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

Ingeniería de características

Equipo #47

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

Fecha de entrega: 5 de Octubre del 2025

Repositorio de GitHub:

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

I. Introducción

El presente avance forma parte del proyecto integrador orientado a la representación del conocimiento sobre defectos vehiculares mediante la transformación de datos provenientes de la National Highway Traffic Safety Administration (NHTSA), específicamente de los conjuntos Complaints, Investigations y Recalls.

Durante las etapas iniciales se desarrolló un Análisis Exploratorio de Datos (EDA) que permitió evaluar la estructura, calidad y completitud de los datos, así como identificar la naturaleza de las variables más relevantes. Dicho análisis incluyó la detección y tratamiento de valores atípicos, la eliminación de duplicados, la imputación de valores faltantes y la estandarización de atributos clave como MAKE, MODEL, YEAR y COMPONENT. A nivel textual, se aplicaron técnicas de limpieza y normalización lingüística que incluyeron la eliminación de ruido, el manejo de expresiones redundantes y la lematización, lo cual permitió preparar las descripciones de quejas (CMPLDESCR) para su análisis semántico posterior.

El tránsito hacia la representación semántica mediante grafos de conocimiento (Knowledge Graphs, KG) responde a la necesidad de modelar las relaciones entre entidades (vehículos, componentes, fabricantes, descripciones, fechas y campañas) de forma más expresiva que en estructuras tabulares. Los KGs han demostrado ser una herramienta eficaz para integrar datos heterogéneos y facilitar el razonamiento automatizado mediante inferencias semánticas y consultas basadas en relaciones (Hogan et al., 2021). En este contexto, su aplicación permite vincular la información técnica de los reportes con el lenguaje natural de las quejas, potenciando la capacidad analítica para detectar asociaciones entre componentes defectuosos, tipos de incidentes y acciones regulatorias.

En esta fase, se avanzó hacia la vectorización semántica de las descripciones textuales y las entidades estructuradas, proceso que se llevó a cabo mediante el uso de modelos de lenguaje basados en transformadores, los cuales permiten capturar la similitud contextual entre palabras y frases (Vaswani et al., 2017). Se emplearon dos modelos complementarios: ModernBERT, un modelo optimizado para manejar contextos largos y eficiente en tareas de similitud semántica (Nomic AI, 2024), y E5-base-v2, un modelo orientado al aprendizaje no supervisado de embeddings textuales, capaz de mapear textos y etiquetas en un mismo espacio vectorial (Wang et al., 2024). La utilización de estos modelos permitió generar incrustaciones semánticas (embeddings) para las descripciones de fallas y las entidades técnicas, lo que hizo posible identificar relaciones implícitas entre los reportes de usuarios y los componentes mecánicos.

En este avance, se definieron las entidades canónicas del dominio automotriz (vehículo, fabricante, modelo, componente y queja), se generaron embeddings representativos de cada una, y se establecieron umbrales de similitud para construir relaciones de correspondencia entre descripciones y componentes. Con ello, se sientan las bases para la consolidación de un grafo de conocimiento vehicular, el cual permitirá realizar búsquedas semánticas, análisis de comunidades y detección de patrones de riesgo en incidentes automotrices.

II. Fundamento teórico y metodológico

2.1 Grafos de conocimiento y embeddings semánticos

Un grafo de conocimiento (Knowledge Graph, KG) es una representación estructurada del conocimiento en forma de red, donde los nodos representan entidades (por ejemplo, vehículos, componentes o fabricantes) y las aristas modelan las relaciones entre ellas, tales como “fabricado por”, “presenta falla en” o “fue investigado en”. Estos grafos no solo almacenan información, sino que permiten razonar sobre las relaciones semánticas entre los elementos, facilitando inferencias, detección de patrones y recuperación contextual de información (Hogan et al., 2021). En el contexto del dominio vehicular, esta estructura permite vincular los datos técnicos y regulatorios con las experiencias narradas por los consumidores, integrando perspectivas cuantitativas y cualitativas sobre los defectos reportados.

El enfoque metodológico adoptado en este proyecto se basa en una integración híbrida de datos estructurados y no estructurados, combinando la información tabular proveniente de los datasets de la NHTSA (e.g., columnas como MAKE, MODEL, YEAR, COMPONENT) con las descripciones textuales de los consumidores (CMPLDESCR). Esta integración es fundamental para representar tanto los hechos explícitos (tipo de componente afectado o el año del vehículo) como el conocimiento implícito presente en el lenguaje natural, el cual suele contener indicios sobre la causa o el contexto del defecto.

Para capturar esta dimensión semántica se emplearon modelos de embeddings, los cuales permiten transformar textos y etiquetas en vectores de alta dimensión que preservan sus relaciones semánticas. Estos embeddings posibilitan medir la similitud entre quejas, componentes y modelos, y construir relaciones basadas en significado en lugar de coincidencia literal. En esta investigación se seleccionaron los modelos ModernBERT (Nomic AI, 2024) y E5-base-v2 (Wang et al., 2024) como núcleo de la representación semántica, por su capacidad para manejar secuencias largas y capturar dependencias contextuales profundas. ModernBERT destaca por su arquitectura optimizada para contextos extensos lo que resulta especialmente útil para descripciones largas de fallas, mientras que E5-base-v2 está diseñado específicamente para tareas de búsqueda y recuperación semántica, alineando textos y etiquetas dentro del mismo espacio vectorial.

Durante la fase de exploración, también se evaluó el modelo BGE-Large-en-v1.5, desarrollado por el equipo de BAAI General Embeddings (BAAI, 2024), reconocido por su alto rendimiento en tareas de recuperación semántica y sentence similarity. Aunque este modelo presenta una mayor capacidad expresiva debido a su tamaño y su entrenamiento con instrucciones, su costo computacional y requerimientos de memoria lo hacen menos viable para la escala del presente proyecto. Sin embargo, su inclusión en el proceso de investigación permitió una comparación más informada sobre las ventajas y compromisos entre precisión semántica y eficiencia computacional.

En conjunto, la combinación del modelado estructurado de entidades y la representación vectorial de texto sienta las bases para la construcción de un grafo de conocimiento vehicular semánticamente enriquecido, en el cual las relaciones no solo reflejan vínculos explícitos entre atributos, sino también asociaciones implícitas inferidas a partir del lenguaje natural contenido en los reportes de quejas. Este enfoque híbrido constituye un paso esencial hacia la comprensión profunda de los patrones de fallas vehiculares, posibilitando la detección de relaciones entre descripciones narrativas y características técnicas, más allá de las limitaciones de los modelos tabulares tradicionales.

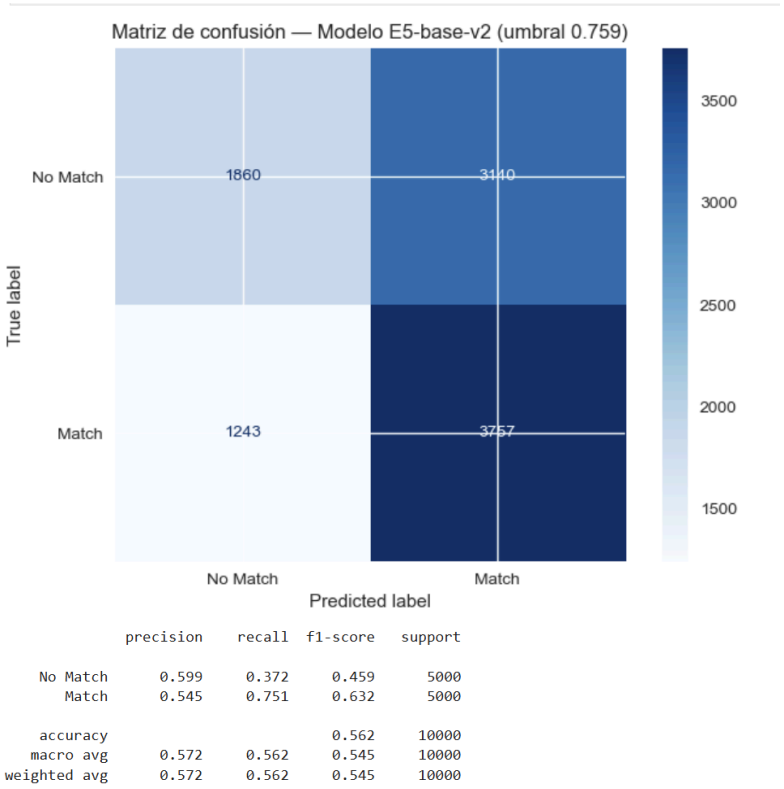
2.2 Modelos de incrustación semántica (embeddings)

Los embeddings semánticos constituyen el núcleo del proceso de vinculación entre los textos no estructurados y las entidades estructuradas presentes en las bases de datos. En este proyecto, se evaluaron distintos modelos de representación semántica con el objetivo de identificar cuál ofrecía el mejor desempeño en tareas de similitud textual y asociación contextual entre descripciones de quejas y componentes vehiculares. Entre los modelos analizados se incluyeron ModernBERT-Large, ModernBERT-Base, y E5-base-v2, además de haberse considerado BGE-large-en-v1.5 por su sólido rendimiento en benchmarks de recuperación semántica (Wang et al., 2023).

Es importante destacar que las siguientes pruebas se generaron utilizando una muestra aleatoria de 5,000 registros del conjunto de datos original. Esta decisión metodológica respondió tanto a consideraciones de rendimiento computacionas. Si bien el conjunto total contiene un volumen significativamente mayor de observaciones, el tamaño de muestra seleccionado resultó suficiente para reflejar patrones semánticos representativos en las relaciones entre las quejas, los componentes y los fabricantes.

El modelo E5-base-v2, desarrollado por Intfloat, fue seleccionado para la generación final de embeddings debido a su equilibrio entre precisión, capacidad de generalización y compatibilidad con arquitecturas optimizadas para semantic retrieval (Wang et al., 2023). Este modelo cuenta con un límite de 512 tokens, basado en la arquitectura Transformer, y ha mostrado consistentemente un desempeño competitivo en el Massive Text Embedding Benchmark (MTEB), ocupando las primeras posiciones en tareas de clustering y semantic similarity. En contraste, ModernBERT-Large, aunque soporta hasta 8192 tokens y presenta mejoras en eficiencia de cómputo (Nomic AI, 2024), obtuvo un desempeño inferior al evaluar la similitud entre descripciones de quejas y componentes, con un ROC-AUC de 0.478 frente al 0.664 obtenido por E5-base-v2.

En la Figura 2.1, se observa la matriz de confusión resultante de la calibración del umbral (0.759) para el modelo E5-base-v2. Este resultado permitió visualizar el comportamiento del modelo ante los pares texto–componente, alcanzando una accuracy global del 56.2% y un F1-score promedio ponderado de 0.545. Si bien el modelo tiende a clasificar más frecuentemente coincidencias (matches), esto refleja una preferencia hacia la sensibilidad en lugar de la precisión, una característica útil en escenarios exploratorios donde se busca maximizar la recuperación de relaciones semánticas.



	precision	recall	f1-score	support
No Match	0.599	0.372	0.459	5000
Match	0.545	0.751	0.632	5000
accuracy			0.562	10000
macro avg	0.572	0.562	0.545	10000
weighted avg	0.572	0.562	0.545	10000

Figura 2.1. Matriz de confusión del modelo E5-base-v2 (umbral 0.759). La figura muestra la distribución de verdaderos positivos, falsos positivos y falsos negativos durante la evaluación del modelo.

La Figura 2.2 muestra la distribución de similitudes coseno para ambos modelos, evidenciando una mayor separación entre los grupos positivos y negativos en el caso del modelo E5-base-v2. Esta dispersión más pronunciada indica una mejor capacidad de diferenciación semántica, elemento crucial para evitar enlaces erróneos entre descripciones ambiguas y entidades vehiculares.

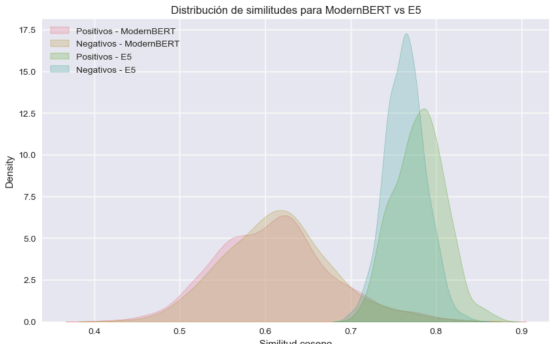


Figura 2.2. Distribución de similitudes coseno para ModernBERT y E5-base-v2. Se aprecia cómo el modelo E5 presenta una densidad más concentrada en las similitudes altas (0.75–0.85), reflejando una mayor coherencia semántica en la correspondencia texto–entidad.

En la Figura 2.3A, se presenta la proyección bidimensional (PCA 2D) de los embeddings generados con el modelo E5-base-v2, coloreados por el componente vehicular asociado a cada queja. Esta visualización permite identificar la distribución semántica de las descripciones de los consumidores en función de los sistemas del vehículo mencionados (por ejemplo, engine, electrical system, power train, entre otros). Se observa una estructura densa con zonas de ligera separación, lo cual indica que las descripciones tienden a compartir patrones léxicos comunes entre distintos componentes, pero con concentraciones más definidas en torno a sistemas específicos como “Air Bags”, “Service Brakes” o “Engine Cooling”. Estas agrupaciones reflejan la capacidad del modelo E5 para capturar relaciones semánticas coherentes entre fallas similares reportadas en diferentes contextos.

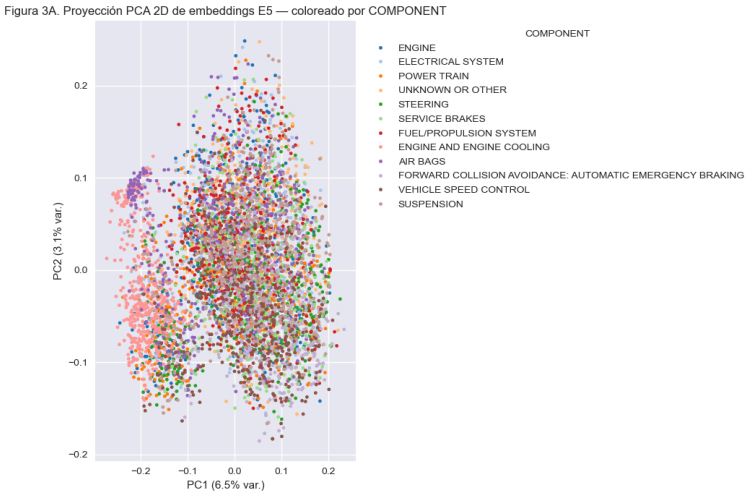


Figura 2.3A. Proyección PCA 2D de los embeddings E5 de descripciones de quejas (muestra estratificada). Cada punto es una queja y el color indica el componente reportado. Se observan agrupamientos semánticos coherentes por componente, lo que respalda el uso de embeddings para enlazar texto con entidades del grafo.

Por otro lado, la Figura 2.3B muestra la misma proyección de embeddings, pero coloreada por la marca del fabricante (MAKE). En esta vista, no se observan conglomerados completamente separados, sino una superposición significativa entre marcas, lo que sugiere que los patrones lingüísticos en las quejas no dependen principalmente del fabricante, sino del tipo de problema descrito. Sin embargo, se pueden notar agrupamientos locales en torno a marcas con comportamientos reportados distintivos, como BMW o Tesla, lo cual podría estar vinculado a la naturaleza técnica y terminológica de sus reportes.

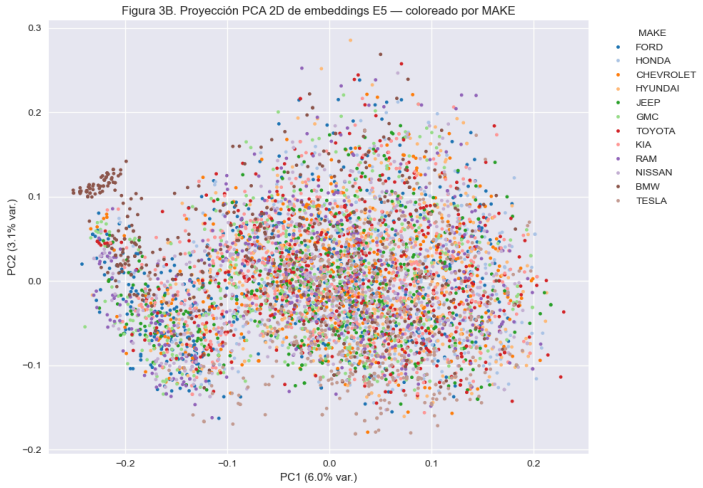


Figura 2.3B. Proyección PCA 2D de los embeddings E5 de quejas, coloreado por fabricante (MAKE). La separación parcial por marca sugiere que los reportes tienden a compartir vocabulario técnico y patrones semánticos por ecosistema de fabricante.

Ambas proyecciones evidencian que el modelo E5 logra representar de forma consistente las similitudes semánticas entre quejas vehiculares, manteniendo una estructura continua donde los vectores de texto de distinta procedencia convergen según el tipo de falla. Este tipo de visualización, además de validar la calidad del embedding, es fundamental para justificar la construcción de grafos semánticos, en los cuales las relaciones entre nodos podrán reflejar de manera explícita las afinidades observadas en el espacio vectorial.

La comparación de los modelos seleccionados se resume en la Tabla 2.1, la cual fue construida a partir del Massive Text Embedding Benchmark (MTEB), un estándar ampliamente utilizado para la evaluación comparativa de modelos de incrustación semántica (Hugging Face, 2025a). Los resultados muestran que E5-base-v2, con un máximo de 512 tokens y 109 millones de parámetros, mantiene un equilibrio óptimo entre rendimiento y eficiencia computacional. Si bien bge-large-en-v1.5 presenta un puntaje ligeramente superior en tareas de retrieval y pair classification, requiere casi el triple de memoria, lo cual limita su escalabilidad en entornos de recursos moderados. En contraste, ModernBERT-base (Alibaba-NLP, 2024) destaca por su capacidad de procesar hasta 8192 tokens, lo que lo convierte en una alternativa idónea para textos extensos; sin embargo, su desempeño general (81.57 en Semantic Textual Similarity) no supera el balance global logrado por E5. En función de estos resultados, la elección del modelo E5-base-v2 se justifica tanto por su robustez semántica como por su eficiencia de cómputo, manteniendo una cobertura contextual suficiente para las descripciones textuales de las quejas vehiculares analizadas.

Rank (Borda)	Model	Zero-shot	Memory Usage (MB)	Number of Parameters	Embedding Dimensions	Max Tokens	Classification	Clustering	Instruction Retrieval	Multilabel Classification	Pair Classification	Reranking	Retrieval	STS
62	bge-large-en-v1.5	99%	1242	335M	1024	512	48.02	41.42	-2.29	17.67	72.04	48.84	39.00	60.14
61	e5-base-v2	100%	418	109M	768	512	49.46	40.10	-3.94	16.06	72.63	47.50	40.49	60.73
207	gte-modernbert-base	99%	284	149M	768	8192	76.99	46.47	76.99	46.47	85.93	59.24	55.33	81.57

Tabla 2.1. Comparativa de desempeño entre modelos de embeddings (bge-large-en-v1.5, e5-base-v2 y gte-modernbert-base) en el benchmark MTEB. Fuente: Hugging Face (2025a); Alibaba-NLP (2024).

III. Procesamiento y generación de embeddings

3.1 Normalización semántica y embeddings de texto

Antes de la generación de embeddings, se llevó a cabo un proceso integral de normalización semántica, el cual incluyó la limpieza, estandarización y tokenización de los textos. Las descripciones de las quejas fueron preprocesadas mediante la eliminación de caracteres especiales, transformación a minúsculas, corrección de espacios y supresión de términos sin valor semántico (como stopwords).

Este paso fue fundamental, ya que permitió asegurar la consistencia léxica entre las descripciones textuales y las entidades canónicas, evitando que variaciones tipográficas o gramaticales influyeran en las medidas de similitud. La normalización incrementa la comparabilidad semántica, garantizando que dos descripciones con contenido equivalente produzcan representaciones vectoriales similares.

Los embeddings generados corresponden tanto a las descripciones limpias de quejas (CLEAN_TEXT) como a las entidades canónicas derivadas de las combinaciones de MAKE, MODEL y YEAR. La generación de vectores se realizó utilizando modelos de lenguaje preentrenados, en particular E5-base-v2 y ModernBERT-Large, los cuales ofrecen representaciones densas en espacios de 768 dimensiones. Estos modelos fueron seleccionados debido a su robustez en tareas de semantic textual similarity (STS) y retrieval, permitiendo cuantificar la cercanía semántica entre las quejas y los componentes reportados.

3.3 Calibración del umbral de similitud

Durante el proceso de vinculación entre las descripciones textuales de quejas y las categorías de componentes, se implementó una calibración basada en la distribución empírica de las similitudes coseno obtenidas entre ambos conjuntos de embeddings. Esta calibración permite definir un umbral de decisión, es decir, un valor mínimo de similitud que indica cuándo una descripción puede considerarse semánticamente asociada a un componente vehicular específico.

En las Figuras 3.1A y 3.1B se muestran los resultados de la calibración para los modelos ModernBERT-Large y E5-base-v2, respectivamente. El modelo ModernBERT-Large alcanzó un umbral sugerido de 0.565, mientras que E5-base-v2 mostró un valor de 0.759, lo cual sugiere que este último requiere una mayor similitud para considerar una relación válida entre texto y componente. En términos prácticos, una similitud de 0.56 implica una correspondencia semántica moderada, donde las descripciones comparten parte del contexto, mientras que valores cercanos o superiores a 0.75 reflejan una relación semántica más fuerte y específica.

El método de calibración se basó en el cálculo de los percentiles de similitudes positivas, una técnica recomendada para evitar la selección arbitraria de umbrales (Reimers & Gurevych, 2019). A partir del percentil 25 de las similitudes más altas, se determinó el umbral que optimiza el equilibrio entre precisión y cobertura semántica. Este enfoque no solo permite una mejor interpretación del comportamiento del modelo, sino que también facilita la comparación entre arquitecturas, como ModernBERT y E5, en términos de sensibilidad y robustez en la detección de relaciones semánticas.

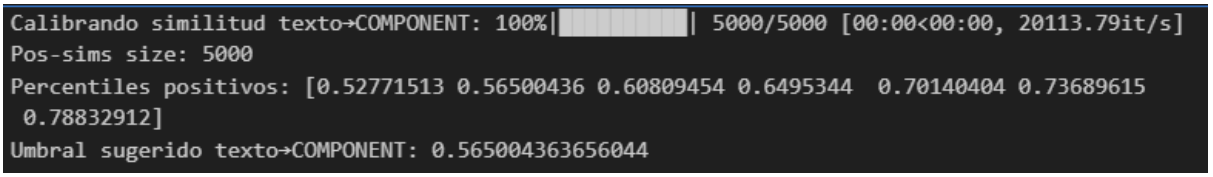


Figura 3.1A. Calibración de similitud texto→COMPONENT utilizando el modelo ModernBERT-Large (umbral 0.565).

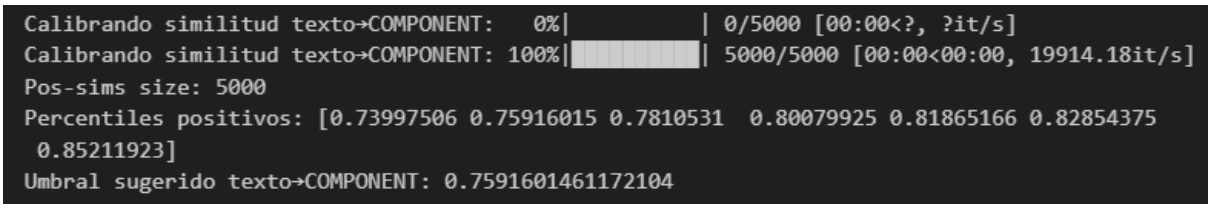


Figura 3.1B. Calibración de similitud texto→COMPONENT utilizando el modelo E5-base-v2 (umbral 0.759).

IV. Normalización y Preparación de Complaints NHTSA para Grafo en Neo4j

En esta etapa se trabajó en la estructuración y normalización del conjunto Complaints proveniente de la Office of Defects Investigation (ODI) de la NHTSA, con el fin de preparar los datos textuales y estructurados para su integración en el grafo de conocimiento. El objetivo fue representar semánticamente las quejas de consumidores y vincularlas con los vehículos y componentes afectados, generando una base sólida para análisis de relaciones y co-ocurrencias de fallas.

4.1. Preprocesamiento y Normalización de Campos

Partiendo del EDA desarrollado en etapas anteriores, se aplicaron transformaciones complementarias para asegurar consistencia semántica y tipológica entre las entidades principales. Las operaciones más relevantes incluyeron:

Marcas (MAKE) y Modelos (MODEL): Se estandarizaron a mayúsculas, se eliminaron espacios redundantes, guiones inconsistentes y anotaciones entre paréntesis. Los modelos se unificaron en variantes canónicas (e.g., F150 → F-150).

Años (YEAR): Se verificó la plausibilidad dentro del rango 1950–2035 y se forzaron los valores a tipo int64.

Componentes (COMPONENT): Se forzaron a mayúsculas, se eliminaron términos genéricos (“UNKNOWN”, “UNSPECIFIED”) y se mantuvieron únicamente componentes válidos identificados durante la limpieza.

Texto de la queja (CLEAN_TEXT): Se aplicaron operaciones de normalización (minúsculas, eliminación de stopwords y signos de puntuación).

Posteriormente, se construyó una clave canónica VEHICLE_KEY, compuesta por la concatenación (MAKE + MODEL + YEAR), empleada para asociar cada queja con su vehículo correspondiente.

4.2. Generación de Embeddings y Vinculación Semántica

A partir del texto limpio (CLEAN_TEXT), se generaron embeddings mediante el modelo E5-base-v2, seleccionado por su desempeño robusto en tareas de similitud textual (ROC-AUC = 0.664). Los embeddings se calcularon tanto para las descripciones de quejas como para los componentes, modelos y vehículos canónicos, generando representaciones vectoriales comparables.

El proceso de vinculación entre texto y entidades estructuradas se basó en la similitud coseno entre los vectores de queja y los de referencia. Para evitar emparejamientos espurios, se implementó una calibración de umbral de similitud (ver Figura 5A), que permitió

definir un valor mínimo de 0.759 para el modelo E5. Las relaciones con puntuaciones por debajo de este umbral fueron descartadas.

Las aristas resultantes se definieron bajo dos relaciones principales:

```
(:Complaint)-[:MENTIONS_COMPONENT]->(:Component)
(:Complaint)-[:MENTIONS_VEHICLE]->(:Vehicle)
```

Este proceso dio origen a los archivos nodes_df.csv y edges_df.csv, exportados como fuente para la carga en Neo4j.

4.3. Modelado en Neo4j

Con base en la estructura conceptual del grafo, se definieron tres tipos principales de nodos: Complaint, Vehicle y Component. Cada uno fue indexado por atributos únicos que aseguran consistencia y evitan duplicidades:

```
CREATE CONSTRAINT complaint_unique IF NOT EXISTS FOR (c:Complaint)
REQUIRE c.id IS UNIQUE;
CREATE CONSTRAINT vehicle_unique   IF NOT EXISTS FOR (v:Vehicle)
REQUIRE (v.make, v.model, v.year) IS UNIQUE;
CREATE CONSTRAINT component_unique IF NOT EXISTS FOR (c:Component)
REQUIRE c.name IS UNIQUE;
```

La carga de datos se efectuó a partir de los archivos complaints_nodes.csv y complaints_edges.csv, importados mediante transacciones por lotes para garantizar estabilidad en la ingesta:

```
CALL {
  LOAD CSV WITH HEADERS FROM 'file:///complaints_edges.csv' AS row
  MERGE (s:Complaint { id: row.source })
  MERGE (t:Component { name: row.target })
  MERGE (s)-[:MENTIONS_COMPONENT { weight: toFloat(row.weight)
}]->(t);
} IN TRANSACTIONS OF 1000 ROWS;

CALL {
  LOAD CSV WITH HEADERS FROM 'file:///complaints_edges_veh.csv' AS
row
  MERGE (s:Complaint { id: row.source })
  MERGE (v:Vehicle { name: row.target })
  MERGE (s)-[:MENTIONS_VEHICLE { weight: toFloat(row.weight)
}]->(v);
} IN TRANSACTIONS OF 1000 ROWS;
```

Este procedimiento garantizó que las quejas quedaran vinculadas de manera unívoca a los vehículos y componentes más probables según la similitud semántica del texto original.

4.4. Validación y Análisis Estructural Inicial

Tras la carga en Neo4j, se verificó la correcta formación de las relaciones mediante consultas de verificación:

```
MATCH (c:Complaint)-[:MENTIONS_COMPONENT]->(x:Component)
RETURN count(c) AS complaints, count(x) AS components, count(*) AS
relations;
MATCH (c:Complaint)-[:MENTIONS_VEHICLE]->(v:Vehicle)
RETURN count(v) AS vehicles, count(*) AS relations;
```

Las pruebas confirmaron la creación de nodos y relaciones con alta cobertura semántica y sin duplicados.

En la Figura 4.1 se puede apreciar la vecindad del componente ENGINE. El nodo de Component actúa como centro y se conecta a múltiples Complaints que lo mencionan y, a su vez, a los Vehicles vinculados por queja. Esta vista tipo “estrella” permite inspeccionar rápidamente qué cohortes de vehículo concentran quejas relacionadas con el motor y facilita la lectura puntual de casos desde la interfaz de Neo4j.

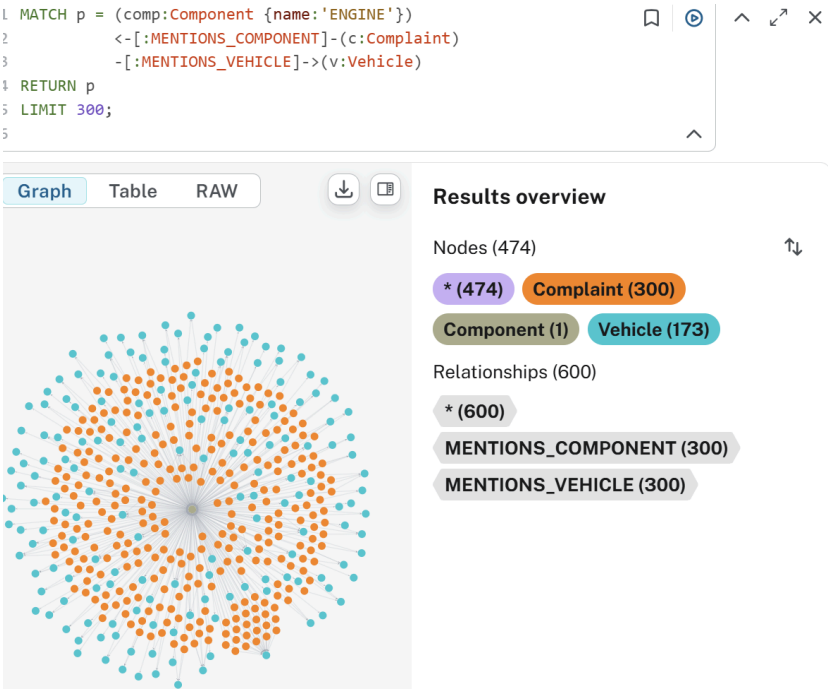


Figura 4.1. Estrella de un componente específico

En la Figura 4.2 se puede apreciar el resumen del esquema del grafo en Neo4j. Se listan las etiquetas de nodo (Complaint, Component, Vehicle, Make, Model), las relaciones disponibles (MENTIONS_COMPONENT, MENTIONS_VEHICLE, ABOUT_MAKE, ABOUT_MODEL, ABOUT_VEHICLE) y las principales propiedades (por ejemplo, text, score, key, year, miles, vehspeed, injured, deaths, faildate, ldate, datea). Esta información verifica la consistencia del modelo y orienta la definición de índices y consultas analíticas.

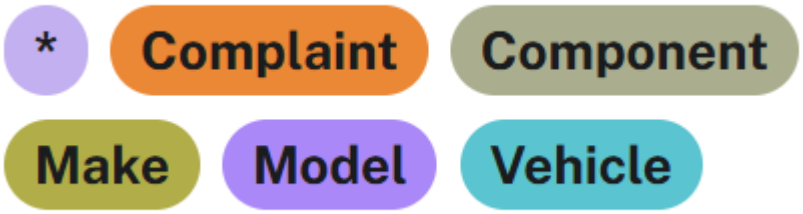


Figura 4.2. Información de esquema y propiedades del grafo (Database Information)

V. Normalización y Preparación de Recalls NHTSA para Grafo en Neo4j

Tras la etapa previa de análisis exploratorio de datos (EDA) y validación inicial, en esta semana también se avanzó en la limpieza profunda y normalización adicional de los datos de recalls de la NHTSA, con el objetivo de preparar un conjunto de datos confiable y estandarizado para su ingesta en un grafo de conocimiento. El trabajo se enfocó en tres ejes: (i) depuración de claves mínimas, (ii) normalización semántica de marcas, modelos y componentes, (iii) construcción de un CSV mínimo y consistente para Neo4j.

5.1. Filtrado y Normalización de campos

Se definieron las claves mínimas requeridas (campaign_no, make, model, year, component). Las observaciones que no cumplían simultáneamente con estas condiciones fueron separadas en un archivo de rechazados (recalls_cleaned_rejected.csv), preservando trazabilidad. El conjunto válido (recalls_cleaned_good.csv) conserva únicamente las filas con claves completas y años de modelo en el rango aceptado (1950–2035). Esta depuración garantizó que cada fila destinada a Neo4j represente una campaña válida y consistente.

Se implementaron transformaciones adicionales en atributos fundamentales:

* Marcas y fabricantes (make, mfgname): Se aplicó estandarización a mayúsculas, eliminación de espacios redundantes y supresión de sufijos corporativos (e.g. Inc., LLC, S.A. de C.V.). Se generó una versión canónica (make_norm) para reducir ambigüedad en la representación de nodos de fabricante.

* Modelos (model): Se eliminaron anotaciones secundarias entre paréntesis. Se unificaron guiones y espacios múltiples. Se incluyeron reemplazos manuales para casos recurrentes (e.g. F150 → F-150, RAM1500 → RAM 1500). Se almacenó en un campo normalizado (model_norm).

* Componentes (component): Se forzó a mayúsculas y se dividió jerárquicamente en hasta tres niveles (comp_l1, comp_l2, comp_l3). Se generó un nivel canónico (component_norm) basado en el primer nivel de la jerarquía. Se construyó además un campo component_group que agrupa componentes en categorías estandarizadas (AIR BAGS, SERVICE BRAKES, POWER TRAIN, etc.).

* Fechas (recall_date, ODATE, DATEA): Se transformaron a formato ISO (YYYY-MM-DD) con manejo explícito de valores vacíos.

* Años (year): Se restringieron al rango plausible 1950–2035 y se marcó como NaN cualquier valor fuera de este rango.

5.2. Verificación de Consistencia y exportación a Neo4j

Se aplicaron múltiples controles para garantizar coherencia:
* No existencia de claves vacías en las columnas mínimas.
* Ausencia de duplicados en campaign_no después de la deduplicación por fecha más reciente.
* Años válidos siempre como enteros (Int64).
* Formato correcto de fechas en ISO.
* Cobertura alta en campos narrativos (subject, consequence).

Finalmente, se construyó un dataset reducido (recalls_neo4j_ready.csv) que concentra las variables esenciales para la fase de grafo:

* campaign_no
* recall_date
* make_norm
* model_norm
* year
* component

Este archivo constituye la versión lista para ingestión en Neo4j, garantizando compatibilidad con la estructura de nodos y relaciones prevista en la modelación del grafo.

5.3. Modelado en Neo4j

Se definió un esquema mínimo y explícito para garantizar unicidad lógica y acelerar consultas. Las restricciones crean índices subyacentes y evitan duplicidades al momento de la ingesta:

```
CREATE CONSTRAINT vehicle_unique  IF NOT EXISTS FOR (v:Vehicle)
REQUIRE (v.make, v.model, v.year) IS UNIQUE;
CREATE CONSTRAINT component_unique IF NOT EXISTS FOR (c:Component)
REQUIRE c.name IS UNIQUE;
CREATE CONSTRAINT recall_unique IF NOT EXISTS FOR (r:Recall)
REQUIRE r.campaign_no IS UNIQUE;

CREATE INDEX vehicle_make   IF NOT EXISTS FOR (v:Vehicle) ON
(v.make);
CREATE INDEX vehicle_model  IF NOT EXISTS FOR (v:Vehicle) ON
(v.model);
CREATE INDEX vehicle_year   IF NOT EXISTS FOR (v:Vehicle) ON
(v.year);
CREATE INDEX component_name IF NOT EXISTS FOR (c:Component) ON
(c.name);
CREATE INDEX recall_date   IF NOT EXISTS FOR (r:Recall)   ON
(r.date);
```

Este diseño asume la cohorte de vehículo como combinación de (make, model, year), el componente por nombre canónico y el recall por su identificador de campaña. Los índices apoyan filtros frecuentes y aceleran agregaciones posteriores.

La ingesta se realizó a partir del archivo normalizado `recalls_neo4j_ready.csv` ubicado en la carpeta de importación de Neo4j. Se utilizaron transacciones por lotes para estabilidad y rendimiento; el procedimiento consolida nodos y relaciones sin duplicaciones:

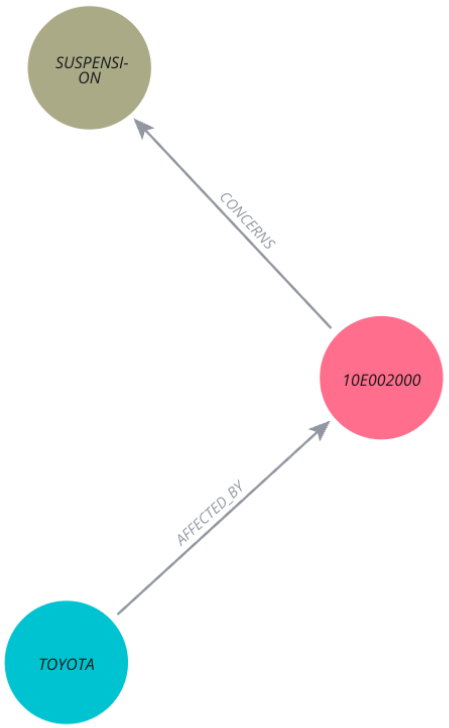
```
CALL {
  LOAD CSV WITH HEADERS FROM 'file:///recalls_neo4j_ready.csv' AS
row
```

```
  WITH
      row,
      toInteger(row.year) AS y,
      CASE WHEN row.recall_date IS NOT NULL AND row.recall_date <>
'' THEN date(row.recall_date) END AS rdate,
      toUpper(trim(row.make_norm))  AS mk,
      toUpper(trim(row.model_norm)) AS mdl,
      toUpper(trim(row.component_norm)) AS comp_name

  MERGE (v:Vehicle { make: mk, model: mdl, year: y })
  MERGE (c:Component { name: comp_name })
  MERGE (r:Recall { campaign_no: row.campaign_no })
      ON CREATE SET r.date = rdate
      ON MATCH  SET r.date = coalesce(r.date, rdate)

  MERGE (v)-[:AFFECTED_BY]->(r)
  MERGE (r)-[:CONCERNS]->(c);
} IN TRANSACTIONS OF 1000 ROWS;
```

El uso de MERGE sobre las claves canónicas asegura idempotencia: las re-ejecuciones no multiplican nodos ni relaciones. Las fechas se parsean en ISO y se aplican mayúsculas/recortes para consistencia.



En la Figura 5.1 se puede apreciar la representación de una relación de datos en un modelo de grafos. Se observa cómo la entidad "fabricante", con el valor TOYOTA, se conecta a la entidad "campaña" (ID 10E002000) mediante el vínculo AFFECTED_BY. Posteriormente, la

campaña se vincula a la entidad "componente", con el valor SUSPENSIÓN, a través de la relación CONCERNS.

5.4. Validación post-ingesta y analítica de grafos (GDS)

Tras la carga, se ejecutaron verificaciones de población y muestreos de relaciones para confirmar la estructura esperada:

```
MATCH (v:Vehicle)  RETURN count(v) AS vehicles;
MATCH (r:Recall)   RETURN count(r) AS recalls;
MATCH (c:Component)RETURN count(c) AS components;

MATCH
(v:Vehicle)-[:AFFECTED_BY]->(r:Recall)-[:CONCERNS]->(c:Component)
RETURN v.make, v.model, v.year, r.campaign_no, c.name
LIMIT 10;
```

v.make	v.model	v.year	r.campaign_no	c.name
TOYOTA	CELICA	1986	10E002000	SUSPENSION
NISSAN	TITAN	2010	10E019000	SUSPENSION
CHRYSLER	SEBRING CONV	2006	10E059000	SUSPENSION
WILSON	LIVESTOCK TRAILER	2003	10V060000	SUSPENSION
PONTIAC	MONTANA	2007	10V110000	SUSPENSION
UNIMOG	U500NA	2005	10V149000	SUSPENSION
DODGE	CARAVAN	2008	10V164000	SUSPENSION
KENWORTH	W900	2010	10V185000	SUSPENSION
NISSAN	TITAN	2010	10V208000	SUSPENSION
SPARTAN	GLADIATOR	2009	10V210000	SUSPENSION

La Tabla 5.1 muestra el resultado de una consulta a una base de datos que une información de vehículos, campañas de revisión y componentes. La tabla contiene registros de recalls donde cada fila representa una tupla con el esquema: (v.make, v.model, v.year, r.campaign_no, c.name). El objetivo de esta data es agregarla para obtener conteos (COUNT) de recalls y así generar métricas de popularidad que ayuden a priorizar el enfoque en dashboards y análisis de datos.

Se incorporó un etiquetado de popularidad (conteo de recalls) para componentes y cohortes de vehículo, útil para priorizar análisis y dashboards:

```
// Conteo y anotación de recalls por componente
MATCH (r:Recall)-[:CONCERNS]->(c:Component)
WITH c, count(*) AS n
SET c.recallsCount = n;
```

```
// Conteo y anotación de recalls por Vehicle (make-model-year)
MATCH (v:Vehicle)-[:AFFECTED_BY]->(r:Recall)
WITH v, count(r) AS n
SET v.recallsCount = n;
```

Para detectar agrupamientos estructurales se proyectó un grafo en memoria y se aplicó Louvain a través de Neo4j Graph Data Science (GDS):

```
CALL gds.graph.project(
  'veh_recall_comp',
  ['Vehicle','Recall','Component'],
  {
     AFFECTED_BY: {type:'AFFECTED_BY', orientation:'UNDIRECTED'},
     CONCERNS:  {type:'CONCERNS',    orientation:'UNDIRECTED'}
  }
);

CALL gds.louvain.write('veh_recall_comp', { writeProperty:
'community' })
YIELD communityCount, modularity;

// Grado de componentes (centralidad estructural)
CALL gds.degree.stream('veh_recall_comp')
YIELD nodeId, score
WITH gds.util.asNode(nodeId) AS n, score
WHERE 'Component' IN labels(n)
RETURN n.name AS component, score AS degree
ORDER BY degree DESC LIMIT 20;
```

El procedimiento asigna a cada nodo una etiqueta de comunidad que puede emplearse para análisis de co-ocurrencia entre componentes y cohortes de vehículos, y permite extraer listas de componentes de alta conectividad (grado) potencialmente críticos en la propagación de fallas.

component	degree
EQUIPMENT	2199.0
ELECTRICAL SYSTEM	1589.0
SERVICE BRAKES	941.0
STRUCTURE	813.0
AIR BAGS	748.0
POWER TRAIN	709.0
FUEL SYSTEM	705.0
STEERING	681.0
SUSPENSION	637.0
EXTERIOR LIGHTING	557.0

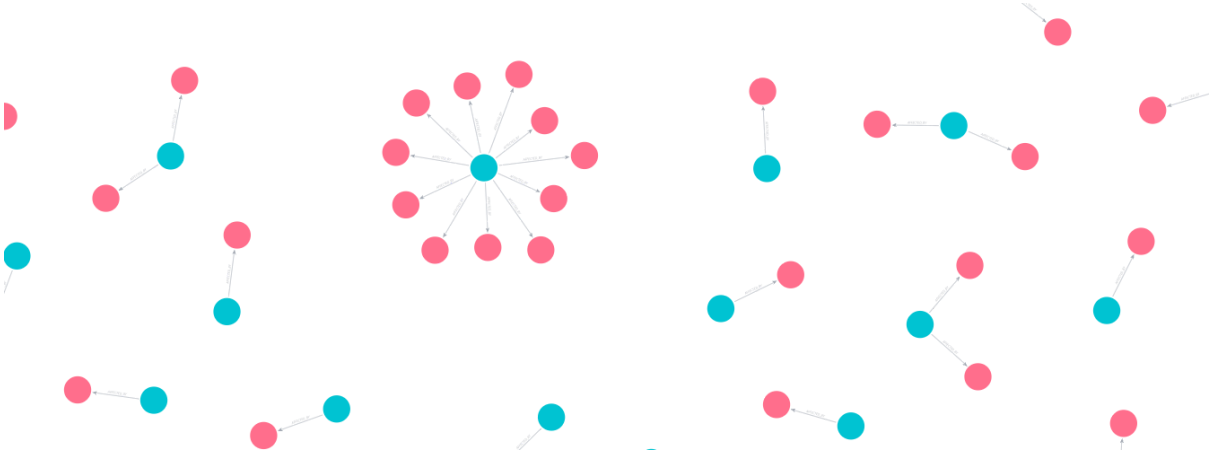
ENGINE AND ENGINE COOLING	535.0
EQUIPMENT:OTHER:LABELS	477.0
SEAT BELTS	402.0
SEATS	381.0
VISIBILITY	333.0
ELECTRICAL SYSTEM:WIRING	311.0
EQUIPMENT:RECREATIONAL VEHICLE/TRAILER	280.0
TIRES	272.0
EQUIPMENT ADAPTIVE/MOBILITY	229.0
LATCHES/LOCKS/LINKAGES	216.0

La Tabla 5.2 muestra los componentes más importantes organizados según su grado de conexión, que indica cuán relevantes y frecuentes son en el grafo de conocimiento.

A fin de explorar la estructura de comunidades detectadas, se ejecutó la consulta:

```
MATCH p = (n)-[e]-(m)
WHERE n.community IS NOT NULL AND m.community = n.community
RETURN p
LIMIT 150;
```

Esta instrucción selecciona todos los paths p formados por dos nodos n y m conectados por una relación e, restringidos a aquellos nodos que poseen la propiedad community y que comparten la misma etiqueta de comunidad. El resultado devuelve un subconjunto limitado (150 paths) que puede visualizarse en el panel de grafo de Neo4j, mostrando clústeres internos de nodos fuertemente relacionados según la partición de Louvain.



La Figura 5.2 es una visualización de Neo4j que muestra **clústeres de comunidad** (Louvain) que revelan grupos de componentes y vehículos con una alta interconexión.

VI. Conclusiones y próximos pasos

La construcción y validación del grafo de recalls en Neo4j permitió pasar de un conjunto tabular de datos a una representación estructural donde vehículos, campañas y componentes se encuentran explícitamente relacionados. Esta transformación posibilita consultas más expresivas que las que se pueden realizar sobre tablas planas, tales como: identificar vehículos con múltiples campañas, detectar componentes recurrentemente implicados o explorar comunidades técnicas emergentes a través de algoritmos de detección de clústeres.

El etiquetado de comunidades mediante Louvain mostró que es factible **agrupar componentes y cohortes de vehículos en conglomerados** que comparten patrones de falla o campañas de retiro comunes. Esta capacidad sugiere la existencia de regularidades estructurales que pueden explotarse en fases posteriores de análisis.

En cuanto a los próximos pasos, se plantean varias líneas de trabajo:

- **Enriquecimiento multimodal**: integrar al grafo otras fuentes de la NHTSA, como quejas (*Complaints*) e investigaciones (*Investigations*), para ampliar la trazabilidad desde la detección temprana hasta la resolución de cada problema.

- **Análisis semántico de narrativas**: incorporar embeddings o modelos de lenguaje para agrupar descripciones textuales de defectos y consecuencias, permitiendo una búsqueda más flexible y la predicción de similitudes entre campañas.

- **Métricas de centralidad y propagación**: aplicar algoritmos adicionales de Graph Data Science (PageRank, Betweenness, etc.) para priorizar componentes críticos en términos de seguridad o volumen de vehículos afectados.

- **Visualización avanzada**: generar vistas interactivas del grafo filtradas por marca, año o categoría de componente, que permitan a analistas e ingenieros identificar

patrones de riesgo de manera intuitiva.

- **Predicción de fallas**: explorar modelos supervisados o semisupervisados que, apoyados en la estructura del grafo, anticipen posibles futuros recalls para ciertas cohortes de vehículos.

En síntesis, el grafo ya construido constituye una **infraestructura analítica sólida** sobre la cual se podrán desplegar modelos predictivos y herramientas de soporte a la toma de decisiones en seguridad vehicular.

VII. Referencias:

NHTSA datasets (Complaints, Investigations, Recalls).

Fuentes teóricas de embeddings y KG (Bordes et al., 2013; Mikolov et al., 2013; Devlin et al., 2019).

Documentación técnica (Hugging Face, ModernBERT, SentenceTransformers).

Alibaba-NLP. (2024). GTE ModernBERT Base. Recuperado de https://huggingface.co/Alibaba-NLP/gte-modernbert-base

Devlin, J., Chang, M. W., Lee, K., & Toutanova, K. (2019). BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. Proceedings of NAACL-HLT, 4171–4186.

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

Nomic AI. (2024). ModernBERT: Efficient and scalable Transformer models for long-context understanding. Recuperado de https://huggingface.co/nomic-ai/ModernBERT

Wang, K., et al. (2023). E5: A multi-task text embedding model for retrieval and classification. arXiv preprint arXiv:2309.09812.

BAAI. (2024). BGE-Large-en-v1.5: General Embeddings Model for English Semantic Representation. Beijing Academy of Artificial Intelligence. https://huggingface.co/BAAI/bge-large-en-v1.5

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