



## TECNOLOGICO DE ESTUDIOS SUPERIORES IXTAPALUCA

Alumno: Alfredo Alejandro Lopez Montejo

Profesor: Ebner Juárez Elías

Materia: Programación lógica y funcional

Semestre: 9

## PRACTICA DE SISTEMA-EXPERTO-DIAGNOSTICO

Como primer paso creamos una carpeta en la raíz de la computadora con el nombre de sistema-experto-diagnostico.

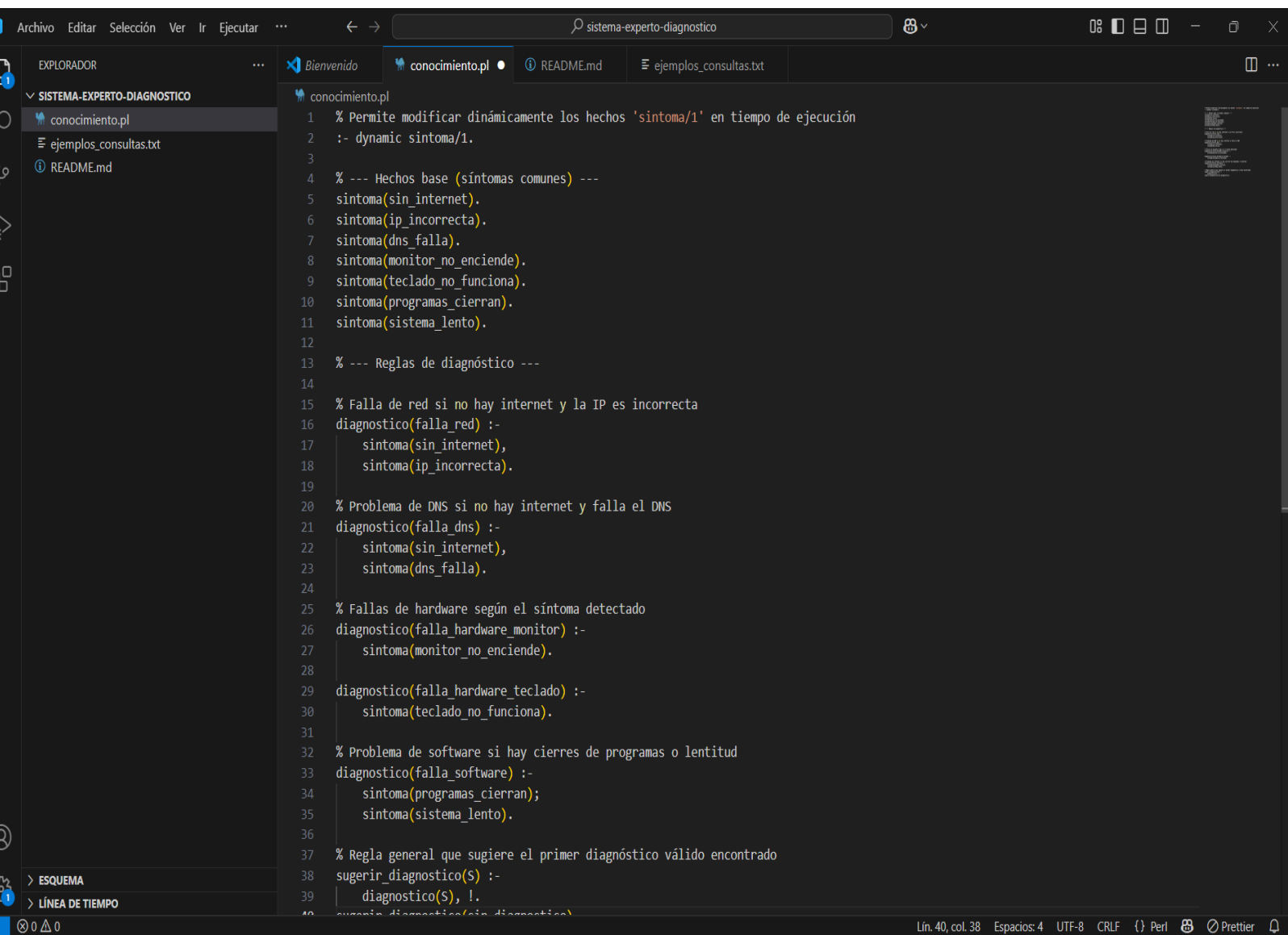
Después de haberla creado, nos vamos a visual studio code y le damos en abrir carpeta y buscamos la carpeta echa y luego creamos 3 archivos adentro.

El primer archivo debes ponerle el nombre de conocimiento.pl

El segundo archivo debe tener el nombre de README.md

El tercero archivo debe tener el nombre de ejemplos\_consultas.txt

Luego abrimos el archivo de conocimiento debes empezar hacer el código como se muestra en la imagen:



```
conocimiento.pl
1 % Permite modificar dinámicamente los hechos 'sintoma/1' en tiempo de ejecución
2 :- dynamic sintoma/1.
3
4 % --- Hechos base (síntomas comunes) ---
5 sintoma(sin_internet).
6 sintoma(ip_incorrecta).
7 sintoma(dns_falla).
8 sintoma(monitor_no_enciende).
9 sintoma(teclado_no_funciona).
10 sintoma(programas_cierran).
11 sintoma(sistema_lento).
12
13 % --- Reglas de diagnóstico ---
14
15 % Falla de red si no hay internet y la IP es incorrecta
16 diagnostico(falla_red) :-
17     sintoma(sin_internet),
18     sintoma(ip_incorrecta).
19
20 % Problema de DNS si no hay internet y falla el DNS
21 diagnostico(falla_dns) :-
22     sintoma(sin_internet),
23     sintoma(dns_falla).
24
25 % Fallas de hardware según el síntoma detectado
26 diagnostico(falla_hardware_monitor) :-
27     sintoma(monitor_no_enciende).
28
29 diagnostico(falla_hardware_teclado) :-
30     sintoma(teclado_no_funciona).
31
32 % Problema de software si hay cierres de programas o lentitud
33 diagnostico(falla_software) :-
34     sintoma(programas_cierran);
35     sintoma(sistema_lento).
36
37 % Regla general que sugiere el primer diagnóstico válido encontrado
38 sugerir_diagnostico(S) :-
39     diagnostico(S), !.
40     sugerir_diagnostico(_).
41
42 % Fin del conocimiento base
```

Después en el archivo de README colocamos lo que quiera que tenga o visualicemos la interfaz del GitHub.

The image shows two side-by-side views of the same README file. The top view is a screenshot of the Visual Studio Code editor. The bottom view is a screenshot of the GitHub web interface.

**VS Code View:**

- File Explorer (left): Shows the project structure with files `conocimiento.pl`, `ejemplos_consultas.txt`, and `README.md`.
- Editor (center): Displays the content of `README.md` with line numbers 1 through 33. The content includes a title, a description, sections for 'Características', 'Requisitos', 'Archivos del Proyecto', and 'Instrucciones de Uso'.
- Terminal/Output (bottom): Shows the status bar with 'Lin. 26, col. 1' and other editor settings.

**GitHub View:**

- File Explorer (left): Shows the repository structure with files `README.md`, `conocimiento.pl`, and `ejemplos_consultas.txt`.
- Preview (center): Displays the rendered HTML version of the README. It features a title with a brain icon, a description, and sections for 'Características', 'Requisitos', 'Archivos del Proyecto', and 'Instrucciones de Uso'.

**README Content:**

```
1 # 🧠 Sistema Experto en Prolog - Diagnóstico de Fallas Técnicas
2
3 Este sistema experto fue desarrollado para ayudar a diagnosticar fallas comunes en un entorno tecnológico (redes, hardware y software) usando lógica declarativa con Prolog. El objetivo es automatizar el diagnóstico inicial que realiza un técnico al recibir reportes de usuarios.
4
5 ---
6
7 ## 🚀 Características
8
9 - Diagnóstico básico a partir de síntomas.
10 - Utiliza una base de conocimientos y reglas lógicas en Prolog.
11 - Enfocado en tres áreas: Redes, Hardware y Software.
12
13 ---
14
15 ## ⚙️ Requisitos
16
17 - [SWI-Prolog](https://www.swi-prolog.org/Download.html)
18
19 ## 📁 Archivos del Proyecto
20
21 - `conocimiento.pl`: Archivo principal con hechos y reglas.
22 - `README.md`: Documentación del proyecto.
23 - `ejemplos_consultas.txt` (opcional): Ejemplos de consultas en Prolog.
24
25 ---
26
27 ## 📖 Instrucciones de Uso
28
29 ### 1. Clonar el repositorio (o descargarlo)
30
31 ```bash
32 git clone https://github.com/TU_USUARIO/sistema-experto-diagnostico.git
33 cd sistema-experto-diagnostico
```

**GitHub Preview:**

## 🧠 Sistema Experto en Prolog - Diagnóstico de Fallas Técnicas

Este sistema experto fue desarrollado para ayudar a diagnosticar fallas comunes en un entorno tecnológico (redes, hardware y software) usando lógica declarativa con **Prolog**. El objetivo es automatizar el diagnóstico inicial que realiza un técnico al recibir reportes de usuarios.

### 🚀 Características

- Diagnóstico básico a partir de síntomas.
- Utiliza una base de conocimientos y reglas lógicas en Prolog.
- Enfocado en tres áreas: Redes, Hardware y Software.

### ⚙️ Requisitos

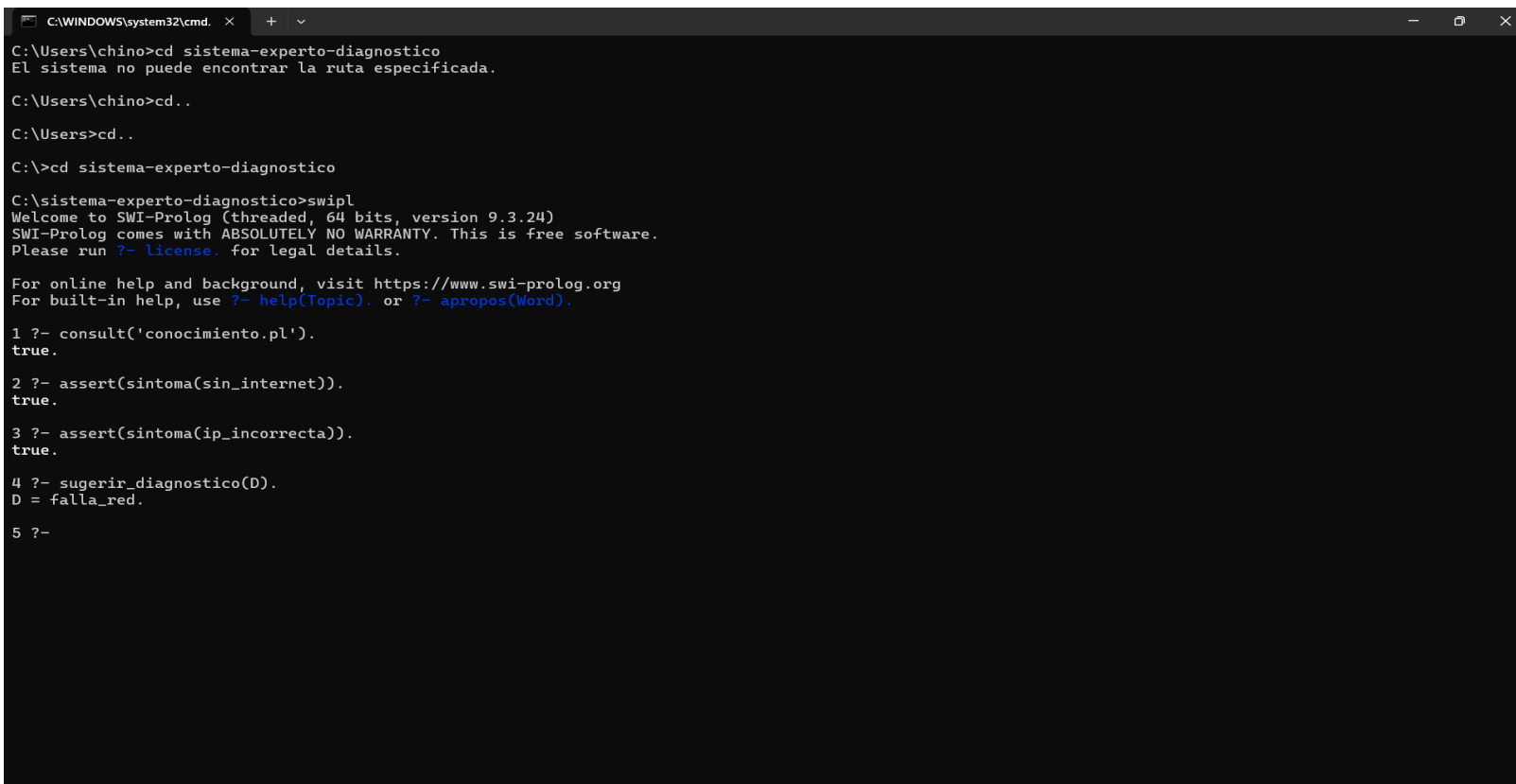
- [SWI-Prolog](https://www.swi-prolog.org/Download.html)

### 📁 Archivos del Proyecto

- `conocimiento.pl`: Archivo principal con hechos y reglas.
- `README.md`: Documentación del proyecto.
- `ejemplos_consultas.txt` (opcional): Ejemplos de consultas en Prolog.

### 📖 Instrucciones de Uso

Y por último debemos abrir el CMD y entrar a la carpeta de la raíz llamada sistema-experto-diagnóstico y después abrir el prolog con el comando swipl, ya abierto el prolog colocamos las consultas como se muestra en la imagen y ya terminamos:



```
C:\WINDOWS\system32\cmd. x + v
C:\Users\chino>cd sistema-experto-diagnostico
El sistema no puede encontrar la ruta especificada.

C:\Users\chino>cd..

C:\Users>cd..

C:\>cd sistema-experto-diagnostico

C:\sistema-experto-diagnostico>swipl
Welcome to SWI-Prolog (threaded, 64 bits, version 9.3.24)
SWI-Prolog comes with ABSOLUTELY NO WARRANTY. This is free software.
Please run ?- license. for legal details.

For online help and background, visit https://www.swi-prolog.org
For built-in help, use ?- help(Topic). or ?- apropos(Word).

1 ?- consult('conocimiento.pl').
true.

2 ?- assert(sintoma(sin_internet)).
true.

3 ?- assert(sintoma(ip_incorrecta)).
true.

4 ?- sugerir_diagnostico(D).
D = falla_red.

5 ?-
```