

**TECNOLÓGICO DE ESTUDIOS SUPERIORES DE IXTAPALUCA ORGANISMO PUBLICO DESCENTRALIZADO DEL ESTADO DE MEXICO**

**Evaluación Segunda Oportunidad: Tercer Parcial**

**INGENIERÍA EN SISTEMAS COMPUTACIONALES**

**Programación lógica y funcional.**

**Profesor: Ebner Juárez Elías.**

**Grupo:1851**

**DEREK CAMILO VALENCIA SOSA**

**Instrucciones**

Desarrollar un sistema experto básico utilizando programación lógica (con Prolog o un entorno similar), que permita diagnosticar fallas comunes en un entorno tecnológico (redes, hardware o software) a partir de una base de conocimiento y reglas lógicas.

Colocar la liga de acceso del repositorio en github, para su revisión de documentación, presentación y Código

**Contexto**

Una empresa de soporte técnico desea automatizar el diagnóstico inicial de fallas que sus técnicos reciben por parte de los usuarios. Para ello, se requiere construir un sistema experto basado en reglas lógicas que reciba síntomas o condiciones reportadas

por el usuario y devuelva un diagnóstico probable.

**Fundamentos del Sistema Experto en Prolog**

Un sistema experto en Prolog se compone principalmente de dos partes:

Base de Conocimiento: Aquí es donde definimos los hechos (síntomas observados) y las reglas (condiciones que llevan a un diagnóstico).

Motor de Inferencia: En Prolog, este motor está integrado en el propio lenguaje a través de su mecanismo de unificación y backtracking. Cuando haces una consulta, Prolog usa este motor para recorrer las reglas y los hechos y llegar a una conclusión.

**Cómo Usar el Sistema (con SWI-Prolog)**

Guarda el código: Copia el código anterior y guárdalo en un archivo llamado diagnostico\_tecnologico.pl.

Abre SWI-Prolog: Inicia tu intérprete de SWI-Prolog (puedes descargarlo e instalarlo si no lo tienes).

Carga el archivo: En la consola de Prolog, escribe lo siguiente y presiona Enter:

consult('diagnostico\_tecnologico.pl').

Con ese comando hacemos la consulta a nuestro documento para poder visualizar o hacer la consulta debemos abrir nuestra terminal y escribimos cd y la dirección de donde creamos el archivo .pl en este caso queda asi

Cd C:\Users\roddr\Desktop\examen ebner2

Ahora bien, haremos las consultas pertinentes para observar y visualizar el funcionamiento del sistema ejecutando el siguiente comando

iniciar\_diagnostico.

**Responde las preguntas:** El sistema empezará a hacerte preguntas sobre los síntomas. Responde con s (sí) o n (no) seguido de un punto y Enter (ejemplo: s.

Conclusión: Un Paso Lógico hacia el Diagnóstico Inteligente

El desarrollo de este sistema experto básico en Prolog demuestra la potencia de la programación lógica para abordar problemas de diagnóstico en entornos tecnológicos. Al estructurar el conocimiento en forma de hechos (síntomas) y reglas lógicas (diagnósticos), hemos creado un motor de inferencia capaz de simular el razonamiento humano.

La facilidad con la que Prolog permite definir relaciones y buscar soluciones a través de la unificación y el backtracking lo convierte en una herramienta ideal para este tipo de aplicaciones. Aunque básico, el sistema ilustra cómo, a partir de la interacción con el usuario y una base de conocimiento bien definida, se pueden identificar fallas comunes en redes, hardware o software y ofrecer sugerencias de solución. Este ejercicio resalta la aplicabilidad de los principios de la inteligencia artificial y los sistemas expertos en la resolución práctica de problemas cotidianos, sentando las bases para sistemas más complejos y autónomos.

Posibles Mejoras Futuras

Expansión de la Base de Conocimiento: Añadir más síntomas, diagnósticos y soluciones para cubrir un espectro más amplio de problemas. Manejo de Incertidumbre: Incorporar factores de certeza o lógica difusa para asignar probabilidades a los diagnósticos cuando la información es incompleta o ambigua. Interfaz Gráfica de Usuario (GUI): Desarrollar una interfaz más amigable que reemplace la interacción por consola, mejorando la experiencia del usuario. Módulo de Explicación: Permitir que el sistema explique el "por qué" de un diagnóstico, mostrando las reglas y hechos que condujeron a la conclusión. Persistencia: Implementar un mecanismo para guardar y cargar bases de conocimiento más grandes o datos de sesiones anteriores.