

# INTELIGENCIA ARTIFICIAL

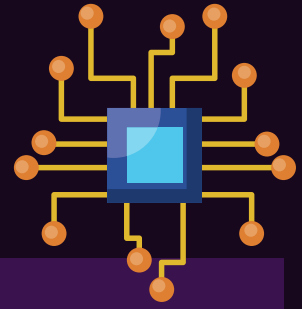
meta 4.1.2: Técnicas de Representación  
del Conocimiento  
garcia chavez erik 01275863  
ingeniería en computacion  
ciclo 2025-1

## Introducción

las técnicas de razonamiento nos ayudan a enseñarle a la computadora el cómo aprender, si queremos que la maquina logre pensar como un experto en cierta área, a que comprenda y relaciona conceptos en un texto complejo o tomar decisiones en tiempo real. las técnicas como los marcos (**frames**), redes semánticas (semantic Nets), y reglas de producción, lo hacen posible y también tratan de imitar la forma en como los humanos estructuramos la información mentalmente, estas técnicas transforman datos crudos en conocimiento accionable, acercándonos a sistemas más inteligentes y autónomos.



## Contenidos generales



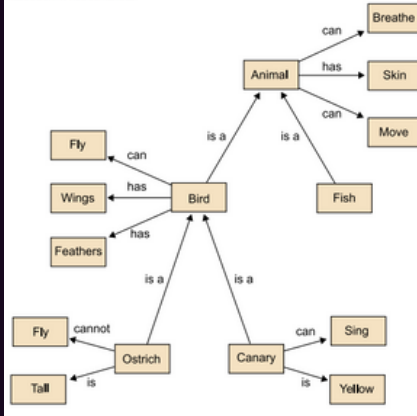
**1** Redes semánticas  
(**Semantic Nets**)

**2** marcos (Frames)

**3** Reglas de produccion

## redes semánticas

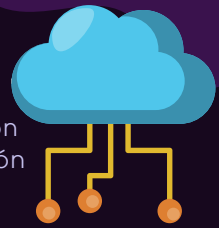
Ilustración 4. Red semántica



este concepto tiene su origen en 1909 con Charles Peirce quien propuso una notación gráfica de nodos y arcos denominada "grafos existenciales".

las redes semánticas es un método de representación del conocimiento basado en relaciones entre objetos. este conocimiento puede ser representado mediante un grafo.

los nodos corresponde a los objetos y los arcos describen las relaciones entre los objetos. cada arco con sus dos nodos forman una tripleta (sujeto, relaciones, objeto). la red solo almacena las relaciones estaticas mas no como usar es infomacion



la semantica intenta describir el segnificado de las palabras y las condiciones en las que los significados pueden interactuar para ser compatibles con los otros aspectos del lenguaje.

una relacion puede ser uni o bidireccional, si se da el caso

tenemos el ejemplo : La red puede indicar que "Lassie es-un Perro" y "Perro es-un Mamífero", pero no explica cómo inferir que "Lassie es-un Mamífero".

una red semantica se ve mas como dar la forma la estrucutra del aprenidaje, pero es necesario algo mucho mas potente apoyandonos con una red semantica, ya que se hacerca mucho a como la mente humana relaciona los objetos a su al rededor y puede deducir todo eso, pero la mente humana lo hace implicitamente

## Marcos (Frames)

un marco (frame) es una colección de atributos que normalmente son llamados ranuras (slots), con valores asociados y posiblemente restricciones entre valores, estos describen alguna entidad del mundo, algunas veces el marco describe una entidad en un sentido absoluto y en otras representa la entidad desde un punto de vista particular



frames unicos no son de gran utilidad:

-Un marco de "automóvil" puede tener un atributo motor, pero para entender qué es un motor, necesitas otro marco que lo detalle.

sistema de marcos:

- Marco "Persona" → Atributo trabajo → Marco "Empleo".
- Marco "Empleo" → Atributo empresa → Marco "Empresa".

2 - un frame unico particularmente no es util, esto porque este describe un un concepto con atributos basico, pero el cocimeinto real es relacional, jerarquico y conextual. por lo que estos siempre estan conectados con otros frames

los sitemas de marcos son colecciones de marcos vinculados, sonde el valor de un atributo en un marco puede ser otro marco

## Marcos (Frames)

fue propuesto por Minsky en 1975 como una forma de representación del conocimiento, está inspirado en como el humano capta y almacena el conocimiento de su entorno.



la idea original se expresa como < Un Frame es una estructura de datos para representar una situación estereotipada, podemos pensar en un Frame como una red de nodos y relaciones. Los 'niveles superiores' de un Frame son fijos y representan cosas que son siempre ciertas en la situación supuesta. Los niveles inferiores tienen muchos terminales - 'slots' que tienen que llenarse con instancias o datos específicos. Las suposiciones por defecto son asociadas débilmente con sus terminales, de forma tal que ellas puedan ser fácilmente removidas ante nuevos elementos que se ajusten mejor a la situación actual.>

los marcos trata de imitar como el humano aprende, un slot es un concepto del mundo real, yo sé que compone ese objeto del mundo real y cómo interactúa con su entorno otros slots, nosotros lo hacemos de manera implícita, nuestro cerebro lo procesa automáticamente

que nos quiere decir la idea original:

que un frame es una plantilla que captura cómo es una situación típica por ejemplo un restaurante y sus atributos como ubicación, tipo, precio, horario, etc.

el frame no es una lista sino más bien como una red estructurada donde los atributos (nodos) están conectados entre sí.

podemos pensar un frame incluso como alguna clase padre, en donde nos dice que la parte superior es fija, son los atributos, la información

inmutable, esencial sobre un concepto, como en POO esta tendría sus clases hijas que son los niveles inferiores, los niveles inferiores heredan

de la parte superior sus atributos pero la parte baja se adapta a casos concretos.

los slots tienen valores predeterminados, pero pueden ser cambiarse si la realidad es

distinta como un ave, tendríamos que pensar que un ave vuela pero

por ejemplo el pinguino es un ave pero no vuela.

## Marcos (Frames)

podemos considerar los frames como una red semántica con numero de posibilidades mucho mayor entre las que destacan especialmente, la capacidad de activación de procesos, y de herencia nomonotonica mediante sobre control, en la que un nodo hijo hereda todos los slots de su padre a menos que se especifique lo contrario.



## Marcos (Frames)

características:

- precisión: los objetos, las relaciones entre objetos y sus propiedades se describen de forma precisa.
- activación dinámica de procesos: es posible adjuntar procedimiento a un frame o algunos de sus componentes de forma que se llamen y ejecuten automáticamente tras la comprobación de cambio de valor o propiedad
- herencia por defecto no-monotónica: Los marcos están conceptualmente relacionados, permitiendo que los atributos de los objetos sean heredados de otros objetos predecesores en la jerarquía.
- modularidad: La base de conocimiento está organizada en componentes claramente diferenciados.

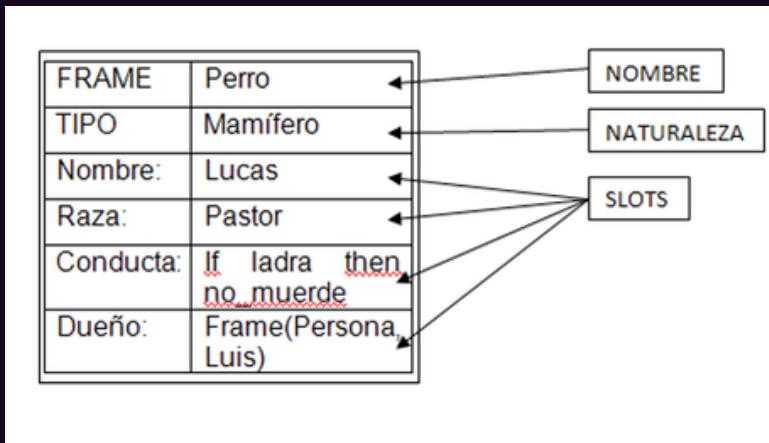
estructura:

- nombre: el cual debe ser único
- padre o naturaleza: equivale a la clase superior a la cual pertenece y es la que le permite el mecanismo de herencia. El frame raíz no tiene padre y normalmente se identifica su tipo con algún atributo, en nuestro caso superior.
- slots o atributos: Constituye el cuerpo y cada uno representa una propiedad estándar o atributo del elemento representado.





## Marcos (Frames)



estructura de un Slot:

- nombre: cada slot debe de tener un nombre unico dentro del frame
- valor del slot: puede ser asignado inicialmente o estar vacío para añadirle valor después
- procedimiento anexos:
  - if - needed: significa que antes de obtener el valor, se ejecuta el procedimiento adjunto y solo si tiene éxito se podrá obtener el valor
  - if - added : antes de asignarle un valor al slot, debe de ejecutarse con éxito el procedimiento, de otra manera no se realizará alguna acción
  - if - deleted: antes de borrar el valor del slot, debe de ejecutarse con éxito el procedimiento, de otra manera no se borrará el valor del slot

un marco de puede ver como una relacion de POO o incluso con BD, muy parecido en su estructura

# Reglas de produccion

## Ejemplos:

### Regla Médica:

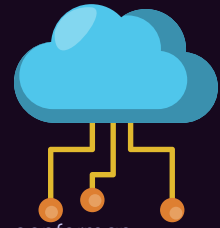
*Si una persona tiene fiebre, tos y dificultad para respirar, ENTONCES diagnosticar posible COVID-19.*

### Regla de un Sistema Experto para Configuración de Equipos:

*Si la impresora está fuera de línea ENTONCES verificar la conexión USB.*

### Regla de un Juego:

*Si el jugador tiene tres cartas iguales ENTONCES puede bajarlas a la mesa.*



también denominadas reglas IF-THEN. conforman una de las técnicas de representación del conocimiento más antiguas. Estas surgen entre la década de los 70's con los primeros sistemas expertos, su simplicidad las hizo populares para modelar conocimiento humano en forma de reglas explícitas.

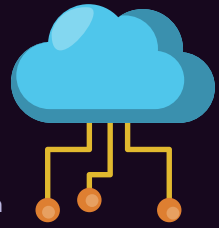
su estructura es simple se vincula a:

**condiciones** (*antecedentes*) con **acciones o conclusiones** (*consecuencias*).

**SI** (condición X se cumple **{IF}** ) **ENTONCES** (ejecutar acción Y o inferir conclusión Z **{THEN}** ).

## Reglas de produccion

es un paradigma de representación aplicado en sistemas expertos, en este sistema los programas emulan la toma de decisiones de un humano partiendo de igual forma como lo haría un humano, tiene el conocimiento necesario para hacerse de igual forma las preguntas si el paciente tiene X condición entonces Debería de tener X tambien y sucesivamente hasta encontrar la respuesta. un ejemplo de un sistema experto puede ser MYCIN un sistema experto médico de los años 70's que diagnosticaba infecciones bacterianas usando reglas.



Ejemplo:

Si "el paciente tiene tos persistente por más de 3 semanas"

Y "pérdida de peso inexplicable"

ENTONCES "sospechar de tuberculosis y ordenar una prueba de esputo".



es muy popular para diagnosticar patologías dentro de la medicina por su naturaleza, las reglas IF-THEN permiten codificar el conocimiento médico en pasos lógicos, vinculando síntomas, pruebas de diagnóstico y tratamientos. como por ejemplo:

## Reglas de produccion: tipos

### encadenamiento hacia adelante (Forward chaining):

el sistema comienza con hechos iniciales y aplica reglas para inferir nuevos hechos hasta llegar a una conclusión

ejemplo:

Hecho : "Juan es un hombre"

Regla: SI una persona es un hombre

ENTONCES es un ser humano

conclusión: "Juan es un ser humano"

### encadenamiento hacia atrás (Backward Chaining):

si el sistema comienza con una hipótesis o conclusión y trabaja hacia atrás para ver si los hechos la apoyan

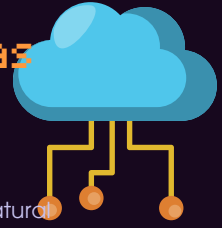
Hipótesis: "Juan es un ser humano"

Regla: "Si una persona es un ser humano

Entonces es un hombre o una mujer"

verificación: "Confirmamos que 'Juan es un hombre' lo que soporta la hipótesis inicial"

# Reglas de producción: ventajas y desventajas



## ventajas:

- permiten representar conocimiento declarativo y procedimental de manera conjunta
- es modular, por lo que se puede añadir, quitar o modificar reglas sin altera todo el sistema
- la forma IF-THEN de la regla tiene con frecuencia una buena correlación con el lenguaje natural
- en algunos sistemas se ha visto que los sistemas basado en reglas consiguen algo parecido al modelo que el hombre utiliza para solucionar por sí mismo el problemas

## desventajas:

- si hay muchas reglas en un sistema algunas pueden solaparse, contradecirse o activarse en cadena de forma imprevista
- las reglas suelen almacenarse como una lista plana sin organización jerárquica o temática. cuando hay miles de reglas se vuelve caótico modificarlas, depurarlas o entender cómo se relacionan entre sí
- no todos los métodos de resolución se adaptan a las reglas IF-THEN. en muchas ocasiones usamos instrucción, experiencia o aprendizaje por analogía, procesos más difíciles de capturar con reglas simples
- validar un sistema basado en reglas implica asegurar de que todas las combinaciones posibles de reglas funcionen correctamente. pero en los sistemas grandes Es casi imposible probar todas las interacciones entre reglas lo que puede dejar errores ocultos.

aplicaciones con fames:

Sistemas de simulación (ej.: representar comportamientos de objetos en un entorno virtual).

Bases de datos orientadas a objetos.

Modelado de escenarios cotidianos (ej.: diagnósticos médicos con síntomas típicos).

aplicaciones con redes semanticas:

Procesamiento de lenguaje natural (entender relaciones en textos).

Sistemas de búsqueda inteligente (ej.: motores de conocimiento como Wikidata).

Ontologías para web semántica.

aplicaciones con reglas de produccion:

Diagnóstico médico o industrial.

Automatización de procesos empresariales (ej.: aprobación de créditos).

Control de sistemas (ej.: regulación de temperatura en tiempo real).



## Bibliografía

- <https://inteligenciartificialmca.wordpress.com/2017/02/28/2-marcos-frames/>
- <https://sistemasinformatica.istmo.tecnm.mx/cursos/inteligencia-artificial/3-reglas-y-busqueda/3-3-reglas-de-produccion/>
-