Diseño de Memoria en Agentes de IA para Proyectos de HAUT

Juan Camilo Rojas Ortiz

Luiggi Valencia Vélez

Santiago Botero García

Escuela Colombiana de Ingeniería Julio Garavito

HAUT_M: Hiperautomatización

Ing. Jerónimo Herrera Carbonell

Noviembre 02, 2025

1. **Resumen del caso de uso**

¿Qué hace su agente dentro del proyecto?

El agente actúa como un **asistente pedagógico inteligente** que apoya a los profesores universitarios en dos fases del proceso de calificación:

- **Fase 1:** Colabora con el docente para diseñar y ajustar rúbricas de evaluación, guiando la definición de criterios, niveles de desempeño y descriptores claros.
- **Fase 2:** Utiliza las rúbricas previamente definidas para evaluar automáticamente entregas de los estudiantes, proporcionando una retroalimentación personalizada y oportuna basada en los criterios establecidos.

De esta forma, el agente combina asistencia pedagógica y evaluación automatizada, reduciendo la carga del profesor y mejorando la calidad del feedback.

¿Qué tipo de decisiones o interacciones requiere recordar?

El agente necesita recordar información que le permita mantener coherencia pedagógica y contextual entre interacciones, como:

- Las rúbricas creadas previamente por cada profesor (criterios, niveles, ejemplos).
- Las preferencias del profesor (estilo de retroalimentación, formato de rúbrica, tono del feedback).
- Las características del curso o asignatura (tipo de actividad, nivel académico, objetivos de aprendizaje).
- Las decisiones de calibración o ajustes discutidos durante la creación de la rúbrica.
- Los historiales de evaluación (para comparar entregas o mantener consistencia en las calificaciones).

Esta memoria permite al agente mantener continuidad, evitar repetir preguntas ya resueltas, y ofrecer sugerencias más precisas en futuras interacciones con el mismo profesor o curso.

2. Diagnóstico del agente actual (si ya lo tienen)

¿Tiene o no capacidad de memoria?

Actualmente, al agente no cuenta con capacidad de memoria persistente. Todas las interacciones y procesos de calificación o generación de rubricas se realizarían en sesiones independientes, lo que significa que el modelo no recuerda decisiones previas del docente ni conserva información de largo plazo.

La información como rubricas, calificaciones o entregas de los estudiantes se gestiona mediante servicios externos como Google Drive y Google Sheets, pero el agente no tiene la capacidad de asociar estos datos de manera contextual o de aprendizaje continuo. En

consecuencia, la personalización de respuestas, la consistencia entré evaluaciones y la mejora progresiva del desempeño del agente se ven limitadas.

¿Qué limitaciones tiene en términos de contexto, aprendizaje o personalización?

A continuación, se describen las principales limitaciones identificadas en términos de contexto, aprendizaje y personalización:

1. El agente no posee memoria contextual que le permitan mantener coherencia entre interacciones. Cada sesión es independiente, lo que impide recordar decisiones previas sobre rubricas, ajustes realizados por el docente o características del curso.
2. No existe un mecanismo de aprendizaje adaptativo que permita al agente mejorar su desempeño con base en experiencia previas. No aprende de las correcciones del docente ni de los patrones de error o similitudes entre entregas, lo que limita su capacidad de evolución.
3. El agente no recuerda las preferencias individuales de los profesores, por lo que las interacciones carecen de un estilo consistente. La falta de memoria personalizada dificulta que el agente se adapte al contexto pedagógico de cada curso o docente.

3. Propuesta de arquitectura de memoria

¿Qué tipo(s) de memoria van a utilizar? (Episódica, Semántica, Procedimental)

Para el agente inteligente propuesto se puede utilizar los tres tipos de memoria:

- **Episódica**: Permitirá registrar y recuperar información relacionada con interacciones específicas entre el profesor y el agente. Esta memoria conservará datos sobre las decisiones tomadas durante la definición de rúbricas, los ajustes realizados en versiones previas, los ejemplos utilizados para ilustrar criterios de evaluación y las preferencias individuales de cada docente. De esta manera, el agente podrá retomar conversaciones previas sin necesidad de repetir información, asegurando una experiencia personalizada y coherente a lo largo del tiempo.
- **Semántica:** Permite almacenar y acceder a rúbricas, criterios de evaluación o políticas institucionales; permitiendo respuestas consistentes. Almacenando hechos esenciales e información fundamental para las respuestas del agente.
- **Procedimental:** Corresponde al conjunto estructurado de flujos de trabajo, reglas y plantillas de prompts que definen *cómo* se realiza el proceso de evaluación. Esta memoria se manifiesta principalmente en los workflows de n8n, que orquestan cada paso, así como en los prompts que guían al modelo en la aplicación de los criterios y el formato de respuesta. Gracias a esta memoria, el agente puede ejecutar evaluaciones de manera coherente, repetible y auditable, garantizando la estandarización del proceso y facilitando su mejora continua. Esta memoria será mayormente estática, los procesos de mejora, ajuste y validación de resultados estarán mediados por un mecanismo human-in-the-loop. En este enfoque, el docente cumple un rol activo como supervisor

y validador del desempeño del agente, revisándo las evaluaciones generadas, confirmando su veracidad y realizando los ajustes necesarios.

Para la definición de tipos de memoria a utilizar en el agente inteligente propuesto, se usaron como base los conceptos de (*Core Concepts*, s. f.). De igual forma, esta arquitectura híbrida con tres tipos de memoria permite prácticas RAG (retrieval-augmented generation) para contextualizar al agente LLM con documentos relevantes, y memorias episódica para aprender de decisiones anteriores. (Lewis et al., 2021).

¿Cómo estructurarían el almacenamiento y recuperación?

El agente contará con una arquitectura de memoria hibrida que combina una base de datos NoSQL distribuida (Airtable) para la gestión de la información estructurada y N8N como entorno de orquestación y memoria contextual. Esta estructura permite manejar tanto información estructurada como datos no estructurados.

En esta arquitectura, la memoria semántica se almacenará en Airtable, aprovechando su facilidad para versionar documentos, mantener trazabilidad y aplicar controles de acceso. La memoria procedimental estará representada por los workflows de n8n, que definen los pasos, condiciones y automatizaciones para la detección de entregas, ejecución de pipelines de evaluación y generación de retroalimentaciones, apoyado de un mecanismo de revisión docente, que permitirá capturar la retroalimentación sobre los resultados generados por el agente. Finalmente, la memoria episódica se gestionará también dentro de n8n, mediante sus módulos de Chat Memory Manager y persistencia de contexto conversacional, permitiendo registrar interacciones relevantes entre el agente y el docente.

El proceso de recuperación sigue un flujo de cuatro etapas:

1. La entrada del usuario se transforma en un embedding o representación semántica.
2. N8N consulta la memoria contextual para recuperar interacciones o episodios previos relacionados.
3. Se accede a los documentos persistentes en la base Airtable.
4. La información consolidada se pasa al LLM para generar una respuesta contextualizada.

Finalmente, las nuevas interacciones se registran nuevamente en la memoria, cerrando el ciclo de recordar-aprender-olvidar.

¿Qué frameworks utilizarían y por qué?

- **Memoria procedimental:**
  Se implementa mediante **flujos de trabajo en n8n**, los cuales definen la secuencia de

acciones que realiza el agente, como activar el proceso, recuperar la rúbrica, enviarla al LLM para la evaluación y guardar los resultados. Este tipo de memoria representa el "cómo" del funcionamiento del agente, asegurando que el proceso de calificación siga pasos consistentes y reproducibles. El uso de n8n facilita la orquestación visual y la modificación del flujo cuando sea necesario.

- **Memoria Semántica:** Se implementa utilizando las integraciones de n8n con **PostgreSQL o Airtable**, donde se almacenan las rúbricas, criterios de evaluación y conceptos pedagógicos relevantes de forma persistente. Esta memoria permite al agente recuperar conocimiento conceptual y factual (por ejemplo, definiciones de dimensiones de evaluación o ejemplos de retroalimentación efectiva), constituyendo el "qué" del conocimiento requerido para la calificación.
- **Memoria Episódica:** Se implementa aprovechando el **contexto o estado de conversación integrado en n8n**, el cual almacena información de corto plazo sobre las interacciones entre el profesor y el agente. Esto permite recordar elementos de sesiones anteriores, como aclaraciones o avances parciales en la definición de una rúbrica. El uso del contexto nativo de n8n simplifica la gestión del estado sin requerir infraestructura adicional (como Redis), ofreciendo una solución eficiente para mantener continuidad en la colaboración entre el profesor y el agente.

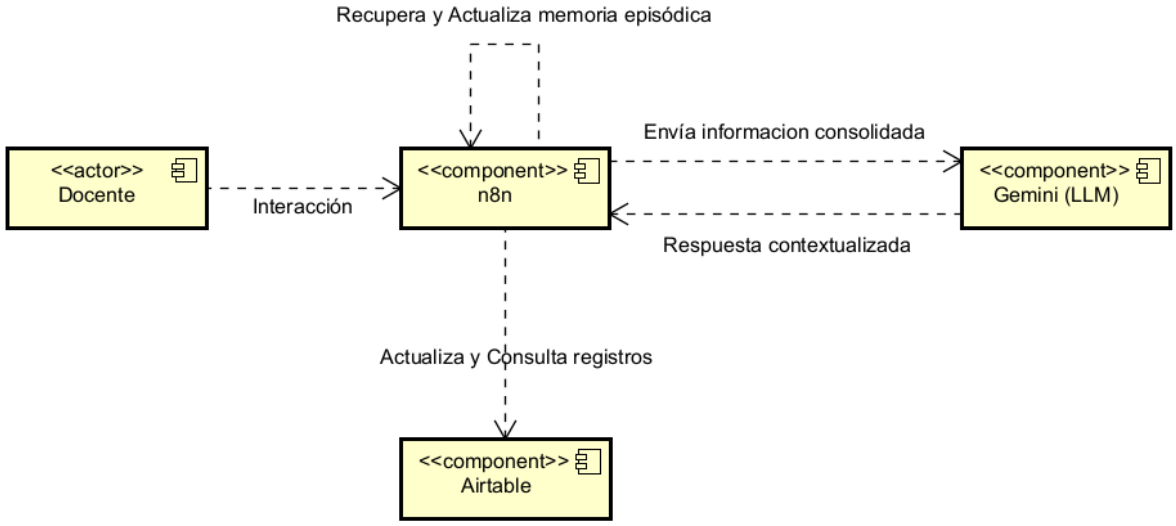
Para el caso de uso propuesto no se considera necesario la implementación de frameworks como LLamaIndex o Langchain, podrían ser útiles para vectorizar documentos que enriquezcan el modelo, no está planteado por el momento dentro de la solución.

¿Cómo manejarían el ciclo recordar-aprender-olvidar?

- **Recordar** implica acceder al conocimiento almacenado en la memoria semántica y episódica, como las rúbricas de evaluación y las interacciones previas del profesor con respecto a la definición de rúbricas. Este proceso garantiza la coherencia en los resultados y la aplicación consistente de criterios pedagógicos, así como la consistencia del contexto, evitando reproceso del docente.
- **Aprender** se refiere a la incorporación o actualización de información relevante, como nuevas rúbricas definidas junto al docente o ajustes en los criterios de evaluación. Este proceso permite que el agente evolucione y mejore su desempeño con el tiempo.
- **Olvidar** corresponde a la eliminación o archivado de información obsoleta, como versiones antiguas de rúbricas o historiales de sesiones sin valor pedagógico. Este mecanismo evita la sobrecarga de memoria y mantiene actualizado el conocimiento del sistema. Además, se deben implementar políticas de retención por categorías, como datos personales identificados, puesto que deben ser borrados o anonimizados para dar cumplimiento a la (*LEY 1581 DE 2012*, s. f.) para la protección de datos personales.

4. Modelo de implementación

Arquitectura técnica (diagrama propuesto)



Flujo de información: interacción → memoria → razonamiento → respuesta

El flujo de información inicia con la interacción del docente con el agente que es recibida por N8N, el entorno de orquestación. N8N primero consulta la memoria episódica y contextual para recuperar interacciones previas e historial relevante del docente. Paralelamente, accede a los documentos estructurados almacenados en Airtable, incluyendo rubricas, configuraciones y registros de calificaciones. Esta información consolidada se envía al modelo LLM, que genera una respuesta contextualizada y pertinente. Finalmente, las nuevas interacciones, ajustes y resultados de retroalimentación se registran nuevamente en N8N y Airtable.

Adicionalmente, durante el proceso de calificación automática, el modelo debe acceder a la rúbrica asignada para evaluar los distintos componentes del entregable presentado por el estudiante, de esta forma el modelo permite automatizar dinámicamente la calificación de distintos tipos de tareas, incluso en distintas asignaturas.

Tipos de datos que almacenan (y mecanismos de seguridad o privacidad)

Dentro de los tipos de datos que se almacenan se tienen los siguientes:

- Metadatos como el identificador docente, curso, tarea, timestamp.
- Artefactos como los entregables de cada estudiante como notebooks, PDFs, Docs, etc.
- De memoria episódica como fragmento de texto de las conversaciones previas con el profesor.
- De memoria semántica como las rúbricas, criterios de evaluación, políticas o documentación.

- De memoria procedimental como la estructura del workflow utilizado para evaluar los entregables y para proveer asistencia al docente durante la creación de rúbricas

En cuanto a los mecanismos de seguridad se propone:

- Cifrado en tránsito (TLS) y en reposo (KMS).
- Control de acceso para que solo el docente y los administradores tengan acceso.
- Anonimización de datos sensibles.
- Consentimiento explícito de los estudiantes para usar entregables en agentes inteligentes.
- Politicas de retención conforme a la Ley 1581de 2012, ofreciendo el derecho a la rectificación o supresión de los datos.

5. Justificación técnica y de negocio

¿Qué ganan como solución? (Velocidad, personalización, satisfacción, eficiencia)

En cuanto a justificación técnica se puede decir que se gana en la mejora de factualidad y contexto al hacer uso de RAG permitiendo que el agente recupere información agente con las rúbricas o decisiones previas del agente y del docente reduciendo así alucinaciones, sesgos del modelo LLM y pérdida de contexto.

En cuanto a justificación de negocio, la inclusión de memoria mejora la propuesta de valor del modelo puesto que le permite adaptarse a distintos tipos de entregas, al cambiar dinámicamente los criterios de evaluación mediante la lectura de rúbricas previamente definidas, adicionalmente la inclusión de memoria episódica se evita reproceso del docente puesto que se recordarán sus interacciones con el modelo al momento de definir las rúbricas, por lo que cuando quiera definir una nueva rúbrica el modelo recordará detalles previamente mencionados como los objetivos la asignatura o la estructura utilizada anteriormente.

¿Qué riesgos evita?

Se puede indicar que los riesgos que se evita con el uso de memoria del agente es que se reducen errores reiterativos al aprender de correcciones realizadas por el docente, se evita dependencia de memoria solo en parámetros del LLM.

Referencias

Core Concepts. (s. f.). Recuperado 29 de octubre de 2025, de https://langchain-ai.github.io/langmem/concepts/conceptual_guide/

Lewis, P., Perez, E., Piktus, A., Petroni, F., Karpukhin, V., Goyal, N., Küttler, H., Lewis, M., Yih, W., Rocktäschel, T., Riedel, S., & Kiela, D. (2021). *Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks* (No. arXiv:2005.11401). arXiv. https://doi.org/10.48550/arXiv.2005.11401

LEY 1581 DE 2012. (s. f.). Recuperado 30 de octubre de 2025, de https://www.suin-juriscol.gov.co/viewDocument.asp?ruta=Leyes%2F1684507&

Vidivelli, S., Ramachandran, M., & Dharunbalaji, A. (2024). Efficiency-Driven Custom Chatbot Development: Unleashing LangChain, RAG, and Performance-Optimized LLM Fusion. *Computers, Materials & Continua*, *80*(2), 2423-2442. https://doi.org/10.32604/cmc.2024.054360

Vithanage, D., Yu, P., Xie, Q., Xu, H., Wang, L., & Deng, C. (2025). A comprehensive evaluation of large language models for information extraction from unstructured electronic health records in residential aged care. *Computers in Biology and Medicine*, *197*, 111013. https://doi.org/10.1016/j.compbiomed.2025.111013