



Modulo 2

Tipos de IA: Búsqueda y Resolución de Problemas

Repaso

🔵 Agente

Un agente es cualquier cosa capaz de **percibir su medioambiente** o entorno a través de sensores y actúa sobre ese entorno a través de actuadores. Por ejemplo, una aspiradora robot tiene sensores para detectar la suciedad y actuadores para moverse y limpiar. Un agente humano recibe la información de su entorno a través de los diferentes sentidos y nuestros actuadores serían brazos, piernas, boca, etc.

El término *percepción* se utiliza en este contexto para indicar que el agente puede recibir entradas en cualquier instante. *Russell y Norvig*

🔵 Agente Racional

Un agente racional es aquel que actúa de manera que **maximiza su desempeño** con base en la información disponible. Para definir si un agente es racional, se considera su función de desempeño, la percepción del entorno, el conocimiento disponible y las acciones posibles, también llamado **REAS (Rendimiento, Entorno, Actuadores, Sensores)**.

La racionalidad depende de:

1. *El objetivo o criterio de desempeño.*
Ejemplo: Maximizar la limpieza en una casa.
2. *El conocimiento previo del entorno.*
Ejemplo: Saber en qué habitaciones suele haber más suciedad.
3. *Las acciones posibles.*
Ejemplo: Moverse hacia una habitación sucia o regresar a la base para recargarse.
4. *Las percepciones actuales.*
Ejemplo: Detectar dónde hay más suciedad en tiempo real.

El entorno de trabajo varía según distintos parámetros. Pueden ser total o parcialmente visibles, deterministas o estocásticos, episódicos o secuenciales, estáticos o dinámicos, discretos o continuos, y formados por un único agente o por varios agentes. *Russell y Norvig*

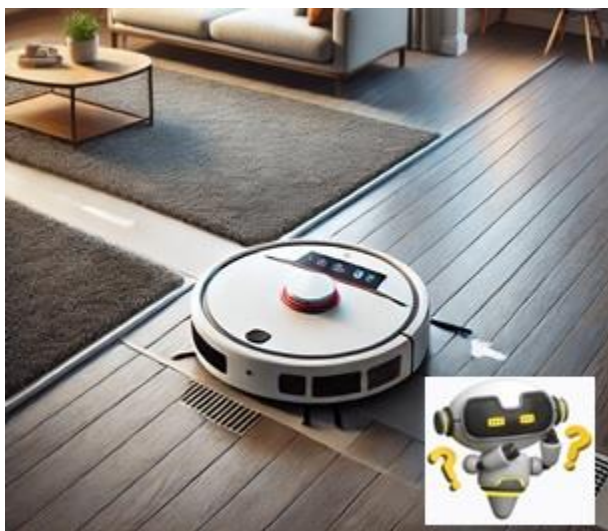
🔵 Agente Inteligente

Aunque un agente inteligente también busca tomar decisiones óptimas, se destaca por su capacidad de aprender y adaptarse con el tiempo. Esto implica que un agente inteligente no solo sigue reglas predefinidas, sino que mejora su desempeño al adquirir nuevos conocimientos a partir de la experiencia. **El agente inteligente es calcular varias medidas, maximizar la utilidad** de sus acciones e ir hacia delante.

*El principio de la **máxima utilidad esperada** (MUE) establece que un agente racional debe elegir aquella acción que maximice la utilidad esperada del agente.*

Recuerdan la pregunta: **¿La aspiradora puede ser considerada un agente racional?**

Sí y no. Es un agente racional en un entorno controlado (*limitado, estático, determinístico*). Si ponemos la misma aspiradora en una habitación con concurrencia de personas, mascotas que van y vienen, seguramente le va a “costar” llegar a ejecutar un trabajo óptimo. **¿Inteligente? Definitivamente NO.** Porque no aprende ni se adapta. Si se le introduce un obstáculo nuevo o un cambio en el entorno (como una habitación adicional) o cualquier entorno que *sea estocástico, continuo y dinámico*, seguirá ejecutando el mismo conjunto de reglas sin mejorar o en el peor de los casos quedará sin actuar porque no sabe cómo procesar tanta información.



¿Qué se considera medio ambiente o entorno?

Russell y Norvig definen el medio ambiente como cualquier cosa fuera del agente que interactúa con él. Para analizar el entorno, proponen clasificarlo en diferentes dimensiones:

- 🧐 **Observable:** Si el agente tiene acceso a toda la información relevante (entorno completamente observable) o solo a una parte de ella (parcialmente observable).
- 🧐 **Determinístico:** Las acciones tienen resultados predecibles o si hay incertidumbre en el resultado (ambiente estocástico).
- 🧐 **Episódico:** Si las decisiones actuales afectan solo el presente (episódico) o si afectan futuros estados (secuencial).
- 🧐 **Dinámico:** Si el entorno cambia mientras el agente actúa (dinámico) o permanece constante (estático).
- 🧐 **Discreto o continuo:** Si los estados y acciones son discretos o continuos. En un juego de ajedrez hay un número finito de estados distintos (discreto). Un taxista conduciendo define un estado continuo y un problema de tiempo continuo.

Búsqueda y Resolución de Problemas

Muchos problemas pueden formularse como problemas de búsqueda. La resolución de problemas en IA se modela como un proceso de búsqueda en un espacio de estados. Cada problema se define por un conjunto de estados, acciones, un estado inicial, un estado objetivo y un camino (o secuencia de acciones) que lleva del estado inicial al estado objetivo.

Ya vimos los **agentes basados en objetivos**, estos agentes también se conocen como **agente resolvente-problemas**. Los agentes resolventes-problemas deciden qué hacer para encontrar secuencias de acciones que conduzcan a los estados deseables.

La clave es explorar los posibles caminos en este espacio para encontrar una solución óptima o al menos adecuada.

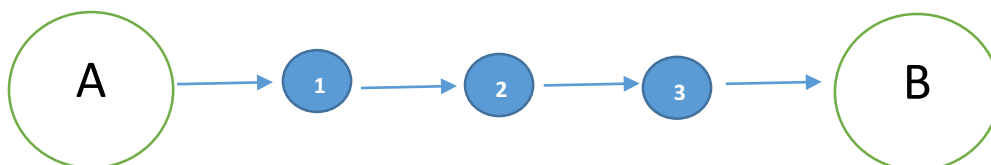
El primer paso para solucionar un problema es la *formulación del objetivo*, basado en la situación actual y la medida de rendimiento del agente.

Consideraremos **un objetivo como un conjunto de estados del mundo** (exactamente aquellos estados que satisfacen el objetivo). Dado un objetivo, la **formulación del problema es el proceso de decidir qué acciones y estados tenemos que considerar**.

Veamos un ejemplo: estás en una ciudad que no conoces y quieres ir a visitar un lugar turístico. . ¿Qué es lo primero que haces? Revisas a qué distancia estás de ese lugar. Si son varios kilómetros, puedes decidir usar el transporte público. Obviamente desconoces qué línea de colectivo debes tomar. Consultas la aplicación del transporte público con sus líneas y recorridos. Ahora sólo queda caminar hasta la parada. Si este lugar turístico no queda “tan lejos”, usas el GPS para ir caminando y sigues las instrucciones.

En general, *un agente con distintas opciones inmediatas de valores desconocidos puede decidir qué hacer, examinando las diferentes secuencias posibles de acciones que le conduzcan a estados de valores conocidos, y entonces escoger la mejor secuencia*.

Este proceso de hallar esta secuencia de A a B, se llama **búsqueda y planificación**. Los automóviles autónomos deben resolver problemas similares.



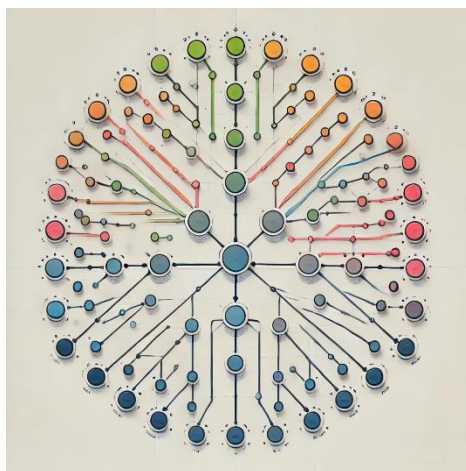
Un *problema* puede definirse, formalmente, por cuatro componentes:

- El **estado inicial** en el que comienza el agente.
- Una descripción de las posibles **acciones** disponibles por el agente.
- El **test objetivo**, el cual determina si un estado es un estado objetivo.
- Una función **costo del camino** que asigna un costo numérico a cada camino.

Un algoritmo de búsqueda toma como entrada un problema y devuelve una **solución** de la forma secuencia de acciones. Una vez que encontramos una solución, se procede a ejecutar las acciones que ésta recomienda. Esta es la llamada fase de **ejecución**.

En la vida diaria hay varias formas distintas de resolver un problema, todas dependen de criterios como tiempo, esfuerzo, coste, etc. Diferentes técnicas de búsqueda pueden llevar a diferentes soluciones.

Un problema se estructura como un grafo donde los nodos representan estados y los arcos representan acciones que mueven de un estado a otro. La búsqueda consiste en recorrer este grafo desde el estado inicial hasta el estado objetivo.



Hay muchas formas de representar los nodos, pero vamos a suponer que un nodo es una estructura de datos con cinco componentes:

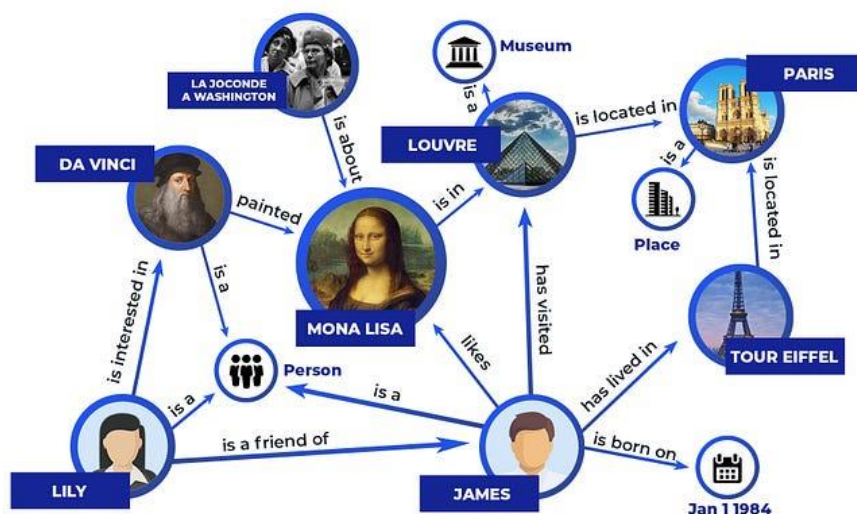
- ESTADO: el estado, del espacio de estados, que corresponde con el nodo;
- NODO PADRE: el nodo en el árbol de búsqueda que ha generado este nodo;
- ACCIÓN: la acción que se aplicará al padre para generar el nodo;

— COSTO DEL CAMINO: el costo, tradicionalmente denotado por $g(n)$, de un camino desde el estado inicial al nodo, indicado por los punteros a los padres; y

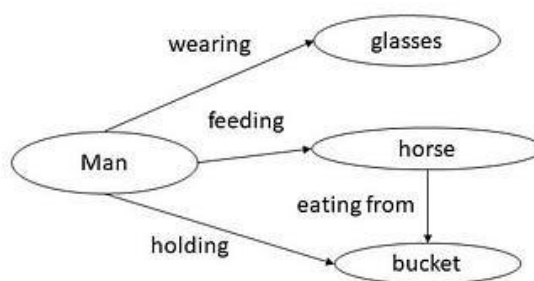
— PROFUNDIDAD: el número de pasos a lo largo del camino desde el estado inicial.

Es importante recordar la distinción entre nodos y estados. Un nodo es una estructura de datos usada para representar el árbol de búsqueda. Un estado corresponde a una configuración del mundo

Knowledge Graphs KG (Grafos de Conocimiento)



Fuente: <https://leslysandra.medium.com/intro-grafos-de-conocimiento-kg-121e37e6694>



KG ha sido usado para la presentación tanto en el procesamiento del lenguaje natural NLP como en la visión por computadora y es la base de Machine Learning.

Formulación de los problemas

La metodología para resolver problemas se ha aplicado a un conjunto amplio de entornos. Podemos definir dos tipos de problemas: problemas de *juguete* y del *mundo-real*.

Un **problema de juguete** se utiliza para ilustrar o ejercitar los métodos de resolución de problemas. Éstos se pueden describir de forma exacta y concisa. Esto significa que diferentes investigadores pueden utilizarlos fácilmente para comparar el funcionamiento de los algoritmos.

Un **problema del mundo-real** es aquel en el que la gente se preocupa por sus soluciones. Ellos tienden a no tener una sola descripción.

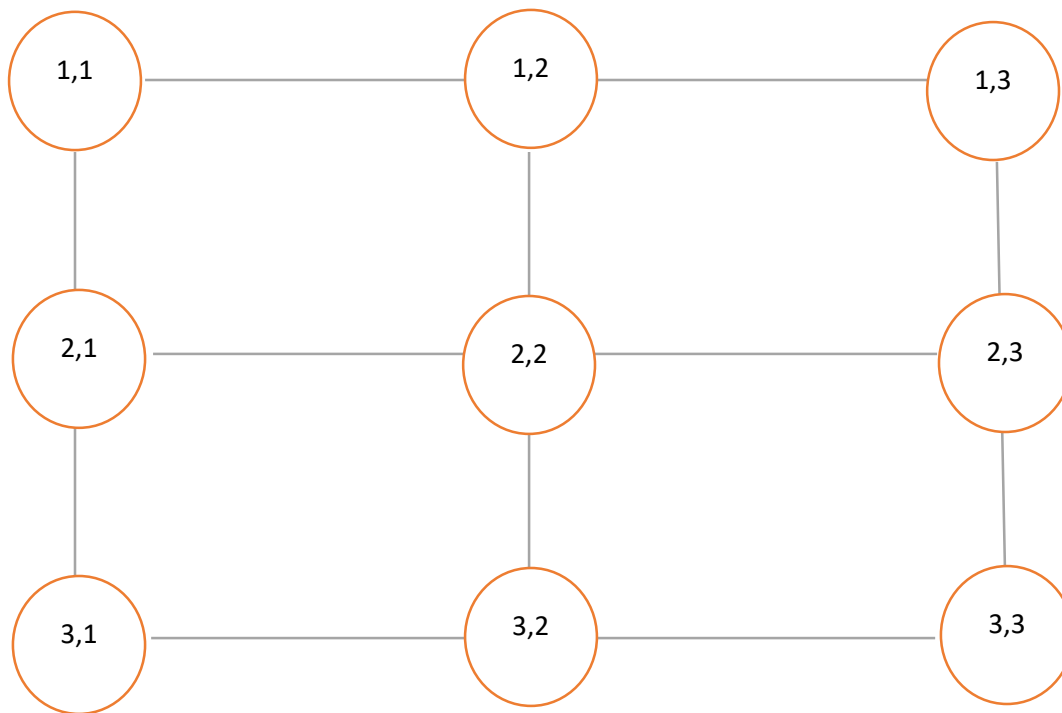
Analicemos un problema de juguete

✚ Tienes un robot que está programado para recolectar frutas en un pequeño huerto rectangular de 3x3 casillas. El huerto tiene tres árboles: uno en cada esquina (superior izquierda, superior derecha y la esquina inferior derecha). El robot siempre empieza en la esquina inferior izquierda (posición (3,1)) y su objetivo es recolectar todas las frutas pasando por cada árbol, terminando en la esquina inferior derecha (3,3).

Reglas:

1. El robot se puede mover arriba, abajo, izquierda o derecha.
2. Solo puede recolectar frutas si pasa por la casilla del árbol.
3. El objetivo es encontrar la secuencia de movimientos óptima (mínima cantidad de pasos) para recolectar todas las frutas y llegar a la esquina inferior derecha.





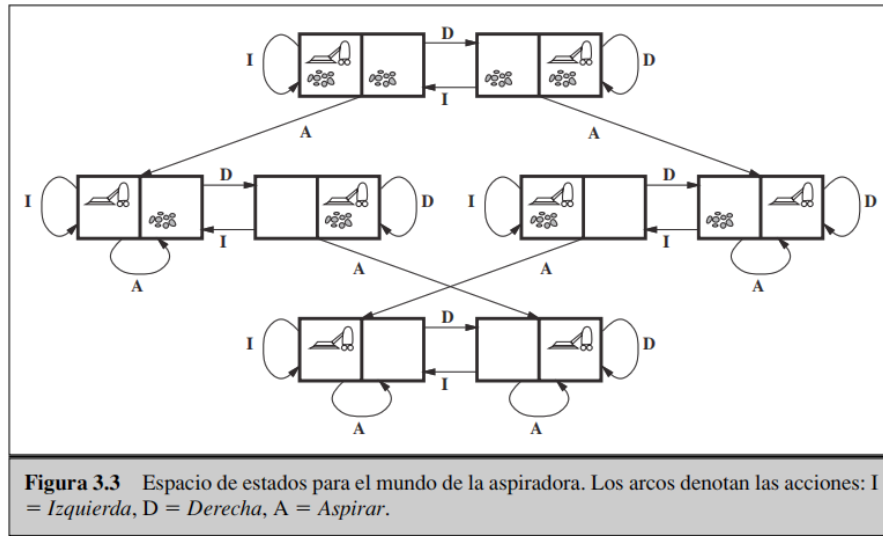
(1,1) - Árbol (1,3) – Árbol (3,1) - Start (3,3) - Árbol / Fin

¿Cuál es la secuencia óptima de movimientos para que el robot recoja todas las frutas y termine en la posición (3,3)?

☑ Otro problema de juguete propuesto por Russell y Norvig:

El mundo de la aspiradora, *introducido en el Capítulo 2. Éste puede formularse como sigue:*

- Estados: *el agente está en una de dos localizaciones, cada una de las cuales puede o no contener suciedad. Así, hay $2 \times 2 \times 2 \times 2 \times 2 \times 2 \times 2 \times 2 = 256$ posibles estados del mundo.*
- Estado inicial: *cualquier estado puede designarse como un estado inicial.*
- Función sucesor: *ésta genera los estados legales que resultan al intentar las tres acciones (Izquierda, Derecha y Aspirar). En la Figura 3.3 se muestra el espacio de estados completo.*
- Test objetivo: *comprueba si todos los cuadrados están limpios.*
- Costo del camino: *cada costo individual es 1, así que el costo del camino es el número de pasos que lo compone.*



Tipos de Búsqueda

☒ Búsqueda No Informada

No tienen información adicional acerca de los estados más allá de la que proporciona la definición del problema.

- 1) **Búsqueda en anchura (Breadth-First Search - BFS):** Explora todos los nodos en un nivel antes de pasar al siguiente. Es completa y encuentra la solución más corta, pero puede requerir mucho espacio de memoria.
- 2) **Búsqueda en profundidad (Depth-First Search - DFS):** Explora lo más profundo posible a lo largo de un camino antes de retroceder y probar otros. Requiere menos memoria, pero no garantiza la solución óptima.
- 3) **Búsqueda de costo uniforme:** Busca expandir el nodo con el costo acumulado más bajo, útil cuando las acciones tienen costos diferentes.

☒ Búsqueda Informada (Heurística)

- **Algoritmo A* (A-star):** Combina el costo acumulado para llegar a un nodo ($g(n)$) con una estimación heurística del costo para llegar al estado objetivo ($h(n)$). Esto proporciona un equilibrio entre exploración y eficiencia.

Para algunos problemas, la repetición de estados es inevitable. Esto incluye todos los problemas donde las acciones son reversibles, como son los problemas de búsqueda de rutas y los puzzles que deslizan sus piezas. Los árboles de la búsqueda para estos problemas son infinitos, pero si podemos parte de los estados repetidos, podemos cortar el árbol de búsqueda en un tamaño finito, generando sólo la parte del árbol que atraviesa el grafo del espacio de estados. Considerando solamente el árbol de búsqueda hasta una profundidad fija, es fácil encontrar casos donde la eliminación de estados repetidos produce una reducción exponencial del coste de la búsqueda.

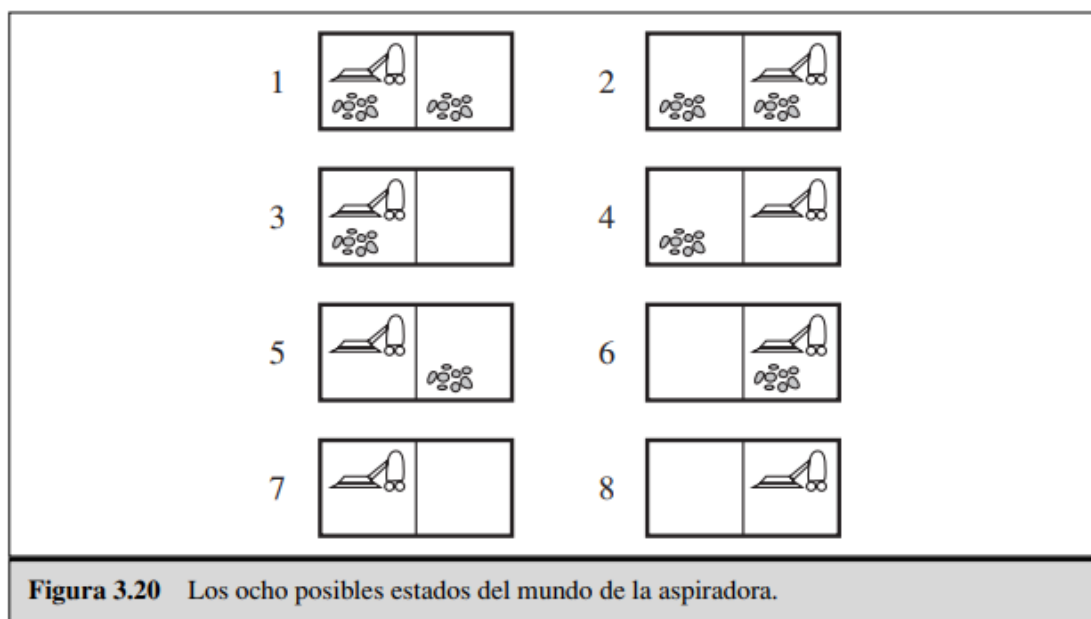


Figura 3.20 Los ocho posibles estados del mundo de la aspiradora.

En estrecha relación con los juegos, las técnicas de búsqueda y planificación fueron un ámbito en el que la IA propició grandes avances en los años sesenta del siglo pasado: algoritmos como el de minimax o la poda Alfa-Beta, desarrollados entonces, siguen siendo la base para la IA orientada a los juegos.

Para resolver

✓ Problema 1

Un robot con una barca de remos tiene que transportar tres mercancías al otro lado de un río: **un zorro, un pollo y un saco de maíz para pollos**. El zorro se comerá al pollo si tiene la oportunidad, y el pollo hará lo propio con el pienso, y ninguno de los dos casos es un resultado deseable. El robot puede evitar que los animales causen daños cuando está a su lado, pero solo él es capaz de manejar la barca de remos, y en la barca solo caben dos mercancías junto con el robot. **¿Cómo puede transportar el robot todas sus mercancías a la orilla opuesta del río?**

Hay cinco objetos que se pueden desplazar: el robot, la barca de remos, el zorro, el pollo y el maíz.



C: cercano

L: Lejano



Resuelve de manera que el zorro no se coma al pollo

☑ Problema 2

El problema de los misioneros y caníbales en general se forma como sigue. **Tres misioneros y tres caníbales** están en un lado de un río, con un **barco** que puede sostener a una o dos personas. Encuentre un modo de conseguir que todos estén en el otro lado, sin dejar alguna vez a un grupo de misioneros en un lugar excedido en número por caníbales. Este problema es famoso en IA porque fue el tema del primer trabajo que aproximó una formulación de problema de un punto de vista analítico (Amarel, 1968).

- Formule el problema de forma precisa, haciendo sólo las distinciones necesarias para asegurar una solución válida. Dibujar un diagrama del espacio de estados complete.
- Investiga los algoritmos de búsqueda propuestos por Russell y Norvig, cual de ellos implementarías para resolver el problema de manera óptima.

Solución de la transición a partir de CCCC

- ✓ (C, C, C, C) → El robot lleva al pollo → (L, C, L, C)
- ✓ (C, C, C, C) → El robot lleva al pollo y al maíz → (L, C, L, L)
- ✓ (C, C, C, C) → El robot lleva al zorro y al maíz → (L, L, C, L)
- ✓ (C, C, C, C) → El robot lleva al zorro y al pollo → (L, L, L, C)