Instituto Tecnológico y de Estudios Superiores de Monterrey.

Escuela de Ingenierías

Maestría en Inteligencia Artificial Aplicada – Proyecto Integrador Grupo 10, Equipo 47.

Profesores:

Dra. Grettel Barceló Alonso
Dr. Luis Eduardo Falcón Morales
Mtra. Verónica Sandra Guzmán de Valle
Dr. Gerardo Jesús Camacho González
Dr. Eusebio Vargas Estrada

Modelo Base

Equipo #47

Erick Eduardo Betancourt Del Angel	A01795545
Lucero Guadalupe Contreras Hernández	A01794502
Erik Morales Hinojosa	A01795110

Fecha de entrega: 12 de Octubre del 2025

Repositorio de GitHub:

https://github.com/erikmoralestec/proyecto-integrador-grafos-de-conocimiento-llm

I. Introducción

El presente informe aborda los resultados de la fase de **evaluación de modelos de *text embedding*** y sienta las bases para la posterior **integración de datos en un Grafo de Conocimiento** utilizando **Neo4j**.

El objetivo principal de esta etapa fue seleccionar el modelo de representación vectorial más robusto para un corpus de **textos técnicos de *recalls* automotrices** de la NHTSA. Esta selección es crucial para asegurar la coherencia semántica al integrar, en fases futuras, quejas de usuarios (*complaints*) y manuales técnicos. El resultado de esta evaluación se utilizará para generar los *embeddings* que servirán como atributos de los nodos de texto en el grafo.

II. Fundamento teórico y metodológico

La representación mediante embeddings permite mapear fragmentos de texto a puntos en un espacio vectorial de alta dimensión, donde la cercanía refleja similitud semántica.
Dado el lenguaje técnico de los *recalls*, el modelo debía capturar relaciones finas entre componentes, fallas y acciones correctivas.
Para ello, se aplicó segmentación por *chunks* de ≈250 palabras, preservando coherencia local.

La metodología *Retrieval-to-Retrieval (R2R)* siguió cuatro pasos:

1. **Indexación**: generación de embeddings para cada documento.

2. **Definición de relevancias (Qrels)**: pares relevantes basados en coincidencia exacta de atributos técnicos (make, model, year, component).

3. **Búsqueda**: cada documento sirvió como consulta para recuperar sus vecinos más cercanos mediante FAISS.

4. **Evaluación**: comparación de resultados con los Qrels usando métricas estándar de recuperación.

Esta metodología permitió medir objetivamente la coherencia semántica de cada modelo antes de integrarlo al grafo.

III. Modelo elegido

Se evaluaron modelos representativos de distintas arquitecturas y tamaños.

Modelo	Parámetros	Dimensiones	Orientación
BAAI/bge-m3	568M	1024	Multitarea, inglés/multilingüe
intfloat/multilingual-e5-large-instruct	560M	1024	Multilingüe instruccional
intfloat/e5-base-v2	300M	768	Inglés general
sentence-transformers/all-MiniLM-L6-v2	66M	384	Ligero, inglés general
mixedbread-ai/mxbai-embed-large-v1	≈1B	1024	Inglés técnico y general
nomic-ai/ModernBERT-base	125M	768	BERT moderno optimizado

Tabla3.1 muestra Parámetros, dimensiones y orientación general

El modelo **mixedbread-ai/mxbai-embed-large-v1** obtuvo la mayor puntuación global (nDCG@10 = 0.645), pero se eligió **multilingual-e5-large-instruct**, que ofreció un equilibrio entre rendimiento y soporte multilingüe, esencial para la futura incorporación de textos en español. Todas las pruebas se ejecutaron bajo condiciones controladas (GPU T4, batch size uniforme y misma dimensionalidad).

Modelo	Recall@10	MRR@10	nDCG@10	Dim	Dispositivo	Tiempo de *embedding* (s)
BAAI/bge-m3	0.79	0.59	0.60	1024	GPU	775.9
intfloat/multilingual-e5-large-instruct	0.81	0.61	0.62	1024	GPU	766.1
sentence-transformers/all-MiniLM-L6-v2	0.82	0.60	0.61	384	GPU	36.0
intfloat/e5-base-v2	0.82	0.60	0.61	768	GPU	177.4
mixedbread-ai/mxbai-embed-large-v1	**0.85**	**0.63**	**0.65**	1024	GPU	581.6

Tabla 3.2 Resultados de comparación de modelos

IV. Métricas

El desempeño del modelo se evaluó mediante varias métricas. En el conjunto de **quejas**, la comparación entre e5-base-v2y un modelo ModernBERT demostró que e5 captura mejor la relación entre el texto de la queja y su componente, alcanzando un **AUC ROC de 0,664** frente a 0,478 para el modelo de referencia. Con un umbral de similitud del 25.º percentil (≈0,759) se obtuvieron una precisión de 0,545, recall de 0,751 y exactitud de 0,562 en la detección de pares correctos. En el grafo final se generaron **178 622 aristas** de queja→componente y **59 548 aristas** de queja→vehículo.

En el conjunto de **investigaciones**, el empleo de e5-large produjo 26 281 nodos de investigación, 160 nodos de componente, 76 de marca, 506 de modelo, 59 de fabricante, 146 de campaña y 2 075 de vehículo. Las similitudes coseno filtradas por los umbrales calibrados generaron **78 843 aristas texto→componente**, 21 585 aristas texto→marca, 26 281 texto→modelo, 23 665 texto→fabricante, 26 281 texto→campaña y 26 110 texto→vehículo, sumando 202 765 aristas semánticas【360848825287383†L1750-L1920】. Además se crearon 30 nodos adicionales (year y state) y 221 910 aristas explícitas que

conectan cada investigación con sus valores exactos y definen relaciones jerárquicas (modelo→marca, componente→modelo, vehículo→año, etc.).

En los **datos de recall**, Las tres métricas empleadas (Recall@10, MRR@10 y nDCG@10) son complementarias y permiten una evaluación robusta del rendimiento de los modelos:

- Recall@10 evalúa la capacidad de cobertura: mide si el modelo logra recuperar los fragmentos relevantes, independientemente del orden.
- MRR@10 captura la precisión temprana: un modelo con alto MRR prioriza correctamente los documentos más relevantes.
- nDCG@10, al ser una métrica logarítmica, equilibra ambas dimensiones ponderando la relevancia por posición. Su robustez frente a ruido y empates la convierte en la métrica más representativa del rendimiento global en tareas de recuperación semántica.

V. Preparación de Grafos en Neo4j

Tras calcular las aristas semánticas, se construyeron tablas de **nodos** y **aristas** adecuadas para su importación en Neo4j. Cada registro de investigación o queja se representa como un nodo con su identificador (investigation_i o complaint_i) y atributos como year, state, miles, injured o deaths. Las entidades categóricas (component, make, model, mfg_name, camp_no, vehicle) se transforman en nodos únicos. Las aristas se etiquetan según la relación que representan: mentions_component_semantic, mentions_make_semantic, etc., y llevan asociada la **similitud coseno** como peso. Para las investigaciones se añadieron aristas **explícitas** basadas en igualdad exacta (e.g. mentions_component_explicit) y relaciones jerárquicas (model_of_make, campaign_targets_component, vehicle_of_year).

Con los embeddings validados, se diseñó la estructura grafo para su importación en Neo4j AuraDB.
Cada entidad principal (Recall, Component, Make, Model) se modeló como nodo único con atributos relevantes.
Las relaciones se dividieron en dos tipos:

1. Explícitas y jerárquicas: derivadas de coincidencias exactas (e.g. MENTIONS_COMPONENT, MODEL_OF_MAKE, VEHICLE_OF_YEAR), que definen la taxonomía del dominio.

2. Semánticas: calculadas mediante similitud coseno entre embeddings (e.g. SIMILAR_TO), que revelan relaciones conceptuales entre campañas de recall.

El resultado es un grafo limpio, navegable y semánticamente enriquecido, listo para consultas híbridas (estructurales y vectoriales) en Neo4j Aura.

VI. Introducción a agentes para construcción de grafos.

La literatura reciente sobre cooperación cognitiva muestra que la coordinación eficaz surge cuando los participantes (humanos o virtuales) comparten y actualizan un marco común de referencia mediante grounding multimodal y reglas de intercambio (D'Avella, Camacho-González, & Tripicchio, 2022). Trasladado al dominio de grafos, un equipo de agentes especializados —planificador, recuperador, razonador sobre grafo y verificador puede mantener ese marco anclado a evidencia estructurada (nodos/aristas) y no estructurada (fragmentos embebidos) para responder consultas complejas y ejecutar tareas de mantenimiento del KG.

Un problema conocido en KG-RAG es la poda temprana: la expansión local y voraz tiende a descartar trayectorias estructuralmente relevantes antes de que el modelo pueda razonar multihop. Los métodos recientes proponen combinar una señal global derivada del grafo (por ejemplo, a través de GNN o métricas de importancia en un subgrafo contextualizado) con una señal local de similitud consulta-relación para priorizar tripletas y preservar caminos críticos antes de construir el prompt (Liu, Wang, & Li, 2025). En evaluación, esta hibridación mejora tanto la recuperación como la precisión de respuesta frente a enfoques puramente locales (Liu et al., 2025). Trabajos paralelos de KG-guiado (p. ej., KG²RAG) refuerzan la idea de usar el KG para expandir y organizar chunks semánticos, incrementando diversidad y coherencia del contexto (Zhu et al., 2025).

6.1 Patrones de consulta y modelado para agentes

La consulta en grafos se fundamenta en patrones que representan conocimiento humano reutilizable; el pattern matching permite navegar, describir y extraer información, incluyendo relaciones implícitas (Zhu, Nisbet, Yin, Wei, & Brilakis, 2025). En dominios donde las aristas necesitan propiedades propias, conviene objetificar la relación como nodo intermedio; ello exige ajustar los patrones básicos y hace que las consultas de relación adopten una estructura cercana al sujeto-predicado-objeto de RDF (Zhu et al., 2025). Aunque el estudio de referencia se centra en IFC-Graph, la metodología es transferible: patrones declarativos y "relaciones objetificadas" favorecen evidencias citables y verificables por Cypher, exactamente lo que demanda un flujo KG-RAG para controlar faithfulness.

Para consultas en lenguaje natural amplio, conviene un flujo Vector→Graph: un Retriever genera semillas desde el índice de multilingual-e5, seguido por un Graph-Reasoner que materializa un subgrafo N-hops en Neo4j y calcula puntajes globales (centralidades/comunidades o, si procede, una GNN ligera) combinados con similitud consulta-relación para reordenar tripletas antes del prompt (Liu et al., 2025). Para consultas con filtros estructurados (make/model/year/component), un flujo Graph→Vector restringe primero el subgrafo y solo después expande evidencia textual. Un Critic/Verifier exige rutas citables en Cypher y políticas de abstención cuando falte evidencia, mitigando alucinación. Este diseño es coherente con el curso "Agentic Knowledge Graph Construction" (DeepLearning.AI, 2025)

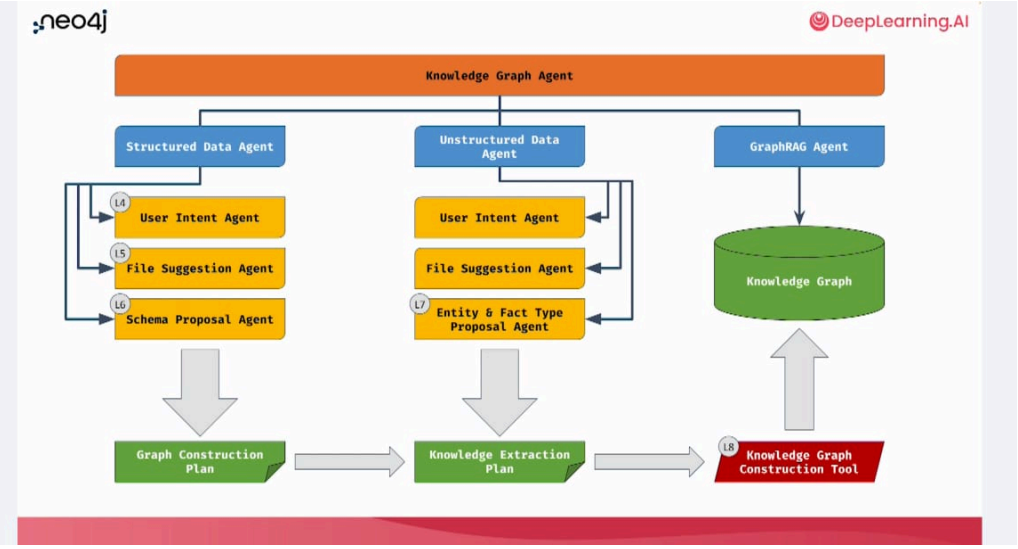

Figura 6.1 Estructura de agentes sugerida por DeepLearning.AI, recuperada el 11 de octubre del 2025.

VI. Conclusiones y próximos pasos

El pipeline implementado logra integrar datos técnicos heterogéneos en un modelo unificado de grafo, preservando coherencia semántica y jerarquía ontológica.
Los próximos pasos incluyen:

- Integrar Complaints e Investigations como capas explícita y semántica del KG con nodos dedicados y relaciones jerárquicas/por similitud hacia Component, Make, Model, Campaign y Vehicle, versionando umbrales y rutas Cypher de verificación para consultas híbridas (KG-RAG).
- Consolidar un conjunto de evaluación de 30–50 preguntas multihop del dominio NHTSA, donde cada ítem incluya filtros estructurados y una ruta Cypher verificable; esto permite juzgar la respuesta por evidencia de grafo y no solo por texto (Zhu et al., 2025).
- Implementar una capa de recuperación híbrida que, para cada consulta, extraiga un subgrafo contextualizado y compute una señal global (centralidades/comunidades, y posteriormente una GNN ligera) combinada con una señal local de similitud consulta-relación para priorizar tripletas antes del prompt (Liu et al., 2025).
- Operativizar un equipo mínimo de agentes (planificador, recuperador, razonador de grafo, verificador) que produzca, por cada pregunta, la traza: subgrafo, puntajes global/local, tripletas seleccionadas, ruta Cypher y respuesta, manteniendo el marco compartido de referencia (D'Avella et al., 2022; Liu et al., 2025).

VII. Referencias:

NHTSA datasets (Complaints, Investigations, Recalls).

Documentación técnica (Hugging Face, Multilingual E5 Text Embeddings, SentenceTransformers).

Hugging Face. (2025a). Massive Text Embedding Benchmark (MTEB) Leaderboard. Recuperado de https://huggingface.co/spaces/mteb/leaderboard

Hogan, A., Blomqvist, E., Cochez, M., d'Amato, C., Melo, G., Gutiérrez, C., … Janowicz, K. (2021). Knowledge Graphs. ACM Computing Surveys, 54(4), 1–37. https://doi.org/10.1145/3447772

D'Avella, S., Camacho-González, G., & Tripicchio, P. (2022). On multi-agent cognitive cooperation: Can virtual agents behave like humans? Neurocomputing, 480, 27–38. https://doi.org/10.1016/j.neucom.2022.01.025

Liu, H., Wang, S., & Li, J. (2025). Knowledge graph retrieval-augmented generation via GNN-guided prompting. In Proceedings of COLM 2025.

Zhu, J., Nisbet, N., Yin, M., Wei, R., & Brilakis, I. (2025). Releasing the power of graph for building information discovery. Automation in Construction, 172, 106034. https://doi.org/10.1016/j.autcon.2025.106034