



unesco



# APLICACIONES DE LA INTELIGENCIA ARTIFICIAL



Yuniesky Coca Bergolla  
Cinthia Cuza Soca  
Miguel Llivina Lavigne





# APLICACIONES DE LA INTELIGENCIA ARTIFICIAL

**Yuniesky Coca Bergolla**  
**Cinthia Cuza Soca**  
**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: 978-959-18-1343-5**

Sello Editor EDUCACIÓN CUBANA

Dirección de Ciencia y Técnica - MINED

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



# ÍNDICE

INTRODUCCIÓN.....	5
SISTEMAS BASADOS EN EL CONOCIMIENTO.....	7
■ Presentación del problema.....	10
■ Solución.....	11
Representación del conocimiento.....	13
Razonamiento.....	15
Tratamiento de la incertidumbre.....	20
Aprendizaje.....	21
MINERÍA DE DATOS.....	23
■ Presentación del problema.....	25
■ Solución.....	25
Representación del conocimiento.....	26
Razonamiento.....	28
Tratamiento de incertidumbre.....	29
Aprendizaje.....	35

PROCESAMIENTO DE LENGUAJE NATURAL .....	41
■ Presentación del problema .....	42
■ Solución .....	43
Representación del conocimiento .....	47
Razonamiento .....	51
Tratamiento de la incertidumbre .....	53
Aprendizaje .....	54
VIDEOJUEGOS .....	57
■ Presentación del problema .....	58
■ Solución .....	59
Representación del conocimiento .....	61
Razonamiento .....	62
Tratamiento de incertidumbre .....	64
Aprendizaje .....	65
CONSIDERACIONES FINALES .....	67
GLOSARIO DE TÉRMINOS .....	71
REFERENCIAS .....	75

---

# INTRODUCCIÓN

---

Este libro es una publicación científica, didáctica, metodológica y educativa, destinada a educadores en general y otros profesionales. Es el último de cuatro libros que tienen como objetivo principal propiciar la incorporación de conceptos e ideas generales sobre Inteligencia Artificial (IA) en las escuelas, familias y comunidades. Este cuarto libro va dirigido a mostrar algunas de las aplicaciones de la Inteligencia Artificial y cómo se reflejan sus cuatro núcleos de conocimiento en la práctica.

Cada capítulo se organiza de manera que se mostrarán ejemplos de problemas y su solución con IA. Primeramente se presenta una introducción sobre el tipo de sistema a desarrollar, seguidamente el problema específico a resolver, la solución general propuesta y de forma particular cómo se abordaron los cuatro núcleos de conocimiento de la IA en cada solución. Todos los ejemplos presentados son aplicaciones reales desarrolladas como trabajos de diploma de estudiantes de 5to año de la carrera Ingeniería en Ciencias Informáticas.

Específicamente el primer capítulo expone un sistema basado en conocimiento. Estos están tan estrechamente relacionados con la IA que en muchas asignaturas de carreras afines a la informática se in-

cluye como un tema a estudiar de la IA. El segundo capítulo presenta un sistema multiagente para tareas de minería de datos. Se utiliza una metodología específica para desarrollar el sistema y se utiliza una biblioteca de algoritmos de IA, de las tantas que existen hoy en diversos lenguajes de programación. El tercer capítulo aborda uno de los problemas más complejos y que sigue siendo un reto para la IA en nuestros días, el procesamiento de lenguaje natural. Se aborda desde el desarrollo de un sistema tutorial inteligente en forma de chat que brinda ayuda y evalúa el avance de los estudiantes. El cuarto capítulo presenta un tema de gran interés para los jóvenes, los videojuegos. Además, es una de las tareas de mayor integralidad en la aplicación de la IA, donde muchas veces se prueban los resultados científicos. Finalmente, se pone a disposición de los lectores un Glosario de Términos y la Bibliografía.

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, 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, 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.



---

# SISTEMAS BASADOS EN EL CONOCIMIENTO

---

En la IA, los métodos de solución de problemas basados en identificar estados y transiciones entre ellos, dejaron de ser suficientes para resolver los problemas que iban apareciendo en la construcción de nuevas aplicaciones. Se hizo necesario conocimiento específico sobre el problema y los dominios de aplicación de interés. Este reconocimiento, por los años 70 del siglo pasado, condujo al desarrollo de los Sistemas Basados en el Conocimiento (SBC) (Gálvez, 1998).

En principio, el conocimiento representado en los SBC es el de los expertos en un dominio específico. Este puede ser expresado mediante relaciones de causa y efecto, las cuales se originan a partir de la experiencia pasada y pueden considerarse heurísticas. Representan conocimiento informal o atajos, que permiten a un experto encontrar una solución a un problema, sin tener que realizar un análisis detallado de una situación particular. Para esto es necesario un análisis de algún problema resuelto de forma exitosa anteriormente o a relaciones que han sido aprendidas al no poder resolver un problema similar. El experto puede no recordar todos los detalles del análisis del problema original, pero puede reconocer el enfoque aplicado.

El conocimiento de un experto sobre un dominio específico puede ser público o privado. El conocimiento público incluye las definiciones, hechos y teorías publicadas. Pero la experticidad usualmente incluye más que esta clase de conocimiento.

*La ética juega un papel fundamental en el conocimiento que un experto brinda para la construcción de un SBC. El reconocimiento de los que aportaron es esencial, incluso para corroborar la calidad del sistema construido.*

Se reconocen en los SBC tres componentes principales, la interfaz de usuario, la base de conocimiento y la máquina de inferencia. A través de la interfaz de usuario se realiza la entrada y salida de información al sistema. En la base de conocimiento es donde se representa el conocimiento sobre el dominio de aplicación. Por último, la máquina de inferencia es el mecanismo que garantiza el proceso de razonamiento que se lleva a cabo. Esta visión clásica, podemos decir, se refiere a sistemas tradicionales surgidos desde los años 80 o un poco antes y que hoy siguen siendo útiles en infinidad de problemas prácticos (Tabla 1) (Gálvez, 1998).

**Tabla 1.** Algunos tipos clásicos de sistemas basados en conocimiento

NOMBRE	FORMA DE REPRESENTACIÓN DEL CONOCIMIENTO	MÉTODO DE SOLUCIÓN DEL PROBLEMA	FUENTES DE CONOCIMIENTO
Sistemas basados en reglas (SBR)	Reglas de producción	Usualmente búsqueda primero en profundidad con dirección backward o forward	Expertos, publicaciones, ejemplos

NOMBRE	FORMA DE REPRESENTACIÓN DEL CONOCIMIENTO	MÉTODO DE SOLUCIÓN DEL PROBLEMA	FUENTES DE CONOCIMIENTO
Sistemas basados en frames (SBF)	Frames	Herencia y procedimientos adjuntos	Expertos, publicaciones, ejemplos
Sistemas basados en casos (SBCasos)	Casos	Razonamiento basado en casos (búsqueda por semejanza y adaptación de las soluciones)	Ejemplos
Sistemas basados en probabilidades (SBP)	Probabilidades o frecuencias	Teorema de Bayes y otras técnicas de inferencia estadística	Ejemplos
Redes expertas	Pesos y alguna otra FRC	Cálculo de niveles de activación de las neuronas	Ejemplos
Sistemas basados en modelos	Modelo del artefacto	Razonamiento basado en modelos	Esquemas estructurales y funcionales del artefacto

En nuestros días, más que extraer el conocimiento de expertos humanos, los esfuerzos se han dirigido a extraerlo de grandes bases de datos de información. Estos sistemas han evolucionado a sistemas más complejos que utilizan métodos específicos de la IA o la combinación de varios. Por ejemplo, la aplicación de algoritmos de aprendizaje automático para extraer conocimiento, la utilización de ontologías para representar el conocimiento y realizar razonamientos que expresen relaciones entre conceptos y otros muchos.

Los sistemas basados en el conocimiento, de forma general, utilizan los cuatro núcleos de conocimiento de la IA. El conocimiento se representa de diversas formas, inicialmente en forma de reglas o

de casos previos, pero hoy se ha extendido a muchas otras formas, incluso particulares para cada aplicación. El mecanismo de inferencia, encargado del proceso de razonamiento, incluye el tratamiento a la incertidumbre y vías para mejorar el desempeño futuro.

En sus inicio estos sistemas incluían un proceso de justificación o explicación del razonamiento llevado a cabo. La utilización de mecanismos más complejos, influyó en hacer más oscuro el proceso de razonamiento. Hoy es un reto hacer el razonamiento en estos sistemas más transparente al usuario, donde sea capaz de explicar cómo llegó a cada resultado.

La utilidad de los sistemas basados en el conocimiento se mantiene aún hoy. Con la ayuda de estos sistemas, personas con poca experiencia pueden resolver problemas que requieren un conocimiento especializado. Además, pueden obtenerse conclusiones y resolver problemas utilizando un cúmulo de información y conocimiento imposible de asimilar por una persona, por cuestiones de tiempo sobre todo.

## Presentación del problema

Una de las preocupaciones constantes de las universidades y centros de estudios superiores es estar a la vanguardia en los métodos de enseñanza y ofrecer oportunidades de prácticas innovadoras que apoyen la preparación de sus estudiantes para su futura vida laboral. El continuo desarrollo tecnológico que las organizaciones viven, particularmente en las áreas de almacenamiento de información, recuperación y comunicación, altera la manera de enseñar y, por supuesto, de aprender.

La enseñanza dentro de las universidades debe permanecer en constante actualización. Las universidades deben ofrecer a sus estudiantes las herramientas para que ellos puedan explorar profundamente el medio laboral en el que se desarrollarán, pero desde su posición en el seno estudiantil. Con estas herramientas, ellos pueden diferenciar, analizar y crear su propio aprendizaje a través de una experiencia directa con el medio, aumentando su capacidad de respuesta y su habilidad para responder a las demandas tecnológicas del medio. Son varias las herramientas diseñadas con fines lúdicos o prácticos que se insertan en los cursos de IA en el mundo. Sin embargo, muy pocas son creadas específicamente para la enseñanza.

Con estos elementos se plantea la necesidad de contar con un sistema basado en el conocimiento, multiparadigma (que incluya la utilización de Prolog como paradigma declarativo) y que facilite la creación y utilización de bases de conocimiento para resolver problemas sobre un dominio especificado.

## Solución

Para dar solución a la problemática se diseña un Agente Lógico (Velázquez, Rosell, Coca, & Cruz, 2016) (Coca, Rosell, & Velazquez, 2017) capaz de resolver problemas basados en reglas o basados en casos, dependiendo de los elementos que el usuario le brinde (Figura 1). Para ello realiza parte de la inferencia siguiendo el paradigma de la programación lógica. A su vez permite que el usuario diseñe el problema que pretende resolver y que elija entre el módulo basado en casos y el módulo basado en reglas para llegar a una solución. El agente calcula el peso de cada uno de los rasgos en los casos almacenados, así como un grado de creencia sobre los hechos almacenados y los inferidos.

Para que el sistema cuente con inteligencia propia, se desarrolló siguiendo un modelo de agente inteligente para realizar el razonamiento. Este agente se encarga de todo el procesamiento lógico del sistema para llegar a una solución. También brinda algunas funcionalidades que ofrecen comodidades al usuario, por ejemplo, saber cuáles son las Bases de conocimiento (BC) más utilizadas, ir aprendiendo más sobre la BC mientras más se use, permitiéndole dar respuestas más certeras y rápidas. Además, el usuario debe indicar cuál será la variable dependiente en la BC, ya que en la mayoría de los problemas que se resuelven de este tipo hay una variable cuyo valor depende del resto.



**Figura 1.** Interfaz gráfica de la aplicación.

El módulo basado en casos muestra una explicación del razonamiento llevado a cabo para llegar a la solución, permitiéndole al usuario saber todo el análisis realizado por el agente para llegar a esa conclusión. Este módulo clasifica un nuevo caso, ayudándose para esto de los casos almacenados en la BC.

Las creencias del agente se definieron a partir de las percepciones del entorno y la BC interna. Se asumió como deseo del agente la solución de un nuevo caso y se definieron las intenciones a partir de cada una de las acciones que lleva a cabo, tanto de cara al usuario como internamente.

## Representación del conocimiento

Un agente lógico debe contar con una BC que represente sus creencias. Estas se pueden representar como un conjunto de expresiones arbitrarias, conocidas como cláusulas. Cada cláusula puede expresarse de la forma:

$$q: p_1 p_2 \dots p_n$$

Esta sentencia expresa que  $q$  será verdadera si cada una de los  $p_j$  ( $j=1, \dots, n$ ) son verdaderos.

Un ejemplo  $E$  se define como un par  $(X, f(X))$ , donde  $X$  es un vector de entrada, y  $f(X)$  es la salida de la función  $f$  aplicada al vector  $X$ . En problemas de clasificación, diagnóstico y muchos otros,  $X$  representa el conjunto de rasgos predictores o independientes (atributos en lo adelante) y  $f(X)$  el rasgo dependiente (objetivo en lo adelante). Para la construcción de árboles de decisión a partir de ejemplos (Mitchell, 1997) se calcula la ganancia de información de cada atributo de  $X$  hacia  $f(X)$ , pero no se asume una interdependencia entre ellos. Por otro lado, el modelo de Bayes Ingenuo (Russell & Norvig, 2004) asume que los atributos son condicionalmente inde-

pendientes entre sí, dado el objetivo. A partir de estos elementos en lo sucesivo se asume que el agente guarda ejemplos y resuelve problemas con atributos de  $X$  condicionalmente independientes entre sí, dada  $f(X)$ . Para simplificar la exposición, se asumen los atributos discretizados.

En la programación lógica es común representar el conocimiento en forma de reglas. Sin embargo, un ejemplo  $E$  puede representarse en forma de hecho como:

*ejemplo*(*id*,*rasgo*(*valor*<sub>0</sub>,*cree*<sub>0</sub> ),*rasgo*(*valor*<sub>1</sub>,*cree*<sub>1</sub> ),...,  
*rasgo*(*valor*<sub>*m*</sub>,*cree*<sub>*m*</sub>))

Donde el primer rasgo es el objetivo y los siguientes son los atributos.

Una representación usual para este tipo de sistema es a través de una matriz donde cada fila representa un ejemplo y cada columna un rasgo, se asume que la primera columna es el objetivo (Tabla 2).

**Tabla 2.** Representación de los casos

	$r_o$	...	$r_i$	...	$r_m$
$e_1$	$v(r_o e_1)$	...	$v(r_i e_1)$	...	$v(r_m e_1)$
⋮					
$e_j$	$v(r_o e_j)$	...	$v(r_i e_j)$	...	$v(r_m e_j)$
⋮					
$e_n$	$v(r_o e_n)$	...	$v(r_i e_n)$	...	$v(r_m e_n)$

Donde cada  $v(r_i e_j)$  representa un valor del conjunto de valores posibles del atributo  $r_i$  y cada  $v(r_o e_j)$  representa un valor del conjunto de valores posibles del objetivo  $r_o$ .

Algunos elementos necesarios para el trabajo posterior del agente será guardar la información necesaria acerca de cada uno de los

rasgos, tanto los atributos como el objetivo. La representación de cada rasgo debe respetar el orden en que aparecen en la estructura *ejemplo* creada, ya que la programación lógica realiza la inferencia siguiendo un orden de primero en profundidad. Los hechos tienen la forma siguiente:

$$\text{rasgo}(i, r_i, t, D, w_i).$$

Donde:

$i$ : Número consecutivo para identificar la posición de cada rasgo en el ejemplo.

$r_i$ : Nombre del rasgo.

$t$ : Tipo de dato, el cual puede ser nominal, real o entero.

$D$ : Dominio. Para el caso de los nominales tiene un conjunto de valores posibles, aquí se incluyen los dicotómicos. Los numéricos tienen el mínimo y el máximo valor.

$w_i$ : Peso del rasgo.

## Razonamiento

Luego de establecerse la notación de la información para la BC se debe tener en cuenta que, en el razonamiento basado en casos, los atributos no aportan de la misma manera al objetivo. Este razonamiento está estrechamente ligado al tratamiento de la incertidumbre. Aunque se especificará el tratamiento de la incertidumbre en un apartado para mostrar el tratamiento que se realiza de cara al usuario, a continuación se verá cómo todo el razonamiento va tratando la incertidumbre de los datos y del propio razonamiento.

La teoría de la información mide el contenido de información en bits. Un bit de información es suficiente para responder una pregunta de tipo sí/no sobre la que no se sabe nada. En general, si las

respuestas posibles  $v_i$  tienen probabilidades  $P(v_i)$ , el contenido de información  $I$  de la respuesta actual viene dado por:

$$I(P(v_1), \dots, P(v_k)) = \sum_{i=1}^k -P(v_i) \log_2 P(v_i)$$

Donde  $k$  es la cantidad de respuestas posibles. Como  $P(v_i)$  toma valores entre 0 y 1, el valor resultante de información está acotado entre 0 y  $\log_2 k$ .

En los sistemas basados en casos la tarea más común es saber cuál es el valor del objetivo en un nuevo problema  $nc$ . En general para calcular la información a partir del análisis del atributo  $r_i$  se tiene:

$$Resto(r_i) = \sum_{j=1}^p \frac{n(C_j)}{n(C)} I(P'(v_1), \dots, P'(v_k))$$

Donde  $n(C_j)$  es la cardinalidad del subconjunto  $C_j$ ,  $n(C)$  es la cardinalidad del conjunto de ejemplos  $C$  y  $P'(v_1), \dots, P'(v_k)$  es la probabilidad de cada respuesta del rasgo objetivo  $r_o$  dentro del subconjunto  $C_j$ .

La ganancia de información del atributo  $r_i$  es la diferencia entre la necesidad de información original y la nueva necesidad de información:

$$Ganancia(r_i) = I(P(v_1), \dots, P(v_k)) - Resto(r_i)$$

El inconveniente fundamental de este cálculo es cuando los valores posibles de un atributo son muchos, ejemplo los atributos identificadores. En esos casos el resultado asignaría el mayor peso a dicho atributo, cuando realmente no aporta al valor del objetivo. Para lograr un mayor aporte de los atributos con similar cantidad de valores posibles al objetivo, se divide el valor obtenido entre el módulo de la diferencia de los valores posibles del atributo ( $r_i$ ) y el rasgo objetivo ( $r_o$ ):

$$GananciaM(r_i) = \frac{Ganancia(r_i)}{1 + |p - k|}$$

Donde  $p$  es la cantidad de valores posibles para el atributo ( $r_i$ ) y  $k$  es la cantidad de valores posibles para el rasgo objetivo  $r_o$ . Luego para eliminar cualquier posibilidad de estar analizando un rasgo identificador, se adiciona una regla dura de manera que asigne:

$$Ganancia(r_i) = 0 \text{ Si } p = n(C)$$

Es decir que la Ganancia del atributo será 0 si la cantidad de valores posibles  $p$  es igual a la cantidad de ejemplos. Con este valor y para garantizar que  $\sum_{i=1}^m w_i = 1$ , se calcula la sumatoria de las ganancias de los atributos

$$G = \sum_{i=1}^m GananciaM(r_i)$$

y a partir de este valor se calcula el peso de cada atributo:

$$w_i = \frac{GananciaM(r_i)}{G}$$

Esta asignación de peso a los atributos es una muestra clara de razonamiento con tratamiento de la incertidumbre.

El módulo desarrollado se encarga de clasificar los nuevos casos teniendo en cuenta los ya almacenados. Para poder comparar el nuevo caso con los existentes se necesita recuperar todos los casos almacenados. Un algoritmo general para recuperar los casos semejantes utiliza una medida  $\delta_i$  (Coca, 2003). Esta brinda un valor de semejanza en el rango  $[0,1]$  entre los valores del atributo  $r_i$  del caso nuevo  $nc$  y el caso con que se compara en la BC  $e_j$ . Algunas de las medidas  $\delta$  más utilizadas son:

Para atributos con valores dicotómicos:

$$\delta_i = \begin{cases} 1 & \text{si } v(r_i e_j) = v(r_i nc) \\ 0 & \text{e.o.c} \end{cases}$$

Para atributos con valores numéricos:

$$\delta_i = 1 - \left| \frac{v(r_i e_j) - v(r_i nc)}{v_{\max}(r_i) - v_{\min}(r_i)} \right|$$

Además, se requiere de una función de semejanza  $f$  para combinar los valores de las semejanzas entre los atributos ponderados por el peso.

$$f\left(w_i, \delta_i\left(v(r_i e_j), v(r_i nc)\right)\right) = \frac{\sum_{i=1}^m w_i + \delta_i\left(v(r_i e_j), v(r_i nc)\right)}{\sum_{i=1}^m w_i}$$

Algunos autores incluyen en la función de semejanza entre casos el tratamiento de la incertidumbre para obtener un valor de semejanza  $\beta$  que incluye en su resultado dicho valor. Sin embargo, en la presente propuesta el tratamiento de la incertidumbre se hace aparte del valor de semejanza calculado. Se realiza toda la inferencia de forma paralela, esto permite brindar un valor de semejanza en base a las creencias del agente y un valor de creencia que tiene el propio agente sobre el resultado obtenido. De esta forma la salida del agente incluirá para cada caso semejante un par (Valor, Creencia). Quedando el algoritmo específico utilizado de la siguiente manera:

**Entrada:**  $nc, \varepsilon, C$  (nuevo problema, valor mínimo de semejanza para recuperar los ejemplos y conjunto de ejemplos de la base)

**Salida:**

$$L_s = \left\{ \left( e'_1, \text{Par}(\beta(nc, e'_1), \text{Cre}_e(\beta)) \right), \dots, \left( e'_1, \text{Par}(\beta(nc, e'_1), \text{Cre}_e(\beta)) \right) \right\}$$

(Lista formada por cada ejemplo semejante y el par formado por la

medida de semejanza entre  $nc$  y cada  $e'$ , y la creencia total del agente respecto al valor obtenido).

Se prosigue a calcular la creencia conjunta de los valores de los rasgos objetivos de  $nc$  y  $e_j$  mediante la regla del producto de la teoría de las probabilidades. Calcular la creencia del valor obtenido en  $\beta(nc, e_j)$  a través de la función  $\theta$ :

$$\theta(nc, e_j) = 1 - |\beta(nc, e_j) - \text{CreeC}(nc, e_j)|$$

Se compara el valor  $\varepsilon$  con  $\beta(nc, e_j)$  y si  $\beta(nc, e_j) > \varepsilon$  se adiciona a la lista el ejemplo, el valor de semejanza con el nuevo problema y la creencia del agente respecto a dicho valor:

$$L_s \leftarrow \{e_j, \text{Par}(\beta(nc, e_j), \theta(nc, e_j))\}$$

Luego de realizar todo el proceso de recuperación se prosigue con la adaptación, la cual consiste en encontrar una solución final al problema a partir de los resultados obtenidos en los casos recuperados. Se tiene en cuenta que la solución depende del dominio del objetivo  $r_o$ , que la lista de casos recuperados  $L_s$  debe ser considerada para la solución y que dicha solución no se realiza de la misma manera para valores nominales, numéricos enteros o numéricos reales. Además, en la presente propuesta se realiza el cálculo del valor final y la creencia que tiene el agente sobre dicho resultado de forma separada. Dicho lo anterior, se realiza el siguiente proceso de adaptación:

**Rasgo nominal:** Se utiliza la moda, es decir, el valor del objetivo  $r_o$  que más se repite dentro de la lista  $L_s$ . Para obtener el valor de creencia se combinan los valores de cada caso que tiene por valor objetivo la moda.

**Rasgo numérico real:** Se calcula el promedio de los valores de los objetivos  $r_o$  de toda la lista  $L_s$ . Como los valores de creencia están asociados a valores que serán promediados, el nuevo valor de creen-

cia se calculará de la misma manera, o sea, calculando el promedio de las creencias de cada valor.

**Rasgo numérico entero:** Se calcula el promedio truncado para el valor final, mientras el valor de creencia se calcula similar a los rasgos numéricos reales.

### ▮ Tratamiento de la incertidumbre

De cara al usuario el agente puede determinar un grado de creencia de cada valor de los atributos, a partir de los ejemplos guardados en su BC. A pesar de ser cuestionada, debido al uso simplificado de las hipótesis de independencia, la teoría de la probabilidad sigue siendo la más aceptada para el tratamiento de la incertidumbre (Russell & Norvig, 2004). Soporta este criterio el hecho de ser la menos dependiente del factor humano y brindar facilidades para el cálculo autónomo por el agente, lo cual la hace superior a otras teorías. Para la presente propuesta se sigue un análisis similar al modelo Bayes Ingenuo, para lo cual se calcula el nivel de creencia del agente sobre cada valor de los atributos, tomando en cuenta dos magnitudes:

- $Cree_p$  Creencia calculada a partir de la probabilidad a priori de la existencia de dicho valor para ese atributo en la BC.
- $Cree_c$  Creencia que toma en cuenta la relación de dicho valor para ese atributo con el valor del rasgo objetivo en ese mismo caso.

Para cada uno de los valores de los atributos en cada ejemplo de la BC se define formalmente cada una de las creencias como:

$$Cree_p(v(r_i)) = P(v(r_i))$$

$$Cree_c(v(r_i)) = P(v(r_i) | v(r_0))$$

Para calcular la creencia total de cada  $v_i$  se utiliza una función para combinarlas, se asume que cada una aporta positivamente a la creencia total, tanto una alta probabilidad de ocurrencia de un valor  $Cree_p$ , como una alta probabilidad de que ese valor aporte a una conclusión  $Cree_c$ , brinda elementos para creer que el valor tiene menos incertidumbre:

$$Sum(x, y) - Prod(x, y) = x + y - x * y$$

De esta forma  $Cree_t = Cree_p + Cree_c - Cree_p * Cree_c$

Luego se debe obtener un valor de creencia para el rasgo objetivo, dada la creencia de cada uno de los valores de los rasgos predictores:

$$Cree_t(v(r_0)|v(r_1), \dots, v(r_m)) = P(v(r_0)) \prod_i Cree_t(v(r_i))$$

La creencia del valor del atributo  $r_i$  ya incluye en su cálculo la probabilidad condicional  $P(v(r_i) | v(r_0))$ .

## Aprendizaje

El aprendizaje más común en este tipo de sistemas se logra mediante la incorporación de nuevos casos a la BC. Este proceso puede hacerse de forma automática, ya sea directamente adicionando cada nuevo caso resuelto o analizando si tiene elementos que aporten de forma significativa a nuevos posibles casos a resolver. Un agente debe ser capaz de decidir cuándo un caso aporta significativamente al sistema, o al menos tener elementos para tomar la decisión. Algunos de los más significativos pueden ser:

- **Tamaño  $n$  de la BC:** Mientras mayor sea, el sistema debe ser más restrictivo en la adición de nuevos casos.

- **Cantidad de casos semejantes recuperados (longitud de  $L_s$ ):** Si es baja, es muy probable que el nuevo caso tenga elementos de interés en la solución de futuros problemas.
- **Valor de semejanza entre el nuevo caso y el caso más semejante de la base ( $\beta(nc, e_j)^*$ ):** Si este valor es bajo se puede asumir que el nuevo caso es significativamente diferente a los guardados.
- **Valor de creencia obtenido del rasgo objetivo del nuevo caso ( $Cree(r_o nc)$ ):** Si se obtiene un valor bajo puede que la solución no sea tan buena, por tanto, su incorporación a la base pudiera afectar en alguna medida las soluciones posteriores.

Una de las ventajas de los sistemas basados en casos está en la posibilidad de tener un módulo de explicación, donde se ofrecen al usuario los elementos fundamentales que permitieron al sistema llegar a un resultado. En el sistema que se presenta se brindan al usuario los elementos expuestos con anterioridad, de esta forma puede determinar la incorporación o no del nuevo caso a la BC.

El aprendizaje del sistema no termina con la incorporación del caso, a partir de ahí se deben actualizar las creencias del agente. Además, con la solución de cada nuevo caso el sistema puede aprender de los valores que trae cada atributo. A partir de lo cual se asume el aprendizaje del agente en dos momentos:

1. **Al realizar una inferencia:** Se actualizan los valores de  $Cree_p$ . Se asume el valor de cada atributo del nuevo caso  $v(r_i nc)$  como información para la actualización de dicho valor de creencia a priori de cada ejemplo de la base.
2. **Al incorporar nuevo ejemplo a la base:** Se actualizan los valores de creencia  $Cree_c$  y  $Cree_p$ , para este último se utiliza el ya actualizado  $Cree_p$ .

---

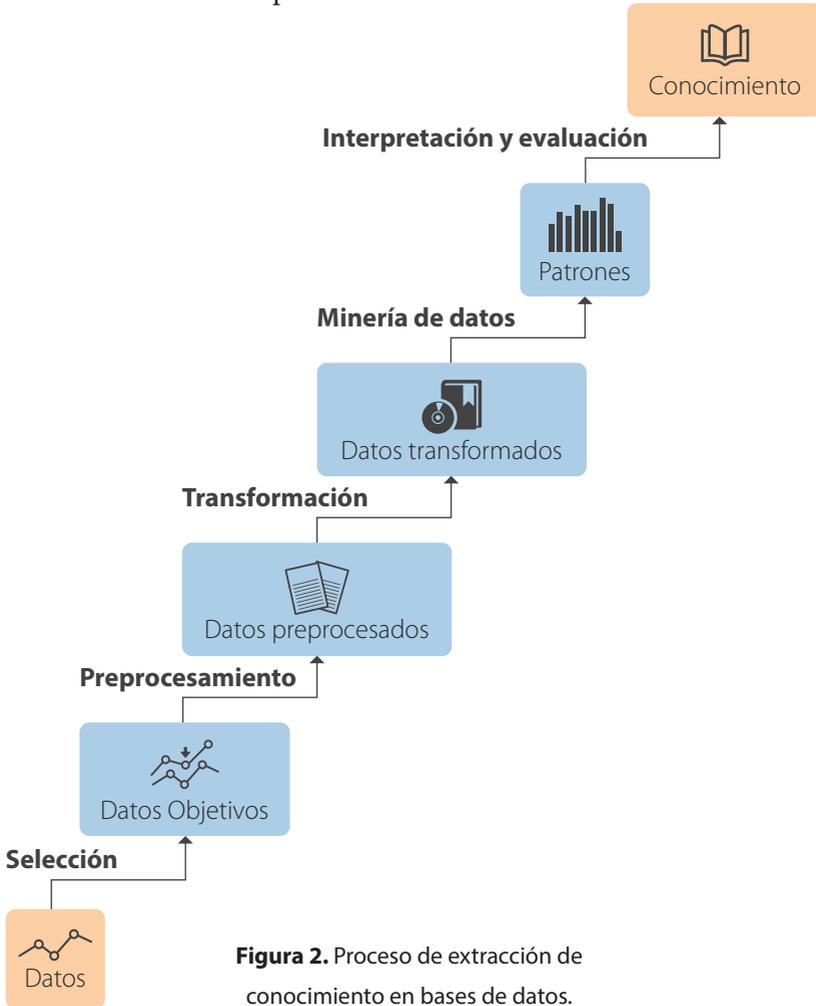
# MINERÍA DE DATOS

---

En la actualidad la cantidad de datos que ha sido almacenada en las bases de datos excede la habilidad humana para reducirlos y analizarlos sin el uso de técnicas de análisis automatizadas. Muchas bases de datos comerciales transaccionales y científicas crecen a una proporción desmedida. El Descubrimiento de Conocimiento en Bases de Datos (KDD, por sus siglas en inglés), atendiendo al criterio de algunos autores, es el proceso completo de extracción de información que va desde la identificación y preparación de los datos hasta la interpretación de los resultados obtenidos. Dentro de todo este proceso se encuentra la minería de datos, la cual utiliza métodos de aprendizaje automático y se dirige a tareas específicas como la inducción de reglas, la clasificación y el agrupamiento, el reconocimiento de patrones, el modelado predictivo, la detección de dependencias, entre otras.

El proceso de KDD (Figura 2) se inicia con la identificación y selección de los datos a utilizar. Para ello hay que determinar qué datos se necesitan, dónde se pueden encontrar y cómo conseguirlos. Una vez que se dispone de datos, se deben seleccionar aquellos que sean útiles para los objetivos propuestos. Luego se preparan los datos aplicando técnicas de preprocesamiento y transformación para

ponerlos en un formato adecuado. Ya obtenidos los datos se procede a la Minería de Datos, donde se aplican los métodos de aprendizaje automático de la IA para lograr los objetivos trazados. Tras este proceso llega el análisis de resultados donde se interpretan se evalúan y se toman decisiones a partir de ellos.



**Figura 2.** Proceso de extracción de conocimiento en bases de datos.

La minería de datos es una tarea de la IA que tradicionalmente se ha asociado directamente al aprendizaje automático. Pero, como se verá a continuación, cualquier aplicación que utilice la minería de datos deberá prestar atención a la representación del conocimiento, al tratamiento de la incertidumbre, al razonamiento y, por supuesto, al aprendizaje.

## Presentación del problema

El Entorno para el Análisis del Conocimiento de la Universidad de Waikato (WEKA) ha sido una de las herramientas más utilizadas en la minería de datos, sobre todo en el ámbito educativo y científico. Facilita la aplicación de técnicas de clasificación y agrupamiento a conjuntos de datos en un formato preestablecido. Sin embargo, seleccionar cuál algoritmo utilizar para resolver un problema concreto puede ser engorroso, pues hay que aplicarle varios para verificar cuál arroja las mejores soluciones.

El objetivo, entonces, es diseñar un sistema capaz de analizar bases de datos, comparando resultados al aplicar algoritmos ya implementados en la biblioteca Weka.

## Solución

Se desarrolló un sistema multiagente (SMA) que sugiere al usuario el mejor algoritmo para resolver un problema de clasificación o agrupamiento, mediante el cálculo de un conjunto de métricas (Gutiérrez, Cedeño, & Coca, 2017) (Figura 3). Se utilizó la metodología de desarrollo de sistemas multiagentes INGENIAS. Esta cuenta con cinco meta-modelos: Agentes, Interacciones, Organización, Entorno, Tareas y Objetivos. Para implementar la solución informática

se utiliza JADE, un conjunto de bibliotecas en lenguaje Java para la creación de aplicaciones y SMA bajo los estándares FIPA (Foundation for Intelligent Physical Agents).



Figura 3. Interfaz de comparación de métodos de clasificación del Sistema multiagente SMEKDB.

## Representación del conocimiento

La primera tarea a llevar a cabo consiste en transformar las bases de datos al tipo de archivo con que trabaja la biblioteca de algoritmos a utilizar. Weka solo trabaja con ficheros con la estructura especificada en el formato ARFF. Por tanto, es necesaria una transformación del conocimiento disponible en las bases de datos para representarlo en dicho formato.

Para su ejecución se analizaron los diferentes tipos de datos de las tablas en el gestor de bases de datos PostgreSQL, con sus equivalentes en Java y luego se asociaron con los aceptados en ARFF (Tabla 3).

**Tabla 3.** Relación de tipos de datos de PostgreSQL, Java y Weka.

TIPO DE POSTGRESQL	TIPO DE JAVA	TIPO DE ATRIBUTO WEKA
Char	String	nominal
char[]		
character varying		
character varying[]		
character[]		
Text		
Boolean	boolean	
Bit		
double precisión	double	numeric
Numeric		
Smallint	short	
Integer	Int	
Bigint	long	
Real	float	
Date	date	date
Time	time	
Text	text	string

## Razonamiento

El razonamiento general del sistema pasa por la interacción de los agentes especificados Smek, Connector, Arff, Classifier y Clusterer. Smek es el agente principal, encargado de la administración del SMA y de la interacción con el usuario. El agente Connector se responsabiliza de la primera etapa del proceso de transformación de bases de datos de PostgreSQL en archivos ARFF. El agente Arff inicia la segunda etapa de dicho proceso con la información suministrada por Connector, y también realiza tareas de edición sobre archivos ARFF, como ya se presentó en el apartado de representación del conocimiento. Por su parte los agentes Classifier y Clusterer ejecutan las operaciones de clasificación y agrupamiento respectivamente.

En el proceso de comparación de clasificadores intervienen los agentes Smek y Classifier. Smek se encarga de recoger las configuraciones de los algoritmos, los métodos de evaluación y la dirección del archivo con los datos. El agente Classifier con la información recibida de Smek construye, evalúa y calcula las métricas para cada algoritmo. Luego compara los valores alcanzados en cada métrica para determinar y proponer el mejor algoritmo de clasificación mostrando los resultados en pantalla. Classifier también permite al usuario visualizar la comparación de las métricas en gráficos para una mejor comprensión visual.

En el proceso de comparación de agrupadores intervienen los agentes Smek y Clusterer. Smek se encarga de recoger las configuraciones de los algoritmos, los métodos de evaluación y la dirección del archivo con los datos. El agente Clusterer con la información recibida de Smek construye, evalúa y calcula las métricas para cada algoritmo. Luego compara los valores alcanzados en cada métrica para determinar y proponer el mejor algoritmo de agrupamiento

mostrando los resultados en pantalla. Clusterer también permite al usuario visualizar la comparación de las métricas en gráficos para una mejor comprensión visual.

La aplicación de los algoritmos y la identificación del mejor para un problema específico tiene un componente alto de tratamiento de la incertidumbre. El propio hecho de determinar las métricas para evaluarlos es tratar la incertidumbre dada por el cálculo de cada método. Por esta razón este razonamiento específico se mostrará en el apartado de tratamiento de la incertidumbre.

## Tratamiento de incertidumbre

Para especificar mejor cada caso se va a dividir en dos apartados: uno para la clasificación y el otro para el agrupamiento. Para llevar a cabo la clasificación de nuevos casos es necesario utilizar algoritmos de aprendizaje supervisado. Mientras, para el agrupamiento se utilizan algoritmos de aprendizaje no supervisado.

### Clasificación

La evaluación de los clasificadores depende de la matriz de confusión. Esta permite visualizar, a través de una tabla de contingencia, la distribución de los errores cometidos por los algoritmos, donde los Verdaderos Positivos (VP) son instancias correctamente reconocidas por el sistema, los Falsos Negativos (FN) son instancias positivas y que el sistema no reconoce como tales, los Falsos Positivos (FP) son instancias negativas pero el sistema dice que no lo son y los Verdaderos Negativos (VN) son instancias negativas y correctamente reconocidas. La matriz de confusión posibilita el cálculo de métricas como:

- Cantidad de instancias clasificadas correcta e incorrectamente  

$$\text{Correctos} = \text{VP} + \text{FN} \qquad \text{Incorrectos} = \text{VN} + \text{FP}$$

- El porcentaje de exactitud: por ciento que representa la cantidad de instancias clasificadas correctamente con respecto al total de instancias (N).

$$\%Exactitud = \frac{Correctos \times 100}{N}$$

- Precisión: proporción de casos positivos predichos que son realmente positivos.

$$Precision = \frac{VP}{FP + VP}$$

- Recall: la razón de casos reales positivos predichos como positivos de manera correcta.

$$Recall = \frac{VP}{FN + VP}$$

- Medida-F: media equilibrada entre la precisión y el recall.

$$Medida\_F = \frac{2 \cdot Precision \cdot Recall}{Precision + Recall}$$

- Estadístico kappa de Cohen: compara una precisión observada con una precisión esperada, las cuales se ilustra más fácilmente mediante el uso de una matriz de confusión. En un ejemplo se tienen 30 casos totales (Tabla 4). De acuerdo con la primera columna 15 se etiquetaron como A, y de acuerdo con la segunda columna 15 se etiquetaron como B. También se aprecia que el modelo clasificó 17 casos como A y 13 casos como B.

**Tabla 4.** Matriz de confusión de un ejemplo con 30 casos.

	CLASE REAL	
CLASE PREDICHA	A	B
A	10	7
B	5	8

La precisión observada ( $P_o$ ) es el número de casos que se clasificaron correctamente y se calcula agregando el número de instancias en las que el clasificador estuvo de acuerdo con la etiqueta real, dividido por el número total de instancias ( $P_o = \frac{10+8}{30} = 0.6$ ). Para determinar la precisión esperada ( $P_e$ ), primeramente, se multiplica la frecuencia marginal de la clase real A y la de la clase predicha A, y se divide por el número total de instancias. La frecuencia marginal es la suma de todas las instancias para esa clase. En este caso, 15 instancias fueron etiquetados como A de acuerdo a la clase real y 17 de acuerdo a la clase predicha, obteniendo como resultado un valor de  $\frac{15 \times 17}{30} = 8.5$ . Luego, se aplica el mismo procedimiento para la clase B alcanzando como resultado un valor de  $\frac{15 \times 13}{30} = 6.5$ . El paso final es sumar todos los valores obtenidos, y dividirlos por el número total de instancias ( $P_e = \frac{8.5+6.5}{30} = 0.5$ ). Finalmente se calcula el valor del estadístico kappa.

$$kappa = \frac{P_o + P_e}{1 - P_e}$$

- Especificidad: mide la proporción de negativos que se identifican correctamente como tales.

$$Especificidad = \frac{VN}{VN + FP}$$

- Costo total y promedio: el costo total es la suma de los FN y los FP. El costo promedio equivale al costo total dividido entre el número total de instancias.
- Coeficiente de correlación de Matthews (MCC, por sus siglas en inglés): es un coeficiente de correlación entre las clasificaciones observadas y predichas que mide la calidad de las clasificaciones binarias. Devuelve un valor entre -1 y 1, donde -1 indica una mala clasificación 1 una clasificación perfecta.

$$MCC = \frac{VP \times VN - FP \times FN}{\sqrt{(VP + FP)(VP + FN)(VN + FP)(VN + FN)}}$$

- AUC: es el área bajo la curva ROC. Un gráfico de Características de Operación del Receptor (ROC, por sus siglas en inglés) es una técnica para visualizar, organizar y seleccionar clasificadores basados en su desempeño. Una curva ROC es una representación bidimensional del rendimiento del clasificador. Para comparar clasificadores, se recomienda reducir el rendimiento de ROC a un solo valor escalar que representa el rendimiento esperado.

El último proceso que se realiza durante la evaluación de la clasificación es tomar en cuenta el error absoluto medio (EAM), la raíz del error cuadrático medio (RMSE), el error absoluto relativo (EAR) y la raíz del error cuadrático relativo (RRSE), que son calculados durante la evaluación de los modelos por WEKA. Estas métricas calculadas servirán también para conocer el clasificador con mejores resultados, y por tanto el que debe seleccionarse.

### Agrupamiento

El agrupamiento puede considerarse un tipo especial de clasificación, aplicado cuando no se tienen clases que predecir. Para medir la calidad del agrupador se emplean las métricas externas y las internas. Las primeras son medidas supervisadas que utilizan criterios no inherentes al conjunto de datos, estas comparan los resultados del agrupador contra información suministrada a priori por el usuario utilizando ciertas medidas de calidad de agrupamiento. Las medidas internas determinan la bondad de un agrupador considerando cuán bien separa y compacta los grupos.

Para calcular todas las métricas externas e internas se debe determinar el número idóneo de grupos (k) que mejor representa la estructura natural de los datos. Utilizando el método empírico se obtiene la ecuación  $k \approx \sqrt{\frac{n}{2}}$ , donde n es la cantidad de datos. El usua-

rio puede escoger la cantidad de datos calculadas, un valor predeterminado o uno deseado por el usuario. Para la evaluación basada en clases se propone usar el número de clases como número de grupos. Si es escogida cualquier otra forma de evaluación solo se tendrán en cuenta la medida interna del coeficiente Silhouette y la cantidad de instancias no agrupadas.

La cohesión mide la similitud dentro de un grupo y la separación mide la distancia a la que se encuentran dichos grupos. El Coeficiente Silhouette trata de minimizar la cohesión y maximizar la separación. Su valor oscila entre -1 y 1, pero se busca un valor positivo pues significa que la cohesión es menor que la separación. Se tiene entonces para un elemento  $O$ , donde  $dist(O, O')$  es la distancia entre  $O$  y  $O'$  y  $|C_i|$  es la cantidad de elementos agrupados en el grupo  $C_i$ :

$$\alpha(O) = \frac{\sum_{O' \in C_i, O \neq O'} dist(O, O')}{|C_i| - 1}$$

El cálculo de las métricas externas solo es posible para la evaluación basada en clases, donde a partir de las mismas se asigna una a cada grupo y se forma una matriz de confusión en la que cada elemento representa la cantidad de instancias de una clase asignada a un grupo.

Las métricas externas se dividen en dos grupos, las orientadas a clasificación y las orientadas a similitud. En el primer grupo se encuentran:

- **Entropía:** grado en que cada grupo está conformado por elementos de una clase. Para cada grupo se determina la distribución de clases de los datos (usando la matriz de confusión). Luego la entropía de cada grupo  $i$  es computada y la entropía total del agrupador. En las ecuaciones  $p_{ij}$  representa la probabilidad de que un miembro del grupo  $i$  pertenezca a la clase  $j$  y

determina  $p_{ij}=m_{ij}/m_i$  donde  $m_{ij}$  es el número de objetos de clase  $j$  en el grupo  $i$  y  $m_i$  el número de objetos en el grupo  $i$ . El número de clases es representado por  $L$ ,  $k$  es el número de grupos y  $m$  el total de datos.

$$e_i = - \sum_{j=1}^L p_{ij} \log_2 p_{ij} \quad e = \sum_{i=1}^k \frac{m_i}{m} e_i$$

- **Pureza:** indica cómo un grupo está constituido por objetos de una clase.

$$p_i = \max_j p_{ij} \quad \text{pureza} = \sum_{i=1}^k \frac{m_i}{m} p_i$$

- **Precisión:** indica cómo un grupo está compuesto por objetos de una clase específica.

$$\text{precisión}(i,j)=p_{ij}$$

- **Recall:** indica cómo un grupo concentra a todos los objetos de una clase específica. El recall se obtiene mediante la fórmula  $\text{recall}(i,j)=m_{ij}/m_j$ .

- **Medida-F:** mide cómo un grupo contiene solo objetos de una clase y a todos sus elementos.

$$\text{medidaF}(i,j)=(2 \times \text{precisión}(i,j) \times \text{recall}(i,j)) / (\text{precisión}(i,j) + \text{recall}(i,j))$$

Las medidas externas orientadas a similitud se determinan con respecto a matrices de similitud y son el índice de Jaccard, el índice de Fowlkes-Mallows, el índice Rand y el índice gamma. En el caso de estadístico gamma es el equivalente al coeficiente de correlación entre la matriz de similitud de grupos y la matriz de similitud de clases. La matriz de similitud de grupos se realiza sobre la estructura de los grupos formados correspondiendo a una similitud, si dos instancias  $i$  y  $j$  están en el mismo grupo (se coloca el valor 1 en la celda  $ij$ ), hay una divergencia si dos instancias  $i$  y  $j$  no están en el mismo grupo (se coloca el valor 0 en la celda  $ij$ ). La matriz de similitud de clases

se realiza sobre la estructura de las clases estructurada de los datos y sigue la misma lógica que para la similitud de grupos.

$$r = \text{correlacion}(X, Y) = \frac{\sum(x - \bar{x})(y - \bar{y})}{\sqrt{\sum(x - \bar{x})^2 \sum(y - \bar{y})^2}}$$

Luego de formadas ambas matrices de similitud es posible extraer la información mediante una tabla donde se muestra el número de pares de elementos asignados para cada caso.

Se puede realizar entonces el cálculo de las métricas faltantes:

$$\text{Jaccard} = \frac{a}{c+b+a}$$

$$\text{Rand} = \frac{a+d}{a+b+c+d} \quad \text{FolkesMallows} = \sqrt{\frac{a}{a+c} \times \frac{a}{a+b}}$$

El sistema desarrollado construye los modelos, los evalúa y posteriormente conforma una tabla con las métricas presentadas. En dicha tabla se comprueba qué modelo obtiene mejores resultados en cada una de las métricas.

## Aprendizaje

El aprendizaje es la esencia de este sistema. Los algoritmos utilizados, disponibles en la biblioteca de algoritmos WEKA, son para la clasificación o el agrupamiento, tareas específicas del aprendizaje automático. Toda la representación, el razonamiento general y el tratamiento de la incertidumbre presentada hasta aquí, son para determinar el mejor algoritmo de aprendizaje para un problema especificado.

Para la clasificación se incluyen los métodos KNN, J48, Bayes ingenuo y SMO. Mientras para el agrupamiento se incluyen CO-BWEB, K-Medias y EM.

El método de los K-vecinos más cercanos es una técnica de clasificación basada en ejemplos. Se suele denominar método y no algoritmo porque es como un esqueleto que admite variaciones dando paso a algoritmos específicos, en este caso se pueden utilizar diversas funciones de proximidad. La función de proximidad puede decidir la clasificación de un nuevo ejemplo atendiendo a la clasificación de los k ejemplos más cercanos. Admite funciones de proximidad que consideren el peso o coste de los atributos que intervienen, permite que los atributos sean simbólicos, fechas o numéricos y puede funcionar bien aunque hayan atributos con valores perdidos.

El algoritmo J48 es utilizado en la construcción de árboles de decisión. Un árbol de decisión puede interpretarse como una serie de reglas obtenidas del análisis de un conjunto de ejemplos, estructurados como vectores de pares ordenados atributo-valor. Cada nodo está etiquetado con un par atributo-valor y las hojas con una clase, de forma que la trayectoria que determinan desde la raíz los pares de un ejemplo de entrenamiento alcanzan una hoja etiquetada con la clase del ejemplo.

La clasificación de un ejemplo nuevo del que se desconoce su clase se hace con la misma técnica, solamente que en ese caso al atributo clase, cuyo valor se desconoce, se le asigna de acuerdo con la etiqueta de la hoja a la que se accede. Este algoritmo es una variante de implementación realizada del C4.5 que a su vez se deriva del ID3.

El algoritmo J48, que implementa WEKA, admite atributos de tipo simbólicos, fechas y numéricos. Permite ejemplos con valores perdidos en los atributos, tanto en el momento de entrenamiento

como en la predicción de un caso, pero la clase siempre debe ser nominal.

El clasificador Bayes Ingenuo se utiliza cuando se quiere clasificar un caso de acuerdo con el valor más probable dados los valores de sus atributos. Estos clasificadores asumen que el efecto de un valor del atributo en una clase dada, es independiente de los valores de los otros atributos. Esta suposición se llama “independencia condicional de clase”. Esta suposición simplifica los cálculos involucrados dado que es una simplificación de la realidad. A pesar del nombre del clasificador y de la simplificación realizada, este método funciona muy bien en muchos casos. Admite atributos de los ejemplos de tipo simbólicos y numéricos.

El método de Optimización Mínima Secuencial (SMO, por sus siglas en inglés) es utilizado para entrenar una Máquina de Soporte Vectorial (SVM por sus siglas en inglés). Este modelo de clasificación funciona relacionando las distintas características de los elementos que se desean clasificar a un espacio vectorial, en el cual trazan un hiperplano que separa los vectores de una clase de los del resto. Las SVM son bastante utilizadas porque tienen un funcionamiento relativamente sencillo, asociado a una serie de propiedades teóricas y prácticas bastante atractivas, además de que pueden ser modificadas con facilidad para abordar problemas con datos que no son separables linealmente o que contienen mucho ruido.

El método SMO reemplaza todos los valores perdidos (o vacíos) de los atributos, transforma los nominales en binarios y los normaliza. Permite atributos simbólicos y numéricos.

Dentro de los métodos de agrupamiento más utilizados esta el K-Media. Para su utilización es imprescindible definir por adelantado el número de grupos que se desea. 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 se defina. Para cada uno de los grupos se calcula el centroide de todas sus instancias, los cuales son tomados como los nuevos centros de sus respectivos grupos. 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-Media se encuentra con un problema cuando los atributos no son numéricos. En respuesta surge la noción de agrupamiento conceptual, basada en un agrupamiento cualitativo frente al agrupamiento cuantitativo. Para ello se toma en cuenta la vecindad entre los elementos de la población.

El método COBWEB forma los conceptos por agrupación de ejemplos con atributos similares. Representa los grupos como una distribución de probabilidad sobre el espacio de los valores de los atributos, generando un árbol de clasificación jerárquica en el que los nodos intermedios definen subconceptos. El objetivo de COBWEB es hallar un conjunto de clases o grupos (subconjuntos de ejemplos) que maximice la utilidad de la categoría (partición del conjunto de ejemplos cuyos miembros son clases).

El algoritmo de Máxima Expectativa (EM por sus siglas en inglés) empieza buscando 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.

Todos los métodos incluidos se basan en aprender, a partir de ejemplos almacenados, para dar solución a un nuevo problema.



---

# PROCESAMIENTO DE LENGUAJE NATURAL

---

El Procesamiento de Lenguaje Natural (PLN) es un enfoque computarizado para analizar texto o voz utilizando un conjunto de teorías y tecnologías. Tiene dos subáreas fundamentales: la comprensión del lenguaje y la generación del lenguaje. La primera división se refiere al análisis del lenguaje con el propósito de producir una representación significativa del mismo, mientras que la segunda se refiere a la producción del lenguaje a partir de una representación especificada.

Esta es un área en pleno desarrollo, soportado en gran medida por la Inteligencia Artificial. Algunos avances recientes como el big-data, las tecnologías móviles, las redes digitales y los Cursos Abiertos Masivos en Línea (MOOCs por sus siglas en inglés) han resultado en la creación de nuevas oportunidades y desafíos de investigación. Algunas aplicaciones comerciales incluyen asistentes de escritura y ambientes de instrucción en línea. Los avances en el PLN y las tecnologías educacionales, así como la disponibilidad de grandes cantidades de texto relevante en el campo educativo y también de datos del habla, han llevado a un interés creciente en usar el PLN para dirigir las necesidades de profesores y estudiantes.

Aunque la tarea sea al final la misma, es muy diferente el tratamiento computacional para el PLN en texto y el PLN en sonido.

El primero ha avanzado más, con el desarrollo de asistentes en dispositivos móviles y sistemas operativos o los llamados chatbot que permiten una interacción automática con el usuario sobre temas específicos.

Desde los inicios de la Inteligencia Artificial, hasta nuestros días con el avance de la robótica, una tarea importante de la IA ha sido el procesamiento de lenguaje natural. Tanto es así, que varios autores lo consideran un núcleo básico de esta. Sin embargo, la realidad es que para lograr el procesamiento de lenguaje natural es necesario representar el conocimiento, tanto de elementos gramaticales del idioma, como del contenido específico que se analice. Es necesario llevar a cabo un razonamiento que incluya la percepción del interlocutor y tomar la decisión de la respuesta a brindar, tomando en cuenta un fuerte tratamiento de la incertidumbre. En cuanto al aprendizaje, hay dos aristas fundamentales, una a largo plazo y otra a corto plazo. A largo plazo es necesario que el sistema adquiera conocimiento sobre algún dominio específico, pero no debe limitarse a un conocimiento estático, sino que su propio desempeño en el entorno, debe ir enriqueciendo su conocimiento sobre dicho dominio. Por otro lado, más a corto plazo, es necesario llevar el control de una conversación específica, el tema que se trata, qué preguntas se han hecho, qué respuestas se han dado, características e intereses del interlocutor, entre muchos otros aspectos que debe ir adquiriendo el sistema en la medida que interactúa con una persona real.

## Presentación del problema

Las plataformas educativas agrupan diversos cursos en línea sobre diferentes asignaturas. Generalmente, se ofrece interactividad entre los estudiantes y los contenidos educativos mediante activida-

des como tareas, ejercicios, cuestionarios y exámenes, favoreciendo el aprendizaje en línea. A través de los foros de discusión existe una comunicación multidireccional de tipo asincrónica y con retroalimentación diferida entre los docentes y los estudiantes. Los foros permiten a los estudiantes intercambiar experiencias e ideas, formular y responder preguntas, exponer situaciones, sintetizar pensamientos, reflexionar y cuestionar, todo ello con la intervención del docente que promueve, apoya y retroalimenta los diálogos.

Si un estudiante desea plantear preguntas y dudas en los foros de discusión, debe esperar por la respuesta del profesor, lo cual no siempre sucede en un lapso de tiempo corto, debido a que el profesor no está conectado a la plataforma en todo momento, lo que provoca que no se favorezca la atención diferenciada en tiempo real. Pocas plataformas, todavía hoy, cuentan con mecanismos que permitan soportar el proceso de tutoría autónoma y brindar retroalimentación en tiempo real a las necesidades cognitivas de los estudiantes utilizando lenguaje natural.

A partir de estas limitantes se propuso diseñar un sistema capaz de interactuar de forma autónoma en forma de texto escrito, para asistir en la enseñanza de temáticas en un dominio especificado.

## Solución

El sistema se desarrolla como un Sistema Tutorial Inteligente (STI) capaz de adaptarse a diferentes dominios de conocimiento (Rojas, Coca, Núñez, & Batista, 2018). El STI permite que el profesor a cargo de un curso dentro de la plataforma educativa pueda incluir una red semántica y asociarla a dicho curso. La red semántica importada representa el conocimiento que domina el STI sobre el contenido del curso. En la propia definición de la red semántica es-

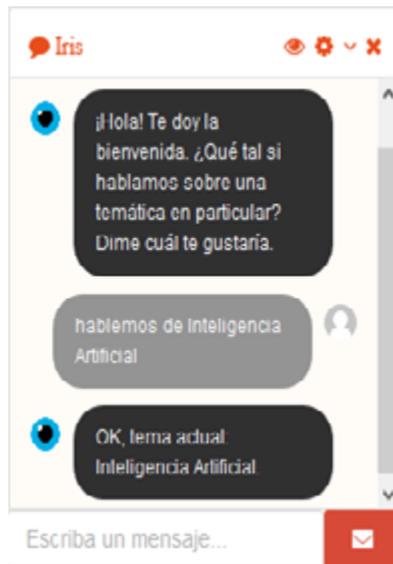
tán especificados los nodos más importantes que el tutor inteligente chequea para establecer el perfil cognitivo de cada estudiante.

El perfil cognitivo es la forma en que el STI establece el modelo del estudiante, permitiéndole tener creencias en cuanto a cuáles son los contenidos que domina y en qué medida lo hace. Esta medida está dada por una distancia para cada nodo importante de la red semántica definida. El estudiante, por su parte, accede a la interfaz de comunicación con el STI, que tiene el funcionamiento y la estética de un chat, mediante el cual intercambia información y conocimientos a través de diálogos en lenguaje natural (Figura 4).

Algunas de las funcionalidades principales del STI de cara al estudiante, son las siguientes:

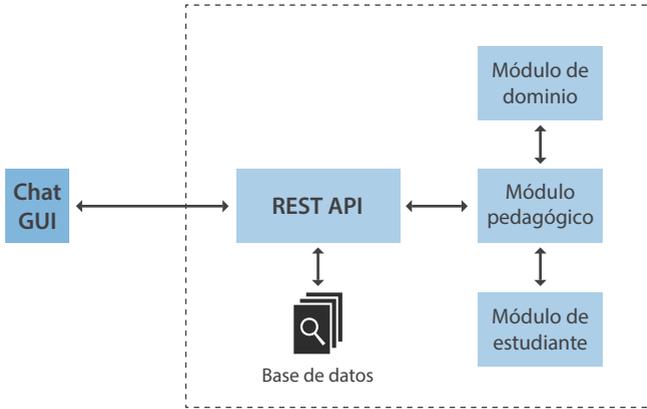
- Saludar, dar la bienvenida y proponer un tema de conversación al iniciar la primera conversación con el estudiante.
- Despedirse y cerrar la conversación al despedirse el estudiante.
- Responder preguntas relacionadas con su identidad.
- Ofrecer la definición de un concepto asociado al tema de conversación actual en caso de que el estudiante lo solicite.
- Recomendar fuentes bibliográficas asociadas a un concepto o contenido cuando el estudiante lo pida.
- Permitir al estudiante proponer un tema de conversación, cambiar el tema actual o realizar preguntas; así como permitirle preguntar por los posibles temas de conversación disponibles y proponer que se cambie el tema de conversación.
- Realizar preguntas de control con el objetivo de actualizar el perfil cognitivo del estudiante.
- Indicar cuándo una respuesta es correcta o incorrecta y brindarle al estudiante la posibilidad de expresar que no conoce la respuesta para una pregunta dada.

- Permitir al estudiante evadir una pregunta y posteriormente insistir en que sea respondida.



**Figura 4.** Entorno chat del sistema tutorial inteligente.

Una arquitectura típica de un STI consiste en una base de conocimiento, el modelo del estudiante, el módulo pedagógico y la interfaz gráfica de usuario. La base de conocimiento, también llamada “modelo de dominio”, es el conocimiento relacionado con el tema que se le enseña al estudiante. El sistema desarrollado respeta la arquitectura típica y crea una interfaz para lograr su utilización en diversas arquitecturas posibles de plataformas educativas (Figura 5).



**Figura 5.** Arquitectura general del sistema tutorial inteligente.

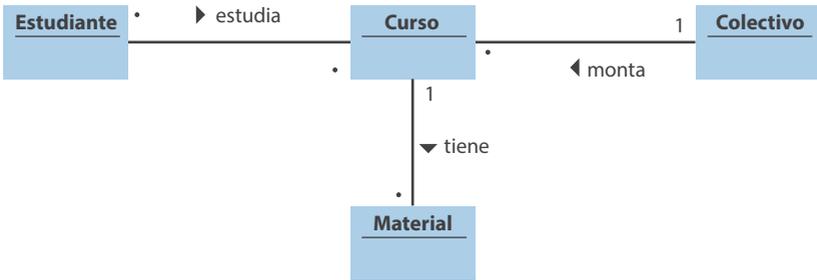
El módulo pedagógico se diseña como un agente BDI con “creencias”, “deseos” e “intenciones”. Este agente pedagógico está conformado de varias piezas internas que le permiten llevar a cabo sus funciones, entre ellas se encuentra un compilador (e intérprete) que permite analizar, compilar y ejecutar la notación intermedia que se genera para cada frase de entrada del usuario.

El módulo de dominio maneja la base de conocimiento en forma de redes semánticas. Todas las acciones que se efectúan sobre la base de conocimiento serán desencadenadas por las decisiones tomadas en el agente pedagógico.

El módulo de estudiante maneja el perfil cognitivo de cada estudiante. Todas las acciones efectuadas sobre este módulo serán también desencadenadas por las decisiones tomadas en el agente pedagógico. Los perfiles cognitivos se modificarán con cada interacción de los estudiantes con el sistema.

## Representación del conocimiento

Este tipo de sistemas debe lidiar con mucho conocimiento. Lo primero es el relacionado con todo el funcionamiento del sistema. Para ello se define un modelo de dominio que representa el conocimiento general (Figura 6).



**Figura 6.** Modelo de dominio del sistema.

El estudiante es la persona que se matricula a los cursos de la plataforma educativa, estudiando sus materiales y que consulta al STI para aclarar dudas y consolidar sus conocimientos. Los cursos son montados en la plataforma educativa y acceden a ellos los estudiantes. El colectivo es un equipo multidisciplinario de profesores e ingenieros del conocimiento que montan el curso y entrenan al STI para que este, a su vez, pueda realizar sus funciones. En tanto un material educativo montado en un curso, se refiere a documentos, libros, figuras y otros recursos como videos y audio.

La otra parte importante de conocimiento a representar es la relativa al contenido específico sobre una temática. En este caso se emplean las redes semánticas. Cada red semántica está definida en un archivo cuyo contenido es un conjunto de datos con información válida, correctamente estructurada y respetando la sintaxis de un

lenguaje de intercambio de datos, en este caso JSON. El formato que se define a continuación consta de un conjunto de reglas que permiten definir cada nodo de la red a incluir, así como sus relaciones con otros nodos, relaciones especiales y otros aspectos:

- **Regla 1.** El archivo contendrá un único objeto de JSON, que representa la red semántica que se define y deberá contener como mínimo un nodo.
- **Regla 2.** Los nodos se definen en el atributo “nodos” dentro del objeto único de red semántica. Este atributo consistirá en un arreglo no vacío de objetos donde cada uno definirá un nodo.
- **Regla 3.** Cada nodo tendrá como atributos obligatorios el nombre, el cual debe ser único, y el tipo para definir el tipo del nodo (concepto o instancia).
- **Regla 4.** Las relaciones entre los nodos se definirán en el atributo opcional “relaciones”. Este atributo consistirá en un arreglo de objetos, en los que cada uno definirá una relación con otro nodo. Tradicionalmente, cada relación se define como una tripla de la forma “nodo origen – relación – nodo destino”, en este formato se declara el nodo origen como poseedor de las relaciones, por lo cual solo se define la relación del mismo con otros nodos. Cada relación puede tener cualquier nombre, exceptuando algunos nombres reservados por el formato, de acuerdo al tipo de nodo.
- **Regla 5.** Cada relación se define como un objeto con un único atributo, que será el nombre de la relación y cuyo valor será un arreglo de los nombres a los cuales apunta dicha relación.
- **Regla 6.** Las relaciones reservadas por el formato y que forman parte de la lógica del tutor inteligente, están asociadas a cada tipo de nodo.
  - **Super:** relación opcional reservada exclusivamente para los nodos de tipo “concepto”. Consiste en un arreglo con los

nombres de los nodos padres del nodo actual, de los cuales hereda todas sus propiedades, permitiendo de esta forma la herencia múltiple.

- **Instancia de:** relación obligatoria reservada exclusivamente para los nodos de tipo “instancia”. Define cuál es el concepto del cual instancia el nodo actual. Un nodo de tipo instancia debe instanciar solo un nodo de tipo concepto y heredará todas las propiedades del nodo que instancia.
- **Regla 7.** Una propiedad se define como una relación entre el “nodo origen”, que es el nodo actual que posee la propiedad, el “nombre” de la propiedad y el “valor”, que representa el nodo destino y servirá para almacenar el valor que toma la propiedad actual (“nodo origen – nombre – valor”).
- **Regla 8.** Las propiedades de un nodo se definen en el atributo “propiedades”, al mismo nivel que los demás atributos. El valor de este atributo será un arreglo donde cada elemento es un objeto propiedad. Cada objeto propiedad poseerá un único atributo, que será el nombre de la misma y cuyo valor podrá ser “atómico”, “múltiple” y “sin valor”:
- **Regla 9.** Un nodo puede o no poseer propiedades asociadas.
- **Regla 10.** Si una propiedad está definida dentro de un nodo de tipo “instancia”, esta solo se aplica a dicho nodo. Si una propiedad está definida dentro de un nodo de tipo “concepto”, esta se aplica tanto al mismo como a todas sus instancias mediante un mecanismo de inferencia automática.
- **Regla 11.** La propiedad “bibliografía” está reservada por el formato, es usada para definir el conjunto de fuentes bibliográficas asociadas a un nodo, definida mediante un arreglo o valor atómico.

Para cada nodo se define un atributo que estará en verdadero si es un nodo esencial, es decir si es un nodo a incluir en el perfil cognitivo del estudiante. Cada uno de estos nodos define un objetivo a cumplir por el estudiante en su interacción con el sistema.

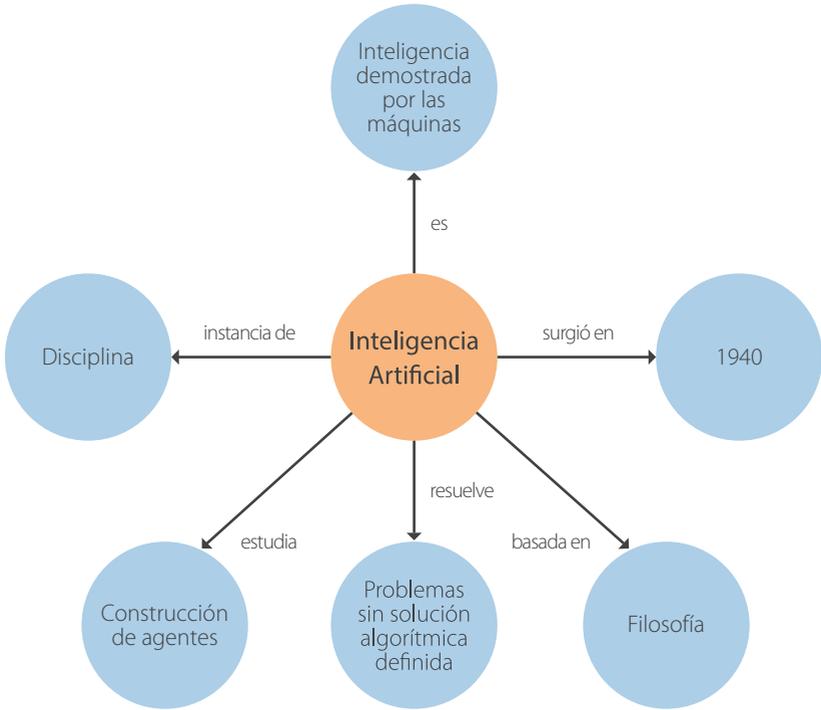
Un ejemplo sencillo de cómo codificar una red semántica se muestra en la siguiente posible caracterización del concepto “Inteligencia Artificial”. En la figura 7 se muestra el código fuente del archivo JSON que define la red mediante el formato definido. La figura 8 muestra la representación gráfica del nodo principal y los nodos generados automáticamente al ser procesado el archivo de red semántica.

```

{
  "nombre": "Inteligencia Artificial",
  "tipo": "concepto",
  "det": "la",
  "super": [
    "disciplina"
  ],
  "relaciones": [
    {
      "es la": ["inteligencia demostrada por las máquinas"]
    },
    {
      "surgió en el año": ["1940"]
    },
    {
      "soluciona": ["problemas sin solución algorítmica definida"]
    },
    {
      "estudia la": ["construcción de agentes"]
    },
    {
      "está basada en": [
        "Filosofía"
      ]
    }
  ],
  "bibliografía": [
    "página web 'http://es.wikipedia.org/Inteligencia Artificial/'",
    "libro 'IA. Un enfoque moderno.' de Russel & Norvig"
  ]
}

```

**Figura 7.** Ejemplo de código de la red semántica



**Figura 8.** Ejemplo de representación de la red semántica de la Figura 7.

## Razonamiento

Para garantizar la generación de respuestas a cada frase de entrada del estudiante, así como la generación y deducción del perfil cognitivo de cada uno durante el proceso de comunicación humano-STI, se diseñó un algoritmo general.

1. El usuario introduce una frase en lenguaje natural a través de la interfaz de usuario definida (chat) la cual se dirige hacia la interfaz del STI.

2. La interfaz redirecciona la frase del usuario hacia el módulo pedagógico, el cual analiza y genera una respuesta, siguiendo los pasos siguientes:
  - a. Se establece un contexto de ejecución que contiene información útil, como usuario, tema de conversación, listado de preguntas pendientes e intenciones.
  - b. Se procesa la frase haciendo uso de la biblioteca de Python spaCy: La frase del usuario pasa a través de diferentes procesos, ejecutando operaciones de PLN como tokenization<sup>1</sup>, el tagging<sup>2</sup>, parsing<sup>3</sup>, el reconocimiento de entidades nombradas, entre otras. Al concluir retorna un objeto que contiene la sentencia original y algunas informaciones de interés.
  - c. Se clasifica la frase de acuerdo a un conjunto de patrones sintácticos (Pregunta, respuesta, bienvenida, etc) y cuando coincide con alguno es traducida a una notación intermedia o comando, que expresa de forma unitaria la o las tareas a realizar, ejemplo /define(Inteligencia Artificial). De esta forma el agente transforma la frase original a una notación que pueda interpretar y ejecutar.
  - d. El compilador analiza léxica y sintácticamente el comando generado, posteriormente el intérprete lo ejecuta según el contexto de ejecución generado en el paso a. Para ello utiliza todo el conocimiento representado tanto del sistema general (modelo del estudiante, curso, etc) como del conocimiento especificado en la red semántica sobre la asignatura específica.

---

<sup>1</sup> Proceso mediante el cual se divide la frase en elementos atómicos o pedazos llamados *tokens*.

<sup>2</sup> Asignación de etiquetas lingüísticas a cada token.

<sup>3</sup> Análisis sintáctico y construcción del árbol de dependencias sintácticas.

- e. Cuando un comando es ejecutado, retorna una respuesta que es generada dinámicamente usando los pedazos de respuestas generadas en los pasos anteriores y un conjunto almacenado de posibles respuestas por tipo.
3. Se devuelve la respuesta final generada.

## Tratamiento de la incertidumbre

En este tipo de sistemas es sumamente importante el tratamiento de la incertidumbre. El primer punto a analizar es cómo se maneja por el sistema el desconocimiento sobre una frase introducida. Para este caso, cuando la frase pasa por el análisis y no logra identificar un patrón lógico definido, se utiliza un patrón de fallo. Cuando se genera la respuesta se utiliza aleatoriamente una respuesta aleatoria de las predefinidas para fallo, por ejemplo: “¿Podrías explicarte mejor?”.

El otro punto de interés para el tratamiento de la incertidumbre es determinar el nivel de cumplimiento de los objetivos definidos para el estudiante. En la red semántica se especifican los nodos básicos que debe dominar cada estudiante. Este proceso lo hace el profesor al introducir la red semántica en el sistema. De esta forma se guarda en el perfil cognitivo del estudiante, para cada nodo de la red a vencer, un valor que se irá modificando en la medida que el estudiante vaya venciendo satisfactoriamente las preguntas asociadas a los nodos relacionados directamente a él.

La medida de distancia está en el rango de 0 a 1, donde 0 representa la total ignorancia acerca de un nodo y 1 representa el total conocimiento, esta distancia es calculada usando la siguiente fórmula:

$$d = \frac{1}{x}$$

En esta expresión  $d$  representa la distancia y  $x$  la cantidad de relaciones pendientes a chequear para un nodo específico, dada por las intenciones del agente pedagógico. Cuando la última intención para un nodo determinado es llevada a cabo y evaluada, la distancia para ese nodo toma valor 1, lo cual representa que el nodo ha sido comprendido completamente de acuerdo al conocimiento almacenado en el sistema. Finalmente, la intención es eliminada de la lista de intenciones pendientes.

## | Aprendizaje

De forma general el aprendizaje en este tipo de sistemas puede asumirse desde diversos puntos de vista. El primer punto es la adquisición de conocimiento sobre un dominio específico, el cual ya se mostró desde la incorporación por un profesor de la red semántica. Toda la información definida en cada archivo de red semántica incorporada al sistema es transformada y posteriormente almacenada en la base de datos relacional. Esto mejora el tiempo de acceso y permite al agente pedagógico moverse sobre la red semántica con mayor facilidad para realizar inferencias y definiciones de conceptos. También, se almacenan las respuestas estáticas que sirven de base en la construcción de respuestas dinámicas y presenta una clase que asocia a cada usuario una respuesta esperada, lo cual se usa para predecir las frases del usuario.

El tratamiento de la incertidumbre dando un seguimiento al perfil cognitivo del estudiante es la principal vía de aprendizaje, ya que va guardando la información de cada estudiante para darle seguimiento y verificar en cada momento si ya venció los objetivos definidos.

Otro elemento de interés relacionado con el aprendizaje en el sistema, es que debe saber responder las preguntas formuladas, procesando la frase del usuario y dando seguimiento a la conversación. Por otro lado, el STI toma en cuenta las preguntas no respondidas por el estudiante para reiterarlas en algún momento de la conversación. Es por eso que almacena un listado para recordar las preguntas que ya el estudiante ha respondido y otro para recordar las preguntas pendientes que se han realizado al usuario y no ha respondido.



---

# VIDEOJUEGOS

---

Los juegos fueron creados desde sus inicios con el propósito de brindar relajación y entretenimiento. Los videojuegos son juegos electrónicos, presentan el mismo fin que los tradicionales, pero, el jugador interactúa con un dispositivo electrónico. Con el paso del tiempo y la creación de determinados videojuegos, así como el mal uso de los mismos, su perspectiva ha tomado un curso equivocado. Suelen verse como un elemento dañino y obsesivo, que afecta la psicología y las emociones. Estos criterios, aunque tienen sus fundamentos, olvidan la gran cantidad de juegos enfocados a la instrucción e introducción de valores positivos en los jugadores.

En la búsqueda de mayor calidad, realismo, libertad e interactividad con el jugador, los videojuegos se han aliado con la IA. Con ella se intenta dar una idea de inteligencia del juego, más que resolver problemas complejos. Se intenta dotar a los personajes de capacidades propias de la inteligencia humana como la percepción, el razonamiento y la toma de decisiones.

Un punto de interés adicional de los videojuegos es que son un área de ejemplo y prueba para otra importante área de aplicación de la IA: La robótica. Un personaje en un videojuego tiene un comportamiento similar a un robot en la vida real. Los criterios teóricos e

incluso prácticos que permiten su comportamiento son similares. La diferencia marcada está en los sistemas de percepción del entorno que tiene cada uno y en las acciones concretas que llevarán a cabo. Pero, luego de obtener la información del entorno, sería similar la representación del conocimiento, el razonamiento, el tratamiento de la incertidumbre y el aprendizaje; los elementos que verdaderamente se refieren a la IA.

## Presentación del problema

A pesar de los avances alcanzados en este campo, no siempre los videojuegos realizados evidencian el uso de la IA. El uso consciente de la IA podría aportar una mayor calidad a los productos desarrollados. Muchas veces se emplean técnicas de IA de forma inconsciente, pues los programadores las utilizan como parte de la implementación del videojuego. Esto se debe, además, al poco conocimiento teórico que existe acerca de la IA, sus técnicas y cómo se aplican en el desarrollo de videojuegos.

Por otra parte, los videojuegos son considerados como la aplicación informática ideal para utilizarse como medio en asignaturas de Inteligencia Artificial. En ellos los estudiantes pueden analizar algunas técnicas, modificarlas o incorporarles parte del código fuente, implementando los algoritmos vistos en clases.

Estos elementos propiciaron la proyección de un videojuego donde se visualice la utilización de los cuatro núcleos de conocimiento de la IA, de manera que sirva de ejemplo, tanto a desarrolladores, como a estudiantes de carreras informáticas.

## Solución

Uno de los juegos clásicos utilizados en libros para la enseñanza de la Inteligencia Artificial es “El mundo de Wumpus”. Este juego se presenta como una buena opción para lograr una herramienta interesante y útil para desarrolladores (Cuza, 2019). Una implementación básica de este juego, de manera que cumpla con las especificaciones de un software libre, permite el estudio y la modificación de su código. Esta característica le permite, además, ser utilizada para la enseñanza en las clases de IA.

Para el desarrollo de la propuesta de solución se emplearon como herramientas y tecnologías el software de diseño 2D Illustrator, el software de diseño 3D Maya y la herramienta Make Human. Se seleccionó el motor de videojuego Unity 3D, cuya función principal es la integración de gráficos 3D y 2D, sonidos, animación, física, scripting e inteligencia. Permite programar en lenguaje C# o Java Scripts y se emplea como entorno de desarrollo integrado (IDE) el MonoDevelop.

El mundo de Wumpus consiste en un tablero formado por cuadrículas, las cuales podrán contener al jugador, el Wumpus, el tesoro, precipicios, brisa, hedor y la entrada y salida. El objetivo principal del agente inteligente es obtener el oro y salir vivo. El diseño del tablero puede realizarse con las dimensiones y la colocación de los elementos en las cuadrículas que se desee, excepto el hedor que debe ubicarse en las casillas adyacentes al Wumpus y la brisa en las adyacentes a los precipicios. El agente podrá moverse hacia arriba, abajo o hacia los lados y también podrá disparar. Muere al caer en un precipicio o si cae en manos del Wumpus. El agente debe memorizar cada información que perciba del entorno, lo cual forma parte de su conocimiento, esto será vital para llegar a lograr su objetivo.

En el videojuego el jugador tiene como objetivos matar al Wumpus y encontrar el tesoro. Para lograr el primer objetivo debe encontrar una poción ubicada en algún lugar del mapa y lanzársela al Wumpus. Existen tres posibles terrenos que se seleccionan de forma aleatoria cada vez que se comience a jugar. El juego cuenta con un Modo Agente y la opción de jugar (Figura 9).

- Modo agente: Juega un personaje predefinido que actúa de forma autónoma mostrando en pantalla el razonamiento lógico llevado a cabo en cada momento. Busca todo el tiempo el tesoro y la poción para luego matar al Wumpus. En la pantalla se muestra el botón de opciones, la poción para el Wumpus y un mapa con la ubicación del jugador. El agente termina si se logran los objetivos definidos o fracasa al caer en un hoyo o ser devorado por el Wumpus.



**Figura 9.** Entorno del videojuego El mundo de Wumpus.

- Jugar: El jugador avanzará por el camino y se guiará por el ambiente para saber por dónde ir. En la pantalla se verá el nivel de vida, el botón de opciones y la poción para el Wumpus. Para

proteger el mundo de Wumpus existe un guardián que será un agente inteligente con la misión de matar a todo aquel que entre en este mundo, por lo que si ve al jugador lo perseguirá y atacará. Tanto el guardián (controlado por el sistema) como el jugador, cuentan con un arma con la que se podrán disparar, esto provoca que el nivel de vida baje. Si el nivel de vida del guardián se agota, muere; pero si la vida que se agota es la del jugador, el juego concluye. El jugador ganará si logra los objetivos y pierde si muere porque se agotó su nivel de vida, por haber caído en un precipicio o por haber sido asesinado por el Wumpus.

## Representación del conocimiento

Como se ha expresado, el conocimiento del juego se estructura, esencialmente, en forma de matriz, donde cada celda representa una habitación del entorno. En cada celda se almacena la información de los elementos que ella contiene. Además de la posición en el entorno tridimensional del juego, cada celda incluye:

- **Estado:** incluye uno o varios de los valores: brisa, hedor, resplandor, precipicio, Wumpus, poción o null.
- **Visitada:** permite saber si una casilla fue visitada o no.
- **Segura:** Son las no visitadas pero que el agente sabe que no presentan un peligro para él.
- **Posible\_Precipicio:** es verdadera cuando existe la posibilidad de que en ella haya un precipicio.
- **Posible\_Wumpus:** es verdadera cuando existe la posibilidad de que en ella se encuentre el Wumpus.
- **varDecision:** Representa, en un valor numérico, cuan segura o insegura es la celda, la de menor valor será la más segura.

Aunque la celda incluye todas estas variables, no todas son utilizadas a la vez por los distintos personajes. El jugador conoce los valores reales de la celda, es decir su estado. Sin embargo, el agente no conoce los valores reales que tiene la celda, sino que lleva su propia matriz donde va guardando la información que deduce de su andar por el entorno. A medida que visitan las celdas, se incorporan elementos a su base de conocimiento que le servirán para determinar la próxima acción a realizar para lograr su objetivo.

El agente, cada vez que llega a una habitación, consulta la celda para saber qué información tiene de la habitación donde se encuentra, la marca como visitada y guarda esta información en su propia matriz. Esta matriz es el conocimiento que va adquiriendo el agente sobre el entorno y se irá actualizando a medida que el agente avance por las distintas habitaciones.

## | Razonamiento

El razonamiento es el proceso que ayuda a la toma de decisión y consiste en deducir información a partir de la ya existente. El agente toma las decisiones a partir del razonamiento que realice con los datos que presenta. Luego de almacenar en su base de conocimiento (matriz) la información de la celda visitada, realiza conjeturas de los posibles estados en las celdas adyacentes, teniendo en cuenta las reglas del juego y los valores de las celdas visitadas anteriormente. Este análisis permite marcar celdas no visitadas como: posible\_ precipicio, posible\_ Wumpus o segura.

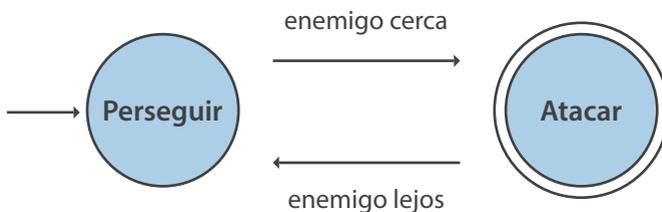
En la opción **Jugar** el guardián utiliza una técnica gráfica de los videojuegos, el *collider*, para detectar cuándo el jugador se encuentra cerca de él y detenerse a dispararle. En caso de que se aleje continuará persiguiendo al jugador. En el **Modo Agente** el personaje obtiene

información de la habitación donde se encuentra para almacenarla en su base de conocimiento y poder decidir hacia qué habitación dirigirse.

Se tuvo en cuenta que el personaje no saliera del tablero con posiciones no válidas como por ejemplo  $x$  igual a un número negativo. Posteriormente el agente va analizando la información que tiene para actualizar los valores en su base de conocimiento. Luego de realizar este análisis, se realiza una revisión de los valores de cada celda para garantizar que la información actualizada no contradiga las reglas del juego. Por último, el agente decide hacia dónde debe ir.

Para el caso del guardián del mundo del Wumpus se utiliza en el razonamiento una sencilla máquina de estados con dos estados (Figura 10):

- **Perseguir:** el guardián tendrá como destino la posición del jugador. En caso de que colisione con el jugador (enemigo cerca) entonces pasará al estado atacar.
- **Atacar:** el guardián disparará al jugador hasta que el jugador no colisione con él (enemigo lejos) o hasta que el jugador muera y termine el juego. Si el jugador se aleja a cierta distancia el guardián pasa al estado perseguir.



**Figura 10.** Máquina de estado del guardián del Mundo de Wumpus.

Para lograr el movimiento realista en el entorno se utilizan los comportamientos de locomoción. Esta técnica se incorpora en las

principales bibliotecas de IA utilizadas en los motores de videojuegos. Las versiones de Unity más recientes incorporan una biblioteca que brinda varios elementos relacionados con la IA. El componente NavMeshAgent es el encargado del movimiento realista por el entorno. Permite que el objeto que se moverá llegue al punto especificado por el camino más corto.

En la opción **Jugar** el guardián debe perseguir al jugador por lo que se especifica como destino la posición del jugador. En el caso del agente en el **Modo Agente** debe avanzar hacia la celda que haya decidido, por lo que se le especifica como destino el centro de dicha celda.

### Tratamiento de incertidumbre

Dado que el agente no conoce todo el terreno, debe tener implementado un mecanismo de tratamiento de incertidumbre para decidir hacia qué habitación dirigirse en cada momento. El agente en caso de encontrarse ante el problema de no tener casillas seguras en las posiciones adyacente entonces debe seleccionar la opción que más posibilidades de ganar tenga. Para esto tendrá en cuenta las clasificaciones realizadas en el razonamiento.

Cada celda tendrá tres variables booleanas (segura, posibleWumpus y posiblePrecipicio) y ocho posibles estados:

- $E_1$ : vacía     $E_3$ : brisa     $E_5$ : poción     $E_7$ : Wumpus
- $E_2$ : null     $E_4$ : hedor     $E_6$ : resplandor     $E_8$ : precipicio

Se define un índice de peligrosidad (indp) para cada celda, el cual será determinado de la forma siguiente:

$$indp = \begin{cases} 2 & \text{para } E_i \quad \setminus t = \{2, \dots, 6\} \\ 5 & \text{para } E_i \quad \text{donde } t = \{7, 8\} \\ 1 & \text{para } E_1 \quad \text{donde } segura = true \\ 4 & \text{para } E_1 \quad \text{donde } posibleWumpus = true \vee posiblePrecipicio = true \end{cases}$$

Para evitar que el jugador realice el mismo camino de forma infinita con el objetivo de no morir, se define la variable **Visitada** como un entero. Cada vez que se visite dicha casilla el valor de esta variable aumenta. Por lo tanto, se determina que el valor de la variable **varDecision** es el índice de peligrosidad visto anteriormente más el valor de la variable visitada por un valor de importancia.

$$\text{varDecision} = \text{indp} + \text{visitada} * \varepsilon$$

El valor de  $\varepsilon$  inicialmente se puede fijar en 0.5 para disminuir el valor de la variable *visitada* y tenga menor peso que el estado de la celda.

## Aprendizaje

El aprendizaje podemos verlo como la manera que tiene el agente de tomar decisiones en base a experiencias pasadas, lo que implica en cada momento guardar la información necesaria para reutilizar en el futuro. El agente aprende a partir de su interacción con el entorno y las consecuencias de las acciones realizadas. A medida que el agente realiza el razonamiento para tomar una decisión, lo va almacenando en su propia matriz para tenerlo en cuenta en el futuro. Si visitó una casilla, la información de la misma se almacena en su base de conocimiento y con esto podrá determinar qué acción realizar. Se muestra entonces que la información que conoce es aprendida por el agente y utilizada para lograr el objetivo del juego.

Al agente se le especifica un comportamiento de aprendizaje. Este comportamiento es sencillo y se basa en revisar las reglas del juego para eliminar inconsistencias y eliminar ruido en la matriz de conocimiento del agente. Por ejemplo, en el juego solo existe un Wumpus, por lo tanto, si ya se definió en qué casilla está el Wumpus

las demás que estén marcadas como **posibleWumpus** se marcarán como **segura**. También se modificará la variable **wumpusEncontrado** como verdadera.

En las posiciones adyacentes al Wumpus hay hedor y en las adyacentes al precipicio se percibe brisa, por lo tanto, al definirse la casilla del Wumpus entonces las adyacentes a él, aunque no hayan sido visitadas tendrán ahora como estado **Hedor**. Lo mismo sucede para el caso del precipicio.

Como puede verse para el aprendizaje no se utiliza una técnica específica tradicional. Esto es muy común en aplicaciones como los videojuegos, donde se desarrollan técnicas y algoritmos para lograr un comportamiento racional en los personajes y en el juego de forma general. Las técnicas complejas muchas veces atentan contra la eficiencia computacional, lo cual es muy importante en estas aplicaciones que deben procesar gráficos, lo cual compite con el tiempo de procesamiento de la inteligencia.

---

## CONSIDERACIONES FINALES

---

La Inteligencia Artificial hoy tiene impacto significativo en, prácticamente, todas las ramas de la ciencia y la sociedad. Cada uno de sus cuatro núcleos de conocimiento: el razonamiento, la representación del conocimiento, el tratamiento de la incertidumbre y el aprendizaje, se integran para lograr sistemas con un comportamiento racional.

Los sistemas basados en el conocimiento han estado estrechamente vinculados al propio desarrollo de la Inteligencia Artificial, de ahí que es común verlos como una temática que se aborda en las asignaturas universitarias de IA. Sin embargo, se aprecia cómo en su concepción y desarrollo es necesario la utilización de los cuatro núcleos de conocimiento de la IA.

La minería de datos, por su parte, es una de las principales áreas de aplicación de la IA en nuestros días. Es común que se asocie directamente al aprendizaje automático, sin embargo, pudimos apreciar cómo en el desarrollo de un sistema, son necesarios, además de aplicar los algoritmos de aprendizaje automático, los otros núcleos de conocimiento de la IA. Es necesario previamente representar o estructurar el conocimiento disponible, no solo los datos y la información, también elementos de dominio que permitan una correcta aplicación de los algoritmos. Es necesario tratar la incertidumbre

que cada vez está más presente, tanto en los datos que se van a analizar, como en los resultados que se obtienen de los algoritmos aplicados. Todo este proceso lleva un razonamiento computacional a partir de estrategias que permitan escoger los datos adecuados, los parámetros para cada algoritmo e incluso el algoritmo a utilizar en un momento determinado.

El procesamiento de lenguaje natural también ha sido considerado por algunos autores como núcleo básico de la IA. La primera contradicción con esta posición es que realmente no es necesario para infinidad de aplicaciones. Sin embargo sí se evidencia que este proceso necesita una fuerte representación del conocimiento que se desee procesar. Es impensable hoy un sistema de propósito tan general que no requiera información sobre un dominio específico. El razonamiento con tratamiento de la incertidumbre es esencial para tomar las decisiones, tanto en la interpretación de frases, como para elaborarlas y dar respuestas coherentes. Se evidencia la necesidad de un aprendizaje con memoria a largo plazo para el aprendizaje del contenido del dominio específico y un aprendizaje, con memoria a corto plazo, para el seguimiento de la información con un interlocutor específico.

Los videojuegos son una de las áreas de aplicación más apasionantes y abarcadora de la Inteligencia Artificial en nuestros días. Si a eso le sumamos que el desarrollo de un comportamiento racional en un personaje de un videojuego es, computacionalmente hablando, similar al de un robot en la vida real, se hace aún más interesante para los que van adquiriendo conocimientos de IA. La diferencia sustancial entre un personaje en videojuego y un robot, es el entorno y por tanto las vías de percepción y actuación en dicho entorno. Quiere esto decir que es similar la forma de representar el conocimiento que se percibió del entorno, es similar el tratamiento de la

incertidumbre que se realiza, son similares las vías de aprendizaje utilizadas y por tanto el razonamiento para tomar decisiones.





## 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

---

- Coca, Y. (2003).** URS-HTA: Sistema para el diagnóstico de la hipertensión arterial usando razonamiento basado en casos en condiciones de incertidumbre. Universidad Central de Las Villas.
- Coca, Y., Rosell, L. B., & Velazquez, A. (2017).** Modelo de agente lógico con inferencia basada en hechos. *Revista Cubana de Ciencias Informáticas*, 11(2), 29–45.
- Cuza, C. (2019).** Videojuego El mundo de Wumpus basado en técnicas de Inteligencia Artificial. Universidad de las Ciencias Informáticas.
- Gálvez, D. (1998).** Curso de Sistemas Basados en el Conocimiento. Grupo de Investigación en Inteligencia Artificial del Departamento de Ciencia de la Computación. Facultad de Matemática, Física y Computación. Universidad Central “Martha Abreu” de Las Villas. Cuba
- Gutiérrez, J. A., Cedeño, Y., & Coca, Y. (2017).** Sistema multiagente para la extracción de conocimiento en bases de datos (Universidad de las Ciencias Informáticas). Retrieved from <https://repositorio.uci.cu/jspui/handle/123456789/9398>
- Mitchell, T. M. (1997).** *Machine Learning*. McGraw-Hill.

- Rojas, C. A., Coca, Y., Núñez, S., & Batista, R. (2018).** Sistema Tutorial Inteligente para la Plataforma educativa XAUCE ZERA. Universidad de las Ciencias Informáticas.
- Russell, S., & Norvig, P. (2004).** Inteligencia Artificial. Un enfoque moderno (2da Edició). Madrid: Pearson Educación.
- Velázquez, A., Rosell, L. B., Coca, Y., & Cruz, O. (2016).** Sistema inteligente para la resolución práctica de problemas de Inteligencia Artificial utilizando la programación lógica (Universidad de las Ciencias Informáticas). Retrieved from <https://repositorio.uci.cu/jspui/handle/123456789/7763>

*"Aplicaciones de la Inteligencia Artificial"* es el último 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. Cada capítulo muestra un ejemplo de problema y su solución.

El primer capítulo expone un sistema basado en conocimiento. Estos están tan estrechamente relacionados con la Inteligencia Artificial que en muchas asignaturas de carreras afines a la informática se incluye como un tema a estudiar. El segundo capítulo presenta un sistema multiagente para tareas de minería de datos. Se utiliza una metodología específica para desarrollar el sistema y se utiliza una reconocida biblioteca de algoritmos, de las tantas que existen hoy en diversos lenguajes de programación. El tercer capítulo aborda uno de los problemas más complejos y que sigue siendo un reto en nuestros días, el procesamiento de lenguaje natural. Se aborda desde el desarrollo de un sistema tutorial inteligente en forma de chat que brinda ayuda y evalúa el avance de los estudiantes. El cuarto capítulo presenta un tema de gran interés para los jóvenes, los videojuegos. Además, es una de las tareas de mayor integralidad en la aplicación de la Inteligencia Artificial.

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: 978-959-18-1343-5

