



INVESTIGACION 2: ARQUITECTURA DE UN SISTEMA EXPERTO

Materia: Sistemas Expertos

Alumno: Eric Uriel Cabrera Hernández 22310206

El experto

El experto en los sistemas expertos no es el software, sino la persona especialista en el área de aplicación, cuyo conocimiento se captura y modela para que el sistema pueda razonar como él.

Durante la construcción del sistema, ese conocimiento del experto se transfiere al sistema por medio del ingeniero del conocimiento, que lo formaliza en una base de conocimiento (reglas, hechos, heurísticas).

El experto humano es indispensable porque:

1. Aporta el conocimiento especializado

- El sistema no “sabe” nada por sí mismo, necesita que alguien con experiencia en el área defina las reglas, hechos y heurísticas.
- Ejemplo: en un sistema médico, el experto (médico especialista) describe síntomas, diagnósticos y tratamientos.

2. Valida el sistema

- Revisa si el razonamiento del sistema coincide con la realidad.
- Confirma que las reglas implementadas sean correctas y que no falten excepciones.

3. Resuelve ambigüedades

- El conocimiento humano muchas veces no es blanco o negro, hay zonas grises que el experto ayuda a aclarar.
- Ejemplo: “si la fiebre es alta y se acompaña de tos, probablemente sea infección respiratoria”.

4. Actualiza el conocimiento

- El experto ayuda a mantener vigente el sistema, porque el conocimiento en un campo puede cambiar con el tiempo

El experto humano aporta su conocimiento a un sistema experto a través de un proceso de adquisición de conocimiento, que normalmente se hace con ayuda del ingeniero del conocimiento.

Formas en que el experto transmite su conocimiento:

1. Entrevistas y consultas

- El ingeniero del conocimiento entrevista al experto, preguntándole cómo resuelve problemas en su campo.
- Ejemplo: “Si un motor no arranca, ¿qué pasos sigue para diagnosticar la falla?”

2. Análisis de casos reales

- El experto explica ejemplos de problemas ya resueltos (casos clínicos, averías, juicios, etc.) y cómo llegó a la solución.
- Esto ayuda a descubrir patrones y reglas.

3. Formalización de reglas y heurísticas

- El conocimiento se traduce en reglas del tipo:
 - Si ocurre A y B, entonces probablemente C.
 - Si el síntoma es X y no se observa Y, entonces descartar Z.

4. Revisión y validación del sistema

- El experto prueba el sistema y verifica que sus conclusiones sean correctas.
- Si encuentra errores o lagunas, aporta nuevas reglas o excepciones.

5. Documentos y experiencia tácita

- El experto también comparte manuales, protocolos, normativas y, sobre todo, su experiencia personal (su “intuición profesional”), que luego se traduce en conocimiento explícito.

Modulo de adquisición de conocimiento

En un sistema experto, el módulo de adquisición de conocimientos es el componente encargado de obtener, organizar y actualizar el conocimiento del experto humano para que pueda almacenarse en la base de conocimiento.

El módulo de adquisición de conocimientos es como el puente entre el experto humano y el sistema experto, encargado de capturar y formalizar el saber del especialista para que el sistema pueda usarlo en el razonamiento.

El módulo de adquisición de conocimiento obtiene el saber del experto para transformarlo en reglas formales que el sistema pueda usar y así razonar como lo haría un especialista humano. Permite agregar nuevas reglas, modificar o eliminar las obsoletas, manteniendo el sistema vigente.

El módulo de adquisición de conocimientos obtiene el conocimiento del experto porque el sistema experto por sí solo no sabe nada, necesita que alguien le “enseñe” la experiencia y reglas del dominio.

¿PARA QUÉ LO OBTIENE?

1. Construir la base de conocimiento

- El conocimiento del experto se traduce en hechos, reglas y procedimientos que el sistema usará para razonar.

2. Representar la experiencia humana

- Captura tanto el conocimiento explícito (manuales, protocolos) como el tácito (intuición, experiencia práctica).

3. Permitir que el sistema rzone como un experto

- Una vez almacenado, el motor de inferencia (subsistema cognímático) puede usarlo para resolver problemas.

4. Actualizar el sistema cuando cambian las condiciones del dominio

- El conocimiento humano evoluciona: aparecen nuevas enfermedades, tecnologías, leyes, etc.
- El módulo de adquisición permite mantener el sistema actualizado con el experto.

Base de conocimientos

La base de conocimientos es el conjunto de hechos, reglas y relaciones que representan el conocimiento de un dominio específico.

En un sistema experto, la base de conocimientos es el corazón del sistema, porque allí se almacena todo el saber del experto humano de manera organizada para que el sistema pueda usarlo.

Contenido típico:

1. Hechos → información concreta del dominio.

- Ejemplo: “Un motor necesita combustible para funcionar”.

2. Reglas de producción (condición → acción).

- Ejemplo: “Si el motor no arranca y hay combustible, entonces revisar la batería”.

3. Heurísticas → atajos o estrategias que usa el experto.

- Ejemplo: “Si el paciente tiene fiebre y tos, lo más probable es que sea una infección respiratoria”.

Función en el sistema experto:

- Proporciona la base de datos de conocimientos especializados.
- Es usada por el motor de inferencia (subsistema cognímático) para razonar y resolver problemas.
- Se puede ampliar o actualizar a medida que el experto aporta nuevo conocimiento.

¿CÓMO SE UTILIZA LA BASE DE CONOCIMIENTO?

1. Ingreso de datos del problema

- El usuario introduce información sobre el caso (síntomas, fallas, condiciones).
- Esa información pasa a la base de hechos.

2. Consulta del motor de inferencia

- El motor de inferencia busca en la base de conocimiento las reglas que se pueden aplicar a los hechos actuales.
- Ejemplo:
 - Hecho: “El paciente tiene fiebre de 39°C”.
 - Regla en la base de conocimiento: “Si la fiebre es > 38°C y hay dolor de garganta → posible amigdalitis”.

3. Aplicación de reglas y razonamiento

- El sistema aplica reglas paso a paso (encadenamiento hacia adelante o hacia atrás).
- A medida que se generan nuevas conclusiones, se agregan a la base de hechos.

4. Generación de conclusiones o recomendaciones

- Finalmente, gracias a la combinación de hechos (caso específico) y conocimiento (reglas generales), el sistema da una respuesta.
- Ejemplo: “El paciente probablemente tiene amigdalitis”.

5. Explicación del razonamiento

- Muchos sistemas expertos incluyen un módulo de explicación, que muestra qué reglas de la base de conocimiento se usaron para llegar a la conclusión

Base de hechos

La base de hechos (también llamada memoria de trabajo) es el componente del sistema experto donde se almacenan los datos específicos del problema que se está resolviendo en ese momento.

Es temporal y dinámica, a diferencia de la base de conocimiento, que es general y permanente.

Características de la base de hechos:

1. Contiene información particular del caso actual.

- Ejemplo: “El paciente Juan tiene fiebre de 39°C”.

2. Se actualiza durante el razonamiento.

- A medida que el motor de inferencia deduce nuevas conclusiones, estas se agregan como hechos.
- Ejemplo: “Juan probablemente tiene amigdalitis”.

3. Es cambiante y temporal.

- Cada vez que se analiza un nuevo caso, la base de hechos se reinicia con la información del nuevo problema.

La base de hechos en un sistema experto no se “programa” como un bloque fijo, sino que se construye dinámicamente cada vez que el sistema trabaja con un caso concreto.

¿CÓMO SE HACE LA BASE DE HECHOS?

1. Ingreso de datos iniciales

- El usuario o un sensor introduce los datos del caso actual.
- Ejemplo: en un sistema médico, el usuario ingresa “*El paciente tiene fiebre de 39°C*”.
- Estos datos iniciales se convierten en los primeros hechos.

2. Representación estructurada

- Los hechos se guardan en un formato que el sistema pueda manejar.
- Suelen expresarse como pares atributo-valor o proposiciones lógicas:
 - *Síntoma(fiebre, 39°C)*
 - *Síntoma(dolor_garganta, sí)*

3. Adición de nuevos hechos durante la inferencia

- El motor de inferencia aplica reglas de la base de conocimiento y genera nuevos hechos.
- Ejemplo:
 - Regla: “*Si fiebre > 38°C y dolor de garganta → probable amigdalitis*”.
 - Nuevo hecho agregado: “*Diagnóstico = amigdalitis probable*”.

4. Actualización dinámica

- La base de hechos se va enriqueciendo con las conclusiones parciales.
- Cada nuevo hecho puede activar más reglas.

5. Reinicio en cada caso

- Cuando el sistema termina de resolver un problema, la base de hechos se limpia.
- El siguiente caso comienza con una base de hechos nueva y vacía.

Motor de inferencia

El motor de inferencia (también llamado subsistema cognitivo o motor de razonamiento) es el cerebro del sistema experto.

Es el módulo encargado de aplicar las reglas de la base de conocimiento sobre los hechos de la base de hechos, para llegar a conclusiones, diagnósticos o recomendaciones.

unciones principales:

1. Seleccionar reglas relevantes

- Busca en la base de conocimiento qué reglas se pueden aplicar con los hechos actuales.

2. Aplicar las reglas

- Usa técnicas de razonamiento (ejemplo: encadenamiento hacia adelante o hacia atrás) para combinar hechos y reglas.

3. Generar nuevos hechos

- A medida que aplica reglas, produce nuevas conclusiones que se añaden a la base de hechos.

4. Controlar el proceso de razonamiento

- Decide qué regla aplicar primero si hay varias posibles.
- Evita bucles o razonamientos redundantes.

MÉTODOS DE INFERENCIA EN SISTEMAS EXPERTOS

El motor de inferencia puede razonar de dos maneras principales: encadenamiento hacia adelante y encadenamiento hacia atrás.

1. Encadenamiento hacia adelante (Forward chaining)

- Es un razonamiento de datos a conclusiones.
- El sistema parte de los hechos conocidos (base de hechos) y aplica las reglas hasta llegar a una nueva conclusión.
- Se usa mucho en diagnóstico, monitoreo y predicción

2. Encadenamiento hacia atrás (Backward chaining)

- Es un razonamiento de meta a datos.
- El sistema parte de una hipótesis o conclusión posible y busca en la base de conocimiento qué hechos deben cumplirse para confirmarla o descartarla.
- Se usa mucho en sistemas de diagnóstico y asesoría, cuando se quiere probar una hipótesis concreta.

¿CÓMO SE HACE UN MOTOR DE INFERENCIA?

1. Definir la representación del conocimiento

Antes de programar el motor, hay que decidir cómo estarán representados los hechos y las reglas:

- Hechos → pares atributo-valor, proposiciones lógicas o declaraciones tipo Síntoma(fiebre, 39°C).

- Reglas → expresadas como *SI (condiciones) ENTONCES (acción/conclusión)*.
 - Ejemplo: *Si fiebre > 38°C y dolor de garganta → probable amigdalitis.*

2. Diseñar la estrategia de inferencia

El motor debe saber cómo aplicar las reglas:

- Encadenamiento hacia adelante → parte de los hechos y genera conclusiones.
- Encadenamiento hacia atrás → parte de una hipótesis y busca comprobarla.
- Algunos motores permiten usar ambos según el problema.

3. Implementar el ciclo de razonamiento

Un motor de inferencia normalmente sigue este ciclo:

1. Buscar reglas aplicables → revisar qué reglas cumplen con los hechos actuales.
2. Seleccionar la regla a aplicar → si hay varias posibles, elegir según prioridad o heurísticas.
3. Ejecutar la regla → añadir la conclusión (nuevo hecho) a la base de hechos.
4. Repetir hasta que no haya más reglas aplicables o se alcance la meta.

4. Resolver conflictos (control de reglas)

Cuando varias reglas pueden aplicarse al mismo tiempo, el motor necesita una estrategia de resolución de conflictos, como:

- Prioridad fija de reglas.
- Orden temporal (la primera que apareció).
- Regla más específica antes que una general.

5. Programación práctica

El motor se puede implementar en distintos lenguajes o entornos:

- Lenguajes de propósito general: Python, Java, C++.
- Lenguajes especializados en IA: Prolog, Lisp.
- Herramientas de sistemas expertos: CLIPS, Jess, Drools.

Modulo de explicación

El módulo de explicación es el componente del sistema experto que se encarga de mostrar al usuario cómo y por qué el sistema llegó a una conclusión o recomendación.

El módulo de explicación permite que el sistema experto sea transparente y confiable, mostrando al usuario qué reglas usó y por qué tomó ciertas decisiones, de manera similar a cómo un humano justificaría su razonamiento.

Funciones principales:

1. Explicar las conclusiones (el qué)

- Justifica qué respuesta dio el sistema.
- Ejemplo: “*El diagnóstico es amigdalitis porque se cumplieron las condiciones fiebre > 38°C y dolor de garganta*”.

2. Explicar el razonamiento (el cómo)

- Muestra las reglas aplicadas y en qué orden se usaron.
- Ejemplo:
 - Regla 1 aplicada: *Si fiebre > 38°C y dolor de garganta → probable amigdalitis*.

3. Explicar las preguntas del sistema (el por qué)

- Si el sistema hace preguntas al usuario, puede explicar por qué solicita esa información.
- Ejemplo: “*Pregunto si hay dolor de garganta porque esta condición es necesaria para evaluar la posible amigdalitis*”.

¿CÓMO FUNCIONA EL MÓDULO DE EXPLICACIÓN?

1. Monitoreo del motor de inferencia

- Mientras el motor de inferencia aplica reglas y genera nuevos hechos, el módulo de explicación registra qué reglas se usaron y en qué orden.

2. Registro de la traza de razonamiento

- Guarda la secuencia de:
 - Hechos iniciales.
 - Reglas aplicadas.
 - Nuevos hechos generados.
- Esto crea una historia del razonamiento.

3. Generación de explicaciones

- A partir de esa traza, el sistema puede responder preguntas del usuario, por ejemplo:
 - ¿Por qué esta conclusión? → Explica qué reglas y hechos la provocaron.
 - ¿Cómo llegó a esta respuesta? → Muestra la cadena de reglas usadas.

- ¿Por qué esta pregunta? → Justifica por qué necesitaba cierta información para continuar.

4. Presentación al usuario

- La explicación se muestra en lenguaje natural o en forma de pasos lógicos.
- Ejemplo:
 - “*El diagnóstico es amigdalitis porque el paciente tiene fiebre de 39°C y dolor de garganta, lo cual cumple la regla R1*”.

Interfaz de usuario

La interfaz de usuario es el módulo que permite la comunicación entre el usuario y el sistema experto. La interfaz de usuario es el módulo que conecta al usuario con el sistema experto, permitiéndole dar datos, recibir resultados y comprender las conclusiones del sistema de manera amigable e interactiva.

FUNCIONES PRINCIPALES:

1. Entrada de información (input)

- Permite al usuario ingresar los hechos del caso actual (síntomas, condiciones, datos técnicos, etc.).

2. Presentación de resultados (output)

- Muestra las conclusiones, diagnósticos o recomendaciones que el sistema genera.

3. Comunicación interactiva

- El sistema puede hacer preguntas al usuario para obtener datos que necesita.

4. Explicaciones del razonamiento

- Permite que el usuario consulte al módulo de explicación, preguntando:

5. Facilidad de uso

- Su objetivo es hacer que el sistema sea comprensible y accesible para personas que no son programadores ni expertos en computación.

¿CÓMO FUNCIONA LA INTERFAZ DE USUARIO?

1. Captura de datos (input)

- El usuario introduce información inicial del problema (síntomas, condiciones, parámetros).
- La interfaz traduce esos datos en un formato comprensible para el sistema y los envía a la base de hechos.
- Ejemplo: el usuario escribe “*El paciente tiene fiebre de 39°C*”.

2. Interacción dinámica

- Si el motor de inferencia necesita más información, la interfaz hace preguntas al usuario.
- Ejemplo:
 - Sistema: “*¿El paciente tiene dolor de garganta? (Sí/No)*”
 - Usuario: “*Sí*”.
- La respuesta se convierte en un nuevo hecho dentro del sistema.

3. Presentación de resultados (output)

- Una vez que el motor de inferencia llega a una conclusión, la interfaz muestra los resultados de manera clara.
- Ejemplo: “*El sistema concluye: el paciente probablemente tiene amigdalitis*”.

4. Explicación al usuario

- La interfaz conecta con el módulo de explicación.
- Si el usuario pregunta “*¿Por qué este diagnóstico?*”, la interfaz traduce la justificación en lenguaje natural.
- Ejemplo:
 - “*Se diagnosticó amigdalitis porque el paciente presenta fiebre de 39°C y dolor de garganta, lo que activa la regla R1*”.

5. Facilidad de uso

- Dependiendo del diseño, puede ser:
 - Texto simple (preguntas y respuestas tipo chat).
 - Gráfica (formularios, menús, botones).
 - Incluso voz o asistentes inteligentes en sistemas modernos.

Sensores y Base de datos

En algunos sistemas expertos, además de los módulos básicos (base de conocimientos, base de hechos, motor de inferencia, módulo de explicación, interfaz de usuario), también se incluyen sensores y bases de datos como fuentes de información externa.

Sensores en un sistema experto

- ¿Qué hacen?
Los sensores sirven para capturar datos del entorno en tiempo real y pasarlo directamente a la base de hechos.

- Ejemplo en un sistema médico: un termómetro digital envía la temperatura del paciente.
- Ejemplo en un sistema industrial: un sensor de presión informa que una máquina está a 120 psi.
- Rol dentro del sistema experto:
Reemplazan o complementan la entrada manual de datos del usuario.
Así, el sistema experto puede tomar decisiones automáticas basadas en información actualizada

Base de datos en un sistema experto

- ¿Qué hace?
La base de datos se usa como un almacén de información estructurada que el sistema experto consulta cuando necesita hechos adicionales.
 - Ejemplo en medicina: historial clínico de pacientes (edad, alergias, diagnósticos anteriores).
 - Ejemplo en negocios: registros de ventas para recomendar estrategias de mercado.
- Diferencia con la base de conocimientos:
 - La base de conocimientos contiene reglas y experiencia del experto (general).
 - La base de datos guarda datos concretos y masivos (históricos, numéricos, externos).

¿CÓMO SE INGRESAN LOS DATOS DE LA BASE DE DATOS?

1. Conexión al sistema experto

El sistema experto necesita un módulo de conexión (software intermedio) que le permita comunicarse con la base de datos.

- En un entorno real se usa SQL, APIs o conectores especiales.

2. Extracción de información relevante

No se trae toda la base de datos, solo los datos necesarios para el caso actual.

- Ejemplo: de una base de datos hospitalaria, se extraen:
 - Nombre del paciente.
 - Edad.

3. Conversión en hechos

Los datos de la base de datos se transforman en hechos que la base de hechos pueda manejar.

- Ejemplo en pseudocódigo:
 - BD: temperatura = 39
 - Hecho en la base de hechos: fiebre(39°C)

4. Carga en la base de hechos

Una vez convertidos, los hechos ingresan en la base de hechos, que es dinámica y se actualiza para cada problema.

5. Uso por el motor de inferencia

Con esos hechos cargados desde la base de datos, el motor de inferencia puede aplicar las reglas de la base de conocimiento y llegar a conclusiones.

¿Cómo se adquiere la información de los sensores?

1. Captura de datos en tiempo real

- Los sensores físicos (de temperatura, presión, humedad, velocidad, etc.) miden constantemente una variable del entorno.
- Cada sensor convierte esa medición en una señal digital que puede ser entendida por un computador.

2. Interfaz de adquisición de datos (hardware + software)

- Los sensores se conectan a través de:
 - Microcontroladores (Arduino, Raspberry Pi, PLCs).
 - Módulos de comunicación (WiFi, Bluetooth, Zigbee, protocolos industriales como Modbus o CAN).
- El sistema experto recibe la información gracias a un software de adquisición de datos que traduce la señal del sensor a un valor usable.

3. Transformación en hechos

Los datos capturados se convierten en hechos dentro de la base de hechos.

- Ejemplo:
 - Sensor: mide 39.2 °C.
 - Hecho: temperatura (39.2).

4. Carga en la base de hechos

Los valores medidos se agregan a la base de hechos para que el motor de inferencia los use junto con las reglas de la base de conocimiento.

5. Uso en el razonamiento

El motor de inferencia aplica reglas sobre esos datos.

- Ejemplo en un invernadero:
 - Sensor de humedad mide 30%.

- Regla: *Si humedad < 40% → activar riego automático.*
- Conclusión: el sistema activa el riego.

Usuario

El usuario es el beneficiario final del sistema, quien consulta al sistema experto porque no tiene el mismo nivel de conocimiento que el experto humano o porque necesita apoyo en el proceso de razonamiento.

El sistema experto funciona como un asistente inteligente, y el usuario es quien lo utiliza para resolver un caso concreto.

FUNCIONES DEL USUARIO EN EL SISTEMA EXPERTO

1. Ingresar información (datos o hechos)

- El usuario proporciona los datos iniciales del problema.
- Ejemplo: en un sistema médico, el usuario ingresa “*Paciente con fiebre de 39°C y dolor de garganta*”.

2. Responder preguntas del sistema

- Durante el razonamiento, el sistema puede necesitar datos adicionales y el usuario los proporciona.
- Ejemplo: “*¿El paciente tiene tos? Sí/No*”.

3. Consultar resultados

- El usuario recibe del sistema conclusiones, diagnósticos o recomendaciones.
- Ejemplo: “*Probable diagnóstico: amigdalitis*”.

4. Solicitar explicaciones

- El usuario puede preguntar “*¿Por qué llegaste a esa conclusión?*”, y el sistema le responde a través del módulo de explicación.

Control de coherencia

El control de coherencia en un sistema experto es un mecanismo que asegura que el conocimiento almacenado en la base de conocimientos no tenga contradicciones, redundancias ni inconsistencias, de modo que el sistema pueda razonar correctamente.

Es el proceso de verificación y validación de las reglas y hechos en un sistema experto, con el fin de:

1. Evitar **contradicciones** (reglas que se anulan entre sí).
2. Evitar **redundancias** (reglas duplicadas o innecesarias).

3. Evitar **inconsistencias** (reglas que llevan a conclusiones opuestas).
4. Garantizar que el sistema produzca resultados lógicos, confiables y predecibles.

¿CÓMO SE HACE EL CONTROL DE COHERENCIA?

1. Revisión manual (por el experto humano y el ingeniero de conocimiento)

- Analizar las reglas escritas y asegurarse de que no se contradigan.
- Requiere reuniones periódicas entre el experto humano y el ingeniero de conocimiento.

2. Validación automática con el motor de inferencia

- Probar el sistema con casos de prueba conocidos y verificar si las conclusiones son correctas.
- Ejemplo: ingresar un caso médico real y ver si el diagnóstico coincide con lo esperado.

3. Detección de conflictos en reglas

- Se usan algoritmos para encontrar:
 - **Conflictos directos** → dos reglas con la misma condición pero conclusiones opuestas.
 - **Redundancias** → reglas que dicen lo mismo.
 - **Lagunas** → situaciones sin reglas que las cubran.

4. Mantenimiento periódico de la base de conocimientos

- A medida que cambian los conocimientos en el dominio (ejemplo: nueva enfermedad o nueva tecnología), se revisa la base de conocimientos para mantener su coherencia.

Ingeniero del conocimiento

El ingeniero del conocimiento es el especialista en informática e inteligencia artificial encargado de construir, organizar y mantener el sistema experto.

Su papel es actuar como puente entre el experto humano (que sabe del dominio del problema) y el sistema experto (el programa que usará ese conocimiento).

Funciones principales del ingeniero del conocimiento

1. Adquisición del conocimiento

- Entrevista al experto humano para obtener las reglas, estrategias y métodos de razonamiento que usa.
- Traduce ese conocimiento en un formato entendible por la computadora.

2. Representación del conocimiento

- Decide cómo almacenar el conocimiento:

- Reglas de producción (SI... ENTONCES...).
- Redes semánticas.
- Marcos (frames).

3. Implementación del sistema experto

- Diseña y programa la base de conocimientos, la base de hechos, el motor de inferencia y otros módulos.

4. Control de coherencia y consistencia

- Verifica que las reglas no sean contradictorias, redundantes o incompletas.

5. Pruebas y validación

- Ejecuta casos de prueba para asegurarse de que el sistema dé conclusiones correctas.

6. Mantenimiento y actualización

- Ajusta la base de conocimientos cuando cambian las condiciones del dominio.
- Ejemplo: en un sistema médico, cuando se descubre un nuevo tratamiento.

Subsistema de ejecución de órdenes

El subsistema de ejecución de órdenes es el componente encargado de llevar a cabo las acciones prácticas o recomendaciones que el sistema experto determina, después de que el motor de inferencia llega a una conclusión.

Es decir, transforma las decisiones del sistema en acciones concretas dentro del entorno o hacia el usuario.

Funciones principales

1. Interpretar las conclusiones del motor de inferencia

- Recibe la salida (diagnóstico, decisión, instrucción) y la convierte en una orden ejecutable.

2. Ejecutar acciones automáticas

- Puede activar dispositivos, enviar comandos a máquinas o interactuar con otros sistemas.
- Ejemplo: en un sistema industrial, activar un ventilador si la temperatura supera un límite.

3. Comunicar recomendaciones al usuario

- Cuando la acción no es automática, se presenta al usuario como una orden o sugerencia.
- Ejemplo: “Administre 500 mg de paracetamol cada 8 horas”.

4. Controlar la interacción con el entorno

- Puede trabajar junto con sensores y actuadores, cerrando el ciclo completo de monitoreo y acción.

Subsistema de aprendizaje

El subsistema de aprendizaje es el módulo encargado de que el sistema experto mejore y actualice sus conocimientos con el tiempo, ya sea incorporando nueva información o corrigiendo reglas existentes.

Permite que el sistema experto no se quede estático, sino que evolucione a medida que cambian las condiciones del entorno o se adquiere más conocimiento del dominio.

Funciones principales

1. Adquisición automática de conocimiento

- Recopila información nueva a partir de:
 - Experiencias pasadas (casos ya resueltos).
 - Interacción con el experto humano.
 - Datos obtenidos de sensores o bases de datos externas.

2. Modificación de la base de conocimientos

- Agrega nuevas reglas, ajusta las existentes o elimina las que ya no son válidas.

3. Corrección de errores

- Si el sistema da conclusiones incorrectas, el subsistema de aprendizaje ajusta la base de conocimientos para evitar repetir el error.

4. Generalización de experiencias

- Aprende patrones de múltiples casos similares y los convierte en nuevas reglas de conocimiento.

5. Mantenimiento de la coherencia

- Al introducir nuevo conocimiento, revisa que no se creen contradicciones en la base de conocimientos.

¿CÓMO FUNCIONA EL SUBSISTEMA DE APRENDIZAJE?

1. Entrada de nueva información

El subsistema recibe conocimiento desde diferentes fuentes:

- Del experto humano, que aporta reglas nuevas o corrige errores.
- De experiencias pasadas (casos ya resueltos).

- De bases de datos o sensores, que generan patrones nuevos.

2. Procesamiento y análisis

- Compara la nueva información con la base de conocimientos actual.
- Detecta:
 - Si la regla ya existe (evita redundancia).
 - Si contradice a otra (control de coherencia).
 - Si es completamente nueva (se incorpora como conocimiento fresco).

3. Actualización de la base de conocimientos

- Si la información es válida, el subsistema agrega, modifica o elimina reglas en la base de conocimientos.
- Esto le permite adaptarse a nuevas situaciones sin que un programador tenga que reescribir todo el sistema.

4. Validación del nuevo conocimiento

- El sistema prueba las nuevas reglas en casos de prueba.
- Si las conclusiones son correctas y coherentes, el conocimiento queda incorporado.

5. Uso en el razonamiento futuro

- A partir de entonces, el motor de inferencia ya puede usar las nuevas reglas para resolver problemas similares.