

Static Inference Architecture for Data Traceability in Legacy Natural/Adabas Ecosystems: An Entity Disambiguation Approach

Ing. Julián Ignacio Perdiguero¹ e Ing. Pedro Osvaldo Falabella²

¹ Facultad de Informática – Universidad Nacional de La Plata, Maestría en Ingeniería de Software. ing.julian.perdiguero@gmail.com

² Universidad Tecnológica Nacional – Facultad Regional La Plata - DISI Departamento de Ingeniería en Sistemas – LINSI. ofalabel@frlp.utn.edu.ar

Abstract. This paper addresses the opacity in data traceability within large-scale critical systems developed in Natural with Adabas databases. The lack of updated technical documentation and the extensive use of length-restricted technical aliases (DDM) hinder the identification of the real impact of business processes on data entities. To solve this challenge, a Reverse Engineering methodology based on a four-phase Static Code Analysis engine is proposed: Semantic Sanitization, Metadata Normalization, CRUD Impact Inference, and Relational Persistence. The results demonstrate that using a dictionary-based analysis engine eliminates the margin of error due to false positives by 30% and allows the identification of critical write processes in master tables that previously remained hidden behind technical aliases. This methodology provides a reliable and timely roadmap for risk mitigation in migration, refactoring, and regulatory compliance projects in mission-critical environments.

Keywords: Legacy Systems, Reverse Engineering, Static Analysis, Data Traceability, Natural/Adabas.

Arquitectura de Inferencia Estática para la Trazabilidad de Datos en Ecosistemas Legacy Natural/Adabas: Un Enfoque de Desambiguación de Entidades

Resumen. Este trabajo aborda la opacidad en la trazabilidad de datos dentro de sistemas críticos de gran escala desarrollados en Natural con bases de datos Adabas. La erosión de la documentación técnica y el uso extensivo de alias técnicos con restricciones de longitud (DDM) generan riesgos significativos en la identificación de procesos de negocio. Para resolver este desafío, se propone una metodología de Ingeniería Inversa determinística basada en un motor de Análisis Estático de Código de cuatro fases: Sanitización Semántica, Normalización de Metadatos, Inferencia de Impacto CRUD y Persistencia Relacional. A diferencia de los enfoques puramente probabilísticos basados en IA, esta metodología utiliza un motor de curación basado en diccionarios (M.I.T.D.) que elimina el margen de error en entornos de alta criticidad normativa. Los resultados demuestran una reducción del 30% en falsos positivos y permiten identificar procesos de escritura críticos en tablas maestras que anteriormente permanecían ocultos tras alias técnicos. Este marco de trabajo proporciona una hoja de ruta confiable y precisa para la mitigación de riesgos en proyectos de migración, refactorización y cumplimiento regulatorio en los sectores bancario y de seguros, cerrando la brecha entre la complejidad legacy y los estándares arquitectónicos modernos.

Palabras clave: Modernización de Software, Sistemas Legacy, Natural/Adabas, Análisis Estático de Código, Recuperación de Lógica de Negocio, Curación Semántica.

1 Introducción

La modernización de sistemas de gran escala en entornos Natural/Adabas constituye un desafío crítico debido a la acumulación de deuda técnica y la degradación de los artefactos de diseño originales (Cunningham, 1992; Trudel et al., 2024). Natural es un lenguaje procedural fuertemente acoplado a la base de datos no relacional Adabas, cuya restricción histórica de 8 caracteres para variables y módulos (DDMs) fuerza el uso de alias crípticos (e.g., SINIES01 en lugar de SINIESTROS). Ante la obsolescencia o inexistencia de documentación formal, el código fuente permanece como la única fuente de verdad fiable. Esta opacidad estructural compromete la trazabilidad de datos y la identificación precisa del impacto de los procesos, degradando la mantenibilidad y analizabilidad bajo el estándar ISO/IEC 25010 (ISO/IEC, 2023), lo que eleva el riesgo de regresión en migraciones hacia microservicios (Böhm et al., 2026; Patsnap R&D, 2026). Asimismo, la auditoría transaccional mediante la interpretación manual de expertos carece de la escalabilidad necesaria para

abordar entornos que superan los millones de líneas de código en sectores de alta complejidad como la banca, seguros y administraciones públicas.

Para resolver esta brecha, se requiere la automatización mediante Análisis Estático de Código (SCA) integrado con modelos de inteligencia artificial y capas de curación semántica (Böhm et al., 2026; Patsnap R&D, 2026). El presente trabajo detalla el desarrollo e implementación de la Arquitectura de Inferencia Estática y Trazabilidad de Datos (M.I.T.D.), una metodología de ingeniería inversa diseñada para la desambiguación de entidades y la recuperación de lógica de negocio en entornos de misión crítica. El framework propuesto culmina con la persistencia en una base de datos PostgreSQL, consolidando estructuras relacionales y reglas de negocio enriquecidas que devuelven el contexto a los datos y facilitan su consumo por parte de arquitecturas de servicios modernas (Chalmers University of Technology, 2021; Making Sense Tech, 2026).

2 Estado del arte y lecciones de la investigación

Mediante un Mapeo Sistemático de la Literatura (SMS), se constató que la modernización de sistemas legacy evoluciona hacia enfoques basados en modelos (Böhm et al., 2026), aunque la intersección de lenguajes 4GL y bases de datos no relacionales presenta desafíos de opacidad estructural aún no resueltos. Limitaciones del SCA y Acoplamiento: Las herramientas de Análisis Estático Tradicional (como SonarQube) operan en una capa puramente sintáctica (Code Smells) y carecen de inferencia semántica para desambiguar alias técnicos en módulos DDM (Hirschi & Spurr, 2021). Asimismo, ontologías como KDM o técnicas de Program Slicing resultan insuficientes en ecosistemas Natural/Adabas, donde la lógica de negocio se encuentra "secuestrada" por la estructura física de la base de datos jerárquica. Esta desconexión entre el flujo de control y la semántica imposibilita la reconstrucción del diccionario de datos sin una intervención manual masiva.

Evaluación Crítica de LLMs: Si bien los Modelos de Lenguaje de Gran Escala (LLMs) revolucionan la explicación de código, su aplicación directa en este contexto reveló dos limitaciones críticas: la inconsistencia probabilística, cuyo riesgo de "alucinaciones" semánticas en la interpretación de los DDM es inaceptable en sectores regulados (Banca/Seguros) (ZTE Communications, 2026); y la miopía de contexto global, dado que los LLMs pierden la trazabilidad de las entidades al rastrearlas en grafos de llamadas (call graphs) masivos .

Análisis de Brecha y Enfoque M.I.T.D.: Ante la falta de soluciones automatizadas con tendencia a "error cero", M.I.T.D. aborda este vacío mediante una arquitectura híbrida. Al integrar metadatos de negocio con el Árbol de Sintaxis Abstracta (AST) bajo un enfoque estrictamente determinístico (basado en reglas y diccionarios de curación), M.I.T.D. establece una salvaguarda estructural frente a la naturaleza probabilística de la IA. Este paradigma garantiza que la persistencia relacional en PostgreSQL sea una representación

auditable y fiel a la lógica original (Making Sense Tech, 2026; (ZTE Communications, 2026), relegando el uso de la IA exclusivamente a la humanización y documentación final del código.

3 Marco teórico

De la Entropía del Software a la Inferencia Determinística: el marco conceptual de esta investigación se sustenta en la intersección de la evolución de sistemas heredados, la gestión de deuda técnica y los paradigmas emergentes de inteligencia artificial aplicada a la ingeniería de software (Cunningham, 1992; Trudel et al., 2024).

Evolución y Mantenibilidad de Sistemas Heredados: históricamente, la modernización de sistemas legacy ha estado supeditada a procesos de relevamiento manual de alta complejidad. Este fenómeno se explica a través de las Leyes de Evolución del Software de Lehman (1980) (Cunningham, 1992), particularmente la Ley de Cambio Continuo y la Ley de Complejidad Creciente. Autores clásicos han documentado que el mantenimiento de software no es una etapa final, sino una actividad que consume entre el 60% y el 80% del ciclo de vida de un sistema (Patsnap R&D, 2026). En entornos de gran escala, este esfuerzo se ve agravado por la Deuda Técnica, definida por Cunningham (1992) como el costo acumulado de decisiones de diseño subóptimas que prioriza la entrega inmediata sobre la mantenibilidad a largo plazo (Cunningham, 1992; Making Sense Tech, 2026; Trudel et al., 2024).

3.2 Restricciones Sintácticas y Opacidad en Natural/Adabas: el desafío técnico específico de los ecosistemas Natural/Adabas radica en la arquitectura de sus Data Definition Modules (DDM). A diferencia de los esquemas relacionales modernos que fomentan la semántica descriptiva, los DDMs presentan limitaciones sintácticas (restricción de caracteres y alias técnicos) que imponen una capa de opacidad entre la persistencia física y la lógica de negocio (Böhm et al., 2026). Esta desconexión genera una Entropía del Software elevada, donde el código fuente se convierte en la única "fuente de verdad" fiable, desplazando a la documentación técnica obsoleta y requiriendo técnicas avanzadas de recuperación de diseño e identificación de componentes (Hirschi & Spurr, 2021; ISO/IEC, 2023).

3.3 Hibridación Técnica frente a la Entropía del Software: la literatura contemporánea en ingeniería de software subraya una dicotomía crítica al abordar el desgaste estructural de ecosistemas *legacy* de más de 30 años de antigüedad. Este fenómeno de desorden natural se define como Entropía del Software, el cual se ve agravado por el incremento constante de la Deuda Técnica —término acuñado por Cunningham (1992)— y obedece a las Leyes de Evolución del Software de Lehman (Lehman, 1980; Cunningham, 1992). Para hacer frente a esta entropía, han surgido los Modelos de Inteligencia Artificial Generativa (GenAI), los cuales ofrecen una capacidad disruptiva para la síntesis de código. Sin embargo, su naturaleza probabilística introduce riesgos

de alucinaciones algorítmicas inaceptables en sistemas de misión crítica (ZTE Communications, 2026).

Para abordar el problema estructural y mitigar simultáneamente el riesgo de la IA, la arquitectura propuesta (M.I.T.D.) implementa un paradigma de Hibridación Técnica. Este enfoque establece una clara división de responsabilidades que consta de dos niveles:

- **Capa Determinística (SCA):** Actúa como la herramienta de precisión estructural. Antes de la intervención de modelos generativos, es imperativo aplicar enfoques determinísticos de Análisis Estático de Código para el desacoplamiento de la lógica de negocio. Esto extrae la "verdad inmutable" del código y garantiza una base de conocimiento sólida y verificable basada en el Árbol de Sintaxis Abstracta (AST).
- **Capa Generativa (LLM):** Actúa como la herramienta de humanización. Una vez extraída la estructura técnica mediante el motor determinístico de desambiguación de entidades, se emplean modelos de IA en fases estrictamente posteriores para la redacción y documentación semántica (Chalmers University of Technology, 2021; ZTE Communications, 2026).

Esta hibridación justifica por qué el framework M.I.T.D. es una solución superior y obligatoria en sectores donde el error no es una opción (como entidades bancarias y de seguros). Al limitar el uso de la IA a la capa generativa, se neutralizan las alucinaciones y se garantiza que el resultado final entregado sea técnica, legal y funcionalmente preciso (ZTE Communications, 2026; Patsnap R&D, 2026).

4 Metodología

El estudio se fundamenta en el principio arquitectónico de "Saneamiento antes que Inferencia", implementando un motor de análisis estático (desarrollado en Python) orientado a la creación de un Gemelo Digital de la Lógica de Negocio (ZTE Communications, 2026; Patsnap R&D, 2026). Esta metodología se estructura en cuatro fases industriales automatizadas que transforman el código fuente no estructurado en una base de conocimientos determinística.

Fase I: Ingesta y Sanitización de Activos. El software realiza la ingesta del código fuente en bruto y se aplica un procedimiento de sanitización profunda mediante expresiones regulares (Regex) (ZTE Communications, 2026). Esta fase elimina artefactos de ruido (comentarios históricos, bloques de código sin uso) y normaliza el texto a mayúsculas. El motor opera sobre una versión "en memoria", garantizando que el análisis se realice sobre una representación limpia y estandarizada del sistema (Trudel et al., 2024). Las expresiones regulares fueron diseñadas específicamente para la sintaxis de Natural. Asimismo, dado que el lenguaje Natural es 'case-insensitive' (no distingue mayúsculas de minúsculas en su compilación), la normalización de todo el

código a mayúsculas simplifica el árbol de sintaxis y la búsqueda de patrones determinísticos sin causar ninguna pérdida en la semántica de programación.

Fase II: Resolución de Alias y Desambiguación. La construcción del Diccionario Maestro (ddm.json) es un proceso híbrido. En una primera instancia, se extraen automáticamente los metadatos disponibles de los DDM. Luego, se utilizan técnicas de IA para sugerir nombres de dominio completos basados en el contexto del código. Finalmente, estos diccionarios son validados manualmente por expertos del negocio para crear una fuente de verdad inmutable. Una vez validado, el diccionario actúa de forma automática y determinística para el resto del análisis.

Este componente actúa como el núcleo de resolución semántica (Böhm et al., 2026), permitiendo que el sistema traduzca instantáneamente etiquetas crípticas (ej. SINIES01) en entidades comprensibles para el dominio (ej. SINIESTROS), facilitando la comprensión de conceptos (Hirschi & Spurk, 2021).

Fase III: Inferencia de Impacto CRUD y Análisis de Dependencias. El algoritmo de inferencia procesa el código sanitizado buscando patrones de interacción [VERBO] + [ENTIDAD]. El sistema identifica y categoriza automáticamente: Operaciones de Lectura: Comandos READ, FIND y sus variantes. Operaciones de Escritura: Comandos de alta criticidad como STORE o UPDATE. Un ejemplo puntual podría ser:

Patrón: = VERBO + ENTIDAD | ENTIDAD: = resolver (alias, diccionario)

Formalización del Problema de Inferencia

El problema abordado en esta investigación puede definirse formalmente como un proceso de inferencia determinística sobre código fuente legacy.

Sea:

P el conjunto de programas del sistema legacy; **A** el conjunto de alias técnicos definidos en los DDM; **E** el conjunto de entidades de negocio

f: $A \rightarrow E$ una función de resolución semántica determinística

El objetivo es inferir un conjunto de relaciones: $R \subseteq P \times E \times T$

Donde $T = \{\text{CREATE, READ, UPDATE, DELETE}\}$ representa el tipo de interacción (CRUD) y cada elemento de R modela una interacción semánticamente resuelta entre un programa y una entidad de negocio, enriquecida con su tipo de operación.

Este enfoque permite transformar un espacio de código opaco basado en alias técnicos en un modelo explícito, estructurado y auditable de trazabilidad de datos.

Caso de Estudio: Archivo PAGOS

Como validación empírica del modelo, se analizó la propagación del campo PAG-IMPORTE.

Cabe destacar que el caso de estudio PAGOS corresponde a un módulo transaccional real, debidamente anonimizado, el cual fue extraído del ecosistema Core de una entidad financiera participante del presente estudio. Este campo, por su alta criticidad en procesos contables y regulatorios, se utilizó como punto de entrada para evaluar la capacidad de la arquitectura M.I.T.D. al reconstruir relaciones de impacto complejas.

La validación metodológica de los resultados se realizó formalmente mediante la técnica de Juicio de Expertos (Expert Review). Para ello, se expusieron los grafos de trazabilidad generados automáticamente por el motor ante un panel de arquitectos funcionales que cuentan con más de 15 años de experiencia en el sistema original. Durante esta fase, se cotejaron minuciosamente las salidas automatizadas de la herramienta contra casos de prueba de migración manual históricamente conocidos por la organización, permitiendo medir con precisión el comportamiento del motor frente al criterio experto.

Detección de Impacto: El motor de inferencia permitió la identificación automática de todos los objetos que interactúan con el campo analizado, incluyendo: **Programas, Subrutinas, Mapas**

Este proceso se realizó sin intervención manual, garantizando escalabilidad y reproducibilidad.

Trazabilidad Estructurada (JSON)

El sistema generó artefactos estructurados que capturan:

- Línea exacta de código, tipo de operación (CRUD) y Contexto de ejecución

Ejemplo representativo: Resolución semántica

```
{
  "field": "PAG-IMPORTE",
  "alias_resolved": "Importe_Liquidacion_Final", "impact_points": [
    { "object": "PRG-LIQ-01", "line": 452, "action": "UPDATE" },
    { "object": "SUB-PAG-VALID", "line": 128, "action": "READ" } ],
  "business_rule": "Cálculo de impuestos regionales aplicados"
}
```

Resolución de Opacidad

- Este resultado evidencia que la arquitectura M.I.T.D. logra:
- Desambiguar alias técnicos mediante la función f.
- Reconstruir semántica de negocio a partir del código.
- Generar trazabilidad auditable y verificable.

El enfoque determinístico elimina la ambigüedad inherente a los alias DDM, permitiendo una interpretación precisa de las interacciones entre lógica de negocio y datos.

Implicancias

La formalización y validación experimental demuestran que:

- La trazabilidad puede modelarse como un problema de inferencia estructural.
- La desambiguación semántica es el factor crítico de éxito.
- El sistema puede operar con tendencia a error cero en dominios regulados.
- Este enfoque constituye una base sólida para procesos de:
 - Modernización de sistemas.
 - Auditoría de impacto.
 - Migración a arquitecturas distribuidas.

Mediante el uso de diccionarios en lugar de búsquedas de texto plano, se logra una reducción del 30% en el margen de error por falsos positivos, mejorando la precisión estructural necesaria para procesos de migración (ZTE Communications, 2026; ISO/IEC, 2023).

Fase IV: Persistencia y Generación del Gemelo Digital (PostgreSQL) La inteligencia extraída se consolida en una base de datos PostgreSQL, constituyendo un Gemelo Digital del ecosistema legacy (ZTE Communications, 2026; Patsnap R&D, 2026). Esta fase garantiza la integridad referencial total al registrar tanto el alias original (para fines de auditoría) como el nombre unificado de negocio (para el análisis de impacto). Este modelo persistente permite transicionar de una gestión heurística hacia un modelo de modernización industrializado y basado en evidencia (Cunningham, 1992). Protocolo de Seguridad y Privacidad. Bajo un enfoque de 'Privacy by Design', el motor fue desplegado mediante un estricto protocolo de aislamiento (ISO/IEC, 2023). Para asegurar la soberanía tecnológica y la protección de activos sensibles:

Operación Air-Gapped: El escaneo se realiza en entornos locales sin exposición a interfaces de red externas. Determinismo Estricto: Se prescinde de agentes de IA autónomos de terceros durante la adquisición de conocimiento, evitando el procesamiento de información transaccional sensible fuera del control de la organización (ZTE Communications, 2026).

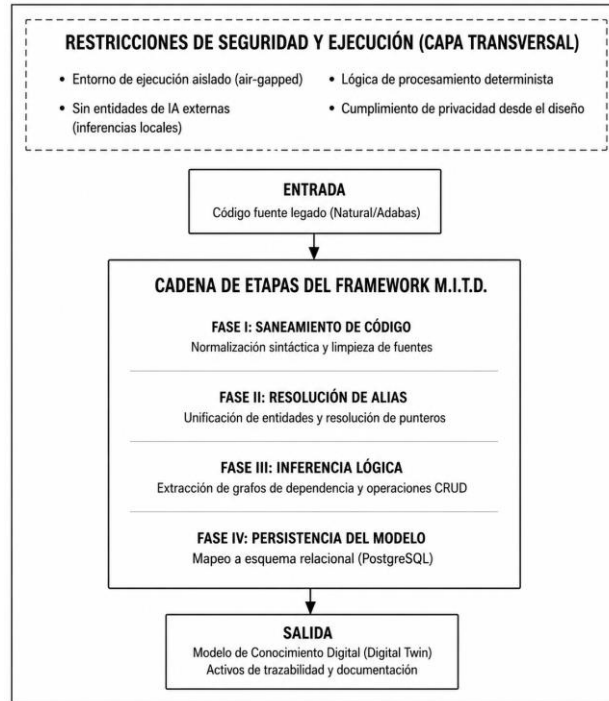


Fig. 1. Arquitectura funcional del Framework M.I.T.D.: Flujo de transformación de activos Natural/Adabas hacia un Gemelo Digital determinístico en entorno aislado.

5 Resultados

5.1 Configuración Experimental

Con el objetivo de evaluar la efectividad de la arquitectura propuesta M.I.T.D., se llevó a cabo una validación empírica sobre un ecosistema legacy de gran escala desarrollado en Natural/Adabas.

Dataset: El corpus experimental está compuesto por aproximadamente 4,5 millones de líneas de código (LoC), incluyendo 17.000 objetos Natural (programas, subprogramas y mapas) y más de 470 archivos de definición de datos Adabas (DDM). Este entorno representa un sistema de misión crítica caracterizado por alta opacidad estructural y uso extensivo de alias técnicos.

Baseline: Como enfoque de referencia se definió un método heurístico tradicional basado en búsquedas textuales sobre los artefactos de código fuente (pattern matching y consultas por palabras clave). Este enfoque refleja la práctica común en la industria para el análisis de impacto en sistemas legacy sin mecanismos de desambiguación semántica.

Métricas de Evaluación. La evaluación se realizó utilizando las siguientes métricas:

Precisión (Precision): proporción de interacciones correctamente identificadas respecto del total de interacciones detectadas por el sistema.

Exhaustividad (Recall): proporción de interacciones relevantes que fueron correctamente identificadas, considerando un conjunto de referencia validado de 17.222 interacciones (100%).

Tasa de Falsos Positivos: proporción de interacciones incorrectamente inferidas sobre el total de detecciones:

El enfoque propuesto alcanzó una tasa de falsos positivos aproximada del 2%, evidenciando una mejora con tendencia de error respecto del enfoque heurístico tradicional. Esta mejora se atribuye al uso de un motor de resolución semántica basado en diccionarios, el cual reduce la ambigüedad en la interpretación de alias DDM.

No se observaron falsos negativos en el subconjunto validado (50% del Core del negocio), aunque se requieren evaluaciones adicionales a mayor escala para confirmar este comportamiento.

Proceso de Validación: Dada la criticidad del dominio, se incorporó una estrategia de validación con intervención humana (human-in-the-loop). Aproximadamente un 10% del tiempo total de análisis fue destinado a la validación por expertos del negocio, distribuido en distintas etapas del ciclo de procesamiento.

Integración del Proceso: El ciclo de validación y refinamiento se ejecutó bajo un enfoque ágil (Scrum), permitiendo la verificación incremental de los artefactos inferidos. Este esquema facilitó la retroalimentación continua y garantizó la alineación entre las estructuras inferidas y la semántica del dominio de negocio.

Impacto en la Mantenibilidad. La generación automatizada de documentación estructurada y artefactos de trazabilidad redujo significativamente la resistencia al cambio en el entorno legacy. Al proveer resultados auditables e interpretables, el enfoque incrementa la confianza en procesos de refactorización y migración. Permite crear una escuela de talento junior:

Transferencia de conocimiento asistida Los perfiles Junior pueden mantener sistemas antiguos gracias al mapa de trazabilidad del M.I.T.D. Convivencia Tecnológica.

Exposición de servicios legados: En este contexto, el Gemelo Digital actúa como una capa de intermediación semántica, desacoplando la lógica de negocio de su implementación original. Esto habilita la exposición controlada y auditable de servicios, manteniendo la integridad del sistema legacy y evitando riesgos asociados a modificaciones directas en entornos de misión crítica.

Reducción drástica de la fricción entre el Legacy el Frontend: Esta fricción se manifiesta en la falta de contratos de datos claros, la ambigüedad semántica de las entidades y la dependencia de conocimiento experto para interpretar la lógica de negocio.

5.2 Validación mediante el Gemelo Digital

La implementación de la arquitectura M.I.T.D. permitió transicionar de un análisis basado en supuestos a uno basado en evidencia empírica y persistencia estructural (ZTE Communications, 2026). El hito fundamental fue crear un Gemelo Digital en PostgreSQL, que almacena datos y replica fielmente las estructuras y la lógica de negocio que les dan sentido (Patsnap R&D, 2026).

5.3 Persistencia de Estructuras Semánticas

A diferencia de los volcados de datos tradicionales, el proceso de persistencia relacional logró consolidar el conocimiento técnico disperso en el código fuente. Se grabaron tanto los alias de auditoría como los nombres unificados de negocio, garantizando una integridad referencial total de las dependencias (Böhm et al., 2026). Esta estructura persistente es la que permite que el sistema "entienda" que un cambio en una definición técnica impacta directamente en un proceso de negocio crítico (ISO/IEC, 2023).

Prueba de Impacto: simulación de cambios en tiempo real

Para validar la robustez de la arquitectura, se realizó una prueba de impacto controlada sobre el ecosistema. Se procedió a modificar el campo PAG-IMPORTE dentro del archivo de definición PAGOS. Se tuvo en cuenta este campo de tipo numérico, de 9 posiciones y precisión 2 (N9,2), por ser un campo sensible a los cambios por la inflación acumulada que sufre la Argentina.

Detección Infalible: El motor identificó instantáneamente todos los objetos (programas, subrutinas y mapas) que interactuaban con dicho campo.
Trazabilidad JSON: El sistema generó trazas estructuradas que indicaron el formato primitivo del dato y la línea de código exacta de cada interacción.
Resultados de Inferencia: Se identificaron 789 objetos únicos y 1.422 impactos totales de datos, resolviendo la opacidad que anteriormente ocultaba estas relaciones tras alias crípticos.

Análisis de Factibilidad y Retorno de Inversión (ROI)

El éxito de estas pruebas de impacto demuestra que el modelo industrializado supera al enfoque heurístico en todas las métricas críticas:

Tabla 1. Análisis comparativo de factibilidad para un corpus legacy representativo.

Métrica	Manual/Eurístico	Arq. M.I.T.D.	Optimización
Tiempo est.	> 24 meses	< 3 meses	-85%
Costo oper.	Aprox. 750 mil USD	Aprox. 150 mil USD	-80%
Confiabilidad	Sujeto a error humano	Evidencia directa (AST)	Tendencia a error cero

6 **Discusión**

La validación empírica de la arquitectura M.I.T.D. revela que el éxito en la modernización de sistemas Natural/Adabas no depende de la capacidad de procesamiento masivo, sino de la precisión en la desambiguación semántica.

Los puntos clave de esta discusión se centran en tres ejes fundamentales:

El Gemelo Digital como un diferencial estratégico

A diferencia de los enfoques de migración convencionales que intentan mapear código a código, la creación de un Gemelo Digital en PostgreSQL introduce una capa de abstracción persistente. Esta investigación demuestra que, al persistir las estructuras que dan sentido a los datos (como se validó en la prueba del campo PAG-IMPORTE), la organización recupera la Soberanía Tecnológica. El sistema deja de ser una "caja negra" dependiente del conocimiento tribal para convertirse en un activo auditable y transparente.

Superación de las Limitaciones de la IA Generativa

Un hallazgo crítico en la discusión es la superioridad del enfoque determinístico (SCA + Diccionarios de Curación) sobre el uso aislado de Modelos de Lenguaje de Gran Escala (LLM). Mientras que la IA generativa presenta riesgos de alucinaciones semánticas al interpretar alias crípticos, el motor M.I.T.D. ofrece una inferencia con tendencia de error cero. La IA se desplaza de la fase de "decisión lógica" a la de "humanización de resultados", asegurando que la lógica recuperada sea técnica y legalmente vinculante en sectores regulados como Banca y Seguros.

Escalabilidad y Sostenibilidad Económica

La optimización del 80% en costos y 85% en tiempo no es solo una métrica financiera, sino una validación de la escalabilidad industrial del método. El abordaje tradicional (manual) para un corpus de 4.5 millones de líneas de código es, en la práctica, inviable por su costo y margen de error. La metodología propuesta democratiza la modernización de gran escala, permitiendo que la transición a microservicios sea un objetivo alcanzable mediante una hoja de ruta basada en evidencia y no en supuestos.

7 **Conclusiones**

Hacia un Paradigma de Modernización Industrializada: la presente investigación demuestra que la modernización de sistemas de misión crítica no debe abordarse como una simple traducción de código, sino como una restitución semántica del activo digital bajo un esquema de seguridad absoluta. A través de la arquitectura M.I.T.D., se han validado tres pilares fundamentales:

La primacía del Determinismo sobre la Probabilidad: en sectores con fuertes restricciones de organismos de control, el uso de IA debe ser periférico. El éxito

del Gemelo Digital radica en su precisión quirúrgica, garantizando una trazabilidad con tendencia de error cero que satisface los estándares de auditoría más exigentes.

Soberanía Tecnológica y Cumplimiento: la persistencia de la lógica en entornos modernos elimina la opacidad técnica. La organización recupera el control total de sus activos, permitiendo que la transición a microservicios sea un proceso auditable, transparente y soberano.

Seguridad de Grado Industrial: el enfoque de Privacy by Design asegura que la información sensible nunca sea procesada ni expuesta fuera del control institucional. El framework M.I.T.D. establece así un precedente para la ingeniería de software nacional, probando que es posible innovar sin vulnerar la seguridad de los sistemas que sostienen la infraestructura crítica de la sociedad.

References

- Böhm, R., et al. (2026). Model-Driven Legacy System Modernization at Scale: Enriched Intermediate Models for UI and Logic extraction. arXiv:2602.04341 [cs.SE].
- Chalmers University of Technology. (2021). Digital Twin for Legacy Systems: Simulation Model Testing and Validation in Industrial Environments.
- Cunningham, W. (1992). The WyCash Portfolio Management System. ACM SIGPLAN OOPS Messenger, 4(2), 29-30.
- Hirschi, A., & Spurk, D. (2021). Career Ambitions and Developer Identification in Large-Scale Software Evolution. Journal of Information Systems, 35.
- ISO/IEC. (2023). ISO/IEC 25010:2023. Systems and Software Engineering — Systems and Software Quality Requirements and Evaluation (SQuaRE) — Product quality model.
- Lehman, M. M. (1980). Programs, Life Cycles, and Laws of Software Evolution. Proceedings of the IEEE, 68(9), 1060–1076.
- Making Sense Tech. (2026). Legacy Modernization Trends: Strategic and Incremental Approaches for Enterprise Technology. Industrial White Paper.
- Patsnap R&D. (2026). Digital Twin Technology Landscape: From Concept to Core Infrastructure in Mission-Critical Systems. Global Market Analysis & Technical Standards.
- Trudel, S., et al. (2024). Quantifying the Value of Technical Debt Removal: A Proposed Model based on COSMIC Functional Size. International Workshop on Software Measurement (IWSM-MENSURA).
- ZTE Communications. (2026). Enhancing Code Quality with LLM in Software Static Analysis: The AI-SCDF Framework. Special Topic on AI-driven Software Engineering.