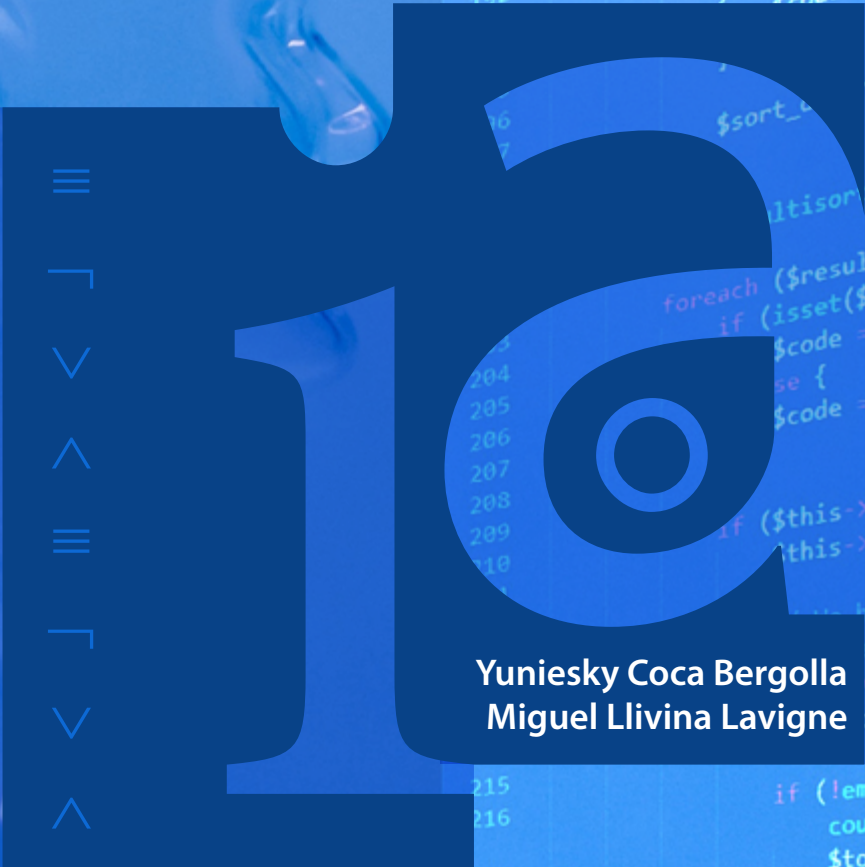


# ¿QUÉ ESTUDIA LA INTELIGENCIA ARTIFICIAL?



Yuniesky Coca Bergolla  
Miguel Llivina Lavigne





# ¿QUÉ ESTUDIA LA INTELIGENCIA ARTIFICIAL?

**Yuniesky Coca Bergolla**  
**Miguel Llivina Lavigne**



EDITORIAL EDUCACIÓN CUBANA

La Habana, noviembre de 2021

Esta obra está bajo una Licencia Creative Commons  
Atribución-NoComercial-CompartirIgual 4.0 Internacional.



© Yuniesky Coca Bergolla, 2021

© UNESCO, 2021

Diseño y Edición:

DI. Ismael Adán Quesada Chow

**ISBN: 978-959-18-1342-8**

Sello Editor EDUCACIÓN CUBANA

Dirección de Ciencia y Técnica - MINED

Calle 17 y O. Vedado. La Habana, Cuba.



# ÍNDICE

INTRODUCCIÓN.....	5
EL RAZONAMIENTO COMPUTACIONAL.....	7
■ La formulación de problemas de estado.....	9
■ Búsqueda a ciegas.....	17
■ Búsqueda heurística.....	18
■ Búsquedas locales.....	26
■ Otras formas de razonamiento en IA.....	30
EL CONOCIMIENTO Y SU REPRESENTACIÓN.....	41
■ Formas específicas de representación del conocimiento.....	42
■ La representación del todo en IA.....	47
TRATAMIENTO DE LA INCERTIDUMBRE.....	53
■ La visión probabilista de la incertidumbre.....	55
■ La incertidumbre mediante factores de certeza.....	68
■ Tratar con el conocimiento difuso.....	70
APRENDIZAJE.....	75
■ Aprendizaje supervisado.....	78
■ Redes neuronales artificiales.....	83
■ Aprendizaje no supervisado y aprendizaje reforzado.....	88
CONSIDERACIONES FINALES.....	93
GLOSARIO DE TÉRMINOS.....	95
REFERENCIAS.....	99





# INTRODUCCIÓN

---

Este libro es una publicación científica, didáctica, metodológica y educativa, destinada a educadores en general y otros profesionales. Es el tercero de cuatro libros que tienen como objetivo principal propiciar la incorporación de conceptos e ideas generales sobre Inteligencia Artificial en las escuelas, familias y comunidades. Este tercer libro va dirigido a las temáticas específicas que aborda la Inteligencia Artificial (IA) desde la ciencia. Para su estudio se han organizado en cuatro núcleos básicos de conocimiento: La representación del conocimiento, el razonamiento, el tratamiento de la incertidumbre y el aprendizaje.

Los capítulos responden a cada uno de los cuatro núcleos de conocimiento. El primer capítulo expone los elementos generales del razonamiento computacional y algunas de las estrategias específicas de la IA. El segundo capítulo aborda algunas de las formas de representación del conocimiento que necesitan los sistemas de IA. El tercer capítulo muestra criterios para el tratamiento de la incertidumbre presente en los problemas que resuelve la IA. El cuarto capítulo presenta los elementos generales del aprendizaje y los tres tipos de aprendizaje automático que se reconocen en la bibliografía

especializada. Finalmente, se pone a disposición de los lectores un Glosario de Términos y la Bibliografía.

Antes de exponer las particularidades de cada núcleo de conocimiento de la IA, se hace necesario una aclaración. La Inteligencia Artificial, como Ciencia de la Computación, se encarga de llevar a un sistema de cómputo un conjunto de órdenes para que este logre un comportamiento racional. Llevar al sistema esas órdenes requiere, primeramente, llevarla al plano mental (humano), organizarlo y plantearlo de manera que sea posteriormente “entendido” por el sistema de cómputo. De ahí la importancia de conocer y especificar de antemano, antes incluso de plantearlo en un lenguaje específico para llevar a la computadora, cómo representar el conocimiento que queremos introducir al sistema, cómo se procederá para resolver el problema que se desea, qué situaciones se encontrará el sistema que puedan hacerlo dudar de su decisión y cómo lidiar con ello y por último cómo podrá mejorar su desempeño en el tiempo.

Defendemos la idea que no es necesario conocer de programación para tener nociones básicas de formulación de problemas y su solución desde una visión de Inteligencia Artificial. Sería útil que personas de diversas áreas tengan estos conocimientos generales para identificar vías de solución a sus problemas apoyados en la Inteligencia Artificial. Mucho más importante es que estas nociones vayan introduciéndose en el pensamiento de los niños. Justo aclarar que solo con formular el problema y su solución general, no será posible construir el sistema, esa sí es tarea de los especialistas y programadores de sistemas de cómputo.

Como antecedentes de esta publicación se elaboró en la Universidad de las Ciencias Informáticas (UCI) un documento base, el cual fue distribuido a especialistas, profesores, investigadores y otros profesionales relacionados con la Inteligencia Artificial. En él



se brindan las ideas generales de posibles temáticas a abordar en edades tempranas, las cuales pudieran ser formalizadas en la educación general o tratadas como temas complementarios en diversos espacios. De igual forma, como parte del IV Taller Internacional de Enseñanza de las Ciencias Informáticas en el marco de la IV Conferencia Científica Internacional de la Universidad de las Ciencias Informáticas UCIENCIA 2021, se efectuó un panel de expertos donde participaron como panelistas, reconocidos profesores e investigadores de IA en Cuba. Todos los criterios fueron tomados en cuenta para enriquecer las ideas que aquí se exponen.

Aprovechamos la ocasión para expresar nuestro agradecimiento a todos los que aportaron opiniones, valoraciones y sugerencias para mejorar esta obra. Sobre todo al grupo de investigación de Inteligencia Artificial de la Universidad de las Ciencias Informáticas, al grupo de debate sobre ética y robótica de la Oficina Regional de la UNESCO en La Habana, integrado por desarrolladores y otros especialistas de los Joven Club de Computación y Electrónica, universidades y la propia oficina. De igual forma al colectivo de la disciplina Inteligencia Computacional de la carrera Ingeniería en Ciencias Informáticas de la UCI.

Esta obra pretende servir de base para contribuciones teóricas y prácticas mediante el desarrollo de actividades educativas curriculares y no curriculares para las instituciones educativas, escuelas, universidades, familias, comunidades y la sociedad en general, así como para las instituciones y organizaciones gubernamentales y no gubernamentales; y de todos los sectores e instituciones de la sociedad. No se trata de enseñar a desarrollar aplicaciones de Inteligencia Artificial, sino, de contribuir al desarrollo del pensamiento computacional y nociones básicas necesarias para incentivar el estudio y profundización en IA con un posicionamiento ético, crítico y creativo,

## INTRODUCCIÓN

desde edades tempranas. Finalmente los exhortamos a leer, estudiar y profundizar en los aspectos que se abordan, para contribuir entre todos al desarrollo de la Inteligencia Artificial como componente esencial para la informatización de la sociedad.

---

# EL RAZONAMIENTO COMPUTACIONAL

---

El razonamiento computacional puede verse como el proceso general llevado a cabo por el sistema para dar solución a un problema. Digamos que seguir un algoritmo general, aplicar métodos generales utilizando estrategias diversas. El término razonamiento computacional está relacionado directamente con el objeto de estudio de la IA, es decir construir sistemas con comportamiento racional. Llamarlo razonamiento implica asumir que el comportamiento del sistema será racional, o sea, encontrará la mejor solución posible a partir del conocimiento que tenga de su entorno. Con estos criterios se puede asumir que:

*El **razonamiento**, desde el punto de vista computacional, es el proceso general llevado a cabo por un sistema para comportarse de manera racional a partir del conocimiento que tenga de su entorno.*

La inteligencia artificial trata de resolver problemas que no tienen una solución algorítmica conocida, o esta es tan compleja que requeriría mucho tiempo para ser resuelta por la computadora, digamos, horas, días o incluso años. Muchos problemas de la vida

real pueden caer en esta clase de problemas. En la IA se reconoce una forma de razonamiento general para dar solución a este tipo de problemas. Los métodos de solución de problemas de IA aplican estrategias de búsqueda de soluciones. Pongamos un ejemplo muy utilizado en los libros de IA.

Suponga que está en París sin un mapa y no habla francés. ¿Cómo llegar hasta la torre Eiffel? Una primera alternativa podría ser tomar aleatoriamente una calle, con la esperanza de que en algún momento se llegará a la torre. Esta búsqueda aleatoria puede llevar a encontrar la torre, pero puede requerir una cantidad infinita de tiempo por la forma arbitraria en la cual se selecciona un camino, incluso pudiera ocurrir que el mismo camino lo tomáramos muchas veces.

Otra alternativa es seguir una a una cada calle. Cuando se alcanza un final se busca una calle paralela y se sigue en dirección opuesta, independientemente de si nos acercamos o alejamos del objetivo. Esta variante de búsqueda a ciegas consideraría todas las variantes posibles. Generalmente se llegará a la solución, pero el costo puede ser muy alto, podemos demorar varios días en llegar a la Torre.

La tercera alternativa utiliza cierto conocimiento sobre el problema para encontrar su solución y mejorar la eficiencia de la búsqueda. Suponiendo que el extremo superior de la torre puede ser visto desde cualquier lugar, podemos tomar la calle que parezca nos lleve en esa dirección. Esta búsqueda dirigida no garantiza encontrar el mejor camino, pero sí nos va guiando para reducir el proceso de búsqueda.

La IA se centra en las búsquedas a ciegas y la búsqueda dirigida, también conocida como búsqueda informada o heurística. Algunos métodos utilizan elementos de aleatoriedad, pero siempre dirigidos a complementar las heurísticas utilizadas.

## La formulación de problemas de estado

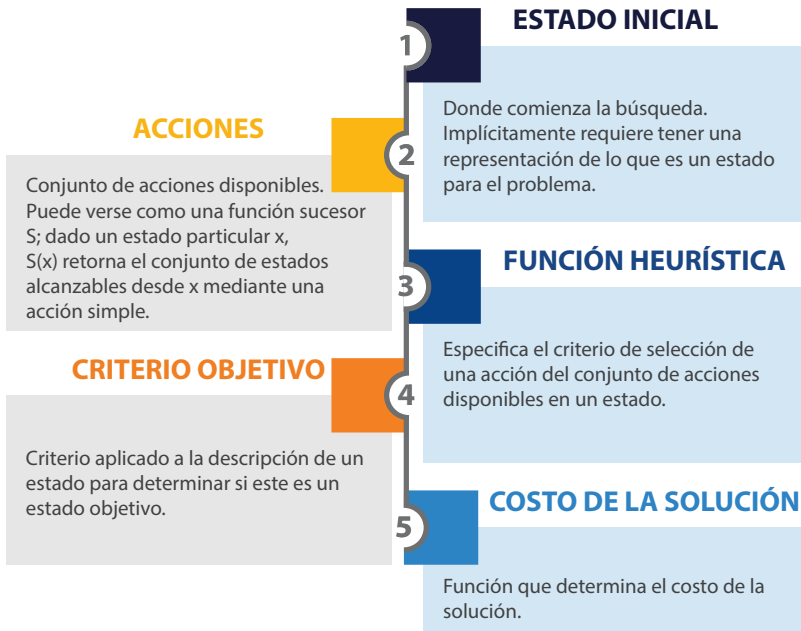
Para dar solución a cualquier problema en la vida, lo primero es tener bien claro ese problema. De ahí que la tarea, en este caso, es formularlo de manera que pueda ser resuelto siguiendo un conjunto de pasos. Derivados de la teoría de autómatas, estos métodos parten de definir una representación del problema en forma de estados. A partir de un estado inicial y aplicando un conjunto de acciones, se va transitando de un estado a otro hasta encontrar la solución al problema.

Para la formulación de un problema de estados es necesario identificar los siguientes elementos:

- **ESTADO:** Información que se tiene en un momento determinado sobre el problema.
- **ESTADO INICIAL:** Donde comienza el análisis.
- **ACCIONES:** Descripción de cómo y hacia qué otro estado se llegará, desde un estado particular. Implícitamente el estado inicial y las acciones definen el **espacio de estados** del problema (conjunto de todos los estados alcanzables desde el estado inicial). El espacio de estados forma un grafo en el cual los nodos son estados y los arcos entre los nodos son acciones. Un **camino** en el espacio de estados es una secuencia de estados conectados por una secuencia de acciones.
- **HEURÍSTICA:** Criterio de selección de una acción.
- **CRITERIO OBJETIVO:** Criterio aplicado a la descripción de un estado para determinar si ya se llegó a una solución. Algunas veces, hay un conjunto explícito de posibles objetivos y el criterio simplemente consiste en revisar si se ha alcanzado uno de ellos.
- **COSTO DE LA SOLUCIÓN:** Valor de cuánto costó llegar a la solución. El caso más simple asume que el costo de una acción

individual es 1, así que el costo de la solución es el número de pasos seguido para llegar a la solución.

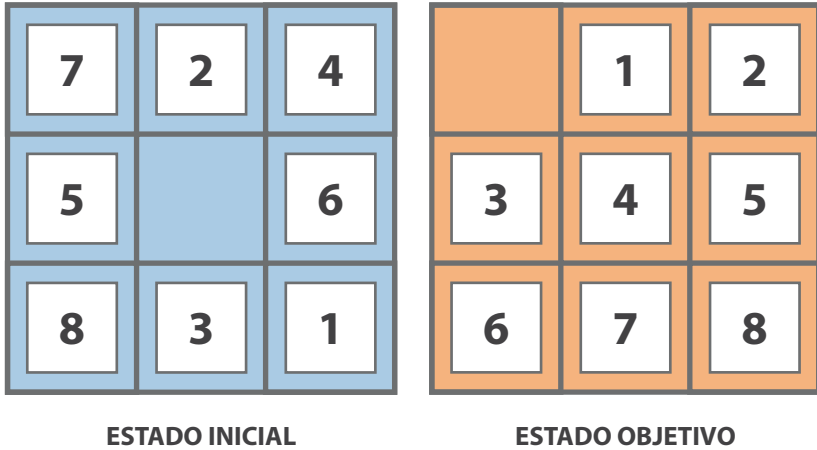
Los elementos anteriores definen un problema de búsqueda en IA (Figura 1). Una solución de un problema puede ser un camino desde el estado inicial a un estado objetivo o un estado al que se ha llegado. La calidad de la solución se mide por el valor del costo de la solución, y una solución óptima tiene el costo más pequeño de entre todas las soluciones posibles. A continuación se especifican algunos ejemplos de problemas que se pueden formular siguiendo estas ideas.



**Figura 1.** Elementos necesarios para modelar un problema de estados.

## Problema de las 8 piezas

Consiste en un tablero de 3 X 3 con ocho fichas numeradas y un espacio en blanco. Una ficha adyacente al espacio en blanco puede deslizarse a este. La meta es alcanzar el estado objetivo especificado (Figura 2).



**Figura 2.** Ejemplo del problema de las 8 piezas.

La formulación estándar es como sigue:

**ESTADOS:** La descripción de un estado especifica la localización de cada una de las ocho fichas y el blanco en cada uno de los nueve cuadrados.

**ESTADO INICIAL:** Cualquier estado puede ser un estado inicial.

**ACCIONES:** Genera los estados legales que resultan de aplicar las cuatro acciones (mover el blanco a la Izquierda, Derecha, Arriba y Abajo).

**CRITERIO OBJETIVO:** Comprueba si el estado coincide con la configuración objetivo.

**COSTO DEL CAMINO:** El costo de cada paso del camino tiene valor 1, así que el costo del camino es el número de pasos.

Este tipo de problemas de piezas deslizantes, a menudo se utilizan como problemas de prueba para los nuevos algoritmos de IA. Estos problemas no tienen, por lo general, métodos significativamente mejores que los algoritmos de búsqueda tradicionales.

Esta variante con 8 piezas tiene 181 440 estados alcanzables y se resuelve fácilmente con los métodos tradicionales de IA. El de 15 piezas (sobre un tablero de 4x4) tiene alrededor de 1,3 trillones de estados y puede resolverse óptimamente en pocos milisegundos por los mejores algoritmos de búsqueda. El de 24 piezas (sobre un tablero de 5 x 5) tiene alrededor de  $10^{25}$  estados, y sus configuraciones aleatorias son difíciles de resolver de manera óptima con computadores y algoritmos comunes.

## Problema de las 8 reinas

Otro ejemplo es el problema de las 8 reinas. Su objetivo es colocar ocho reinas en un tablero de ajedrez de manera que cada reina no ataque a ninguna otra. (Una reina ataca alguna pieza si está en la misma fila, columna o diagonal).

Aunque existen algoritmos eficientes específicos para este problema y para el problema general de las  $n$  reinas, sigue siendo un problema interesante para los algoritmos de búsqueda. Existen dos formulaciones principales para resolver este problema. Una formulación incremental que utiliza operadores que aumentan la descripción del estado, comenzando con un estado vacío; para el problema de las 8 reinas, esto significa que cada acción añade una reina al estado. La otra formulación utiliza los estados completos, es decir,



comienza con las ocho reinas en el tablero y las mueve para buscar la configuración deseada. En cualquier caso, el coste del camino no tiene ningún interés porque solamente cuenta el estado final. La primera formulación incremental que se puede realizar es la siguiente:

**ESTADOS:** Cualquier combinación de cero a ocho reinas en el tablero es un estado.

**ESTADO INICIAL:** Ninguna reina sobre el tablero.

**ACCIONES:** Añadir una reina a cualquier cuadrado vacío.

**CRITERIO OBJETIVO:** Ocho reinas sobre el tablero, ninguna es atacada.

En esta formulación, se tienen aproximadamente  $3 \times 10^{14}$  posibles combinaciones a investigar.

Una mejor formulación sería no colocar reina en un cuadrado que esté atacado:

**ESTADOS:** La combinación de  $n$  reinas ( $0 \leq n \leq 8$ ), una por columna desde la columna más a la izquierda, sin que una reina ataque a otra.

**ACCIONES:** Añadir una reina en cualquier cuadrado en la columna más a la izquierda vacía tal que no sea atacada por cualquier otra reina.

Esta formulación reduce el espacio de estados de las 8 reinas de  $3 \times 10^{14}$  a 2 057, y las soluciones son fáciles de encontrar. Para 100 reinas la formulación inicial tiene 10400 estados, mientras que las formulaciones mejoradas tienen cerca de 1052 estados. Esta es una reducción enorme, pero el espacio de estados sigue siendo demasiado grande.

## El mundo de Wumpus

Un ejemplo de problema también de juego, pero representando elementos de “posible realidad”, es El mundo de Wumpus. Este juego representa una cueva compuesta por habitaciones conectadas mediante pasillos. Escondido en algún lugar de la cueva está el Wumpus, una bestia que se come a cualquiera que entre en su habitación. El Wumpus puede ser derribado por la flecha de un agente, y éste sólo dispone de una. Algunas habitaciones contienen hoyos sin fondo que atrapan a aquel que deambula por dichas habitaciones (menos al Wumpus, que es demasiado grande para caer en ellos). El premio en este entorno es la posibilidad de encontrar un tesoro. Aunque el mundo de Wumpus pertenece más al ámbito de los juegos por computador, es un entorno perfecto de juegos de mesa (Figura 3).














4	 Hedor		 Brisa	 Hoyo
3		  ORO	 Hoyo	 Brisa
2	 Hedor		 Brisa	
1	INICIO	 Brisa	 Hoyo	 Brisa
	1	2	3	4

Figura 3. Ejemplo del mundo de Wumpus.

La principal dificultad para el agente es su ignorancia inicial acerca de la configuración del entorno; para superar esta ignorancia el agente requerirá razonamiento lógico. Una formalización del problema sería como sigue:

**ESTADOS:** Una configuración sencilla se limitará a la posición del agente. Sin embargo para resolver el problema será necesario estados más complejos, que incluyan el conocimiento que tiene el agente de su entorno.

**ESTADO INICIAL:** Habitación de inicio de la cueva.

**ACCIONES:** Devuelve una acción de movimiento a una habitación contigua a la actual. Esta función debe tomar en cuenta las consecuencias de cada posible movimiento.

**CRITERIO OBJETIVO:** Puede ser un objetivo complejo: matar al wumpus y obtener el premio, o asumir solo uno de ellos.

**COSTO DEL CAMINO:** El costo en sí, puede ser solo la cantidad de pasos dados en la cueva.

Como puede verse este problema tiene variantes de formulación que dependerán de la especificación que se desee y los objetivos a lograr.

## El problema de las rutas

Quizás el ejemplo práctico más utilizado en IA es el problema de búsqueda de una ruta. Este está definido en términos de posiciones y transiciones a lo largo de ellas. Los algoritmos de búsqueda de rutas se han utilizado en una variedad de aplicaciones, tales como rutas en redes de computadores, planificación de operaciones militares, en sistemas de planificación de viajes de líneas aéreas, entre muchas otras. Estos problemas pueden ser muy sencillos de especificar si se simplifican las condiciones, pero en la realidad son muy complejos.

Consideremos un ejemplo simplificado de un problema de viajes de líneas aéreas:

**ESTADOS:** Cada estado está representado por una localización (por ejemplo, un aeropuerto, una ciudad).

**ESTADO INICIAL:** Una localización especificada por el problema.

**ACCIONES:** Tomar un camino desde el estado actual a otro.

**CRITERIO OBJETIVO:** Destino especificado por el problema.

**COSTO DEL CAMINO:** Depende del costo especificado en el problema (Puede ser dinero, tiempo de viaje, kilómetros recorridos, gasto de combustible, etc.).

Dada la formulación de un problema, la próxima acción es encontrar una solución, lo cual consiste en generar nuevos estados a partir del estado donde se encuentra el agente. Este proceso se denomina expandir el estado. La expansión puede producir uno o varios nuevos estados. Cuando hay un solo estado posible, se toma este y se continúa, cuando hay varias opciones para continuar la búsqueda, es necesario tomar una decisión para seleccionarlo.

Esta es la esencia de la búsqueda, seleccionar una opción y poner las otras alternativas a un lado para retomarlas más tarde, si la primera selección no conduce a una solución. La selección de cuál estado expandir primero se determina por la estrategia de búsqueda.

La búsqueda consiste en ir construyendo un espacio de búsqueda (la parte del espacio de estado que deja de ser abstracta). La raíz del espacio de búsqueda se corresponde con el estado inicial. Los nodos hojas del árbol corresponden a estados que no tienen sucesores en el árbol, porque no han sido expandidos o porque no tienen sucesores. En cada paso, el algoritmo de búsqueda selecciona un nodo hoja por expandir.

## Búsqueda a ciegas

La búsqueda a ciegas es aquella donde no existe ninguna información para decidir qué nodo expandir, no se conoce la cantidad de pasos o el costo del camino desde el estado actual hasta el objetivo.

Dentro de esta, la búsqueda exhaustiva examina el espacio de estado completamente, de una manera ordenada, usando todos los operadores y generando todos los sucesores posibles para encontrar la solución deseada. Los dos métodos básicos de esta búsqueda son la búsqueda primero a lo ancho y la búsqueda primero en profundidad.

Una búsqueda **primero a lo ancho** comienza por generar todos los sucesores de la raíz del nodo. Luego, la búsqueda continúa expandiendo todos los nodos de cada nivel. La esencia del método es examinar todas las soluciones potenciales a un nivel antes de pasar al nivel inferior (Figura 4).

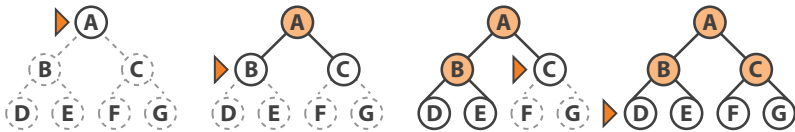


Figura 4. Búsqueda primero a lo ancho.

La búsqueda **primero en profundidad** procede generando primeramente un sucesor del nodo raíz y luego un sucesor de este, y así continúa extendiendo este camino hasta que termina (Figura 5). El inconveniente de la búsqueda primero en profundidad es que puede hacer una elección equivocada y obtener un camino muy largo (o infinito) aun cuando una elección diferente llevaría a una solución cerca de la raíz del árbol de búsqueda.

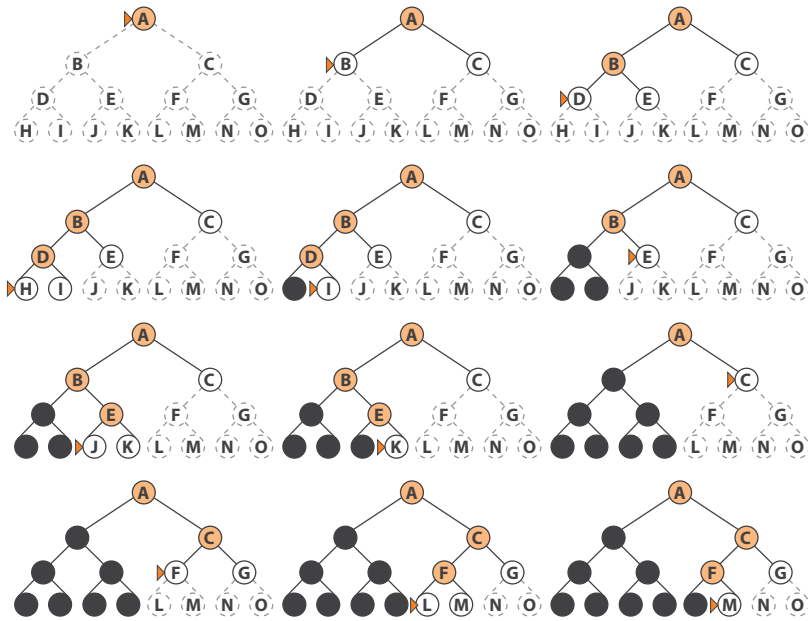


Figura 5. Búsqueda primero en profundidad.

## Búsqueda heurística

Una heurística es un criterio que evalúa la posibilidad de que una búsqueda vaya en la dirección correcta. Los métodos de búsqueda heurísticos están orientados a reducir la cantidad de búsqueda requerida para encontrar una solución. Cuando un problema es presentado como un árbol de búsqueda, el enfoque heurístico intenta reducir el tamaño del árbol cortando nodos poco prometedores.

Estos métodos se llaman métodos fuertes, al incorporar conocimiento en la búsqueda de la solución. La heurística usada en un problema dado, puede no ser totalmente aplicable o ser inaplicable en otro dominio o tarea. Generalmente, los métodos de búsqueda

heurísticos se basan en maximizar o minimizar algunos aspectos del problema.

Suponga que un hombre se encuentra en una extensa llanura y tiene sed, en ese momento ha llegado a una pequeña elevación que es la única en esa región y se sube a ella. Desde la elevación, el hombre observa lo siguiente:

**NORTE:** Vegetación amarilla.

**SUR:** Vegetación verde y movimiento de animales.

**ESTE:** Vegetación amarilla.

**OESTE:** Vegetación verde.

La vegetación verde es un indicio de que hay humedad, por lo que es muy probable que exista agua en la superficie o subterránea. El movimiento de animales puede indicar que ellos se dirigen allí buscando agua, lo cual sugiere que el agua está en la superficie. Con esta información el hombre decide dirigirse al sur, esto constituye una heurística.

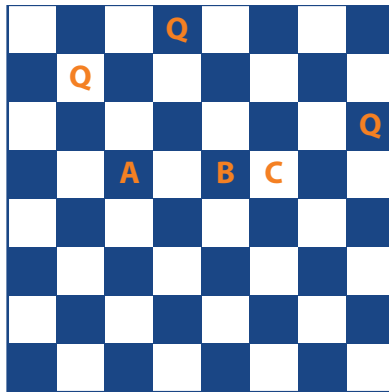
La heurística no garantiza que siempre se tome la dirección de la búsqueda correcta, por eso este enfoque no es óptimo sino suficientemente bueno. Frecuentemente, son mejores los métodos heurísticos que los métodos de búsqueda a ciegas, aunque, muchas veces no se llega a la mejor solución existente. Muchas veces la brecha entre la solución óptima y una generada por heurísticas puede ser grande.

Hay una familia de algoritmos denominada *Primero el mejor* que utilizan una heurística que se basa en el costo estimado del camino menos costoso desde un nodo cualquiera a un nodo objetivo. La búsqueda primero el mejor trata de expandir el nodo más cercano al objetivo, alegando que probablemente conduzca rápidamente a una solución. Así, evalúa los nodos utilizando el valor heurístico. Para ilustrar mejor veamos cómo funciona este método en algunos de los ejemplos formulados previamente.

## El problema de las ocho reinas

Asumiendo la variante incremental del juego de las 8 reinas, poner las reinas una a una de modo que no se ataquen mutuamente, supóngase que se tienen situadas tres reinas y tenemos que decidir dónde colocar la cuarta (Figura 6). El papel de la heurística aquí será ofrecer un criterio para decidir cuál de las tres posiciones indicadas es la mejor para conducir a una solución satisfactoria.

Al deducir una heurística para este problema podemos razonar que para poder colocar las ocho reinas tenemos que dejar libres la mayor cantidad de opciones que sea posible para futuras adiciones de reinas. Esto significa determinar la cantidad de casillas en las filas no utilizadas que quedarán no atacadas al colocar la cuarta reina en A, B o C. Una casilla candidata será preferida si deja la mayor cantidad de casillas no atacadas en el resto del tablero. Consecuentemente, el número de casillas no atacadas constituye una medida de su mérito, es decir, es la función de evaluación heurística para este problema.



**Figura 6.** Ejemplo de situación del juego de las 8-reinas.



Al calcular la función heurística para A, B y C vemos que la mejor opción es C:

$$f(A) = 8$$

$$f(B) = 9$$

$$f(C) = 10$$

Otra alternativa o enfoque heurístico es tomar en cuenta las casillas no atacadas de las filas no usadas. Aquellas con pocas casillas no atacadas tienden a quedar bloqueadas más rápidamente que las filas con muchas casillas no atacadas. Consecuentemente, si queremos minimizar la posibilidad de un futuro bloqueo, debemos fijarnos en la cantidad de casillas no atacadas de la fila con menor número de casillas no atacadas. Para esta función tenemos:

$$f'(A) = 1$$

$$f'(B) = 1$$

$$f'(C) = 2$$

Con esta heurística C también es la alternativa preferida.

## El problema de las 8 piezas

Como se vio antes, el objetivo de las piezas es deslizar las fichas horizontalmente o verticalmente al espacio vacío, hasta que la configuración empareje con la configuración objetivo.

En el reordenamiento sólo se permite desplazar fichas a la casilla vacía, siempre que sea adyacente, es decir la regla es:

- Una pieza puede moverse de la posición X a la Y, si la posición X es adyacente a Y y la posición Y está vacía.

Una solución típica para este problema requiere alrededor de veinte pasos, aunque este número varía dependiendo del estado inicial. El factor de ramificación es aproximadamente tres

(cuando la casilla vacía está en el centro hay cuatro movimientos posibles, y cuando está en una esquina hay dos). Esto significa que una búsqueda exhaustiva a profundidad veinte podría requerir cerca de  $3^{20} = 3.5 * 10^9$  estados. Eliminando los estados repetidos quedarían  $9! = 362880$  ordenamientos diferentes. Esta es aún una cantidad grande de estados, por lo que se requiere encontrar una buena función heurística.

Un razonamiento para construir la función heurística para este problema es estimar qué tan cerca está un estado del objetivo. Hay dos variantes comúnmente usadas para estimar la proximidad de un estado a otro:

$h_1$  = número de piezas mal colocadas. Para la Figura 2, las 8 piezas están fuera de su posición, así que el estado inicial tiene  $h_1 = 8$ .

$h_2$  = suma de las distancias de las piezas a sus posiciones en el objetivo. Como las piezas no pueden moverse en diagonal, la distancia que contaremos será la suma de las distancias horizontales y verticales. Esta medida es conocida como distancia de Manhattan. Esta también es admisible, porque cualquier movimiento que se puede hacer es mover una pieza un paso más cerca del objetivo. Las piezas 1 a 8 en el estado inicial nos dan una distancia de Manhattan de

$$h_2 = 3+1+2+2+2+3+3+2=18$$

Como era de esperar, ninguna sobrestima el coste solución verdadero, que es 26.

Las heurísticas  $h_1$  (piezas mal colocadas) y  $h_2$  (distancia de Manhattan) son heurísticas bastante buenas para 8 piezas. Tanto  $h_1$  como  $h_2$  son estimaciones de la longitud del camino restante para este problema.

## El problema de las rutas

Asumamos que para el problema de las rutas se brinda un mapa con las carreteras existentes entre ciudades, queremos encontrar el camino más corto entre las ciudades A y B, conociendo las longitudes de las carreteras entre las ciudades. Estas distancias las podemos colocar en una tabla de modo que  $d(i,j)$  indica la distancia por carretera entre las ciudades  $i$  y  $j$ .

Como no tenemos la distancia de la Ciudad A a la B que es nuestro objetivo, se puede estimar la distancia directa en el mapa entre dos ciudades, por ejemplo, la distancia aérea (en línea recta) de D a B de C a B y así con cada ciudad hasta nuestro objetivo. Las menores distancias aéreas serán las mejores candidatas a tomar de escala en el camino hacia B.

Esta información se puede representar en una tabla adicional con la distancia aérea entre las ciudades. Esta tabla  $h(i)$  contiene la distancia aérea de la ciudad  $i$  al objetivo B.

A la forma más ampliamente conocida de la búsqueda primero el mejor se le llama búsqueda  $A^*$  (pronunciada «búsqueda A-estrella»). Evalúa los nodos combinando  $g(n)$ , el costo para alcanzar el nodo, y  $h(n)$ , el costo estimado de ir al nodo objetivo:  $f(n)=g(n)+h(n)$ .

Ya que  $g(n)$  representa el costo del camino desde el nodo inicio al nodo  $n$ , y  $h(n)$  el coste estimado del camino más barato desde  $n$  al objetivo, tenemos:

$f(n)$ = costo más barato estimado de la solución a través de  $n$ .

Así, si tratamos de encontrar la solución más barata, es razonable intentar primero el nodo con el valor más bajo de  $g(n)+h(n)$ . Resulta que esta estrategia es muy razonable; siempre que la función heurística  $h(n)$  satisfaga ciertas condiciones, la búsqueda  $A^*$  es completa y óptima. Para ilustrar pongamos un ejemplo.

Juan quiere recorrer un laberinto. Él comienza en la entrada A y debe terminar en la salida F. Los senderos en el laberinto son unidireccionales. Esto significa que si se llega desde A hasta B por un sendero no se puede regresar a A desde B por el mismo sendero. Juan consume un tiempo de caminata determinado por cada sendero de acuerdo a la longitud del mismo. Además, antes de entrar en el laberinto, a Juan le dan una medida estimada del tiempo que tardará en salir desde cada lugar (MET). Entre cada par de lugares hay, a lo sumo, un sendero unidireccional que los conecta directamente.

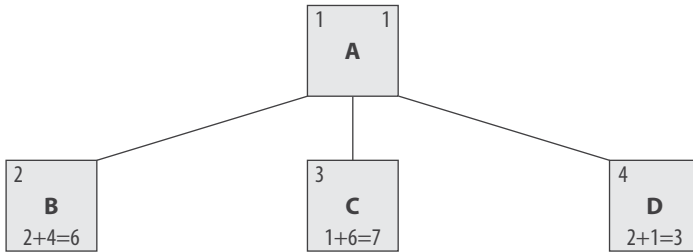
En la Tabla 1 se muestra la relación de lugares y las conexiones unidireccionales entre los mismos.

**Tabla 1.** Datos del problema de ejemplo para métodos de búsqueda

LUGARES	MET	SE LLEGA A	TIEMPO
A	2	B	4
		C	6
		D	1
B	2	C	5
		E	5
C	1	F	3
		G	4
D	2	B	7
		E	6
E	1	F	3
F	0	-	-
G	10	-	-

Aplicando el método A\* vemos en la Figura 7 que se parte de A, se generan los tres nodos alcanzables desde este (B, C y D), se especifica que es el primer nodo generado y el primer nodo expandido.

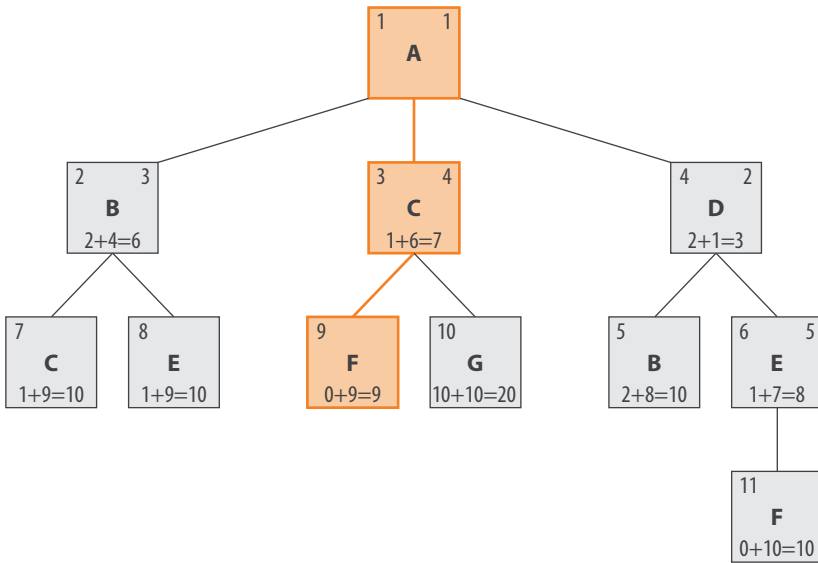
Al analizar B se observa que el valor ofrecido de inicio de cuánto le demorará para llegar al final es 2 (heurística), mientras que el tiempo de moverse de A hasta B es 4. Por lo que el valor de la función heurística es 6.



**Figura 7.** Primer nivel del árbol de solución del ejemplo de problema de búsqueda.

De igual forma se analiza C y D. Al analizar los tres valores obtenidos de funciones heurísticas se observa que D tiene el menor valor (3), lo que quiere decir que sería el nodo a través del cual se demoraría menos tiempo en llegar a la solución. Se marca como segundo expandido y se generan los nodos B y E (Figura 8), accesibles desde D.

Analizando todos los nodos hojas hasta el momento: B (desde A), C, B (desde D) y E, se sigue el proceso para ir generando todo el árbol de solución hasta llegar a un punto donde el nodo hoja con menor valor de función heurística sea F. En ese momento se determina el camino para llegar desde A hasta F (A - C - F).



**Figura 8.** Árbol de solución del ejemplo de algoritmos de búsqueda.

## Búsquedas locales

Los métodos vistos hasta aquí se encargan de explorar todo el espacio de búsqueda en cada momento. Esta forma sistemática se logra manteniendo uno o más caminos en memoria y registrando qué alternativas se han explorado en cada punto a lo largo del camino y cuáles no. Cuando se encuentra un objetivo, el camino a ese objetivo también constituye una solución al problema.

Existe un conjunto de problemas para los cuales el camino que conduce a la solución no es importante. Esto ocurre, generalmente, en problemas donde la descripción del estado contiene toda la información necesaria para una solución. La idea es comenzar con

una configuración completa y hacer modificaciones para mejorar su calidad. Esta clase de problemas es muy utilizada en aplicaciones para el diseño de circuitos integrados, disposición del suelo de una fábrica, programación del trabajo en tiendas, programación automática, optimización de redes de telecomunicaciones, dirigir un vehículo, entre muchas otras.

En el juego de las ocho reinas, con la formulación que parte de una distribución de estas sobre el tablero y se van moviendo tratando de reducir el número de ataques, no es necesario conocer los pasos dados, sino la configuración final del tablero. Si no importa el camino al objetivo, podemos considerar una clase diferente de algoritmos que no se preocupen de los caminos. Los algoritmos de búsqueda local funcionan con un solo estado actual y generalmente se mueve sólo a los vecinos del estado actual. Aunque los algoritmos de búsqueda local no son sistemáticos, muchas veces pueden encontrar soluciones razonables en espacios de estados grandes o infinitos (continuos) para los cuales son inadecuados los algoritmos sistemáticos.

Además de encontrar los objetivos, los algoritmos de búsqueda local son útiles para resolver problemas de optimización, en los cuales el objetivo es encontrar el mejor estado según una función objetivo.

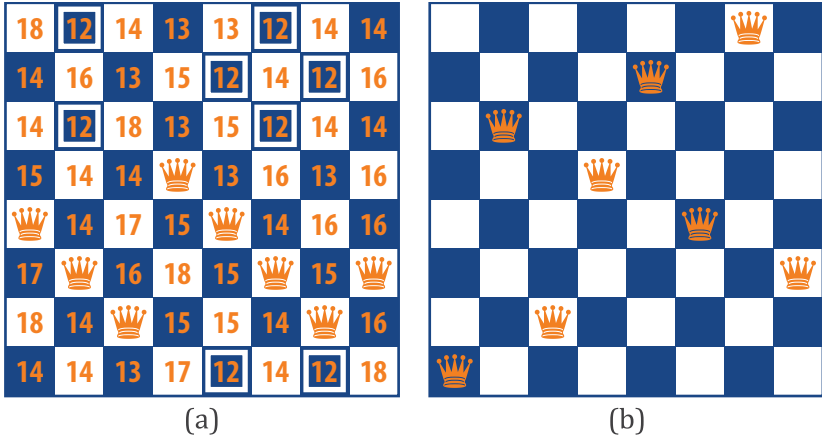
Uno de los métodos más conocidos para este tipo de problemas es el algoritmo de búsqueda de ascensión de colinas. Aunque tiene variantes que modifican ciertas partes del algoritmo, el más básico va buscando si alguno de los sucesores del estado actual es mejor que este, si lo encuentra lo toma como estado actual. Una mejora de este algoritmo analiza todos los sucesores del estado actual y escoge el mejor para hacerlo estado actual.

Este algoritmo no mantiene un árbol de búsqueda, sino una estructura de datos del nodo actual que necesita sólo el registro del estado y su valor de función objetivo. La ascensión de colinas no mira más allá de los vecinos inmediatos del estado actual.

Para ilustrar la ascensión de colinas, usaremos el problema de las 8 reinas. Los algoritmos de búsqueda local típicamente usan una formulación de estados completa. En este ejemplo, cada estado tiene a ocho reinas sobre el tablero, una por columna. Una acción devuelve el estado generado al mover una reina a otro cuadrado en la misma columna. De ahí que cada estado tiene  $8$  (cantidad de reinas)  $\times$   $7$  (cantidad restante de casillas en la misma columna) =  $56$  sucesores. La función heurística  $h$  es el número de pares de reinas que se atacan la una a la otra, directa o indirectamente. El mínimo global de esta función es cero, que ocurre en soluciones perfectas, es decir, cuando ninguna reina ataca a otra. La Figura 9 (a) muestra un estado con  $h = 17$  ya que hay 17 pares de reinas que se atacan. La figura también muestra los valores de todos sus sucesores obtenidos al mover una reina dentro de su columna. Se puede ver que los mejores valores son de  $12$ , por ejemplo, para el estado sucesor donde la reina de la segunda columna está en la primera o tercera fila. Los algoritmos de ascensión de colinas eligen típicamente al azar entre el conjunto de los mejores sucesores, si hay más de uno.

La Figura 9 (b) muestra un mínimo local en el espacio de estados; el estado tiene  $h = 1$  ya que hay solo un par de reinas que se atacan, pero cada sucesor tiene un coste más alto. Esta es una muestra de un resultado final del algoritmo que no brinda una solución al problema.





**Figura 9.** Ejemplo de búsqueda para el problema de las 8 reinas.

A veces a la ascensión de colinas se le llama búsqueda local voraz porque toma un estado vecino bueno sin pensar hacia dónde ir después. La ascensión de colinas a menudo hace el progreso muy rápido hacia una solución, porque es por lo general bastante fácil mejorar un estado malo. Por ejemplo, desde el estado de la Figura 9 (a), se realizan solamente cinco pasos para alcanzar el estado de la Figura 9 (b), que tiene  $h = 1$  y es casi una solución.

Se han inventado muchas otras variantes de la ascensión de colinas. La ascensión de colinas estocástica escoge aleatoriamente de entre los movimientos ascendentes; la probabilidad de selección puede variar con la pendiente del movimiento ascendente. Éste por lo general converge más despacio que la subida más rápida, pero en algunos casos encuentra mejores soluciones. La ascensión de colinas de primera opción implementa una ascensión de colinas estocástica generando sucesores al azar hasta que se genera uno que es mejor que el estado actual. Esta es una buena estrategia cuando un estado tiene muchos (por ejemplo, miles) sucesores. Estos algoritmos de

ascensión de colinas son incompletos, a menudo dejan de encontrar un objetivo, cuando éste existe, debido a que pueden estancarse sobre máximos locales.

## Otras formas de razonamiento en IA

Las formas de razonamiento en la IA son mucho más diversas que los ejemplos más clásicos mostrados aquí. Algunos métodos de razonamiento general en IA van dirigidos a resolver problemas específicos, otros utilizan estrategias de búsqueda de conjuntos de soluciones, generalmente inspirados en la naturaleza. Los algoritmos que parten de una población de soluciones candidatas llevan a cabo un proceso de optimización de dichas soluciones. Dentro de los más reconocidos están los algoritmos genéticos, las colonias de hormigas y las bandadas de pájaros.

Otro tipo de problema específico, pero muy popular en IA, es el que va dirigido a entornos competitivos, en los cuales los objetivos del agente están en conflicto. Estos problemas son denominados de búsqueda entre adversarios, a menudo conocidos como juegos. La teoría matemática de juegos, asume a cualquier entorno multiagente como un juego, siempre que el impacto de cada agente sobre los demás sea significativo, sin tener en cuenta si los agentes son cooperativos o competitivos (Russell & Norvig, 2004). Por otro lado la industria del videojuego tiene hoy una gran aliada en la IA, ya que aplica muchos de los resultados de ella en lograr el comportamiento inteligente de los personajes.

Sin embargo, tradicionalmente en IA se han considerado un tipo muy específico de juegos para ser resueltos con los métodos de búsqueda. Son nombrados de diversas formas: juegos de suma cero, de dos jugadores, por turnos, determinista o de información perfecta.

Se puede decir que son entornos deterministas, totalmente observables, donde hay dos agentes y sus acciones deben alternar. En estos agentes los valores de utilidad, al final del juego, son siempre iguales y opuestos. Por ejemplo, si un jugador gana un juego de ajedrez (+1), el otro jugador necesariamente pierde (-1).

En esta clase de juegos intervienen dos jugadores con información completa. Hay dos jugadores adversarios quienes alternan sus movimientos, cada uno ve las fallas de su oponente y sus propios éxitos. En cada turno las reglas del juego definen los movimientos legales y el efecto que tendrá cada movimiento posible. Los juegos comienzan a partir de un estado inicial especificado y termina en una posición que puede ser declarada ganadora, perdedora o tabla.

Desde la perspectiva de un problema de búsqueda, se tiene un árbol de juego que es una representación explícita de todos los posibles caminos de una partida. El nodo raíz es la posición inicial del juego, sus sucesores son las posiciones que el primer jugador puede alcanzar, los sucesores de estos nodos son las posiciones resultantes de la réplica del segundo jugador, y así sucesivamente. Los nodos terminales u hojas representan posiciones ganadoras, perdedoras o tablas. Cada camino de la raíz a un nodo terminal representa una partida completa del juego. El problema está en las dimensiones de este árbol de juego; por ejemplo, en el juego de ajedrez se ha determinado que el factor de ramificación promedio es de alrededor de 35 (Cantidad de jugadas posibles en un momento determinado del juego) y que en un juego promedio cada jugador realiza 50 movimientos, por tanto, para examinar el árbol de juego completo es necesario examinar  $35^{100}$  posiciones.

Un programa que realice una búsqueda a ciegas no podrá seleccionar ni siquiera su primer movimiento. Es necesario algún procedimiento de búsqueda heurística. La técnica utilizada es desarrollar

una parte del árbol de juego hasta una profundidad determinada en función de la potencia de cálculo disponible y de la estabilidad de la posición, evaluar estáticamente las posiciones alcanzadas y luego propagar hacia la raíz del árbol estos valores. Se utilizan diferentes vías que permiten evaluar seriamente las jugadas posibles, explorando solamente un pequeño porcentaje del árbol de búsqueda. El procedimiento de propagación hacia la raíz de los valores estimados para las hojas del árbol más conocido es el MINMAX, y el método más viejo para simplificar el árbol es el Alfa - Beta.

Este método es muy interesante y va necesariamente a desarrollar el pensamiento crítico, donde cada opción en un momento determinado deberá tomar en cuenta tanto las posibilidades reales de la persona o agente como las de su entorno, en este caso su contrincante.

La Figura 10 muestra parte del árbol de búsqueda para el juego del tic - tac - toe (también conocido como tres en raya). El número debajo de cada nodo hoja indica el valor de utilidad del estado terminal desde el punto de vista de Max, valores altos son buenos para Max y malos para Min.

En la práctica resulta imposible tener el árbol de juego y aplicar la función de utilidad a los nodos terminales. Para resolver este problema se sustituye el criterio de terminación por un corte a una profundidad dada y la función de utilidad por una función de evaluación heurística que se aplica a las hojas del árbol.

Una función de evaluación calcula un estimado de la utilidad de una posición desde el punto de vista de uno de los jugadores, en este caso Max. La idea se basa en que los jugadores de ajedrez y de otros juegos, han desarrollado maneras de juzgar las posibilidades de una posición. El resultado del juego depende extremadamente de la calidad de la función de evaluación. La función de evaluación

debe estar de acuerdo con la función de utilidad sobre los estados terminales, y debe ser calculable con un esfuerzo razonable.

Debe haber un compromiso entre la precisión de la función de evaluación y su costo en tiempo. Una tercera característica es que la función de evaluación debe reflejar con precisión la oportunidad real de ganar.

Una forma de construir la función de evaluación es usando la expresión:

$$W_1 * f_1 + W_2 * f_2 + \dots + W_n * f_n$$

donde  $W_i$  son los pesos y los  $f_i$  son los rasgos de una posición particular.

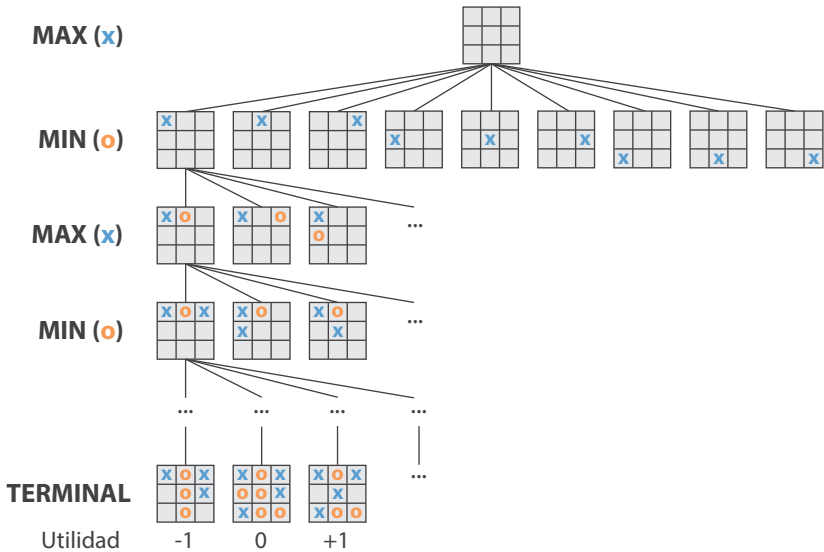
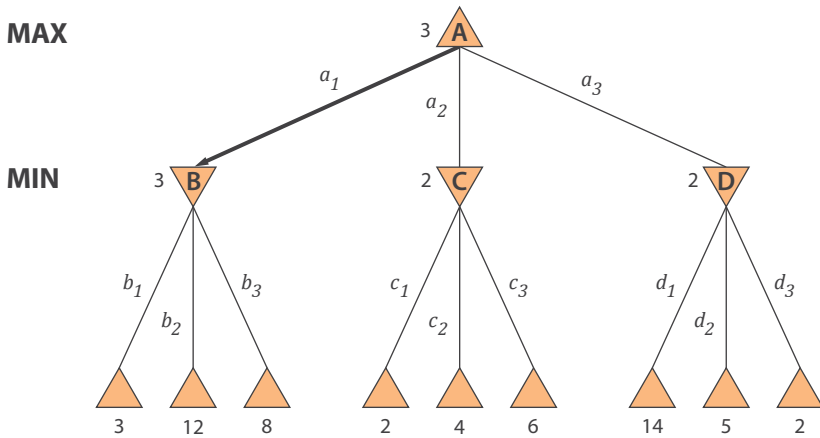


Figura 10. Árbol parcial del juego Tic-Tac-Toe.

Por ejemplo, en el juego de ajedrez, los pesos podrían ser: 1 para el peón, 3 para el alfil, 3 para el caballo, 5 para las torres y 9 para la

reina, y las  $f_i$  serían la cantidad de cada tipo de pieza. Turing propuso otra función de evaluación para el ajedrez: sumar los valores de las piezas negras (N), sumar los valores de las piezas blancas (B) y calcular el cociente B/N.

En la Figura 11 se muestra una parte de un árbol de juego con los nodos hojas evaluados con alguna función de evaluación estática.



**Figura 11.** Ejemplo simplificado de un árbol de juego.

Falta analizar el procedimiento para propagar los valores calculados con la función de evaluación para los nodos hojas del árbol de búsqueda generado hasta un nivel dado hacia la raíz de dicho árbol.

Incluso un juego simple como tic-tac-toe es demasiado complejo para dibujar el árbol de juego entero, por tanto, analicemos el juego trivial representado en la Figura 11. Los movimientos posibles para MAX, en el nodo raíz, se etiquetan por  $a_1, a_2, a_3$ . Las respuestas posibles  $a_1$ , para MIN, son  $b_1, b_2, b_3$ . Este juego particular finaliza después de un movimiento para MAX y MIN. Las utilidades de los estados terminales en este juego varían desde dos a 14.

Considerando un árbol de juegos, la estrategia óptima puede determinarse examinando un valor minimax de cada nodo. El valor minimax de un nodo es la utilidad (para MAX) de estar en el estado correspondiente, asumiendo que ambos jugadores juegan óptimamente desde allí al final del juego.

El valor minimax de un estado terminal (hoja) es su utilidad. Considerando una opción, MAX preferirá moverse a un estado de valor máximo, mientras que MIN prefiere un estado de valor mínimo. Entonces el valor minimax quedaría como:

- Utilidad del nodo, si es un nodo terminal.
- Máximo de los valores minimax de los sucesores del nodo, si es un nodo de MAX.
- Mínimo de los valores minimax de los sucesores del nodo, si es un nodo de MIN.

Aplicando estas ideas al árbol de juego de la Figura 11. Los nodos terminales se etiquetan por sus valores de utilidad. El primer nodo de MIN, etiquetado B, tiene tres sucesores con valores 3, 12 y 8, entonces su valor minimax es 3. Los otros dos nodos de MIN tienen un valor minimax de 2. El nodo raíz es un nodo MAX; sus sucesores tienen valores minimax de 3, 2 y 2; entonces tiene un valor minimax de 3.

Con esto podemos identificar la decisión de MAX en la raíz: la acción  $a_1$  es la opción óptima porque conduce al sucesor con el valor minimax más alto.

Esta definición de juego óptimo para MAX supone que MIN también juega óptimamente (maximiza los resultados del caso-peor para MAX). ¿Y si MIN no juega óptimamente? Entonces se puede demostrar que MAX lo hará aún mejor. Puede haber otras estrategias contra oponentes subóptimos que lo hagan mejor que la estrategia

minimax; pero estas estrategias necesariamente lo hacen peor contra oponentes óptimos.

Veamos otro ejemplo para ilustrar un problema desde el inicio:

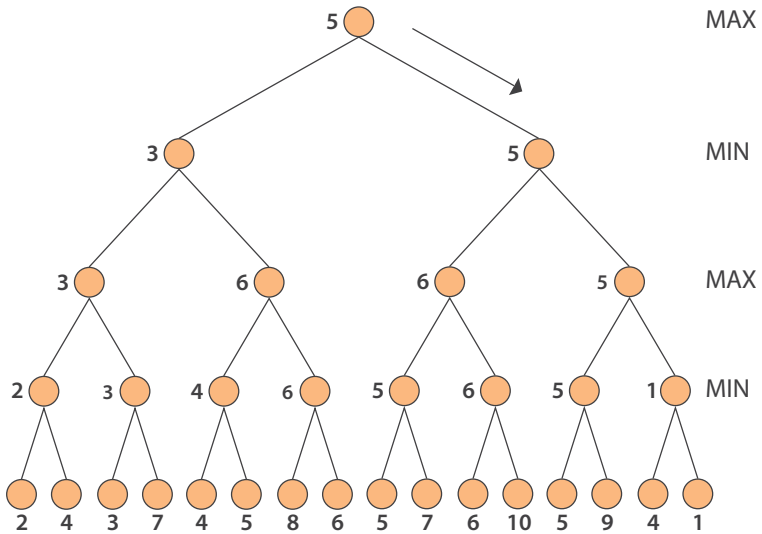
En el árbol de variantes de un juego de mesa creado por un grupo de estudiantes de la Universidad de Oriente, se utiliza una función de evaluación estática para medir la calidad de las posiciones surgidas. En una de las posiciones obtenidas en el juego, se utilizó un árbol de altura 4 para determinar la jugada a realizar por el jugador en turno, que maximiza la función estática a su favor. El factor de ramificación en cada posición es 2. Se desea determinar la mejor jugada para el jugador en turno. A continuación se presenta la evaluación estática de los nodos terminales de dicho árbol.

2	4	3	7	4	5	8	6	5	7	6	10	5	9	4	1
---	---	---	---	---	---	---	---	---	---	---	----	---	---	---	---

Lo primero es representar el árbol de juego, se especifica que son cuatro niveles con factor de ramificación 2 por lo que el árbol queda como se muestra en la Figura 12. Los valores de la función estática se refieren a cada nodo hoja. El algoritmo comienza a ejecutarse analizando el nivel superior a las hojas, en este caso como se puede ver es de minimizar, por lo que el primer nodo de ese nivel analiza los dos primeros nodos hojas, el menor de ellos tiene valor 2 por lo que es el valor minimax que se asigna. De la misma manera se analizan todos los nodos de ese nivel. Posteriormente se sube al siguiente nivel y así por niveles de abajo a arriba hasta el nodo raíz. En este punto se observa que los valores de los nodos son 3 y 5 por lo que la mejor jugada será la que representa el nodo con valor 5.

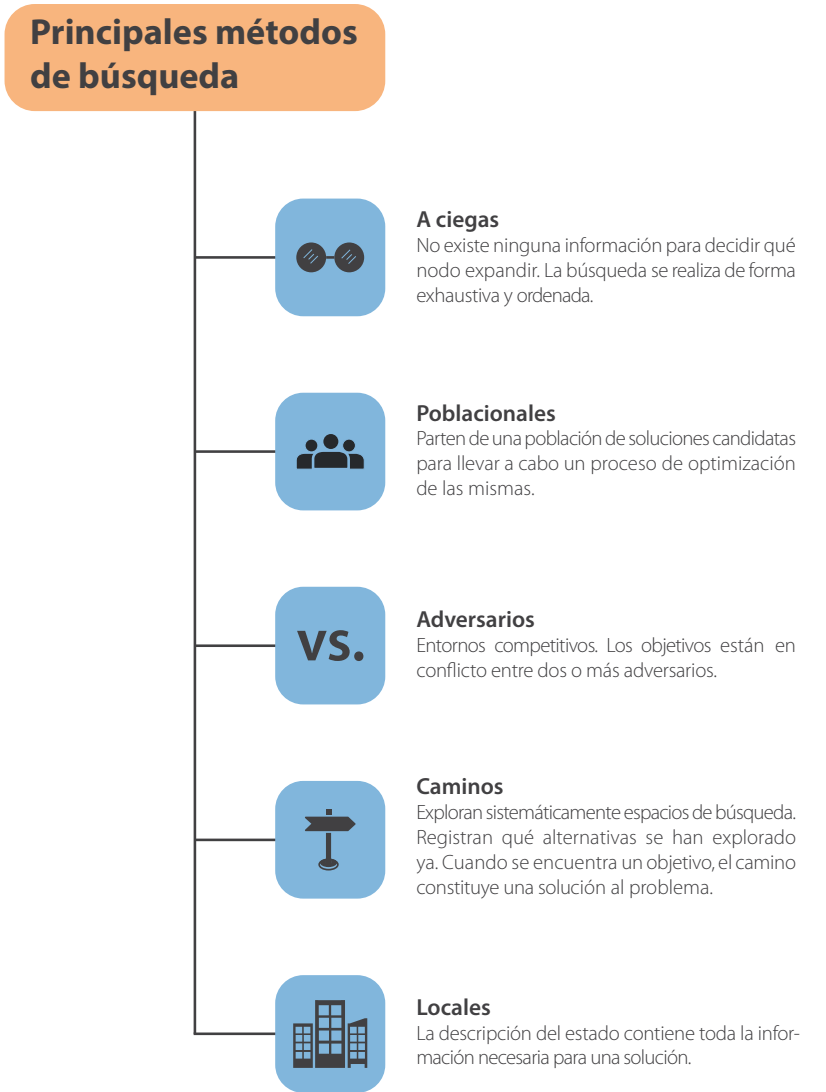
Esto representa que, si el contrario realiza sus mejores jugadas posibles, el jugador puede llegar a una posición del juego con valor 5 de la función estática.





**Figura 12.** Árbol de juego del ejemplo de aplicación del algoritmo Min-Max.

Hasta aquí hemos visto los principales métodos de búsqueda en IA (Figura 13). Pero, en la IA se reconocen otras formas de razonamiento muy ligadas a la representación que se hace del conocimiento, por lo tanto las abordaremos con más detalles en el apartado de representación del conocimiento. Un ejemplo es el razonamiento basado en reglas (RBR), el cual utiliza especificaciones de consecuencias o conclusiones dadas ciertas evidencias. También se abordará con más profundidad en el apartado de tratamiento de la incertidumbre.



**Figura 13.** Principales métodos de búsqueda en IA.

Otro ejemplo es el razonamiento basado en casos (RBC), este utiliza ejemplos estructurados convenientemente ya resueltos para dar solución a nuevos problemas. Este es un razonamiento complejo que utiliza conocimiento estructurado, debe tratar con incertidumbre en los datos y en el propio razonamiento e incorpora elementos de aprendizaje. Dada esta integralidad en el razonamiento y su utilización, generalmente, para problemas de clasificación o regresión (tareas del aprendizaje automático) se volverá a abordar en el apartado de aprendizaje.



---

# EL CONOCIMIENTO Y SU REPRESENTACIÓN

---

La representación del conocimiento necesario para desarrollar una aplicación informática, requiere estructurar el conocimiento sobre un dominio de aplicación, de modo que los problemas sobre ese dominio, sean más fáciles de resolver. Para ello se debe desarrollar una notación lo suficientemente precisa mediante la cual se pueda representar el conocimiento a utilizar por el sistema. A esa notación se le ha llamado en la IA Forma de Representación del Conocimiento (FRC). Podemos decir entonces que:

***La Representación del Conocimiento** es el proceso de estructurar el conocimiento mediante una notación suficientemente precisa como para ser utilizado por el sistema en su comportamiento.*

Hasta el momento se ha visto la representación de un conocimiento muy básico sobre un problema de estados. El conocimiento que se tenía sobre el problema se estructuró de manera que pudiéramos guardar y trabajar con la información que teníamos de los estados, las acciones para moverse de un estado a otro y el criterio objetivo que permitía evaluar si ya estábamos en una solución. Por

esta vía se pueden resolver muchos problemas en IA, pero, de más está decir, que el propio desarrollo de la IA le exige resolver problemas donde hay que lidiar con mucho más conocimiento que requiere ser organizado, estructurado y utilizado en la solución de problemas más complejos.

### **Toda FRC debería cumplir ciertos criterios (Gálvez, 1998):**

- Describir el dominio de una manera natural, reflejando tanto como sea posible la estructura de los objetos, los hechos y las relaciones entre ellos.
- Aceptar conocimiento empírico, teórico o heurístico, y combinar el conocimiento declarativo con el procedimental, de acuerdo con los requerimientos de la aplicación.
- Estratificar el conocimiento de acuerdo con su significado y funciones.
- Comprensión por el usuario, del conocimiento que está almacenado.
- Utilización efectiva del conocimiento almacenado.

Estas son características deseadas, pero, es muy difícil que una única FRC cumpla con todas ellas. Como tarea de suma importancia dentro de la IA, es un gran campo abierto de investigaciones en nuestros días.

## **Formas específicas de representación del conocimiento**

La lógica es considerada una forma de representar el conocimiento de las más antiguas, es una subdivisión de la Filosofía. En

el razonamiento lógico, deben ser utilizados algunos métodos para convertir dicho razonamiento en una forma adecuada que sea manipulada por la computadora. La máxima expresión de estas ideas en IA fue la creación del lenguaje Prolog, el cual, aún hoy, se utiliza y tiene como componente esencial la representación del conocimiento en forma de hechos y reglas.

Las reglas de producción son muy populares en la IA, para ellas se han desarrollado varias herramientas computacionales. En su forma más sencilla, una regla de producción es un par (A, B) que puede representarse como  $A \Rightarrow B$ .

La interpretación de una regla es que, si el antecedente se puede satisfacer, entonces se obtiene el consecuente. Utilizan un formato SI-ENTONCES para representar el conocimiento, la parte SI de una regla es la condición, también llamada premisa o antecedente, y la parte ENTONCES de la regla es la acción, conclusión o consecuente. Permite inferir un conjunto de hechos nuevos si se verifican las condiciones establecidas en la parte SI.

Veamos un ejemplo de razonamiento utilizando una representación del conocimiento en forma de reglas de producción.

Se tiene el siguiente conjunto de reglas:

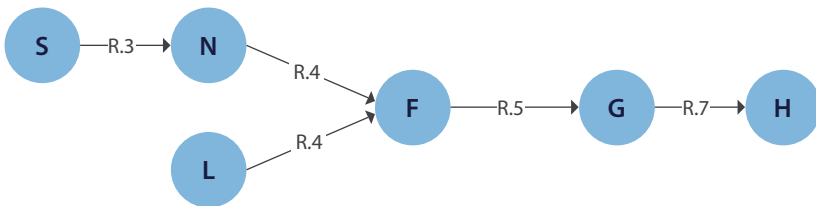
1. **Si** se prepara la tierra (P) y llueve (L) **entonces** se tiene buena tierra (T)
2. **Si** se riegan semillas (S) y se tiene buena tierra (T) **entonces** crecen las plantas (C)
3. **Si** se riegan semillas (S) **entonces** nacen las plantas (N)
4. **Si** llueve (L) y nacen las plantas (N) **entonces** florecen las plantas (F)
5. **Si** florecen las plantas (F) **entonces** salen los frutos (G)
6. **Si** crecen las plantas (C) **entonces** salen los frutos (G)
7. **Si** salen los frutos (G) **entonces** se comen los frutos (H)

Se sabe que se regaron semillas (S) y llueve (L), se desea saber si se comerán los frutos (H).

El proceso de razonamiento comienza buscando la regla que se satisfaga a partir de los hechos conocidos. La regla uno (1) no se satisface ya que no se conoce si se preparó la tierra. La regla dos (2) tampoco ya que no se conoce si se tiene buena tierra. Sin embargo, la regla tres (3) solo tiene como antecedente que se riegan semillas y es un hecho conocido, por tanto, se infiere el hecho *nacen las plantas*. Siguiendo el mismo procedimiento se van activando las reglas e infiriendo los hechos como se muestra en la Tabla 2 y Figura 14.

**Tabla 2.** Solución al ejemplo de inferencia en reglas de producción

REGLA QUE SE ACTIVA	HECHO QUE INFIERE
3	N
4	F
5	G
7	H



**Figura 14.**

Con el propósito de superar las limitaciones de otros modelos, se desarrolló el esquema de representación estructurada en marcos. Este permitió agrupar información y establecer relaciones entre esos



grupos. Esta representación también pretende dotarnos de expectativas ante nuevas situaciones, usando el conocimiento existente sobre situaciones previas.

La idea fundamental es contar con un esquema donde el conocimiento sea codificado en paquetes o estructuras, que podremos llamar marcos, casos, o simplemente elementos, los cuales se encuentran interrelacionados entre sí. Digamos entonces que un marco es una estructura de datos que contiene información acerca de un objeto. Ofrece una representación estereotipada de un objeto o una clase de objetos. De forma general es una colección de conocimiento que consiste de un nombre y sus propiedades, las cuales son un conjunto de pares atributo/valor que representan un objeto.

Esta forma de representación del conocimiento se asocia a la idea de la programación orientada a objetos, donde se utilizan conceptos de esta, como las instancias que especifican los objetos concretos de un marco; o la herencia, que permite reutilizar información entre marcos relacionados directamente.

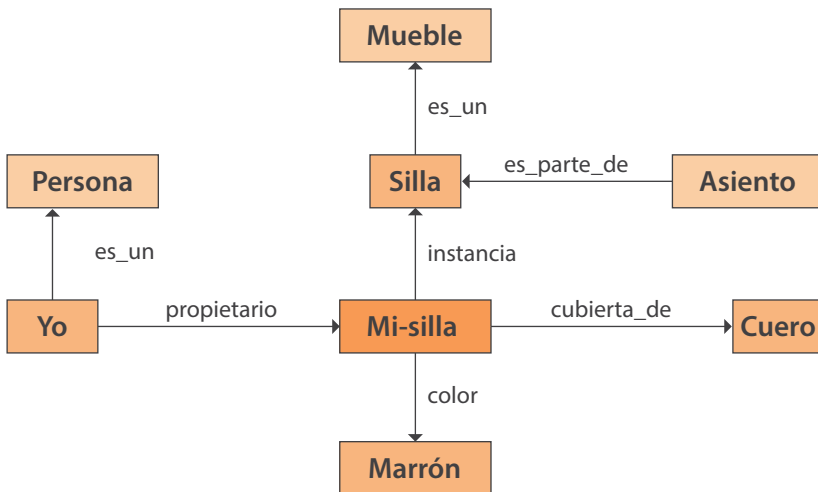
Muchas veces nos encontramos con problemas donde es necesario representar el conocimiento a través de conceptos, rasgos y las relaciones que pueden establecerse entre ellos. La representación mediante redes semánticas pone énfasis en la conectividad del conocimiento, similar a la construcción de un diccionario.

Una red semántica consta de puntos o nodos, conectados por enlaces o arcos que describen las relaciones entre los nodos. Los nodos representan conceptos de entidades, atributos, eventos y estados. A los arcos se asocian etiquetas que especifican los tipos de relaciones. De ahí que, frecuentemente, se ilustra una red semántica como un grafo etiquetado dirigido, aunque se han propuesto diversas versiones de redes semánticas.

La representación del conocimiento en una red es interesante y potente pero también muy complejo. Las redes se basan en dos condiciones:

1. El conocimiento se representa por nodos en un grafo no jerárquico, por lo que todos los nodos tienen la misma importancia y cualquiera puede ser usado como punto de partida.
2. Los nodos están organizados de modo que tipos similares de conocimiento son adyacentes.

Un ejemplo sencillo de red semántica se muestra en la Figura 15. Este ejemplo muestra el concepto silla. Se aprecian como relaciones de mayor interés: es\_un, parte\_de e instancia.



**Figura 15.** Ejemplo sencillo de red semántica del concepto silla.

Las redes semánticas han sido propuestas como un mecanismo para simular algunas de las características asociativas de la memoria humana. Muchos investigadores en IA consideran que las redes semánticas son el formalismo más apropiado para representar el

conocimiento humano. Es cierto que se han utilizado en varios tipos de problemas, incluidos esquemas de representación de significados de oraciones en lenguaje natural.

## La representación del todo en IA

La tarea de representarlo todo en el mundo es muy compleja, imposible aún en nuestros días. La IA necesita representar desde el problema hasta su solución, incluso hay autores que incluyen la representación de ciertos eventos futuros dentro de la propia IA.

La formulación de problemas de estados es una primera noción de representar un problema para darle solución. Aunque ya hemos visto algunas técnicas generales de representación del conocimiento en IA y elementos básicos generales de formular problemas, este tema debe tener una mirada más integral.

Un ejemplo útil para ver la importancia de la representación del problema son los métodos poblacionales como los algoritmos genéticos. Un algoritmo genético (AG) es un método inspirado en la evolución de las especies. Para ello parte de un conjunto de individuos o población y aplica operadores, donde los más comunes son la selección, el cruzamiento y la mutación. Esta es una búsqueda local en la que se parte de una población de soluciones candidatas donde los estados sucesores se generan combinando dos estados padres, más que modificar un solo estado. Un AG comienza con un conjunto de  $k$  estados generados aleatoriamente. Cada estado, o individuo, está representado como una cadena sobre un alfabeto finito, siendo más común las cadenas de ceros y unos.

Un ejemplo de formulación de estado en el problema de las ocho reinas debe especificar las posiciones de las ocho reinas, cada una en una columna de ocho cuadrados, y se requieren  $8 \times \log_2 8 = 24$  bits.

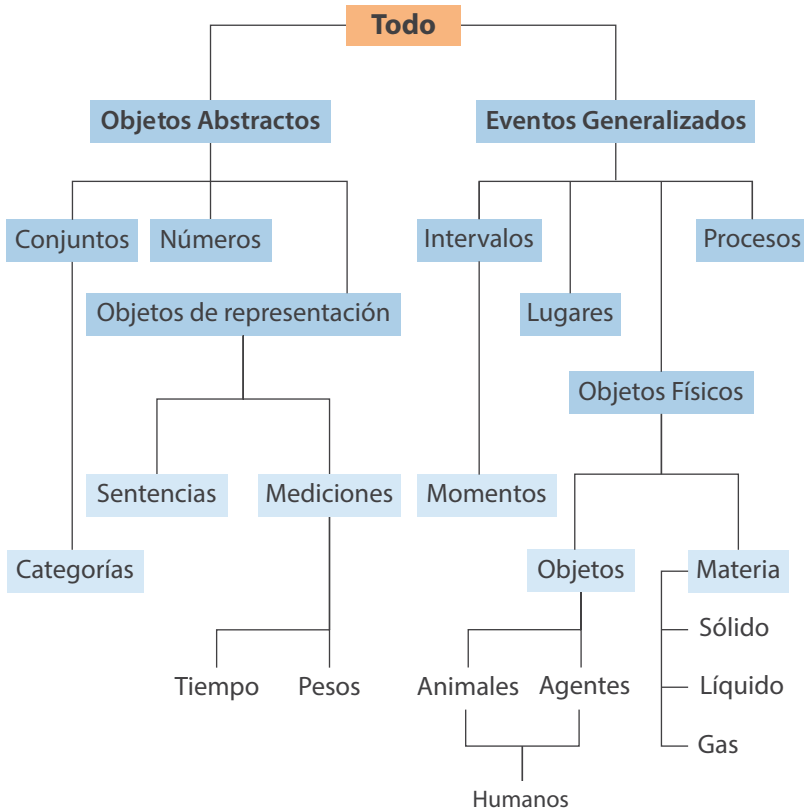
Asumiendo la configuración de reinas de la Figura 9 (b) donde los primeros 8 bits representan la primera columna con los cuadrados de arriba a abajo. Se representa con 1 la posición donde está la reina y con cero la celda donde no hay reina. Quedaría el individuo de la siguiente manera:

```
000000010010000000000010000100000100000000001000100000  
0000000100
```

Como se puede apreciar, este es un individuo bastante grande. Trabajar con varios de ellos como población, al menos, se hace engorroso visualmente.

Otra posibilidad es representar el estado como ocho dígitos, cada uno en el rango de uno a ocho. De esta forma, cada dígito representa la posición de la reina en cada columna. Para el mismo ejemplo la representación quedaría: 83742516.

Las investigaciones han avanzado hacia un modelo que permita incorporar nuevo conocimiento, sea cual sea el dominio. Por ejemplo, se han trabajado varios conceptos para definir lo que significa un objeto físico, los detalles de diferentes tipos de objetos (robots, televisores o cualquier otra cosa), de forma que pueda ser rellenado posteriormente. El marco de trabajo general para los conceptos se llama ontología superior, representando en los grafos los conceptos generales en la parte superior y los conceptos más específicos debajo de ellos (Russell & Norvig, 2004), como en la Figura 16.



**Figura 16.** Representación ontológica general simplificada.

Una ontología es una especificación formal explícita de una conceptualización compartida (Guerrero, 2012). Es un modelo abstracto que representa el conocimiento aceptado por un grupo de personas sobre algún fenómeno en el mundo. Para ello define explícitamente los conceptos relevantes y las restricciones sobre su aplicación, de forma que sean entendibles por la máquina.

En esencia una ontología es un conjunto de conceptos (clases), ordenados jerárquicamente y las sentencias sobre ellos, normalmente escritos en términos de la Lógica Descriptiva. Por tanto en una ontología se pueden identificar como componentes, a las clases, sus instancias y las relaciones entre ellas. Esta triada (clase-relaciones-instancias) se descompone en más elementos, así a las relaciones que siempre tienen lugar, en cualquier estado de la ontología y que se expresan con enunciados tautológicos, se les denomina axiomas o cuando la relación es funcional entonces se dice que se está en presencia de una función.

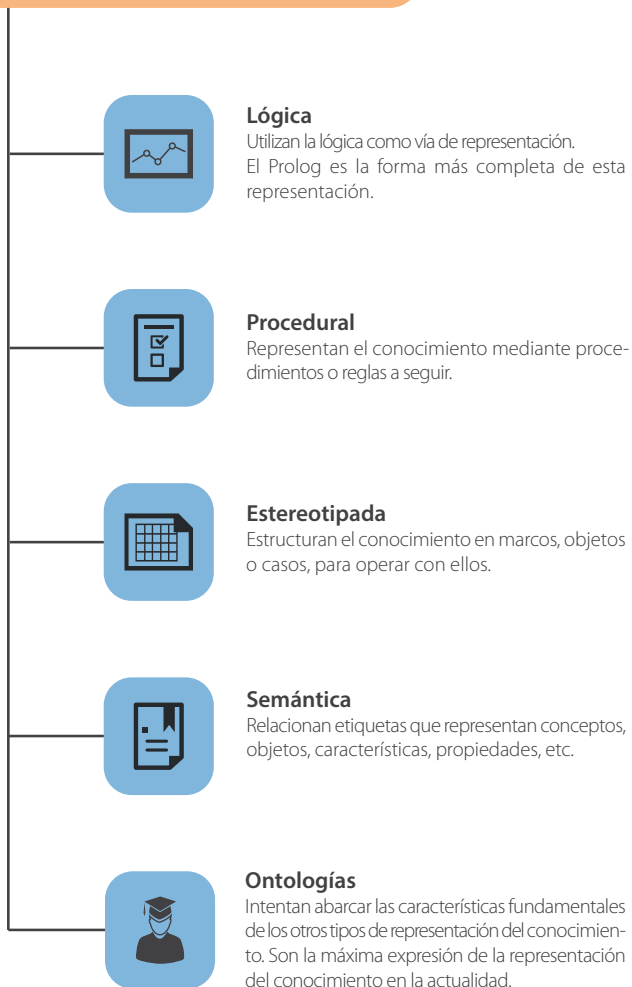
### Los cuatro componentes básicos de las ontologías son:

- **Clases:** representan conceptos y están usualmente organizadas en jerarquías a través de las cuales se puede aplicar los mecanismos de herencia.
- **Relaciones:** representan un tipo de asociación entre los conceptos del dominio.
- **Axiomas formales:** sirven para modelar sentencias que siempre son ciertas. Habitualmente son empleados para representar el conocimiento que no es posible hacerlo con los otros componentes. También sirven para verificar la consistencia de la propia ontología o del conocimiento guardado en la base de conocimiento y para inferir conocimiento nuevo.
- Las ontologías usualmente contienen relaciones binarias. El conjunto donde están definidos los valores del primer argumento de la relación binaria es conocido como el dominio, mientras que el conjunto donde se toman los valores del segundo argumento se denomina rango.

- **Las instancias:** son usadas para representar a los elementos o individuos en una ontología.

En la práctica, las ontologías representan un dominio específico limitado. Sin embargo, la representación del conocimiento a gran escala necesita una ontología de propósito general para organizar y unir varios dominios de conocimiento específicos, lo cual es una gran tarea de la IA en nuestros días. Aunque las formas de representación del conocimiento más utilizadas se pudieran resumir en cinco tipos principales o generales (Figura 17), es importante reforzar la idea de que la representación del conocimiento en IA estará directamente asociada al problema específico que se desea resolver. Para lo cual, buscará las vías y formas ajustadas a las condiciones, características del conocimiento a representar y necesidades propias de la aplicación a desarrollar.

## Principales formas utilizadas para representar el conocimiento



**Figura 17.** Principales formas utilizadas para representar el conocimiento



---

# TRATAMIENTO DE LA INCERTIDUMBRE

---

Si un agente conoce bien su entorno, algunos pasos lógicos garantizarán obtener buenos resultados. Pero, casi nunca se tiene acceso a toda la verdad sobre el entorno. Por tanto, el agente necesita saber comportarse en condiciones de incertidumbre.

La incertidumbre no se puede analizar por separado del conocimiento que se representa o del razonamiento que se realice. De ahí que este núcleo de conocimiento de la IA debe abordarse desde la propia representación del conocimiento y el razonamiento que se realiza, incluso la forma de aprendizaje está estrechamente relacionada con el tratamiento que se realice de la incertidumbre.

El término muy utilizado de razonamiento bajo incertidumbre se refiere, en una manera general, a problemas de representar y razonar con conocimiento que no puede ser expresado como afirmaciones categóricas. De alguna manera ya hemos mostrado en la IA formas de tratar la incertidumbre, al introducir heurísticas en la solución de problemas, estamos introduciendo información imprecisa, simplificada, que nos permite acercarnos a la solución de un problema. El razonamiento ya mostrado de los algoritmos que utilizan heurísticas es una forma de tratar con la incertidumbre que el propio problema o la solución formulada introduce.

Para algunos autores el razonamiento bajo incertidumbre no es una simple subárea de la IA, sino su propia base o fundamento. En este libro se asume el tratamiento de la incertidumbre como uno de los cuatro núcleos básicos de la IA contribuyendo directamente al razonamiento, a la representación del conocimiento y al aprendizaje. Algunos de los métodos que veremos en este apartado son razonamientos que se basan en el tratamiento de la incertidumbre para la toma de decisiones. Con estos criterios se propone que:

*El tratamiento de la incertidumbre es el proceso de identificar criterios para tomar decisiones al representar y razonar cuando no se tiene todo el conocimiento sobre el entorno.*

La incertidumbre puede manifestarse de diversas formas y ser provocada por diferentes causas. A grandes rasgos, se pueden reconocer tres grandes categorías de incertidumbre (Gálvez, 1998):

1. La provocada por información o conocimiento impreciso.
2. La provocada por información incompleta.
3. La provocada por conceptos o palabras que son inherentemente inexactos.

En el primer caso nos referimos al hecho de tener que hacer inferencias a partir de datos, de los cuales no estamos completamente seguros (por errores de medición, de transmisión, etc.) o de relaciones de causalidad no totalmente seguras (un conjunto de sistemas puede ayudar a indicar un diagnóstico particular, sin ser conclusivos). El segundo caso es cuando no se tiene toda la información o resulta muy costoso obtenerla o considerarla. El último caso se refiere al hecho de tener que usar en la inferencia conceptos generales como *alto*, *largo*.

Se han identificado algunas razones que dan lugar a la incertidumbre (R. E. Bello et al., 2002):

**Pereza:** Cuando determinar el conjunto completo de opciones implica demasiado trabajo.

**Ignorancia teórica:** Cuando la ciencia no cuenta aún con una teoría completa sobre ese problema y su solución.

**Ignorancia práctica:** Cuando conociendo todas las opciones y teorías, se hace improbable recopilar toda la información que se requiere.

En estos casos, un sistema racional puede tener sólo un **grado de creencia** del conocimiento que posee. La presencia de incertidumbre cambia radicalmente el modo en que se toman las decisiones.

### La visión probabilista de la incertidumbre

Para algunos autores (Álvarez, 1989) (Russell & Norvig, 2004) la principal herramienta para tratar con grados de creencia es la teoría de la probabilidad. La probabilidad proporciona una manera de resumir la incertidumbre que se deriva de nuestra pereza e ignorancia. Asignar probabilidad 0 a una sentencia determinada corresponde a una creencia de que la sentencia es falsa, mientras que asignar una probabilidad de 1 corresponde a una creencia de que la sentencia es cierta. Las probabilidades entre 0 y 1 corresponden a grados intermedios de creencia en la veracidad de la sentencia. La sentencia en sí misma es verdadera o falsa. El grado de creencia es diferente de un grado de veracidad. Una probabilidad de 0,8 no significa que una sentencia sea 80 por ciento verdadero, sino un 80 por ciento de grado de creencia.

Las creencias dependen de las percepciones que se han recibido hasta el momento, es decir, el conocimiento que se tiene en un momento determinado. Estas percepciones constituyen la evidencia sobre la que se basan las afirmaciones de probabilidades. Cuando se reciben nuevas percepciones, las valoraciones de probabilidad se actualizan para reflejar la nueva evidencia. Antes de que la evidencia se obtenga, hablaremos de probabilidad a priori o incondicional; después de obtener la evidencia, hablaremos de probabilidad a posteriori o condicional. En la mayoría de los casos, un agente tendrá evidencia de sus percepciones y le será útil calcular la probabilidad de las consecuencias de sus acciones.

De la teoría de probabilidades se tiene que la probabilidad de que ocurra un evento X es la razón del número de ocurrencias favorables y el número total de eventos y se denota por  $P(X)$ .

### Existen tres axiomas básicos conocidos como axiomas de Kolmogorov:

1. Todas las probabilidades están entre 0 y 1. Para cualquier proposición a:

$$0 \leq P(a) \leq 1$$

2. Las proposiciones necesariamente ciertas (es decir, válidas) tienen probabilidad 1, y las proposiciones necesariamente falsas (es decir, insatisfactibles) tienen probabilidad 0.

$$P(\text{cierto})=1 \quad P(\text{falso})=0$$

3. La probabilidad de una disyunción viene dada por:

$$P(a \vee b) = P(a) + P(b) - P(a \wedge b)$$

Si consideramos que los casos donde a es válido, junto con los casos donde b es válido, cubren todos los casos donde a y b

son válidos; pero la suma de los dos conjuntos de casos cuenta su intersección dos veces, así que necesitamos restar  $P(a \wedge b)$ .

Muchas veces se tiene un evento con un conjunto de resultados posibles. En este caso es necesario conocer todos los posibles resultados (completitud). De igual forma, no se pueden dar dos resultados distintos simultáneamente, es decir deben ser mutuamente excluyentes. Esto se refiere al término variable aleatoria. Ejemplo la variable aleatoria  $M$ ="Resultado de tirar moneda al aire" se puede hablar de dos resultados posibles  $M = \text{cara}$  y  $M = \text{escudo}$ . La probabilidad de cada uno se expresa como  $P(M=\text{cara})$  y  $P(M=\text{escudo})$ . De forma simplificada se puede expresar  $P(\text{cara})$  y  $P(\text{escudo})$ . Si una variable aleatoria, por ejemplo  $N$ ="Tiene Covid19" toma solo valores verdadero o falso se puede expresar  $P(\text{Covid19}=\text{verdadero})$  como  $P(\text{Covid19})$  y  $P(\text{Covid19}=\text{falso})$  como  $P(\neg\text{Covid19})$ .

Para una variable aleatoria la probabilidad de cada valor que pueda tomar se llama distribución de probabilidad. Por ejemplo para la variable  $L$ ="Llueve" se tiene:

LLUEVE	P(LLUEVE)
Verdadero	0.1
Falso	0.9

Cuando se tienen más de una variable se necesita asignar probabilidades a cada posible combinación de los valores de las variables. Al listado de todos esos valores se llama la distribución conjunta del conjunto de variables. Un ejemplo con la variable  $L$ ="Llueve" y  $S$ ="Salgo a la calle".

L	S	P(L,S)
Llueve	Salgo	0.01
Llueve	¬ Salgo	0.09
¬ Llueve	Salgo	0.2
¬ Llueve	¬ Salgo	0.7

A veces esta teoría puede usarse con éxito en la IA. Una regla puede tener varios antecedentes, cada uno con su probabilidad. Pudiera calcularse la probabilidad múltiple del antecedente, multiplicando las probabilidades de cada una de las condiciones, pero esto sólo es cierto si son independientes unas con otras. Esto generalmente no ocurre, por lo que se utiliza un procedimiento especial conocido como la regla de Bayes, el cual permite calcular la probabilidad de que ocurra un evento dado que ha ocurrido otro evento.

Las probabilidades condicionales pueden definirse en términos de probabilidades no condicionales mediante la ecuación:

$$P(a|b) = \frac{P(a \wedge b)}{P(b)}$$

Siempre que  $P(b) > 0$ . De esta se deriva la llamada regla del producto:

$$P(a \wedge b) = P(a|b)P(b)$$

La regla proviene del hecho de que para saber si a y b son ciertos, necesitamos que b sea cierto, y también necesitamos que a sea cierto dado b, de aquí también se obtiene:

$$P(a \wedge b) = P(b|a)P(a)$$

En algunos casos, es más fácil razonar en términos de probabilidades a priori de conjunciones, pero mayoritariamente se utilizan

probabilidades condicionales como instrumento para inferencia probabilista.

Derivado de aquí se puede llegar a:

$$P(b|a) = \frac{P(a \vee b)P(b)}{P(a)}$$

Esta ecuación se conoce como regla o teorema de Bayes y es la base de los sistemas de IA modernos con inferencia probabilística. A primera vista la regla de Bayes puede no parece muy útil, ya que requiere tres términos (una probabilidad condicional y dos probabilidades incondicionales) sólo para calcular una probabilidad. Sin embargo, es útil en la práctica porque hay muchos casos donde se dispone de buenas estimaciones probabilistas para estos tres números y necesitamos calcular el cuarto.

Veamos un ejemplo, un estomatólogo cuenta con un dominio de tres variables booleanas: Dolor-de-muelas, Caries e Infectarse (Por ejemplo, por mala manipulación del médico). La distribución conjunta completa es una tabla de dimensión 2X2X2 como se muestra en la Tabla 3.

**Tabla 3.** Distribución conjunta completa de tres variables.

	DOLOR DE MUELAS		¬DOLOR DE MUELAS	
	Infectarse	¬infectarse	infectarse	¬infectarse
Caries	0,108	0,012	0,072	0,008
¬caries	0,016	0,064	0,144	0,576

Las probabilidades de la distribución conjunta suman 1, como se requiere por los axiomas de la probabilidad. La tabla brinda un modo directo de calcular la probabilidad de cualquier proposición,

simple o compleja: simplemente identificamos aquellos sucesos atómicos en los que la proposición es cierta y sumamos sus probabilidades. Por ejemplo, hay seis sucesos en los que se verifica caries  $\vee$  dolor-de-muelas: toda la fila donde hay “Caries” y las dos columnas de “dolor de muelas”. Las celdas que se corresponden a los dos se cuentan una sola vez:

$$P(\text{caries} \vee \text{dolor-de-muelas}) = 0,108 + 0,012 + 0,072 + 0,008 + 0,016 + 0,064 = 0,28$$

Podemos calcular la probabilidad de una caries, dada la evidencia de dolor de muelas, como sigue:

$$P(\text{caries}|\text{dolor}) = \frac{P(\text{caries} \wedge \text{dolor})}{P(\text{dolor})}$$

$$\frac{0,108 + 0,012}{0,108 + 0,012 + 0,016 + 0,064} = 0,6$$

Se puede además calcular la probabilidad de que no hay caries, dado un dolor de muelas:

$$P(\neg \text{caries}|\text{dolor}) = \frac{P(\neg \text{caries} \wedge \text{dolor})}{P(\text{dolor})}$$

$$\frac{0,016 + 0,064}{0,108 + 0,012 + 0,016 + 0,064} = 0,4$$

De esta manera es muy fácil, pero, contar con una tabla de las distribuciones conjuntas completa es muchas veces complicado, por lo que puede hacerse más práctico trabajar con las probabilidades condicionales. Veamos otro ejemplo.

Un médico sabe que, de cada 100 pacientes con la enfermedad de meningitis, en 50 causa al paciente que el cuello se agarrote. El médico también conoce que la probabilidad a priori de que un paciente tenga meningitis es  $1/50.000$  y la probabilidad a priori de que



cualquier paciente tenga un cuello agarrotado es  $1/20$ . El médico desea saber la probabilidad de que un paciente que llegue a consulta con el cuello agarrotado, tenga meningitis.

Tomemos a  $s$  como la proposición de que el paciente tiene un cuello agarrotado y  $m$  como la proposición de que el paciente tiene meningitis. Tenemos:

$$P(s|m) = 0,5$$

$$P(m) = 1/50000$$

$$P(s) = 1/20$$

$$P(m|s) = \frac{P(s|m)P(m)}{P(s)} = \frac{0,5 * 1/50000}{1/20} = 0,0002$$

Es decir, esperamos que sólo un paciente de entre 5.000 con un cuello agarrotado tenga meningitis.

La regla de Bayes puede ser un elemento de evidencia útil para responder a preguntas probabilistas condicionales. La información probabilista está muchas veces disponible de la forma  $P(\text{efecto}/\text{causa})$ .

Pero como dijimos no siempre se tiene la distribución conjunta completa. Si hay  $n$  posibles variables de evidencia, entonces hay  $2n$  posibles combinaciones de valores observados para los que necesitaríamos conocer las probabilidades condicionales.

La noción de independencia nos simplifica este problema. Sería muy útil si *Dolor-de-muelas* e *Infectarse* fueran independientes. Pero no lo son: si el médico infecta la muela debió ocurrir porque el paciente tiene una caries y si tiene una caries probablemente causa un dolor de muelas. Sin embargo, estas variables son independientes

dada la presencia o la ausencia de caries. Cada una está directamente causada por la caries, pero ninguna de ellas tiene un efecto directo sobre la otra: el dolor de muelas depende del estado de los nervios de la muela, mientras que la precisión del médico depende de su habilidad, para la cual es irrelevante el dolor de muelas.

Este ejemplo ilustra un patrón de ocurrencia común en el que una sola causa influye directamente en un número de efectos, todos ellos independientes condicionalmente, dada la causa. La distribución conjunta completa puede escribirse como:

$$P(\text{Causa}, \text{Efecto}_1, \dots, \text{Efecto}_n) = P(\text{Causa}) \prod_i P(\text{Efecto}_i \mid \text{Causa})$$

Esta distribución de probabilidad se llama modelo simple-Bayes o Bayes ingenuo. En la práctica, estos sistemas pueden trabajar sorprendentemente bien, incluso cuando los supuestos de independencia no son ciertos.

Este enfoque es ampliamente utilizado en la IA, aunque requiere gran cantidad de datos para estimar las probabilidades a priori y a posteriori, lo cual a veces no es fácil de obtener. La evaluación de la fórmula de Bayes en un dominio complejo considera muchas probabilidades, por lo que requiere gran cantidad de cálculos. De igual forma si se añade un cambio a la base de conocimientos existente hay que cambiar las probabilidades, pues la suma de las probabilidades de todos los resultados posibles, o sea, la suma de las probabilidades de todas las hipótesis posibles que revelan la evidencia debe ser 1.

Los modelos de tipo probabilístico han sido criticados, por el elevado número de parámetros y por las dificultades de estimarlos a partir de los datos. Existe un criterio bastante generalizado en que el modelo general de dependencia no es práctico en la gran mayoría de los casos. De igual forma, los modelos de independencia son dema-

siado simples para reproducir muchos casos prácticos. Sin embargo, entre estos dos extremos, la dependencia general y la independencia, existen muchas posibilidades de aplicación de estas ideas.

Existen modelos mucho más completos de estas mismas ideas. Para representar las dependencias entre las variables y para mostrar una descripción escueta de cualquier distribución de probabilidad conjunta completa se utiliza una estructura de datos llamada **red bayesiana**.

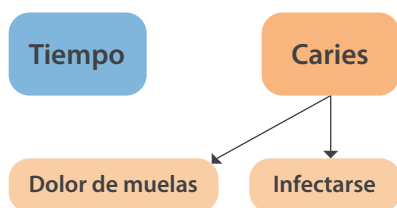
**Una red bayesiana es un grafo dirigido en el que cada nodo está comentado con información probabilista cuantitativa. La especificación completa es como sigue:**

1. Un conjunto de variables aleatorias forma los nodos de la red. Las variables pueden ser discretas o continuas.
2. Un conjunto de enlaces dirigidos o flechas conectan pares de nodos. Si hay una flecha de un nodo  $X$  a un nodo  $Y$ , se dice que  $X$  es un padre de  $Y$ .
3. Cada nodo  $X_i$  tiene una distribución de probabilidad condicionada  $P(X_i \vee \text{Padres}(X_i))$  que cuantifica el efecto de los padres del nodo.
4. El grafo no tiene ciclos dirigidos (y así es un grafo acíclico dirigido).

El conjunto de nodos y enlaces, conocido como topología de la red, especifica las relaciones de independencia condicional que se tienen en el dominio. El significado intuitivo de una flecha en una red de este tipo es que  $X$  tiene una influencia directa sobre  $Y$ . Por lo general es sencillo para un experto del dominio decidir qué influencias directas existen entre los componentes, mucho más sencillo que

especificar las probabilidades. Una vez que la topología de la red bayesiana está diseñada, necesitamos sólo especificar una distribución de probabilidad condicional para cada variable, dados sus padres. La combinación de la topología y las distribuciones condicionales son suficientes para especificar la distribución conjunta completa para todas las variables.

Adicionemos a la distribución conjunta completa de la Tabla 3 una cuarta variable: *Tiempo*. Esta representa el estado del tiempo con cuatro posibles valores: soleado, nublado, lluvioso o tormentoso. Como puede apreciarse el *Tiempo* es independiente de las otras variables; además, el *Dolor-de-muelas* e *Infectarse* son independientes condicionalmente, dada *Caries*. Estas relaciones se presentan en la estructura de la red bayesiana que se muestra en la Figura 18.



**Figura 18.** Red bayesiana simple.

Formalmente, la independencia condicional de *Dolor-de-muelas* e *Infectarse* dada *Caries* está indicada por la ausencia de un enlace entre *Dolor-de-muelas* e *Infectarse*. Intuitivamente, la red representa el hecho de que *Caries* es una causa directa de *Dolor-de-muelas* e *Infectarse*, aunque no exista una relación causal directa entre *Dolor-de-muelas* e *Infectarse*.

Ahora consideremos el siguiente ejemplo adaptado de (Russell & Norvig, 2004) que asume el problema de un residente en la ciudad de Los Ángeles, donde es frecuente la ocurrencia de sismos. Esta

persona tiene una alarma antirrobo instalada en casa, es bastante fiable en la detección de un robo, pero también responde en ocasiones a pequeños terremotos. Dos vecinos, John y Mary, han prometido llamarlo al trabajo cuando oigan la alarma. John siempre llama cuando oye la alarma, pero a veces confunde el timbre del teléfono con la alarma y lo llama también. A Mary, por otro lado, le gusta la música bastante fuerte y a veces no oye para nada la alarma. La red bayesiana para este dominio se presenta en la Figura 19.

Primeramente, analicemos la topología de la red. En el caso de esta red-robo, la topología muestra que robos y terremotos afectan directamente a la probabilidad de que la alarma se dispare, pero tanto la llamada de John como la de Mary dependen sólo de la alarma.

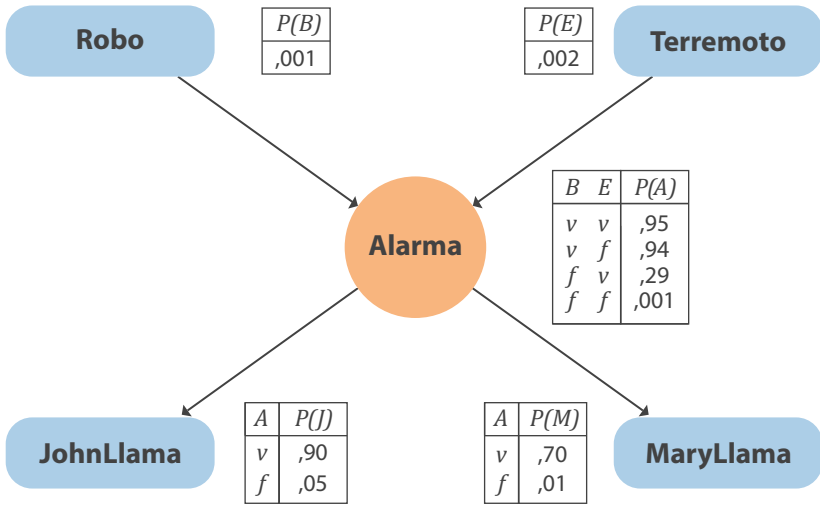


Figura 19. Red bayesiana "alarma".

Así, la red representa nuestras suposiciones de que ellos no perciben ningún robo directamente, que ellos no perciben terremotos menores, y que ellos no dialogan antes de la llamada.

La red no tiene nodos que correspondan a la música alta que está escuchando en ese momento Mary o al timbre del teléfono y la confusión de John. Estos factores están resumidos en la incertidumbre asociada a los enlaces de Alarma hasta JohnLlama y MaryLlama. Esto muestra tanto la pereza como la ignorancia en funcionamiento: sería mucho trabajo descubrir porqué esos factores serían más o menos posibles en cualquier caso particular, y de todas formas no tenemos un modo razonable para obtener la información relevante.

Las probabilidades resumen un conjunto potencialmente infinito de circunstancias en las que la alarma podría fallar y no se dispara (humedad alta, fallo de la corriente, baterías agotadas, cables cortados, un ratón en el interior atasca la campana, etc.) o John o Mary podrían fallar y no llaman para informarnos (salir a comer, estar de vacaciones, temporalmente sordos, pasa un helicóptero, etc.). De este modo, un agente pequeño puede hacer frente a un mundo muy grande, al menos aproximadamente. El grado de aproximación puede mejorarse si introducimos información relevante adicional.

En cuanto a las distribuciones condicionales que se muestran en la Figura 19. Cada distribución se muestra como una tabla de probabilidad condicional (TPC). Cada fila de una TPC contiene la probabilidad condicional de cada valor del nodo para un caso de condicionamiento. Un caso de condicionamiento es una combinación posible de valores de los nodos padres. Cada fila debe sumar 1, ya que las entradas representan un conjunto exhaustivo de casos para la variable. Para variables booleanas, una vez que conoce que la probabilidad de un valor verdad es  $p$ , la probabilidad de falso debe ser  $1-p$ , de esta manera omitimos el segundo número a menudo, como en la Figura 19. En general, una tabla para una variable booleana con  $k$  padres booleanos contiene  $2^k$  probabilidades explícitas por

separado. Un nodo sin padres tiene sólo una fila, que representa las probabilidades a priori de cada posible valor de la variable.

Una red bayesiana proporciona una descripción completa del dominio. Cada entrada de la distribución de probabilidad conjunta puede calcularse a partir de la información de la red. Una entrada genérica en la distribución conjunta es la probabilidad de una conjunción de asignaciones concretas a cada variable, tal como  $P(X_1=x_1 \wedge \dots \wedge X_n=x_n)$ . Para esta usaremos la notación abreviada  $P(x_1, \dots, x_n)$ . El valor de esta entrada está dado por la fórmula:

$$P(x_1, \dots, x_n) = \prod_{i=1}^n P(x_i \mid \text{padres}(X_i))$$

donde  $\text{padres}(X_i)$  denota los valores específicos de las variables de  $\text{padres}(X_i)$ . Así, cada entrada de la distribución conjunta está representada por el producto de los elementos apropiados de las tablas de las probabilidades condicionales (TPCs) de la red bayesiana. Las TPCs proporcionan así una representación descompuesta de la distribución conjunta. Para ilustrar esto, podemos calcular la probabilidad de que la alarma ha sonado, pero no ha ocurrido ni un robo ni un terremoto, y tanto John como Mary llaman. Usamos nombres para las variables con una sola letra:

$$P(j \wedge m \wedge a \wedge \neg b \wedge \neg e) = P(j \mid a)P(m \mid a)P(a \mid \neg b \wedge \neg e)P(\neg b)P(\neg e) = 0,90 * 0,70 * 0,001 * 0,999 * 0,998 = 0,00062.$$

Como ya se explicó, la distribución conjunta completa puede usarse para responder cualquier pregunta sobre el dominio. Si una red bayesiana es una representación de la distribución conjunta, entonces esta puede también ser utilizada para responder cualquier pregunta, a partir de la sumatoria de todas las entradas conjuntas relevantes.

La buena noticia para la creación y la inferencia en redes bayesianas es que ya existen bibliotecas en varios lenguajes de programación que lo traen implementado. Conocer su funcionamiento básico es útil para configurar y utilizarlas para la solución de problemas.

## La incertidumbre mediante factores de certeza

Uno de los modelos más sencillos surgidos en un momento en el que se consideraba un problema la escasez de información necesaria en los sistemas probabilistas, fue la del tratamiento de la incertidumbre en reglas de producción mediante factores de certeza. El modelo basado en factores de certeza (FC) introduce los conceptos de credibilidad y descredibilidad. Estos conceptos son mutuamente independientes y por eso no pueden ser combinados.

Los Factores de Certeza para el cálculo de la incertidumbre están basados en el enfoque bayesiano, pero no requiere una base de datos estadística. Entre las razones que explican su amplio uso en los años 70 y 80 fue la de ser utilizado satisfactoriamente en problemas causales y de diagnóstico, usar valores de certidumbre brindados por expertos y formularse mediante expresiones no complejas.

Este método se basa en definir un valor de certeza en la veracidad de las evidencias, las hipótesis y las reglas que se definen en el sistema. Un modelo básico de manipulación de la incertidumbre basado en factores de certeza se define de la forma siguiente:

Cada regla tiene asociado un factor de certeza  $CF$

### 1. Cálculo de certidumbre de la condición:

Si la condición  $C$  es una conjunción de premisas  $P_1, \dots, P_n$  con factores de certidumbre  $CF(P_1), \dots, CF(P_n)$  entonces  $CF(C) = \min(CF(P_1), \dots, CF(P_n))$ .



Si la condición C es una disyunción de premisas  $P_1, \dots, P_n$  con factores de certidumbre  $CF(P_1), \dots, CF(P_n)$  entonces  $CF(C) = \max(CF(P_1), \dots, CF(P_n))$ .

Si existen combinaciones de conjunciones y disyunciones se determinan de forma independiente.

2. Cálculo de la certidumbre de la conclusión de una regla.

$$CF(\text{Conclusión}) = CF(\text{condición}) * CF(\text{regla})$$

3. Si una conclusión se deriva por dos reglas, teniendo en cada derivación una certidumbre diferente, entonces la certidumbre de la conclusión es:

$$CF(CF_1, CF_2) = \begin{cases} CF_1 + CF_2(1 - CF_1) & \text{si } CF_1 \text{ y } CF_2 > 0 \\ CF(-CF_1 - CF_2) & \text{si } CF_1 \text{ y } CF_2 < 0 \\ \frac{CF_1 + CF_2}{1 - \min(|CF_1|, |CF_2|)} & \text{si } CF_1 \text{ o } CF_2 < 0 \end{cases}$$

Siguiendo estas ideas surgieron sistemas expertos desde finales de los años 70, que hicieron aportes a estos cálculos. Este modelo ha perdido popularidad por la necesidad de contar con información especializada de expertos y presentar varias insuficiencias en la combinación de las certezas. Sin embargo, sigue siendo útil en la práctica, sobre todo en sistemas no muy complejos y donde se puedan determinar los valores de certeza por alguna vía confiable.

Veamos un ejemplo sencillo:

Sean las reglas

R1: IF fiebre (F) and tos (T) THEN COVID\_19 (C) CF(0.7)

R2: IF fiebre (F) or tos (T) THEN COVID\_19 (C) CF(0.6)

Conociendo que son verdaderas las evidencias F y T con certidumbres

$$CF(F) = 0.8 \quad CF(T) = 0.5$$

$$CF(F \text{ y } T) = \min(CF(F), CF(T)) = \min(0.8, 0.5) = 0.5$$

$$CF(F \text{ o } T) = \max(CF(F), CF(T)) = \max(0.8, 0.5) = 0.8$$

$$\begin{aligned} \text{CF(C):R1} &= 0.5 * 0.6 = 0.3 \\ \text{CF(C):R2} &= 0.8 * 0.7 = 0.56 \end{aligned}$$

$$\begin{aligned} \text{CF(C)} &= \text{CF(C):R1} + \text{CF(C):R2}(1 - \text{CF(C):R1}) \\ &= 0.3 + 0.56 * (1 - 0.3) \\ &= 0.3 + 0.56 * 0.7 \\ &= 0.3 + 0.392 \\ &= 0.692 \end{aligned}$$

Este razonamiento se interpreta de la siguiente manera: A partir del conocimiento que se tiene representado en las reglas, si una persona tiene fiebre con una certeza de 0.8 y tos con una certeza de 0.5 entonces tiene COVID-19 con una certeza de 0.69. Es claro que un sistema simplificado como este, no toma en cuenta muchas otras variables que implicarían variar la certeza en cada caso. Por ejemplo, si estamos en una etapa epidémica de la enfermedad, estas certezas deben ser mucho más altas que si la incidencia de la enfermedad está controlada.

## Tratar con el conocimiento difuso

La mayoría de los fenómenos del mundo son imprecisos, es decir, tienen un cierto grado de borrosidad en la descripción de su naturaleza. Esta imprecisión puede estar asociada con su forma, posición, momento, color, textura, o incluso la semántica que describe qué son ellos. En cualquier caso, un mismo concepto, por ejemplo, *caliente* tendría diferentes grados de imprecisión en diferentes contextos o tiempos. No es lo mismo un día *caliente* en verano que en invierno, o un verano *caliente* en Noruega que en Cuba.

Para algunos autores (Gálvez, 1998) (R. E. Bello et al., 2002) la incertidumbre es un concepto multidimensional. Esa naturaleza multidimensional no fue reconocida mientras se concebía únicamente en términos de la teoría de las probabilidades, en la cual se manifiesta solo una de sus dimensiones. Para estos autores diversas teorías matemáticas modelan las situaciones problemáticas según la naturaleza de la incertidumbre. Cada teoría matemática es capaz de capturar solamente cierto tipo específico de incertidumbre.

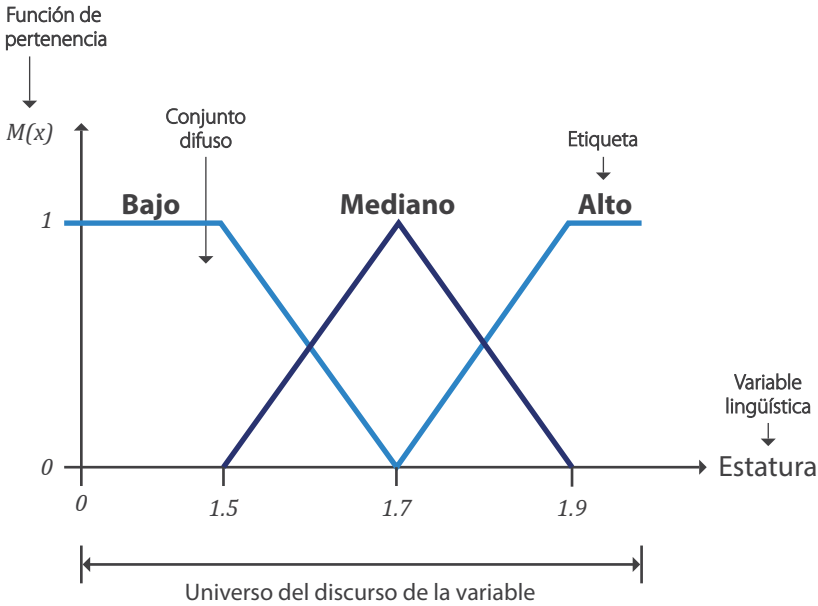
La teoría de las probabilidades tiene que ver con eventos definidos claramente (falsos o verdaderos) donde la fuente de ambigüedad es la incertidumbre en si ocurrirá o no, en si ocurrió o no. En los métodos borrosos la ambigüedad se refiere a la naturaleza del evento, en lugar de en su ocurrencia.

Los grados de veracidad, a diferencia de los grados de creencia, son la materia de la lógica borrosa o difusa. El enfoque de razonamiento difuso tiene como punto inicial la teoría de conjuntos difusos. La teoría de conjuntos clásica plantea que, para cualquier conjunto y cualquier elemento, el elemento pertenece absolutamente o no al conjunto. Un conjunto difuso, en cambio, tiene como dominio el intervalo  $[0,1]$ . Los puntos cercanos a 0 están menos en el conjunto difuso, en tanto que los puntos que se aproximan a 1 están más en dicho conjunto (R. Bello & Verdegay, 2010).

Una variable lingüística, por ejemplo, *Estatura*, puede considerarse como el conjunto de todos los términos que pueden hacer referencia a la estatura, como “Bajo”, “Mediano”, “Alto”, etc cada uno de estos términos representan conjuntos difusos. El universo de discurso  $U$  de esta variable va, en general, desde cero hasta dos metros (Figura 20).

Se pueden definir **reglas difusas** que combinan uno o más conjuntos borrosos de entrada, llamados **antecedentes o premisas**, y

se les asocian un conjunto difuso de salida, llamado **consecuente** o **consecuencia**. Los conjuntos difusos de la premisa se asocian mediante conjuntivas lógicas como y, o, etc.



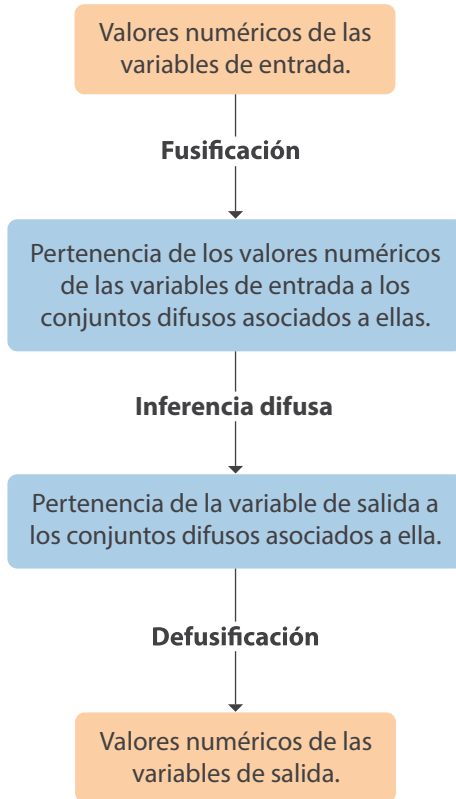
**Figura 20.** Ejemplo de variable lingüística Estatura.

Las reglas difusas permiten expresar el conocimiento que se dispone sobre la relación entre antecedentes y consecuentes. Para expresar este conocimiento de forma completa normalmente se precisa de varias reglas, que se agrupan formando una base de reglas, es decir, el conjunto de reglas que expresan las relaciones conocidas entre antecedentes y consecuentes como se vio en el apartado de representación del conocimiento en forma de reglas.

Un sistema de inferencia difuso o borroso (SIB) es un sistema computacional basado en los conceptos de la teoría de conjuntos difusos, reglas difusas y razonamiento difuso. La estructura básica de un SIB consiste de tres componentes: una base de reglas difusas, una base de datos y un mecanismo de razonamiento el cual ejecuta el procedimiento de inferencia.

Un esquema para representar el proceso que realiza un SIB parte de los valores numéricos de las variables, hasta llegar al valor numérico de la variable resultado (Figura 21). El proceso se puede resumir en los siguientes pasos:

1. Determinar la pertenencia del valor de la variable o variables de los antecedentes de las reglas difusas a los conjuntos difusos representados por las etiquetas de dichas variables. Este paso se conoce con un anglicismo bastante generalizado, el término **fusificación**.
2. Determinar los valores de pertenencia de la variable del consecuente a cada conjunto difuso asociado a ella. Esto es, realizar la **inferencia difusa**. Para ello se determina el valor del antecedente a partir de los conjuntos borrosos implicados, utilizando operaciones para la conjunción en la regla (intersección en los conjuntos), disyunción (Unión) o negación (complemento). Se calcula el valor del consecuente en cada regla, tomando en cuenta el valor del antecedente y el valor de la regla. Por último se integran los resultados de las reglas para obtener un valor global de pertenencia a cada conjunto difuso de la variable del consecuente.
3. Determinar el valor numérico de la variable del consecuente. A este proceso se le denomina concisión, aunque igualmente se utiliza el anglicismo **defusificación**.



**Figura 21.** Esquema general del funcionamiento de la Lógica Difusa.

Una de las principales dificultades en los sistemas de inferencia difuso es determinar las funciones de pertenencia a los conjuntos. Este paso se puede realizar a partir de información que se tenga sobre el dominio, ya sea por expertos o información disponible en bases de datos.

---

# APRENDIZAJE

---

La idea del aprendizaje en los agentes consiste en utilizar las percepciones no sólo para actuar, sino también para mejorar su habilidad para actuar en el futuro. El aprendizaje ocurre cuando el agente observa sus interacciones con el mundo y sus procesos de toma de decisiones. Puede ir desde la trivial memorización de las experiencias, hasta la creación de teorías científicas (Russell & Norvig, 2004). De una forma sencilla se puede decir que:

*El aprendizaje, en un agente, es la capacidad de utilizar las percepciones de su entorno para mejorar su comportamiento actual y futuro.*

Para muchos investigadores, el método más adecuado de crear sistemas novedosos es construir máquinas que aprendan y después enseñarlas. El aprendizaje permite que el agente se desarrolle en medios inicialmente desconocidos, logrando ser más competente en la medida que actúa. Los agentes o sistemas racionales de forma general, pueden diseñarse a partir de una gran variedad de componentes o módulos, y estos se pueden implementar de muchas formas, por lo que, en la práctica, existen infinidad de métodos de aprendizaje.

Para mostrar diversas formas de aprendizaje pongamos el ejemplo de un agente entrenándose para convertirse en un taxista (Russell & Norvig, 2004). Cada vez que el instructor grite ¡Frena!, el agente puede aprender una regla condición-acción que le indique cuándo frenar (1). Si se le informa al agente que ciertas imágenes de cámaras contienen autobuses, puede aprender a reconocerlos (2). Si realiza acciones y observa sus resultados, por ejemplo, frenar bruscamente en pavimento mojado, puede aprender las consecuencias de sus acciones (3). Si no recibe propina de pasajeros tras una conducción brusca, puede aprender un componente útil de utilidad para su comportamiento (4).

*El tipo de información disponible y utilizada por el agente para el aprendizaje es, por lo general, el factor más importante a la hora de determinar el tipo de problema de aprendizaje que tiene que resolver el agente. Tomando en cuenta estos criterios se reconocen tres tipos de aprendizaje: **supervisado, no supervisado y por refuerzo** (Figura 22).*

El problema de aprendizaje supervisado consiste en aprender a partir de ejemplos con sus entradas y sus salidas. Los casos (1), (2) y (3) son ejemplos de problemas de aprendizaje supervisado. En el caso (1), el agente aprende la regla condición-acción para frenar, es decir, una función que, a partir del estado, genera una salida booleana (frenar o no frenar). En el caso (2), el agente aprende una función que, a partir de una imagen, genera una salida booleana (si la imagen contiene o no un autobús). En el caso (3), aprende una función que, a partir del estado y las acciones para frenar, genera la distancia a la que debe frenar. En los casos (1) y (2), un profesor suministra el valor correcto de la salida de cada ejemplo; en el ter-



cero, el valor de la salida proviene de lo que el agente percibe. En entornos totalmente observables, el agente siempre puede observar los efectos de sus acciones, por lo tanto, puede utilizar métodos de aprendizaje supervisado para aprender a predecirlos. En entornos que son parcialmente observables, el problema es más difícil, ya que los efectos más inmediatos pueden ser invisibles.



**Figura 22.** Principales tipos de aprendizaje en la IA.

El problema de aprendizaje no supervisado consiste en aprender a partir de patrones de entradas para los que no se especifican los valores de sus salidas. Por ejemplo, el agente taxista debería desarro-

llar gradualmente los conceptos de días de tráfico bueno y de días de tráfico malo, sin que le hayan sido dados ejemplos. Un agente de aprendizaje supervisado puro no puede aprender qué hacer, porque no tiene información de lo que es una acción correcta o un estado deseable.

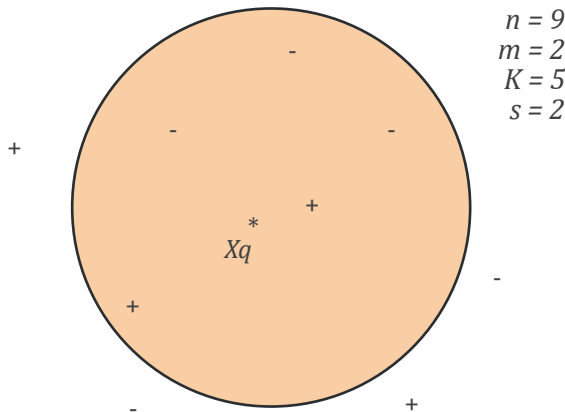
El problema del aprendizaje por refuerzo, es el más general de las tres categorías. En vez de que un profesor indique al agente qué hacer, el agente de aprendizaje por refuerzo debe aprender a partir del refuerzo. Por ejemplo, la falta de propina al final del viaje, o una gran multa por golpear la parte trasera de otro carro, le brinda al agente algunas indicaciones de que su comportamiento no es el deseable. El aprendizaje por refuerzo típicamente incluye el subproblema de aprender cómo se comporta el entorno.

La forma de representar la información aprendida, también juega un papel importante para determinar el algoritmo de aprendizaje. Cualquiera de los componentes de un agente puede ser representado usando cualquiera de los esquemas de representación mostrados en este libro.

## Aprendizaje supervisado

Por ser una de las formas más sencillas y a la vez más utilizadas en la IA, vamos a mostrar el funcionamiento general del aprendizaje supervisado. Un método de aprendizaje supervisado toma como entrada un objeto o una situación descrita a través de un conjunto de atributos y devuelve el valor previsto de la salida, dada la entrada. Los atributos de entrada pueden ser discretos o continuos. El valor de la salida puede ser a su vez discreto o continuo; aprender una función de valores discretos se denomina clasificación; aprender una función continua se denomina regresión.

Uno de los métodos más simples de clasificación es el de los *k* vecinos más cercanos (KNN por sus siglas en inglés). Este método se basa en clasificar un elemento a partir de su cercanía (semejanza) con un conjunto de elementos o problemas ya resueltos. En la Figura 23 se muestra un ejemplo con una base de ejemplos con nueve instancias ( $n=9$ ), las cuales tienen como atributos las coordenadas de posición en el espacio de dos dimensiones ( $m=2$ ), se toman los cinco elementos más semejantes ( $k=5$ ) para asignar uno de los dos valores posibles de la clase ( $s=2$ ): + o -. En este caso la clase a asignar al punto  $X_q$  sería -, ya que, de las cinco instancias más cercanas, tres de ellas son de dicha clase. De haber utilizado  $K = 1$  se hubiera asignado la clase + por ser la clase de la instancia más cercana al punto  $X_q$ .



**Figura 23.** Ejemplo de clasificación con algoritmo KNN.

El razonamiento basado en casos (RBC) pertenece a la misma familia de métodos que el KNN. Comparten como características

que analizan la base de conocimiento al llegar un nuevo problema a resolver y lo resuelven a partir de la comparación con las instancias en la base de ejemplos. La diferencia principal es que el razonamiento basado en casos utiliza una descripción más elaborada de las instancias y por tanto el método de recuperar las instancias más complejo. Generalmente el RBC utiliza medidas de semejanza y no de distancia entre las instancias.

La Base de Casos contiene las experiencias, ejemplos o casos a partir de los cuales el sistema hace sus inferencias. Tradicionalmente esta base puede ser generada por entrevistas a expertos humanos, o por un procedimiento automático o semiautomático que construye los casos desde datos existentes registrados en una base de datos.

Luego de tener la base de datos, el problema es dar solución a un nuevo caso. El proceso de recuperación consiste en determinar los casos más semejantes que se encuentran en la base de casos. Para determinar una medida de semejanza entre dos casos, se han desarrollado varias técnicas. La más sencilla consiste en contar el número de rasgos predictores similares entre ambos; sin embargo, se presenta el problema de que la importancia de los rasgos predictores puede ser diferente en distintas circunstancias y el propio dominio de los rasgos exige una comparación más detallada.

Para resolver estos problemas se definen funciones de comparación de rasgos utilizando un conjunto de heurísticas que permitan determinar cuáles rasgos predictores tienen mayor importancia o peso en la determinación del rasgo objetivo. Se formula una función de semejanza o distancia que considere la semejanza entre cada uno de los rasgos predictores, teniendo en cuenta su relevancia.

Con el valor de semejanza entre cada caso de la base y el nuevo caso a resolver y siguiendo algún criterio, se seleccionan los más semejantes. Uno de los criterios es recuperar una cantidad de casos

fija, es decir una estrategia KNN. Este criterio garantiza recuperar una cantidad de casos más semejantes, pero no toma en cuenta cuan semejantes son. De ahí que otro criterio es seleccionar los casos que tengan valor de semejanza menor que un umbral. Este criterio tiene como inconveniente que ningún caso de la base cumpla dicha restricción o que sean muchos los que cumplan con la misma. Es útil, entonces, utilizar variantes combinadas dependiendo del problema que se esté resolviendo.

Después de la determinación de los casos más semejantes, se procede a la adaptación de las soluciones contenidas en ellos. En algunas aplicaciones, estas soluciones pueden emplearse directamente como solución al nuevo problema, pero comúnmente necesitan ser modificadas.

*Un aspecto de gran interés en los Sistemas Basados en Casos es la posibilidad de justificar sus decisiones, dado que están basadas en casos reales. Este método es de los pocos que atenúa una de las cuestiones éticas abordadas de la IA. La explicabilidad de los métodos de IA es un reto en la actualidad (R. Bello, 2019) y a la vez una de sus exigencias éticas (Conferencia General, 2019).*

Para problemas con un solo rasgo objetivo es más sencillo encontrar una solución. Cuando el rasgo objetivo es nominal, puede asignarse al nuevo problema la moda, si el valor es un número real, pudiera utilizarse el promedio, incluso un promedio pesado tomando en cuenta la semejanza de cada uno al caso a resolver. Sin embargo, en problemas reales, muchas veces este proceso de adaptación es más complejo. Incluso, puede ser necesario hacer una transformación de una o varias de las soluciones de los casos más semejantes de manera que se ajuste a la nueva situación.

Una vez propuesta la solución en el proceso de adaptación, debe ser evaluada. El proceso de evaluación requiere un elevado conocimiento del dominio. Una vía de evaluar la nueva solución es aplicarla y ver qué sucede. Un agente debe ser autónomo en este proceso por tanto la evaluación, más que dar una medida de calidad, brinda criterios de evaluación. También, es conocido este proceso como la explicación del resultado. Algunos de los elementos a tomar en cuenta son las diferencias entre el caso resuelto y los utilizados en la adaptación y que llevaron a decidir la solución o dar las características del proceso, como la cantidad de casos escogidos, el valor de semejanza con ellos, el criterio de adaptación utilizado. Después de este proceso la nueva solución pudiera ser adaptada nuevamente. Por último, el nuevo problema puede ser incorporado a la base de casos a través del proceso de actualización de la memoria de casos.

*El proceso de razonamiento basado en casos puede explotar múltiples tipos de conocimiento. El aprendizaje está asociado a la selección, representación y organización de los casos, a la estrategia de recuperación, a la estrategia de adaptación y a la actualización de la base a partir de los casos resueltos.*

Aunque estos métodos, en su funcionamiento básico, analizan para cada nuevo problema todos los casos de su base de conocimiento, un gran número de métodos de aprendizaje supervisado crean un modelo a partir de los datos de entrenamiento, lo cual simplifica el trabajo a la hora de resolver los nuevos problemas. Ejemplos de estos son los métodos existentes para crear árboles de decisión, redes bayesianas, máquinas de soporte vectorial o incluso, algoritmos para entrenar redes neuronales, las cuales, en su gran mayoría, sí requieren una construcción manual.

## Redes neuronales artificiales

Para algunos investigadores, en las redes neuronales está la verdadera Inteligencia Artificial, los criterios para ello son bien fundamentados. Las redes neuronales son una simulación del funcionamiento del cerebro humano, el modelo que en la naturaleza ha demostrado ser más capaz, aunque aún poco conocido su funcionamiento interno. Si le sumamos a este intento, que sus resultados, desde el surgimiento de la IA, hasta nuestros días, han demostrado efectividad e ir evolucionando, la convierten en un área de constante desarrollo e interés. De redes neuronales se han desarrollado modelos, desde totalmente estáticos y específicos como las primeras de McCulloch y Pitts, hasta las utilizadas en nuestros días en tareas de *Big Data* o aprendizaje profundo.

Específicamente, una neurona es una célula del cerebro cuya función principal es la recogida, procesamiento y emisión de señales eléctricas. Se piensa que la capacidad de procesamiento de información del cerebro proviene principalmente de redes de este tipo de neuronas. Desde un punto de vista informal, la neurona se activa cuando una combinación lineal de sus entradas excede un determinado umbral.

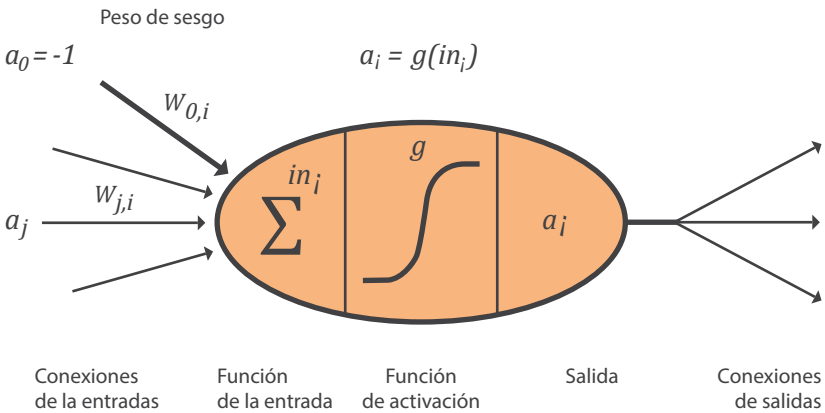
Las redes neuronales artificiales están compuestas de nodos o unidades conectadas a través de conexiones dirigidas (Figura 24). Una conexión de la unidad  $j$  a la unidad  $i$  sirve para propagar la activación  $a_j$  de  $j$  a  $i$ . Además, cada conexión tiene un peso numérico  $W_{j,i}$  asociado, que determina la fuerza y el signo de la conexión. Cada unidad  $i$  calcula una suma ponderada de sus entradas:

$$x_i = \sum_{j=0}^n W_{j,i} a_j$$

Luego aplica una función de activación  $g$  a esta suma para producir la salida:

$$a_i = g(x_i) = g\left(\sum_{j=0}^n W_{j,i} a_j\right)$$

Un punto de interés es la inclusión de un peso de sesgo  $W_{j,i}$  conectado a una entrada fija  $a_0 = -1$ .

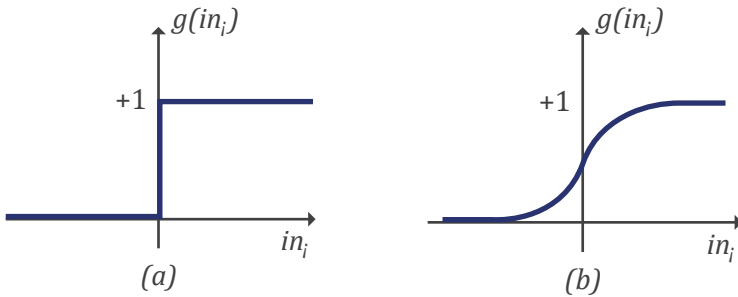


**Figura 24.** Estructura de neurona artificial.

La función de activación  $g$  se diseña con dos objetivos. Primero, queremos que la unidad esté activa (cercana a +1) cuando se proporcionen las entradas correctas, e inactiva (cercana a 0) cuando se den las entradas erróneas. Segundo, la activación tiene que ser no lineal, para evitar que la red neuronal colapse con el procesamiento. En la Figura 25 se muestran dos posibles funciones  $g$ : la función umbral y la función sigmoide (también conocida como función logística). Ambas funciones tienen un umbral de cero; los pesos de sesgo  $W_{0,i}$  constituyen el umbral real de la unidad, es decir, la unidad se activa

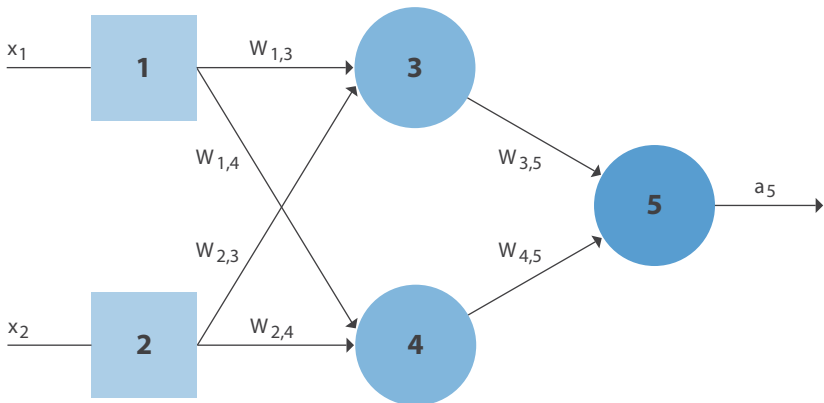


cuando la suma de los pesos de las entradas reales  $\sum_{j=1}^n W_{j,i} a_j$  (incluyendo la entrada del sesgo) excede  $W_{0,i}$ .



**Figura 25.** Ejemplos de funciones de activación de neuronas.

Una red con neuronas como estas que se han presentado con alimentación hacia delante, representa una función de sus entradas. Veamos como ejemplo la sencilla red mostrada en la Figura 26, que tiene dos unidades de entrada, dos unidades ocultas y una unidad de salida. Para mantener simplicidad, se han omitido las unidades de sesgo.



**Figura 26.** Red simple con dos entradas y una salida.

Dado un vector de entrada  $x=(x_1, x_2)$ , las activaciones de las unidades de entrada se ponen directamente  $a=(a_1, a_2)=(x_1, x_2)$ . A partir de ahí cada neurona de la capa intermedia (3 y 4) calculan las salidas  $a_3$  y  $a_4$  que son las entradas de la neurona de la capa de salida (5), la cual se activa para determinar el valor de  $a_5$ . Visto todo este proceso como una combinación lineal de toda la red quedaría:

$$a_5 = g(w_{3,5}a_3 + w_{4,5}a_4) = g(w_{3,5}g(w_{1,3}a_1 + w_{2,3}a_2) + w_{4,5}g(w_{1,4}a_1 + w_{2,4}a_2))$$

De esta forma se expresa la salida de la red como una suma  $a_5$ , en función de las entradas de la red. Es importante ver que los pesos de la red actúan como parámetros de la función; escribiendo  $w$  para los parámetros, la red calcula  $h_w(x)$ . Ajustando los pesos, cambiamos la función que representa la red. Esta es la manera en que se produce el aprendizaje en las redes neuronales.

Para resolver un problema concreto con este tipo de redes neuronales, es preciso tener claro algunos elementos. Primero, cada neurona de la capa de entrada representa una variable del problema a resolver, mientras que, en la capa de salida, cada neurona representa una variable objetivo o un valor posible de la variable objetivo (depende de cómo se modele el problema). Para determinar la cantidad de capas y neuronas en cada capa intermedia, no existe un criterio definido, generalmente, se determina siguiendo un proceso de pruebas para determinar la mejor topología para cada problema.

Una red neuronal se puede usar para clasificación o para regresión. Para clasificaciones booleanas con entradas continuas (por ejemplo, con unidades sigmoides), es tradicional tener una única unidad de salida, con un valor por encima de 0,5 interpretado como una clase y con un valor por debajo de 0,5 como otra. Para clasificación en  $k$ -clases, se puede dividir el rango de la unidad de salida en  $k$  partes, pero es más común tener  $k$  unidades de salida separadas,

donde el valor de cada una representa la verosimilitud relativa de esa clase dada una entrada actual. Una red con todas las entradas conectadas directamente a las salidas se denomina red neuronal de una sola capa, o red Perceptrón.

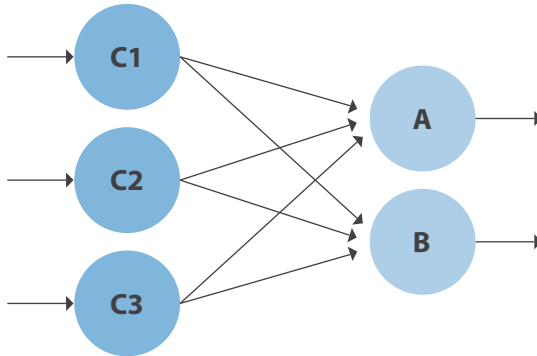
Para ilustrar cómo se utiliza una red neuronal en un problema específico, mostraremos un ejemplo sencillo.

En un laboratorio de biotecnología se enfrentan a un problema de clasificación. Se sabe que existen tres características que, su aparición o no, pueden definir el tipo de virus. La Tabla 4 presenta la información que hasta el momento tienen, con la cual desean crear un modelo de red neuronal Perceptrón Simple para predecir la clase del virus en nuevas muestras que puedan llegar.

**Tabla 4.** Datos del problema de clasificación

CARACTERÍSTICA 1	CARACTERÍSTICA 2	CARACTERÍSTICA 3	TIPO DE VIRUS
si	no	no	A
si	no	si	A
no	no	no	B
si	si	si	B

Lo primero es saber que necesitamos tres neuronas de entrada, una para cada característica de los virus, las cuales tendrán dos posibles valores (si o no). De igual forma debemos incluir en la red dos neuronas de salida, una para cada clase de virus. Siendo una red neuronal simple de tipo perceptrón quedaría como en la Figura 27.



**Figura 27.** Red neuronal del ejemplo del virus.

Con esta estructura se asignan pesos iniciales a las conexiones entre cada neurona de entrada y cada neurona de salida y se inicia un proceso de entrenamiento de la red a partir de los datos de entrada y salida disponibles en la Tabla 4. Al concluir el entrenamiento ya tendremos una red neuronal con la que podemos, a partir de los datos de un nuevo virus, es decir, de conocer si tiene presente o no sus tres características, determinar si pertenece al tipo A o B.

## Aprendizaje no supervisado y aprendizaje reforzado

El problema de aprendizaje no supervisado consiste en aprender a partir de patrones de entradas para los que no se especifican los valores de sus salidas. Uno de los problemas más representativos del aprendizaje no supervisado es el agrupamiento (*clustering* en inglés). El agrupamiento no supervisado es el problema de discernir múltiples categorías en una colección de objetos. Digamos que proporciona una hipótesis que explica un agrupamiento de los datos.

Una familia de métodos para esta tarea es la que realiza un particionamiento de los datos. Dado  $D$ , un conjunto de datos con  $n$  instancias y  $k$  el número de grupos a formar. Un algoritmo de particionamiento organiza las  $n$  instancias en  $k$  particiones ( $k \leq n$ ), donde cada partición representa un grupo (*cluster*).

El algoritmo k-medias busca una partición óptima de los datos minimizando el criterio de la suma del error cuadrado con un procedimiento iterativo de optimización, el cual pertenece a la categoría de algoritmos de búsqueda escalador de colina. Para su realización es imprescindible definir por adelantado el número de grupos que se desea en un parámetro  $k$ , para esto se seleccionan aleatoriamente  $k$  elementos que representan la media o centro de cada grupo. Posteriormente cada ejemplo o instancia es asignado al grupo más cercano acorde con la distancia Euclidiana, de Manhattan, o cualquier otra, que la separa de él. Para cada uno de los grupos así construidos, se calcula el centroide de todas sus instancias. Estos centroides son tomados como los nuevos centros de sus respectivos grupos. Finalmente se repite el proceso completo con los nuevos centros de los grupos. La iteración continúa hasta que se repite la asignación de los mismos ejemplos a los mismos grupos, ya que los puntos centrales de los grupos se han estabilizado y permanecerán invariables después de cada iteración.

El algoritmo k-medias se encuentra con un problema cuando los atributos no son numéricos, ya que en ese caso la distancia entre ejemplos no está tan clara. Para resolver este problema surge el agrupamiento conceptual, que utiliza para justificar la necesidad de un agrupamiento cualitativo frente al agrupamiento cuantitativo. En este tipo de agrupamiento, una partición de los datos es buena si cada clase tiene una buena interpretación conceptual.

El método COBWEB forma los conceptos por agrupación de ejemplos con atributos similares. El objetivo es hallar un conjunto de clases o grupos que maximice la utilidad de la categoría.

Los dos algoritmos de agrupamiento anteriores se caracterizan por una dependencia entre el resultado y el orden de los ejemplos y por una tendencia al sobreajuste. Una aproximación estadística para la tarea de agrupamiento busca el grupo de grupos más probables dados los datos. Los ejemplos tienen ciertas probabilidades de pertenecer a un grupo. La base de este tipo de agrupamiento se encuentra en el modelo estadístico llamado mezcla de distribuciones. Cada distribución representa la probabilidad de que un objeto tenga un conjunto particular de pares atributo-valor, si se supiera que es miembro de ese grupo. Se tienen  $k$  distribuciones de probabilidad que representan los  $k$  grupos. El algoritmo de la máxima expectativa (EM por sus siglas en inglés) empieza adivinando los parámetros de las distribuciones (es decir, las probabilidades de que un objeto pertenezca a una clase) y, a continuación, los utiliza para calcular las probabilidades de que cada objeto pertenezca a un grupo y usa esas probabilidades para re-estimar los parámetros de las probabilidades, hasta converger.

Estos métodos de aprendizaje no supervisado han tenido un gran auge en nuestros días. Con el cúmulo de información que se dispone en redes digitales de comunicación, en aplicaciones y sitios en la web de forma general, se ha hecho una necesidad extraer conocimiento útil para tomar decisiones. En la utilización de estos algoritmos y la recopilación de los datos que utilizan, se refuerza la necesidad de cumplir con los principios éticos definidos por la UNESCO para el desarrollo de los sistemas con Inteligencia Artificial.

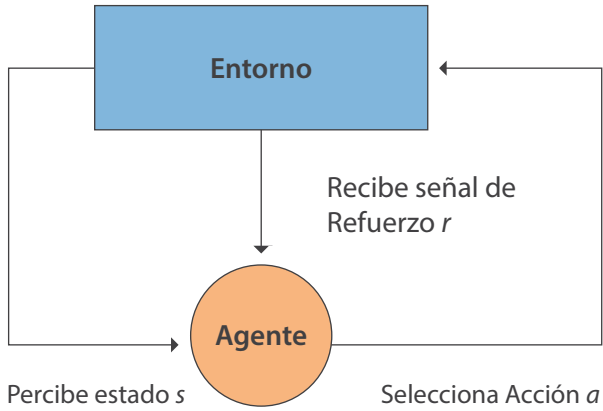
El otro tipo reconocido de aprendizaje automático es el aprendizaje por refuerzo. En este tipo de aprendizaje el agente debe aprender qué hacer, o sea, decidir las acciones a ejecutar, a partir de las situaciones del entorno. De esta forma deberá maximizar una señal de recompensa numérica. En las demás formas de aprendizaje automático se le dice al agente qué acciones debe realizar, en el reforzado debe descubrir, mediante prueba y error, qué acciones producen la mayor recompensa. En los casos más interesantes y difíciles, las acciones pueden afectar no sólo a la recompensa inmediata sino también a la siguiente situación y, a través de ella, a todas las recompensas posteriores. Estos son los rasgos más importantes del aprendizaje por refuerzo.

*Los métodos de aprendizaje por refuerzo exploran el entorno a lo largo del tiempo para llegar a una política deseada (Martínez, 2012).*

La Figura 28 muestra un modelo general de aprendizaje por refuerzo. En cada paso de la interacción, el agente percibe el estado actual  $s$  de su entorno, y luego selecciona una acción  $a$  para cambiar este estado. Esta transición genera una señal de refuerzo  $r$ , que es recibida por el agente. La tarea del agente es aprender una política para elegir acciones en cada estado para recibir la máxima recompensa acumulada a largo plazo.

Este tipo de aprendizaje proporciona un enfoque flexible para el diseño de agentes racionales en situaciones en las que otro tipo de aprendizaje no es práctico. Puede aplicarse a problemas para los que no se dispone de conocimientos sobre el dominio de aplicación o cuya obtención es costosa. Una tarea común de aprendizaje por refuerzo es el control de robots. En el diseño de robots autónomos, normalmente, se carece de conocimientos suficientes sobre el en-

torno para planificar las acciones, o datos previos para diseñar una política de control del robot. En este caso, el objetivo es permitir al robot generar políticas de decisión eficaces a medida que explora su entorno (Martínez, 2012).



**Figura 28.** Modelo general de aprendizaje por refuerzo.





## CONSIDERACIONES FINALES

---

La Inteligencia Artificial abarca una gran cantidad de técnicas y estrategias encaminadas a resolver problemas para los cuales no existe un algoritmo definido o estos se hacen computacionalmente intratables. Buscando organizar las temáticas que estudia, se agruparon en cuatro núcleos de conocimiento: El razonamiento, la representación del conocimiento, el tratamiento de la incertidumbre y el aprendizaje.

El razonamiento en IA va dirigido a buscar soluciones, no siempre mediante un algoritmo predefinido, sino utilizando estrategias que permitan acercarse a la solución deseada. Un método muy particular del razonamiento en la IA es la identificación de determinados criterios que evalúen la posibilidad de que una búsqueda vaya en la dirección correcta, a lo cual se le denomina heurística.

La Inteligencia Artificial trabaja con el conocimiento, para lo cual requiere de vías que permitan su estructuración mediante una notación suficientemente precisa como para ser utilizado por el sistema. Aunque sigue siendo interés de la IA encontrar vías para representarlo todo, aún hoy se limitan a un conjunto de formas generales de representación del conocimiento sobre determinados dominios de aplicación. La tarea sigue siendo encontrar para cada caso, la

mejor manera de estructurar el conocimiento para ser utilizado en el comportamiento del sistema.

La incertidumbre está presente en las informaciones y los datos con que trabajan los sistemas computacionales. La IA se encarga de lidiar con esa incertidumbre para tomar decisiones racionales. La teoría de la probabilidad, aunque ha tenido sus detractores, sigue siendo la más aceptada por su relación con el concepto de creencia en los agentes. No se debe olvidar que la incertidumbre tiene aristas diferentes, donde otras teorías, como la lógica difusa, juegan un papel esencial.

El aprendizaje de un sistema o agente permite utilizar las percepciones realizadas del entorno, para mejorar su comportamiento. Esta posición va más allá del análisis tradicional que se ha defendido del aprendizaje automático dentro de la IA, donde se clasifica en Supervisado, No Supervisado y Reforzado. Partiendo de la memorización, cualquier técnica o estrategia utilizada por el sistema para mejorar su comportamiento, puede considerarse aprendizaje.



## GLOSARIO DE TÉRMINOS

---

**Acciones (de un agente):** Evidencias del comportamiento del agente en su entorno.

**Agente (racional):** Entidad computacional que se comporta a partir de maximizar una medida de rendimiento cuyo valor depende de las percepciones realizadas de su entorno.

**Aplicaciones (de la IA):** Áreas donde se evidencia la utilización práctica de los conocimientos sobre una materia, relacionándose con otras materias científicas o ingenieriles. Las aplicaciones de la IA incluyen otras ramas de las ciencias de la computación, otras ciencias y sectores específicos de la economía y la sociedad. Resaltan en nuestros días el procesamiento de lenguaje natural, la minería de datos, los videojuegos y la robótica.

**Aprendizaje (en un agente):** Capacidad de utilizar las percepciones de su entorno para mejorar su comportamiento actual y futuro.

**Aprendizaje no supervisado:** El sistema se comporta a partir de patrones de entradas para los que no se especifican los valores de sus salidas.

**Aprendizaje reforzado:** El sistema actúa a partir de la información que obtiene de su propio comportamiento en el entorno.

**Aprendizaje supervisado:** El sistema se comporta a partir de la utilización de ejemplos con sus entradas y sus salidas.

**Creencia (de un agente):** Representan el conocimiento que se tiene del entorno. Desde un punto de vista informático, son la forma de representar el estado del entorno.

**Comportamiento racional (de un agente):** Función del agente que proyecta una percepción dada, en una acción.

**Deseos (de un agente):** Son los objetivos del agente. Un objetivo se asocia al valor de una variable, un registro, o una expresión simbólica en alguna lógica. Representa un estado final deseado.

**Entorno (de un sistema computacional):** Conjunto de datos, informaciones, conocimiento, circunstancias, factores, etc., que influyen en el estado o desarrollo de un sistema computacional.

**Heurística:** Criterio que evalúa la posibilidad de que una búsqueda vaya en la dirección correcta.

**Incertidumbre (en IA):** Falta de conocimiento sobre el entorno.

**Inteligencia Artificial:** Es la Ciencia de la Computación encargada de aplicar métodos de representación del conocimiento, razonamiento, tratamiento de la incertidumbre y aprendizaje, en el desarrollo de sistemas informáticos con comportamiento racional.

**Intenciones (de un agente):** Conjunto de planes especificados por el agente para alcanzar los objetivos propuestos, a partir de las creencias existentes.

**Medida de rendimiento (de un agente):** Criterio utilizado para valorar el desempeño del agente en su entorno.

**Minería de datos:** Aplicación práctica de la Inteligencia Artificial. Usualmente se asocia al aprendizaje automático, aunque realmente requiere de los cuatro núcleos de conocimiento de la IA.

**Núcleo de conocimientos (de la IA):** Temáticas básicas en las que se estructura una materia o disciplina para su estudio. Los núcleos, estrechamente relacionados entre sí, de la IA son: la Representación del Conocimiento, el Razonamiento, el Tratamiento de la Incertidumbre y el Aprendizaje.

**Percepción (de un agente):** Son las entradas que recibe el agente en cualquier instante. La secuencia de percepciones de un agente refleja el historial completo de lo que el agente ha recibido.

**Procesamiento de lenguaje natural:** Aplicación de la IA encargada de analizar texto o voz utilizando un conjunto de teorías y tecnologías. Incluye la comprensión del lenguaje y la generación del lenguaje. Aunque algunos lo consideran un núcleo de conocimiento de la IA, lo cierto es que requiere de los cuatro núcleos de conocimiento de la IA y otras áreas para ser abordado.

**Prueba de Turing:** Un sistema supera la prueba si un evaluador humano no es capaz de distinguir si las respuestas, a una serie de preguntas planteadas, son de una persona o del sistema.

**Racionalidad:** Se considera que un sistema o agente es racional si hace lo correcto, en función de su conocimiento.

**Razonamiento (computacional):** Proceso general llevado a cabo por un sistema para comportarse de manera racional a partir del conocimiento que tenga de su entorno.

**Red neuronal artificial:** Modelo matemático desarrollado en el área de la Inteligencia Artificial que trata de representar el funcionamiento del cerebro. Se utiliza para tareas de clasificación y

regresión fundamentalmente, incluye un proceso previo de entrenamiento. Ha acompañado los principales resultados de la IA desde sus inicios hasta nuestros días, por lo que es una de las técnicas más reconocidas en este campo.

**Representación del conocimiento:** Proceso de estructurar el conocimiento mediante una notación suficientemente precisa como para ser utilizado por el sistema en su comportamiento.

**Robótica:** Área de aplicación de la IA con mayor complejidad. Integra otras aplicaciones directas de la IA como la visión por computador y el procesamiento de lenguaje natural, así como acciones complejas como el movimiento, equilibrio, tacto, entre otras. Requiere integración con áreas como la mecánica, la electrónica y muchas otras ciencias e ingenierías.

**Sistema multiagente:** Sistema computacional donde varios agentes actúan en el mismo entorno para dar cumplimiento a tareas específicas.

**Tratamiento de la incertidumbre:** Proceso de identificar criterios para tomar decisiones al representar y razonar cuando no se tiene todo el conocimiento sobre el entorno.

**Utilidad (de un agente):** Medida de rendimiento que indica lo deseables que son los estados posibles del agente y las acciones que va a ejecutar para llegar a ellos.

**Visión por computador:** Aplicación de la IA que analiza imágenes o secuencia de imágenes para tomar decisiones sobre ellas. Requiere integrar otras áreas como los gráficos por computador y la física, en muchos casos, además, la biología e incluso la psicología.

---

## REFERENCIAS

---

- Álvarez, E. (1989).** Modelos probabilísticos para utilización en sistemas expertos. Universidad de Cantabria.
- Bello, R. (2019).** Trends of Artificial Intelligence: challenge in Education. International Conference on Interactive Collaborative and Blended Learning, 51. Santiago de Cuba.
- Bello, R. E., García, Z. Z., García, M. M., & Reynoso, A. (2002).** Aplicaciones de la Inteligencia Artificial (Primera ed). Guadalajara.
- Bello, R., & Verdegay, J. L. (2010).** Los conjuntos aproximados en el contexto de la Soft Computing. Revista Cubana de Ciencias Informáticas, 4(1-2), 5-24.
- Conferencia General. (2019).** Estudio preliminar sobre un posible instrumento normativo relativo a la ética de la inteligencia artificial. Paris: UNESCO.
- Gálvez, D. (1998).** Sistemas Basados en el Conocimiento. Santa Clara. Cuba: Grupo de Investigación en Inteligencia Artificial. Universidad Central de Las Villas.
- Guerrero, R. S. (2012).** Ontología para la representación de las preferencias del estudiante en la actividad de aprendizaje en entornos virtuales. Universidad de La Habana.

- Martínez, Y. (2012).** A Generic Multi-Agent Reinforcement Learning Approach for Scheduling Problems. Vrije Universiteit Brussel.
- Russell, S., & Norvig, P. (2004).** Inteligencia Artificial. Un enfoque moderno (2da Edició). Madrid: Pearson Educación.





“¿Qué estudia la Inteligencia Artificial?” es el tercero de cuatro libros que tienen como objetivo principal propiciar la incorporación de conceptos e ideas generales sobre Inteligencia Artificial en las escuelas. Va dirigido a las temáticas específicas que aborda la Inteligencia Artificial desde la ciencia. Para su estudio se han organizado en cuatro núcleos básicos de conocimiento: La representación del conocimiento, el razonamiento, el tratamiento de la incertidumbre y el aprendizaje.

Se exponen los elementos generales del razonamiento computacional y algunas de las estrategias específicas de la Inteligencia Artificial. Aborda algunas de las formas de representación del conocimiento que necesitan los sistemas inteligentes y algunos criterios para el tratamiento de la incertidumbre presente en los problemas que resuelve. Presenta los elementos generales del aprendizaje y los tres tipos de aprendizaje automático que se reconocen en la bibliografía especializada.

Pretendemos que esta obra sirva de base para contribuciones teóricas y prácticas mediante el desarrollo de actividades educativas curriculares y no curriculares para las instituciones educativas, escuelas, universidades, familias, comunidades y la sociedad en general, así como para las instituciones y organizaciones gubernamentales y no gubernamentales; y de todos los sectores e instituciones de la sociedad. No se trata de enseñar a desarrollar aplicaciones de Inteligencia Artificial, sino, de contribuir al desarrollo del pensamiento computacional y nociones básicas necesarias para incentivar el estudio y profundización en IA con un posicionamiento ético, crítico y creativo, desde edades tempranas.

ISBN: 978-959-18-1342-8



9 789591 813428