

**Alumnos:**

Abraham Gael Herrera Quiñones

Eduardo Garcia Noriz

**Carrera:**

Ingeniería en sistemas computacionales

**Materia:**

Inteligencia artificial

**Maestro**

**Zuriel Dathan Mora Felix**

**Reporte de sistema experto de diagnostico de fallas electronicas**

**Unidad 3 tarea 4**

**Grupo:**

09:00-10:00

Sistema Experto para el Diagnóstico de Fallas Electrónicas

Introducción

Este documento presenta el desarrollo de un sistema experto diseñado para diagnosticar fallas en dispositivos electrónicos a partir de síntomas observables. El sistema utiliza una base de conocimiento con variables proposicionales y un conjunto de reglas de inferencia representadas en lógica proposicional. Su propósito es asistir a técnicos o estudiantes en la identificación rápida y eficiente de posibles problemas eléctricos o electrónicos.

El sistema fue desarrollado en el lenguaje de programación Python e implementa una interfaz gráfica utilizando Tkinter. Se incluyeron 17 síntomas distintos como variables proposicionales, y se definieron 12 reglas de inferencia. Cada regla consiste en una combinación lógica de condiciones que, al cumplirse, generan un diagnóstico específico. El sistema permite al usuario seleccionar los síntomas que observa en un dispositivo, y luego infiere posibles fallas basándose en la base de conocimiento.

El sistema está organizado dentro de una clase llamada SistemaExpertoElectronicaGUI, que contiene tanto la lógica de diagnóstico como la parte visual. Se utiliza una base de conocimiento formada por un diccionario que define los síntomas posibles y una lista de reglas que indican qué combinaciones de síntomas generan un diagnóstico.

1. **Componentes Principales**

* crear\_interfaz(): Se encarga de mostrar todos los síntomas disponibles como casillas seleccionables y de preparar los botones para ejecutar el diagnóstico o limpiar la selección.
* diagnosticar(): Toma los síntomas seleccionados por el usuario, revisa cuáles reglas se cumplen y genera los diagnósticos posibles.
* mostrar\_diagnosticos(): Muestra los resultados obtenidos de forma clara, indicando qué regla se activó, qué condiciones se cumplieron y cuál es la conclusión.
* limpiar(): Permite reiniciar la selección de síntomas para volver a usar el sistema.

1. **Funcionamiento del Motor de Diagnóstico**

El sistema recorre todas las reglas y evalúa si se cumplen según los síntomas que el usuario seleccionó. Soporta condiciones simples (como "A"), negaciones (por ejemplo "!B") y también condiciones con alternativas (como "E|F"). Si una regla se cumple, se agrega el diagnóstico correspondiente a los resultados.

1. **Interfaz de Usuario**

Se desarrolló una interfaz gráfica usando Tkinter. Esta interfaz permite seleccionar los síntomas fácilmente, sin necesidad de escribir nada. Además, se puede realizar más de un diagnóstico simplemente limpiando y repitiendo el proceso. Es intuitiva y pensada para ser utilizada por estudiantes o técnicos.

1. **Evaluación de Reglas Combinadas**

Las reglas pueden tener varias condiciones combinadas con operadores lógicos. El sistema puede manejarlas correctamente y mostrar siempre qué condiciones activaron el diagnóstico.

|  |  |  |
| --- | --- | --- |
| Regla | Condición necesaria | Diagnóstico que da |
| R1 | A y B | Revisar el cable de alimentación, enchufe o fuente externa |
| R2 | A y C y NO B | Posible falla en fuente interna o placa principal |
| R3 | D y C | Reemplazar fusible y verificar continuidad |
| R4 | D y (E o F) | Hay un corto circuito interno |
| R5 | G y H | Falla en la fuente conmutada (switching) |
| R6 | I | Sustituir capacitores y verificar circuito de regulación |
| R7 | J | Verificar pistas, soldaduras frías y conexiones internas |
| R8 | K | No energizar el equipo hasta reemplazar el componente afectado |
| R9 | L y M | Componente abierto o dañado internamente |
| R10 | N y O | Resistencia fuera de tolerancia, debe ser reemplazada |
| R11 | P | Transistor en corto, debe reemplazarse |
| R12 | Q | Verificar diodo zener, transistor regulador y capacitores de filtro |

# Visualización de interfaz del Sistema experto

Primer ejemplo es el de la regla R1 de A y B que son desde las variables proposicionales ya definidas.

Interfaz de usuario gráfica

El contenido generado por IA puede ser incorrecto.

Ejemplo 2 R2:

Interfaz de usuario gráfica, Aplicación

El contenido generado por IA puede ser incorrecto.

Ejemplo 3 R3:

Interfaz de usuario gráfica, Aplicación

El contenido generado por IA puede ser incorrecto.

Ejemplo 4 R4:

AQUI vemos que hay un detalle que es que si la condicion con solo cualquiera de los 2 sintomas se cumple, entra este mismo diagnostico. Sintoma E: “olor a quemado” y F: “componente visualmente dañado” cualquiera de estas 2 combinada con D fusible dañado sale el mismo diagnostico que es muy común.

Interfaz de usuario gráfica, Texto, Aplicación

El contenido generado por IA puede ser incorrecto.

Y asi sucesivamente con las demas reglas que se definieron el codigo se adjunto en archive diferente a esta por temas de longitud de codigo.

# Conclusión

El desarrollo de este sistema experto proporciona una herramienta útil para el diagnóstico preliminar de fallas electrónicas. Permite identificar problemas comunes a partir de síntomas observados de manera sencilla y eficiente. Además, su interfaz gráfica lo hace accesible a estudiantes y técnicos sin necesidad de conocimientos avanzados de programación. Este proyecto demuestra cómo la inteligencia artificial puede aplicarse en el ámbito técnico para facilitar la toma de decisiones.