



**TECNOLÓGICO  
NACIONAL DE MÉXICO**



**Instituto Tecnológico de Culiacán**

**Carrera:** Ingeniería en Sistemas Computacionales

**Materia:** Inteligencia Artificial

**Profesor:** Zuriel Dathan Mora Félix

**Tarea: Sistema Experto: Orienciación Vocacional**

**Grupo:**

11:00 AM – 12:00 PM

**Equipo:**

García Pérez José Ángel

Verdugo Bermúdez Sebastián

# Sistema Experto: Orientación Vocacional

## Introducción

Este sistema experto implementado en Prolog es una herramienta de orientación vocacional diseñada para ayudar a los estudiantes a identificar carreras profesionales que se ajusten a sus intereses, habilidades y personalidad. El programa utiliza un enfoque basado en reglas para analizar las respuestas del usuario y sugerir las opciones académicas más adecuadas.

### Características principales:

**Amplia cobertura de áreas profesionales:** El sistema cubre 10 grandes áreas vocacionales con más de 50 carreras específicas.

**Sistema de preguntas adaptativo:** Preguntas dinámicas basadas en respuestas previas.

**Mecanismo de inferencia:** Utiliza lógica de predicados para determinar carreras coincidentes.

**Interfaz conversacional:** Interacción mediante preguntas/respuestas simples ("si"/"no").

**Integración con Python:** Puede ser ejecutado desde Python usando la biblioteca PySWIP.

## Funcionamiento con PySWIP (Python-Prolog Interface)

El sistema puede ser invocado desde Python mediante el siguiente código:

```
from pyswip import Prolog

# Crear la instancia de Prolog
prolog = Prolog()

# Cargar el archivo Prolog
prolog.consult("OrientadorVocacional.pl")

# Ejecutar la consulta principal
resultados = list(prolog.query("orientar"))

# Mostrar resultados
for resultado in resultados:
    print("Carrera recomendada:", resultado)
```

## **Código Prolog**

**orientar :-**

```
limpiar_respuestas,  
writeln("Bienvenido al sistema experto de orientacion vocacional"),  
predecir(Carrera),  
write('La carrera que mas se ajusta a tu perfil es: '),  
write(Carrera), write('.'), nl, !.
```

El predicado **orientar** es la regla principal del sistema experto que coordina el proceso de recomendación vocacional: primero limpia cualquier dato previo con **limpiar\_respuestas**, luego muestra un mensaje de bienvenida y mediante **predecir(Carrera)** realiza el cuestionario interactivo que analiza los intereses y aptitudes del usuario para determinar la carrera más adecuada, la cual finalmente muestra con un mensaje formateado ("La carrera que más se ajusta..."), donde el operador de corte (!) asegura que el sistema devuelva solo la primera y más relevante recomendación encontrada sin buscar alternativas adicionales.

### **% Predicciones posibles**

#### **% Carreras para Ciencias de la Salud**

```
predecir(medicina) :- medicina.  
predecir(enfermeria) :- enfermeria.  
predecir(psicologia) :- psicologia.  
predecir(nutricion) :- nutricion.
```

#### **% Carreras para Ingeniería y Tecnología**

```
predecir(ingenieria_civil) :- ingenieria_civil.  
predecir(ingenieria_mecanica) :- ingenieria_mecanica.  
predecir(ingenieria_industrial) :- ingenieria_industrial.  
predecir(ingenieria_electronica) :- ingenieria_electronica.  
predecir(ingenieria_mecatronica) :- ingenieria_mecatronica.  
predecir(ingenieria_sistemas) :- ingenieria_sistemas.
```

Este fragmento define las posibles carreras que el sistema puede recomendar, agrupadas en dos áreas principales: Ciencias de la Salud (medicina, enfermería, psicología, nutrición) e Ingeniería y Tecnología (civil, mecánica, industrial, electrónica, mecatrónica, sistemas). Cada regla predecir(X) vincula una carrera con su respectivo predicado de especialización (como medicina o ingenieria\_civil), que luego se evalúa mediante reglas específicas que determinan si el perfil del usuario coincide con esa opción. Es la base del sistema de recomendación vocacional.

## **% Reglas de cada carrera (Ciencias de la Salud)**

### **medicina :-**

**pertenece\_a\_salud,**  
**verificar(diagnosticar\_enfermedades),**  
**verificar(atender\_pacientes),**  
**verificar(salvar\_vidas).**

### **enfermeria :-**

**pertenece\_a\_salud,**  
**verificar(cuidar\_pacientes),**  
**verificar(apoyar\_tratamientos),**  
**verificar(contacto\_con\_personas).**

### **psicologia :-**

**pertenece\_a\_salud,**  
**verificar(interés\_mente\_humana),**  
**verificar(ayudar\_problemas\_emocionales).**

Este código establece los criterios para recomendar carreras de Ciencias de la Salud donde cada profesión tiene requisitos específicos: medicina exige aptitud para diagnosticar enfermedades atender pacientes y salvar vidas, enfermería requiere vocación de cuidado apoyo en tratamientos y contacto humano mientras psicología busca interés por comprender la mente humana y ayudar en problemas emocionales, todas comparten el prerrequisito de pertenecer al área de salud y cada una se activa cuando el usuario cumple con los atributos característicos de esa carrera mediante el sistema de verificación de pregunta

**% Área Ciencias de la Salud**

**pertenece\_a\_salud :-**

**verificar(ayudar\_salud), !.**

**pertenece\_a\_salud :-**

**verificar(interés\_cuerpo),**

**verificar(interés\_medicina),**

**verificar(interés\_bienestar), !.**

**pertenece\_a\_salud :-**

**verificar(empático),**

**verificar(observador),**

**verificar(responsable).**

**% Área Ingeniería y Tecnología**

**pertenece\_a\_ing\_tec :-**

**verificar(resolver\_problemas\_técnicos), !.**

**pertenece\_a\_ing\_tec :-**

**verificar(mejorar\_procesos), !.**

Este código define las condiciones para identificar si un usuario pertenece al área de Ciencias de la Salud o Ingeniería y Tecnología mediante un sistema de reglas jerárquicas con cortes (!) que detienen la búsqueda al encontrar un criterio válido: para salud verifica primero la disposición a ayudar (ayudar\_salud), luego intereses específicos (cuerpo, medicina, bienestar) y finalmente cualidades personales (empático, observador, responsable); para ingeniería prioriza habilidades de resolución de problemas técnicos y mejora de procesos, mostrando cómo el sistema clasifica al usuario en áreas amplias antes de evaluar carreras específicas.

**% Verificación con memorización**

**verificar(Característica) :-**

**si(Característica), !.**

**verificar(Característica) :-**

**no(Característica), !, fail.**

**verificar(Característica) :-**

**preguntar(Característica).**

Este código implementa un sistema de verificación con memoria que evalúa características del usuario en tres pasos: primero consulta si ya se registró una respuesta afirmativa (si) y acepta la característica inmediatamente con corte (!), luego verifica si existe una respuesta negativa (no) que hace fallar la regla (fail), y finalmente si no hay datos almacenados activa el predicado preguntar para consultar interactivamente al usuario y almacenar su respuesta, optimizando así el proceso al evitar repreguntas mediante memorización de respuestas previas.

**% Preguntar al usuario (solo acepta "si" o "no")**

**preguntar(Pregunta) :-**

**pregunta\_texto(Pregunta, Texto),  
repetir\_pregunta(Pregunta, Texto).**

**% Repite la pregunta hasta que se reciba "si" o "no"**

**repetir\_pregunta(Pregunta, Texto) :-**

**write(Texto), write(' (si/no): '),  
read\_line\_to\_string(user\_input, Resp),  
nl,  
(Resp = "si" ->  
assertz(si(Pregunta))  
; Resp = "no" ->  
assertz(no(Pregunta)), fail  
; writeln("Por favor, responde solo con 'si' o 'no'."),  
repetir\_pregunta(Pregunta, Texto)  
).**

Este código maneja la interacción con el usuario mediante un sistema robusto de preguntas y respuestas: primero obtiene el texto de la pregunta a mostrar (pregunta\_texto), luego inicia un bucle (repetir\_pregunta) que persiste hasta recibir una respuesta válida ("si" o "no"), almacenando las respuestas en la base de conocimientos (con assertz) como hechos si o no; si el usuario ingresa una respuesta inválida, muestra un mensaje de error y repite la pregunta, garantizando así que solo se procesen respuestas válidas mientras construye dinámicamente el perfil del usuario para la orientación vocacional.

## **% Preguntas para area Ciencias de la Salud**

**pregunta\_texto(ayudar\_salud, "Te interesa ayudar a las personas a mejorar su salud?").**

**pregunta\_texto(interés\_cuerpo, "Te interesa el cuerpo humano?").**

**pregunta\_texto(interés\_medicina, "Te interesa la medicina?").**

**pregunta\_texto(interés\_bienestar, "Te interesa el bienestar?").**

**pregunta\_texto(empático, "Eres empático?").**

Este código define el mapeo entre identificadores lógicos de preguntas (como `ayudar_salud`) y su formulación legible para el usuario en español, actuando como una base de datos de preguntas para el área de Ciencias de la Salud: cada hecho `pregunta_texto` relaciona un átomo simbólico (usado internamente por el sistema) con su correspondiente pregunta textual que se mostrará al usuario ("¿Te interesa ayudar...?"), permitiendo separar la lógica del programa de los textos interactivos y facilitando la traducción o modificación de las preguntas sin alterar el funcionamiento del sistema experto.

## **% Limpiar respuestas anteriores**

**limpiar\_respuestas :-**

**retract(si(\_)), fail.**

**limpiar\_respuestas :-**

**retract(no(\_)), fail.**

**limpiar\_respuestas.**

Este código define el predicado `limpiar_respuestas` que elimina todas las respuestas almacenadas anteriormente en la base de datos dinámica del sistema. Funciona en tres partes: primero intenta retirar (`retract`) todos los hechos `si` (usando `fail` para forzar el retroceso y continuar buscando más coincidencias), luego hace lo mismo con los hechos `no`, y finalmente la cláusula vacía `limpiar_respuestas`. asegura que el predicado siempre tenga éxito después de limpiar todas las respuestas, preparando así el sistema para una nueva sesión de orientación vocacional con un estado limpio.

# Funcionamiento del Programa

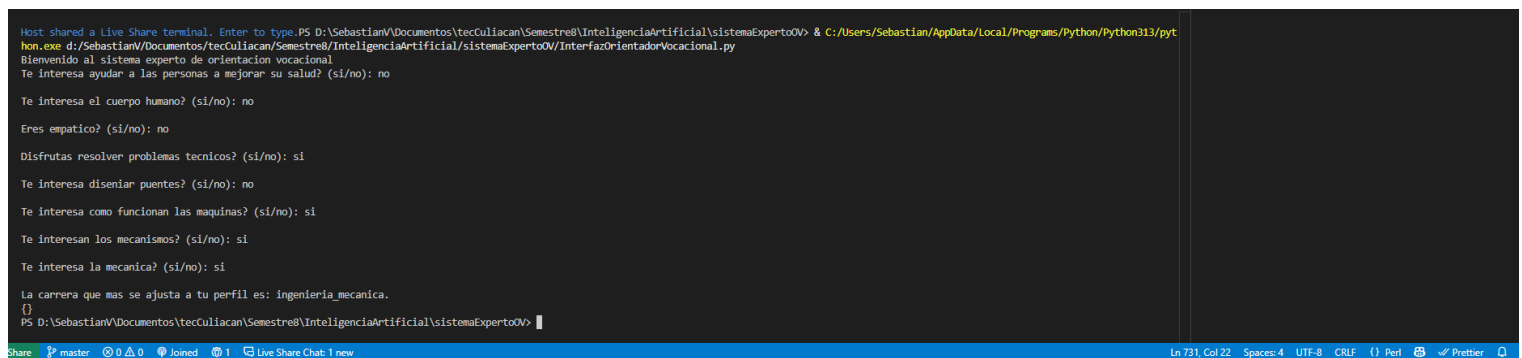
## Ejemplo 1



```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS COMMENTS
PS D:\Sebastian\Documentos\tecCuliacan\Semestre8\InteligenciaArtificial\sistemaExpertoOv> & C:/Users/Sebastian/AppData/Local/Programs/Python/Python313/python.exe d:/Sebastian/Documentos/tecCuliacan/Semestre8/InteligenciaArtificial/sistemaExpertoOv/InterfazOrientadorVocacional.py
Bienvenido al sistema experto de orientacion vocacional
¿Te interesa ayudar a las personas a mejorar su salud? (si/no): si
¿Quieres diagnosticar enfermedades? (si/no): si
¿Quieres atender pacientes? (si/no): si
¿Quieres salvar vidas? (si/no): si
La carrera que mas se ajusta a tu perfil es: medicina.
{}
PS D:\Sebastian\Documentos\tecCuliacan\Semestre8\InteligenciaArtificial\sistemaExpertoOv>
```

En esta imagen realizamos un primer ejemplo en donde respondimos a las preguntas del sistema experto que nos gusta ayudar a las personas en mejorar su salud, así como diagnosticar enfermedades, atender pacientes y salvar vidas como estamos muy enfocados en esta área nos arroja que de acuerdo a nuestras respuestas la carrera ideal de acuerdo a nuestro perfil es la “Medicina”.

## Ejemplo 2



```
Host shared a Live Share terminal. Enter to type.PS D:\Sebastian\Documentos\tecCuliacan\Semestre8\InteligenciaArtificial\sistemaExpertoOv> & C:/Users/Sebastian/AppData/Local/Programs/Python/Python313/python.exe d:/Sebastian/Documentos/tecCuliacan/Semestre8/InteligenciaArtificial/sistemaExpertoOv/InterfazOrientadorVocacional.py
Bienvenido al sistema experto de orientacion vocacional
Te interesa ayudar a las personas a mejorar su salud? (si/no): no
Te interesa el cuerpo humano? (si/no): no
Eres empático? (si/no): no
Disfrutas resolver problemas técnicos? (si/no): si
Te interesa diseñar puentes? (si/no): no
Te interesa como funcionan las máquinas? (si/no): si
Te interesan los mecanismos? (si/no): si
Te interesa la mecánica? (si/no): si
La carrera que mas se ajusta a tu perfil es: ingeniería_mecánica.
{}
PS D:\Sebastian\Documentos\tecCuliacan\Semestre8\InteligenciaArtificial\sistemaExpertoOv>
```

En esta imagen hicimos otro ejemplo para determinar una vocación, dimos respuesta a que no nos interesaba nada relacionado con la salud, pero si nos gusta resolver problemas técnicos sin embargo no nos interesa diseñar puentes, pero si el funcionamiento de las maquinas, los mecanismos y la mecánica en conclusión nos arroja que la carrera que mas se ajusta a nuestro perfil de acuerdo a nuestra respuesta fue la “Ingeniería Mecánica”.



### Ejemplo 3



```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL COMMENTS

Te interesa como funcionan las máquinas? (si/no): no
Te gusta trabajar con numeros? (si/no): no
Te interesa como funciona el dinero? (si/no): no
Te gusta organizar? (si/no): no
Te gusta mejorar procesos? (si/no): no
Te gusta mejorar procesos? (si/no): no
Te interesa como funcionan las máquinas? (si/no): no
Te gusta trabajar con numeros? (si/no): no
Te interesa como funciona el dinero? (si/no): no
Te gusta organizar? (si/no): no
eres analítico? (si/no): no
Te interesa entender la sociedad? (si/no): si
Te interesa entender a las personas? (si/no): si
Te interesa la historia? (si/no): si
Te interesa la justicia? (si/no): si
Te interesan las leyes? (si/no): si
Te interesa defender causas? (si/no): si

La carrera que mas se ajusta a tu perfil es: derecho.
```

En este ultimo ejemplo decidimos responder que no nos interesaba nada relacionado a la salud, tampoco a la ingeniería, tecnología o las finanzas ya que nos pregunta si somos una persona analítica, sin embargo, si nos gusta entender a la sociedad, entender a las personas, la historia, la justicia, las leyes y defender causas por ello nuestro perfil que mas se asemeja es la carrera de “Derecho”