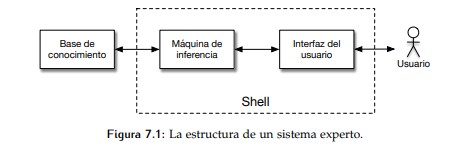
**Proyecto Final: Sistema Experto Personalizado en Prolog y Python**

**Condiciones generales del proyecto final**

* Grupos de máximo 2 personas.
* Fecha y hora de entrega: 22 de mayo de 2025, 6:30 pm.
* Fecha de sustentación: 27 de mayo de 2025, 6:30 pm (30% de la nota del taller).
* Formato de entrega: Los archivos .pl y .py deben ser comprimidos en archivo .zip. El nombre del archivo .zip será el nombre y apellido de cada integrante. Ejemplo SantiagoOrozco-LuisLópez.zip. Se debe incluir un documento explicando como se ejecuta la aplicación.

**Introducción**

En inteligencia artificial, un sistema experto es un sistema computacional que emula la capacidad de tomar decisiones de un humano experto. Un sistema experto está compuesto de una base de conocimiento y un motor de inferencia, el cual recibe las características de una entidad o un problema. Comparando el conocimiento previo existente en la base de conocimientos con las características de la entidad es posible identificarla y, por consecuencia, producir un diagnóstico o deducción. En el siguiente enlace podrá encontrar un ejemplo de sistema experto: https://swish.swi-prolog.org/p/PL2019-I.swinb



Fuente: https://www.uv.mx/personal/aguerra/files/2019/04/rc-notas-07.pdf.

**Descripción General:**

Este proyecto tiene como objetivo el desarrollo de un **sistema experto** utilizando el lenguaje de programación **Prolog y Python (pyswip)**, que esté diseñado para abordar un tema cotidiano no se podrá repetir con algún tema ya presentado el semestre pasado, ya que es personal y obedecerá a los interés particulares de cada estudiante. El sistema experto deberá ser capaz de **brindar recomendaciones, resolver problemas o proporcionar asistencia** en base a un conjunto de reglas y hechos que reflejen la experiencia o el conocimiento específico sobre el tema elegido por el estudiante.

**Temática:**

Cada estudiante debe elegir un tema **personal y específico** que no sea general ni relacionado con temáticas de amplio conocimiento público (por ejemplo, no puede ser sobre temas como "salud", "educación" o "cultura general"). El tema debe ser algo que el estudiante pueda desarrollar a partir de su propio conocimiento o experiencia (¿Qué temas te gustan leer o aprender? ¿En qué eres experto?). Algunos ejemplos de temas posibles son (estos ejemplos ya fueron utilizados el semestre pasado):

* **Recomendaciones para rutinas de deportes**: Asesoramiento sobre cómo programar las rutinas del gimnasio, entrenar futbol, practicar ciclismo de ruta, etc.
* **Recomendaciones de recetas o buenos hábitos de alimentación**: Un sistema que sugiera recetas de cocina basado en los ingredientes disponibles en la casa o hábitos de alimentación de acuerdo al objetivo a lograr (ganar/perder peso, etc).
* **Planificación de viajes personales**: Recomendaciones sobre itinerarios de viaje en fincas de Caldas basados en preferencias personales (por ejemplo, distancia, presupuesto).

El sistema deberá ser capaz de hacer inferencias sobre la situación del usuario (basándose en hechos proporcionados como datos de entrada), y tomar decisiones o sugerir soluciones según las reglas definidas.

**Componentes del Sistema Experto:**

A continuación se describen los componentes, no obstante, se recomienda realizar tener los datos (base de conocimiento) amplia, con un motor de inferencia , interfaz de usuario y módulo de recomendaciones adecuado acorde a la complejidad de un proyecto final.

1. **Base de Conocimiento**:
   * Hechos: Datos que representan el conocimiento específico sobre el tema.
   * Reglas: Instrucciones o relaciones lógicas que permiten inferir nuevos hechos o tomar decisiones.
2. **Motor de Inferencia**:
   * El motor de inferencia se encargará de hacer deducciones a partir de los hechos y reglas proporcionados por el usuario. El sistema será capaz de proporcionar respuestas o recomendaciones basadas en la entrada del usuario.
3. **Interfaz de Usuario**:
   * El sistema debe ser interactivo, permitiendo al usuario ingresar datos y obtener respuestas.
   * La interfaz puede ser simple (línea de comandos) o un interfaz gráfica sencilla, si se prefiere.
4. **Módulo de Recomendación o Asesoría**:
   * Dependiendo del tema elegido, el sistema deberá ser capaz de proporcionar recomendaciones o guiar al usuario en la toma de decisiones, basándose en el conocimiento almacenado en la base de conocimiento.
5. **Sistema de Explicaciones**:
   * El sistema debe ser capaz de explicar sus decisiones o recomendaciones, proporcionando una justificación basada en las reglas y hechos utilizados.

**Rubrica de Evaluación:**

|  |  |
| --- | --- |
| **Criterio** | **Descripción** |
| **Correcta**  **implementación en**  **Prolog** | El código debe estar correctamente escrito en Prolog, utilizando hechos y reglas lógicas que permitan inferir conclusiones. |
| **Relevancia del tema elegido** | El tema debe ser personal, relevante y específico para el estudiante, evitando temas genéricos. |
| **Complejidad de las reglas y hechos** | La cantidad de reglas y hechos, su interrelación y la capacidad de hacer inferencias complejas. |
| **Interfaz de usuario** | La facilidad de uso y la interacción del usuario con el sistema. Se puede optar por una interfaz de línea de comandos o gráfica. |
| **Explicación de las decisiones** | El sistema debe ser capaz de explicar sus decisiones o recomendaciones de manera clara. |
| **Documentación y presentación** | Se evaluará la claridad del código, la documentación escrita y la presentación del proyecto final. |

**Ejemplo de Proyecto (no se puede realizar sobre este tema):**

**Tema**: Asesoría sobre la planificación de estudios

* **Base de Conocimiento**:
  + Hechos: estudiante(nombre, carrera, horas\_disponibles, nivel\_estudio).
  + Reglas:
    - planificar\_estudios(X) :- estudiante(X, \_, H, N), H >= 4, N == 'Licenciatura', escribir\_plan\_de\_estudios(X).
    - planificar\_estudios(X) :- estudiante(X, \_, H, N), H < 4, N == 'Licenciatura', sugerir\_horas\_extra(X).
* **Motor de Inferencia**: En función de las horas disponibles y el nivel de estudio del estudiante, el sistema determinará si el estudiante tiene suficiente tiempo para realizar un plan de estudio completo o si necesita sugerencias para incrementar su tiempo de estudio.
* **Interfaz de Usuario**: El sistema pedirá información como el nombre del estudiante, su carrera, las horas disponibles para estudiar y su nivel de estudios, luego proporcionará un plan de estudio sugerido o recomendaciones de tiempo adicional.

**Ejemplo de interacción con el sistema:**

Bienvenido al sistema experto de planificación de estudios.

Por favor, ingrese su nombre: Santiago

¿En qué carrera está? Ingeniería de Sistemas

¿Cuántas horas tiene disponibles para estudiar paradigmas de programación por semana? 2

¿Cuál es su nivel de estudio? Ingeniería

El sistema sugiere que debería estudiar más horas para cumplir con el taller. Se recomienda estudiar al menos 2 horas diarias.