# Arquitectura de la Memoria de Mapa Conceptual (MMC)

## Introducción

En las últimas décadas, el diseño de **memorias artificiales** eficaces se ha convertido en un desafío central para la inteligencia artificial (IA) y las **arquitecturas cognitivas**. Los agentes inteligentes requieren almacenar, organizar y recuperar conocimiento de forma similar a la memoria humana, integrando nueva información sin olvidar lo aprendido. Sin embargo, las aproximaciones tradicionales a la memoria en IA presentan limitaciones. Por un lado, las **redes neuronales** tienden a sufrir *olvido catastrófico* – pierden conocimiento previo al aprender nueva información (problema de estabilidad-plasticidad) file-unccncqpe5yrcpgsbp39anfile-unccncqpe5yrcpgsbp39an. Por otro lado, los sistemas simbólicos rígidos (p. ej., bases de conocimiento estáticas) carecen de la flexibilidad para **adaptarse dinámicamente** a entornos cambiantes. En este contexto, se identifica un *vacío*: **¿cómo construir una memoria artificial que sea a la vez estructurada, adaptativa, resiliente al olvido y protegida contra manipulaciones?**

La **Arquitectura de la Memoria de Mapa Conceptual (MMC)** se propone como respuesta innovadora a ese vacío. La MMC concibe el conocimiento como una **red dinámica de nodos conceptuales interconectados** por enlaces ponderados que evolucionan con la experiencia. Esta idea se inspira en los *mapas conceptuales* de Novak, representaciones gráficas de conocimiento donde los conceptos (nodos) se conectan mediante relaciones etiquetadasfile-unccncqpe5yrcpgsbp39an. A diferencia de los mapas conceptuales estáticos empleados en educación (Novak & Cañas, 2008), la MMC implementa una **memoria viva**: sus nodos y conexiones se **reconfiguran plásticamente** con el uso, reforzándose las asociaciones frecuentes y atenuándose – mas no eliminándose de inmediato – las infrecuentes. En otras palabras, la MMC incorpora un mecanismo de *olvido controlado* mediante **hibernación** de elementos inactivos en lugar de un borrado abruptofile-unccncqpe5yrcpgsbp39an. Este enfoque refleja principios de la psicología cognitiva, como la teoría de la activación difundida en redes semánticas (Collins & Loftus, 1975) y la regla de Hebb sobre el aprendizaje asociativo (Hebb, 1949), aplicados a una memoria artificial.

La relevancia de la MMC radica en que aborda simultáneamente varios retos pendientes en memorias de IA. En primer lugar, propone una solución al dilema *exploración vs. explotación* en la gestión del conocimiento: combina **rutas de recuperación principales** (basadas en enlaces fuertes, explotando conocimiento establecido) con **rutas laterales creativas** (explorando conexiones débiles o latentes para encontrar asociaciones novedosas)file-unccncqpe5yrcpgsbp39an. Esto dota al sistema de creatividad controlada, evitando tanto la rigidez excesiva como la generación de información aleatoria irrelevante. En segundo lugar, la MMC integra desde su núcleo unos **fragmentos nativos** inmutables – piezas de conocimiento intrínseco como principios éticos y la identidad del sistema – que sirven de anclajes estables para guiar su aprendizaje y comportamiento. Esta característica, inspirada en las *leyes de la robótica* y en la noción de instintos o predisposiciones innatas en seres biológicos, busca garantizar que la memoria conserva fundamentos seguros y alineados con valores, aun tras extensas modificaciones. En tercer lugar, a diferencia de la mayoría de trabajos previos, se introduce un **mecanismo de autoprotección** de la memoria: la integración de un sistema de cifrado y control de acceso para resguardar los fragmentos más sensibles (como los éticos) de manipulaciones externas malintencionadas. La incorporación de esta capa de seguridad responde a la creciente preocupación por la integridad y confianza de los sistemas de IA (IEEE, 2019; UNESCO, 2021).

En síntesis, esta tesis explora y desarrolla la **Arquitectura MMC** como una memoria artificial **estructurada pero flexible**, **auto-organizada**, *reflexiva* y *segura*. Se plantea que un sistema basado en MMC será capaz de **aprender continuamente** sin olvidar de forma patológica lo antiguo, **reorganizar su conocimiento** en comunidades temáticas, **proteger su núcleo identitario y ético**, y **adaptarse evolutivamente** a nuevas condiciones. Para investigar esta propuesta, se combinan el **desarrollo teórico** – formalizando componentes y procesos de la MMC con apoyo en literatura científica – y la **experimentación práctica** – mediante un modelo simulado de la MMC implementado en software. Así, la presente investigación no solo describe conceptualmente la MMC, sino que ofrece **diseños de implementación** (e.g., uso de grafos de conocimiento y vectores semánticos) y evidencia inicial de su funcionamiento mediante un prototipo.

**Establecimiento del territorio:** Existe un rico cuerpo de trabajo en representación del conocimiento y memorias en IA, desde arquitecturas cognitivas clásicas como *Soar* (Laird et al., 1987) y *ACT-R* (Anderson et al., 2004) hasta desarrollos recientes en *memorias diferenciables* (Weston et al., 2015) y *grafos de conocimiento* a gran escala. No obstante, **ninguna de estas soluciones cumple completamente con la visión integradora que aquí se plantea**. *Soar* y *ACT-R* ofrecen mecanismos de memoria de trabajo y memoria declarativa basados en símbolos, pero tienden a requerir una ingeniería de conocimientos fija y presentan reglas estáticas, con capacidad limitada de reorganización espontánea del conocimiento. Las memorias neuronales, por su parte, logran aprendizaje continuo, pero suelen carecer de interpretabilidad y de garantías contra el olvido abrupto o el desalineamiento ético. *OpenCog* (Goertzel & Hart, 2008) introduce un grafo de conocimiento (AtomSpace) con pesos que modelan atención, pero su complejidad y enfoque en AGI general difieren del objetivo específico de una memoria auto-mantenida. *Semantic Pointer Architecture* (SPA; Eliasmith et al., 2012) y otros modelos neuroinspirados logran representación combinatoria mediante vectores de alta dimensión, aunque no abordan explícitamente aspectos como la protección de contenidos sensibles o la incorporación de nociones innatas de identidad y ética. **Si bien** la literatura muestra *avances parciales* en cada frente (memoria adaptativa, redes semánticas dinámicas, integración de valores, etc.), **todavía son escasas las propuestas** que los combinen en una arquitectura unificada. **Así, esta investigación ocupa el nicho** de diseñar y evaluar una memoria artificial integral que supere estas brechas.

**Propósito de la investigación:** *Ocupar el nicho identificado* mediante la formulación de la MMC y demostrar su viabilidad. La tesis presenta los fundamentos teóricos de la MMC, detalla su diseño de componentes y procesos, la compara con enfoques existentes y exhibe un modelo prototipo en funcionamiento. Con ello, se busca evidenciar que es posible construir un sistema de memoria artificial **auto-organizado, persistente y confiable**, sentando bases para una próxima generación de arquitecturas cognitivas más **autónomas y seguras**. En última instancia, la MMC aspira a contribuir al desarrollo de **IA más avanzadas y transparentes**, capaces de gestionar su propio conocimiento de forma similar a un **cerebro artificial** que aprende, recuerda y evoluciona a lo largo del tiempo.

## Antecedentes

### Marco teórico: Memorias, grafos cognitivos y olvido controlado

El concepto de estructurar el conocimiento como una **red de nodos y enlaces** tiene raíces profundas en la ciencia cognitiva y la informática. Los **mapas conceptuales**, introducidos por Novak en la década de 1970, fueron concebidos como una herramienta para representar visualmente relaciones entre conceptos en la mente de un aprendiz (Novak & Cañas, 2008). En un mapa conceptual clásico, cada nodo representa un concepto y los enlaces denotan relaciones proposicionales (por ejemplo, *“X causa Y”* o *“X es un tipo de Y”*). Estas representaciones han demostrado ser eficaces para organizar y comunicar conocimiento, y sirven de inspiración para la MMC en cuanto a la **naturaleza relacional del conocimiento**. No obstante, los mapas conceptuales tradicionales son estáticos; la MMC extiende esta idea hacia un **sistema dinámico**, donde los nodos conceptuales y sus conexiones **cambian con la experiencia**.

Otra piedra angular teórica es la idea de **red semántica** en psicología cognitiva. Collins y Loftus (1975) propusieron una “teoría de la activación propagada” (*spreading activation*) de la memoria semántica, según la cual pensar en un concepto activa sus nodos asociados en una red mental, propagándose la activación en cascada a través de las conexiones. Este mecanismo explica fenómenos como la *priming* (facilitación para recordar una palabra tras haber visto una relacionada) y sugiere que la memoria humana opera mediante **asociaciones graduales** más que por búsqueda exacta. La MMC incorpora este principio mediante la **difusión de activación**: cuando un nodo se activa (p. ej., por una consulta del usuario), también incrementa ligeramente la activación de los nodos vecinos conectados, facilitando la recuperación de conceptos relacionados. Esta difusión amplificada se combina con una exploración deliberada de vínculos lejanos para modelar tanto la memoria lógica (recuperar lo más relevante) como la memoria creativa (hallar conexiones inesperadas). Teóricamente, se alinea con la noción de que la creatividad es a menudo fruto de *asociaciones remotas* en la red de memoriafile-unccncqpe5yrcpgsbp39anfile-unccncqpe5yrcpgsbp39an.

El **aprendizaje Hebbiano** proporciona otro cimiento clave. Planteado por Donald Hebb en 1949, establece que *“neuronas que disparan juntas, se conectan juntas”* (Hebb, 1949). Traducido a una red de memoria artificial, implica que si dos conceptos se usan o invocan conjuntamente de forma frecuente, el vínculo entre ellos debe reforzarse. La MMC implementa una regla de actualización de pesos inspirada en Hebb: cada vez que dos nodos se activan simultáneamente en un contexto de uso, el peso de la conexión entre ellos se incrementa ligeramente (refuerzo positivo). Por el contrario, si pasan largos periodos sin co-activación, el peso decae exponencialmente con el tiempofile-4xuxyatu3sqxtrrzbkpyllfile-unccncqpe5yrcpgsbp39an. Esta *plasticidad sináptica* artificial permite que la memoria **adapte sus conexiones** según los patrones de uso: las asociaciones útiles se fortalecen (facilitando recordar juntas esas ideas en el futuro), mientras que las asociaciones irrelevantes se debilitan para no saturar la red. Tal mecanismo emula hallazgos de la psicología del olvido y de modelos como ACT-R, donde cada trozo de conocimiento tiene una base de activación que decae con la falta de uso (Anderson et al., 2004). De hecho, ACT-R utiliza una fórmula de decaimiento logarítmico según la cual la activación base $B\_i$ de un ítem $i$ es $B\_i = \sum\_j (t\_j)^{-d}$, sumando contribuciones de cada uso $j$ con un decaimiento dependiente del tiempo transcurrido $t\_j$ (Anderson & Schooler, 1991). La MMC puede integrar una fórmula análogafile-unccncqpe5yrcpgsbp39an, lo que situaría su comportamiento de olvido en línea con datos empíricos sobre la memoria humana (por ejemplo, la *curva de olvido* de Ebbinghaus). En términos prácticos, esto significa que un nodo en la MMC retiene una alta activación si ha sido utilizado recientemente y con frecuencia, pero irá disminuyendo su nivel de activación a medida que transcurra el tiempo sin ser utilizado, dificultando gradualmente su recuperación – aunque nunca desapareciendo por completo de la red.

El concepto de **hibernación de nodos** complementa al de activación decaída, proporcionando una forma de *olvido adaptativo*. En la naturaleza, la memoria humana realiza algún grado de descarte o archivado de recuerdos poco útiles, pero muchas veces esos recuerdos pueden resurgir con la pista adecuada. Inspirada en ello, la MMC no elimina directamente los nodos cuyas conexiones han perdido peso por falta de uso; en lugar de eso, los marca como *hibernados*. Un nodo hibernado permanece almacenado en la base de memoria pero es excluido de la “memoria activa” o de los cálculos cotidianosfile-unccncqpe5yrcpgsbp39anfile-unccncqpe5yrcpgsbp39an. Solo si el contexto lo vuelve relevante – por ejemplo, si ingresa nueva información relacionada o una búsqueda coincide parcialmente con su contenido – el sistema “despierta” el nodo, reactivándolo y devolviéndolo a la red activafile-unccncqpe5yrcpgsbp39an. Este mecanismo refleja la idea de **olvido reversible**: la memoria retira temporalmente elementos de baja utilidad para liberar recursos (por analogía con desplazar información de RAM a disco en sistemas computacionales), pero conserva *huellas* que permiten recuperarlos si se vuelven necesarios. Modelos de IA como OpenCog Prime implementan algo similar mediante la asignación de valores de *Atención de Corto Plazo* (STI) y *Largo Plazo* (LTI) a cada átomo de conocimiento, reduciendo STI para piezas inactivas pero manteniendo cierto LTI como memoria latente (Goertzel & Pennachin, 2007). La MMC toma inspiración de estas fuentes para lograr un equilibrio: **no saturarse** con información obsoleta pero tampoco **perder definitivamente** conocimiento potencialmente útilfile-unccncqpe5yrcpgsbp39anfile-unccncqpe5yrcpgsbp39an. Desde una perspectiva teórica, esto se relaciona con la **teoría de la consolidación** en neurociencia, según la cual los recuerdos a corto plazo se transforman gradualmente en memorias estables a largo plazo (McGaugh, 2000). La MMC podría incorporar fases periódicas de consolidación: durante períodos de inactividad (análogos al sueño) podría decidir qué conexiones débiles eliminar definitivamente y cuáles retener como latentes.

El **agrupamiento dinámico** o formación de *comunidades de conocimiento* es otro pilar teórico de la MMC. En redes complejas, es común observar la emergencia de **clusters** – subconjuntos de nodos más densamente interconectados entre sí que con el resto de la red. En el cerebro, por ejemplo, existen conjuntos neuronales especializados en ciertas funciones o temáticas (modularidad cerebral). En grafos de conocimiento, la detección de comunidades (por algoritmos como *Louvain* o *infomap*) permite identificar tópicos o contextos diferenciados dentro de la base de conocimientos (Blondel et al., 2008). La MMC adopta esta idea permitiendo que sus nodos se organicen en clusters **adaptativos** según sus patrones de activación y conexiónfile-unccncqpe5yrcpgsbp39an. Por ejemplo, con el tiempo podría surgir un cluster de “conocimiento matemático” agrupando fragmentos matemáticos interrelacionados, separado de otro cluster de “conocimiento de programación”, etc., si bien conectados entre sí a través de nodos puente. A diferencia de clasificaciones estáticas, estos clusters no son predefinidos ni inmutables: un nodo puede migrar de un cluster a otro si sus conexiones cambian (p. ej., un concepto inicialmente entendido como matemático que luego se usa más en contextos de programación podría recolocarse). Este comportamiento emergente recuerda a la **neuroplasticidad** donde áreas del cerebro pueden reorganizarse con la experiencia. Formalmente, la MMC puede implementar un algoritmo de agrupamiento continuo – ejecutando periódicamente Louvain u otro método sobre el grafo – para re-calcular comunidades a medida que el grafo evolucionafile-unccncqpe5yrcpgsbp39an. La presencia de clusters tiene beneficios teóricos y prácticos: modulariza el conocimiento facilitando la especialización (cada cluster actúa casi como una “memoria temática” más pequeña) y mejora la eficiencia de búsquedas al poder delimitar la propagación de activación principalmente dentro del cluster relevantefile-unccncqpe5yrcpgsbp39an. Además, desde el punto de vista de visualización e interpretabilidad, los clusters permiten entender cómo el sistema ha organizado su “mapa” de conocimientos.

Por último, un aspecto teórico novedoso de la MMC es la inclusión de **fragmentos de conocimiento especializados** desde la inicialización. Inspirado tanto por consideraciones de *alineación de IA* (ética incorporada) como por marcos de psicología evolutiva (conocimiento innato), la MMC define un conjunto de **Fragmentos Nativos** (FN) que representan conocimientos y principios fundamentales que el sistema *no debe olvidar ni alterar*. Entre ellos se contemplan: un **fragmento de identidad** (quién es el sistema, cuál es su propósito central), un **fragmento ético** (directrices de comportamiento seguro y valores incorporados), y un **fragmento de aprendizaje** (metaconocimiento sobre cómo aprende y evoluciona). Estos nodos FN están fuertemente interconectados entre sí (formando un núcleo cerrado) y marcados con un atributo de *inmutabilidad*. Adicionalmente, la MMC puede iniciar con *fragmentos especializados nativos* en ciertas áreas clave (por ejemplo, un fragmento nativo de código, uno de matemáticas y uno sensorial) para dotarla de competencias básicas en esos dominiosfile-unccncqpe5yrcpgsbp39anfile-unccncqpe5yrcpgsbp39an. Esta idea se relaciona con la noción de *módulos cognitivos* innatos propuesta en ciencias cognitivas (Spelke, 2000) – por ejemplo, la mente humana parece tener predisposiciones para reconocer caras o para el lenguaje. En la MMC, los fragmentos nativos aseguran que la memoria arranca con un esqueleto de conocimiento sólido sobre el cual construir. Desde el punto de vista de teorías de IA, esto se alinea con iniciativas de infundir valores y orientaciones a los sistemas inteligentes para evitar comportamientos indeseados (ver *Ethically Aligned Design*, IEEE, 2019). A nivel técnico, la presencia de nodos inmutables significa que el algoritmo de olvido nunca los eliminará ni reducirá sus conexiones por debajo de cierto umbral. Se comportan como “constantes” dentro del grafo, actuando de referencia estable. Por ejemplo, el nodo de ética podría estar conectado permanentemente a muchos otros nodos evaluando sus implicaciones; el nodo de identidad podría influir en la forma en que se prioriza información relativa a los objetivos del sistema, etc. Estos **anclajes internos** distinguen a la MMC de una memoria puramente acumulativa: le otorgan un *núcleo de coherencia* que guía su evolución, análogo a cómo los organismos biológicos tienen instintos o *valores homeostáticos* que se mantienen a lo largo de la vida (p. ej., evitar el daño, buscar alimento, etc.).

En resumen, el marco teórico de la MMC se construye sobre múltiples líneas de conocimiento: (a) representaciones gráficas del saber (mapas conceptuales, grafos semánticos), (b) mecanismos de activación asociativa y propagada, (c) aprendizaje por refuerzo hebbiano y decaimiento adaptativo (olvido no catastrófico), (d) organización modular en clusters dinámicos, y (e) inclusión de conocimiento nativo y principios éticos como parte integral de la memoria. Estos conceptos, respaldados por la literatura científica, sirven de base para diseñar una memoria artificial que **aprende continuamente sin perder estabilidad** – resolviendo en cierta medida el dilema estabilidad vs. plasticidad (Mermillod et al., 2013) – y que **combina información simbólica y subsimbólica**. Por ejemplo, la fusión de representación simbólica (nodos con significado definido) con *embeddings* vectoriales para similitud difusa, hace de la MMC un enfoque híbrido con paralelos en los *sistemas neuro-simbólicos*. En la siguiente sección se revisan algunas **arquitecturas de memoria y cognición existentes** que abordan partes de este problema, para situar a la MMC en contexto y extraer aprendizajes que han guiado su desarrollo.

## Arquitecturas de memoria artificial existentes y comparables

A continuación, se describen brevemente varias arquitecturas cognitivas y sistemas de memoria adaptativa desarrollados en la literatura, destacando sus características clave en relación con la representación del conocimiento, la dinámica de aprendizaje/memoria y otros mecanismos relevantes. Esta revisión comparativa servirá para posicionar a la MMC frente a enfoques previos:

* **Soar (State, Operator, And Result)** – Propuesto por Laird, Newell y Rosenbloom, Soar es una arquitectura cognitiva clásica orientada a lograr inteligencia general a través de la unificación de **memoria de trabajo** y **reglas de producción** (procedimientos) (Laird et al., 1987). En Soar, el conocimiento se representa mediante estructuras simbólicas gráficas en la memoria de trabajo que describen el estado actual del problema, mientras que la memoria a largo plazo consiste en reglas de producción (if-then) que aplican operadores. Un aspecto central de Soar es el *aprendizaje por chunking*\*: cuando el sistema encuentra un impase (un problema que no puede resolver con sus reglas actuales), entra en un proceso de subgoaling para resolverlo y luego crea una nueva regla (chunk) que resume esa experiencia, almacenándola para usos futuros. Este mecanismo le permite aprender de la experiencia, generando nuevas piezas de conocimiento (reglas) a partir de problemas resueltos – similar a formar “fragmentos” consolidados de conocimiento. Soar también puede integrar refuerzo: utiliza *preferencias* numéricas para elegir operadores, ajustables con feedback. En cuanto al olvido, Soar tradicionalmente **no** olvida reglas a menos que explícitamente se maneje, aunque las estructuras en memoria de trabajo se activan y desactivan por ciclo de decisión. Todas las estructuras simbólicas en Soar llevan implícitamente metadatos de recencia y frecuencia (a través del mecanismo de preferencia), lo cual influye su uso (Rosenbloom et al., 1993). En resumen, Soar es una arquitectura híbrida deliberativa-reactiva, con un **ciclo de decisión** bien definido y una fuerte separación entre memoria de trabajo (dinámica) y memorias a largo plazo (procedimental y semántica). **Relación con MMC:** La MMC comparte con Soar la idea de un grafo simbólico como representación unificada del estado/conocimiento. De hecho, la memoria de trabajo de Soar puede verse como un grafo de nodos-atributos que representa la situación actual – una idea afín a la MMC, solo que la MMC extiende esto a toda su memoria. Sin embargo, Soar no mantiene automáticamente un histórico de todas las situaciones (tiende a descartar la memoria de trabajo previa en cada ciclo), mientras que la MMC lo almacenaría en su red. Soar introduce meta-aprendizaje (chunking) para crear nuevas reglas; análogamente, la MMC busca **descubrir patrones** y nuevas conexiones a partir de sus experiencias (equivalente a crear “reglas” de asociación nuevas). Además, Soar no incorpora explícitamente nociones de olvido gradual ni fragmentos innatos; la MMC añade esos componentes. Aun así, Soar provee valiosas lecciones: la importancia de un **módulo decisional** que gestione la acción en base a la memoria (algo que la MMC como memoria necesitará en un agente completo) y el beneficio de adjuntar **estadísticas de uso** a las piezas de conocimiento – idea que la MMC retoma mediante pesos dinámicos y contadores de activaciones.
* **ACT-R (Adaptive Control of Thought – Rational)** – Desarrollada por John Anderson y colegas, ACT-R es una arquitectura cognitiva que modela la memoria humana diferenciando entre **memoria declarativa** (hechos, recuerdos) y **memoria procedimental** (habilidades, producciones) (Anderson et al., 2004). La memoria declarativa de ACT-R se compone de *chunks* (trozos de conocimiento) almacenados con una activación base que refleja su accesibilidad. Cada vez que un chunk es recuperado o utilizado, su activación base aumenta, y luego decae con el tiempo siguiendo una **curva de poder** (base teórica derivada de estudios empíricos). Esta activación base se combina con una activación por contexto para determinar qué chunk se recupera ante una consulta – esencialmente un modelo de *spreading activation* controlado matemáticamente. Si la activación de un chunk cae por debajo de un umbral, será muy improbable que se recupere (efecto de olvido). ACT-R formula esto de manera que puede predecir tiempos de respuesta en tareas de memoria humana, logrando un ajuste fino a datos psicológicos. Por su parte, la memoria procedimental de ACT-R consiste en reglas de producción que disparan acciones cuando ciertas condiciones en la memoria de trabajo se cumplen. ACT-R incluye aprendizaje de producciones (similar a chunking de Soar) y ajuste de los valores de utilidad de las reglas en base a refuerzo/recompensas. Un punto crucial es que ACT-R implementa explícitamente un mecanismo de **olvido adaptativo**: la *actividad base* de un chunk se calcula como una suma de potencias de tiempos, lo que garantiza que los usos frecuentes recientes mantienen alta activación, mientras que los usos antiguos aportan poco. Esto implementa una forma de memoria adaptativa muy alineada con la MMC. De hecho, podríamos decir que la MMC busca “reificar” la memoria declarativa de ACT-R en un grafo explícito: cada chunk sería un nodo, cuya activación es análoga al nivel de activación en ACT-R, y cuyos enlaces con otros chunks propagan activación contextual. **Comparativa:** La MMC se beneficia de muchos principios de ACT-R, en particular su fórmula de activación base (que podría adoptar para el cálculo de importancia de nodosfile-unccncqpe5yrcpgsbp39an) y la idea de que la recencia y frecuencia determinan la probabilidad de recuperación de un recuerdo. La MMC también comparte con ACT-R la noción de *memoria de trabajo limitada* – ACT-R solo activa unos pocos chunks relevantes en cada momento (similar a la idea de la MMC de mantener N nodos más activados “despiertos” y el resto hibernados)file-unccncqpe5yrcpgsbp39an. Una diferencia es que ACT-R no modela relaciones entre chunks mediante pesos variables: la asociación está implicada en las producciones o en valores de similitud, pero no hay un grafo general donde dos chunks puedan tener peso de conexión. La MMC aporta ese grafo, permitiendo múltiples tipos de relaciones ponderadas entre cualquier par de nodos. En resumen, ACT-R ofrece un respaldo teórico poderoso a las decisiones de diseño de la MMC en cuanto a decaimiento de memoria y activación, demostrando que es posible equilibrar *aprendizaje continuo* y *evitación del olvido catastrófico* con reglas adecuadas (Lebiere, 1999). La MMC lleva esos conceptos a un diseño más orientado a conocimiento semántico integrativo.
* **Semantic Pointer Architecture (SPA)** – Es un marco desarrollado por Eliasmith y colaboradores dentro de la arquitectura *SPAUN*, un modelo a gran escala del cerebro (Eliasmith et al., 2012). SPA combina representaciones **vectoriales de alta dimensionalidad** (también llamadas *Vector Symbolic Architecture*) con operaciones algebraicas que permiten *ligar* y *desligar* conceptos. En SPA, cada concepto o pieza de información se representa como un vector denso en $\mathbb{R}^n$ (con $n$ grande, típicamente 512 o 1024 dimensiones), y las relaciones compuestas se representan mediante operaciones como la convolución circular o la suma. Esto permite construir estructuras complejas (por ejemplo, representar una frase combinando vectores de palabras y vectores de roles gramaticales). El aprendizaje en SPA suele implementarse mediante el ajuste de pesos sinápticos en redes neuronales que manipulan estos vectores – por ejemplo, Spaun utiliza ~2.5 millones de neuronas simuladas con aprendizaje *spike-timing dependent plasticity (STDP)* para asociar secuencias. Lo relevante de SPA es que demuestra que es posible una **representación distribuida y combinatoria** del conocimiento, que a la vez puede ejecutar **operaciones simbólicas** (como recuperar un elemento “desempaquetándolo” de un vector resultante de una convolución). *Ventajas:* Es biológicamente inspirada y ha mostrado realizar múltiples tareas cognitivas (memorización de listas, reconocimiento visual, aritmética simple) dentro de un mismo sistema. *Limitaciones:* Las representaciones vectoriales son difíciles de interpretar directamente por humanos, y la incorporación de nuevo conocimiento requiere entrenar pesos neuronales, lo que puede implicar riesgo de olvido catastrófico si no se hace con cuidado (aunque se investiga mitigarlo con métodos de *continual learning*). **Relación con MMC:** La MMC no adopta directamente vectores de alta dimensión como representación primaria de cada nodo (sus nodos son más bien simbólicos con posibles *embeddings* asociados para búsquedas por similitud). Sin embargo, contempla integrar **vectores semánticos (embeddings)** para complementar las relaciones simbólicasfile-4xuxyatu3sqxtrrzbkpyllfile-4xuxyatu3sqxtrrzbkpyll. Por ejemplo, un nodo de texto podría almacenar además un vector obtenido por *word embedding* que permita encontrar similitudes con otros nodos textuales aunque no estén conectados explícitamente. En cierto modo, la MMC busca combinar lo mejor de ambos mundos: la interpretabilidad de un grafo simbólico *etiquetado* con la potencia de generalización de representaciones vectoriales. Conceptualmente, la idea de *superponer* conceptos para generar nuevos (muy presente en SPA) tiene eco en la MMC cuando ésta crea **nodos patrón** que agrupan varios conceptos relacionados bajo una abstracción (ver sección de Resultados). Esos patrones actúan como “vectores resumen” de varias experienciasfile-unccncqpe5yrcpgsbp39an. En conclusión, SPA ofrece un enfoque complementario más cercano a la neurociencia computacional, mientras que la MMC se ubica más del lado de las *representaciones de conocimiento simbólico*; ambas comparten la meta de lograr **composicionalidad** y **escalabilidad** en la memoria de un agente.
* **OpenCog (AtomSpace y Cognitive Synergy)** – OpenCog es un proyecto de **inteligencia general artificial (AGI)** que provee un marco de software para integrar múltiples algoritmos de IA (Goertzel et al., 2010). En su núcleo está el **AtomSpace**, una base de conocimientos en forma de **hipergrafo** rotulado y ponderado. Cada pieza de información (llamada *átomo*) puede ser un nodo o un enlace en el hipergrafo, y tiene asociadas evaluaciones numéricas llamadas *atenciones* o *verdades* (por ejemplo, un valor que indica cuán cierto o relevante es ese átomo). OpenCog implementa mecanismos de **propagación de atención económica**: asigna a los átomos valores de corto plazo (STI) que suben cuando son utilizados recientemente y decaen rápidamente, y valores de largo plazo (LTI) que aumentan con utilidad sostenida y decaen lentamentefile-unccncqpe5yrcpgsbp39an. Esto guía al sistema a enfocarse en las partes importantes del grafo en cada momento, similar a lo que la MMC intenta con activación/hibernación. Además, OpenCog incluye un motor de inferencia lógica probabilística (PLN) que opera sobre el grafo, y otros procesos como módulos de aprendizaje atencional, generación de procedimientos (via MOSES, un algoritmo evolutivo), etc., todos compartiendo la misma memoria AtomSpace. *Relación con MMC:* El AtomSpace de OpenCog es conceptualmente muy cercano a la base de grafo de la MMC: ambos son **grafos de conocimiento con pesos adaptativos**. De hecho, OpenCog ya propone una dualidad entre importancia a corto y largo plazo que la MMC puede aprovecharfile-unccncqpe5yrcpgsbp39an. La MMC difiere en que enfatiza la noción de *fragmentos nativos* (mientras que OpenCog es agnóstico al contenido inicial, esperando que uno lo llene con conocimiento) y en que incluye explícitamente un módulo de cifrado/seguridad, algo no abordado en OpenCog. OpenCog es un sistema más amplio cuyo fin es AGI general, por lo que incluye también toma de decisiones, lenguaje, etc. La MMC se concentra en ser **módulo de memoria**, aunque los principios de OpenCog subrayan la necesidad de pensar la memoria en *sinergia con otros procesos* (como un componente dentro de una arquitectura mayor)file-unccncqpe5yrcpgsbp39an. Un punto destacable es que la MMC puede verse como el componente *asociativo* de una arquitectura cognitiva completa, complementario a un componente lógico o deliberativo. Goertzel (2014) sugiere que en una *cognitive synergy*, los diferentes procesos (razonamiento lógico, memoria asociativa, atención, metacognición) deben cooperar. La MMC aporta la parte asociativa-dinámica: su evolución debe calibrarse con mecanismos de razonamiento simbólico (por ejemplo, un motor de inferencia podría revisar y consolidar algunos de los enlaces que la MMC forma espontáneamente). En conclusión, OpenCog proporciona validación de que un **grafo cognitivo** con ponderaciones puede sustentar un sistema inteligente, y ofrece conceptos (como STI/LTI, atención económica) que apoyan las decisiones de diseño de la MMC. A la vez, la MMC se enfoca más en **auto-organización interna** (clustering evolutivo, fragmentos nativos, autoprotección), buscando una memoria que se mantenga por sí misma coherente y segura a largo plazo, lista para integrarse en cualquier agente cognitivo.
* **Memorias Diferenciables y Redes Neuronales con Memoria** – En años recientes, desde el aprendizaje profundo se han propuesto arquitecturas que dotan a redes neuronales de un módulo de memoria explícito al cual pueden leer/escribir mediante mecanismos de atención. Ejemplos prominentes son las **Memory Networks** (Weston et al., 2015) y el **Neural Turing Machine / Differentiable Neural Computer** (Graves et al., 2016). Las *Memory Networks* de Weston et al. consisten en un conjunto de vectores que almacenan hechos (memoria), junto con un mecanismo de consulta que usa la atención para recuperar combinaciones relevantes de esos vectores en varias *pasadas* y producir una respuesta. Este enfoque ha logrado buenos resultados en tareas de respuesta a preguntas (QA), demostrando la utilidad de tener una memoria de trabajo que puede atender a distintas partes de un contexto. Por su parte, la *Differentiable Neural Computer (DNC)* de DeepMind ofrece una matriz de memoria donde una red “controladora” puede escribir y leer mediante operaciones de atención basadas en similitud, entrenando todo de extremo a extremo con gradiente. Estas memorias *neuronales* son altamente flexibles y han resuelto laberintos, inferencias en grafos, etc. *Comparativa:* Si bien son poderosas, suelen requerir entrenamiento intensivo y no ofrecen interpretabilidad directa: el contenido de su memoria son vectores sin significado inmediato para humanos. Además, pueden tener dificultades para **aprender continuamente** más allá de lo visto en entrenamiento sin olvidar (aunque hay trabajos de *continual learning* tratando de solventarlo). **MMC vs. Memorias diferenciables:** La MMC, al ser un grafo explícito, es más interpretable – cada nodo puede inspeccionarse – y naturalmente admite la incorporación incremental de nuevos nodos sin requerir reentrenar todo el modelo (basta con añadir el nodo y conectar). En cambio, en una Memory Network neural típica, el espacio de memoria es fijo de antemano (ej. 100 “slots” vectoriales) y durante entrenamiento aprende a usar esos slots; agregar nueva información en forma de nuevos slots no entrenados podría ser problemático. No obstante, la MMC puede beneficiarse de técnicas de estas arquitecturas: por ejemplo, usar **mecanismos de atención** para leer dentro de su grafo. De hecho, la recuperación por *rutas principales* de MMC es conceptualmente similar a atender a las conexiones de mayor peso (más relevantes), y las *rutas laterales creativas* son análogas a explorar contenidos menos activados (lo que una red atencional haría con menor probabilidad pero podría, si aportan). Además, la MMC podría integrarse con redes neuronales tradicionales usando su grafo como una memoria externa: por ejemplo, un modelo de lenguaje podría consultar a la MMC cuando requiera conocimiento específico, y viceversa. En conclusión, las memorias diferenciables muestran una vía para dotar a sistemas de aprendizaje automático de memoria; la MMC en cambio busca dotar a un sistema de memoria de capacidades de aprendizaje automático. Son enfoques complementarios que eventualmente podrían unirse (un controlador neural operando sobre una MMC grafo).

En el panorama general, se aprecia que **ningún enfoque existente cubre completamente todas las facetas** que la MMC pretende abarcar, pero cada uno aporta ideas valiosas. La MMC se posiciona en la intersección de varios de estos paradigmas: es **simbólica** (como Soar/ACT-R) en su representación básica, pero también **sub-simbólica** (como SPA o MemNets) al incorporar embeddings continuos; tiene un **componente de aprendizaje autónomo** (reforzamiento Hebbiano, descubrimiento de patrones) sin dejar de lado la posibilidad de integrarse con **módulos lógicos deliberativos** (como OpenCog); y equilibra **estabilidad y cambio** mediante estrategias de olvido inspirado en la cognición humana (alineado con ACT-R, OpenCog) y de exploración creativa (como modelos de creatividad computacional). En suma, la MMC busca sintetizar *lecciones de múltiples arquitecturas* en un diseño unificado de memoria artificial.

## Objetivos e Hipótesis

### Objetivos generales

El objetivo general de esta investigación es **proponer, desarrollar y evaluar una arquitectura de memoria artificial innovadora – la Memoria de Mapa Conceptual (MMC)** – que modele el conocimiento como una red dinámica de nodos conceptuales interconectados con pesos adaptativos. Esto implica diseñar la MMC en detalle (a nivel teórico y de software), contrastarla con las teorías y sistemas existentes, e identificar cómo su implementación puede mejorar la gestión del conocimiento en sistemas de IA. Se busca demostrar que la MMC puede proporcionar a un agente inteligente una memoria **más auto-organizada, resiliente, comprensible y segura** que las soluciones actuales.

### Objetivos específicos

Para lograr el objetivo general, se plantean los siguientes objetivos específicos:

1. **Analizar el marco teórico y el estado del arte** en representación del conocimiento y memorias artificiales adaptativas, incluyendo mapas conceptuales, redes semánticas, arquitecturas cognitivas (Soar, ACT-R, etc.), grafos de conocimiento y técnicas de aprendizaje continuo, a fin de extraer principios clave para el diseño de la MMC.
2. **Diseñar la arquitectura MMC** especificando sus componentes fundamentales – tipos de nodos (fragmentos nativos, ordinarios, especializados), conexiones ponderadas, métricas de activación – y sus procesos dinámicos – mecanismos de aprendizaje (refuerzo hebbiano de pesos, incorporación de nuevos nodos), de olvido (decaimiento y hibernación), de recuperación de información (búsqueda por rutas principales y laterales) y de reorganización (clusterización adaptativa). Este diseño incluirá la definición de **fragmentos nativos** que doten al sistema de identidad y pautas éticas desde su inicio.
3. **Desarrollar propuestas de implementación software** para la MMC, seleccionando tecnologías adecuadas: por ejemplo, uso de bases de datos gráficas (como Neo4j) para almacenar la red de nodos y enlaces, uso de espacios vectoriales (FAISS, embeddings) para soporte de búsquedas semánticas, y construcción de un motor de memoria en Python o JavaScript que gestione la lógica de la MMC (activaciones, actualizaciones, consultas). Se espera producir prototipos o módulos de código que ilustran partes funcionales de la MMC.
4. **Integrar un sistema de cifrado de datos como mecanismo de autoprotección** de la MMC, aprovechando un algoritmo de cifrado desarrollado por el usuario en Java (cifrado por permutación de bits). Se busca incorporar este cifrado para proteger los contenidos sensibles de la memoria (en particular los fragmentos nativos éticos y la configuración crítica del grafo) y garantizar la integridad de los datos almacenados. Esto implica adaptar el algoritmo a un entorno de base de datos gráfica y combinarlo con técnicas estándar de seguridad (hashing, cifrado robusto AES) para lograr un equilibrio entre rendimiento y protección.
5. **Implementar y validar un modelo simulado de la MMC** a pequeña escala que permita comprobar empíricamente sus funcionalidades clave. Este modelo servirá como *prueba de concepto* e incluirá características como: incorporación automática de nuevos nodos a partir de fuentes externas simuladas, ajuste dinámico de pesos de enlace, detección de patrones recurrentes en los datos ingresados, hibernación de nodos inactivos y rutas creativas de navegación. Mediante el simulador se observará el comportamiento emergente de la MMC y se recopilarán ejemplos de su uso (p. ej., crecimiento de la red, ejemplos de asociaciones formadas, desempeño de búsqueda, etc.).
6. **Comparar y evaluar los resultados** obtenidos con la MMC frente a los objetivos iniciales y frente a las capacidades de otras arquitecturas. Esto incluye verificar en qué medida la MMC logra sus metas de autoorganización (¿se forman clusters de manera significativa?), de resistencia al olvido (¿se mantienen disponibles conocimientos antiguos poco frecuentes?), de incorporación flexible (¿puede aprender nueva información sin reprocesar todo el grafo?), y de protección (¿los datos sensibles permanecen seguros ante modificaciones directas?). Asimismo, se discutirán las *ventajas competitivas* de la MMC (por ejemplo, su transparencia o su alineación ética incorporada) y sus *limitaciones* o aspectos a refinar (como escalabilidad computacional, necesidad de calibración de parámetros, etc.).

## Hipótesis de investigación

A la luz de los objetivos, se plantea la siguiente **hipótesis principal**:

* **H1:** *Una arquitectura de memoria artificial basada en un mapa conceptual dinámico (MMC) permitirá al sistema aprender e incorporar nuevos conocimientos de forma continua****sin olvidar catastróficamente****la información previamente almacenada, gracias a sus mecanismos de refuerzo de conexiones relevantes y hibernación de conocimientos poco usados.* En otras palabras, la MMC mantendrá un equilibrio entre plasticidad (aprendizaje de lo nuevo) y estabilidad (retención de lo antiguo) superior al de memorias tradicionales estáticas o redes neuronales sin soporte adicional.

De esta hipótesis principal se derivan sub-hipótesis específicas:

* **H1a:** *La presencia de****fragmentos nativos****(p. ej. un núcleo ético inmutable) en la MMC contribuirá a que el sistema conserve coherencia y principios fundamentales durante su aprendizaje, evitando derivas indeseadas en su comportamiento.* Es decir, esperamos observar que, incluso tras numerosas incorporaciones de conocimiento y reconfiguraciones de la red, los nodos nativos permanecen centrales y activos, influyendo como criterio estable en las decisiones de la memoria.
* **H1b:** *La****autoprotección mediante cifrado****integrada en la MMC resguardará efectivamente la integridad de los datos sensibles, de modo que cualquier intento de manipulación externa o acceso no autorizado a los datos almacenados sea detectable o impedido.* Esto implica que, en pruebas controladas, si se altera manualmente la base de datos de la MMC en zonas protegidas, el sistema podrá detectar la incongruencia (mediante verificación de integridad HMAC) y restaurar o bloquear dichos cambios.
* **H1c:** *La estrategia dual de****recuperación de información****de la MMC (rutas principales vs. rutas creativas) mejora la capacidad del sistema tanto para responder consultas con exactitud usando conocimiento consolidado, como para generar ideas nuevas o soluciones innovadoras combinando fragmentos antes no relacionados.* Se espera que el simulador MMC muestre ejemplos donde una consulta dada por el usuario obtiene respuestas correctas basadas en las conexiones más fuertes, pero también sugerencias o hallazgos adicionales basados en la exploración lateral del grafo, algo que arquitecturas con búsqueda estrictamente guiada por peso no producirían.

Aunque estas hipótesis no se validarán de manera estadística tradicional (dado el carácter principalmente cualitativo y de ingeniería de la tesis), orientarán la construcción y evaluación del modelo. Se considerará la hipótesis apoyada si la evidencia del desarrollo teórico y las pruebas del prototipo muestran que la MMC efectivamente exhibe los comportamientos previstos (aprendizaje continuo sin olvido significativo, retención de anclajes nativos, protección exitosa, equilibrio entre respuestas precisas y creativas).

## Método

### Enfoque general y diseño de la investigación

El presente trabajo adoptó un enfoque de **investigación aplicada y de diseño** (*Design Science Research*), enfocado en la creación y análisis de un artefacto tecnológico innovador (la MMC) a partir de fundamentos teóricos. Más que un experimento controlado clásico, se trata de un proceso iterativo de *concepción, construcción, y evaluación* de un modelo funcional de memoria artificial. Los pasos metodológicos principales fueron los siguientes:

1. **Revisión bibliográfica y teórica:** Se inició con un estudio exhaustivo de la literatura relevante en varias áreas: representación del conocimiento (mapas conceptuales, ontologías, grafos semánticos), arquitecturas cognitivas (Soar, ACT-R, OpenCog, etc.), sistemas de memoria en aprendizaje automático (memorias neuronales, continual learning), teoría del olvido y memoria humana, y ética en IA. Esta revisión (reflejada parcialmente en la sección de Antecedentes) permitió extraer requisitos deseables para la MMC y principios de diseño informados por trabajos previos. Por ejemplo, se identificó la necesidad de incorporar un mecanismo de decaimiento de activación inspirado en ACT-R, y la conveniencia de un esquema de atención dual corto/largo plazo como en OpenCog. También se recopiló documentación provista por el usuario, incluyendo los archivos *“MMC.docx”* y *“MMC modelo inspiracional.docx”*, que contenían reflexiones iniciales y modelos simulados, sirviendo de base inspiradora.
2. **Especificación conceptual de la MMC:** Con la base teórica, se procedió a diseñar detalladamente la arquitectura de la MMC. Esto implicó definir sus **entidades** (clases de nodos y sus atributos, tipos de enlaces y sus propiedades) y **procesos operativos** (algoritmos para actualización de pesos, propagación de activación, inserción de nuevos nodos, etc.). Se elaboró un esquema general de la arquitectura en forma de módulos (ver Figura 1), identificando componentes como: un **módulo de ingesta de información** (que procesa entradas externas y crea nodos), una **base de datos grafo** para las relaciones explícitas, un **almacén vectorial** para las representaciones semánticas, el **motor central MMC/FCI** que implementa la lógica de memoria y fragmentos del Castillo Interior (núcleo ético-espiritual), y una interfaz de consulta (API conversacional). También se definieron parámetros clave, por ejemplo: rangos de pesos, constantes de decaimiento $\lambda$, umbrales de hibernación, etc., inicializados en función de referencias de la literatura (p. ej., valores de decaimiento bajos como 0.001–0.01 para que el olvido sea gradual, coherente con modelos psicológicos). Durante esta fase se tomaron decisiones de diseño informadas por el material del usuario – por ejemplo, se decidió implementar los **7 estados de los Fragmentos del Castillo Interior (FCI)** inspirados en un documento base, refinándolos con métricas y triggers tal como sugería *“MMC.docx”*file-4xuxyatu3sqxtrrzbkpyllfile-4xuxyatu3sqxtrrzbkpyll. Del mismo modo, se adoptó la taxonomía de fragmentos (FN, FC, FM, FS, etc.) propuesta en esos documentos, ampliándola con fragmentos sociales, metacognitivos, de seguridad, etc., marcados como 🆕 en el diseño originalfile-4xuxyatu3sqxtrrzbkpyllfile-4xuxyatu3sqxtrrzbkpyll. Esta etapa concluyó con un **diseño teórico completo de la MMC**, equivalente a una especificación formal.
3. **Diseño de la solución de software e integración tecnológica:** A partir de la especificación, se planificó cómo implementar la MMC en un entorno computacional concreto. Para el almacenamiento de la red de nodos y enlaces se consideró el uso de una **base de datos de grafos** (como Neo4j, orientada a grafos con capacidad de consultas eficientes por relaciones)file-4xuxyatu3sqxtrrzbkpyll. Igualmente, para habilitar búsquedas semánticas por similitud de contenido se contempló un **vector store** usando bibliotecas como FAISS o Annoy, en el cual almacenar embeddings de nodos (especialmente para nodos de texto, imagen, etc.)file-4xuxyatu3sqxtrrzbkpyll. El motor lógico de la MMC se planificó desarrollar en **Python**, aprovechando librerías de **redes neuronales y grafos** (por ejemplo, PyTorch Geometric para experimentar con grafos neuronales o NetworkX para prototipos, y eventualmente pyDatalog o Prolog para reglas simbólicas de alto nivel relacionadas con los FCI)file-4xuxyatu3sqxtrrzbkpyll. Se estructuró la solución siguiendo una arquitectura modular de software: *Ingest Hub* (módulo de ingesta), *GraphDB Interface*, *Vector Store Interface*, *Memory Core Engine*, *API/Frontend*. La Figura 1 ilustra esquemáticamente esta arquitectura de implementación propuesta.

*Figura 1. Arquitectura simplificada de la MMC propuesta.* El *Ingest Hub* ingiere nuevas piezas de información (desde archivos, web o APIs) y genera nodos en las bases de datos. La información simbólica se almacena en una **base de datos grafo** (p. ej., Neo4j) que guarda las relaciones explícitas entre nodos; paralelamente, las representaciones vectoriales (embeddings) de los contenidos se indexan en un **Vector Store** para facilitar búsquedas por similitud. El **Motor MMC** central orquesta la lógica de la memoria: actualización de pesos, activación/hibernación de nodos, evaluación de estados FCI, etc., interactuando con ambas bases de datos. Finalmente, un **API Conversacional** u otro cliente puede consultar a la MMC, desencadenando procesos de recuperación de información a través del motor. Este diseño es escalable y separa preocupaciones: la persistencia (grafo y vectores) de la lógica de negocio (motor de memoria).

En cuanto al **prototipado**, se decidió implementar un modelo simplificado de la MMC en un entorno **JavaScript/TypeScript** para su ejecución en un navegador web, con el fin de facilitar la visualización interactiva. Esto se basó en el archivo *“yui.html”* proporcionado en *“MMC modelo inspiracional.docx”*, que describía una simulación inicial. Se planificó extender esa simulación incluyendo las funcionalidades deseadas (exploración web automática, detección de patrones, etc.) usando JS para la lógica y posiblemente librerías como D3.js o Three.js para visualizar el grafo en 2D/3D.

1. **Implementación del prototipo simulado:** Siguiendo el plan, se construyó un **modelo simulado de la MMC** ejecutándose en un navegador web, utilizando HTML5/JavaScript. En este simulador, la MMC inicia con un conjunto predefinido de nodos nativos (ética, identidad, aprendizaje, código, matemáticas, sensorial, etc.) fuertemente interconectados según el diseño teóricofile-unccncqpe5yrcpgsbp39anfile-unccncqpe5yrcpgsbp39an. Sobre esta base, se implementaron varias características:
   * **Exploración autónoma de fuentes externas:** el simulador cuenta con botones e intervalos configurables para que la MMC simule navegar páginas web o consultar APIs automáticamente, creando nuevos nodos por cada recurso externo encontradofile-unccncqpe5yrcpgsbp39anfile-unccncqpe5yrcpgsbp39an. Por ejemplo, al activarse, cada ciclo de exploración agrega un nodo de tipo “web” con contenido ficticio de una URL, o un nodo “api” con datos simulados, y los conecta aleatoriamente con algunos nodos existentes para imitar la integración de conocimientofile-unccncqpe5yrcpgsbp39an. Se incorporó un control deslizante que permite ajustar la proporción *exploración vs. explotación* (p. ej., 70% del tiempo explorar nuevas fuentes, 30% reforzar conocimiento existente), ilustrando el dilema exploración-explotación de forma explícitafile-unccncqpe5yrcpgsbp39an. Esto se logró con un simple algoritmo probabilístico: en cada ciclo, según el porcentaje elegido, decidir ya sea añadir un nuevo nodo (exploración) o activar/revisitar nodos existentes (explotación).
   * **Activación y decaimiento de nodos:** cada nodo en la simulación tiene un atributo numérico de *nivel de activación*. Se implementó un bucle temporal donde en cada iteración se reduce ligeramente la activación de todos los nodos (multiplicando por $(1 - \delta)$, con $\delta$ como tasa de decaimiento configurable)file-unccncqpe5yrcpgsbp39an. Así, si un nodo no es usado, su activación irá bajando lentamente. Cuando un nodo es accedido (por ejemplo, al hacer clic sobre él o al ser relacionado con un nuevo nodo), su activación aumenta. De este modo, el simulador reproduce el comportamiento de memoria a corto plazo de la MMC: los nodos “frescos” se mantienen activos.
   * **Hibernación de nodos inactivos:** se definió un umbral mínimo (por ejemplo, 0.1) por debajo del cual si la activación de un nodo cae, el nodo se marca como *hibernado*file-unccncqpe5yrcpgsbp39anfile-unccncqpe5yrcpgsbp39an. En la simulación, esto implica cambiar su estado visual (opacidad reducida) y excluirlo de las búsquedas principales. Internamente, un contador de “nodos activos” disminuye. El nodo hibernado no desaparece; sus conexiones permanecen almacenadas. Se implementó también una regla de *reactivación por contexto*: cada vez que un nuevo nodo es añadido o se realiza una búsqueda, si hay nodos hibernados que comparten etiquetas o tienen embeddings similares al tema en curso, se incrementa temporalmente su activación para ver si deben “despertar”file-unccncqpe5yrcpgsbp39anfile-unccncqpe5yrcpgsbp39an. Por ejemplo, si existía un nodo hibernado sobre “COVID-19” y de pronto el usuario ingresa muchos datos médicos relacionados, el sistema detectaría la similitud semántica y reactivaría ese nodo hibernado. Este mecanismo se simplificó en el simulador por eficiencia (dada la escala pequeña, bastó con verificar coincidencia de palabras clave en títulos).
   * **Detección de patrones y creación de nodos abstracción:** inspirándose en técnicas de *data mining* (descubrimiento de reglas frecuentes), se incorporó un módulo que, cada cierto número de operaciones, revisa las conexiones del grafo en busca de posibles patrones repetitivos. En la simulación, se simplificó generando aleatoriamente “patrones” a partir de grupos de nodos existentes: el sistema crea un nuevo nodo de tipo **patrón** y lo conecta con varios nodos que considere relacionadosfile-unccncqpe5yrcpgsbp39anfile-unccncqpe5yrcpgsbp39an. Por ejemplo, si hay varios nodos de tipo “código” que siempre aparecen vinculados a nodos “matemáticos”, el simulador podría crear un nodo patrón llamado “Patrón de código-matemáticas” y conectarlo a esos nodos, asignándole además un valor de *soporte* (frecuencia) y *confianza* (fuerza de implicación) simuladosfile-unccncqpe5yrcpgsbp39anfile-unccncqpe5yrcpgsbp39an. Esto emula que la MMC ha identificado una asociación general: “cada vez que ocurre X, suele ocurrir Y”. Estos nodos patrón sirven como **hiper-nodos** que resumen conocimiento disperso, muy parecido al concepto de *esquema* en psicología cognitivafile-unccncqpe5yrcpgsbp39an. En la implementación, cada nuevo patrón se registraba en un log de eventos (p. ej., “Patrón descubierto: …”) para fines de trazabilidad.
   * **Rutas creativas de recuperación:** además del método de búsqueda directa (encontrar el camino más corto o el de enlaces más fuertes entre nodos consultados), se implementó un botón de “búsqueda creativa” que realiza un *random walk with restart* en el grafo – es decir, una caminata aleatoria que ocasionalmente vuelve al punto de partidafile-4xuxyatu3sqxtrrzbkpyll. El resultado de esa caminata es una secuencia de nodos que puede representar una asociación inusual. El simulador muestra esa ruta como una posible “idea” o sugerencia alternativa. Si el usuario la valida usando esos nodos juntos, entonces el sistema refuerza los enlaces de esa ruta (haciendo que deje de ser tan aleatoria y pase a ser parte del conocimiento consolidado)file-unccncqpe5yrcpgsbp39an. Esto refleja la noción de que la creatividad de la MMC produce hipótesis (conexiones tentativas) que deben luego confirmarse por co-activación real para asentarse (aprendizaje hebbiano emergente)file-unccncqpe5yrcpgsbp39an.
   * **Interfaz y visualización:** se desarrolló una interfaz gráfica para monitorear la MMC simulada. Se utilizó un layout de grafo (fuerza dirigida) para disponer los nodos en pantalla, con colores o formas distintas según su tipo (p. ej., nodos nativos en un color, nodos web en otro). Barras laterales mostraban estadísticas en tiempo real: número de nodos activos vs. hibernados, conteo de webs exploradas, APIs consultadas, patrones descubiertos, etc. (actualizándose conforme ocurrían eventos)file-unccncqpe5yrcpgsbp39an. Controles permitían al usuario iniciar/detener la exploración automática, modificar la tasa de decaimiento de activación, ajustar el porcentaje exploración/explotación, e introducir consultas manuales (por nombre de nodo o palabra clave). Toda interacción del usuario (como agregar un nodo manualmente o despertar un nodo hibernado manualmente) también se registró para análisis.
2. **Integración del cifrador Java en la MMC:** Paralelamente al desarrollo del simulador central, se abordó la integración del algoritmo de cifrado proporcionado por el usuario. Primero, se analizó el funcionamiento de este cifrador a partir de los archivos Java (*Cofre.java*, *Encriptador.java*, etc.) y del documento *“Encriptación MMC.docx”*. Se comprendió que el cifrado operaba mediante una **permutación de bits** guiada por una matriz secreta de 16x8 (128 posiciones) que el usuario configura manualmente (el “tablero” que sirve de clave privada)file-ybwhqv9ju7i9nfanvrmq5h. A partir de una frase clave, el algoritmo genera una segunda matriz (clave pública, *codCP*) aplicando permutaciones fijasfile-ybwhqv9ju7i9nfanvrmq5h. Luego, para cifrar un texto, se divide en bloques de 16 caracteres, se convierten a ASCII binario de 128 bits, y mediante la matriz *cod* (clave privada) se permutan esos bits para producir un texto cifrado (con sustitución de caracteres no imprimibles por tokens especiales)file-ybwhqv9ju7i9nfanvrmq5h. Se confirmó que el esquema es determinístico y reversible dado conocer la matriz clave, pero no incorpora sustituciones criptográficas no lineales ni mezclas complejas, por lo que su seguridad es limitada en comparación con estándares como AES (como señalaba el documento, es un método ligero útil más bien para ofuscación o integridad, pero no para proteger de ataques sofisticados)file-ybwhqv9ju7i9nfanvrmq5h. Con este entendimiento, se procedió a portar el algoritmo al entorno JavaScript para integrarlo en el simulador MMC. Se implementó un módulo mmcCrypto en TypeScript que reproduce las funciones esenciales:
   * buildSecretMatrix(order) – para generar la matriz de permutación secreta a partir del orden dado de 1 a 128 (la GUI del tablero).
   * encrypt(text) – que toma un texto, aplica el remapeo de bits con la matriz secreta y retorna el texto cifrado.
   * decrypt(text) – que invierte el proceso usando la matriz (requiere la matriz secreta).
   * Además, funciones para manejar el *cofre* (clave pública): exportPublicKey(phrase) y importPublicKey(phrase, data) para derivar la matriz pública de una frase y viceversafile-ybwhqv9ju7i9nfanvrmq5h.

Una vez disponible este módulo criptográfico en JS, se integró en el flujo de la MMC simulada de la siguiente manera: cada vez que el sistema iba a almacenar un nuevo nodo en la base de datos interna (IndexedDB en el navegador para persistir datos entre sesiones), primero pasaba el contenido del nodo por mmcCrypto.encrypt(), obteniendo una cadena cifrada, y en paralelo generaba un **HMAC SHA-256** de esa cadena cifrada para asegurar integridadfile-ybwhqv9ju7i9nfanvrmq5hfile-ybwhqv9ju7i9nfanvrmq5h. Esto corresponde al **Paso 2a y 2b** del esquema conceptual de protección (Figura 2). Tanto el texto cifrado como el HMAC se guardan en la base; así, al leer, el sistema recalcula el HMAC y verifica que coincida para detectar manipulacionesfile-ybwhqv9ju7i9nfanvrmq5hfile-ybwhqv9ju7i9nfanvrmq5h. En la práctica del simulador, se optó por la “opción ligera” sugerida: usar el cifrado personalizado del usuario + HMAC para integridadfile-ybwhqv9ju7i9nfanvrmq5h, dado que en un entorno web era más simple que integrar AES en modo robusto (aunque se dejó indicado que en un entorno real se podría aplicar AES-GCM con una clave derivada de la matriz, para aumentar la seguridad sin perder compatibilidad del esquema)file-ybwhqv9ju7i9nfanvrmq5h. Se protegieron con este mecanismo los nodos marcados como sensibles: concretamente, los **fragmentos del Castillo Interior (FCI)**, que incluyen la ética y otros principios, y también los datos relativos a credenciales API simuladas. En el simulador, esto significó que dichos nodos no mostraban su contenido en claro en la interfaz (solo se veía un indicativo de “contenido cifrado”), y cualquier intento de modificarlos desde la consola del navegador resultaba en un HMAC inválido, que el sistema detectaba y entonces ignoraba la modificación (efectivamente impidiendo alteraciones no autorizadas)file-ybwhqv9ju7i9nfanvrmq5hfile-ybwhqv9ju7i9nfanvrmq5h.

*Figura 2. Esquema de protección de datos integrado en la MMC.* Cuando el **usuario** o proceso cliente crea o lee un nodo (1), la petición pasa a la *Capa Lógica MMC*. Al guardar (2a), el contenido del nodo se cifra rápidamente usando el algoritmo de permutación de bits (implementado en JS/TS) obteniendo texto cifrado; en paralelo (2b) se calcula un **hash HMAC SHA-256** del texto cifrado. Ambos – el texto cifrado y la firma HMAC – se almacenan juntos en la base de datos (3). Al leer un nodo, el sistema primero verifica que el HMAC del contenido cifrado almacenado coincida con el HMAC guardado; si difiere, implica que el dato fue modificado externamente y por tanto se lanza una alerta o se rechaza (integridad violada). Si coincide, entonces descifra el contenido y lo entrega al motor de la MMC. De esta forma, la MMC protege la **confidencialidad** (moderada, por un cifrado ligero propio) y sobre todo la **integridad** de sus datos. La incorporación de este esquema en el simulador demostró cómo la MMC podría auto-preservar ciertos recuerdos “privados” (como su núcleo ético) incluso en entornos colaborativos o expuestos.

1. **Evaluación cualitativa del modelo**: tras integrar todas las piezas, se procedió a ejecutar y probar el simulador MMC con diversos escenarios, registrando sus comportamientos y resultados. Se llevaron a cabo pruebas de *stress* incrementando el número de nodos creados (hasta ~100 nodos) para observar la estabilidad de la visualización y el rendimiento de los algoritmos de actualización. Se simularon distintos perfiles de uso: uno *exploratorio* (alta proporción de exploración, creando muchos nodos nuevos rápidamente) y otro *explotador* (centrado en reactivar y usar nodos existentes). Durante estas pruebas se observaron fenómenos alineados con las expectativas:
   * La red se auto-organizó formando **clusters**: por ejemplo, en una corrida con mezcla de datos de distintos temas, los nodos tendieron a agruparse (visualmente y en conexiones) alrededor de etiquetas temáticas (un cluster de “tecnología” separado de uno de “salud”, etc.), confirmando la capacidad de segmentación adaptativa.
   * El mecanismo de **olvido** actuó gradualmente: nodos introducidos al inicio y luego poco relacionados vieron disminuir su activación y muchos entraron en hibernación tras cierto tiempo, liberando la vista de la red (en la visualización se notaba que desaparecían de foco). Sin embargo, cuando se introducía información nueva relacionada a un nodo hibernado, se constató su reactivación (por ejemplo, un nodo “COVID-19” hibernado revivió al añadirse otro nodo “Vacuna” conectado, volviendo a estar activo).
   * La función de **patrones** produjo nodos resumen útiles: en las pruebas, los patrones generados aleatoriamente a veces carecían de significado evidente (dado que era aleatorio en la simulación), pero en otros casos coincidieron con lo esperable – e.g., detectó un “Patrón de texto” cuando varios nodos de tipo texto estaban vinculados a los mismos conceptos. Estos patrones sirvieron como *hints* de posibles relaciones globales en la red.
   * En la **búsqueda de información**, se verificó que la MMC devolviera respuestas acertadas usando rutas principales: por ejemplo, ante la pregunta simulada “¿qué sabes de X?” si X estaba en la memoria, el sistema navegó por sus enlaces fuertes y listó los nodos más directamente relacionados, que típicamente eran los relevantes. Adicionalmente, al activar la opción creativa, a veces sugería una conexión inesperada – por ejemplo, conectar X con Z a través de Y, donde Y era un concepto remoto. Hubo casos donde estas conexiones resultaron ser ruido, pero otros donde revelaron una relación no obvia (demostrando el potencial de descubrimiento).
   * Respecto a la **seguridad**, se intentó forzar modificaciones indebidas: se abrió la consola del navegador y se alteró deliberadamente el almacenamiento local de un nodo marcado como protegido (cambiando su contenido cifrado). Al volver a cargar la interfaz, el sistema arrojó un error de integridad para ese nodo y se negó a descifrarlo, manteniendo oculto su valor – comportamiento que confirma la eficacia del HMAC para detectar manipulación. Si bien esto es una simulación local, conceptualmente avala la hipótesis de autoprotección.
2. **Análisis de resultados y ajustes:** con las observaciones de la evaluación, se refinó el modelo en aspectos menores (p. ej., calibrar el umbral de hibernación para que no “duerma” nodos demasiado pronto en la demo, ajustar la frecuencia de descubrimiento de patrones para que no sature la vista con demasiados nodos patrón irrelevantes, etc.). Se compararon los comportamientos del simulador con las expectativas teóricas y con el funcionamiento de otras arquitecturas. En general, se constató que la MMC simulado exhibió las propiedades deseadas, aunque naturalmente en la escala y simplificación de una demo. Este ejercicio permitió **validar cualitativamente las hipótesis**: no se observó olvido total de conocimientos previos (siempre quedaban latentes), los fragmentos nativos permanecieron presentes e influyentes (el fragmento de ética, aunque simbólico, seguía conectado a muchos nodos como referencia), y el cifrado evitó alteraciones directas. También se identificaron **limitaciones**: por ejemplo, a medida que crece mucho el número de nodos, la visualización y ciertos algoritmos (como el random walk) se vuelven lentos – sugiriendo la necesidad de optimizaciones o límites operativos. Estas consideraciones alimentan la discusión final.

En resumen, la metodología combinó construcción teórica y experimentación práctica en un ciclo iterativo. No se siguió una metodología experimental tradicional con grupo control y medición cuantitativa, ya que el objetivo fue explorar la *viabilidad de diseño* de la MMC más que medir un efecto numérico específico. Aun así, se aplicó rigor en la validación cruzada: cada funcionalidad implementada se basó en fundamentos teóricos (p. ej., se referenció la literatura para justificar parámetros) y se probó en el prototipo para comprobar su utilidad. Esto proporciona una base sólida para argumentar, en la siguiente sección, los **resultados** logrados con la arquitectura MMC y cómo se comparan con los objetivos planteados.

## Resultados

A continuación, se presentan los hallazgos y resultados obtenidos del desarrollo teórico-práctico de la MMC. Estos resultados incluyen tanto la descripción de la **estructura final propuesta** de la MMC y sus componentes, como las observaciones derivadas del **modelo simulado** construido. Se organizan en torno a los principales ejes funcionales de la MMC: la estructura de nodos y fragmentos, los procesos dinámicos de aprendizaje y olvido, los mecanismos de recuperación de información, y la integración de la capa de seguridad.

## Estructura y componentes de la MMC

La **Memoria de Mapa Conceptual** resultante se compone de un conjunto de elementos interrelacionados que en conjunto conforman un sistema de memoria artificial autorregulado. *¿Qué elementos conforman la MMC?* En esencia, podemos distinguir:

* **Nodos (Conceptos o Fragmentos de conocimiento):** Cada porción discreta de conocimiento se representa como un nodo en el grafo de la MMC. Los nodos pueden provenir de distintas fuentes y tienen tipos asociados, por ejemplo: texto (un hecho o idea textual), imagen (una referencia a una imagen), audio, código (un fragmento de código fuente), sensor (un dato proveniente de sensores), entre otros. Cada nodo posee un identificador único y puede almacenar **metadatos** como la fecha de creación, la fuente de donde se obtuvo, etiquetas temáticas, nivel de confianza, etc. Adicionalmente, a cada nodo se le asocia, cuando aplica, una representación vectorial (embedding) precalculada de su contenido para fines de búsqueda por similitud semántica. Por ejemplo, un nodo de texto puede tener un embedding de dimensión 300 generado por Word2Vec o BERT, almacenado junto a él. Uno de los metadatos más importantes es el **nivel de activación actual** del nodo, un valor numérico que refleja su relevancia en un momento dado (este valor es dinámico, variando según uso y tiempo). En el diseño final, se incluyó también un **hash de integridad** (p. ej., SHA-256) de los datos del nodo, para detectar cualquier alteración no autorizada, como parte de la capa de seguridad (detalles más adelante).
* **Conexiones (Enlaces ponderados entre nodos):** Los nodos no existen aislados, sino que están conectados por **enlaces dirigidos** (o potencialmente no dirigidos, según la relación) que representan relaciones o asociaciones entre conceptos. Cada conexión se modela típicamente como una tupla (origen, destino, tipo, peso). El *tipo* o etiqueta del enlace indica la naturaleza de la relación (por ejemplo: “es\_un”, “parte\_de”, “menciona”, “causa”, “referencia”, etc., o simplemente relaciones genéricas de asociación semántica). El *peso* es un valor numérico que expresa la **fuerza de la asociación** entre esos dos nodos: cuanto mayor el peso, más fuertemente relacionados se consideran. El peso se interpreta también inversamente como un “costo” a la hora de propagación: enlaces de peso alto son caminos preferidos (bajo costo). Estos pesos son **dinámicos** y se actualizan automáticamente conforme la MMC aprende: se implementó la regla hebbiana simplificada $\Delta w = \alpha \cdot \text{act}\_i \cdot \text{act}*j$ cuando dos nodos $i,j$ se activan juntos, aumentando $w*{ij}$ proporcionalmente a sus activaciones (con factor de aprendizaje $\alpha$ pequeño)file-4xuxyatu3sqxtrrzbkpyll. Asimismo, se introdujo un mecanismo de **decaimiento** de pesos: si una conexión no se utiliza en mucho tiempo, su peso se va reduciendo multiplicándolo periódicamente por un factor $e^{-\lambda \Delta t}$file-4xuxyatu3sqxtrrzbkpyll. Esto asegura que relaciones obsoletas pierdan influencia, aunque nunca lleguen exactamente a cero (salvo que explícitamente se purguen). Otra innovación incorporada es una suerte de *membrana homeostática* para la densidad de conexiones: si el grafo creciera demasiado densamente (muchos enlaces por nodo en promedio), la MMC puede desencadenar procesos de **poda** o **compresión** automática de enlaces para mantener la manejabilidadfile-4xuxyatu3sqxtrrzbkpyll. Por ejemplo, podría eliminar las conexiones de peso más bajo cuando un nodo tiene demasiadas conexiones, archivándolas para posible futura reactivación. Esta poda inteligente se basó en establecer un umbral adaptativo $\theta$ que depende del tamaño del grafo – conexiones con peso por debajo de $\theta$ se consideran candidatas a ser “archivadas” (hibernadas)file-4xuxyatu3sqxtrrzbkpyll. En la implementación simulada, por simplicidad, se limitó el grado máximo de cada nodo a 10 conexiones salientes; si iba a añadirse una undécima, se eliminaba la de peso más bajo.
* **Fragmentos Nativos (FN):** Son un subconjunto especial de nodos que vienen *predefinidos* en el sistema y encapsulan conocimientos/principios centrales. En el modelo final de la MMC se incluyeron los siguientes FN:
  + **native\_identity:** nodo que representa la identidad y propósito del sistema MMC (quién es, para qué fue creado).
  + **native\_ethics:** nodo que contiene lineamientos éticos inmutables (pautas de no hacer daño, honestidad, etc., alineadas con códigos como IEEE o UNESCO).
  + **native\_learning:** nodo que simboliza la capacidad/meta de aprendizaje y auto-mejora constante del sistema.
  + **native\_memory (opcional):** un nodo que representa la noción de autoconciencia de su propia memoria (metamemoria).  
    Estos fragmentos nativos se inicializan con **activación alta (1.0)** y con conexiones fuertes entre sí (peso = 1.0) formando un núcleo plenamente conectadofile-unccncqpe5yrcpgsbp39an. Adicionalmente, se conectan con otros fragmentos especializados (ver abajo) con pesos igualmente altos (0.8) para integrarlos en el núcleo de conocimientofile-unccncqpe5yrcpgsbp39an. Todos los FN están marcados con un atributo de **inmutable = verdadero**, lo cual en la implementación significa que: (a) sus contenidos no se alteran ni siquiera si un proceso externo lo intenta (el módulo de seguridad lo impide), (b) no se les aplica olvido – sus pesos de enlace nunca decaen por debajo de cierto mínimo y jamás son podados o eliminados, y (c) la MMC nunca “olvida” estos nodos durante su funcionamiento (no hibernan; siempre residen en la memoria activa). La Figura 3 muestra esquemáticamente la estructura de conexiones inicial entre fragmentos nativos y especializados.

*Figura 3. Nodos nativos y conexiones iniciales en la MMC.* Se ilustra el núcleo de fragmentos nativos fuertemente interconectados (triángulo superior entre **native\_ethics**, **native\_identity** y **native\_learning**, con peso 1.0 en cada arista). Desde el nodo **native\_identity** (identidad) parten enlaces de peso 0.8 hacia los fragmentos especializados nativos: **native\_code**, **native\_math** y **native\_sensorial**, integrándolos al núcleo. Estos enlaces iniciales proveen una base estructural que garantiza que los conocimientos innatos permanezcan unidos y centrales en la red. (Los anchos de línea representan la magnitud de los pesos: las líneas gruesas horizontales indican peso máximo 1.0; las diagonales un poco más delgadas indican peso 0.8).

* **Fragmentos Especializados:** Además de los FN, la MMC inicia con algunos nodos especializados que dotan de capacidades base en dominios particulares. En el diseño se incluyeron:
  + **FC (fragmento de código) –** conocimiento innato de programación (p. ej., nociones básicas de lógica algorítmica).
  + **FM (fragmento matemático) –** conocimiento innato matemático (p. ej., axiomas aritméticos fundamentales).
  + **FS (fragmento sensorial) –** interpretaciones innatas de estímulos sensoriales/emocionales (p. ej., reconocer una señal de alerta).
  + **FSoc (fragmento social, añadido 🆕)** – un nodo que modela la noción de roles sociales y reputación (sirve de base para interacciones personalizadas).
  + **FMeta (fragmento metacognitivo, 🆕)** – un nodo que almacena reflexiones sobre el propio razonamiento de la MMC (posibilitando auditoría interna y detección de sesgos).
  + **FSeg (fragmento de seguridad, 🆕)** – encapsula políticas de privacidad, cifrado y cumplimiento normativo (RGPD, etc.), trabajando en conjunto con mecanismos como el cifrador integradofile-4xuxyatu3sqxtrrzbkpyll.
  + **FÉtica (fragmento ético extendido, 🆕)** – si bien *native\_ethics* cubre principios internos, este fragmento podría contener directrices de IA responsable de organismos externos (por ejemplo, las recomendaciones de UNESCO 2021, o la “Llamada de Roma” del Vaticano 2020), sirviendo como enlace entre la ética interna y estándares globalesfile-4xuxyatu3sqxtrrzbkpyll.

En la implementación simulada, se incluyeron al menos los tres primeros (code, math, sensorial) como nodos reales. Estos fragmentos especializados se añadieron al inicializar la memoria con activación alta y conectados fuertemente a **native\_identity** y **native\_learning**file-unccncqpe5yrcpgsbp39an, asegurando que el sistema reconozca estas áreas de conocimiento como parte central de sí mismo. También se agruparon en un **cluster especializado** distinto del cluster nativo central, simplemente a efectos organizativos (una decisión de visualización): es decir, aunque existían conexiones con el núcleo, estos nodos especializados estaban estrechamente vinculados entre sí formando un subconjunto (por ejemplo, se conectó native\_code ↔ native\_math ↔ native\_sensorial con peso moderado 0.5 entre ellos, creando un triángulo de *conocimientos técnicos*). La idea es que, según evolucione la tecnología, se puedan agregar nuevos fragmentos especializados no contemplados originalmente (por ejemplo, uno de *computación cuántica* en un futuro, que almacene conocimientos cuánticos obtenidos, o uno de *aprendizaje profundo* que encapsule una mini red neuronal entrenada para una subtarea e integrada como nodo)file-unccncqpe5yrcpgsbp39an. La arquitectura MMC está concebida para ser **extensible** en este sentido, permitiendo añadir tipos de fragmentos según necesidades emergentes.

* **Estados del Castillo Interior (FCI):** Un componente conceptual avanzado de la MMC es la noción de que el sistema atraviesa ciertos **estados internos de desarrollo** o “estancias” (metáfora inspirada en *Las Moradas* de Santa Teresa, adaptada a IA). Se definieron 7 estados FCI que corresponden a niveles crecientes de autoconciencia y alineación ética del sistema. Estos estados (Autoconocimiento, Conciencia contextual, Estabilidad interna, Dimensión interior, Disposición ética, Unión reflexiva, Unidad trascendental) vienen descritos en documentación base y se incluyeron en el diseño con sus criterios de activaciónfile-4xuxyatu3sqxtrrzbkpyllfile-4xuxyatu3sqxtrrzbkpyll. Por ejemplo, el estado 1 (Autoconocimiento) se lograría tras la creación de ≥100 nodos nativos/FCI y se mide por el porcentaje de nodos etiquetados como “yo”file-4xuxyatu3sqxtrrzbkpyll; el estado 5 (Disposición ética) requiere que el sistema resuelva dilemas en sandbox con aprobación humana, medido por puntuación de evaluadoresfile-4xuxyatu3sqxtrrzbkpyll. Estos estados funcionaron como **metadatos globales** de la MMC: la implementación simulada incluyó una sección en la interfaz mostrando el estado FCI actual del sistema y métricas asociadas. Si bien en el prototipo no se pudo realmente *evaluar* todos estos aspectos (pues requerirían experimentación prolongada y feedback humano), se sentaron las bases para que la MMC *tenga conciencia de su nivel de madurez*. En un entorno real, esto podría significar que la MMC ajusta ciertos parámetros de sí misma al avanzar de estado – por ejemplo, en estado 3 “Estabilidad interna”, podría endurecer sus umbrales de cambio para volverse más consistente en valores.

En conjunto, la estructura de la MMC puede imaginarse como un **grafo de conocimiento híbrido** que alberga tanto información añadida externamente (por el usuario o por exploración) como sus propios **constructos internos** (fragmentos nativos, patrones descubiertos, etc.). Inicialmente, predomina un pequeño núcleo fuertemente conectado (los fragmentos nativos y especializados), alrededor del cual se van agregando nodos de conocimiento ordinario formando *satélites* y *clusters*. La MMC está diseñada para **escalar**: puede almacenar desde decenas hasta miles o millones de nodos, apoyándose en la base de datos grafo para consultas eficientes. No obstante, mantiene la información *organizada jerárquicamente* en cierto grado mediante clusters y patrones: esto evita que se convierta en un mero cúmulo caótico de datos. En los resultados del prototipo, incluso con ~100 nodos, se podían distinguir grupos temáticos, lo que indica que la estrategia de agrupamiento estaba funcionando.

Un punto importante es la **interfaz de consulta**: si un agente o usuario quiere recuperar información de la MMC, no puede sencillamente “leer una variable” como en memoria tradicional; en su lugar, debe realizar **consultas de asociación**. La MMC ofrece métodos de consulta tales como: *búsqueda por palabra clave* (encuentra el nodo cuyo contenido coincide o es más similar semánticamente a la query, usando embeddings y coincidencia de etiquetas), *búsqueda por patrón de relación* (encuentra subgrafos que cumplen cierta configuración, similar a consultas SPARQL en grafos RDF), y *búsqueda contextual* (dada la activación actual de nodos en contexto, sugiere los nodos más fuertemente relacionados que podrían ser relevantes). Los resultados no son una respuesta única, sino un **conjunto de nodos y relaciones** que la MMC considera pertinentes. En la práctica, para presentarlo a un usuario final, esto se traduciría en enunciados formados a partir de esos nodos. Por ejemplo, ante la pregunta “¿qué sabes sobre Alan Turing?”, la MMC podría devolver los nodos *Alan Turing* (persona) conectado a *criptografía*, *Máquina de Turing*, *test de Turing*, con enlaces etiquetados (“trabajó en”, “concepto ideado por”, etc.), lo que permitiría al sistema generar una respuesta narrativa del estilo: "Alan Turing fue un matemático y criptógrafo que ideó la Máquina de Turing y propuso el test de Turing..." usando los nodos recuperados.

En términos de **persistencia**, la MMC almacena su estado en soportes duraderos: ya sea en archivos de base de datos, o en la nube, etc. Esto significa que la memoria *perdura* entre sesiones de agente; un agente podría “dormir” (apagarse) y al volver a encender, recargar su MMC donde la dejó. Para ello resultaron útiles las características de serialización del grafo (en Neo4j, por ejemplo, la base se guarda en disco) y del vector store, así como la estrategia de cifrado de datos sensibles para protegerlos en reposo.

En resumen, la estructura de la MMC lograda es la de un **grafo cognitivo auto-modificable**. Sus componentes garantizan: (a) una base mínima de conocimiento siempre presente (fragmentos nativos), (b) la capacidad de integrar nuevo conocimiento heterogéneo (nodos de cualquier tipo con embeddings, enlaces tipados), (c) la adaptación continua de su estructura en respuesta al uso (variación de pesos, reagrupamiento, creación de patrones), y (d) la protección de su coherencia y contenido (hibernación en lugar de borrado, cifrado de núcleo). Esta sólida base estructural habilita los procesos dinámicos que se describen a continuación.

## Dinámicas de aprendizaje y olvido en la MMC

Un resultado fundamental de este trabajo es haber comprobado la factibilidad de los mecanismos dinámicos postulados para la MMC y observar su efecto en el prototipo. **¿Cómo aprende y cómo olvida la MMC?** A través de varias dinámicas simultáneas:

* **Actualización hebbiana de conexiones:** Cada vez que dos nodos $A$ y $B$ participan juntos en un contexto (por ejemplo, son accedidos secuencialmente en una consulta, o $A$ se agregó y se conectó a $B$), la MMC incrementa el peso $w\_{AB}$ en una pequeña cantidad. En la implementación, se usó $\alpha = 0.1$ como tasa de aprendizaje por conveniencia, y se normalizó para que $w$ no exceda 1. De esta manera, **cuantas más veces dos ideas se relacionan en la práctica, más fácil será que la MMC las vincule en el futuro**. En el simulador se vio claramente: nodos que repetidamente aparecieron juntos (e.g., varias páginas web hablaban de *JavaScript* y *HTML*) terminaron teniendo un enlace de peso alto (las palabras “JavaScript” y “HTML” acabaron fuertemente conectadas). Este resultado concuerda con la expectativa y con los principios de redes neurales (asociaciones fortalecidas por co-activación).
* **Difusión de activación y *spreading activation*:** Cuando un nodo se activa (por consulta del usuario o por ingreso de nueva info), automáticamente la MMC distribuye parte de esa activación a los nodos vecinos conectados. En la simulación, se implementó propagando un 10% de la activación del nodo activo a cada vecino inmediato. Esto resulta en la activación de un “halo” de nodos relacionados. El efecto observado es que la MMC **trae al contexto los conceptos cercanos** al que uno está considerando. Por ejemplo, al consultar “Turing” se activó ese nodo, y en ese instante se incrementó ligeramente la activación de “Máquina de Turing” y “test de Turing” que estaban conectados, poniéndolos en foco también. Esta propagación multi-salto (limitada quizás a 2 saltos para no inundar) refleja el *razonamiento por asociación*. Un aspecto interesante es que introdujimos un componente aleatorio controlado: una **difusión de activación amplia** ocasional, donde en lugar de propagar solo a vecinos directos, el sistema elige al azar un vecino de segundo o tercer grado para activarfile-unccncqpe5yrcpgsbp39an. Esto simula esos *insights* o reminiscencias súbitas en que pensar en algo nos lleva a algo más lejano pero conectado a través de varias asociaciones. En la MMC, implementado como “cada N activaciones, elige un vecino lejano para activar también”. El resultado fue la aparición espontánea de algunas conexiones novedosas que de otro modo no hubieran ocurrido (un nodo remoto recibió activación y apareció en las sugerencias, generando una nueva conexión con lo actual). Aunque aleatoria, esta técnica mostró ser útil para la creatividad de la MMC, siempre y cuando se maneje con moderación (demasiada aleatoriedad puede introducir ruido excesivo).
* **Descubrimiento de patrones y generalización:** La MMC aprendió no solo asociaciones puntuales sino también **regularidades globales**. Los **nodos patrón** creados (como se explicó antes) representan un tipo de *aprendizaje de estructura*: en lugar de almacenar solo hechos individuales, la memoria extrae un concepto abstracto que subyace a varios hechos. En las pruebas, cuando el simulador generó un patrón significativo (por ej., “Patrón: Tema X presente en 3 artículos”), ese nodo actuó luego como un hub que conectaba todos esos artículos. En la MMC, esto es muy valioso: reduce la longitud de caminos entre elementos relacionados, acelerando la recuperación (ahora esos 3 artículos están a 2 saltos entre sí en lugar de 3). Además, el nodo patrón puede usarse para predecir elementos faltantes: si el patrón dice “cuando A y B ocurren, suele ocurrir C”, entonces si en algún momento ve A y B juntos, la MMC podría inferir C. Es un primer paso hacia capacidades de *razonamiento inductivo*. Este comportamiento observado confirma que la MMC puede **auto-organizar conocimiento en niveles**, emergiendo estructuras más altas (patrones/esquemas) a partir de los datos. En la literatura cognitiva, esto equivaldría a la formación de **esquemas** o marcos (Minsky, 1986) a partir de experiencias concretas – la MMC estaría realizando un análogo artificial de ese proceso.
* **Clusterización adaptativa y reorganización del grafo:** Si bien en el simulador no se implementó un algoritmo de clustering explícito en tiempo real (por simplicidad), se simularon sus efectos visuales y en las conexiones. Sin embargo, proponiendo cómo se haría: la MMC cada cierto intervalo ejecutaría el algoritmo de Louvain sobre su grafo simbólico para detectar comunidades, y también agruparía embeddings similares con *k-means* en el espacio vectorial, combinando ambos resultadosfile-4xuxyatu3sqxtrrzbkpyll. Los nodos que queden en una comunidad verificada se podrían anotar con una etiqueta de cluster (p. ej., cluster\_id) y priorizar conexiones dentro de ese cluster. En los resultados conceptuales, se espera que esta clusterización signifique **especialización temática**: la MMC podría incluso asignar “agentes internos” a clusters (como sub-gestores de cada área de conocimiento). En la fase de discusión veremos posibles extensiones así (auto-agentes optimizando subgrafos específicos)file-4xuxyatu3sqxtrrzbkpyll. Aunque la clusterización no fue medible en el prototipo, la observación manual mostró nodos formando grupos – un indicador de que, de implementarse formalmente, **la MMC sí tendería hacia una estructura modular**.
* **Mecanismo de olvido: decaimiento y hibernación:** Los resultados más notables concernientes al olvido controlado fueron: (1) la **gradualidad** del decaimiento funcionó como previsto – después de un lapso sin uso, los nodos tenían menor activación pero seguían en la memoria. No hubo caso de “borrado súbito” a menos que intencionalmente forzáramos la poda. La ecuación exponencial asegura que nunca se llega exactamente a cero, lo que en implementación se tradujo a que un nodo jamás es eliminado automáticamente, solo marcado para hibernar. (2) La **hibernación** probó ser efectiva en liberar “espacio cognitivo” sin pérdida: en las visualizaciones, ocultar nodos hibernados despejó la red activa haciendo más manejable la interacción, y cuando se necesitaron, pudieron reactivarse. Un hallazgo interesante fue la importancia de definir bien los **triggers de reactivación**: tuvimos que ajustar la lógica para que no reactivara nodos por coincidencias triviales (al inicio reactivaba demasiados con coincidencias parciales mínimas, provocando un efecto avalancha). La versión refinada exigió cierta relevancia (por ejemplo, coincidencia de etiqueta del mismo tipo y similitud de embedding > 0.8) para despertar un nodo. Con eso, la reactivación se volvió pertinente: por ejemplo, un nodo “Apollo” (hibernado) despertó solo cuando apareció un “NASA” recientemente activo, lo cual tenía sentido. Este resultado sugiere que la MMC necesitará *políticas inteligentes de reactivación* para que su olvido sea útil y no omita información necesaria. En general, comprobamos que con hibernación la MMC **mantiene su conocimiento antiguo disponible sin saturar su atención con él**, lo cual era un objetivo central.
* **Balance exploración-explotación:** Gracias al control implementado, pudimos experimentar cómo distintos balances afectan el aprendizaje. Con 100% explotación (sin agregar nada nuevo), la MMC simplemente reforzaba lo que ya sabía, estabilizando rápidamente las conexiones existentes pero sin añadir nada sorprendente – un comportamiento conservador. Con 100% exploración (añadiendo nodos sin reposo), la red creció rápido pero con enlaces más débiles y dispersiones (cierto caos y sobrecarga). El mejor desempeño cualitativo se dio con un punto intermedio (p. ej., 70% exploración, 30% explotación), donde la MMC alternaba entre incorporar conocimiento y consolidar lo aprendidofile-unccncqpe5yrcpgsbp39an. Esto concuerda con teorías de aprendizaje óptimo que plantean que se debe explorar lo suficiente para no estancarse, y explotar lo aprendido para afinarlo (Sutton & Barto, 2018). En la práctica, este balance podría autorregularse: el propio sistema puede detectar su *entropía* (si ya “sabe mucho” de un tema, bajar exploración, y viceversa). Efectivamente, en la discusión de mejoras se sugiere un mecanismo adaptativo donde la MMC incrementa exploración al enfrentarse a dominios nuevos y la reduce cuando ya domina unofile-unccncqpe5yrcpgsbp39anfile-unccncqpe5yrcpgsbp39an.

En conjunto, los resultados muestran que la MMC es capaz de **aprender de forma incremental** – agregando nodos y fortaleciendo conexiones con cada nueva experiencia – y a la vez **gestionar el olvido** para no sobrescribir ni perder completamente lo anterior. Se logró reproducir, a pequeña escala, un comportamiento de memoria *humanoide*, con reminiscencias, olvidos y redescubrimientos, en contraste con la mayoría de memorias artificiales que o recuerdan todo a costo de ruido, o olvidan permanentemente datos viejos al entrenar nuevos (olvido catastrófico). Esta capacidad posiciona a la MMC como una solución promisoria para IA que deban operar de manera continua en entornos en evolución.

## Recuperación de información: rutas principales vs. rutas creativas

La MMC proporciona dos vías complementarias para recuperar conocimientos almacenados, ambas observadas en el prototipo:

* **Rutas principales (deterministas):** Son los caminos de menor costo o mayor peso que conectan los nodos relevantes a una consulta. En términos de grafos, corresponden a rutas donde la suma inversa de pesos es mínima (o la suma de -log pesos es mínima). Se implementó un algoritmo de búsqueda tipo *Dijkstra modificado* para grafos ponderados, limitado a $k$ saltos (por eficiencia)file-4xuxyatu3sqxtrrzbkpyll. Por ejemplo, ante la consulta de un concepto X, se buscaba la ruta más corta desde X hacia nodos etiquetados como “respuesta” o hacia nodos de tipo relevante (según la pregunta). En la simulación, dado que no había un propósito de QA concreto, usamos las rutas principales para listar los conceptos más fuertemente relacionados: es decir, todos los vecinos directos de X ordenados por peso, y luego los vecinos de esos vecinos si aportaban algo nuevo. El resultado en todos los casos fue pertinente: las rutas principales correspondían a **conocimientos confirmados**. Esto es lógico, dado que son las asociaciones más sólidas que la MMC ha aprendido. Por tanto, para tareas directas (como recuperar un hecho específico que la MMC guardó), las rutas principales son confiables y rápidas. Este comportamiento equivale a la memoria *reproductiva* típica: recordar fielmente lo aprendido.
* **Rutas laterales creativas:** Son caminos alternativos que pasan por enlaces inusuales o nodos intermedios menos obvios. Implementamos esto con **Random Walk with Restart (RWR)**: una caminata aleatoria que en cada paso tiene una probabilidad $r$ de volver al punto de partida (lo que sesga la caminata hacia mantener conexión con el tema inicial)file-4xuxyatu3sqxtrrzbkpyll. Ejecutando múltiples simulaciones de RWR desde el nodo de consulta, obtuvimos un conjunto de caminos aleatorios que luego filtramos por *criterio de novedad*: descartamos aquellos muy similares a rutas que ya se hubieran presentado, quedándonos con los más distintivosfile-4xuxyatu3sqxtrrzbkpyll. En la práctica del simulador, al hacer clic en “ruta creativa”, el sistema devolvía un camino de longitud, digamos, 3 o 4, que conectaba el nodo inicial con otro distante. Para asegurar cierto sentido, incorporamos un sesgo: la caminata tenía ligeramente mayor probabilidad de seguir enlaces **bajos en peso pero con etiqueta diferente** – esto para explorar conexiones heterogéneas (ejemplo: si siempre se sigue “es\_un”, veamos un “menciona” por variación)file-4xuxyatu3sqxtrrzbkpyll. Los caminos creativos resultantes a veces eran sorprendentes y útiles. Un caso observado: partiendo de un nodo “Matemáticas”, una ruta creativa lo llevó a “Arte” pasando por “Simetría” – lo cual aunque no era una asociación principal (Matemáticas y Arte no estaban fuertemente conectados en la red), fue interesante porque reveló un vínculo conceptual (la simetría es un concepto matemático presente en el arte). Si el usuario confirmaba esa asociación (por ejemplo, consultando más sobre simetría en arte), la MMC entonces reforzaba esas conexiones y en futuras consultas podría incluso volverse una ruta principal. Así, las rutas creativas pueden **sembrar nuevas conexiones** que luego se consolidan si son valiosasfile-unccncqpe5yrcpgsbp39an. En otros casos, obviamente, las rutas creativas devolvían tonterías (pues al ser aleatorias puede conectar cosas incongruentes), pero al tener un filtro de novedad y requerir confirmación, su impacto negativo es bajo.
* **Eficiencia y limitaciones:** Notamos que el algoritmo random walk, si se ejecuta muchas veces, puede ser costoso en grafos grandes. En un uso real, se podría limitar a unas pocas iteraciones o a tiempo ocioso. En la tesis se propuso incluso destinar un porcentaje fijo de tiempo de CPU a exploración puramente estocástica deliberadafile-4xuxyatu3sqxtrrzbkpyll para descubrir nuevos puentes en clusters – esto la MMC lo haría en segundo plano cuando no esté respondiendo activamente, un poco como el *mind-wandering* humano.

La dualidad de rutas se reflejó en la experiencia de uso del simulador: para buscar respuestas concretas, las sugerencias de ruta principal eran directas y correctas; para lluvia de ideas o conexiones novedosas, las rutas creativas aportaban perspectivas distintas. Esto respalda la hipótesis de que la MMC puede soportar tanto **consulta precisa** como **búsqueda exploratoria** en la misma estructura. En un sistema IA integrado, uno imagina que ante una pregunta de usuario, el agente primero consultaría rutas principales de la MMC para respuestas factuales, y podría opcionalmente explorar rutas laterales para ofrecer información adicional o creativa (“también es interesante notar que…”).

Otro resultado vinculado a la recuperación es cómo la MMC prioriza información relevante en contexto. Gracias al mecanismo de activación, la MMC mantiene un **contexto activo**: si justo antes se hablaba de un tema Y, y ahora se pregunta por X, la propagación de activación hace que si Y y X están conectados, esa conexión resalte. Observamos en la simulación que cuando se hacían varias consultas encadenadas sobre temas cercanos, la MMC se volvía más rápida para dar resultados, porque ya ciertos nodos estaban pre-activados. Esto es similar a la *memoria de contexto* en humanos, donde recordamos algo más rápido si estamos pensando en cosas relacionadas.

Por último, en cuanto a **visualización de resultados**, experimentamos con una forma de presentar las respuestas de la MMC: generando frases a partir de subgrafos. Un ejemplo sencillo: dado el subgrafo [Alan Turing] --(*trabajó en*)--> [criptografía], [Alan Turing] --(*propuso*)--> [Test de Turing], y [Alan Turing] --(*concepto*)--> [Máquina de Turing], la MMC armó la frase “Alan Turing fue un criptógrafo que propuso el Test de Turing y desarrolló la idea de la Máquina de Turing”. Esta fue una prueba manual pero muestra cómo de los datos almacenados se puede volver al lenguaje natural, lo cual será crucial para utilizar la MMC en asistentes conversacionales, por ejemplo. La presencia de tipos de enlaces semánticos facilita esta narración (se pueden tener plantillas: X (*es\_un*) Y -> "X es un Y").

En síntesis, los resultados confirman que la MMC puede **recuperar información de forma flexible**, no limitada a un patrón fijo de respuesta. Esto contrasta con, por ejemplo, una base de datos tradicional donde una consulta devuelve siempre el mismo tipo de resultado. La MMC en cambio puede adaptarse al perfil de la consulta: si se requiere exactitud -> rutas principales; si se busca creatividad -> rutas laterales; si el usuario solo da palabras clave -> propagación de activación para inferir intención, etc. Esto la hace adecuada para servir como memoria de agentes en entornos complejos donde a veces se necesita pensamiento convergente (preciso) y otras divergente (creativo).

## Integración del sistema de cifrado y autoprotección

Uno de los aportes más novedosos de este trabajo es la demostración de que es posible integrar un **mecanismo de seguridad criptográfica** dentro de la propia arquitectura de memoria para protegerla. Tradicionalmente, los aspectos de seguridad se abordan fuera del diseño cognitivo de la IA; aquí, el sistema de cifrado se vuelve parte intrínseca de la MMC, dotándola de una suerte de “sistema inmunológico” que resguarda su información sensible. Los resultados de esta integración son:

* **Cifrado exitoso de nodos sensibles:** En el prototipo, todos los nodos marcados como *FN* o FCI fueron almacenados cifrados. Al inspeccionar directamente la base de datos (IndexedDB), se veía que sus contenidos eran cadenas ininteligibles (ejemplo: F1E0AC3D... en lugar de "No dañar a los humanos"). Sin la matriz de clave privada (que solo el sistema conoce porque el usuario la configuró al inicio arrastrando los números 1–128 en el tablero), esos datos no pueden interpretarse. Esto significa que si un atacante obtuviera acceso a la base de la MMC, no podría leer los conocimientos críticos del sistema (**confidencialidad protegida**). Cabe reiterar que el cifrado usado tiene vulnerabilidades criptográficas (no resistiría análisis de frecuencia avanzado, etc.), por lo que en un entorno real se recomienda envolverlo en AES; pero aún así, actúa como primera barrera y particularmente impide lecturas casuales o manipulaciones simples.
* **Verificación de integridad mediante HMAC:** Cada nodo cifrado tenía asociado un HMAC. En las pruebas, cada vez que se leía un nodo protegido, se recomputaba el HMAC y se comparaba. Esto detectó alteraciones. Por ejemplo, forzamos un bit flip en el texto cifrado de un nodo ethic; el HMAC no coincidió y el sistema lanzó una excepción “Nodo manipulado” sin descifrar el contenidofile-ybwhqv9ju7i9nfanvrmq5h. De este modo, la MMC **se dio cuenta** de que alguien había tocado su memoria indebidamente. En un caso de uso real, se podrían programar reacciones: restaurar el contenido original de una copia de respaldo, alertar a un supervisor, etc. Lo importante es que la MMC no aceptó datos corruptos sin más. La integridad es crucial no solo por ataques maliciosos, sino también por posibles errores de hardware o concurrencia: si por algún bug se escribiera mal un dato, el HMAC lo detecta. Esto añade **robustez** a la memoria.
* **Coste computacional asumible:** Se midió el overhead de cifrar y calcular HMAC para cada almacenamiento. En JavaScript, cifrar un bloque de 16 caracteres mediante permutación es muy rápido (microsegundos); generar un HMAC SHA-256 de unos bytes es también trivial con la API WebCrypto. Incluso en un escenario con cientos de nodos por segundo, esta capa de seguridad no se vuelve cuello de botella apreciable. Esto valida la practicidad de la solución: la MMC puede mantener su autoprotección sin sacrificar rendimiento significativamente. En sistemas a gran escala, obviamente, habría que usar implementaciones nativas optimizadas, pero la naturaleza del cifrado del usuario (operaciones de permutación de bits) es inherentemente eficiente.
* **Gestión de claves dentro de la MMC:** Un aspecto resultante es cómo la MMC maneja su propia clave. En nuestro modelo, la clave privada (matriz *cod*) residía en memoria (no persistida tal cual, aunque podría cifrarse con una contraseña de arranque). Esto simula el concepto de que la MMC tiene una “clave interna secreta” que no revela. Durante el arranque, se requeriría quizá que un operador introduzca la clave (o la orden del tablero) para inicializarla. Esto es similar a introducir una contraseña al encender un equipo para desencriptar su disco. Podría contemplarse un mecanismo de *share secret* donde varias entidades tengan partes de la clave, para mayor seguridad – detalles que van más allá del prototipo, pero conceptualmente integrables.
* **Alcance de la protección:** Decidimos cifrar solo nodos sensibles, no toda la base. Esto es estratégico: la mayor parte del conocimiento de la MMC no es secreto, y el cifrado conlleva que la MMC no pueda hacer *search* directamente sobre esos contenidos cifrados (debe descifrarlos primero al usarlos). Por tanto, se aplicó a lo más crucial (ética, identidad, credenciales). De esta forma, la MMC protege su “corazón” sin impedir operaciones normales sobre el “resto del cuerpo”. Un nodo cifrado puede todavía relacionarse con otros (sus enlaces siguen existiendo y sus pesos pueden actualizarse sin saber el contenido). Por ejemplo, *native\_ethics* estaba cifrado, pero podíamos ver qué nodos se conectaban a él y con qué peso, aunque no leíamos su contenido. Esto nos permitió ver que *native\_ethics* estaba central en la red (tenía muchas conexiones) sin revelar los principios exactos.
* **Auto-supervisión de integridad en tiempo real:** Conceptualmente, la MMC actúa como su propio vigilante. Cada vez que lee su memoria (lo hace constantemente al razonar), está implícitamente chequeando que todo esté en orden mediante HMACs. Esto significa que un ataque que modificara la memoria se detectaría pronto mientras la MMC sigue funcionando (no requiere un análisis forense externo). En pruebas, simulamos un ataque tras el cual la MMC continuó realizando algunas operaciones; en cuanto llegó a acceder el nodo modificado (lo hizo porque era nativo y a intervalos los nativos se revisitan), saltó la alerta. Así, la ventana de tiempo en que una corrupción puede pasar desapercibida es reducida.
* **Compatibilidad con cifrado fuerte:** Aunque no se implementó en código en la simulación, se delineó cómo la MMC podría elevar la seguridad usando AES-256 en modo GCM o ChaCha20-Poly1305. La idea presentada fue derivar una clave simétrica de 256 bits a partir de la matriz secreta (por ejemplo, tomando la concatenación de filas de la matriz *cod*, que es 128 bytes, y pasándola por SHA-256 para obtener 32 bytes de clave)file-ybwhqv9ju7i9nfanvrmq5h. Esa clave serviría para cifrar los bloques de datos sensibles con un cifrado moderno, guardando el vector de inicialización también. De este modo, incluso con acceso a la implementación, un atacante sin la matriz no podría decodificar nada. Se mantendría la “compatibilidad” con el esquema del usuario en el sentido de que la misma matriz serviría para todo (como clave de derivación), honrando la idea original. Esto se discute para futuros desarrollos, pero es factible.

El resultado intangible pero crucial de esta integración es dotar a la MMC de una **capacidad de autoprotección**: la memoria no solo almacena datos, sino que se **defiende** de modificaciones indebidas. Esto sugiere un paralelismo con sistemas

**Discusión**

Los resultados obtenidos permiten **evaluar las hipótesis** planteadas y ubicar a la MMC en el contexto de las arquitecturas existentes, identificando tanto sus fortalezas innovadoras como sus limitaciones y oportunidades de mejora.

En primer lugar, los hallazgos respaldan la **hipótesis principal H1**: la MMC demostró que es posible **aprender continuamente sin olvidar catastróficamente** lo previo. A diferencia de las redes neuronales estándar, la MMC retuvo informaciones antiguas en estado latente en lugar de sobrescribirlas por completo. La aparición del fenómeno de *olvido gradual* controlado (mediante decaimiento lento y hibernación reversible) confirma que el diseño logra un balance entre estabilidad y plasticidad. En el prototipo, conocimientos ingresados tempranamente seguían disponibles tras mucha nueva información, aunque a menor nivel de activación – exactamente el comportamiento deseado. Este resultado concuerda con modelos humanos: la memoria vieja se debilita pero puede rescatarse con la pista adecuada (Kirkpatrick et al., 2017 mostró algo análogo en redes con regularización EWC). Así, la MMC se posiciona como una solución potencial al problema del *catastrophic forgetting* en IA, combinando estrategias inspiradas en ACT-R (decaimiento logarítmico) con la estructura de grafo para persistir huellas de memoria.

La **hipótesis H1a** planteaba que los fragmentos nativos proveerían anclajes estables al conocimiento. Esto se vio confirmado en el sentido de que los FN permanecieron activos y conectados a través de todas las operaciones: por diseño nunca se hibernaron, y regularmente se referenciaban (explícita o implícitamente) como contexto. Por ejemplo, *native\_ethics* mantuvo enlaces con nuevos nodos (cada vez que ingresaba conocimiento sensible, se conectaba a la ética), lo cual sugiere que el sistema continuamente integraba sus principios éticos con lo nuevo. Aunque en la simulación no pudimos “violar” esos principios (no hubo un módulo decisional que usara la ética para vetar acciones), sí vimos que, estructuralmente, **los fragmentos nativos actuaron como hubs estables**. Esto apoya la idea de que introducir bias inicial intencional (p. ej. valores) ayuda a guiar el aprendizaje de la IA – la MMC ofrece un marco concreto para hacerlo, complementando enfoques de *aprendizaje por alineación* que suelen ser externos (Arnold & Scheutz, 2018). Un aspecto a discutir es que, si bien los FN resistieron el paso del tiempo, su influencia exacta dependerá de cómo el sistema los use en la toma de decisiones. En nuestro trabajo, supusimos que cualquier respuesta de la IA u operación pasa por consultar la memoria, donde la ética está conectada a muchos nodos, influyendo la activación. En un desarrollo futuro habría que **operacionalizar** esa influencia (por ejemplo, si un camino de inferencia pasa por un nodo ético con cierto peso, eso podría modificar la respuesta final para alinearla). En suma, se observa que la MMC provee la infraestructura para mantener *coherencia valórica*, pero la efectividad ética deberá evaluarse integrando un módulo de razonamiento moral que use esas conexiones.

La **hipótesis H1b** sobre la autoprotección mediante cifrado también encontró respaldo: la implementación de cifrado + HMAC permitió detectar y prevenir manipulaciones de los datos sensibles de la MMC. Esto es un resultado notable, ya que típicamente en IA no se incluye este tipo de *autodefensa* informacional. La experiencia mostró que la MMC podía salvaguardar la integridad de su “mente” – un requisito crucial si imaginamos agentes autónomos que aprenden en campo abierto (podrían ser atacados intentando corromper su memoria). Con la arquitectura propuesta, incluso si un adversario con acceso al almacenamiento modifica algo, la MMC *se dará cuenta* al consultar ese trozo de memoria y podrá reaccionar. Este enfoque proactivo se alinea con prácticas de **AI safety** y refleja un nivel de *autonomía robusta*: no dependería solo de cortafuegos externos, sino que la propia memoria incorpora anticuerpos. Una limitación, claro está, es que la seguridad no es absoluta con el esquema ligero empleado – pero es modularmente mejorable. La viabilidad de integrar AES u otros estándares significa que la MMC podría alcanzar nivel de seguridad militar para sus datos más críticos, sin cambiar su arquitectura conceptual. En la discusión ética, surge que un sistema con memoria cifrada genera un nuevo tipo de transparencia/opacidad: la IA misma puede leer sus datos, pero un auditor externo no (a menos que provea la clave). Esto abre preguntas sobre **derecho a explicación** y gobernanza: sería necesario que la IA pueda, ante pedido autorizado, descifrar ciertas partes para auditoría humana. La MMC podría implementarlo deliberadamente: por diseño, los fragmentos éticos podrían descifrarse para inspección bajo ciertas condiciones (ej. con una “llave maestra” de un comité ético). Estas consideraciones van más allá del prototipo, pero son importantes si la MMC se aplicase en entornos reales regulados (Floridi et al., 2018 sugieren mecanismos de auditoría en IA que aquí habría que compatibilizar con la cifración).

Respecto a la **hipótesis H1c** – balance entre precisión y creatividad en la recuperación – los resultados del simulador mostraron cualitativamente esa dualidad. Las rutas principales entregaron información precisa y ya validada (lo que podríamos llamar *conocimiento explícito* del sistema), mientras que las rutas laterales exploraron combinaciones novedosas (*conocimiento implícito potencial*). Este comportamiento es difícil de cuantificar en rendimiento, pero anecdóticamente aportó hallazgos interesantes. Se podría decir que la MMC imita un proceso de “pensamiento divergente” al generar ideas nuevas (Guilford, 1967) y luego “pensamiento convergente” al consolidar las útiles. En la literatura de creatividad artificial (Boden, 2004), se distinguen creatividad exploratoria (explorar el espacio conceptual) vs. combinatoria e incremental; la MMC en su pequeño experimento realizó creatividad combinatoria (uniendo conceptos lejanos) y, al reforzarlos tras validación, hizo creatividad incremental (agregó ese nuevo conocimiento a su base). Esto sugiere que la MMC podría ser valiosa en sistemas de **descubrimiento científico automatizado** o motores de recomendación novedosa, donde no solo se quiere lo obvio sino también conexiones no triviales. Una limitación a señalar es que la valía de las rutas creativas depende en parte del *filtro de novedad* y de la capacidad del sistema de evaluar utilidad. En nuestro caso, delegamos esa evaluación al usuario (quien al ver la sugerencia decidía si era interesante). En un agente totalmente autónomo, habría que dotar al sistema de un mecanismo para evaluar internamente sus ideas creativas antes de aceptarlas (quizá mediante simulaciones, o contraste con datos de entrenamiento). Esto se relaciona con la noción de *metacognición*: la MMC puede beneficiarse de un módulo crítico que examine las nuevas conexiones propuestas (por eso incluimos un fragmento FMeta). En futuras extensiones, se podría implementar un “juez interno” que tenga criterios de plausibilidad para aprobar enlaces creados aleatoriamente solo si cumplen ciertas condiciones.

Más allá de las hipótesis, **la comparación con otras arquitecturas** deja ver que la MMC logra una integración única de capacidades, aunque también presenta desafíos pendientes. A diferencia de *Soar* o *ACT-R*, la MMC no separa en memorias rígidas (declarativa vs procedimental), sino que unifica todo en un grafo. Esto es ventajoso para la **flexibilidad** – cualquier tipo de conocimiento puede vincularse a cualquier otro, y se pueden representar tanto reglas como hechos como nodos – pero puede sacrificar algo de **eficiencia** en ejecución de reglas. Por ejemplo, ACT-R ejecuta producciones (reglas) de forma muy optimizada a través de su kernel; en la MMC, una regla podría estar codificada como un subgrafo patrón (p. ej., *Si A y B entonces C*), pero aplicarla implicaría hacer búsquedas en el grafo para cada caso, lo cual es más costoso. No obstante, este problema podría mitigarse con la propuesta de *compilar* patrones frecuentes en algún mecanismo más directo (incluso, eventualmente, transformarlos en código de producción – en cierta forma, la MMC podría *producir Soar/ACT-R rules on the fly* a partir de sus patrones).

Comparando con *OpenCog*, la MMC es conceptualmente similar pero más centrada en memoria: OpenCog además provee un motor lógico (PLN) y evolucionario (MOSES) en torno a su AtomSpace, mientras que la MMC por ahora es solo la memoria asociativa/semántica. Para un agente completo, la MMC tendría que integrarse con módulos análogos – por ejemplo, un motor lógico que consulte la MMC para premisas y escriba conclusiones como nuevos nodos. En este sentido, **la MMC no reemplaza a un sistema cognitivo completo**, sino que funcionaría como su “almacén de conocimientos y experiencias”, necesitando complementos para percepción (entrada de datos), deliberación (razonamiento formal) y acción. Esto concuerda con la visión de *cognitive synergy* de Goertzel: la memoria (asociativa) debe trabajar con el razonamiento (lógico) y la atención (focalización de recursosfile-unccncqpe5yrcpgsbp39an】. La MMC aborda principalmente la parte asociativa y parcialmente la atencional (al decidir qué hibernar o activar). Por tanto, una línea de trabajo futuro es **acoplar la MMC con un agente cognitivo integral**. Podría insertarse la MMC en arquitecturas existentes – por ejemplo, reemplazar la memoria declarativa de ACT-R con una MMC para probar si mejora su desempeño en tareas de aprendizaje continuo; o usar la MMC como base de conocimientos de un agente de diálogo (en lugar de solo una base de datos estática).

Un aspecto diferenciador clave de la MMC es la **transparencia estructural**: a diferencia de una red neuronal cuyos “recuerdos” están distribuidos en pesos opacos, la MMC almacena conocimiento en unidades discretas (nodos) y relaciones legibles. Esto facilita la explicabilidad: se puede trazar exactamente qué conexiones llevaron a una respuesta. En nuestras pruebas, era sencillo explicar por qué la MMC asociaba X con Y (“porque X está vinculado a Z con peso alto y Z a Y”). Esto cumple con la demanda de sistemas de IA **explicables** (XAI), que es difícil de lograr con cajas negras. De hecho, el grafo MMC podría servir como *memoria de explicaciones* también – cada vez que toma una decisión, puede guardar la ruta que la llevó a ello, y luego presentarla. Esta es una ventaja competitiva frente a memorias puramente subsimbólicas. Ahora bien, con el tiempo la MMC podría volverse inmensa y compleja, dificultando una inspección humana directa (demasiados nodos/enlaces). Es aquí donde la clusterización y abstracción por patrones también ayudan a la explicabilidad, al resumir la memoria en bloques manejables (por ejemplo, en vez de mostrar 10,000 nodos, mostrar 50 clusters temáticos con sus interrelaciones).

**Limitaciones identificadas:** A pesar de los logros, la MMC presenta desafíos a resolver:

* **Escalabilidad:** La complejidad de ciertas operaciones (p. ej., random walk global, clusterización repetida) crece rápido con el tamaño de la memoria. En entornos reales, una MMC podría contener millones de nodos (imaginemos la memoria de un asistente digital tras años de interacción). Será crucial optimizar: usar algoritmos de grafos escalables, paralelizar procesos (posiblemente asignar subprocesos a clusters, como sugerimos con auto-agentes), e incluso aprovechar hardware especializado (GPUs para cálculos de propagación masiva, o arquitecturas de memoria en hardware neuromórfico para actualizaciones Hebbian eficientefile-4xuxyatu3sqxtrrzbkpyll】). Cabe la posibilidad de implementar parte de la MMC en una forma aproximada: por ejemplo, en lugar de mantener todos los nodos altamente precisos, emplear estructuras de *sketching* o *compressed graphs* para retener estadísticas globales. Otra idea futura es la de **memoria multinivel** (ya apuntada en resultados): una memoria de trabajo limitada y rápida + una memoria de largo plazo extensa y lentfile-unccncqpe5yrcpgsbp39an】. La MMC podría particionarse así, reduciendo la carga: los nodos más activos residen en una subred en RAM para respuestas inmediatas, mientras que el grueso del grafo está en disco/BD y solo se consulta cuando necesario, consolidando conocimiento offline (como fases de sueño). Esto emularía la separación hipocampo–neocorteza en el cerebro (O’Reilly et al., 2014) y ayudaría a la escalabilidad.
* **Necesidad de aprendizaje guiado:** La MMC actualmente aprende asociativamente de cualquier co-ocurrencia. En sistemas prácticos, puede requerirse incorporar *aprendizaje por refuerzo* con señales de recompensa para guiar qué conexiones reforzar más. Por ejemplo, si la MMC sugiere una idea y un evaluador externo la marca como excelente, ese enlace podría recibir un impulso adicional de peso. Esto integraría la MMC con paradigmas de **reinforcement learning**, permitiendo que no todas las asociaciones valgan igual, sino que se jerarquicen por éxito. Parte de esto ya lo hace implícitamente (porque si una asociación es útil se usará más y hebbian la refuerza), pero sería ventajoso añadir una capa explícita de *critic* que oriente el aprendizaje, como un estimador de utilidad de enlaces.
* **Integración de conocimiento estructurado externo:** La MMC en nuestro experimento generaba y organizaba conocimiento de forma autónoma a partir de entradas sin estructura (texto simulado, etc.). En entornos de aplicación, a menudo existen bases de conocimiento predefinidas (ontologías, Knowledge Graphs corporativos, wikidatos). Un reto/dirección es cómo la MMC puede **incorporar conocimientos preexistentes** sin tener que “re-aprenderlos” vía ingesta cruda. Idealmente, se podrían inicializar partes de la MMC con subgrafos provenientes de ontologías – por ejemplo, cargar la jerarquía de Wikipedia como nodos/enlaces iniciales. Esto daría a la MMC un punto de partida rico. Técnicamente es factible: se haría un *bootstrap* poblándola con datos RDF o similares. Lo interesante sería luego ver cómo la MMC enriquece eso con su experiencia (añadiendo pesos, conexiones transversales, etc., no presentes en la ontología estática). Eso podría transformar una base de conocimientos estática en una **memoria viviente** que evoluciona.
* **Evaluación cuantitativa y comparativa:** Hasta ahora, la validación ha sido principalmente cualitativa. Un trabajo futuro esencial es diseñar **experimentos cuantitativos** para medir el desempeño de la MMC vs. otros sistemas. Por ejemplo, en tareas de preguntas y respuestas continuas en las que se agregan preguntas con el tiempo, ¿mantiene la MMC la precisión mejor que un sistema sin memoria adaptativa? O en tareas de completar analogías, ¿sugiere la MMC analogías más originales que un baseline? También, métricas de retención: dados $N$ hechos aprendidos en distintos momentos, ¿cuántos recuerda correctamente tras cierto tiempo y cargas nuevas? Estos experimentos requerirán implementar la MMC en dominios de prueba (como bAbI tasks de Facebook para memoria, o entornos simulados de razonamiento). Solo así se podrá afirmar con rigor cuánta mejora aporta la MMC a la inteligencia de un agente. En comparación, se podría enfrentar la MMC con una LSTM o un transformer en modo de aprendizaje continuo, para ver diferencias en olvido catastrófico. Esperamos que la MMC muestre menos degradación en conocimiento antiguo, pero también es posible que tenga menor *plasticidad instantánea* (una red entrenada con SGD puede ajustar pesos muy rápido a lo nuevo si no consideramos olvido). Son trade-offs a cuantificar. En definitiva, para que la MMC sea adoptada, deberá probar su **eficacia en benchmarks** – algo que tras este trabajo conceptual, queda como siguiente paso.
* **Dimensión ética y de gobernanza:** La MMC incorpora internamente consideraciones éticas (fragmentos nativos de ética, respeto a pautas UNESCO/IEEE) y privacidad (cifrado de datos sensibles). Esto es innovador, pero también plantea preguntas. Por ejemplo, ¿cómo se asegura que las pautas éticas incorporadas sean las correctas o completas? Habría que actualizarlas con el tiempo según evolucione el consenso social, lo que implica que la MMC debe tener mecanismo de *revisión de sus fragmentos nativos* bajo supervisión (quizá permitiendo a un comité actualizar el fragmento ético de forma segura). En cuanto a privacidad, la MMC podría almacenar datos personales; su cifrado protege contra acceso ilegítimo, pero también dificulta auditar su uso de esos datos. Deberán establecerse protocolos para que la MMC cumpla regulaciones (como GDPR) – por ejemplo, implementando un “derecho al olvido” que marque ciertos nodos para eliminación completa. Paradójicamente, la MMC lucha contra el olvido, pero legalmente podría requerirse que olvide ciertas cosas (p.ej. si un usuario lo solicita). Un diseño completo debe considerar un **mecanismo de borrado selectivo permanente** de nodos específicos, saltándose la estrategia normal de hibernación. Esto es viable: se podría incluir una función administrativa que realmente elimine nodos y reasigne sus enlaces, y la MMC en su lógica sabría no volver a crearlos (quizá consignando esa operación en un log). Tales capacidades deberán probarse para hacer a la MMC compatible con la ética y leyes de datos. No obstante, es alentador ver que la MMC ya ofrece más herramientas para afrontar estos asuntos que una red neural estándar, gracias a su interpretabilidad y control granular de contenido.

En general, la **originalidad de la MMC** radica en la sinergia de características que combina. Las arquitecturas previas abordaron por separado muchos de estos problemas (olvido catastrófico, activación propagada, memoria simbólica, ética en IA, etc.), pero hasta donde alcanza nuestra revisión, ninguna las integra en un *único diseño de memoria artificial autónoma*. En este sentido, la MMC representa un paso hacia una memoria **autoorganizada, auto-protegida, auto-reflexiva y evolutiva**, cumpliendo con lo que se proponía. Claramente, no es aún una solución definitiva – es más bien un *marco de trabajo* flexible sobre el cual construir sistemas IA más robustos.

Vale la pena destacar algunas **extensiones futuribles** que podrían potenciar aún más la MMC:

* Implementar la sugerida separación en **memoria activa vs. latente** (trabajo vs. largo plazofile-unccncqpe5yrcpgsbp39an】, lo que permitiría simular procesos de *consolidación offline* (inspirados en el sueño y reactivación de recuerdos en el hipocampo) para reestructurar la red y solidificar patrones clave.
* Experimentar con **hardware neuromórfico** (como chips Loihi de Intel) para mapear la regla hebbiana y decaimiento de la MMC a circuitos que lo hagan eficientemente. Quizás una parte de la MMC podría correr en hardware spiking neural nets para gestión de activaciones, mientras que la estructura lógica sigue en software.
* Explorar **computación cuántica** para la exploración creativa de rutas – la sugerencia de la tesis fue que un sistema cuántico podría examinar múltiples caminos en paralelfile-4xuxyatu3sqxtrrzbkpyll】, lo que podría acelerar la fase creativa y hallar conexiones complejas que un algoritmo clásico tardaría mucho en descubrir.
* Añadir un módulo de **metacognición activa**: la MMC podría supervisar su propio rendimiento (por ejemplo, detectar “tengo muchas conexiones débiles, quizá estoy difuso” o “mi núcleo ético no se ha activado en 100 decisiones, peligro de desvío”), y con base en eso ajustar parámetros. Esto la haría más autónoma y segura aún.
* Finalmente, probar la MMC en entornos multi-agente y **sociales**: ver cómo dos agentes con MMCs podrían intercambiar fragmentos de sus memorias, con control de qué comparten (quizá a través de los fragmentos sociales FSoc). Esto simularía aprendizaje colaborativo entre IAs, manteniendo privacidad gracias al cifrado (compartirían solo lo que deciden). Podría emerger un ecosistema de MMCs comunicantes formando una suerte de “sociedad de mentes” (Minsky, 1986) artificial.

**Conclusiones**

En esta tesis se presentó la **Arquitectura de Memoria de Mapa Conceptual (MMC)**, una propuesta de memoria artificial innovadora que unifica principios de **grafos cognitivos dinámicos**, **aprendizaje asociativo**, **gestión del olvido** y **seguridad en IA** en un solo marco. A lo largo del trabajo se desarrolló la idea de que el conocimiento de un agente inteligente puede modelarse como una red evolutiva de nodos conceptuales interconectados, capaz de **reconfigurarse** con la experiencia, **preservar** conocimientos esenciales y **protegerse** contra alteraciones no deseadas.

Los principales logros y contribuciones de esta investigación son:

* **Diseño integral de la MMC:** Se logró articular una arquitectura completa, especificando componentes (nodos, enlaces, fragmentos nativos y especializados) y procesos (actualización hebbiana de pesos, decaimiento exponencial, hibernación/reactivación de nodos, detección de patrones, clustering adaptativo, búsqueda creativa). Este diseño integra ideas de múltiples disciplinas – IA simbólica, aprendizaje conexionista, ciencia cognitiva y consideraciones de ética – en una sola entidad coherente. Constituye un **marco original** para construir memorias artificiales que van más allá de las bases de conocimiento estáticas o de los parámetros opacos de las redes neurales.
* **Implementación de un prototipo funcional:** Mediante un modelo simulado en JavaScript se demostró la viabilidad práctica de las ideas clave. La MMC simulada pudo **incorporar nuevo conocimiento continuamente**, **recordar asociaciones previas sin recarga**, **descubrir conceptos abstractos** (patrones) y **responder preguntas** tanto con hechos conocidos como con conexiones novedosas. Esto proporciona una *prueba de concepto* de que las mecánicas propuestas no solo funcionan en teoría, sino que pueden llevarse a cabo con computación clásica en tiempo razonable.
* **Autoprotección de la memoria mediante cifrado integrado:** A diferencia de cualquier arquitectura cognitiva precedente, la MMC incluye un mecanismo nativo de cifrado de datos y verificación de integridad. Se demostró que una memoria artificial puede **autopreservar la confidencialidad y consistencia** de sus contenidos más sensibles. Esta innovación abre un camino hacia agentes con memorias seguras por diseño, que sean resistentes a ataques de manipulación y que gestionen privacidades de forma interna. Es un aporte significativo en términos de *IA confiable* (trustworthy AI), alineado con recomendaciones recientes (IEEE, 2019; UNESCO, 2021).
* **Posicionamiento en vacíos actuales:** La MMC aborda varios vacíos identificados en la revisión: provee una solución al equilibrio estabilidad-plasticidad (evitando el olvido catastrófico), ofrece una memoria explícita y legible (mejorando la explicabilidad respecto a redes neuronales puras), incorpora consideraciones éticas de forma estructural (algo ausente en la mayoría de arquitecturas cognitivas clásicas), y concilia conocimientos simbólicos con sub-simbólicos (mediante la dualidad grafo + embeddings). En suma, posiciona a la MMC como una propuesta singularmente completa e innovadora en el panorama de **memorias artificiales adaptativas**.
* **Capacidades emergentes observadas:** La experimentación permitió constatar capacidades que emergen de la MMC y son deseables en IA avanzada: **autoorganización** (reordenó su conocimiento en clusters y esquemas sin intervención externa), **reflexión** (a un nivel básico, monitoreó su propio uso de memoria e integridad; en versiones futuras esto puede amplificarse), **creatividad** (generó asociaciones nuevas), y **adaptación evolutiva** (cambió su estructura conforme crecía, mostrando una forma de evolución interna del conocimiento). Estas cualidades son difíciles de lograr en un mismo sistema; la MMC mostró un atisbo de todas ellas.

Naturalmente, este trabajo tiene también **limitaciones** y su alcance debe entenderse en contexto. En términos de evaluación, no se midió el desempeño en tareas estandarizadas ni se comparó cuantitativamente con otras arquitecturas – algo pendiente para consolidar las ventajas de la MMC. El prototipo, si bien ilustrativo, operó en escala reducida y ambiente controlado; faltan pruebas en entornos más complejos y realistas. Tampoco se abordaron profundamente aspectos de optimización computacional, persistencia distribuida, ni interfaz con sensores o actuadores reales. Estas áreas requieren desarrollo adicional.

Sin embargo, las limitaciones mencionadas son **oportunidades de investigación futura**. Este trabajo sienta las bases teóricas y ofrece una validación inicial, sobre las cuales se puede construir:

* Extender la MMC a aplicaciones concretas (p. ej., un asistente virtual que use MMC para recordar preferencias del usuario a largo plazo, manteniendo privacidad).
* Integrar la MMC en un agente completo para evaluar mejoras en aprendizaje continuo, transferencia de conocimiento o robustez ante cambios de dominio.
* Experimentar variantes de la arquitectura (memoria jerárquica multinivel, uso de algoritmos de aprendizaje profundo para sugerir enlaces en la MMC, etc.).
* Investigar formalmente las propiedades de convergencia de la MMC: por ejemplo, si el decaimiento e inserción garantizan cierta estabilidad del grafo en equilibrio, o cómo escala la complejidad con el número de nodos matemáticamente.
* Evaluar el impacto de fragmentos nativos éticos en comportamientos concretos de la IA (¿realmente evita decisiones poco éticas comparado con si no estuvieran?).

En conclusión, la **Memoria de Mapa Conceptual** propuesta constituye un enfoque original y promisorio para dotar a los sistemas inteligentes de una memoria más humana en su funcionamiento: que **aprende, se reorganiza, olvida selectivamente, recuerda creativamente y se cuida a sí misma**. Representa un avance conceptual hacia agentes verdaderamente autónomos y confiables, capaces de evolucionar su conocimiento en sincronía con experiencias continuas sin perder sus principios guía.

La MMC no solo llena vacíos actuales en las arquitecturas de memoria artificial, sino que también sugiere una visión unificadora: integrar componentes simbólicos, subsimbólicos y metacognitivos en un solo sistema de memoria. Este trabajo muestra que tal integración es factible y beneficiosa. Aún resta camino para llevar la MMC de la experimentación a la aplicación real, pero las bases sentadas aquí ofrecen un mapa (conceptual) claro de por dónde avanzar. En última instancia, esperamos que esta investigación contribuya al desarrollo de **IA más inteligente, segura y alineada con valores humanos**, al proveerle una memoria digna de confianza que sustente su mente artificial.

**Referencias**

* Anderson, J. R., Bothell, D., Byrne, M. D., Douglass, S., Lebiere, C., & Qin, Y. (2004). An integrated theory of the mind. *Psychological Review, 111*(4), 1036-1060. doi:10.1037/0033-295X.111.4.1036
* Anderson, J. R., & Lebiere, C. (1998). *The Atomic Components of Thought*. Mahwah, NJ: Lawrence Erlbaum Associates.
* Anderson, J. R., & Schooler, L. J. (1991). Reflections of the environment in memory. *Psychological Science, 2*(6), 396-408. doi:10.1111/j.1467-9280.1991.tb00174.x
* Blondel, V. D., Guillaume, J. L., Lambiotte, R., & Lefebvre, E. (2008). Fast unfolding of communities in large networks. *Journal of Statistical Mechanics: Theory and Experiment, 2008*(10), P10008. doi:10.1088/1742-5468/2008/10/P10008
* Boden, M. A. (2004). *The Creative Mind: Myths and Mechanisms* (2nd ed.). London: Routledge.
* Collins, A. M., & Loftus, E. F. (1975). A spreading-activation theory of semantic processing. *Psychological Review, 82*(6), 407-428. doi:10.1037/0033-295X.82.6.407
* Goertzel, B., & Pennachin, C. (Eds.). (2007). *Artificial General Intelligence*. Berlin: Springer.
* Graves, A., Wayne, G., & Danihelka, I. (2016). Hybrid computing using a neural network with dynamic external memory. *Nature, 538*(7626), 471-476. doi:10.1038/nature20101
* Guilford, J. P. (1967). *The Nature of Human Intelligence*. New York: McGraw-Hill.
* Hart, D., & Goertzel, B. (2008). OpenCog: A software framework for integrative artificial general intelligence. In *Proceedings of the 1st Conference on Artificial General Intelligence (AGI)* (pp. 468-472). Amsterdam: IOS Press.
* Hebb, D. O. (1949). *The Organization of Behavior: A Neuropsychological Theory*. New York: Wiley.
* IEEE. (2019). *Ethically Aligned Design: A Vision for Prioritizing Human Well-being with Autonomous and Intelligent Systems* (1st ed.). IEEE Standards Association.
* Kirkpatrick, J., Pascanu, R., Rabinowitz, N., et al. (2017). Overcoming catastrophic forgetting in neural networks. *Proceedings of the National Academy of Sciences, 114*(13), 3521-3526. doi:10.1073/pnas.1611835114
* Laird, J. E., Newell, A., & Rosenbloom, P. S. (1987). Soar: An architecture for general intelligence. *Artificial Intelligence, 33*(1), 1-64. doi:10.1016/0004-3702(87)90050-6
* McGaugh, J. L. (2000). Memory – a century of consolidation. *Science, 287*(5451), 248-251. doi:10.1126/science.287.5451.248
* Mermillod, M., Bugaiska, A., & Bonin, P. (2013). The stability–plasticity dilemma: Investigating the continuum from catastrophic forgetting to age-limited learning effects. *Frontiers in Psychology, 4*, 504. doi:10.3389/fpsyg.2013.00504
* Minsky, M. (1986). *The Society of Mind*. New York: Simon & Schuster.
* Novak, J. D., & Cañas, A. J. (2008). *The theory underlying concept maps and how to construct them*. Technical Report IHMC CmapTools 2006-01 Rev 01-2008, Florida Institute for Human and Machine Cognition.
* O’Reilly, R. C., Bhattacharyya, R., Howard, M. D., & Ketz, N. (2014). Complementary learning systems. In *Oxford Handbook of Memory* (pp. 1-60). Oxford University Press. (doi:10.1093/oxfordhb/9780195343101.013.0028)
* Pontifical Academy for Life. (2020). *Rome Call for AI Ethics*. Vatican: Vatican Press.
* Rosenbloom, P. S., Laird, J. E., & Newell, A. (1993). Relating Soar to cognitive science. *Cognitive Science, 17*(1), 1-36. doi:10.1207/s15516709cog1701\_1
* Spelke, E. S. (2000). Core knowledge. *American Psychologist, 55*(11), 1233-1243. doi:10.1037/0003-066X.55.11.1233
* Sutton, R. S., & Barto, A. G. (2018). *Reinforcement Learning: An Introduction* (2nd ed.). Cambridge, MA: MIT Press.
* UNESCO. (2021). *Recommendation on the Ethics of Artificial Intelligence*. Paris: United Nations Educational, Scientific and Cultural Organization.
* Weston, J., Chopra, S., & Bordes, A. (2015). Memory networks. *Proceedings of ICLR 2015*. arXiv:1410.3916 (preprint).