
Proyecto Final: "Co-piloto Jurídico Híbrido: Análisis de Precedentes Legales mediante RAG Híbrido y LLMs Locales"

Presenta: Ignacio David Vázquez Pérez

Titulares del Diplomado: M. en C. Tania Gisela Alcántara Medina M. en C. Fernando Javier Aguilar Canto M. en C. José Alberto Torres León

Fecha de Entrega: 22 de octubre de 2025

Índice

1. **Introducción**
    - 1.1. Contexto del Problema
    - 1.2. Importancia del Proyecto
    - 1.3. Objetivos Específicos
    - 1.4. Alcance y Limitaciones
2. **Marco Teórico**
    - 2.1. Procesamiento de Lenguaje Natural (PLN)
    - 2.2. Embeddings Semánticos (Sentence Transformers)
    - 2.3. Bases de Datos Vectoriales (ChromaDB)
    - 2.4. Modelos de Lenguaje Grandes (LLM)
    - 2.5. Generación Aumentada por Recuperación (RAG)
3. **Metodología**
    - 3.1. Descripción General del Proceso
    - 3.2. Datos: Origen y Descripción
    - 3.3. Preprocesamiento y Extracción de Datos
    - 3.4. Arquitectura del Modelo: Pipeline RAG con Re-ranking Híbrido
    - 3.5. Criterios y Métricas de Evaluación
    - 3.6. Herramientas y Tecnologías
4. **Implementación**
    - 4.1. Fase 1: Procesamiento de PDFs y Paralelización
    - 4.2. Fase 2: Indexación en la Base de Datos Vectorial

    - 4.3. Fase 3: Inferencia (Clasificación Híbrida y Generación Múltiple)
5. **Resultados**
    - 5.1. Consulta de Ejemplo
    - 5.2. Resultado del Modelo (Análisis de Precedentes Generado)
    - 5.3. Comparación con los Objetivos
6. **Análisis y Discusión**
    - 6.1. Interpretación de los Resultados
    - 6.2. Evaluación de Éxito y Limitaciones del Modelo
    - 6.3. Propuestas de Mejora y Trabajo Futuro
    - 6.4. Escenarios de Aplicación
7. **Conclusiones**
    - 7.1. Resumen de Resultados
    - 7.2. Cumplimiento de Objetivos
    - 7.3. Impacto Profesional
8. **Referencias Bibliográficas**
9. **Anexos**
    - 9.1. Código Fuente Principal (Script v9.8)
    - 9.2. Prompts Refinados del Sistema

---

1. Introducción

1.1. Contexto del Problema

El dominio legal es uno de los campos más intensivos en información textual. Diariamente, se generan miles de sentencias, resoluciones y otros documentos jurídicos que forman un cuerpo de conocimiento vasto y complejo conocido como jurisprudencia. Para los profesionales del derecho, la capacidad de navegar eficientemente este océano de información es una necesidad fundamental para fundamentar argumentos y tomar decisiones informadas. Los métodos tradicionales basados en palabras clave a menudo son insuficientes para capturar la intención o el fondo semántico de un caso.

1.2. Importancia del Proyecto

Este proyecto, "Co-piloto Jurídico Híbrido", aborda la necesidad de un sistema de recuperación de información legal más inteligente y contextual. Su importancia radica en:

- **Eficiencia:** Automatiza el procesamiento de grandes volúmenes de PDFs, ahorrando incontables horas de trabajo manual.
- **Precisión Semántica:** Utiliza técnicas de IA (Embeddings) para permitir búsquedas basadas en el significado de un caso, no solo en palabras clave.
- **Privacidad y Costo:** Implementa un flujo de trabajo completo con modelos de código abierto (Sentence Transformers, Llama 3.1) y herramientas locales (Ollama,

ChromaDB), garantizando que los datos sensibles nunca abandonen la máquina del usuario y eliminando costos recurrentes de APIs.
- **Generación de Valor Avanzado:** El sistema va más allá de la simple recuperación. Primero, identifica la categoría legal más probable de la consulta mediante un re-ranking híbrido numérico. Luego, recupera los tres precedentes más relevantes dentro de esa categoría y genera un análisis detallado de cada uno, actuando como un verdadero co-piloto.

1.3. Objetivos Específicos

1. **Objetivo 1:** Desarrollar un pipeline robusto y paralelizado (`ProcessPoolExecutor`) para la ingesta y preprocesamiento de un corpus de documentos PDF, incluyendo la extracción de metadatos, muestreo estratégico y limpieza de texto.
2. **Objetivo 2:** Implementar un sistema de clasificación híbrido (`classify_dynamically`) que utilice una combinación de búsqueda semántica (Coseno) y métricas léxicas (Jaccard, KEM) para identificar la categoría de acción/delito más relevante para una consulta de usuario.
3. **Objetivo 3:** Configurar el sistema RAG para usar la categoría predicha como un filtro de metadatos estricto en `ChromaDB`, recuperando y analizando los N=3 precedentes más relevantes.
4. **Objetivo 4:** Desarrollar un conjunto de prompts dinámicos por materia (Penal, Civil, Familiar, Mercantil) para guiar al LLM (`llama3.1:8b`) en la extracción de hechos y la generación de informes analíticos para cada precedente.
5. **Objetivo 5:** Priorizar el uso de GPU (CUDA) para la generación de embeddings, optimizando el rendimiento de la indexación y la inferencia.

1.4. Alcance y Limitaciones

Alcance:

- Procesa todos los archivos `.pdf` ubicados en un directorio específico.
- Extrae metadatos (materia, acción) basándose en la nomenclatura de archivos.
- Implementa un muestreo estratégico (primeras, últimas y páginas intermedias) para PDFs largos.
- Limpia el texto y extrae secciones clave (Vistos, Considerandos, Resuelve) mediante Expresiones Regulares.
- Utiliza un re-ranking híbrido-numérico (Coseno, Jaccard, KEM, Concisión) para clasificar la consulta del usuario.
- Recupera y genera un informe analítico detallado para cada uno de los 3 precedentes más relevantes encontrados tras el filtrado.
- Está diseñado y probado para textos en español.

Limitaciones:

- **Calidad del Re-ranking:** La precisión del filtrado inicial depende de la efectividad de la fórmula de re-ranking híbrido y sus pesos. Una fórmula mal ponderada puede llevar a una clasificación de categoría incorrecta.
- **Dependencia de Metadatos:** La efectividad del filtrado depende críticamente de la calidad de los metadatos extraídos de la nomenclatura de archivos. Errores en el nombre del archivo (datos de origen) conducen a resultados de búsqueda incorrectos.
- **Calidad del PDF:** El sistema depende de `PyPDF2` y no puede procesar PDFs que sean imágenes escaneadas (requeriría OCR).
- **Capacidad del LLM:** Se utiliza `llama3.1:8b`. Aunque potente, ocasionalmente falla en generar informes, requiriendo un Control de Calidad (`FAILURE_KEYWORDS`) para descartar resultados inválidos.

---

2. Marco Teórico

2.1. Procesamiento de Lenguaje Natural (PLN)

El PLN es una rama de la IA que se ocupa de la interacción entre las computadoras y el lenguaje humano. En este proyecto, el PLN se aplica en:

- **Tokenización:** Realizada implícitamente por el modelo de embeddings y `spaCy`.
- **Lematización:** Uso de `spaCy` (`es_core_news_sm`) para reducir palabras a su raíz (ej. "lesiones", "lesionado" -> "lesión") y así calcular métricas léxicas como Jaccard y KEM.
- **Extracción de Información:** Uso de Regex para identificar y aislar secciones semánticas específicas (ej. `CONSIDERANDO`, `RESUELVE`).

2.2. Embeddings Semánticos (Sentence Transformers)

Un embedding es una representación vectorial de texto. La distancia entre vectores en este espacio refleja la similitud semántica. El proyecto utiliza `sentence-transformers/all-mpnet-base-v2`, un modelo potente que "entiende" el significado del texto y lo proyecta a un vector de 768 dimensiones, permitiendo que consultas semánticas encuentren documentos relevantes aunque no compartan las mismas palabras clave.

2.3. Bases de Datos Vectoriales (ChromaDB)

Una vez que los documentos se convierten en vectores, se necesita una base de datos especializada. `ChromaDB` es la base de datos vectorial utilizada. Es de código abierto, nativa de IA y persistente, ideal para desarrollos locales. Almacena los embeddings y permite búsquedas de similitud eficientes (ANN), además de filtrado por metadatos.

2.4. Modelos de Lenguaje Grandes (LLM)

Los LLM son redes neuronales masivas entrenadas en trillones de palabras, capaces de generar texto coherente y seguir instrucciones (prompts).

- **Ollama:** Es el framework utilizado para servir modelos LLM de forma local, exponiéndolos a través de un API.
- **Llama 3.1:** El modelo `llama3.1:8b` es el generador de texto de este proyecto. Es un modelo de código abierto de última generación, muy competente en tareas de razonamiento y seguimiento de instrucciones en español.

2.5. Generación Aumentada por Recuperación (RAG)

El RAG es el paradigma central de este proyecto. Resuelve el problema de que los LLMs no conocen información privada o reciente (como las sentencias de este corpus). El flujo RAG funciona en tres pasos:

1. **Recuperación (Retrieve):** Se usa el modelo de embeddings y la base de datos vectorial (ChromaDB) para recuperar los N documentos (precedentes) más relevantes.
2. **Aumento (Augment):** El texto del documento recuperado se inserta en el prompt que se enviará al LLM.
3. **Generación (Generate):** Se le pide al LLM que responda a la consulta basándose únicamente en el contexto proporcionado.

Este enfoque "ancla" al LLM a los hechos del precedente, forzándolo a generar un análisis basado en información verídica y reduciendo la alucinación.

---

3. Metodología

3.1. Descripción General del Proceso

El proyecto sigue un pipeline estructurado en tres fases:

1. **Fase de Ingesta y Procesamiento:** Se escanea un directorio de PDFs. Cada PDF es leído (con muestreo estratégico), limpiado (Regex) y su texto relevante es extraído. El resultado (texto limpio + metadatos) se guarda en lotes en formato Parquet de forma paralelizada.
2. **Fase de Indexación:** Se leen los archivos Parquet. El texto limpio de cada documento se convierte en un embedding vectorial (priorizando GPU). Este embedding, junto con el texto y los metadatos, se almacena en ChromaDB.
3. **Fase de Inferencia (Análisis RAG Híbrido):**

    - **Etapa 1: Clasificación Híbrida:** La consulta del usuario se usa para realizar una búsqueda semántica inicial contra un caché de *categorías* (no de documentos). Los resultados (Top 20) se someten a un re-ranking numérico (Jaccard, KEM, etc.) para seleccionar la *mejor categoría* para filtrar.
    - **Etapa 2: Recuperación Filtrada:** Se realiza una búsqueda semántica en ChromaDB, pero filtrada (cláusula `where`) para incluir solo documentos que coincidan con la categoría ganadora. Se recuperan los N=3 precedentes más relevantes.
    - **Etapa 3: Generación Múltiple:** El sistema itera sobre los 3 documentos recuperados. Para cada uno, utiliza el LLM para extraer hechos clave y generar un informe analítico estructurado.
    - **Etapa 4: Consolidación:** Todos los informes se consolidan en una única respuesta.

3.2. Datos: Origen y Descripción

- **Origen:** El conjunto de datos (`dataset_stjjalisco_sentencias`) es una colección de resoluciones y sentencias judiciales en formato PDF.
- **Descripción:** Documentos de texto no estructurado que varían en longitud, con la estructura típica de una sentencia (Vistos, Considerandos, Resolutivos).
- **Metadatos Adicionales:** El script extrae la materia y la acción/delito de la nomenclatura de los archivos PDF (`parse_metadata_from_filename`).

3.3. Preprocesamiento y Extracción de Datos

El preprocesamiento es un paso crítico:

- **Lectura de PDF:** Se utiliza `PyPDF2`.
- **Muestreo Estratégico de Páginas:** Para documentos de más de 50 páginas, se leen las primeras 10, las últimas 15 y una muestra de páginas intermedias.
- **Limpieza de Ruido:** Se usa `RE_NOISE_F1` para eliminar texto irrelevante como "página X", "toca...", "expediente...", etc.
- **Extracción de Secciones Clave:** Se utiliza `RE_STRUCTURAL_KEYWORDS_F1` y `RE_FINAL_BOILERPLATE_F1` para aislar el cuerpo jurídico principal (desde "VISTOS" o "RESULTANDO" hasta "NOTIFÍQUESE" o "ASÍ LO RESOLVIÓ").

3.4. Arquitectura del Modelo: Pipeline RAG con Re-ranking Híbrido

La arquitectura es un sistema multi-componente:

- **Clasificador Híbrido (Función `classify_dynamically`):**

    - **Componente Semántico:** Utiliza `cosine_similarity` entre la consulta y los embeddings de *todas las categorías* (1053 en total) para obtener un Top 20 de candidatos.
    - **Componente Léxico:** Utiliza `spaCy` para lematizar la consulta y los 20 candidatos. Calcula:
        - `calculate_jaccard_similarity`: Similitud de tokens compartidos.
        - `calculate_key_entity_match` (KEM): Recompensa la coincidencia de entidades clave extraídas de la consulta.
        - `concision_score`: Penaliza categorías con demasiados tokens (ej. `1 / sqrt(num_tokens)`).
    - **Ponderación:** Una fórmula combina estos puntajes con pesos definidos (`COSINE_WEIGHT`, `JACCARD_WEIGHT`, `KEM_WEIGHT`, `CONCISION_WEIGHT`) para producir un `hybrid_score` final. La categoría con el puntaje más alto es seleccionada.
- **Recuperador (Retriever):**
    - **Codificador:** `sentence-transformers/all-mpnet-base-v2`.
    - **Índice:** `ChromaDB` (cliente persistente).
- **Generador (Generator):**
    - **LLM:** `llama3.1:8b`.
    - **Servidor:** `Ollama`.
- **Orquestador (Script de Python):**
    - Maneja el flujo: llama a `classify_dynamically` para obtener el filtro, consulta `ChromaDB` con el filtro (`where=...`), itera sobre los N=3 resultados, y orquesta las llamadas al LLM para la extracción (`PROMPT_EXTRACCION_...`) y generación (`PROMPT_REPORTE_...`) de cada informe.

3.5. Criterios y Métricas de Evaluación

- **Métrica de Clasificación/Recuperación:**
    - **Métrica Propuesta:** Precisión@k (Precision@k) / Tasa de Aciertos (Hit Rate).
    - **Justificación:** Mide si el documento "dorado" (ideal) se encuentra dentro de los k=3 documentos recuperados tras el filtrado.
- **Métrica de Generación:**
    - **Métrica Propuesta:** Evaluación Cualitativa Humana.
    - **Justificación:** La métrica más importante es la calidad, coherencia y fidelidad del informe final. Un humano debe calificar los informes generados en una escala (ej. 1-5) según su coherencia, fidelidad (que no alucine hechos) y utilidad.

3.6. Herramientas y Tecnologías

- **Núcleo de IA:** `torch`, `transformers`, `spacy`.
- **Base de Datos Vectorial:** `chromadb`.

- **Procesamiento de Datos:** `pandas`, `numpy`, `pyarrow`, `pyarrow.parquet`.
- **Lectura de PDF:** `PyPDF2`.
- **Manejo de Archivos:** `os`, `glob`, `re`, `math`.
- **Inferencia LLM:** `requests`.
- **Paralelización:** `concurrent.futures.ProcessPoolExecutor`.

---

4. Implementación

4.1. Fase 1: Procesamiento de PDFs y Paralelización

La función `fase1_procesar_pdfs` orquesta esta fase. Utiliza `glob` para encontrar todos los PDFs y los divide en lotes. De forma crucial, utiliza `concurrent.futures.ProcessPoolExecutor` para distribuir el procesamiento de los PDFs de un lote entre todos los núcleos de CPU disponibles. Cada proceso hijo llama a `process_single_pdf`, que aplica la lógica de extracción (`extract_text_from_pdf`), limpieza (`extract_and_clean_legal_text`) y parseo de metadatos (`parse_metadata_from_filename`). Los resultados limpios se guardan en formato `.parquet`.

4.2. Fase 2: Indexación en la Base de Datos Vectorial

La función `fase2_indexar_rag` crea el índice vectorial.

1. **Conexión a DB:** Se inicializa `chromadb.PersistentClient` y se elimina cualquier colección existente para garantizar una indexación limpia.
2. **Carga de Modelo:** Se llama a `get_embedding_model_and_tokenizer`, que detecta y mueve el modelo de embeddings a la GPU (CUDA) si está disponible.
3. **Generación de Embeddings:** Lee cada lote `.parquet`. Para gestionar la memoria de la GPU, la función `generate_embeddings_in_sub_batches` divide los textos en sub-lotes (`INDEXING_BATCH_SIZE = 50`) para la codificación.
4. **Indexación:** Los `embeddings`, `documents` (texto), `metadatas` (ej. `'materia_principal': 'penal'`) e `ids` se cargan en ChromaDB usando `collection.add()`.

4.3. Fase 3: Inferencia (Clasificación Híbrida y Generación Múltiple)

Este es el núcleo del co-piloto, orquestado por el bloque `if __name__ == "__main__":`.

1. **Carga de Caché:** El sistema primero intenta cargar un caché de categorías y sus embeddings (`CACHE_FILE_CATEGORIES`, `CACHE_FILE_EMBEDDINGS`). Si no existe, lo genera llamando a `get_real_categories_from_db` y `generate_embeddings` sobre las 1053 categorías únicas encontradas.
2. **Clasificación Híbrida:** Se llama a `classify_dynamically` con la consulta del usuario. Esta función realiza la búsqueda semántica (Coseno) contra el caché de categorías (Top 20), y luego aplica el re-ranking numérico (Jaccard, KEM, etc.) para seleccionar la mejor categoría (ej. `'PENAL robo-calificado-y-privacion-ilegal-de-la-libertad'`).
3. **Recuperación Filtrada:** Se llama a `generate_multianalysis_report_from_rag`. Esta función ejecuta `collection.query()` usando el embedding de la consulta, `n_results=10`, y, crucialmente, `where={"delito_o_accion": categoria_seleccionada}`.
4. **Búsqueda Adaptativa:** Si la búsqueda filtrada devuelve menos de 3 resultados, el sistema "relaja" el filtro y vuelve a ejecutar la consulta sin la cláusula `where`, basándose únicamente en la similitud semántica para encontrar precedentes.
5. **Bucle de Generación:** El sistema itera sobre los precedentes encontrados (hasta 3 válidos). Para cada uno:
    - Determina la materia (ej. 'PENAL').
    - Llama al LLM con el `PROMPT_EXTRACCION_PENAL` para obtener hechos estructurados.
    - Realiza un QC (`FAILURE_KEYWORDS`) para descartar fallos del LLM.
    - Llama al LLM con el `PROMPT_REPORTE_PENAL` usando los hechos extraídos.
    - Realiza otro QC y, si es válido, añade el informe a la lista `valid_reports`.
6. **Consolidación:** Los 3 informes válidos se unen con un separador y se presentan al usuario.

---

5. Resultados

5.1. Consulta de Ejemplo

Se utilizó la siguiente consulta de prueba para simular un nuevo caso penal:

> "Se acusa a un individuo de robo calificado por haber sustraído con violencia un teléfono móvil en la vía pública durante la noche. La víctima lo identificó plenamente y existen cámaras de seguridad que registraron el hecho."

5.2. Resultado del Modelo (Análisis de Precedentes Generado)

Al ejecutar el script (Fase 3) con la consulta anterior, el sistema generó la siguiente salida:

Markdown
Resultados del Co-piloto Jurídico
Consulta Analizada: > Se acusa a un individuo de robo calificado por haber sustraído con violencia un teléfono móvil en la vía pública durante la noche. La víctima lo identificó plenamente y existen cámaras de seguridad que registraron el hecho.
Categoría Final para Filtro: `PENAL robo-calificado-y-privacion-ilegal-de-la-libertad`
Tiempo de Procesamiento: `124.54 segundos`
***
📚 Análisis Detallado de Precedentes

📜 Análisis del Precedente 1/3
***

📊 Metadatos y Coincidencia
* **ID del Documento:**
`2023-06-29-134149_penal_robo-calificado-y-privacion-ilegal-de-la-libertad.pdf`
* **Materia Principal:** `PENAL`
* **Acción/Delito Indexado:** `robo-calificado-y-privacion-ilegal-de-la-libertad`
* **Distancia Semántica:** `5.5663`
* **Fuente del Contenido:** Filtro Estricto

---
informe de análisis penal
1. resumen ejecutivo
el imputado fue vinculado a proceso por un delito de privación ilegítima de la libertad, y el tribunal decidió mantenerlo en prisión preventiva debido a que consideró que no había medidas cautelares suficientes para garantizar su comparecencia y proteger a la víctima. la decisión fue proporcional, adecuada y respetuosa de derechos humanos.

2. datos de identificación del proceso
- tribunal: [no se proporciona el nombre específico del tribunal en los hechos extraídos]
- expediente: [no se proporciona el número de expediente en los hechos extraídos]
- etapa procesal: vinculación a proceso por un delito de privación ilegítima de la libertad
- delito(s): privación ilegítima de la libertad

3. partes involucradas
- acusado(s): el imputado (no se proporciona el nombre específico)
- víctima(s)/ofendido(s): [no se menciona a una víctima o ofendido en los hechos extraídos]

4. análisis de hechos y decisión
hechos clave:
el imputado fue vinculado a proceso por un delito de privación ilegítima de la libertad, y la defensa solicitó medidas cautelares diversas a la prisión preventiva. sin embargo, el ministerio

público y el juez consideraron que estas medidas no eran suficientes para garantizar la comparecencia del imputado y proteger a la víctima.

decisión del tribunal:
la decisión del tribunal fue mantener al imputado en prisión preventiva debido a la necesidad de proteger a la víctima.

razonamiento del tribunal (análisis):
el tribunal fundamentó su decisión en el análisis de la petición ministerial y lo debatido por la defensa. consideraron que la prisión preventiva justificada era necesaria para garantizar la comparecencia del imputado y proteger a la víctima, lo cual se ajusta a los principios de proporcionalidad y adecuación en el marco legal penal. la decisión fue respetuosa con los derechos humanos, ya que buscó equilibrar las necesidades de seguridad pública con las garantías procesales del imputado.

aplicación legal:
la aplicación de la ley se centró en la interpretación y aplicación de los artículos legales relacionados con la prisión preventiva y las medidas cautelares. el tribunal debió considerar el grado de riesgo que representaba el imputado para la víctima, así como la necesidad de garantizar su comparecencia en el proceso penal.

conclusión:
la decisión del tribunal fue razonada y ajustada a los principios legales. la prisión preventiva se aplicó de manera proporcional y adecuada para proteger a la víctima, lo cual es un aspecto fundamental en el marco legal penal.

📜 Análisis del Precedente 2/3
***

📊 Metadatos y Coincidencia
* **ID del Documento:**
`2025-06-30-141336_penal_robo-calificado-y-privacion-ilegal-de-la-libertad.pdf`
* **Materia Principal:** `PENAL`
* **Acción/Delito Indexado:** `robo-calificado-y-privacion-ilegal-de-la-libertad`
* **Distancia Semántica:** `7.1502`
* **Fuente del Contenido:** Filtro Estricto

---
informe de análisis penal
1. resumen ejecutivo
el delito principal fue robo calificado, previsto por el numeral 233 en relación a 236 fracciones i, vii y ix, del código penal del estado de jalisco, así como la privación ilegal de la libertad, previsto

por el numeral 193 del mismo código. la decisión final del tribunal fue revocar la resolución emitida en audiencia de revisión de medidas cautelares dictada por el juez décimo de control.

2. datos de identificación del proceso
- tribunal: juez décimo de control.
- expediente: no se proporciona el número de expediente.
- etapa procesal: audiencia de revisión de medidas cautelares.
- delito(s): robo calificado y privación ilegal de la libertad.

3. partes involucradas
- acusado(s): [nombre del imputado] (no se proporciona el nombre completo) y [nombre del imputado] (no se proporciona el nombre completo).
- víctima(s)/ofendido(s): no se proporciona la información de la víctima.

4. análisis de hechos y decisión
hechos clave:
se revocó la resolución emitida en audiencia de revisión de medidas cautelares dictada por el juez décimo de control, lo que implica que se reconsideraron las medidas cautelares aplicadas a los imputados. la causa corresponde a [no.17]eliminado_nombre_del_imputado[97] y [no.18]eliminado_nombre_del_imputado[97], quienes se les imputa el delito de robo calificado, previsto por el numeral 233 en relación a 236 fracciones i, vii y ix, del código penal del estado de jalisco.

decisión del tribunal:
la medida cautelar de prisión preventiva oficiosa subsiste en contra de los imputados. se instruyó al juez de control que lleve a cabo las gestiones necesarias para dar cumplimiento a la resolución dictada por el órgano colegiado.

razonamiento del tribunal (análisis):
la decisión del tribunal se basa en la consideración de que los imputados están involucrados en un delito grave, como es el robo calificado y la privación ilegal de la libertad. la medida cautelar de prisión preventiva oficiosa se mantiene debido a la gravedad de los delitos y la necesidad de asegurar la presencia de los imputados en el proceso penal.

la resolución emitida por el juez décimo de control fue revocada, lo que implica que se reconsideraron las medidas cautelares aplicadas. esto sugiere que el tribunal consideró que la decisión anterior no era justa o necesaria, y que se requirió una nueva evaluación de la situación.

en cuanto a los fundamentos legales, se aplicaron los artículos 233 y 236 del código penal del estado de jalisco, que establecen las penas para el robo calificado. además, se consideró el artículo 193, que prevé la pena por privación ilegal de la libertad.

en resumen, la decisión del tribunal se basa en la gravedad de los delitos y la necesidad de asegurar la presencia de los imputados en el proceso penal. la revocación de la resolución emitida por el juez décimo de control sugiere que se requirió una nueva evaluación de la situación, considerando las pruebas y fundamentos legales aplicables.

📜 Análisis del Precedente 3/3
***

📊 Metadatos y Coincidencia
* **ID del Documento:**
`2024-01-08-103634_penal_robo-calificado-y-privacion-ilegal-de-la-libertad.pdf`
* **Materia Principal:** `PENAL`
* **Acción/Delito Indexado:** `robo-calificado-y-privacion-ilegal-de-la-libertad`
* **Distancia Semántica:** `7.1842`
* **Fuente del Contenido:** Filtro Estricto

---
informe de análisis penal
1. resumen ejecutivo
la sentencia judicial analizada ordena anotar el acta de nacimiento de una persona con un nombre diferente al registrado originalmente, ajustando así el documento a la verdadera realidad social y protegiendo la estabilidad y bienestar de la familia.

2. datos de identificación del proceso
- tribunal: tribunal superior
- expediente: no se proporciona número de expediente en los hechos extraídos.
- etapa procesal: revisión de oficio
- delito(s): anotación de acta de nacimiento con nombre diferente al registrado originalmente.

3. partes involucradas
- acusado(s): no se menciona a un acusado en los hechos extraídos.
- víctima(s)/ofendido(s): no se menciona a una víctima o ofendido en los hechos extraídos.

4. análisis de hechos y decisión
hechos clave:
la demandante probó su acción de anotación sin oposición de los demandados, lo que llevó al tribunal a ordenar el cambio del nombre en el acta de nacimiento. el cambio se justificó porque la registrada es conocida en diferentes ámbitos de su vida con el nuevo nombre y se trata de la misma persona.

decisión del tribunal:

se ordenó anotar el acta de nacimiento con un nombre diferente al registrado originalmente, ajustando así el documento a la verdadera realidad social.

razonamiento del tribunal (análisis):
la decisión del tribunal se basó en la necesidad de proteger la estabilidad y bienestar de la familia. el cambio del nombre no implica actuar de mala fe ni se contraría la moral, ya que el objetivo es ajustar el documento a la realidad social actual. la prueba más importante fue la acción de anotación presentada por la demandante sin oposición de los demandados, lo que demostró la necesidad del cambio.

la aplicación de las normas jurídicas se basó en el código civil para el distrito federal (artículo 135, fracción ii) y el código de procedimientos civiles (numerales 83, 85, 89d, 419 y 457). estas normas permitieron al tribunal ajustar el acta de nacimiento a la realidad social actual, protegiendo así la estabilidad y bienestar de la familia.

en conclusión, la sentencia judicial analizada tiene como objetivo proteger la estabilidad y bienestar de la familia, ajustando el acta de nacimiento a la verdadera realidad social. la decisión del tribunal se basó en la necesidad de adaptar el documento a la realidad actual, sin implicar mala fe ni contradecir la moral.

5.3. Comparación con los Objetivos

- **Objetivo 1 (Pipeline PDFs Paralelizado):** Cumplido. La Fase 1 procesó los PDFs en lotes usando `ProcessPoolExecutor`.
- **Objetivo 2 (Clasificación Híbrida):** Cumplido. La función `classify_dynamically` implementó el re-ranking numérico (Coseno+Jaccard+KEM) y seleccionó una categoría (`PENAL robo-calificado...`) para filtrar.
- **Objetivo 3 (RAG N=3 Filtrado):** Cumplido. El sistema usó la categoría para filtrar y recuperó 3 precedentes, generando un informe para cada uno.
- **Objetivo 4 (Prompts Dinámicos):** Cumplido. El sistema seleccionó los prompts `PROMPT_EXTRACCION_PENAL` y `PROMPT_REPORTE_PENAL` basados en la materia del precedente.
- **Objetivo 5 (GPU):** Cumplido. La salida del script confirma: "Modelo en dispositivo: cuda".

---

6. Análisis y Discusión

6.1. Interpretación de los Resultados

El resultado de la consulta de ejemplo (`robo calificado`) es revelador.

1. **Éxito en Clasificación:** El clasificador híbrido funcionó excelentemente. Ante una consulta sobre "robo calificado con violencia", seleccionó `PENAL robo-calificado-y-privacion-ilegal-de-la-libertad` como la mejor categoría-filtro. Esto es muy relevante, ya que la "privación ilegal de la libertad" es un delito común asociado a robos con violencia (ej. secuestro exprés, amagar).
2. **Éxito en Recuperación (Parcial):** Los Precedentes 1 y 2 fueron altamente relevantes. Ambos trataban sobre "robo calificado" y "privación ilegal de la libertad", coincidiendo perfectamente con el filtro.
3. **Fallo en Recuperación (Precedente 3):** El Precedente 3 es un *fallo claro y evidente* del sistema. A pesar de tener los metadatos `PENAL` y `robo-calificado-y-privacion-ilegal-de-la-libertad`, el contenido real del documento (y el informe generado por el LLM) trata sobre una "anotación de acta de nacimiento".

Conclusión del Análisis: Esto demuestra que la *Fuente del Contenido* (el documento PDF) tenía metadatos incorrectos en su nombre de archivo. El Precedente 3 estaba mal etiquetado en el dataset original. El sistema RAG *funcionó perfectamente* según sus instrucciones (filtró por la etiqueta correcta y encontró el documento), pero el resultado fue incorrecto debido a datos de entrada erróneos ("basura entra, basura sale").

6.2. Evaluación de Éxito y Limitaciones del Modelo

Puntos Exitosos:

- **Inteligencia en la Recuperación:** El pipeline "Clasificar-Híbrido-luego-Recuperar-Filtrado" es un éxito. Es mucho más rápido que usar un LLM para clasificar en cada consulta y más preciso que la búsqueda semántica simple.
- **Análisis Comparativo:** Proporcionar un análisis detallado de 3 precedentes ofrece un valor inmenso al usuario, permitiéndole ver matices.
- **Robustez del QC:** El sistema detectó y descartó fallos del LLM durante la generación de informes (se observa en los logs de la consola que se descartaron precedentes fallidos antes de encontrar 3 válidos).

Limitaciones Identificadas:

- **Dependencia de Metadatos (Crítica):** Como demostró el Precedente 3, la mayor limitación del sistema es su total dependencia de que la *nomenclatura del archivo PDF sea correcta*.
- **Afinación de Pesos Híbridos:** Los pesos (`JACCARD_WEIGHT = 0.40`, etc.) se establecieron heurísticamente. Podrían no ser óptimos para todas las consultas.
- **Falta de Síntesis:** El sistema presenta 3 informes separados, pero no genera un cuarto resumen que los *compare* y *sintetice*.

6.3. Propuestas de Mejora y Trabajo Futuro

1. **Mejora de Metadatos (Prioridad 1):** Eliminar la dependencia de la nomenclatura de archivos. La Fase 1 debería usar un LLM con el `PROMPT_CLASIFICAR_MATERIA` y `PROMPT_EXTRACCION_ENTIDADES` *durante la ingesta* para generar y guardar metadatos fiables basados en el *contenido real* del PDF, no en su nombre.
2. **Síntesis de Múltiples Documentos:** Añadir una "Etapa 4" a la inferencia. Después de generar los 3 informes, enviar esos 3 informes a un prompt final de síntesis: *"Compara estos 3 precedentes: ¿cuál es el más relevante para la consulta del usuario y por qué? ¿Existen contradicciones entre ellos?"*.
3. **Implementación de Evaluación Rigurosa:** Construir un "golden set" (conjunto de datos de prueba) para implementar las métricas MRR y ROUGE-L definidas en la metodología.
4. **Preprocesamiento Avanzado:** Integrar OCR (ej. Tesseract) para manejar PDFs escaneados.

6.4. Escenarios de Aplicación

- **Despachos de Abogados:** Para que los abogados encuentren rápidamente precedentes relevantes y preparen borradores de argumentos.
- **Departamentos Jurídicos de Empresas:** Para analizar litigios pasados y evaluar riesgos en contratos actuales.
- **Juzgados y Tribunales:** Para que los jueces y secretarios encuentren sentencias previas y aseguren la coherencia en sus fallos.
- **Academia:** Para que estudiantes e investigadores analicen grandes volúmenes de jurisprudencia.

---

7. Conclusiones

7.1. Resumen de Resultados

El proyecto implementó exitosamente un pipeline RAG avanzado que utiliza un re-ranking híbrido-numérico para clasificar consultas. El sistema es capaz de realizar una búsqueda semántica filtrada por metadatos, recuperar 3 precedentes relevantes y generar informes analíticos para cada uno, todo en un entorno local y privado que garantiza la confidencialidad de los datos.

7.2. Cumplimiento de Objetivos

Todos los objetivos principales fueron cumplidos. Se desarrolló un pipeline de ingesta paralelizado, se implementó el sistema RAG con clasificación híbrida-numérica, se configuró el análisis múltiple (N=3) y se optimizó el uso de GPU.

7.3. Impacto Profesional

El desarrollo de este proyecto representó un desafío significativo que consolidó la aplicación de los conocimientos del Diplomado. Se enfrentaron problemas reales como la limpieza de datos (Regex), la orquestación de múltiples modelos (Embeddings + LLM) y el diseño de arquitecturas de recuperación inteligentes (re-ranking híbrido y filtrado). Este proyecto proporciona una comprensión profunda de las arquitecturas RAG avanzadas, fundamentales en la industria actual, y da la confianza para diseñar soluciones de IA complejas, privadas y eficientes.

---

8. Referencias Bibliográficas

- ChromaDB. (2023). *Chroma - The AI-native open-source embedding database*. Recuperado de https://www.trychroma.com/
- Hugging Face. (s.f.). *sentence-transformers/all-mpnet-base-v2*. Recuperado de https://huggingface.co/sentence-transformers/all-mpnet-base-v2
- Lewis, P., Perez, E., Piktus, A., et al. (2020). *Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks*. Advances in Neural Information Processing Systems, 33.
- Ollama. (2023). *Ollama: Get up and running with large language models locally*. Recuperado de https://ollama.com/
- Reimers, N., & Gurevych, I. (2019). *Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks*. Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing.
- Touvron, H., Martin, L., Stone, K., et al. (2023). *Llama 2: Open Foundation and Fine-Tuned Chat Models*.

---

9. Anexos

9.1. Código Fuente Principal (Script v9.8)