



**TECNOLÓGICO  
NACIONAL DE MÉXICO**



## Evidencia-03 Sistema Experto

### **Alumnos:**

- Lara Beltrán Jorge Alberto
- Pacheco Pérez Diego

### **Docente:**

Zuriel Dathan Mora Félix

### **Materia:**

Inteligencia Artificial

En el documento Excel anexado al proyecto se encuentran la definición de las proposiciones, así como las tablas donde se comprueban las reglas, vienen columnas donde están los símbolos necesarios de cada enfermedad, así como otra columna donde viene el numero de verdaderos que estos se refieren a cuantos síntomas son verdaderos y luego el porcentaje de certeza (el que se muestra en la ejecución) el cual se calcula con la siguiente formula:

$$certeza(D) = \frac{\#(condiciones\ verdaderas\ en\ el\ antecedente\ de\ D)}{\#(condiciones\ totales\ en\ el\ antecedente\ de\ D)} \times 100$$

Ejemplo (COVID-19): antecedente tiene 4 literales (F, T, AN, FTG).

- Si solo F y T son verdaderas  $\rightarrow$  certeza =  $2 / 4 = 50\%$ .

Para ejecutar el proyecto simplemente ejecute el archivo main.py

**Archivo:** main.py

Este archivo actúa como **punto de entrada del sistema experto**.

Crea la ventana principal de la aplicación usando **Tkinter** y carga la interfaz gráfica definida en interfaz.py.

**Funcionamiento:**

1. Se inicializa una ventana raíz (`root = tk.Tk()`).
2. Se crea una instancia de `InterfazExperto`, que define toda la interfaz gráfica.
3. Se ejecuta el bucle principal de Tkinter (`root.mainloop()`), que mantiene activa la ventana.

**Archivo:** explicacion.py

Este módulo implementa la clase `Explicacion`, responsable de **registrar y generar trazas del razonamiento del sistema**.

**Clase:** `Explicacion`

- **Atributo trazas:** lista que almacena los pasos de razonamiento (reglas activadas, síntomas y conclusiones).
- **registrar\_regla(condiciones, conclusion):** guarda cada regla que se cumplió durante el diagnóstico.
- **generar\_explicacion():** construye un texto explicativo legible para el usuario, mostrando los síntomas detectados y las enfermedades inferidas.

### Ejemplo de salida:

1. Se cumplieron los síntomas: tos, fiebre, fatiga  
-> Hay probabilidad de: Covid-19

### Archivo: base\_conocimiento.py

Contiene la **base de conocimiento del sistema experto**.

1. **descripciones:** diccionario que traduce los nombres técnicos de los síntomas a frases legibles por el usuario.
2. "tos\_productiva": "tos con flema"
3. **reglas:** lista de tuplas que representan las **reglas de inferencia** del sistema. Cada regla tiene una lista de condiciones (síntomas) y una conclusión (diagnóstico probable).
4. (["tos", "tos\_productiva", "duracion\_sintomas\_menor\_3\_semanas"], "bronquitis aguda")

Estas reglas permiten que el motor de inferencia compare los síntomas del paciente con los patrones de enfermedades respiratorias.

### Archivo: motor\_inferencia.py

Define la clase MotorInferencia, que ejecuta la **lógica de razonamiento** del sistema experto.

### Flujo del razonamiento:

1. Recibe los **hechos** (síntomas ingresados por el usuario).
2. Recorre las reglas de la base de conocimiento.
3. Calcula una **probabilidad** según cuántos síntomas coinciden con cada enfermedad.
4. Registra las coincidencias en el objeto Explicacion para justificar los resultados.
5. Devuelve una lista de posibles diagnósticos ordenados por probabilidad.

### Ejemplo:

Si una regla tiene 4 síntomas y el usuario presenta 3,  
la probabilidad será  $(3/4) \cdot 100 = 75\%$ .

## Archivo: interfaz.py

Este archivo implementa la **interfaz gráfica del usuario (GUI)** mediante Tkinter.

### Clase principal: InterfazExperto

- Crea un formulario visual donde el usuario puede ingresar:
  - Edad, sexo y saturación de oxígeno.
  - Síntomas seleccionados mediante casillas de verificación.
- Botones principales:
  - **“Diagnosticar”** → ejecuta el motor de inferencia y muestra los resultados.
  - **“Ver explicación”** → muestra en una ventana aparte el razonamiento detallado.

### Funciones principales:

- diagnosticar():
  - Obtiene los síntomas seleccionados.
  - Evalúa edad y saturación para añadir hechos adicionales (“edad”, “saturacion\_baja”).
  - Llama a MotorInferencia.diagnosticar\_con\_probabilidad().
  - Muestra el diagnóstico y las recomendaciones médicas asociadas.
- mostrar\_explicacion():
  - Muestra el razonamiento generado por la clase Explicacion.

### Ejemplo de salida del diagnóstico:

Sexo: Femenino | Edad: 65 | Saturación: 90%

Diagnóstico(s) presuntivo(s):

- EPOC — Certeza: 66.7%
  - ↳ Recomendación: Consultar con un neumólogo. Posible uso de broncodilatadores y control periódico de oxígeno.