y la interfaz de usuario respectivamente.

A. Lógica del Programa - ProLog

En esta sección del programa se detalla todo lo necesario para el funcionamiento del sistema experto, y está hecho para funcionar independiente de la interfaz del usuario.

1) *Abrir el IDE*: El primer requerimiento es tener instalado el IDE SWI-Prolog si no se tiene instalado se puede descargar del siguiente enlace: <http://www.swi-prolog.org/download/stable> Una vez instalado se ejecuta el programa y se espera que cargue.

2) *Cargar el Archivo DrLog*: Para cargar el archivo de DrLog lo que hacemos es que una vez tenemos abierto "SWI-Prolog" vamos a la barra de herramientas, a "File" y luego a "Consultar..." y buscamos el archivo en el ordenador. Una vez hecho esto el programa correrá automáticamente.

3) *Interactuar con DrLog*: Para iniciar la conversación basta con decir "hola" o "Buenos Días" entonces comenzará una conversación en lenguaje natural.

B. Interfaz de Java

Para la interfaz de java se necesita tener instalado Netbeans, lo que procede es abrir el proyecto desde el IDE (Netbeans), y luego darle "ejecutar", aparecerá una ventana de inicio, y seguidamente la interfaz que interactúa con el usuario. De manera similar a la forma de interactuar con Prolog se interactúa con la interfaz de Java, se hace de forma natural.

C. Iniciar la interfaz de Usuario

cómo ejecutar el programa.

II. HECHOS Y REGLAS

A. Hechos

Para la base de datos de DrLog, se toman como hechos las causas, tratamientos y prevenciones, esto porque se facilita a la hora de darle al paciente la información, y solo se accede a los mismos buscando como primer parámetro la enfermedad que se le ha diagnosticado previamente al paciente, y como segundo parámetro una lista con la frase que coincide con la información que el paciente está solicitando. También se toman como hechos las conversaciones las cuales constituyen el parser, se evalúan diferentes entradas posibles del usuario como primer parámetro del hecho, como segundo parámetro se encuentra la respuesta predefinida que debe dar DrLog, tanto para saludos como para cuando se está en la consulta formalmente.

B. Reglas

Cuando se trata de las enfermedades y los síntomas, estos son almacenados en forma de lista como reglas, dependiendo de su enfermedad haciendo uso de la función miembro, para identificar y evaluar si los síntomas dados por los pacientes concuerdan con alguna de las enfermedades disponibles, si concuerdan, se puede concluir que el paciente padece de esa enfermedad, si el paciente está interesado en las causas, síntomas y prevenciones seguirá preguntando en forma de conversación.

III. ESTRUCTURAS DE DATOS UTILIZADAS

A. Listas

Para la base de datos del programa, se usaron listas como Estructuras de Datos para enumerar los síntomas de las enfermedades al momento de declararlas. Así, cuando el cliente de DrLog envía una consulta desde la interfaz gráfica, el sistema experto puede buscar en las listas de la base de datos y encontrar los síntomas especificados por el usuario. De esta manera, la lista que contenga más síntomas que coincidan, es la lista que corresponde a la enfermedad consultada. Así mismo, se establece una lista para cada una de las causas, tratamientos y prevenciones de dichas enfermedades, las cuales también pueden ser consultadas mediante la interfaz gráfica. Estas estructuras funcionan de la misma manera que las reglas de las enfermedades, a pesar de estar escritas como hechos. También hay manejo de listas al utilizar el diccionario. En el caso del diccionario, el mismo recibe dos entradas, la frase dicha por el usuario, la cual corresponde a una lista compuesta por la palabra clave y variables anónimas, tal que, el programa pueda

identificar las mismas. El segundo parámetro corresponde a la respuesta del sistema experto, formada por una lista cuyos elementos son concatenados para conformar un solo string que constituye la respuesta.

IV. PROBLEMAS CONOCIDOS

A. *El paciente debe dar las 3 enfermedades en una sola línea*

Cuando el paciente está en la consulta, este debe incluir las 3 enfermedades necesarias para el diagnóstico, no puede incluir una enfermedad y después otra, ya que, DrLog busca en la lista de síntomas solo una vez, por lo tanto para dar la enfermedad con mayor coincidencias se deben comparar con los tres síntomas en una sola iteración.

B. *interfaz Java*

El problema con la interfaz de Java es que aunque conecta con la base de Datos de Prolog, a la hora de interactuar con con el usuario, cuando se le hace una solicitud, la interfaz se congela y la consola sigue corriendo y solo se puede interactuar desde la consola de Netbeans.

V. ACTIVIDADES REALIZADAS POR LOS ESTUDIANTES

A. *Bitácora general*

Día / Horas / Descripción

16 de marzo / 7:00-9:00 p.m. / Se realizó la división de labores, se leyó el enunciado y se definieron las enfermedades.
 19 de marzo / 9:00-12:00 p.m. / Se realizó la entrega de la investigación de las enfermedades, sus síntomas, causas, tratamientos y prevenciones. Se inicia la base del conocimiento.
 20 de marzo / 5:00-8:00 p.m. / Se continúa con la base del conocimiento, agregando síntomas, tratamientos y prevenciones de las enfermedades. Se inicia la investigación sobre gramáticas de libre contexto.
 21 de marzo / 5:00-11:00 p.m. / Se realizan pruebas y ejemplos de las gramáticas de libre contexto. Se finaliza la base del conocimiento y se escogen palabras claves.
 24 de marzo / 4:00-8:00 p.m. / Se experimentan con pruebas y errores de la notación BNF de las gramáticas de libre contexto, se decide intercambiar la notación BNF por la implementación de un diccionario basado en las gramáticas de cláusulas definidas (DCG), las cuales son de libre contexto.
 26 de marzo / 7:00-9:00 p.m. / Se inicia a reestructurar la base del conocimiento para el diccionario, se trabaja la nueva notación y se definen funciones de manejos de string para ello.
 27 de marzo / 2:30-7:30 p.m. / Se realizan pruebas con el diccionario que resultan exitosas, se inicia la implementación de la interfaz gráfica de Java y la unión del diccionario con la base del conocimiento.
 28 de marzo / 10:30 p.m.-3:00 a.m. / Se concluyó la integración del diccionario de gramáticas de cláusulas definidas con la base de conocimiento, se realiza el ligado de Java y Prolog, se codifican los query.
 29 de marzo / 4:00 p.m.-2:00 a.m. / Se realizan pruebas con la interfaz gráfica y el manejo de strings desde Java a

Prolog, se redefinen los query y el uso del diccionario. Se realizan intentos y soluciones de los bugs de la interfaz gráfica.

B. *Actividades individuales de Daniel Prieto*

Día / Horas / Descripción

16 de marzo / 7:00-9:00 p.m. / Se realizó la división de labores, se leyó el enunciado y se definieron las enfermedades.
 17 de marzo / 8:00-10:00 p.m. / Se investigan las enfermedades asignadas para el proyecto.
 18 de marzo / 5:00-6:30 p.m. / Se continúa con la investigación de enfermedades, se inicia la investigación sobre aspectos clave de la tarea.
 19 de marzo / 5:00-11:00 p.m. / Se entregan las enfermedades con sus respectivas características y se inicia su adición a la base del conocimiento mediante hechos y reglas.
 20 de marzo / 4:00-8:00 p.m. / Se termina la adición de las enfermedades y sus características en la base del conocimiento.
 21 de marzo / 9:00 a.m.-1:00 p.m. / Se inicia la investigación de las gramáticas de libre contexto y se realizan ejemplos básicos de la notación BNF.
 24 de marzo / 6:30-9:30 p.m. / Se descarta la notación BNF y se inicia a implementar un diccionario, el cual requiere manejo de strings en Prolog. Se realiza una investigación para las soluciones de la implementación del diccionario.
 26 de marzo / 10:00 p.m.-1:00 a.m. / Las enfermedades previamente agregadas a la base del conocimiento se actualizan con nueva sintaxis, se continúa investigando soluciones para la implementación del diccionario.
 27 de marzo / 2:30-7:30 p.m. / Se realizan pruebas con el diccionario que resultan exitosas, se inicia la implementación de la interfaz gráfica de Java y la unión del diccionario con la base del conocimiento.
 28 de marzo / 10:30 p.m.-3:00 a.m. / Se concluyó la integración del diccionario de gramáticas de cláusulas definidas con la base de conocimiento, se realiza el ligado de Java y Prolog, se codifican los query.
 29 de marzo / 4:00 p.m.-2:00 a.m. / Se realizan pruebas con la interfaz gráfica y el manejo de strings desde Java a Prolog, se redefinen los query y el uso del diccionario. Se realizan intentos y soluciones de los bugs de la interfaz gráfica.

C. *Actividades individuales de Karla Rivera*

16 de marzo 7 a 9 p.m.: Asignación de enfermedades por parte de cada uno. 17 de marzo 5 a 7 p.m.: Buscar enfermedades con las tablas respectivas de cada enfermedad. 18 de marzo 10 a 11:30 p.m.: Buscar información sobre BNF. 20 de marzo 3 a 6 p.m.: Buscar información y aplicación de pequeños ejemplos de BNF. 22 de marzo 1 a 3 p.m.: Ejemplos de BNF con sistema experto para implementar en la tarea. 24 de marzo 4 a 8 p.m.: Seguir intentando BNF con sistema experto, descartar la opción de BNF e implementar un diccionario lo más independiente posible. 26 de marzo 9:30 p.m. a 12 a.m.: Acomodar la base de datos de forma que se adapte al diccionario. 28 de marzo 10:30 p.m. a 3 a.m.:

Trabajando en errores, uso de readline, diccionario y función principal de drLog para hacer la serie de conversaciones. 29 de marzo 3:30 p.m. a 9 a.m: Solución de errores en prolog y base de datos, junto con pruebas de drLog en prolog.

D. Actividades individuales de Mariano Muñoz

16 de marzo, 7 a 9 pm: le solicito a mis compañeros que busquen 5 enfermedades incluyendo causas, tratamientos y síntomas.

17 de marzo, 5 a 9 pm: busco las enfermedades que me corresponden y completo el cuadro asignado en el Onedrive del equipo.

18 de marzo 9 a 12 pm: Me instruyo en la parte de Sistemas Expertos, para tener una idea general de como debería funcionar el sistema, el Dr Log.

19 de marzo 10 a 12 pm: realizamos una video llamada para comprobar la funcionalidad de Git y para complementar la información de enfermedades que buscamos individualmente.

22 de marzo 9 pm a 23 de marzo 3 am: busco información de como leer una sentencia en Prolog, encuentro un módulo llamado readline de código abierto, que en Java sería como un paquete, lo importo a Prolog y verifico su funcionamiento.

23, 24 y 25 de marzo: No puedo avanzar por el hecho de que tengo examen de Física 4 y lab de CE.

27 de marzo 9 pm a 28 marzo 3 am: comenzamos a unir las partes existentes para ir complementando nuestros códigos.

28 de marzo 9 pm a 29 marzo 3 am.:se realizan actividades varias como la creación de la documentación y revisión de las porciones de código. 29 a 30 de marzo, de las 9 pm a las 9 am: se termina la documentación y se terminan los detalles posibles de la interfaz de Java.

VI. PROBLEMAS ENCONTRADOS

A. Implementación de BNF

En este caso no se pudo hacer la Implementación de BNF con el sistema experto, por lo que se recurre a una solución alternativa que le llamamos diccionario, donde, a partir de frases base se complementa la conversación con el paciente y se le logra dar lo que el paciente necesita saber acerca de su enfermedad.

B. Problema del uso de los Query

En este caso los Query tuvieron una mala integración con la liga entre Prolog y Java, por lo que ejecutaba una excepción de Prolog al ser ejecutados. Sin embargo, esto se resolvió mediante un bloque try/catch para manejar la excepción. En el caso de la interfaz, los Query también tuvieron un gran problema de integración. Esto fue debido a que la interfaz gráfica trabajaba con el diccionario de términos basado en la notación DCG, y los Query trabajan con hechos y reglas. Esto se solucionó mediante el uso de los strings de la interfaz gráfica como parámetro, los cuales fueron utilizados para obtener la solución correcta.

VII. CONCLUSIONES Y RECOMENDACIONES

Para la parte de prolog, el sistema experto no funciona con BNF para la conversación y palabras clave, por lo que se hace uso de un diccionario que pueda hacer comandos básicos para hacer pruebas fundamentales con la base de datos.

Los enlaces entre lenguajes son de suma importancia, podemos asemejarlo a las simbiosis que realizan los seres vivos, que se complementan y se ayudan unos a otros para lograr una meta común, en esta ocasión pudimos observar la complementariedad de Prolog con Java para sistemas expertos, lo cual nos demuestra que podemos tener bases de Datos aisladas de nuestra interfaz lo que hace que podamos hacer uso de módulos de bases de Datos en la misma interfaz, cambiando las solicitudes a las bases de datos.

Se recomienda siempre tener en cuenta las amplias posibilidades del uso de mas de un lenguaje para la resolución de un problema, ya que un lenguaje nos puede proveer ventajas en interfaz(java) y otro ventajas en las bases de datos (prolog).

VIII. BIBLIOGRAFÍA

<https://personal.us.es/fsoler/papers/05capgram.pdf/>
<https://www.cs.us.es/~jalonso/cursos/ia2-00/temas/tema-05.pdf/>
<https://www.cs.us.es/~jalonso/cursos/d-pl-03/temas/tema-5.pdf/>
http://oa.upm.es/34929/1/TD_GARCIA_SERRANO_ANA.pdf/
<https://www.cs.us.es/~jalonso/cursos/pd-02/temas/tema-4.pdf>

IX. ANEXOS