

UNIVERSIDAD TÉCNICA FEDERICO SANTA MARÍA
DEPARTAMENTO DE INFORMÁTICA
SANTIAGO - CHILE



“APLICACIÓN WEB CON INTELIGENCIA ARTIFICIAL PARA RECOMENDACIÓN DE VISUALIZACIONES Y NARRATIVAS EXPLICATIVAS EN DATOS TABULARES”

FELIPE ANDRE LEÓN CÓRDOVA

MEMORIA PARA OPTAR AL TÍTULO DE
INGENIERO CIVIL EN INFORMÁTICA

Profesor Guía: José Luis Martí Lara
Profesor Correferente: Mauricio Olivares Faúndez

Julio - 2026



CONSTANCIA DE VALIDACIÓN Y CONFIDENCIALIDAD DE MONOGRAFÍA A REPOSITORIO ACADÉMICO

1.- IDENTIFICACIÓN DEL TRABAJO ACADÉMICO

Tipo de monografía (marcar una opción): ☒ Memoria o trabajo de título ☐ Tesis de Postgrado

Título del trabajo: Aplicación web con inteligencia artificial para recomendación de visualizaciones y narrativas explicativas en datos tabulares

Nombre del candidato(a): Felipe Andre León Córdova

Carrera / Grado: Ingeniería Civil Informática

Campus: San Joaquín

Departamento: Informática

2.- VALIDACIÓN DEL PROFESOR GUÍA/DIRECTOR DE TESIS

Yo, José Luis Martí Lara, en mi calidad de profesor(a) guía/director(a) del trabajo académico mencionado anteriormente **DEJO CONSTANCIA** que:

- He revisado esta versión del documento y corresponde a la versión final aprobada del trabajo.
- El trabajo cumple con los requisitos académicos y de formato establecidos por la institución.

3.- EVALUACIÓN DE CONFIDENCIALIDAD POR PROPIEDAD INDUSTRIAL (marcar una opción)

☒ El trabajo **NO contiene** información que amerite confidencialidad y puede ser publicado de inmediato en repositorio con acceso abierto.

☐ El trabajo **CONTIENE** información con potenciales implicancias de propiedad industrial o intelectual y requiere un periodo de confidencialidad (**embargo**) por (**marcar una opción**):

☐ 6 meses ☐ 12 meses ☐ 2 años ☐ 3 años ☐ 5 años ☐ 10 años

Fundamentación de la necesidad de confidencialidad (obligatorio si se solicita embargo):

4.- FIRMAS

Profesor(a) guía o director(a) de memoria o tesis:

Fecha: 28 - 07 - 2026

Firma:

Estudiante o Candidato(a):

Fecha: 30/07/2026

Firma:

Este formulario debe ser insertado como página 2 de la memoria o tesis, completado y firmado por estudiante y profesor(a) antes de la entrega en portal PRISMA de Biblioteca USM.

DEDICATORIA

A mi familia, por ser el pilar fundamental de mi vida. Desde siempre se preocuparon de que fuera una persona responsable y que buscara la excelencia en cada cosa que realizara.

Me enseñaron el valor del esfuerzo, la importancia de la honestidad y que las cosas importantes requieren dedicación y paciencia. Cada enseñanza, cada consejo, cada sacrificio que hicieron por mí me trajo hasta acá. Este logro es tanto mío como de ustedes, porque sin su apoyo incondicional y su ejemplo, nada de esto habría sido posible. Gracias por estar siempre presentes, por apoyarme en los momentos difíciles y por celebrar conmigo cada pequeño avance. Sin ustedes, no estaría aquí.

AGRADECIMIENTOS

A mi familia, por estar siempre ahí. Gracias por el apoyo constante durante todos estos años, por preocuparse de cómo me iba, por estar atentos a cómo avanzaba este proyecto y por impulsarme siempre a dar lo mejor. Cada paso de este camino lo recorrimos juntos, y este trabajo es en cierto modo el resultado de todo lo que me han dado.

A mis compañeros de universidad, con quienes compartí las largas horas de estudio, los trabajos en grupo y el estrés de los exámenes. Gracias por el compañerismo y por hacer que esta carrera fuera una etapa que vale la pena recordar.

Al profesor José Luis Martí Lara, por su guía y apoyo durante el desarrollo de esta memoria.

RESUMEN

Resumen— Esta memoria presenta el diseño, implementación y evaluación de una aplicación web automatizada para recomendación de visualizaciones y generación de narrativas explicativas sobre datos tabulares. La aplicación web integra un motor determinista —inferencia semántica, perfilamiento estadístico, generación de candidatos y *ranking* multiobjetivo— con un componente generativo acotado (LLM) que enriquece metadatos y redacta explicaciones ancladas a hechos. La arquitectura neuro-simbólica propuesta separa estrictamente el cálculo del lenguaje: el núcleo determinista asegura reproducibilidad y trazabilidad, mientras que el LLM aporta flexibilidad semántica sin inventar estadísticas. La aplicación web implementa siete métricas de interesancia, selección diversa mediante MMR y auditoría explícita de decisiones. La validación se realizó sobre *datasets* sintéticos de distintos dominios, midiendo pertinencia y diversidad de las sugerencias, tiempos de respuesta, trazabilidad de las decisiones y fidelidad factual de las narrativas generadas. Los resultados sugieren que el enfoque puede reducir la fricción inicial del EDA manteniendo control humano y explicabilidad de las recomendaciones.

Palabras Clave— Recomendación de visualizaciones; Análisis exploratorio de datos; Sistemas neuro-simbólicos; Narrativas explicativas; Inteligencia artificial generativa.

ABSTRACT

Abstract— This thesis presents the design, implementation, and evaluation of an automated web application for visualization recommendation and explanatory narrative generation on tabular data. The web application integrates a deterministic engine—semantic inference, statistical profiling, candidate generation, and multi-objective *ranking*—with a constrained generative component (LLM) that enriches metadata and writes fact-grounded explanations. The proposed neuro-symbolic architecture strictly separates computation from language: the deterministic core ensures reproducibility and traceability, while the LLM provides semantic flexibility without fabricating statistics. The web application implements seven interestingness metrics, diverse selection through MMR, and explicit decision auditing. Validation was conducted on synthetic *datasets* from different domains, measuring suggestion relevance and diversity, response times, decision traceability, and factual faithfulness of the generated narratives. Results suggest that this approach can reduce the initial friction of EDA while maintaining human control and recommendation explainability.

Keywords— *Visualization recommendation; Exploratory data analysis; Neuro-symbolic systems; Explanatory narratives; Generative artificial intelligence.*

GLOSARIO

- BI** Inteligencia de Negocios (*Business Intelligence*). Conjunto de herramientas y procesos que integran recolección, análisis y visualización de datos para apoyar la toma de decisiones en organizaciones.
- CSV** Valores Separados por Comas (*Comma-Separated Values*). Formato de archivo de texto plano que almacena datos tabulares, donde cada línea corresponde a una fila y los campos se delimitan por comas.
- EDA** Análisis Exploratorio de Datos (*Exploratory Data Analysis*). Proceso iterativo orientado a comprender la estructura, calidad y patrones de un conjunto de datos mediante resúmenes estadísticos y visualizaciones, previo a la construcción de modelos o conclusiones formales.
- HCI** Interacción Humano-Computador (*Human-Computer Interaction*). Disciplina que estudia el diseño, evaluación e implementación de sistemas computacionales interactivos, con énfasis en la usabilidad y la experiencia del usuario.
- IA** Inteligencia Artificial. Rama de la computación que desarrolla sistemas capaces de realizar tareas que requieren capacidades cognitivas humanas, como el aprendizaje, el razonamiento y la generación de lenguaje.
- IQR** Rango Intercuartílico (*Interquartile Range*). Medida de dispersión estadística definida como la diferencia entre el tercer y el primer cuartil, utilizada para describir la variabilidad de una distribución e identificar valores atípicos mediante la regla de Tukey.
- JSON** Notación de Objetos JavaScript (*JavaScript Object Notation*). Formato ligero de intercambio de datos basado en texto, estructurado en pares clave-valor, ampliamente utilizado para la comunicación entre sistemas y la serialización de información.
- LLM** Modelo de Lenguaje de Gran Escala (*Large Language Model*). Modelo de inteligencia artificial entrenado con grandes volúmenes de texto, capaz de generar, resumir y transformar lenguaje natural. En esta memoria su uso se acota a funciones de enriquecimiento semántico y redacción de narrativas sobre hechos ya calculados.
- MMR** Relevancia Marginal Máxima (*Maximal Marginal Relevance*). Técnica de selección que balancea relevancia y diversidad en un conjunto de resultados, reduciendo la redundancia mediante la penalización de elementos similares a los ya seleccionados.
- NLQ** Consulta en Lenguaje Natural (*Natural Language Query*). Modalidad de interacción con sistemas de datos que permite formular preguntas en lenguaje natural, sin requerir conocimiento de lenguajes de consulta formales.

ÍNDICE DE CONTENIDOS

RESUMEN	IV
ABSTRACT	IV
GLOSARIO	VI
ÍNDICE DE FIGURAS	X
ÍNDICE DE TABLAS	X
INTRODUCCIÓN	1
CAPÍTULO 1: DEFINICIÓN DEL PROBLEMA	3
1.1 Contexto de Aplicación y Línea Disciplinar	4
1.2 Situación Actual: ¿Qué se hace hoy y cómo se hace?	4
1.2.1 Herramientas de BI y analítica empresarial	4
1.2.2 Herramientas de programación	5
1.3 Actores Involucrados	5
1.4 Dificultades Actuales	5
1.5 Magnitud del Problema y Oportunidad	6
1.6 Competencia y Soluciones Existentes	7
1.7 Formulación del Problema	9
1.8 Objetivos y Alcances	9
1.8.1 Objetivo general	9
1.8.2 Objetivos específicos	10
1.8.3 Alcances y limitaciones	11
CAPÍTULO 2: MARCO CONCEPTUAL	12
2.1 Estado del arte	12
2.1.1 Enfoques basados en reglas y heurísticas	12
2.1.2 Enfoques basados en LLM/NLI	12
2.1.3 Enfoques híbridos (neuro-simbólicos)	13
2.1.4 Implicancias para el diseño de la solución	13
2.1.5 Síntesis conceptual y orientación de la memoria	13
2.2 Datos tabulares y significado	14
2.3 Análisis Exploratorio de Datos	15
2.4 Perfilamiento y calidad de datos	17
2.5 Visualización y analítica visual	18
2.5.2 Lecciones desde herramientas existentes	19
2.6 Gramáticas declarativas de visualización y trazabilidad	20
2.7 Recomendación de visualizaciones	21
2.8 Perfilamiento semántico con LLM	22

2.9	Ponderación por intención	22
2.10	Narrativa anclada a hechos	23
2.11	Auditoría de decisiones	23
2.12	Lenguaje natural para análisis visual	24
2.13	Desarrollo de aplicación web para analítica visual	24
CAPÍTULO 3: PROPUESTA DE SOLUCIÓN		26
3.1	Análisis de requisitos de la aplicación	26
3.2	Arquitectura general y principios de diseño	27
3.3	Paso 0: Ingesta y perfilamiento estadístico-semántico	27
3.4	Paso 1: Interpretación de intención y columnas foco	30
3.4.1	Modo dirigido y modo de exploración libre	30
3.4.2	Metas analíticas detectadas y foco de columnas	31
3.4.3	Elegibilidad de columnas y priorización a nivel de vistas	31
3.5	Paso 2: Generación, evaluación y <i>ranking</i> de visualizaciones	32
3.5.1	Generación sistemática de vistas candidatas (univariadas y bivariadas)	32
3.5.2	Métricas principales por tipo de vista (definiciones y sentido analítico)	35
3.5.3	Puntaje determinista: métrica principal + calidad, y ajuste por intención	37
3.5.4	Re- <i>ranking</i> por diversidad y cobertura: MMR (anti-redundancia)	40
3.6	Paso 3: Narrativa explicativa anclada en evidencia	42
3.6.1	Rol acotado del componente generativo (<i>grounding</i> y no-interferencia)	42
3.6.2	Narrativa descriptiva vs. prescriptiva	43
3.6.3	Paquete factual y contrato JSON (entrada y salida)	43
3.7	Paso 4: Auditoría, descartes y transparencia	46
3.7.1	Qué evidencia se expone (y cómo se vincula al <i>ranking</i>)	46
3.7.2	Descartes y alternativas (por qué otras vistas no quedaron)	47
3.7.3	Justificación perceptual (reglas de lectura del gráfico)	48
3.7.4	Exportación reproducible de evidencia	49
3.8	Capa de presentación e interactividad (<i>Streamlit</i>)	49
3.9	Síntesis: diferenciación frente a soluciones existentes	51
CAPÍTULO 4: VALIDACIÓN DE LA SOLUCIÓN		53
4.1	Entorno experimental	53
4.1.1	<i>Dataset</i> HR (People Analytics)	54
4.1.2	<i>Dataset</i> Retail Chile	55
4.1.3	<i>Dataset</i> Energy (Consumo Eléctrico)	56
4.2	Validación técnica	56
4.2.1	Pertinencia de las sugerencias	56
4.2.2	Tiempo de respuesta	59
4.2.3	Diversidad y cobertura	60
4.2.4	Transparencia y trazabilidad	61
4.2.5	Narrativa fiel a la evidencia	62
4.3	Discusión	62

CONCLUSIONES	64
REFERENCIAS BIBLIOGRÁFICAS	68
ANEXOS	71
Anexo A: Justificación de constantes de semi-saturación	71
Anexo B: Prompts utilizados por el componente generativo	72
B.1 Prompt de sistema para enriquecimiento de catálogo	72
B.2 Prompt de usuario para enriquecimiento de catálogo	72
B.3 Prompt de sistema para generación de narrativas en lote	72
B.4 Prompt de usuario para generación de narrativas en lote	72

ÍNDICE DE FIGURAS

1 Ciclo iterativo del EDA.	16
2 Jerarquía de efectividad perceptual de Cleveland y McGill [Cleveland y McGill, 1984].	18
3 Brecha semántica entre datos y visualización.	19
4 Articulación conceptual del marco teórico.	20
5 Arquitectura general de la solución propuesta.	28
6 Resumen inicial y catálogo semántico de la aplicación web, usando como ejemplo el <i>dataset</i> HR de recursos humanos.	30
7 Recomendación visual con narrativa y siguiente acción.	45
8 Panel de transparencia.	47
9 Registro de descartes de visualizaciones.	48

ÍNDICE DE TABLAS

1 Implicancias de diseño derivadas del estado del arte.	14
2 Productos del Paso 0 y su evidencia asociada.	29
3 Constantes de semi-saturación C por métrica.	35
4 Conjuntos de datos utilizados en la validación.	53
5 Columnas del <i>dataset</i> HR (People Analytics, 25 000 filas).	54
6 Columnas del <i>dataset</i> Retail Chile (200 000 filas).	55
7 Columnas del <i>dataset</i> Energy (540 000 filas).	56
8 Sugerencias para <i>dataset</i> HR (modo libre, $K = 6$).	57
9 Sugerencias para <i>dataset</i> HR (modo dirigido, $K = 6$).	58
10 Sugerencias para <i>dataset</i> Retail Chile (modo libre, $K = 6$).	58
11 Tiempos por fase (<i>dataset</i> HR, Gemini 2.5 Flash Lite y Ollama con qwen2.5:0.5b).	59
12 Tiempos del motor determinista por <i>dataset</i> (3 repeticiones).	60
13 Diversidad del conjunto de sugerencias ($K = 6$).	60
14 Información de trazabilidad (boxplot de salarios por nivel).	61
15 Justificación de constantes de semi-saturación por métrica.	71

INTRODUCCIÓN

La toma de decisiones en organizaciones públicas y privadas depende cada vez más del análisis de datos. Equipos de distintas áreas —finanzas, operaciones, recursos humanos, planificación— trabajan cotidianamente con datos tabulares provenientes de sistemas transaccionales, planillas y reportes operacionales. Sin embargo, el valor de estos datos no se materializa automáticamente: antes de construir modelos o tableros, es necesario comprender la estructura del conjunto de datos, evaluar su calidad, detectar patrones relevantes e identificar anomalías. Este proceso, conocido como Análisis Exploratorio de Datos (EDA), constituye una etapa crítica que consume tiempo y requiere habilidades tanto técnicas como de comunicación.

En la práctica, el EDA enfrenta fricciones persistentes. Primero, existe una brecha semántica entre la forma en que las personas formulan preguntas (en lenguaje natural, a menudo incompletas) y la estructura rígida de los datos tabulares. Segundo, incluso cuando se logra producir una visualización razonable, suele faltar el componente narrativo que explique qué se observó, por qué importa y qué evidencia lo respalda. Tercero, la incorporación de asistentes basados en inteligencia artificial generativa ha introducido nuevas capacidades, pero también riesgos: estos sistemas pueden producir texto convincente pero incorrecto si no existen mecanismos de control y anclaje a hechos verificables.

Esta memoria aborda el problema de facilitar la exploración y comprensión de datos tabulares mediante una aplicación web automatizada que integra recomendación de visualizaciones con generación de narrativas explicativas. El enfoque adoptado es neuro-simbólico: combina un motor determinista —responsable de la inferencia semántica, el perfilamiento estadístico, la generación de candidatos, el cálculo de métricas y el *ranking*— con un componente generativo acotado que enriquece metadatos y redacta explicaciones sobre hechos ya calculados. Esta separación estricta entre cálculo y lenguaje permite mantener reproducibilidad, trazabilidad y control sobre las decisiones de la aplicación web.

La aplicación web propuesta implementa un *pipeline* organizado en cinco etapas: perfilamiento semántico y estadístico del *dataset*, con construcción de un catálogo por columna; interpretación de intención del usuario y planificación determinista de la exploración; generación sistemática de vistas candidatas con *scoring* multiobjetivo basado en siete métricas de interesancia y selección diversa mediante *Maximal Marginal Relevance*; generación de narrativas prescriptivas ancladas a evidencia; y auditoría explícita que documenta qué se recomendó, por qué y qué alternativas se descartaron.

La metodología de desarrollo siguió un enfoque iterativo, alternando implementación con verificación de alineación respecto a los objetivos específicos definidos. Se utilizó Python como lenguaje principal, con pandas para manipulación de datos, SciPy para cálculos estadísticos, Altair/Vega-Lite para especificaciones declarativas de visualizaciones y Streamlit para la capa de presentación interactiva. La integración con modelos de lenguaje soporta

proveedores en nube (*Gemini*) y ejecución local mediante Ollama. La evaluación se realizó sobre *datasets* sintéticos de distintos dominios, midiendo pertinencia y diversidad de las sugerencias visuales, tiempos de respuesta por etapa, trazabilidad de las decisiones y fidelidad factual de las narrativas generadas.

El presente documento se organiza de la siguiente manera. El capítulo 1 presenta la definición del problema, incluyendo el contexto de aplicación, las dificultades actuales, la formulación del problema y los objetivos con sus alcances. El capítulo 2 desarrolla el marco conceptual que sustenta la propuesta, abarcando desde perfilamiento y EDA hasta arquitecturas neuro-simbólicas y control del riesgo en sistemas con componentes generativos. El capítulo 3 describe en detalle la propuesta de solución, explicando la arquitectura, las cinco etapas del *pipeline*, las métricas implementadas, el algoritmo de selección diversa y la capa de presentación. El capítulo 4 presenta la validación de la solución, documentando las pruebas realizadas y los resultados obtenidos frente a cada objetivo específico. Finalmente, se exponen las conclusiones, incluyendo los principales aportes, las limitaciones identificadas y las líneas de trabajo futuro.

CAPÍTULO 1

DEFINICIÓN DEL PROBLEMA

En los últimos años, la toma de decisiones en organizaciones públicas y privadas se ha vuelto crecientemente dependiente del análisis de datos. Esta tendencia no se limita a áreas técnicas: equipos de finanzas, operaciones, marketing, recursos humanos, auditoría y planificación trabajan con datos tabulares (planillas, exportaciones de ERP/CRM, registros operacionales, etc.) para reportes, seguimiento de indicadores, evaluación de riesgos y priorización de iniciativas. Sin embargo, el valor de estos datos no se materializa automáticamente. Antes de construir modelos predictivos o tableros consolidados, es indispensable realizar un proceso de Análisis Exploratorio de Datos (EDA), cuyo propósito es comprender la estructura del conjunto de datos, evaluar su calidad, detectar patrones, identificar valores atípicos y formular hipótesis de análisis [NIST/SEMATECH, 2023].

A pesar de su relevancia, el EDA y la comunicación de hallazgos siguen siendo etapas difíciles de ejecutar en forma eficiente cuando se trabaja con datos reales: heterogéneos, incompletos, con columnas ambiguas y semántica de negocio implícita. En términos prácticos, gran parte del costo y el riesgo del análisis no proviene de “hacer un gráfico”, sino de decidir qué visualizar y por qué, conectar esa vista con una intención analítica del usuario y producir una explicación textual que sea útil, verificable y consistente con la evidencia observada.

La problemática se intensifica por dos razones principales. Primero, existe una brecha persistente entre la forma en que las personas formulan preguntas (lenguaje natural, a veces vago o incompleto) y la forma rígida en que están estructurados los datos (tablas, columnas, tipos y relaciones). Segundo, incluso cuando se logra producir una visualización razonable, suele faltar un componente crucial: la narración y explicación de lo observado para que el resultado sea comprensible, trazable y accionable para distintos perfiles (analistas, jefaturas y tomadores de decisión). Esta necesidad se vuelve más evidente con la incorporación de asistentes basados en IA generativa, donde el texto puede escalar la comunicación, pero también introducir errores plausibles si no existen mecanismos de control y anclaje a hechos [National Institute of Standards and Technology (NIST), 2024].¹

La importancia del problema no se limita a reducir tiempo operativo. En contextos organizacionales, una visualización mal escogida o una explicación convincente pero no respaldada puede orientar decisiones sobre recursos, riesgos o prioridades con una confianza injustificada. Por el contrario, un flujo que explicita evidencia, supuestos, criterios de selección y razones de descarte permite que analistas y tomadores de decisión revisen el proceso que llevó al resultado, no solo consuman una salida final. Esta memoria se enfoca precisamente en esa dimensión: convertir la exploración inicial de datos tabulares en un proceso más claro, trazable y replicable.

¹En este documento, “anclaje” se entiende como la restricción explícita de la narrativa a hechos medibles (estadísticas, conteos, percentiles, correlaciones, tendencias) derivados del *dataset* y del proceso de análisis.

1.1 Contexto de Aplicación y Línea Disciplinar

Esta memoria se sitúa en la intersección de ciencia de datos y analítica, visualización de datos e interacción humano-computador (HCI): por un lado, por la necesidad de caracterizar y resumir información tabular; por otro, por el rol de las representaciones gráficas como interfaz cognitiva para detectar patrones y comunicar resultados; y, finalmente, por el desafío de traducir intenciones humanas a acciones de análisis y reducir fricción en el flujo de trabajo.

En este marco, el problema no se reduce a “generar gráficos”, sino a habilitar descubrimiento y explicación de manera consistente en escenarios cotidianos donde:

- Los datos provienen de múltiples fuentes y llegan con distinta calidad y convenciones
- La persona usuaria no necesariamente sabe qué buscar al inicio (exploración abierta)
- La comunicación final requiere justificar qué se observó, con evidencia verificable y con un lenguaje entendible para la audiencia

1.2 Situación Actual: ¿Qué se hace hoy y cómo se hace?

En la práctica, el EDA y la construcción de una narrativa explicativa se realizan hoy, de manera predominante, mediante dos familias de herramientas, las cuales se presentan a continuación.

1.2.1 Herramientas de BI y analítica empresarial

El flujo típico en entornos de negocio comienza con la extracción de datos a formatos tabulares (CSV/Excel) y su exploración en herramientas de inteligencia de negocios (BI). Allí se producen filtros, agregaciones, tablas dinámicas y visualizaciones base. Este enfoque facilita el acceso, pero el resultado depende fuertemente de la pericia de quien analiza: seleccionar variables pertinentes, aplicar agregaciones correctas, evitar comparaciones engañosas y reconocer limitaciones del dato.

En años recientes, estas plataformas han incorporado capacidades de asistencia con IA generativa. Sin embargo, la documentación y práctica de uso muestran una condición crítica: la efectividad de estas experiencias depende de que exista un modelo semántico o contexto preparado (definiciones, relaciones, descripciones y reglas de negocio). En caso contrario, se reconoce el riesgo de resultados genéricos o incorrectos, y se recomienda preparar adecuadamente el modelo y la información contextual [Microsoft, 2025]. Este punto refleja que el desafío no es solo “tener un LLM”, sino conectar de manera robusta el lenguaje del usuario con la estructura y semántica de los datos.

Adicionalmente, la evolución de funcionalidades basadas en lenguaje natural no ha sido lineal. Por ejemplo, Tableau retiró *Ask Data* en *Tableau Cloud* y en versiones recientes de *Tableau Server*, indicando una transición hacia nuevas experiencias basadas en avances en tecnologías de lenguaje natural [Tableau, 2024]. Esto sugiere que la industria percibe el valor del paradigma, pero aún existe tensión entre utilidad, mantenibilidad, y consistencia del comportamiento en el tiempo.

1.2.2 Herramientas de programación

En equipos técnicos, el EDA suele realizarse en *notebooks* mediante código. Este enfoque ofrece flexibilidad, reproducibilidad y control fino, pero introduce barreras para usuarios no programadores y aumenta el costo de iterar: cada nueva hipótesis implica escribir código, probar, ajustar, elegir codificaciones visuales y, luego, redactar conclusiones. En escenarios con presión de tiempo, esta fricción reduce la exploración efectiva y tiende a concentrar la capacidad analítica en pocos perfiles especializados.

1.3 Actores Involucrados

El problema involucra múltiples perfiles con necesidades distintas:

- **Usuarios de negocio** (control de gestión, planificación, operaciones, finanzas, etc.), cuyo objetivo es responder preguntas y tomar decisiones con rapidez. Conocen el dominio, pero no necesariamente dominan estadística, modelamiento de datos o diseño de visualizaciones.
- **Analistas de datos y científicos de datos**, quienes pueden manejar herramientas técnicas, pero enfrentan presión por producir resultados reproducibles, comunicables y auditables. Además, suelen actuar como puente entre el lenguaje del negocio y el lenguaje del dato, lo cual amplifica su carga cuando la organización escala el uso de datos.
- **Tomadores de decisión** (jefaturas, gerencias, líderes), cuyo foco es la confianza: necesitan entender qué significa un hallazgo, qué evidencia lo respalda, qué supuestos existen y qué limitaciones podrían afectar la decisión.

1.4 Dificultades Actuales

A partir del contexto descrito, se identifican cuatro dificultades concretas que obstaculizan el flujo de análisis exploratorio y que justifican la necesidad de una aplicación web asistida.

- **Brecha semántica entre intención y estructura tabular.** Una dificultad central es que las preguntas humanas raramente se formulan con precisión técnica. Expresiones como “¿qué explica la variación de ventas?” o “¿dónde están los atrasos?” no especifican columnas, filtros, granularidad temporal ni criterios de agregación. Transformar la intención en operaciones de análisis requiere un puente semántico entre lenguaje natural y estructura tabular (nombres de columnas, unidades, roles y relaciones), que típicamente no está formalizado.
- **Selección de visualizaciones y riesgo de malas prácticas.** Otra fuente relevante de error es la selección de una visualización inadecuada. Un gráfico puede ser técnicamente correcto pero comunicacionalmente defectuoso, o puede inducir interpretaciones erróneas por escalas, agregaciones inapropiadas o codificaciones visuales poco efectivas. En contextos con presión de tiempo, estos errores se vuelven más probables: se prioriza rapidez por sobre validación. En consecuencia, el proceso requiere mecanismos que orienten hacia buenas prácticas y que hagan explícitos los supuestos que justifican una visualización por sobre otra.
- **Narración manual y riesgo de generación automática.** Incluso cuando se encuentran patrones interesantes, traducirlos a una explicación clara exige habilidades de comunicación y un orden lógico: qué se observó, por qué importa, qué evidencia lo respalda, qué alternativas se consideran y qué limitaciones existen. En la práctica, este paso suele hacerse manualmente y queda desacoplado del proceso que originó los cálculos, reduciendo trazabilidad y reproducibilidad.

Con la incorporación de modelos generativos, es posible automatizar parte de esta narración, pero aparece un riesgo reconocido: los sistemas generativos pueden producir contenido plausible pero incorrecto si no existe control sobre el uso de hechos, fuentes y restricciones [NIST, 2024]. En un contexto de analítica, esto es especialmente sensible, porque una narrativa convincente puede amplificar errores y afectar decisiones.

- **Latencia y continuidad del flujo de análisis.** La exploración de datos es un proceso iterativo: se prueba una vista, se ajusta un filtro, se cambia una variable, se compara contra otra, etc. Desde la disciplina de HCI, se reconoce que la latencia afecta directamente la continuidad cognitiva del usuario: respuestas muy lentas rompen el flujo y reducen la cantidad de iteraciones que las personas están dispuestas a realizar [Nielsen, 2009b]. Por ello, más allá de la corrección, la practicidad del proceso depende de mantener tiempos de respuesta compatibles con un uso interactivo.

1.5 Magnitud del Problema y Oportunidad

La oportunidad del problema se observa por la expansión sostenida del uso de datos en funciones no técnicas, la adopción acelerada de experiencias de IA en herramientas de analítica empresarial y la creciente necesidad de transparencia y control cuando se integran componentes generativos. La aparición de asistentes como Copilot en Power BI evidencia una direc-

ción clara del mercado: convertir preguntas en análisis y comunicación de manera más directa, pero con dependencias críticas de preparación semántica y validación [Microsoft, 2025]. En paralelo, los marcos de gestión de riesgos de IA enfatizan que, cuando se despliegan capacidades generativas, deben incorporarse medidas de gobernanza de datos, trazabilidad y control proporcional al riesgo de uso. [NIST, 2024]

1.6 Competencia y Soluciones Existentes

Existen soluciones comerciales y de código abierto que cubren partes del problema: asistentes en herramientas BI, sistemas de preguntas/respuestas (NLQ/QA) sobre modelos semánticos o capas métricas, bibliotecas y flujos de EDA automatizado, y propuestas académicas de recomendación de visualizaciones. Sin embargo, al observar el flujo completo que esta memoria busca resolver (intención → selección de vistas → explicación verificable → trazabilidad y descarte), persisten brechas relevantes en escenarios cotidianos con datos tabulares “crudos” (CSV/Excel) y usuarios con presión de tiempo. A continuación se revisan estas familias de soluciones y sus limitaciones.

- **Asistentes en herramientas BI (Power BI, Tableau y similares).** Las plataformas de BI han incorporado capacidades de IA para acelerar preguntas y generación de contenido, pero su efectividad depende de condiciones fuertes: disponibilidad de un modelo semántico preparado, definiciones consistentes y curación de metadatos. En ausencia de ese contexto, la interpretación tiende a degradarse (ambigüedad de columnas, unidades implícitas, granularidad no declarada), y la experiencia se vuelve frágil o genérica [Microsoft, 2025]. Además, estos asistentes suelen priorizar la conveniencia (“respuesta rápida”) sobre trazabilidad: el usuario ve un resultado, pero no necesariamente un rastro explícito de qué métricas se calcularon, qué alternativas se evaluaron y por qué se descartaron.
- **NLQ/QA sobre capas métricas y modelos semánticos.** Sistemas de preguntas/respuestas sobre “capas semánticas” (p. ej., catálogos, métricas predefinidas, o un *semantic layer*) reducen la barrera de entrada, pero desplazan el problema hacia la preparación: si el significado no está explicitado (sinónimos, jerarquías, unidades, relaciones), el sistema debe adivinar. En la práctica, esto produce dependencia de ingeniería previa del modelo (que no siempre existe en escenarios de planillas) y respuestas difíciles de auditar cuando el mapeo entre texto y datos no queda expresado como evidencia verificable. En contraste, esta memoria plantea un enfoque en el que el “contexto” se construye automáticamente como catálogo semántico liviano, y el motor determinista conserva un rastro de decisiones.
- **Bibliotecas y flujos de EDA automatizado (reportes).** El ecosistema Python/R ofrece herramientas que generan reportes automáticos de EDA (perfilamiento, correlaciones, faltantes, *outliers*) de forma rápida. Estas soluciones son útiles para panoramas

iniciales, pero suelen presentar limitaciones frente al objetivo de esta memoria: tienden a producir salidas densas (un “documento”) más que un conjunto pequeño de recomendaciones rankeadas y diversas, no incorporan de forma consistente el eslabón intención del usuario → priorización de vistas (pesos dinámicos), y no estructuran la narrativa como un artefacto verificable, sino como descripciones o *insights* genéricos. En términos prácticos, estas herramientas reducen trabajo mecánico, pero no cierran el ciclo intención–*ranking*–explicación–auditoría.

- **Propuestas académicas de recomendación de visualizaciones.** Trabajos como Lux formalizan recomendaciones “always-on” en *dataframes*, con generación de vistas candidatas y priorización por medidas de *interestingness* [Lee *et al.*, 2022]. No obstante, muchas propuestas académicas se enfocan en el motor de *ranking* o en la interacción, y dejan como aspecto secundario (o no operacionalizan) requerimientos de despliegue típicos del problema aplicado aquí: narrativa prescriptiva y anclada a hechos para audiencias no técnicas, controles explícitos para minimizar riesgo cuando se integra generación automática, y una capa de transparencia que explique selección y descarte en términos entendibles y verificables.

A pesar de las soluciones anteriores, aún persisten brechas que motivan el enfoque ingenieril de esta memoria:

- **Dependencia del contexto semántico y del “modelo preparado”.** Muchas soluciones funcionan bien cuando existe un modelo semántico curado; en datos tabulares crudos (CSV/Excel), la desambiguación (unidad, rol, jerarquía, granularidad) se vuelve frágil y el sistema tiende a respuestas genéricas o inconsistentes [Microsoft, 2025].
- **Foco parcial.** Es común encontrar herramientas que resuelven “solo visualización” o “solo resumen”. Menos frecuente es una integración coherente y evaluable de punta a punta: intención → generación y *ranking* de vistas → narrativa verificable → auditoría con descarte. Cuando existe, suele ser opaca (caja negra) o dependiente de contexto manual.
- **Trazabilidad y reproducibilidad como salidas de primera clase.** En herramientas con componentes generativos, el desafío no es solo producir texto, sino acotar el texto a evidencia calculada y preservar un rastro mínimo que permita responder “¿por qué este gráfico?” y “¿por qué no otro?”. Este requisito es crítico en analítica por el riesgo de explicaciones plausibles pero incorrectas, y por la necesidad de control proporcional al riesgo [NIST, 2024].
- **Evolución y discontinuidad de funcionalidades NLQ en productos.** La industria ha mostrado iteraciones y reemplazos de experiencias de lenguaje natural, incluyendo retiro de funcionalidades históricas, lo que evidencia una madurez en curso y posibles cambios de comportamiento por estrategia de producto [Tableau, 2024].

- **Interactividad y continuidad del flujo analítico.** Reportes automáticos ayudan, pero no siempre soportan un ciclo iterativo rápido guiado por el usuario (refinar intención, ajustar umbrales, comparar vistas, exportar evidencia) con latencias compatibles con uso interactivo [Nielsen, 2009a].

En consecuencia, existe espacio para abordar el problema desde una perspectiva de ingeniería de software: diseñar una solución que conecte exploración visual con explicación textual y que mantenga control explícito sobre supuestos, evidencias y decisiones.

1.7 Formulación del Problema

Considerando lo anterior, el problema que aborda esta memoria puede formularse de la siguiente manera:

Existe una brecha entre las necesidades de exploración y comprensión de datos tabulares (formuladas en lenguaje natural y bajo presión de tiempo) y las capacidades prácticas de las herramientas actuales para producir visualizaciones y narrativas explicativas que sean útiles, verificables y transparentes respecto de sus decisiones.

De esta formulación se desprenden tres dimensiones inseparables del problema: exploración (descubrir patrones sin saber exactamente qué buscar), explicación (convertir hallazgos en narrativa comprensible y anclada a evidencia) y confianza (tolerancia baja a errores plausibles, necesidad de trazabilidad y control humano en decisiones).

1.8 Objetivos y Alcances

Los objetivos de esta memoria se formulan para responder a las dimensiones del problema identificadas: exploración, explicación y confianza. El enfoque se alinea con la definición de usabilidad como efectividad, eficiencia y satisfacción en un contexto de uso definido [International Organization for Standardization (ISO), 2018].

1.8.1 Objetivo general

El objetivo general de esta memoria es diseñar, implementar y evaluar una aplicación web automatizada, apoyada en inteligencia artificial, que a partir de datos tabulares proponga visualizaciones complementarias, narrativas explicativas y evidencia de sus decisiones, con

el fin de apoyar las primeras etapas del Análisis Exploratorio de Datos (EDA). La evaluación se realizará mediante criterios técnicos de pertinencia, diversidad, tiempo de respuesta, trazabilidad y fidelidad factual de las narrativas.

1.8.2 Objetivos específicos

- **OE1. Pertinencia y diversidad de las sugerencias visuales.** Proponer visualizaciones compatibles con los tipos semánticos de las columnas, con la métrica analítica utilizada y con la intención declarada cuando exista una consulta del usuario. Además, el conjunto final debe cubrir distintos ángulos de análisis, evitando redundancia entre vistas. Su cumplimiento se verificará revisando las sugerencias generadas por *dataset*, el cambio de *ranking* en modo dirigido, la cantidad de tipos de gráfico distintos, las columnas involucradas y las acciones analíticas cubiertas.
- **OE2. Tiempo de respuesta e interactividad.** Mantener un flujo de uso compatible con la exploración iterativa, distinguiendo las etapas que componen el tiempo de respuesta de la aplicación: enriquecimiento semántico, motor determinista y generación de narrativas. Su cumplimiento se verificará midiendo el tiempo hasta obtener las primeras visualizaciones y métricas disponibles para el usuario, además de identificar qué componentes concentran la latencia bajo distintos proveedores LLM y *datasets*.
- **OE3. Transparencia y trazabilidad.** Incorporar evidencia explícita que permita revisar cada recomendación y comprender por qué una visualización fue sugerida o descartada. Su cumplimiento se verificará inspeccionando si las sugerencias incluyen tipo de gráfico, columnas utilizadas, métrica principal, valor calculado, *score*, calidad del dato, transformaciones aplicadas y razones de descarte cuando corresponda. Este objetivo se alinea con recomendaciones de gestión de riesgos y control en sistemas de IA, especialmente cuando se integra generación automática de texto [NIST, 2024].
- **OE4. Narrativa explicativa fiel a la evidencia.** Generar narrativas que resuman la visualización, expliciten la evidencia principal y propongan un siguiente paso de análisis sin incorporar cifras, relaciones o interpretaciones que no estén respaldadas por los insumos calculados. Su cumplimiento se verificará revisando una muestra de narrativas generadas por la aplicación y contrastando sus afirmaciones con el paquete factual entregado al componente generativo. Dado el riesgo de contenido plausible pero incorrecto en componentes generativos, se exige control explícito sobre los hechos utilizados [NIST, 2024].

Cada objetivo específico contribuye directamente a una dimensión del objetivo general:

- **OE1** cubre la dimensión de “proponer visualizaciones complementarias”: evalúa pertinencia respecto de tipos semánticos, intención declarada y diversidad del conjunto recomendado.

- **OE2** cubre la dimensión de “apoyar la exploración”: mide tiempos por etapa y distingue el tiempo hasta disponer de visualizaciones iniciales de los componentes que concentran la latencia.
- **OE3** cubre la dimensión de “rastros de decisiones para favorecer transparencia”: verifica si cada recomendación puede revisarse mediante métricas, puntajes, calidad del dato, transformaciones y razones de descarte.
- **OE4** cubre la dimensión de “narrativa explicativa” y “resultados verificables”: contrasta las explicaciones generadas con los hechos calculados por el motor determinista, evitando invención de cifras o relaciones.

1.8.3 Alcances y limitaciones

El alcance de la memoria se centra en conjuntos de datos tabulares y tareas típicas de EDA: caracterización de variables, detección de patrones, comparación entre dimensiones y comunicación de hallazgos mediante visualizaciones y texto explicativo. No se busca reemplazar plataformas de BI empresariales ni resolver gobernanza de datos organizacional completo; el foco está en la integración efectiva entre exploración visual, narrativa y trazabilidad. Como restricción de diseño, la aplicación web se limita a narrativas estrictamente ancladas a hechos calculados por el motor determinista, evitando la generación libre de contenido por parte del componente generativo [NIST, 2024].

CAPÍTULO 2

MARCO CONCEPTUAL

Este capítulo establece los fundamentos conceptuales de la memoria: cómo pasar desde datos tabulares crudos hacia visualizaciones recomendadas, explicadas y auditables. Su organización sigue tres niveles. Primero se revisa el estado del arte para delimitar qué han resuelto los trabajos existentes y qué brechas siguen abiertas. Luego se presentan conceptos base para entender el dominio: datos tabulares, EDA, perfilamiento, calidad de datos y visualización. Finalmente, se desarrollan conceptos específicos que sustentan la propuesta: gramáticas declarativas, recomendación de visualizaciones, apoyo semántico con LLM, narrativa, auditoría y desarrollo de aplicación web.

2.1 Estado del arte

La automatización de análisis visual sobre datos tabulares puede organizarse en tres enfoques. Cada uno resuelve una parte del problema, pero presenta límites cuando se exige cubrir simultáneamente recomendación, explicación, lenguaje natural y trazabilidad.

2.1.1 Enfoques basados en reglas y heurísticas

Los enfoques basados en reglas, como Voyager [Wongsuphasawat *et al.*, 2016, Wongsuphasawat *et al.*, 2017], Lux [Lee *et al.*, 2022], Draco [Moritz *et al.*, 2019] y DeepEye [Luo *et al.*, 2018], generan vistas válidas mediante tipos de datos, heurísticas, restricciones o modelos de *ranking*. Su principal fortaleza es la trazabilidad: las decisiones pueden auditarse y reproducirse. Su limitación es semántica: no capturan bien intención implícita, ambigüedad del lenguaje natural ni significado de dominio sin una capa adicional de interpretación.

2.1.2 Enfoques basados en LLM/NLI

Los enfoques basados en lenguaje natural, como NL4DV [Narechania *et al.*, 2020], LIDA [Gao *et al.*, 2021] y propuestas asistidas por LLM [Cheng y Satyanarayan, 2023], traducen preguntas a especificaciones o código de visualización. Su fortaleza es la flexibilidad semántica: resuelven sinonimia, referencias implícitas y mapeo a columnas. Su riesgo es la opacidad: pueden producir especificaciones inválidas, inventar transformaciones o generar resultados difíciles de verificar sin validación externa.

2.1.3 Enfoques híbridos (neuro-simbólicos)

Los enfoques híbridos combinan interpretación semántica con ejecución controlada. Data Formulator [Wang *et al.*, 2023], por ejemplo, separa intención de alto nivel y transformación de datos, pero se orienta principalmente a la autoría de gráficos. Esta línea reduce algunos riesgos de los LLM puros, aunque introduce complejidad de integración: requiere interfaces claras para que la capa generativa no contamine cálculos, validaciones ni decisiones de *ranking*.

2.1.4 Implicancias para el diseño de la solución

La revisión anterior muestra que ninguna familia técnica resuelve por sí sola el flujo completo que se busca abordar. Las reglas y heurísticas aportan control, pero requieren apoyo semántico; los enfoques de lenguaje natural aportan flexibilidad, pero necesitan validación; y los enfoques híbridos muestran una dirección prometedora, aunque demandan fronteras claras entre interpretación y cálculo. La Tabla 1 sintetiza estas implicancias de diseño.

2.1.5 Síntesis conceptual y orientación de la memoria

A partir de estas implicancias, esta memoria se orienta conceptualmente hacia un enfoque neuro-simbólico acotado:

1. La interpretación semántica puede apoyar el mapeo entre lenguaje natural y columnas, pero debe mantenerse separada de los cálculos verificables.
2. La generación y selección de vistas requiere reglas explícitas, métricas de interés y criterios de diversidad.
3. La narrativa debe construirse sobre hechos calculados, evitando que el modelo generativo invente resultados.
4. La auditoría debe exponer evidencia cuantitativa, reglas aplicadas y razones de selección o descarte.

Esta orientación se alinea con la tendencia de sistemas neuro-simbólicos en visualización [Setlur *et al.*, 2023], pero enfatiza la separación estricta entre interpretación, cálculo y explicación como mecanismo de control de calidad y trazabilidad.

Con estas brechas como contexto, las siguientes secciones presentan primero los conceptos base necesarios para comprender el problema antes de abordar los mecanismos específicos de recomendación, narrativa y trazabilidad.

Enfoque	Aporte	Implicancia para el diseño
Reglas y heurísticas de recomendación visual	Permiten generar vistas válidas y explicar por qué una codificación es pertinente según tipos de dato, agregaciones y restricciones de legibilidad.	Conviene mantener un motor determinista para generación de candidatos, filtros de validez y reglas perceptuales.
Métricas de <i>interestingness</i> y <i>ranking</i>	Entregan señales comparables para priorizar patrones explorables, como asimetría, concentración, tendencia o asociación.	La priorización debe combinar señal analítica y calidad de datos, evitando depender solo de intuición visual.
Diversificación mediante MMR	Reduce redundancia y favorece cobertura entre vistas cuando varias alternativas tienen puntajes similares [Carbonell y Goldstein, 1998].	La selección final debe optimizar el conjunto completo, no solo ordenar vistas por puntaje individual.
Gramáticas declarativas de visualización	Representan gráficos como especificaciones inspeccionables y reproducibles, separando estructura visual de renderizado [Satyanarayan <i>et al.</i> , 2017].	La vista recomendada debe conservarse como especificación auditable, no solo como imagen final.
Lenguaje natural y LLM	Ayudan a interpretar intención, sinónimos y significado de columnas, pero pueden introducir ambigüedad o respuestas no verificables.	Su uso debe acotarse a orientación semántica sobre metadatos e intención, manteniendo validación externa.
Trazabilidad del proceso	Permite reconstruir por qué una vista fue generada, priorizada o descartada, en lugar de presentar solo el resultado final.	Las decisiones deben registrarse con evidencia, puntajes, transformaciones y criterios explícitos.

Tabla 1: Implicancias de diseño derivadas del estado del arte.

2.2 Datos tabulares y significado

En contextos organizacionales, gran parte del análisis cotidiano se realiza sobre datos tabulares: reportes, planillas, archivos CSV y exportaciones desde sistemas transaccionales. Computacionalmente, estos datos suelen representarse como tablas o *dataframes*, donde cada fila corresponde a una observación y cada columna a un atributo. Este formato facilita filtrado, agregación, unión y visualización, por lo que es especialmente útil para EDA y prototipado.

Sin embargo, el valor analítico de un *dataframe* no se deduce solo por sus tipos físicos. Una columna numérica puede ser una medida o un código; una columna textual puede ser una categoría, texto libre o un identificador. Por ello se distingue entre tipo físico y tipo semán-

tico, entendido como la naturaleza analítica de una columna (p. ej., cuantitativa, categórica, temporal o geográfica). También se consideran roles semánticos como identificador, moneda, porcentaje o estado, que aportan contexto sobre cómo interpretar la columna y qué operaciones son apropiadas (por ejemplo, evitar promediar identificadores). Esta distinción es crítica en cualquier sistema que recomiende visualizaciones: evita operaciones sin significado, restringe codificaciones visuales incoherentes y reduce sugerencias “correctas sintácticamente” pero erróneas conceptualmente.

De forma práctica, en esta memoria se distinguen tres niveles de significado, con implicancias directas para el motor:

1. **Esquema (estructura):** nombres de columnas, tipos físicos, tamaño, claves aparentes y relaciones potenciales. Por ejemplo, un patrón de unicidad alta o incremento monótono puede sugerir un identificador; un sufijo como “_id” o un conjunto alfanumérico de longitud fija puede indicar clave; y la presencia de columnas pares (latitud/longitud) puede indicar georreferenciación.
2. **Propiedades del contenido (estadística y calidad):** distribuciones, cardinalidad, faltantes, *outliers*, asimetría, concentración, medidas de relación y asociación, duplicidad. Este nivel permite detectar patrones explorables (p. ej., dominancia de categorías, colas largas) y advertir limitaciones (p. ej., alta proporción de nulos).
3. **Semántica de dominio (interpretación):** qué representa la columna en el mundo real, unidad/escala y relación conceptual con otras columnas. Aquí aparecen jerarquías *comuna*→*región*→*país*, familias de variables (financieras, demográficas) y equivalencias semánticas (sinónimos y alias).

La contribución conceptual relevante es que la solución no debe tomar decisiones importantes con un solo nivel: para recomendar y explicar de manera confiable, necesita combinar evidencia del esquema, de la estadística y de la semántica. Esta combinación es precisamente lo que permite que la recomendación sea útil y defendible: cada vista sugerida se puede justificar tanto por su validez (tipos y reglas) como por su interés (métrica) y por su pertinencia (intención/foco).

A partir de esta distinción, el problema deja de ser solo representar columnas en gráficos y pasa a ser un proceso de exploración guiada, donde las propiedades del *dataset* se transforman progresivamente en preguntas, hipótesis y vistas visuales.

2.3 Análisis Exploratorio de Datos

El Análisis Exploratorio de Datos (EDA) es el proceso iterativo de comprender un *dataset* antes de construir modelos finales o tableros. Incluye describir distribuciones, identificar

valores extremos, detectar inconsistencias, estudiar relaciones entre variables y formular hipótesis. Un rasgo esencial del EDA es que no es lineal: se avanza por ciclos cortos donde una observación conduce a una nueva pregunta, y esa nueva pregunta redefine qué vista es relevante observar (Figura 1) [Tukey, 1977].

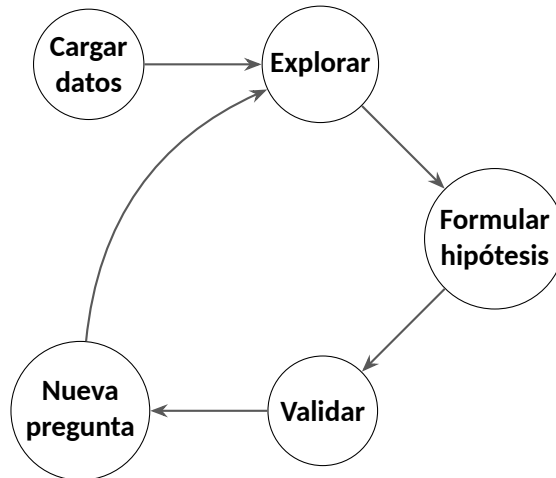


Figura 1: Ciclo iterativo del EDA.

En la práctica, el EDA alterna exploración abierta y análisis dirigido por preguntas. Esta memoria adopta esa dualidad como requisito de la solución:

1. **Modo abierto:** el usuario carga datos y requiere un panorama rápido de qué variables o combinaciones son informativas. Aquí el sistema debe funcionar como un “arranque” que reduce la fricción inicial: mostrar distribuciones básicas, comparaciones categóricas relevantes y algunas relaciones bivariadas para orientar preguntas.
2. **Modo dirigido:** el usuario formula una pregunta o declara un interés. Aquí la exploración debe adaptarse, priorizando métricas y tipos de vista coherentes con esa meta analítica.

La consecuencia conceptual es que un buen soporte a EDA no se limita a “mostrar una vista”, sino a orquestar un conjunto de vistas complementarias que cubran ángulos distintos. La orquestación responde a que el usuario no consume los gráficos como piezas aisladas: construye una interpretación al comparar vistas y al observar consistencias o contradicciones entre ellas. Por eso, diversidad y cobertura son parte fundamental del soporte al razonamiento, como se profundiza a continuación.

2.3.1 Diversidad y no-redundancia

En recomendación visual, no basta con mostrar las vistas de mayor puntaje individual. El conjunto final debe cubrir perspectivas distintas del *dataset*: distribuciones, relaciones, seg-

mentos o tendencias que se complementen entre sí. Por eso la diversidad no es un criterio estético, sino una condición para apoyar el razonamiento exploratorio sin saturar al usuario con variantes del mismo patrón.

Este equilibrio entre relevancia y novedad se formaliza en recuperación de información mediante *Maximal Marginal Relevance (MMR)* [Carbonell y Goldstein, 1998]. En visualización, la similitud puede estimarse por solapamiento de variables, tipo de gráfico y tarea analítica representada. Con ello, el sistema puede penalizar redundancia y reservar espacio para vistas que aporten evidencia distinta.

Sin embargo, una vista diversa no necesariamente es confiable. Antes de puntuar, comparar o recomendar visualizaciones, es necesario identificar señales de calidad, completitud y estructura del *dataset*. Esta necesidad introduce el perfilamiento de datos como componente conceptual del proceso exploratorio.

2.4 Perfilamiento y calidad de datos

En datos organizacionales, el EDA suele incorporar perfilamiento de datos (*data profiling*): extracción de resúmenes sistemáticos sobre completitud, consistencia, duplicidad, cardinalidad, concentración y patrones anómalos. Su valor es transformar un *dataset* “opaco” en señales interpretables para decidir qué comparar, filtrar o validar.

En esta memoria se distinguen dos niveles de perfilamiento:

1. **Perfilamiento descriptivo:** métricas que resumen (min/max, conteos, % faltantes, cardinalidad, frecuencias top- k , duplicados, rangos plausibles).
2. **Perfilamiento explicativo:** interpretaciones y recomendaciones derivadas (p. ej., “alta concentración sugiere segmentación adicional”, “cola larga sugiere inspección de *outliers*”, “faltantes condicionan comparaciones por grupo”).

Esta distinción es clave: una vista puede tener señal estadística, pero inducir interpretaciones erróneas si se apoya en datos incompletos o inconsistentes. Por eso, la calidad debe operar como restricción o penalización dentro del *scoring* y reflejarse luego en la narrativa.

Una vez establecidas las condiciones mínimas de calidad e interpretación de los datos, esas señales deben traducirse en representaciones visuales comprensibles. La visualización y la analítica visual entregan principios sobre percepción, interacción y razonamiento humano que orientan la selección de gráficos.

2.5 Visualización y analítica visual

La analítica visual combina métodos automáticos de análisis con interacción humana para apoyar razonamiento, exploración y toma de decisiones sobre datos [Keim *et al.*, 2008]. Conceptualmente, la visualización no es un producto final: es un medio de pensamiento, en tanto permite explorar señales, validar supuestos y comunicar hallazgos [Munzner, 2014].

Un principio central es que las visualizaciones deben privilegiar codificaciones perceptualmente efectivas. En términos prácticos, cuando la tarea implica comparación cuantitativa, codificaciones basadas en posición/longitud suelen ser preferibles a codificaciones basadas en área/ángulo, porque reducen error perceptual [Cleveland y McGill, 1984, Heer y Bostock, 2010, Mackinlay, 1986]. Para un motor automático, esto se traduce en reglas: no basta con generar muchas vistas; se debe seleccionar aquellas que tienen mejor legibilidad para la tarea analítica. Esta memoria adopta el criterio de efectividad perceptual como parte del “contrato” de la solución propuesta: la recomendación debe ser defendible no solo estadísticamente, sino también desde el diseño. La Figura 2 resume la jerarquía resultante.

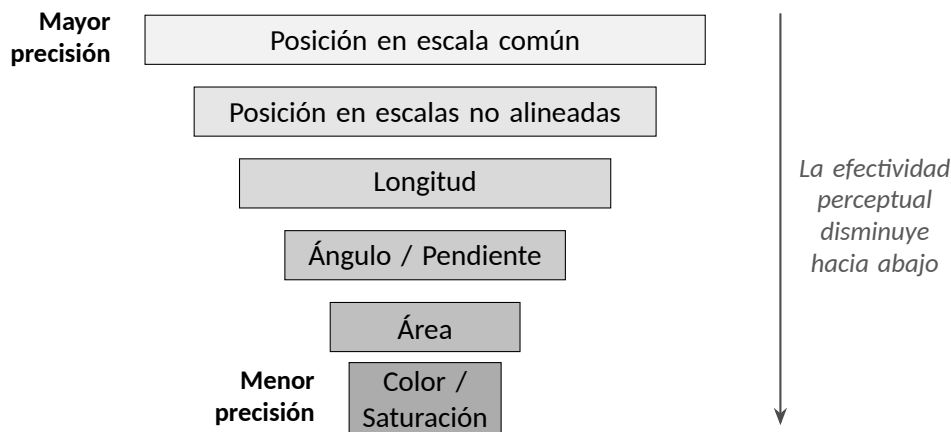


Figura 2: Jerarquía de efectividad perceptual de Cleveland y McGill [Cleveland y McGill, 1984].

Finalmente, la analítica visual presupone control humano: el usuario debe poder cuestionar, reinterpretar y decidir. Por eso la transparencia importa: si la solución sugiere una vista, debe ser posible responder “por qué esta vista es la primera”, “qué evidencia la respalda” y “qué alternativa cercana existía”. Esto conecta directamente con la auditoría.

2.5.1 Brecha semántica entre datos y visualización

Las secciones anteriores muestran que recomendar visualizaciones sobre datos tabulares no consiste solo en generar gráficos válidos. El sistema debe interpretar datos crudos, reconocer

significado semántico, seleccionar codificaciones visuales adecuadas y mantener trazabilidad sobre sus decisiones. Esta relación puede entenderse como una brecha semántica entre el dato disponible y una visualización significativa, como se resume en la Figura 3.

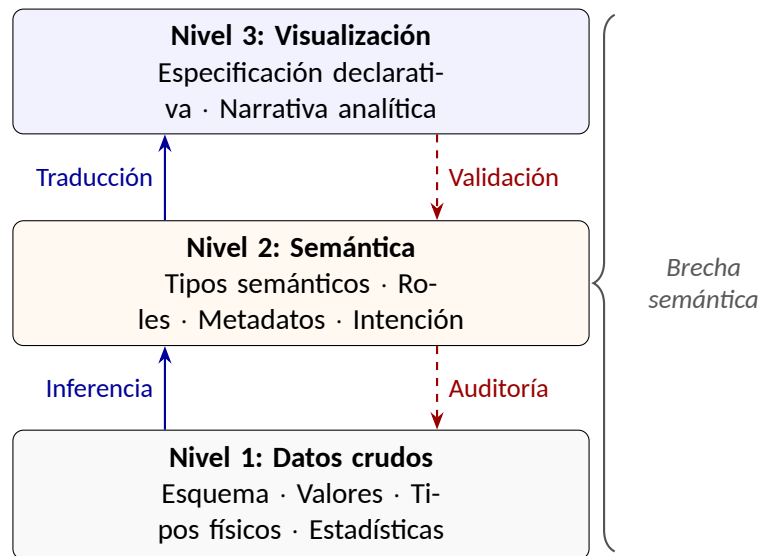


Figura 3: Brecha semántica entre datos y visualización.

La Figura 3 sintetiza el recorrido conceptual desarrollado hasta este punto: desde los datos crudos y sus propiedades observables, hacia una capa semántica que permite interpretar columnas, roles e intención, y finalmente hacia visualizaciones explicables. Las flechas descendentes representan la necesidad de validar y auditar las decisiones tomadas durante ese proceso.

Esta brecha permite leer los enfoques existentes según qué parte del recorrido abordan: algunos priorizan reglas y heurísticas, otros se apoyan en lenguaje natural, y otros combinan interpretación semántica con mecanismos de validación y control.

2.5.2 Lecciones desde herramientas existentes

En las herramientas existentes aparece la misma tensión entre facilidad de uso y control técnico. Power BI Copilot documenta limitaciones asociadas a calidad del modelo semántico y condiciones de uso [Microsoft, 2025]; Tableau Ask Data ofreció consulta en lenguaje natural, pero fue retirado [Tableau, 2024]. Estas experiencias muestran que el lenguaje natural reduce barreras de entrada, pero no resuelve por sí solo el problema completo: sigue dependiendo de modelos de datos bien definidos, validación de resultados, explicaciones confiables y mecanismos claros de trazabilidad.

Desde la perspectiva de esta memoria, dichas experiencias refuerzan una decisión metodológica: el lenguaje natural debe orientar la exploración, no reemplazar el cálculo verificable.

Por ello, el aporte no consiste en delegar toda la construcción de visualizaciones a un modelo generativo, sino en combinar un motor determinista auditable con una capa semántica y narrativa acotada. Esta cadena de dependencias puede visualizarse como un flujo vertical donde cada capa se apoya en la anterior y alimenta la siguiente. La Figura 4 resume la articulación entre los fundamentos conceptuales del capítulo.

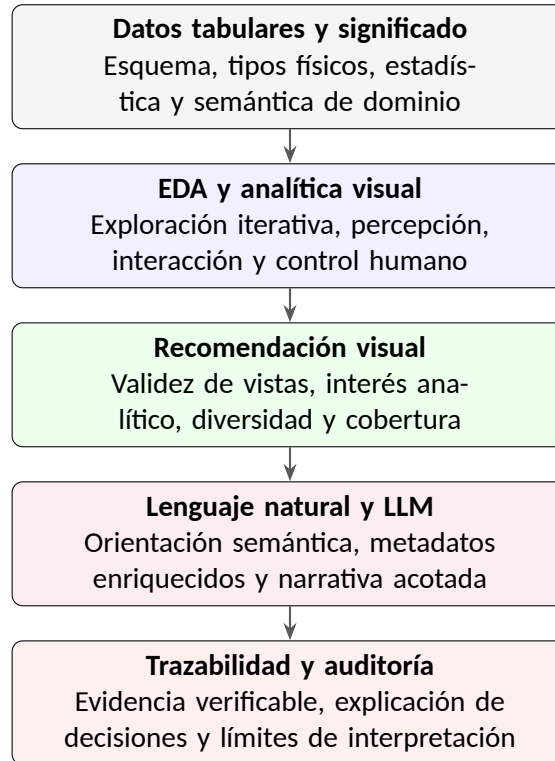


Figura 4: Articulación conceptual del marco teórico.

El mapa permite observar que la comprensión de los datos constituye la base sobre la que se construye el resto del análisis: sin un perfil confiable, las capas superiores (recomendación, lenguaje natural, auditoría) carecen de base verificable. Una vez establecidos los conceptos base del EDA y la visualización, las secciones siguientes avanzan hacia los mecanismos específicos que permiten operacionalizar el enfoque conceptual de la memoria: gramáticas declarativas, recomendación de visualizaciones, perfilamiento semántico, ponderación por intención, narrativa y auditoría.

2.6 Gramáticas declarativas de visualización y trazabilidad

Para automatizar generación y evaluación de gráficos, conviene usar representaciones declarativas: datos, transformaciones, marca, canales, agregación y escalas. Vega-Lite [Satyanarayan *et al.*, 2017] permite describir visualizaciones de forma inspeccionable y reproducible. Así, la evidencia no queda reducida a una imagen final, sino que se conserva

como una especificación auditable.

La ventaja conceptual es separar el qué se visualiza del cómo se dibuja. Una especificación puede declarar campos, agregaciones y codificaciones, y esa estructura permite validar reglas antes de renderizar: tipos incompatibles, agregaciones sin sentido o cardinalidades ilegibles pueden descartarse o ajustarse de forma explícita.

Además, la especificación declarativa funciona como contrato entre componentes: el motor genera candidatos, la interfaz los renderiza y la auditoría explica qué campos y transformaciones se usaron. Definida esta representación común, el problema siguiente es seleccionar qué especificaciones conviene generar y mostrar.

2.7 Recomendación de visualizaciones

Un sistema de recomendación de visualizaciones puede entenderse como una búsqueda guiada sobre un espacio amplio de vistas posibles. Conceptualmente, debe responder tres preguntas:

1. **¿Qué vistas son válidas?** Según tipos y reglas: por ejemplo, línea para temporalidad, barras para comparación categórica, dispersión para relación cuantitativa.
2. **¿Qué vistas son informativas?** Según métricas explícitas de *interestingness*: detectores de patrones explorables (asimetría, desigualdad, asociación, tendencia).
3. **¿Qué conjunto final se muestra?** Según diversidad: evitar redundancia y maximizar cobertura de columnas, tipos de vista y “ángulos” del *dataset*.

Como el espacio de vistas crece rápidamente con el número de columnas, la selección debe evitar redundancia y saturación. Lux [Lee *et al.*, 2022] es un referente porque integra intención, reglas de diseño y métricas de *interestingness* en flujos con *dataframes*. Para esta memoria, su valor es metodológico: muestra cómo generar candidatos y priorizarlos de forma interpretable, aunque no aborda por sí solo narrativa generativa, auditoría detallada ni separación explícita entre cálculo determinista y lenguaje.

Por ello, el *ranking* no debe entenderse como una ordenación puramente estadística. Debe equilibrar interés, validez, calidad del dato, diversidad y capacidad de explicación. No obstante, estos criterios requieren comprender no solo la estructura estadística de las columnas, sino también su significado en el dominio. De ahí la necesidad de un perfilamiento semántico con LLM, entendido como componente acotado a la interpretación de columnas, roles y relaciones.

2.8 Perfilamiento semántico con LLM

Más allá de inferir tipos por heurísticas, un LLM puede aproximar significado de dominio y generar metadatos interpretables. En *datasets* sin modelo semántico, esta capa ayuda a conectar la intención del usuario con columnas reales.

Esta memoria distingue cuatro capacidades útiles:

1. **Roles:** inferir si una columna representa cliente, producto, ubicación, riesgo, atraso, precio, etc.
2. **Metadatos:** describir columnas, unidades y supuestos de interpretación.
3. **Sinónimos:** mapear expresiones del usuario a campos reales.
4. **Relaciones:** sugerir jerarquías o familias de variables.

Este perfilamiento semántico complementa al estadístico: el primero intenta inferir qué representa una columna, mientras el segundo describe cómo se comportan sus valores. Como un LLM puede producir texto plausible pero incorrecto, su salida se trata como metadato auxiliar y auditable, no como fuente de verdad estadística. El motor determinista mantiene el control sobre cálculos, validez de vistas y selección final.

Con esta capa semántica disponible, la intención del usuario puede utilizarse para ajustar prioridades sin alterar los cálculos de base. Esta idea se entiende como una ponderación controlada sobre señales previamente calculadas.

2.9 Ponderación por intención

En motores de recomendación, una vista puede ser interesante por asimetría, desigualdad, tendencia o relación bivariada. El problema es que un *ranking* con pesos fijos puede ser correcto en términos generales, pero no necesariamente alinearse con la intención del usuario.

La propuesta conceptual es traducir la intención textual a prioridades verificables. Por ejemplo:

- Si el usuario pregunta por “equidad” o “distribución desigual”, se priorizan señales de concentración/desbalance.
- Si el usuario pregunta “qué influye en X”, se priorizan relaciones bivariadas (correlaciones o asociaciones) y vistas que incluyan explícitamente *X*.
- Si el usuario pregunta por “atrasos”, se prioriza la selección semántica de variables cercanas a atraso y las vistas temporales o por segmento para localizar variación.

Así, la intención no ejecuta consultas libres: modifica prioridades del *ranking*. Conceptualmente, opera como una política de reponderación sobre señales ya calculadas: puede aumentar el puntaje de vistas que incluyan una columna foco, favorecer ciertos tipos de análisis o penalizar vistas lejanas a la intención. Como la modificación se expresa mediante factores explícitos, el sistema puede explicar no solo por qué una vista era informativa, sino también por qué fue priorizada para una pregunta concreta.

La priorización, sin embargo, no basta por sí sola: el usuario también necesita comprender qué muestra una vista, qué evidencia la respalda y qué limitaciones presenta. En ese punto, la narrativa analítica cumple un rol explicativo, siempre que permanezca anclada a hechos calculados.

2.10 Narrativa anclada a hechos

Una narrativa analítica útil no solo describe un gráfico: resume el hallazgo, explicita evidencia, declara limitaciones y sugiere un siguiente paso razonable. En EDA, esta narrativa ayuda a transformar una vista aislada en una interpretación accionable, conectando patrones visuales con preguntas posteriores.

Para evitar explicaciones no verificables, la narrativa debe estar anclada a hechos calculados. El motor determinista produce estadísticas, métricas, puntajes, validaciones y transformaciones aplicadas; el modelo generativo redacta lenguaje condicionado por esos hechos, sin inventar números ni alterar decisiones del motor. Esta separación permite usar la capacidad expresiva del LLM sin convertirlo en fuente autónoma de verdad.

También es relevante distinguir entre narrativa descriptiva y prescriptiva. La primera comunica qué se observa; la segunda sugiere cómo profundizar o validar el hallazgo. En esta memoria, la prescripción debe ser conservadora y coherente con evidencia disponible. Para que esta explicación sea confiable, debe poder rastrearse hasta las decisiones y evidencias que la originan, lo que exige una capa de auditoría.

2.11 Auditoría de decisiones

Para evitar una recomendación tipo “caja negra”, se requiere una capa de auditoría que responda por qué se seleccionó una vista, qué evidencia la respalda y por qué se descartó una alternativa. Conceptualmente, esta auditoría se apoya en dos insumos:

1. **Evidencia cuantitativa:** métricas de *interestingness*, adecuación por tipos, penalizaciones por calidad, y mecanismos de diversidad (anti-redundancia).
2. **Principios de diseño:** reglas perceptuales y buenas prácticas (p. ej., priorizar longitud/posición en comparaciones y evitar tortas cuando se requiere precisión).

Para mejorar inspección y usabilidad, la evidencia numérica puede acompañarse de una explicación breve en lenguaje natural (por ejemplo, “seleccionado por alta asimetría” o “alta concentración en pocas categorías”), derivada de forma determinista a partir de la métrica principal y su magnitud. Este principio de explicabilidad por diseño se alinea con marcos de inteligencia artificial explicable [Phillips *et al.*, 2021] y traduce señales técnicas a una justificación comprensible.

Finalmente, aunque el lenguaje natural aparece asociado al perfilamiento semántico, la ponderación por intención y la narrativa, conviene precisar su rol general dentro del análisis visual: orientar la interacción sin reemplazar las decisiones que deben permanecer bajo control determinista.

2.12 Lenguaje natural para análisis visual

Las interfaces en lenguaje natural reducen barrera técnica porque permiten expresar una intención analítica sin conocer una sintaxis de consulta ni una herramienta específica. Sin embargo, también introducen ambigüedad: una misma pregunta puede tener varias interpretaciones, una palabra puede referirse a distintas columnas y el usuario puede usar sinónimos o conceptos de dominio que no aparecen literalmente en el *dataset*.

En esta memoria, el lenguaje natural se usa como señal de orientación: identifica metas analíticas y columnas foco, pero no ejecuta consultas libres ni define resultados. Este acotamiento es deliberado. Al usarlo como mecanismo de priorización, la solución aprovecha su valor semántico sin delegarle decisiones críticas. Así, la interacción natural mejora acceso y pertinencia, mientras las reglas deterministas mantienen control sobre el análisis.

2.13 Desarrollo de aplicación web para analítica visual

En esta memoria, la solución no se limita a definir un algoritmo de recomendación: se materializa como una aplicación web de analítica visual. Esto exige traducir los conceptos anteriores en un flujo usable, donde el usuario pueda cargar datos, recibir una caracterización inicial, explorar visualizaciones sugeridas y revisar la evidencia que respalda cada resultado. La aplicación, por lo tanto, debe resolver tanto el procesamiento interno como la forma en que ese procesamiento se vuelve comprensible durante la interacción.

Para lograrlo, la aplicación necesita construir artefactos intermedios que ordenen el análisis. Entre ellos se encuentran el perfil del *dataset*, el catálogo semántico, las vistas candidatas, los puntajes de priorización, las narrativas y los descartes. Estos artefactos cumplen una función doble: por una parte, permiten dividir el proceso en etapas claras; por otra, dejan evidencia suficiente para explicar qué ocurrió en cada paso. Así, el diseño de datos no se reduce al archivo cargado por el usuario, sino que incluye las representaciones internas que hacen

posible recomendar, explicar y auditar.

La interfaz debe hacer visible ese proceso sin sobrecargar al usuario. En una aplicación orientada a EDA, el flujo de uso debe permitir avanzar desde una acción simple —cargar un archivo— hacia resultados inspeccionables: resumen inicial, sugerencias visuales, explicación y panel de transparencia. Esto se relaciona con principios de usabilidad y visualización de información, donde se busca reducir carga cognitiva y mantener el control humano sobre la exploración [Ware, 2013]. La interfaz, en este sentido, no es solo una capa de presentación: organiza la experiencia de análisis y define qué tan fácil resulta revisar o cuestionar una recomendación.

Desde esta perspectiva, el desarrollo de la aplicación conecta tres aspectos: la organización interna del análisis, la forma en que los resultados se presentan al usuario y los criterios con que luego se valida la solución. No basta con que la aplicación ejecute correctamente; también debe producir recomendaciones pertinentes, responder en tiempos compatibles con una exploración iterativa y mantener las narrativas ancladas a evidencia calculada. Por ello, el capítulo siguiente describe la propuesta como un flujo de software completo, desde los requisitos y la arquitectura hasta la interacción y la validación experimental.

CAPÍTULO 3

PROPUESTA DE SOLUCIÓN

A partir del problema definido previamente y de los fundamentos conceptuales revisados, esta propuesta presenta una aplicación web interactiva para apoyar las primeras etapas del Análisis Exploratorio de Datos (EDA) sobre datos tabulares. La aplicación permite cargar archivos CSV o Excel y, opcionalmente, declarar una intención de análisis en lenguaje natural. Desde esa entrada, el sistema construye un perfil estadístico y semántico del *dataset*, genera visualizaciones candidatas, prioriza un conjunto acotado de recomendaciones y acompaña cada resultado con una explicación textual basada en evidencia calculada.

El resultado de la aplicación no es una colección de gráficos, sino una sesión de análisis compuesta por artefactos complementarios: perfil de datos, catálogo semántico, visualizaciones recomendadas, criterios de selección, narrativas explicativas y registros de auditoría. Esta organización busca reducir la fricción inicial del EDA sin reemplazar el juicio del analista, manteniendo visibles las decisiones relevantes del sistema.

Para lograrlo, la solución se implementa como un *pipeline* neuro-simbólico acotado. El componente determinista concentra las operaciones verificables —perfilamiento, generación de candidatos, cálculo de métricas, *ranking*, selección diversa y auditoría—, mientras que el componente generativo se utiliza solo para enriquecer metadatos y redactar explicaciones a partir de hechos ya calculados. Esta separación permite aprovechar lenguaje natural para interpretación y comunicación sin delegarle al LLM el cálculo ni la decisión final de las visualizaciones.

3.1 Análisis de requisitos de la aplicación

Antes de definir la arquitectura, se establecen los requisitos que orientan el desarrollo de la aplicación web. Estos requisitos se derivan del problema identificado: apoyar las primeras etapas del EDA cuando el usuario enfrenta un *dataset* tabular y necesita un punto de partida comprensible, verificable y rápido para explorar sus datos.

Desde el punto de vista funcional, la aplicación debe cubrir cuatro capacidades principales: recibir datos tabulares, construir una caracterización inicial del *dataset*, recomendar visualizaciones priorizadas y entregar evidencia que permita revisar esas recomendaciones. Estas capacidades definen el flujo principal de uso: ingreso de datos, análisis automático, presentación de resultados e inspección posterior.

También existen requisitos no funcionales que condicionan el diseño: trazabilidad, separación entre cálculo y lenguaje natural, minimización de datos compartidos con el LLM, legibilidad visual y tiempos compatibles con una exploración iterativa.

3.2 Arquitectura general y principios de diseño

La propuesta se guía por cuatro principios no funcionales. El primero es separar estrictamente cálculo y lenguaje: el cálculo determinista (inferencia semántica, perfilamiento, estadísticas, métricas, *ranking* y selección de vistas) queda separado de la redacción. El rol del LLM es enriquecer metadatos del catálogo (descripción/sinónimos/unidad) y transformar hechos ya calculados en lenguaje natural para las narrativas. El LLM no calcula métricas ni decide el *ranking*; a lo más puede influir indirectamente en el mapeo de intención al ampliar sinónimos del catálogo. Esta separación se alinea con la taxonomía de sistemas neuro-simbólicos propuesta por Kautz, donde los módulos simbólicos y neuronales se acoplan de forma modular para combinar exactitud lógica y flexibilidad semántica [Kautz, 2020].

El segundo principio es la privacidad por diseño y la minimización de datos compartidos con el LLM. La aplicación web envía metadatos del catálogo y evidencia compacta derivada del análisis (p. ej. valores frecuentes/top- k por columna cuando aplica o percentiles), no el *dataset* completo. Este enfoque reduce exposición, pero no implementa anonimización automática; por lo tanto, si el *dataset* contiene información sensible, se recomienda usar un LLM local. Esto se alinea con recomendaciones de gobernanza/gestión de riesgo en IA [NIST, 2023].

El tercer principio es tratar la usabilidad y la legibilidad como restricciones de diseño. La exploración debe producir salidas inspeccionables por humanos: evitar sobrecarga visual (demasiadas categorías, *bins* excesivos, ejes saturados) y evitar evidencia numérica frágil (métricas calculadas con muy pocos datos). Por ello, la aplicación web mantiene restricciones orientadas a legibilidad (p. ej. top- k en barras, máximo de categorías por eje en mapas de calor, máximo de *bins* en histogramas) y validez mínima (mínimo de observaciones válidas para ciertas métricas), sin alterar el cómputo determinista ni “inventar” resultados.

Finalmente, la trazabilidad se trata como una salida de primera clase. Cada recomendación entrega evidencia: métrica principal y su valor, calidad de dato utilizada, ajustes por intención/foco, y razones de descarte/penalización por redundancia [Phillips et al., 2021].

La Figura 5 muestra la arquitectura general y el flujo principal de la solución. La interfaz de usuario concentra la interacción inicial, mientras que el núcleo de análisis organiza las etapas de perfilamiento, priorización, generación de explicaciones y auditoría. En este flujo, el componente generativo actúa como apoyo acotado: contribuye al enriquecimiento semántico y a la redacción de explicaciones, pero no interviene en el cálculo de métricas ni en la selección de visualizaciones.

3.3 Paso 0: Ingesta y perfilamiento estadístico-semántico

Al cargar un *dataset* tabular, la aplicación web realiza un perfilamiento por columna para obtener una “radiografía” inicial de los datos. El resultado resume, para cada variable, su

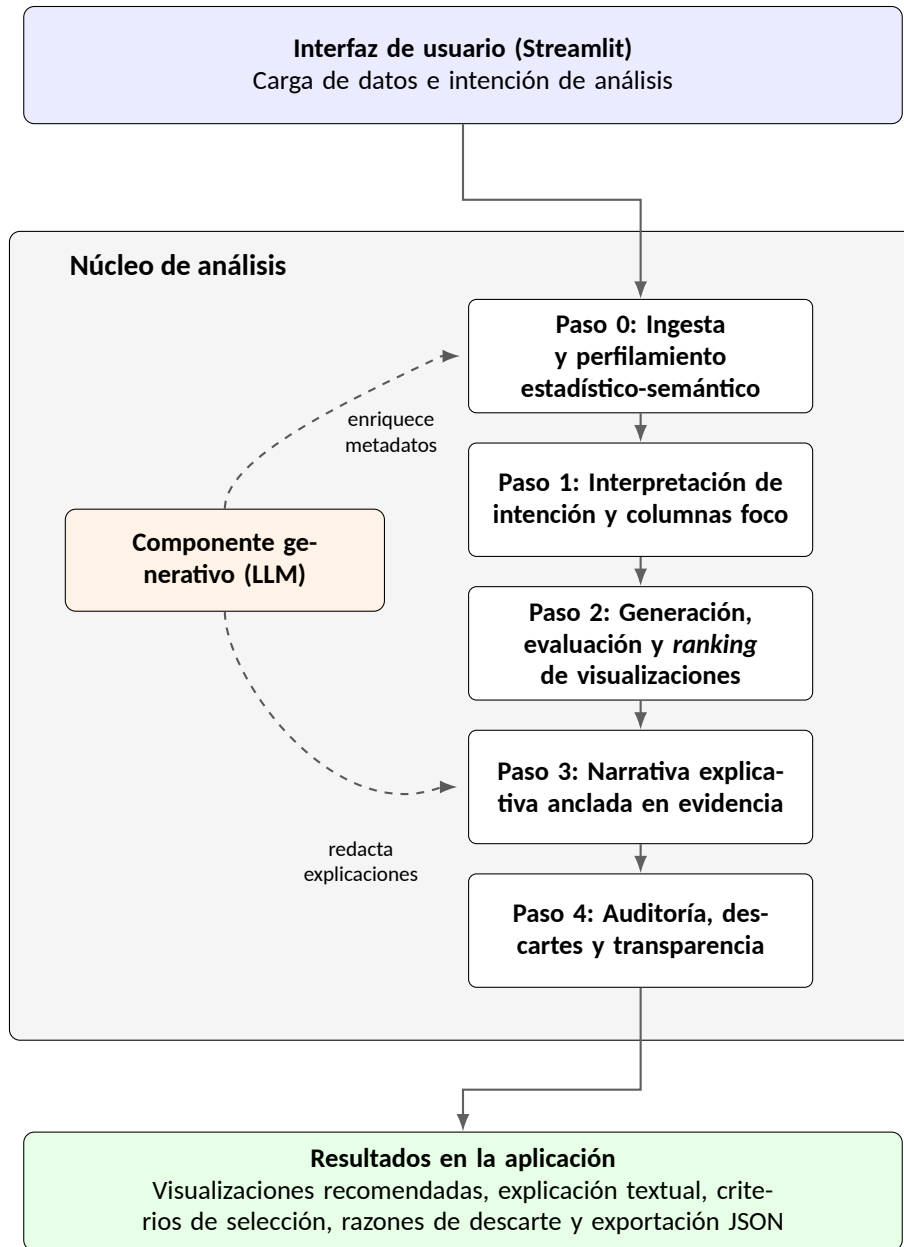


Figura 5: Arquitectura general de la solución propuesta.

completitud (proporción no nula), su cardinalidad (cantidad/proporción de valores distintos) y estadísticas acordes al tipo de dato: en columnas categóricas, distribución de frecuencias y categorías dominantes (mostradas con top- k por legibilidad); en columnas cuantitativas, rangos y percentiles que describen dispersión y extremos; y en temporales, rango temporal. Este perfil actúa como base factual de la aplicación web: alimenta la generación y filtrado de visualizaciones, y entrega evidencia verificable que luego se reutiliza para explicar los hallazgos.

El Paso 0 produce artefactos estructurados que se reutilizan en los pasos siguientes. La Tabla 2 resume qué información se guarda y con qué propósito.

Producto	Qué contiene
Tipificación semántica por columna	Para cada columna se asigna un tipo semántico principal: <i>cuantitativa</i> , <i>nominal/categorica</i> , <i>temporal</i> , <i>geográfica</i> o <i>desconocida</i> . Este tipo determina qué visualizaciones son pertinentes en pasos posteriores.
Metadatos de calidad y contexto por columna	Se registran señales para decidir si una columna es utilizable y cómo tratarla: proporción no nula (completitud), cantidad/proporción de valores distintos (cardinalidad) y pistas semánticas simples (p. ej. si parece identificador, porcentaje, moneda o estado). En columnas geográficas, se puede registrar una jerarquía sugerida a partir de nombres (p. ej. local→región→país) para auditoría.
Perfil global del <i>dataset</i> (resumen + alertas)	Tamaño del <i>dataset</i> y señales globales útiles para EDA (por ejemplo, proporción de nulos y duplicados detectados por el perfil). Incluye alertas de calidad para advertir sobre interpretaciones frágiles.
Perfil estadístico por columna (evidencia numérica)	Un resumen por columna que actúa como “base factual”. <i>Cuantitativas</i> : rangos y percentiles (p. ej. mediana y percentiles altos) para entender dispersión y extremos. <i>Nominales/Geográficas</i> : top- k categorías más frecuentes y sus frecuencias para ver dominancia/concentración (k acotado para mantener legibilidad). <i>Temporales</i> : rango temporal (mínimo/máximo) cuando el parseo es exitoso.
Catálogo semántico por columna (descripción y vocabulario)	Una ficha por columna con: tipo semántico, rol analítico — <i>medida</i> , cuando representa una magnitud agregable, o <i>dimensión</i> , cuando agrupa el análisis—, descripción breve, sinónimos (para reconocer columnas desde intención en lenguaje natural) y unidad cuando aplica (p. ej. %, moneda). Este catálogo se enriquece sistemáticamente con un LLM para robustecer descripción/sinónimos/unidad; este aporte no altera la base estadística ni las reglas deterministas de <i>scoring</i> , pero puede mejorar el mapeo entre intención y columnas; dicha selección se refleja en las columnas foco detectadas y en los ajustes por intención/foco expuestos en la auditoría.

Tabla 2: Productos del Paso 0 y su evidencia asociada.

La Figura 6 muestra cómo estos productos aparecen en la interfaz usando como ejemplo el *dataset* HR, un conjunto sintético de recursos humanos con variables de salario, desempeño, contrato, región y capacitación. Este conjunto se presenta aquí solo para ilustrar la interfaz; sus características y uso experimental se describen con mayor detalle en la validación de la solución. El resumen del *dataset* entrega una lectura rápida de la sesión, mientras que el catálogo semántico deja visible la interpretación que la aplicación realiza sobre las columnas antes de recomendar visualizaciones.

Resumen del dataset (N1)

Dataset con 200000 filas y 28 columnas. Nulos aprox. (muestra) 9%. Tipos: 2 cuantitativas, 21 categóricas, 1 temporales, 4 geográficas. Alertas: Columnas con muchos nulos: return_reason (≈96% nulos), promo_code (≈82% nulos), delivery_days (≈67% nulos).

Intención detectada

```
{
  "mode": "free"
  "goals": [...],
  "focus_fields": [],
  "target_fields": [],
  "matched_fields": {}
}
```

Detalles técnicos

Catálogo semántico							
Perfilamiento semántico automático.							
	columna	tipo_semantico	rol	rol_semantico	unidad	descripcion	sinonimos
0	transaction_id	nominal	dimension	identifier		Identificador único de cada transacción de compra.	ID Transacción, Código de Ver
1	customer_id	nominal	dimension	identifier		Identificador único de cada cliente.	ID Cliente, Código de Cliente
2	customer_segment	nominal	dimension	dimension		Segmento al que pertenece el cliente (ej. Masivo, PyME, Premium).	Segmento Cliente, Categoría
3	gender	nominal	dimension	dimension		Género del cliente.	Sexo
4	age	nominal	dimension	dimension	años	Edad del cliente.	Años
5	region	geographic	dimension	dimension		Región geográfica donde se realizó la transacción o reside el cliente.	Zona Geográfica
6	comuna	geographic	dimension	dimension		Comuna geográfica donde se realizó la transacción o reside el cliente.	Localidad

Figura 6: Resumen inicial y catálogo semántico de la aplicación web, usando como ejemplo el *dataset* HR de recursos humanos.

3.4 Paso 1: Interpretación de intención y columnas foco

Una vez cargado el *dataset* y construido el catálogo semántico, la aplicación web determina cómo orientar la exploración. Este paso no genera visualizaciones ni calcula métricas: define la estrategia de búsqueda según el contexto del usuario.

3.4.1 Modo dirigido y modo de exploración libre

La aplicación web soporta dos modos. En modo dirigido, el usuario formula una intención/pregunta y esa información se usa para priorizar recomendaciones alineadas con lo que desea observar. En modo libre, no hay una pregunta explícita, por lo que la aplicación web debe entregar un punto de partida general y seguro para iniciar el ciclo de EDA, sin sesgar la exploración hacia una hipótesis específica.

Operativamente, en modo libre no se aplican ajustes por intención: las recomendaciones se ordenan según la señal estadística de cada vista, la calidad del dato y la selección diversa. La reponderación guiada por objetivos y columnas foco queda reservada para el modo dirigido.

3.4.2 Metas analíticas detectadas y foco de columnas

Cuando hay intención textual, la aplicación web la traduce a:

- **Metas analíticas** (*goals*). Son etiquetas que representan qué tipo de *insight* busca el usuario y sirven para priorizar métricas coherentes con ese objetivo:
 - *distribution*: entender la forma de una variable (concentración, colas, asimetría, dispersión). Es la meta por defecto: cuando no hay intención específica, el motor opera en modo neutro priorizando cualquier señal estadística destacada sin sesgo hacia un tipo de patrón.
 - *outliers*: encontrar valores atípicos o anomalías (casos extremos que merecen inspección).
 - *inequality*: detectar desbalance o dominancia (por ejemplo, pocas categorías concentran gran parte del total).
 - *trend*: identificar evolución temporal (tendencia creciente/decreciente o cambios sostenidos).
 - *relationship*: estudiar relación entre variables (asociación entre dos cuantitativas, o dependencia entre categorías).

La detección se realiza por palabras clave, por lo que el proceso es reproducible y auditable.

- **Columnas foco** (*focus_fields*). Son las columnas más relacionadas con la pregunta del usuario. Se detectan por coincidencia entre el texto, los nombres de columnas y los sinónimos del catálogo. En la implementación, se seleccionan como columnas foco todas las columnas cuyo *score* de coincidencia supera un umbral mínimo (0,4 por defecto), ordenadas por dicho *score*. El umbral filtra coincidencias débiles (p. ej. solapamientos accidentales de tokens) pero permite menciones parciales o sinónimos razonables.

El resultado de este paso es un plan de priorización que indica qué buscar y dónde buscar. Este plan no altera métricas ni cálculos del motor; solo condiciona la exploración posterior mediante reponderación y selección de columnas/candidatos.

3.4.3 Elegibilidad de columnas y priorización a nivel de vistas

Si un *dataset* tiene p columnas, el número de combinaciones bivariadas posibles crece como $\mathcal{O}(p^2)$. Para mantener el flujo trazable y centrado en evidencia, la aplicación web separa: qué columnas son elegibles (reglas deterministas de validez/interpretabilidad) y cómo se priorizan los hallazgos (*scoring* por vista y selección diversa).

Antes de instanciar vistas, se aplican reglas explícitas para evitar candidatos no informativos o engañosos:

- **Columnas completamente nulas:** se excluyen por ausencia total de evidencia.
- **Identificadores:** se excluyen por defecto (baja utilidad analítica), salvo que el usuario los mencione explícitamente en la intención; en ese caso se habilitan para no bloquear exploración guiada.
- **Complejidad mínima por vista:** además de excluir columnas completamente nulas, la aplicación web descarta una vista candidata si alguna de sus columnas requeridas tiene completitud (proporción no nula) menor a un umbral configurable. En esta implementación, el umbral se fija en (0,6), para evitar que métricas y gráficos se basen en evidencia excesivamente incompleta.

3.5 Paso 2: Generación, evaluación y *ranking* de visualizaciones

Una vez definidas las columnas, la aplicación web ejecuta un motor de recomendación visual inspirado en Lux [Lee *et al.*, 2022] y en la línea de sistemas de recomendación exploratoria como Voyager [Wongsuphasawat *et al.*, 2016]. El motor: genera vistas candidatas coherentes con tipos semánticos y reglas explícitas, computa una métrica principal por vista (*interestingness*), calcula un *score* determinista y selecciona un conjunto final diverso (MMR).

Las vistas se representan internamente como especificaciones declarativas tipo Vega-Lite (vía Altair), lo que permite inspeccionar el “qué” se graficó (variables, agregación y codificación) y mantener trazabilidad del diseño. En la exportación del análisis se preserva evidencia estructurada: variables usadas, agregaciones y transformaciones aplicadas (*binning*, *top-k*, agregación/reescalamiento temporal), junto con métricas y puntajes.

3.5.1 Generación sistemática de vistas candidatas (univariadas y bivariadas)

A partir de las columnas, el motor instancia vistas candidatas de manera determinista según tipos semánticos y reglas explícitas. La enumeración completa de todas las combinaciones posibles puede producir resultados redundantes y sobrecargar la lectura; por ello, la aplicación web incorpora restricciones de legibilidad (*top-k*, *bins* máximos, categorías máximas por eje) y requisitos de evidencia mínima (mínimo de observaciones válidas, umbral de completitud 0,6) antes de considerar ciertas vistas.

A partir del perfilamiento semántico, el motor particiona C en tres subconjuntos: Q (cuantitativas), N (nominales/catóricas) y T (temporales). Esta partición impone restricciones de validez (qué variables pueden combinarse) y guía la selección de familias de vistas (qué

codificaciones tienen sentido para cada tipo). La instanciación aplica reglas explícitas y defendibles, coherentes con tipos semánticos y principios clásicos de efectividad perceptual [Cleveland y McGill, 1984, Mackinlay, 1986, Munzner, 2014].

Bajo los criterios anteriores, el motor genera las siguientes familias de vistas:

■ Univariadas:

- Para cada $q \in Q$ se instancian:
 - **Histograma (X cuantitativo).** Para caracterizar distribución. El *binning* se controla con un máximo de 30 *bins* (límite superior para evitar sobrefragmentación y ruido visual). Esta idea es consistente con la literatura de elección de *bins* [Scott, 1979, Freedman y Diaconis, 1981], donde el número de *bins* debe balancear resolución y estabilidad.
 - **Boxplot (X cuantitativa).** Para exponer dispersión y *outliers* usando la regla de Tukey (IQR 1,5) [Tukey, 1977].
- Para cada $n \in N$ se instancian:
 - **Barras de frecuencia (X nominal).** Para detectar concentración/dominancia. Se usa *top-k* con $k = 25$ por defecto para evitar gráficos con demasiadas categorías (limitación de legibilidad y comparación) [Cleveland y McGill, 1984, Ware, 2013].

■ Bivariadas:

- $N \times Q$. Para cada par $(n, q) \in N \times Q$ se instancian:
 - **Barras agregadas (X nominal $\rightarrow Y$ cuantitativo).** Se instancian barras agregadas calculando suma/promedio por categoría, para comparar magnitudes entre grupos. Se usa *mean* o *sum* según heurística del rol semántico (p. ej. montos $\rightarrow$ *sum*, porcentajes $\rightarrow$ *mean*) y se limita a *top-k* categorías para preservar comparabilidad [Cleveland y McGill, 1984, Mackinlay, 1986]. Para mantener legibilidad y comparabilidad, en vistas categóricas se limita el número de categorías mostradas (*top-k*, por defecto 25).
 - **Boxplot (X nominal $\rightarrow Y$ cuantitativa).** Cuando la cardinalidad es manejable, se instancian *boxplots* por categoría para comparar distribuciones entre grupos. En este caso, además se exige un mínimo de observaciones por grupo (por defecto 5) y al menos dos categorías válidas; esto evita *boxplots* dominados por grupos demasiado pequeños y reduce interpretaciones engañosas.
- $T \times Q$. Para cada par $(t, q) \in T \times Q$ se instancian:
 - **Línea (X temporal $\rightarrow Y$ cuantitativo).** Se instancian líneas agregando Y por intervalos temporales. La granularidad del intervalo puede seleccionarse en modo *auto* a partir del rango temporal observado (por ejemplo diario/semanal/mensual/trimestral), de modo que la serie resulte legible y la

métrica de tendencia se calcule sobre una secuencia agregada. Esta transformación queda registrada como parte de la evidencia y se reutiliza al renderizar la vista.

La granularidad (tamaño del *bin*) se elige automáticamente según el *span*² temporal del *dataset*: si el rango cubre pocos días, se agregan valores a nivel diario; si cubre semanas o meses, se agregan a nivel semanal o mensual; y si cubre varios meses/años, se agregan a nivel trimestral. Con ello, la serie mantiene una cantidad de puntos manejable, preserva la lectura de tendencia y evita tanto la sobrecarga (demasiados puntos) como la pérdida excesiva de detalle (agregación demasiado gruesa) [Heer y Shneiderman, 2012].

- $Q \times Q$. Evaluar todas las dispersiones $\binom{|Q|}{2}$ puede introducir redundancia cuando muchas variables están débilmente relacionadas. Por ello, para $(q_i, q_j) \in Q \times Q$:
 - **Dispersión (X cuantitativo $\times Y$ cuantitativo).** Se instancian dispersiones para estudiar relación monotónica. El motor utiliza ρ_S^2 como señal de *fuerza* de asociación (acotada en $[0, 1]$) y conserva ρ_S como evidencia para distinguir dirección. Además, se exige un mínimo de observaciones válidas por par; si no se alcanza ese mínimo, el par no se considera para evitar estimaciones frágiles.
- $N \times N$. El número de pares $\binom{|N|}{2}$ puede ser alto y, además, mapas de calor con demasiadas categorías por eje se vuelven ininterpretables y generan tablas de contingencia muy dispersas. Por ello, para $(n_i, n_j) \in N \times N$ se instancian:
 - **Mapa de calor (X nominal $\times Y$ nominal).** Se instancian mapas de calor de contingencia para asociación cuando ambas cardinalidades son acotadas [Munzner, 2014, Ware, 2013]. Para preservar legibilidad, se acota el número de categorías por eje (por defecto hasta 12). Sobre la tabla de contingencia resultante, se usa Cramér's V como medida de asociación [Cramér, 1946]. Además, se exige un mínimo de observaciones válidas por par para evitar conclusiones dominadas por conteos muy pequeños; si no se cumple, la vista se descarta por evidencia insuficiente.

Los límites de cardinalidad (top- k en barras y máximo de categorías por eje en mapas de calor), el máximo de *bins* en histogramas y los mínimos de evidencia (p. ej. mínimo de observaciones válidas) se utilizan para mantener legibilidad y evitar estimaciones frágiles. Estos valores son configurables.

Cabe notar que la aplicación web puede inferir el tipo semántico *geográfico* (y metadatos como jerarquía local/región/país) como parte del perfil y del catálogo. En esta propuesta, el motor no genera visualizaciones geoespaciales (p. ej. mapas) ni instancias específicas para variables etiquetadas como geográficas; su rol es aportar contexto y trazabilidad semántica.

²En este contexto, el *span* temporal corresponde a la extensión total del intervalo cubierto por la variable temporal, es decir, la diferencia entre el instante mínimo y máximo observado en el *dataset*.

3.5.2 Métricas principales por tipo de vista (definiciones y sentido analítico)

Cada vista candidata v tiene una métrica principal coherente con lo que ese tipo de gráfico puede revelar de forma natural. La aplicación web conserva tanto el valor crudo como evidencia (`metric_value`) como su versión normalizada en $[0, 1)$ como `metric_score` para *ranking*.

Para comparar señales heterogéneas sin perder resolución por saturación, cada función de *scoring* convierte el valor crudo $m(v)$ en un *score* normalizado mediante una función saturante de la forma:

$$s_m(v) = \frac{m(v)}{m(v) + C},$$

donde $C > 0$ es una constante de semi-saturación específica para cada métrica: el *score* alcanza 0,5 cuando $m(v) = C$. Esta función es análoga a la saturación usada en BM25 para evitar que una señal extrema domine completamente el *ranking* [Robertson y Zaragoza, 2009].

La función mapea $[0, \infty)$ a $[0, 1)$, mantiene el orden relativo de los valores crudos, conserva resolución en señales moderadas y suaviza el efecto de valores muy altos. La elección de C se fundamenta en el punto donde cada métrica indica una señal “moderadamente interesante”. La Tabla 3 resume los valores utilizados; su justificación detallada se presenta en el Anexo A.

Métrica	Vista	C
skewness	histogram	1,5
outlier_ratio	boxplot univ.	0,05
median_spread	boxplot biv.	1,0
unevenness	bar	0,3
trend_spearman	line	0,3
monotonicity	scatter	0,3
association	mapa de calor	0,3

Tabla 3: Constantes de semi-saturación C por métrica.

Las métricas de cada tipo de gráfico son las siguientes:

S1. Asimetría en histograma (forma de distribución). El histograma es el gráfico canónico para inspeccionar forma de distribución (colas, sesgo, concentración). La asimetría muestral $\text{skew}(X)$ resume en un escalar cuán “cargada” está una cola respecto a la otra, lo que es útil para detectar colas pronunciadas, posibles mezclas de subpoblaciones o necesidad de transformaciones (p. ej. log). Esta métrica es estándar para cuantificar asimetría [Joanes y Gill, 1998]. En la aplicación web se guarda el $\text{skew}(X)$ firmado como evidencia y se usa su magnitud para *scoring*:

$$m(v) = |\text{skew}(X)|.$$

Interpretación: valores bajos (≈ 0) sugieren distribución aproximadamente simétrica; valores medios indican asimetría moderada (potenciales subgrupos o desigualdad); valores altos implican colas pronunciadas y mayor probabilidad de valores extremos (posibles *outliers* o fenómenos altamente desbalanceados).

S2. Valores atípicos robustos en *boxplot* univariado (anomalías/colas). El *boxplot* fue diseñado para mostrar mediana, dispersión robusta y *outliers* de manera compacta, usando la regla de Tukey (IQR 1,5) [Tukey, 1977]. Por ello, una señal natural es la proporción de observaciones fuera de las “vallas”:

$$\text{out_ratio}(X) = \mathbb{P}(X < Q_1 - 1,5 \cdot IQR \vee X > Q_3 + 1,5 \cdot IQR),$$

$$m(v) = \text{out_ratio}(X).$$

Interpretación: valores bajos indican pocos casos atípicos (distribución “estable”); valores medios sugieren colas relevantes (puntos a inspeccionar); valores altos implican alta densidad de *outliers* (posibles anomalías, errores de captura o colas pesadas).

S3. Separación de medianas en *boxplot* nominal $\rightarrow$ cuantitativa (heterogeneidad entre grupos). Cuando el objetivo es comparar distribuciones por categoría, el *boxplot* por grupo permite ver diferencias sistemáticas y robustas. En ese contexto, una señal directa es cuán separadas están las medianas entre categorías (usando top- k y un mínimo de muestras por grupo para evitar conclusiones frágiles). Se define:

$$m(v) = \text{spread_norm} = \frac{\text{máx}(\tilde{y}_c) - \text{mín}(\tilde{y}_c)}{\text{máx}(\sigma_y, 1)},$$

donde $\tilde{y}_c$ es la mediana de Y en cada categoría c y σ_y es una escala global (desviación estándar) para volver la señal comparable entre variables.

Interpretación: valores bajos indican categorías con comportamiento similar (poca diferenciación); valores medios sugieren segmentación moderada; valores altos indican diferencias fuertes entre grupos (oportunidades de segmentación y decisiones por categoría).

S4. Desbalance en barras de frecuencia o agregadas (concentración/dominancia). Las barras son efectivas para comparar magnitudes por categoría, y una pregunta típica en EDA es si existe dominancia (pocas categorías concentran la masa). Para capturar esa concentración se usa una medida de “desbalance” respecto a la media:

$$m(v) = \text{uneven}(V) = \frac{\sqrt{\sum_i (v_i - \bar{v})^2}}{\sum_i |v_i|},$$

donde v_i son frecuencias (o valores agregados) por categoría y $\bar{v}$ es su media.

Interpretación: valores bajos indican distribución relativamente balanceada entre categorías; valores medios reflejan cierta concentración; valores altos suelen revelar “top categorías” dominantes (*insight* accionable para priorización, Pareto, y detección de categorías que explican gran parte del fenómeno).

S5. Tendencia en línea temporal (consistencia de evolución). En una serie temporal interesa si hay una tendencia sostenida (creciente o decreciente). Para reducir ruido, Y se agrega en *bins* temporales y luego se mide monotonía con Spearman entre el índice temporal y la serie agregada [Spearman, 1904]:

$$m(v) = |\rho_S|.$$

Interpretación: valores bajos indican que no hay tendencia monotónica (posible estacionalidad/volatilidad); valores medios sugieren tendencia parcial; valores altos indican tendencia consistente. El signo de ρ_S (guardado como evidencia) distingue alza vs. baja.

S6. Monotonía en dispersión (fuerza de relación). En un gráfico de dispersión, la señal analítica más directa es si existe relación monotónica entre X y Y . Spearman ρ_S es apropiado en EDA porque mide asociación monotónica sin exigir linealidad [Spearman, 1904]. Se usa:

$$m(v) = \rho_S(X, Y)^2.$$

El cuadrado enfatiza fuerza (no dirección), mantiene la señal en $[0, 1]$; y se conserva el signo de ρ_S como evidencia.

Interpretación: valores bajos sugieren ausencia de relación monotónica clara; valores medios indican asociación moderada; valores altos sugieren relación fuerte (útil para hipótesis, segmentación o modelado).

S7. Asociación categórica en mapa de calor (dependencia entre categorías). Un mapa de calor nominal $\times$ nominal muestra co-ocurrencias; la pregunta analítica es si hay dependencia entre variables categóricas. Cramér’s V cuantifica esa asociación en $[0, 1]$ sobre una tabla de contingencia [Cramér, 1946]:

$$m(v) = V.$$

Interpretación: valores cercanos a 0 sugieren independencia aproximada; valores medios indican asociación parcial (segmentos relevantes); valores altos señalan dependencia fuerte (combinaciones de categorías informativas).

3.5.3 Puntaje determinista: métrica principal + calidad, y ajuste por intención

El puntaje de una vista se calcula en dos etapas: primero, un *score* base determinista que combina métrica principal y calidad; luego, un ajuste multiplicativo guiado por inten-

ción/foco. La combinación lineal ponderada es un enfoque estándar en decisión multiatributo cuando se desea un compromiso interpretable entre criterios [Keeney y Raiffa, 1976].

En la primera etapa se calcula un *score* base que combina señal analítica y calidad de datos. Sea v una vista candidata. Se define:

- $s_m(v) \in [0, 1]$: *score* de señal (*interestingness*) normalizado de la métrica principal.
- $s_q(v) \in [0, 1]$: *score* de calidad, definido como la completitud mínima (proporción no nula) entre las columnas requeridas por la vista.

El *score* base se calcula como:

$$Score_{base}(v) = 0,75 \cdot s_m(v) + 0,25 \cdot s_q(v).$$

La elección 0,75/0,25 prioriza la señal analítica (*interestingness*) pero mantiene una penalización explícita por baja calidad; ambos pesos son configurables.

En la segunda etapa, si el usuario entrega intención, la aplicación web aplica un ajuste multiplicativo al puntaje base por metas analíticas y columnas foco:

$$Score_{final}(v) = Score_{base}(v) \cdot \alpha_{metric}(v) \cdot \alpha_{fields}(v),$$

siguiendo la idea de reponderar recomendaciones por objetivos y por columnas mencionadas en la consulta [Lee et al., 2022, Wongsuphasawat et al., 2016].

Para que esta reponderación sea auditable, el ajuste se separa en dos factores: uno define qué señal estadística conviene priorizar según la meta analítica detectada, y el otro define dónde buscar según las columnas mencionadas o relacionadas con la intención.

- $\alpha_{metric}(v)$ (qué señal priorizar). Se obtiene por una tabla de pesos según metas detectadas (*goals*). Si la vista v tiene una métrica principal $metric(v)$, entonces:

$$\alpha_{metric}(v) = w_{goal}(metric(v)).$$

Cada *goal* ajusta (repondera) la importancia de las señales definidas en (S1)–(S7). En términos prácticos, la aplicación web utiliza un conjunto de pesos por métrica y los aplica según el objetivo detectado.

- **distribution**: mantiene pesos neutros y no aplica una bonificación específica sobre métricas.
- **outliers**: prioriza señales de atípicos/colas en cuantitativas (S2, S1).
- **inequality**: prioriza señales de dominancia/concentración y de diferencias robustas entre grupos (S4, S3).

- **trend**: prioriza la señal de tendencia temporal (S5).
- **relationship**: prioriza señales de relación entre dos cuantitativas o dependencia entre dos categóricas (S6, S7).

Si la meta detectada es *distribution* o no se detecta un objetivo especializado, se dejan pesos neutros (factor 1,0) y el orden lo determina el score base, salvo por eventuales ajustes de foco cuando existan columnas mencionadas.

- $\alpha_{\text{fields}}(v)$ (dónde buscar). Se calcula usando las columnas foco F detectadas desde el texto (*matching* contra nombres y sinónimos del catálogo). Sea $U(v)$ el conjunto de columnas usadas por la vista:
 - Si la vista usa al menos una columna de F ($U(v) \cap F \neq \emptyset$), entonces se considera “alineada” con la pregunta y se aplica una bonificación: $\alpha_{\text{fields}}(v) = 1 + b$ (por defecto $b = 0,25$).
 - Si la vista no usa ninguna columna de F ($U(v) \cap F = \emptyset$), entonces se considera “menos relevante para la pregunta” y se aplica una penalización: $\alpha_{\text{fields}}(v) = p$ (por defecto $p = 0,65$).
 - Adicionalmente, si la intención recibida incluye columnas objetivo (*target fields*) T y la vista no las utiliza ($U(v) \cap T = \emptyset$), se aplica un factor de penalización adicional $p_t = 0,5$ sobre α_{fields} . Esto permite reforzar la focalización cuando una integración o etapa previa entrega explícitamente una variable de interés.

Este ajuste guía el *ranking* sin romper el criterio determinista. Un gráfico con puntaje base bajo (por baja evidencia o mala calidad) no pasa a ser “bueno” solo por coincidir con la intención; la intención solo reordena candidatos ya válidos. Además, los factores α quedan registrados como evidencia en la auditoría (pesos por métrica y por columnas), permitiendo explicar por qué una vista subió o bajó.

Dado que los factores multiplicativos α_{metric} y α_{fields} pueden exceder 1, el producto $Score_{\text{final}}$ podría superar 1,0, rompiendo el rango $[0, 1)$ garantizado por la función saturante. Para mantener la comparabilidad sin perder información de ordenamiento, se aplica una re-normalización por máximo: si el score máximo observado en el conjunto de candidatos supera 1,0, todos los scores se dividen por dicho máximo:

$$Score'_{\text{final}}(v) = \begin{cases} \frac{Score_{\text{final}}(v)}{\max_{v'} Score_{\text{final}}(v')} & \text{si } \max_{v'} Score_{\text{final}}(v') > 1, \\ Score_{\text{final}}(v) & \text{en caso contrario.} \end{cases}$$

De este modo, en modo libre —donde no hay ajuste por intención— los scores originales se preservan intactos; en modo dirigido el *ranking* relativo se mantiene exactamente y todos los scores quedan en $[0, 1)$; y no se pierde resolución entre candidatos cercanos al tope, a diferencia de un truncamiento a 1,0 que igualaría candidatos distintos.

3.5.4 Re-ranking por diversidad y cobertura: MMR (anti-redundancia)

El *ranking* por puntaje (relevancia) tiende a producir resultados redundantes: muchas vistas con el mismo tipo de gráfico o con las mismas columnas, variando solo detalles menores. Para cumplir OE1 (pertinencia y diversidad) sin perder relevancia, se aplica una etapa explícita de re-ranking diversificado mediante *Maximal Marginal Relevance* (MMR) [Carbonell y Goldstein, 1998]. La motivación es evitar sobrecarga cognitiva y ruido visual al presentar demasiadas vistas similares [Munzner, 2014, Ware, 2013], además de cubrir distintos “ángulos” analíticos (distribución, relación, tendencia) con un presupuesto pequeño de sugerencias.

MMR se aplica una vez que cada vista candidata ya cuenta con un puntaje final $Score_{final}$ (métrica principal + calidad, y ajuste por intención/foco si existe). Luego ejecuta selección diversificada hasta completar un presupuesto fijo K ; los candidatos válidos restantes se conservan como alternativas y se etiquetan como “no seleccionado en top- K ”, distinguiéndolos de descartes por redundancia.

La selección se realiza con una regla greedy que balancea relevancia y diversidad. Sea S el conjunto de vistas ya seleccionadas. La regla MMR se define como:

$$MMR(c) = \lambda \cdot Score_{final}(c) - (1 - \lambda) \cdot \max_{s \in S} \text{sim}(c, s),$$

donde $\lambda \in [0, 1]$ controla el balance entre relevancia y no-redundancia. En la implementación se usa $\lambda = 0,75$, lo que prioriza relevancia (3:1) pero incorpora diversidad de manera consistente.

La similitud se calcula como una suma ponderada de tres señales simples, explícitas y audita-
bles:

$$\begin{aligned} \text{sim}(a, b) = & w_c \cdot \mathbb{I}[\text{chart}(a) = \text{chart}(b)] \\ & + w_f \cdot J(\text{fields}(a), \text{fields}(b)) \\ & + w_a \cdot \mathbb{I}[\text{action}(a) = \text{action}(b)]. \end{aligned}$$

con pesos por defecto $(w_c, w_f, w_a) = (0,4, 0,5, 0,15)$. Aquí:

- $\mathbb{I}[\cdot]$ es la función indicadora (1 si se cumple, 0 si no).
- $\text{fields}(\cdot)$ es el conjunto de columnas usadas por la vista, principalmente los campos x e y de cada candidato.
- $J(A, B) = \frac{|A \cap B|}{|A \cup B|}$ es la similitud de Jaccard [Jaccard, 1901].
- $\text{action}(\cdot)$ corresponde al objetivo analítico etiquetado en el candidato (p. ej. *distribution*, *relationship*, *trend*), y se usa como una señal adicional de redundancia (peso menor) para favorecer cobertura entre metas.

El mayor peso se asigna a solapamiento de campos ($w_f = 0,5$) porque reutilizar las mismas columnas suele producir *insights* muy similares (misma “historia” estadística). El tipo de gráfico ($w_c = 0,4$) captura redundancia perceptual: repetir el mismo *encoding* muchas veces reduce variedad. La acción ($w_a = 0,15$) es un correctivo suave: dos vistas pueden compartir meta (p. ej. “distribución”) y aun así ser útiles si cambian los campos, por lo que su peso se mantiene menor.

El algoritmo usado es greedy (como en la formulación original de MMR [Carbonell y Goldstein, 1998]), por eficiencia y trazabilidad:

1. Se ordenan los candidatos por $Score_{final}$ y se agrega a S el mejor candidato (máxima relevancia).
2. Mientras $|S| < K$ (con $K = \text{max_suggestions} = 6$ por defecto) y existan candidatos:
 - a) Para cada candidato restante c , se calcula su similitud máxima con S : $\max_{s \in S} \text{sim}(c, s)$.
 - b) Se calcula $\text{MMR}(c)$ y se elige el candidato con mayor valor.
 - c) Se añade ese candidato a S y se actualiza el conteo por tipo de gráfico para respetar el máximo por tipo.

Este procedimiento produce un conjunto final pequeño y diverso sin necesidad de resolver una optimización combinatoria exacta (lo cual sería más costoso y menos transparente). En términos prácticos, MMR evita que el top- K quede dominado por variaciones del mismo patrón, manteniendo resultados explorables y comparables [Munzner, 2014, Ware, 2013].

La etapa MMR determina cuáles vistas entran al conjunto final (cobertura y no redundancia). Una vez seleccionado ese conjunto, la lista que se muestra al usuario se ordena por $Score_{final}$ descendente. Es importante considerar que esta post-ordenación no cambia la pertenencia del top- K .

Además del término de diversidad, se aplica una restricción explícita: a lo sumo $\text{max_same_chart_type} = 2$ vistas por tipo de gráfico (por defecto, es configurable por el usuario). Esto actúa como un control de redundancia fuerte: incluso si varios histogramas (o barras) tienen puntajes altos, la aplicación web fuerza variedad de *encodings* para mejorar cobertura (OE1) y reducir saturación visual [Munzner, 2014, Ware, 2013]. Si en algún punto ningún candidato restante puede seleccionarse por violar esta restricción, los restantes se marcan como descartados por “redundancia por tipo”.

Tras seleccionar S , los candidatos no seleccionados se etiquetan para auditoría (OE3). En particular:

- **Redundancia por similitud:** si $\max_{s \in S} \text{sim}(c, s) \geq 0,6$, la vista se marca como “alta

similitud” y se reporta el valor de similitud (umbral moderado-alto con los pesos anteriores).

- **No seleccionado en top- K :** candidato válido que no entra en el conjunto final tras la selección diversificada.
- **Redundancia por tipo:** si el límite `max_same_chart_type` impide escoger candidatos en un paso, los restantes se descartan por exceso de repetición del tipo.

Los hiperparámetros por defecto están diseñados para un compromiso práctico entre exploración y claridad:

- $K = 6$ limita la lista final a un tamaño manejable para inspección humana sin saturar la interfaz, consistente con recomendaciones generales sobre carga perceptual en visualización [Munzner, 2014, Ware, 2013].
- $\lambda = 0,75$ preserva el *ranking* por relevancia como señal principal, incorporando una penalización de redundancia lo suficientemente fuerte para cambiar el orden solo cuando dos vistas son muy parecidas (principio de “diversificar sin sacrificar utilidad”) [Carbonell y Goldstein, 1998].
- `max_same_chart_type = 2` fuerza cobertura entre *encodings* (histograma, barras, línea, etc.), evitando que una sola familia domine el set final cuando el *dataset* favorece esa familia.
- El umbral $\text{sim} \geq 0,6$ actúa como criterio operativo de redundancia: típicamente requiere coincidencia del tipo y/o un solapamiento de campos suficientemente alto como para que las vistas compartan gran parte de la misma evidencia [Jaccard, 1901].

3.6 Paso 3: Narrativa explicativa anclada en evidencia

Una vez seleccionado el conjunto diverso de vistas, la aplicación web genera explicaciones textuales para cada una. El diseño de este paso prioriza la fidelidad a los hechos calculados y evita que el componente generativo introduzca conclusiones no verificables.

3.6.1 Rol acotado del componente generativo (*grounding* y no-interferencia)

La narrativa se diseña bajo el principio de anclaje a hechos (*grounding*): el componente generativo no calcula estadísticas, no decide el *ranking*, no altera transformaciones, ni introduce conclusiones causales. Su rol se limita a redactar explicaciones legibles y accionables a partir de evidencia ya computada por el motor determinista, reduciendo el riesgo de “inventar” números o patrones inexistentes y manteniendo trazabilidad [Phillips *et al.*, 2021].

3.6.2 Narrativa descriptiva vs. prescriptiva

La aplicación web entrega dos capas complementarias:

- **Narrativa descriptiva (N2):** explica qué patrón se observa en la vista seleccionada y lo respalda con 1-2 cifras provenientes de la evidencia agregada.
- **Prescripción analítica (N3):** propone exclusivamente un siguiente paso de EDA (p. ej. segmentar, filtrar, comparar, abrir una vista alternativa o revisar calidad), evitando recomendaciones de negocio (p. ej. “bajar precios”) que requieren contexto externo que la aplicación web no posee.

Con esto, el cálculo queda en el motor determinista y el lenguaje natural queda limitado a comunicar evidencia ya verificada, manteniendo auditable el *ranking*.

3.6.3 Paquete factual y contrato JSON (entrada y salida)

Para asegurar *grounding*, el LLM recibe únicamente un paquete factual estructurado, derivado de artefactos ya calculados por la aplicación web. Operativamente, el prompt incorpora:

- **Resumen base del *dataset* (N1):** texto compacto construido desde el perfil y el catálogo (tamaño, nulos aproximados, tipos semánticos y alertas).
- **Hechos por vista (JSON):** una lista de ítems donde cada ítem contiene *id* y *facts*. *facts* incluye metadatos de la vista, señal principal y su contexto (alto/medio/bajo), además de evidencia agregada/compactada con transformaciones y cifras ya computadas.

El siguiente fragmento muestra la estructura del JSON de entrada por vista:

```
[
  {
    "id": "0",                                // id de la vista (índice en el top-K)
    "facts": {
      "intent_goals": [                      // metas analíticas detectadas (si existen)
        "distribution",
        "inequality"
      ],
      "view": {
        "chart_type": "bar",                // tipo de gráfico recomendado
        "aggregation": "sum",               // agregación aplicada (si corresponde)
        "aggregation_label_es": "total",
      }
    }
  }
]
```

```

"x": {
  "name": "product_subcategory",      // nombre real de la columna
  "label": "Subcategoría de producto", // etiqueta humana para narrativa
  "description": "...",               // descripción (heurística/LLM)
  "unit": "",
  "semantic_type": "nominal",         // tipo semántico inferido
  "role": "dimension",                // measure/dimension
  "synonyms": ["..."]                // sinónimos para coincidencia
  ↪ semántica
},

"y": {
  "name": "unit_price_clp",
  "label": "Precio unitario (CLP)",
  "description": "...",
  "unit": "CLP",
  "semantic_type": "quantitative",
  "role": "measure",
  "synonyms": ["..."]
}
},

"signal": {
  "metric_code": "unevenness",        // métrica principal (código interno)
  "metric_label_es": "Dominancia/desbalance entre categorías",
  "metric_value": 0.42,               // valor crudo (evidencia)
  "level": "alto",                   // interpretación cualitativa
  ↪ (alto/medio/bajo)
  "meaning_es": "Dominancia/desbalance entre categorías."
},

"evidence_data": {
  // evidencia compactada y transformaciones aplicadas
  "top_categories": {"...": 5050.0, "...": 38935.0},
  "n_valid": 12345                    // número de observaciones válidas
  ↪ usadas
}
}
}
]

```

Como salida, el LLM debe devolver solo JSON válido indexado por id. Cada entrada incluye exactamente: `summary` (2-3 frases), `evidence` (4-6 frases con 1-2 cifras) y `next_action` (1-2 frases analíticas).

```

{
  "0": {
    "summary": "...",      // resumen breve
    "evidence": "...",     // explicación con 1-2 cifras de evidencia
    "next_action": "..." // siguiente paso analítico (EDA)
  }
}

```

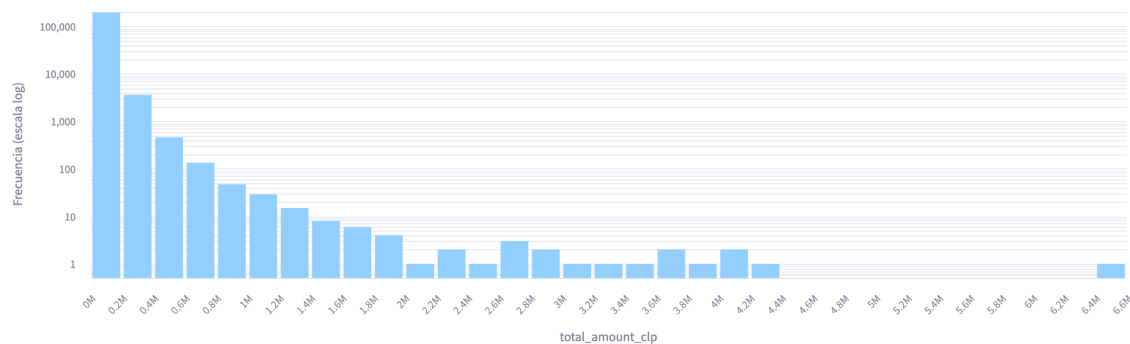
}

Este formato estructurado mejora la estabilidad de integración (parseo automático), refuerza la disciplina de “explicar con evidencia” y permite auditar que el texto se basa en hechos calculados [Phillips *et al.*, 2021]. Los *prompts* concretos utilizados para el enriquecimiento del catálogo y para la generación de narrativas se incluyen en los Anexos, de modo que el contrato entre el motor determinista y el componente generativo pueda ser inspeccionado.

La narrativa generada no se considera un resultado aislado, sino una capa comunicacional sobre evidencia previamente calculada. Por ello, cada explicación debe poder vincularse con la vista, la métrica principal, las transformaciones aplicadas y los criterios de selección utilizados por el motor. Esta vinculación motiva el Paso 4, donde la aplicación expone dichos elementos como parte del panel de transparencia.

La Figura 7 ilustra el resultado visible de esta etapa. En una misma vista se presenta la recomendación seleccionada, la métrica que la activó, el gráfico correspondiente, la narrativa descriptiva y una acción siguiente de EDA. Con ello, el usuario puede pasar de una recomendación técnica a una interpretación accionable sin perder el vínculo con la evidencia calculada.

Vista seleccionada



Narrativa (N2)

Se analizaron 200,000 transacciones. El monto total promedio es de 45,397.90 CLP. La asimetría es de 18.60, lo cual indica una cola muy larga hacia los valores altos. El percentil 50 (mediana) es de 28,261.5 CLP, mientras que el percentil 95 alcanza los 136,554.5 CLP. La mayoría de las transacciones (196,411) se encuentran en el primer rango de monto, lo que confirma la concentración en valores bajos.

Siguiente acción (N3)

Investigar las transacciones de alto valor para entender qué las impulsa y si representan un comportamiento anómalo o una oportunidad de negocio.

Figura 7: Recomendación visual con narrativa y siguiente acción.

3.7 Paso 4: Auditoría, descartes y transparencia

La aplicación web no solo entrega recomendaciones: documenta explícitamente el razonamiento detrás de cada una. Esto permite que el usuario inspeccione, cuestione o valide las decisiones del motor, cumpliendo con el requisito de trazabilidad.

3.7.1 Qué evidencia se expone (y cómo se vincula al *ranking*)

La auditoría tiene como objetivo que cada recomendación sea inspeccionable y defendible. Para cada vista seleccionada se expone, como mínimo:

1. **Qué se graficó (especificación de la vista).** Se informa el tipo de gráfico y los campos utilizados (ejes y, cuando corresponde, la agregación aplicada). Además, se muestran los tipos semánticos inferidos para las columnas involucradas, como parte del contexto interpretativo.
2. **Qué señal la activó (métrica principal).** Se reporta la métrica principal asociada al tipo de gráfico y su valor crudo (como evidencia). Adicionalmente, se entrega una interpretación cualitativa simple (por ejemplo *bajo/medio/alto*) que ayuda a traducir la señal numérica a lenguaje natural en la narrativa N2. Este contexto no altera el *ranking*; solo guía la explicación.
3. **Qué transformaciones se aplicaron (trazabilidad del dato).** Se documentan las transformaciones que condicionan la vista: recorte a top- k categorías, *binning* de histogramas, agregación/reescalamiento temporal y estadísticos/percentiles utilizados.
4. **Cómo se llegó al puntaje final (descomposición del *ranking*).** Se expone cómo se compone el puntaje: contribución de la métrica principal ya normalizada a una escala comparable, factor de calidad del dato (basado en completitud de las columnas requeridas) y puntaje base resultante. Cuando se aplica una normalización robusta, se mantiene trazabilidad indicando la versión “antes de normalizar” y el método/parámetros utilizados. Si existe intención, se reportan los factores de reponderación aplicados (por objetivo/métrica y por alineación con columnas foco), de modo que el puntaje final pueda rastrearse a señales concretas.

Además, la transparencia no se limita a valores numéricos: el panel incluye una explicación breve en lenguaje natural que traduce la evidencia (por ejemplo, “asimetría alta sugiere cola larga”, “dominancia de pocas categorías” o “tendencia marcada en el tiempo”) y explicita si hubo bonificación o penalización por intención/foco. Esto facilita la interpretación para usuarios no técnicos sin perder trazabilidad.

Este diseño sigue el principio de explicabilidad: no basta con mostrar una recomendación, sino que debe quedar claro por qué fue recomendada y qué evidencia la sustenta [Phillips *et al.*, 2021].

La Figura 8 muestra la materialización de este panel en la interfaz. Además de la explicación legible, la aplicación expone la señal principal, la regla perceptual, la calidad del dato, los puntajes y el detalle estructurado de la vista.

Siguiente acción (N3)

Investigar las transacciones de alto valor para entender qué las impulsa y si representan un comportamiento anómalo o una oportunidad de negocio.

Transparencia

- Señal principal: Asimetría alta sugiere cola larga / outliers (skew=18.605).
- Regla perceptual: Histograma para evaluar distribución cuantitativa y detectar colas.
- Calidad del dato para esta vista: 100%.
- Puntaje base (señal + calidad): 0.944.
- Puntaje final: 0.944.

```
{
  "chart_type": "histogram"
  "fields": {
    "x": "total_amount_clp"
    "y": NULL
    "agg": NULL
  }
  "fields_full": NULL
  "semantic_types": {
    "x": "quantitative"
    "y": NULL
  }
  "score": 0.9440432430757695
  "metric_name": "skewness"
  "metric_value": 18.60481060443391
  "payload": {
    "n": 200000
    "mean": 45397.904185
    "std": 68241.29121257867
    "skewness": 18.60481060443391
  }
}
```

Figura 8: Panel de transparencia.

3.7.2 Descartes y alternativas (por qué otras vistas no quedaron)

Para fortalecer OE3, la aplicación web no solo muestra lo seleccionado, sino que conserva un rastro de alternativas descartadas con su razón:

- **Descartes por calidad:** vistas que requieren columnas con demasiados nulos; operativamente, se descartan si alguna columna requerida tiene completitud menor a 0,6.

- **Descartes por redundancia:** vistas muy similares a otras ya seleccionadas (por similitud de campos/tipo de gráfico) o descartadas por límite de repetición por tipo.
- **No seleccionado por presupuesto:** candidato válido, pero que no entra en el conjunto final por el balance relevancia/diversidad bajo un presupuesto fijo de recomendaciones.

El listado completo de candidatos descartados se presenta en el panel de trazabilidad (Detalles técnicos), junto con su razón de descarte. Esto evita duplicar alternativas en el flujo principal y mantiene una auditoría consistente (OE3).

La Figura 9 muestra este rastro de decisión. En lugar de ocultar las alternativas que no fueron mostradas, la aplicación conserva candidatos descartados e indica la razón operativa, por ejemplo columnas con demasiados nulos o vistas que no entraron en el presupuesto final de recomendaciones.

▼ Descartes para transparencia

Aquí queda el rastro de por qué se descartaron candidatos.

	chart_type	x	y	agg	score	discard_reason
0	bar	delivery_days	None	count	0	Columna 'delivery_days' con demasiados nulos.
1	bar	delivery_days	total_amount_clp	sum	0	Columna 'delivery_days' con demasiados nulos.
2	boxplot	delivery_days	total_amount_clp	None	0	Columna 'delivery_days' con demasiados nulos.
3	bar	delivery_days	unit_price_clp	sum	0	Columna 'delivery_days' con demasiados nulos.
4	boxplot	delivery_days	unit_price_clp	None	0	Columna 'delivery_days' con demasiados nulos.
5	bar	return_reason	None	count	0	Columna 'return_reason' con demasiados nulos.
6	bar	return_reason	total_amount_clp	sum	0	Columna 'return_reason' con demasiados nulos.
7	boxplot	return_reason	total_amount_clp	None	0	Columna 'return_reason' con demasiados nulos.
8	bar	return_reason	unit_price_clp	sum	0	Columna 'return_reason' con demasiados nulos.
9	boxplot	return_reason	unit_price_clp	None	0	Columna 'return_reason' con demasiados nulos.

Figura 9: Registro de descartes de visualizaciones.

3.7.3 Justificación perceptual (reglas de lectura del gráfico)

La auditoría también incluye una justificación perceptual textual por tipo de gráfico (p. ej. por qué una línea es adecuada para continuidad temporal o por qué barras son adecuadas para comparación categórica). Estas justificaciones se basan en principios clásicos de efectividad perceptual y elección de *encodings*, reforzando que la aplicación web no propone gráficos arbitrarios, sino que sigue reglas defendibles [Cleveland y McGill, 1984, Mackinlay, 1986, Munzner, 2014, Ware, 2013].

3.7.4 Exportación reproducible de evidencia

Finalmente, la aplicación web permite exportar un JSON con los principales artefactos del análisis: resumen del *dataset*, catálogo semántico, intención detectada, sugerencias (incluyendo evidencia y transformaciones asociadas), narrativas generadas por el componente LLM, descartes (con razones) y metadatos de ejecución (por ejemplo, tiempo total y contadores principales del proceso). Este archivo deja trazabilidad suficiente para revisar una sesión, depurar el comportamiento del motor y contrastar los resultados obtenidos con la evidencia que los originó [Phillips *et al.*, 2021].

3.8 Capa de presentación e interactividad (*Streamlit*)

La capa de presentación implementa el flujo anterior en una interfaz Streamlit. Desde ella, el usuario carga un archivo CSV/Excel, opcionalmente ingresa una intención de análisis, ajusta parámetros acotados del motor (como `max_suggestions` y `max_same_chart_type`) y configura el proveedor LLM que utilizará el flujo generativo, ya sea en nube o local mediante Ollama.

La sección de resultados reúne los artefactos descritos en los pasos anteriores: resumen del *dataset*, catálogo semántico, visualizaciones sugeridas, narrativa N2/N3, panel de transparencia, alternativas descartadas y exportación reproducible. Así, la interfaz funciona como punto de entrada y también como superficie de inspección del proceso.

La aplicación permite exportar los principales artefactos del análisis como un archivo JSON para auditoría, replicación y anexos. A continuación se muestra un ejemplo ilustrativo (con comentarios) del formato exportado:

```
{
  "dataset_summary": "Dataset con 123456 filas y 42 columnas. Nulos aprox. 18%.
  ↳ Tipos: 10 cuantitativas, 25 categóricas, 2 temporales, 5 geográficas. Alertas:
  ↳ Columnas con muchos nulos: delivery_days (aprox. 62% nulos), promo_code
  ↳ (aprox. 55% nulos).",
  "semantic_catalog": [
    {
      "column": "unit_price_clp",
      "semantic_type": "quantitative",
      "role": "measure",
      "semantic_role": "currency",
      "unit": "CLP",
      "unit_confidence": 0.7,
      "description": "Precio unitario en pesos chilenos.",
      "synonyms": ["precio unitario", "precio", "unitario"],
      "geo_hierarchy": null
    }
  ]
}
```

```
],  
  
"intent": {  
  "mode": "directed",  
  "goals": ["distribution"],  
  "focus_fields": ["unit_price_clp", "product_subcategory"],  
  "target_fields": [],  
  "matched_fields": {  
    "unit_price_clp": 0.83,  
    "product_subcategory": 0.65  
  }  
},  
  
"suggestions": [  
  {  
    "chart_type": "bar",  
    "fields": { "x": "product_subcategory", "y": "unit_price_clp", "agg": "sum" },  
    "score": 0.731,  
    "metric_name": "unevenness",  
    "metric_value": 0.42,  
    "payload": {  
      "quality_score": 0.94,  
      "metric_score": 0.88,  
      "base_score": 0.84,  
      "metric_weight": 1.0,  
      "field_weight": 1.25,  
      "intent_goals": ["distribution"],  
      "top_categories": ["..."],  
      "top_values": [5050.0, 38935.0, "..."],  
      "unevenness": 0.42,  
      "top_gap": 33885.0,  
      "transform": { "top_k": 25, "agg": "sum" }  
    }  
  }  
],  
  
"discarded": [  
  {  
    "chart_type": "scatter",  
    "x": "a",  
    "y": "b",  
    "agg": null,  
    "score": 0.031,  
    "discard_reason": "No seleccionado en top-K (ranking/diversidad)."  
  }  
],  
  
"narrative_selected": {  
  "index": 0,  
  "summary": "...",  
  "evidence": "...",  
  "next_action": "..."  
},
```

```
"timings": {  
  "elapsed_time": 1.23,  
  "n_candidates": 320,  
  "n_selected": 6,  
  "n_discarded": 314,  
  "query": "..."  
}  
}
```

Este artefacto permite reconstruir qué se recomendó y con qué evidencia (métricas, transformaciones y razones de descarte), junto con metadatos de ejecución, manteniendo una trazabilidad completa.

3.9 Síntesis: diferenciación frente a soluciones existentes

La propuesta se diferencia por integrar, en un mismo flujo, componentes que suelen aparecer separados en herramientas de EDA automático:

- **Descubrimiento visual determinista y reproducible:** la generación de candidatos, el cálculo de métricas, la normalización, el *ranking* y la selección diversa siguen reglas explícitas y parámetros configurables, lo que permite repetir resultados y explicar decisiones.
- **Neuro-simbiosis acotada:** el LLM no calcula estadísticas ni define las métricas del motor, ni reemplaza el *score* determinista; se utiliza para enriquecer metadatos (descripción/sinónimos/unidad) y para redactar narrativas ancladas a hechos. En la práctica, estos metadatos pueden influir indirectamente en el *matching* de intención (y por tanto en la reponderación por foco), pero el cómputo estadístico y el *score* base permanecen verificables y reproducibles [Phillips *et al.*, 2021].
- **Semántica auto-construida sin modelado BI previo:** la aplicación web infiere tipos semánticos y pistas de rol (p. ej. identificador/porcentaje/moneda/estado) y construye un catálogo por columna que facilita interpretación y mapeo de intención, sin depender de un modelo manual de negocio para arrancar.
- ***ranking* adaptativo por intención como reponderación verificable:** la intención del usuario no “reemplaza” el motor, sino que repondera métricas y columnas foco de manera explícita (factores registrados en el *payload*), manteniendo el *pipeline* rastreable y sin inventar estadísticas.
- **Auditoría explícita (OE3) con alternativas descartadas:** además de mostrar las vistas seleccionadas, se preservan y explicitan razones de descarte (calidad, redundancia por MMR, límite por tipo), fortaleciendo confianza y defensa técnica.

En conjunto, la solución documenta no solo el resultado visual, sino también el proceso que llevó a cada recomendación, permitiendo revisar, reproducir y adaptar el enfoque a otros *datasets* tabulares.

CAPÍTULO 4

VALIDACIÓN DE LA SOLUCIÓN

En este capítulo se presenta la validación de la aplicación web desarrollada, con el propósito de analizar en qué medida responde a los objetivos específicos formulados anteriormente. La estructura sigue tres ejes: primero se describe el entorno experimental y los conjuntos de datos diseñados para las pruebas, luego se reportan los resultados de la validación técnica —pertinencia y diversidad de las sugerencias, tiempo de respuesta, trazabilidad de las decisiones y fidelidad factual de las narrativas— y finalmente se discuten los resultados obtenidos.

4.1 Entorno experimental

La evaluación se realizó sobre tres conjuntos de *datasets* sintéticos, cada uno pensado para ejercitar capacidades distintas del motor de recomendación. Se optó por datos sintéticos porque esto permite conocer de antemano la estructura y los patrones presentes, lo que facilita contrastar las sugerencias de la aplicación web contra lo esperado. Además, al controlar la generación se pudieron incluir intencionalmente casos como *outliers*, distribuciones asimétricas y relaciones bivariadas conocidas. Los tamaños van desde 25 000 hasta 540 000 filas, de modo que también se pueda observar el comportamiento de la aplicación web ante volúmenes distintos. La Tabla 4 resume las características generales de cada *dataset*.

Dataset	Filas	Columnas	Características principales
HR (People Analytics)	25 000	17	Sueldos, niveles jerárquicos, fechas de contratación, attrition, contratos. Incluye columnas cuantitativas, nominales y temporales.
Retail Chile	200 000	28	Transacciones de venta con montos, categorías de producto, geografía (comuna/región), fechas, IVA. Alta variedad de tipos semánticos.
Energy (Consumo)	540 000	10	Serie temporal de consumo eléctrico por hogar, con temperatura, plan tarifario, tipo de medidor. Predominan relaciones categóricas y temporales.

Tabla 4: Conjuntos de datos utilizados en la validación.

Los archivos utilizados en esta validación fueron publicados como un conjunto de datos en Zenodo, incluyendo los tres CSV sintéticos empleados en las pruebas: HR, Retail Chile y Energy [León, 2026].

A continuación se describen las columnas de cada *dataset* con mayor detalle, ya que la composición de variables es relevante para interpretar las sugerencias que la aplicación web genera.

4.1.1 Dataset HR (People Analytics)

Este *dataset* simula un registro de empleados de una organización, con 25 000 filas y 17 columnas. La Tabla 5 detalla cada columna.

Columna	Tipo	Descripción
employee_id	string	Identificador único del empleado.
manager_id	string	Identificador del jefe directo.
department	string	Departamento (RRHH, Operaciones, Finanzas, etc.).
job_level	string	Nivel jerárquico (Junior, SemiSenior, Senior, Lider, Gerencia).
contract_type	string	Tipo de contrato (Indefinido, PlazoFijo, Honorarios).
hire_date	date	Fecha de contratación.
tenure_years	float	Antigüedad en años.
region	string	Región geográfica del empleado.
comuna	string	Comuna del empleado.
remote_ratio	float	Proporción de trabajo remoto (0 a 1).
monthly_salary_clp	float	Sueldo mensual en pesos chilenos.
bonus_pct	float	Porcentaje de bono.
performance_score	float	Evaluación de desempeño (escala 1-5).
training_hours_ytd	float	Horas de capacitación acumuladas en el año.
engagement_score	float	Puntaje de compromiso (escala 1-5).
attrition	boolean	Indicador de desvinculación.
exit_reason	string	Motivo de salida (cuando aplica).

Tabla 5: Columnas del *dataset* HR (People Analytics, 25 000 filas).

El *dataset* mezcla columnas cuantitativas (salario, bono, desempeño, horas de capacitación), nominales (departamento, nivel, tipo de contrato, región), una temporal (fecha de contratación) y dos identificadores (employee_id, manager_id). Esta combinación permite a la aplicación web generar candidatos de los seis tipos de gráfico implementados.

4.1.2 Dataset Retail Chile

Simula 200 000 transacciones de venta en comercio minorista, con 28 columnas que abarcan información del cliente, la tienda, el producto y la transacción. La Tabla 6 resume la estructura.

Columna	Tipo	Descripción
transaction_id	string	Identificador único de transacción.
customer_id	string	Identificador del cliente.
customer_segment	string	Segmento (Estudiante, Profesional, etc.).
gender	string	Género del cliente.
age	int	Edad del cliente.
region	string	Región geográfica.
comuna	string	Comuna.
latitude	float	Latitud de la tienda.
longitude	float	Longitud de la tienda.
store_id	string	Identificador de tienda.
store_type	string	Tipo de tienda (Express, Hiper, etc.).
channel	string	Canal de venta (Tienda, Online).
payment_method	string	Medio de pago.
product_id	string	Identificador de producto.
product_category	string	Categoría de producto.
product_subcategory	string	Subcategoría.
brand	string	Marca.
unit_price_clp	float	Precio unitario en pesos chilenos.
quantity	int	Cantidad de unidades.
discount_pct	float	Porcentaje de descuento.
tax_pct	float	Porcentaje de IVA.
total_amount_clp	float	Monto total de la transacción.
purchase_datetime	datetime	Fecha y hora de compra.
delivery_days	int	Días de despacho (cuando aplica).
returned	boolean	Indicador de devolución.
return_reason	string	Motivo de devolución.
satisfaction_score	float	Puntaje de satisfacción (escala 1-5).
promo_code	string	Código promocional aplicado.

Tabla 6: Columnas del *dataset* Retail Chile (200 000 filas).

Con 28 columnas y alta variedad de tipos semánticos —identificadores, montos, porcentajes, categorías jerárquicas, coordenadas geográficas y una marca temporal— este *dataset* genera un espacio de candidatos considerablemente mayor que los otros dos, lo que lo convierte en el caso más exigente en términos de combinaciones a evaluar.

4.1.3 Dataset Energy (Consumo Eléctrico)

Contiene 540 000 registros diarios de consumo eléctrico por hogar, con solo 10 columnas. La Tabla 7 presenta la estructura.

Columna	Tipo	Descripción
household_id	string	Identificador del hogar.
date	date	Fecha de la lectura.
region	string	Región geográfica.
comuna	string	Comuna.
meter_type	string	Tipo de medidor (Residencial, Comercial).
tariff_plan	string	Plan tarifario (TarifaA, TarifaB, etc.).
temperature_c	float	Temperatura ambiente en grados Celsius.
kwh	float	Consumo eléctrico en kilowatt-hora.
peak_flag	boolean	Indicador de hora punta.
bill_clp	float	Monto de la cuenta en pesos chilenos.

Tabla 7: Columnas del *dataset* Energy (540 000 filas).

A pesar de tener el mayor volumen de filas, este *dataset* es el más estrecho (10 columnas). Las variables cuantitativas son pocas (temperatura, consumo, monto) y las nominales tienen baja cardinalidad, de modo que el espacio de candidatos es reducido. Su valor para la validación está en probar el comportamiento de la aplicación web con series temporales largas y en verificar que el motor sigue siendo rápido a pesar del alto número de filas.

La evaluación principal utilizó Gemini 2.5 Flash Lite, un modelo que balancea calidad con tiempo de respuesta. Además, para contrastar una alternativa local, se ejecutó el mismo *benchmark* con Ollama y qwen2.5:0.5b.

4.2 Validación técnica

La validación técnica se organiza en torno a los criterios definidos en los objetivos específicos: pertinencia y diversidad de las sugerencias visuales, tiempo de respuesta, trazabilidad de las decisiones y fidelidad factual de las narrativas generadas.

4.2.1 Pertinencia de las sugerencias

Para evaluar la pertinencia (OE1), se ejecutó la aplicación web sobre cada *dataset* y se revisaron las sugerencias resultantes desde tres ángulos: que los tipos de gráfico asignados sean coherentes con los tipos semánticos de las columnas involucradas, que las métricas calculadas destaquen patrones estadísticamente relevantes, y que la aplicación web responda de

forma diferenciada cuando se especifica una intención analítica frente a cuando opera en modo libre.

La Tabla 8 muestra las seis sugerencias generadas para el *dataset* HR en modo libre, es decir, sin que el usuario declare ninguna intención particular.

#	Tipo	Columnas	Métrica	score
1	boxplot bivariado	job_level, monthly_salary_clp	median_spread = 2,43	0,781
2	bar frecuencia	attrition	unevenness = 0,56	0,738
3	bar agregada	contract_type, monthly_salary_clp	unevenness = 0,55	0,736
4	histograma	monthly_salary_clp	skewness = 1,87	0,666
5	boxplot univariado	monthly_salary_clp	outlier_ratio = 0,05	0,607
6	línea	hire_date, monthly_salary_clp	trend = 0,13	0,473

Tabla 8: Sugerencias para *dataset* HR (modo libre, $K = 6$).

Al revisar estas sugerencias se observa coherencia con lo que un analista esperaría del dominio. La primera vista —un boxplot bivariado de salario agrupado por nivel jerárquico— pone en evidencia que la dispersión de medianas entre niveles es alta (2,43 veces la desviación estándar global), lo cual tiene sentido en una estructura de compensación estratificada. Las barras de attrition y tipo de contrato revelan desbalances en la distribución de categorías, mientras que el histograma de salarios detecta asimetría positiva (skewness de 1,87), consistente con la cola derecha que producen los sueldos de gerencia. El boxplot univariado de salario en posición 5 complementa al bivariado desde otra óptica: en lugar de comparar medianas entre grupos, detecta que un 5 % de las observaciones cae fuera de las vallas de Tukey, respaldando la presencia de valores extremos. La línea temporal, en posición 6, muestra una tendencia leve en salarios según fecha de contratación. En conjunto, las seis vistas abordan aspectos analíticos distintos sin repetirse.

Para verificar el modo dirigido, se ejecutó la aplicación web con la intención “analizar *outliers* en salario”. La Tabla 9 muestra el *ranking* resultante.

En esta consulta, el motor identificó el objetivo ($goals=['outliers']$) y reponderó las vistas: el histograma de salarios pasó de la posición 4 a la primera posición con un *score* ajustado de 0,999 —superior al boxplot bivariado que lideraba en modo libre con *score* 0,781— porque las métricas de asimetría y proporción de *outliers* recibieron mayor peso ($\alpha_{metric} = 1,5$). Además, el boxplot univariado de salario con métrica *outlier_ratio* = 0,05 subió de la posición 5 a la posición 2 con *score* ajustado de 0,910, muy por encima de su *score* en modo libre (0,607). Las vistas cuyas métricas no están asociadas al objetivo *outliers* mantuvieron su *score* base sin cambios. Este comportamiento muestra que, para el caso evaluado, la reponderación por intención prioriza las vistas más alineadas con lo que el usuario declara querer

#	Tipo	Columnas	Métrica	score
1	histograma	monthly_salary_clp	skewness = 1,87	0,999
2	boxplot univariado	monthly_salary_clp	outlier_ratio = 0,05	0,910
3	boxplot bivariado	job_level, monthly_salary_clp	median_spread = 2,43	0,781
4	bar frecuencia	attrition	unevenness = 0,56	0,738
5	bar agregada	contract_type, monthly_salary_clp	unevenness = 0,55	0,736
6	línea	hire_date, monthly_salary_clp	trend = 0,13	0,473

Tabla 9: Sugerencias para *dataset* HR (modo dirigido, $K = 6$).

observar, sin alterar las que no corresponden al objetivo.

La Tabla 10 presenta las sugerencias para el *dataset* Retail, que con sus 28 columnas ofrece mayor variedad de combinaciones posibles.

#	Tipo	Columnas	Métrica	Score
1	histograma	total_amount_clp	skewness = 18,60	0,944
2	dispersión	total_amount_clp, unit_price_clp	monotonicity = 0,70	0,774
3	bar agregada	tax_pct, unit_price_clp	unevenness = 0,65	0,764
4	bar frecuencia	returned	unevenness = 0,65	0,763
5	mapa de calor	channel, store_type	association = 0,59	0,747
6	boxplot univariado	unit_price_clp	outlier_ratio = 0,09	0,741

Tabla 10: Sugerencias para *dataset* Retail Chile (modo libre, $K = 6$).

El valor de skewness de 18,60 en `total_amount_clp` refleja la cola derecha pronunciada que es habitual en datos transaccionales, donde unas pocas compras concentran montos muy superiores al promedio. La dispersión en posición 2 detecta una relación monótonica fuerte ($\rho^2 = 0,70$) entre monto total y precio unitario, lo que tiene sentido dado que el monto depende directamente del precio. La barra de impuestos por precio unitario (posición 3) y la barra de devoluciones (posición 4) identifican desbalances categoriales relevantes. El mapa de calor en posición 5 revela una asociación fuerte ($V = 0,59$) entre canal de venta y tipo de tienda, mientras que el boxplot univariado de precio en posición 6 detecta un 9 % de *outliers*, señalando valores atípicos en la distribución de precios. Cabe notar que MMR sigue maximizando diversidad en el conjunto seleccionado (seis tipos de gráfico distintos), y la presentación final se ordena por *score* para que la tabla sea consistente con el *ranking* mostrado.

4.2.2 Tiempo de respuesta

Para evaluar el cumplimiento del OE2, se ejecutó un *benchmark* que mide las fases principales de la aplicación web por separado: enriquecimiento semántico (que incluye la llamada al LLM para construir el catálogo), motor determinista (candidatos, *scoring* y selección MMR) y generación de narrativas. La Tabla 11 presenta el desglose para el *dataset* HR con un modelo en la nube (Gemini 2.5 Flash Lite) y con una alternativa local ejecutada mediante Ollama (qwen2.5:0.5b).

Fase	Operación	Gemini 2.5 (s)	Ollama qwen (s)
Fase 1: Enriquecimiento semántico	Inferencia de tipos	0,43	0,39
	Perfilado estadístico	0,08	0,08
	Catálogo semántico (LLM)	6,58	9,08
	Subtotal Fase 1	7,09	9,55
Fase 2: Motor determinista	Candidatos + <i>scoring</i> + MMR	0,25	0,26
Fase 3: Narrativas (LLM)	Narrativa individual de referencia	2,64	5,00
	Estimación secuencial para 6 narrativas	15,83	29,97
	Total de referencia (enriquecimiento + motor + 1 narrativa)	9,98	14,81
	Estimación secuencial (enriquecimiento + motor + 6 narrativas)	23,18	39,79

Tabla 11: Tiempos por fase (*dataset* HR, Gemini 2.5 Flash Lite y Ollama con qwen2.5:0.5b).

Lo que más llama la atención de estos números es la distribución de la latencia. En ambos casos, la Fase 1 concentra la mayor parte del tiempo cuando se genera una sola narrativa de referencia, y dentro de ella la llamada al LLM para el catálogo semántico domina casi por completo; la inferencia de tipos y el perfilado estadístico juntos se mantienen bajo 0,6 s porque son cálculos locales. La alternativa local con qwen2.5:0.5b tardó más que Gemini en esa fase (9,08 s frente a 6,58 s). La Fase 2 —el núcleo heurístico que genera candidatos, calcula métricas y selecciona con MMR— se mantiene prácticamente igual entre proveedores (0,25–0,26 s), lo que confirma que, para HR, la parte determinista opera en tiempos compatibles con interacción. La Fase 3 depende del proveedor LLM: la narrativa individual fue más lenta en Ollama (5,00 s) que en Gemini (2,64 s), por lo que la estimación secuencial para 6 narrativas aumenta de 23,18 s a 39,79 s. En la interfaz actual este costo se reduce operativamente mediante una solicitud por lote y ejecución en segundo plano, por lo que no debe interpretarse como espera bloqueante completa.

La Tabla 12 muestra los tiempos del motor determinista aislado —sin considerar las fases de LLM— para los tres *datasets*, promediados sobre 3 repeticiones.

<i>Dataset</i>	Filas	Cols	Motor (s)	Inferencia (s)	Candidatos (s)
HR (25K)	25 000	17	0,65	0,43	0,09
Retail (200K)	200 000	28	10,34	5,49	2,82
Energy (540K)	540 000	10	2,68	0,41	0,98

Tabla 12: Tiempos del motor determinista por *dataset* (3 repeticiones).

Un punto interesante es que el tiempo del motor escala más con el número de columnas que con el número de filas. El *dataset* Energy tiene 540 000 filas pero solo 10 columnas y se procesa en 2,68 s; en cambio, el *dataset* Retail tiene menos filas (200 000) pero 28 columnas y requiere 10,34 s. Esto se explica porque el espacio de combinaciones bivariadas crece cuadráticamente con el número de columnas, y cada par candidato necesita cálculos estadísticos sobre los datos.

En cuanto a la experiencia percibida por el usuario, la estimación secuencial para una ejecución con 6 narrativas no refleja fielmente lo que ocurre en la interfaz, gracias al diseño de carga progresiva. Una vez que terminan la Fase 1 y la Fase 2 —es decir, a los 7–8 segundos aproximadamente con Gemini y cerca de 10 segundos con la alternativa local en el caso HR medido— el usuario ya puede ver las visualizaciones renderizadas junto con sus métricas y el perfil del *dataset*. Mientras tanto, la Fase 3 opera en segundo plano. El tiempo efectivo de espera para iniciar la exploración es, por tanto, el de las dos primeras fases; la narrativa completa queda disponible de forma posterior y no bloquea la inspección inicial de las vistas.

4.2.3 Diversidad y cobertura

Un aspecto central del OE1 es que las sugerencias no sean variaciones del mismo patrón. Para evaluarlo se contaron los tipos de gráfico distintos, las columnas únicas involucradas y las acciones analíticas cubiertas en cada conjunto final. La Tabla 13 resume estos indicadores.

<i>Dataset</i>	Tipos de gráfico	Columnas únicas	Acciones distintas
HR (25K)	4	5	3 (distribution, relationship, trend)
Retail (200K)	5	6	2 (distribution, relationship)
Energy (540K)	2	3	2 (distribution, relationship)

Tabla 13: Diversidad del conjunto de sugerencias ($K = 6$).

En los *datasets* HR y Retail, donde la variedad de columnas y tipos semánticos es mayor, la aplicación web logra cubrir entre 4 y 5 tipos de gráfico distintos y al menos 2 acciones analíticas diferentes con solo 6 sugerencias. En el caso de Energy, la diversidad es necesariamente menor: con solo 10 columnas —de las cuales 3 categóricas presentan alta asociación entre sí— hay menos patrones distintos que el motor pueda descubrir.

El rol del algoritmo MMR en estos resultados es claro. Sin él, el top-6 del *dataset* HR se poblaría casi por completo de boxplots, ya que existen múltiples combinaciones de categoría y variable cuantitativa con *scores* altos. La restricción *max_same_chart_type=2* junto con la penalización por similitud entre vistas obligan a que como máximo aparezcan 2 boxplots, dejando espacio para histogramas, barras, líneas y mapas de calor.

4.2.4 Transparencia y trazabilidad

El OE3 exige que cada sugerencia esté acompañada de evidencia suficiente para responder “¿por qué esta vista?”. Para verificar que esto se cumple, se inspeccionó la información de trazabilidad asociada a cada *insight* generado. La Tabla 14 muestra un ejemplo concreto con la sugerencia del boxplot bivariado de salarios por nivel jerárquico.

Campo	Valor
<i>Dataset</i>	HR (People Analytics)
Tipo de gráfico	boxplot bivariado
Columnas	job_level (nominal), monthly_salary_clp (cuantitativa)
Acción analítica	relationship
Métrica principal	median_spread = 2,432
Interpretación	Rango de medianas entre categorías, normalizado por desviación estándar global
Score métrica	0,709 (normalizado: $2,432 / (2,432 + 1,0)$)
Score calidad	1,0 (sin nulos, cardinalidad adecuada)
Score base	0,781 ($0,75 \times 0,709 + 0,25 \times 1,0$)
Categorías mostradas	Junior, SemiSenior, Senior, Lider, Gerencia (5)
Observaciones	$n = 25\,000$
Mediana mínima	\$876 000 (Junior)
Mediana máxima	\$2 815 500 (Gerencia)

Tabla 14: Información de trazabilidad (boxplot de salarios por nivel).

Esta estructura permite reconstruir la cadena completa de razonamiento: qué métrica justificó la selección y cuál fue su valor calculado, cómo se compuso el *score* final a partir de la combinación ponderada entre métrica y calidad, qué transformaciones se aplicaron a los datos (por ejemplo, top-*k* de categorías o binning temporal), y qué estadísticas concretas respaldan la interpretación. En la práctica, cada *insight* exportado incluye estos campos, de modo que un revisor puede auditar la recomendación sin necesidad de re-ejecutar el *pipeline*.

Además de la evidencia asociada a las vistas seleccionadas, la aplicación web registra las razones por las que se descartaron alternativas. Entre las causas más frecuentes se observaron:

baja calidad de datos (proporción alta de nulos en las columnas involucradas), alta similitud con una vista ya seleccionada (redundancia detectada por MMR), exceso del límite por tipo de gráfico (`max_same_chart_type=2`), o simplemente un `score` insuficiente para entrar en el top- K . Este registro de descartes es relevante porque completa la trazabilidad: no solo se explica qué se eligió y por qué, sino también qué se dejó fuera y cuál fue el motivo.

4.2.5 Narrativa fiel a la evidencia

La evaluación de la narrativa (OE4) se centró en verificar tres propiedades del componente generativo: que recibe como insumo únicamente evidencia calculada por el motor determinista —sin acceso al *dataset* crudo—, que produce texto estructurado con campos definidos (resumen, evidencia, siguiente acción sugerida), y que no introduce cifras ni interpretaciones que no estén en el paquete factual de entrada.

La separación entre cálculo y lenguaje que se describió en el Capítulo 3 es lo que hace posible este control. El prompt que recibe el LLM contiene un paquete JSON con el tipo de gráfico, las columnas, la métrica principal y su valor, estadísticas de respaldo (percentiles, top- k , coeficientes) y las transformaciones aplicadas. A partir de estos hechos, el modelo debe redactar una explicación legible sin agregar información que no esté presente.

A modo de ejemplo, para el boxplot bivariado de salarios por nivel jerárquico del *dataset* HR, la narrativa generada fue:

“Se observa una alta dispersión en los salarios mensuales entre los diferentes niveles jerárquicos. La mediana salarial de Gerencia (\$2.815.500) y la de Junior (\$876.000) difieren en 2,43 desviaciones estándar, indicando una estructura de compensación estratificada.”

Todos los valores citados —las medianas por categoría y el spread normalizado de 2,43— provienen del motor determinista y están disponibles en la ficha de trazabilidad de la Tabla 14. La narrativa no agrega explicaciones causales ni estadísticas no proporcionadas, lo que es consistente con el diseño de anclaje a hechos. En la muestra revisada de 18 narrativas generadas (6 por cada *dataset*), no se detectaron cifras inventadas ni afirmaciones no respaldadas por el paquete factual.

4.3 Discusión

En conjunto, los resultados permiten sostener que la aplicación web responde razonablemente a los objetivos específicos planteados bajo las condiciones experimentales definidas, aunque con matices que conviene explicitar.

Respecto a la pertinencia (OE1), las sugerencias generadas en los tres *datasets* resultaron coherentes con los tipos de datos presentes y con lo que un analista esperaría encontrar al explorar cada dominio. El modo dirigido mostró, en el caso evaluado, que la reponderación por intención puede reordenar el *ranking* para promover vistas alineadas con una consulta como “analizar *outliers* en salario”. En cuanto a los tiempos (OE2), las mediciones disponibles muestran que el usuario puede recibir visualizaciones y métricas iniciales en menos de 8 segundos para el caso HR; además, los tiempos del motor determinista varían de forma importante con el número de columnas, llegando a alrededor de 10 segundos en Retail. La trazabilidad (OE3) se verifica en la estructura de evidencia que acompaña a cada *insight*: métrica, *score*, calidad y razones de descarte están disponibles como campos exportables. La diversidad (dentro de OE1) se apoya en el algoritmo MMR y en los límites por tipo de gráfico, que reducen la saturación de vistas redundantes cuando el espacio de candidatos es amplio. En la muestra revisada, las narrativas (OE4) respetaron el requisito de anclaje a hechos, sin cifras inventadas ni interpretaciones fuera del paquete factual.

4.3.1 Limitaciones observadas

A lo largo de la validación se identificaron varias limitaciones. Estas limitaciones se refieren al comportamiento observado bajo las condiciones de prueba definidas; el capítulo de conclusiones complementa este análisis con limitaciones de alcance general de la propuesta. La más visible es la dependencia de un proveedor LLM configurado para las fases de enriquecimiento semántico y generación de narrativas. La aplicación web puede usar un proveedor en nube o una alternativa local mediante Ollama, pero ambas fases requieren este componente generativo. Asociada a este requisito está la latencia de las llamadas al LLM, que concentra la mayor parte del tiempo observado o estimado en las fases generativas y cuya variabilidad depende del proveedor, del modelo utilizado y de las condiciones de ejecución.

En cuanto al motor determinista, se observó que su escalabilidad depende principalmente del número de columnas, ya que las combinaciones bivariadas crecen cuadráticamente. Para *datasets* con más de 50 columnas, el espacio de candidatos podría volverse costoso si no se implementa una priorización adicional en la fase de generación. También se detectó que la calidad de la inferencia de intención está ligada a la calidad del catálogo semántico: si los nombres de columnas son códigos internos o abreviaciones poco descriptivas, la aplicación web puede fallar en mapear la consulta del usuario a las columnas correctas. El enriquecimiento con LLM mitiga parcialmente este problema, pero no lo resuelve por completo.

CONCLUSIONES

Síntesis del trabajo realizado

El objetivo de esta memoria fue diseñar, implementar y evaluar una aplicación web que apoye las primeras etapas del Análisis Exploratorio de Datos (EDA) sobre datos tabulares. La solución propone visualizaciones complementarias, genera narrativas explicativas y conserva evidencia suficiente para inspeccionar por qué una vista fue recomendada o descartada.

La decisión de diseño más relevante fue separar estrictamente cálculo y lenguaje. El motor determinista concentra inferencia semántica, perfilamiento estadístico, generación de candidatos, cálculo de métricas, *ranking*, selección diversa y auditoría. El componente generativo, en cambio, se mantiene acotado a dos funciones: enriquecer metadatos del catálogo y redactar explicaciones sobre hechos previamente calculados. Esta frontera permite aprovechar la flexibilidad lingüística de los LLM sin delegarles decisiones analíticas ni cálculos estadísticos.

El *pipeline* resultante se organiza en cinco etapas: ingesta y perfilamiento del *dataset*; interpretación determinista de intención y columnas foco; generación, *scoring* y selección diversa de visualizaciones; generación de narrativas ancladas a evidencia; y auditoría/exportación de resultados. La implementación se realizó en Python, utilizando pandas para manipulación de datos, SciPy para cálculos estadísticos, Altair/Vega-Lite para visualizaciones declarativas y Streamlit para la interfaz web.

Resultados obtenidos e interpretación

La validación se realizó sobre tres *datasets* sintéticos de dominios distintos: recursos humanos, retail y consumo energético. En los casos evaluados, las sugerencias generadas fueron coherentes con los tipos de datos presentes y con patrones esperables de cada dominio, como asimetrías en montos, desbalances categóricos, relaciones monotónicas y diferencias entre grupos. El modo dirigido mostró que la reponderación por intención puede modificar el *ranking* para priorizar vistas alineadas con una consulta específica, como el análisis de *outliers* en salarios.

En rendimiento, las mediciones disponibles muestran que el costo no se distribuye de forma uniforme. El motor determinista fue rápido en el caso HR y aumentó en Retail, donde el número de columnas genera un espacio de candidatos mayor. Esto sugiere una observación importante para el uso práctico: la escalabilidad depende más del ancho del *dataset* que solo del número de filas. Las fases basadas en LLM concentran la mayor parte de la latencia observada o estimada; por ello, la interfaz usa carga progresiva, de modo que las visualizaciones y métricas puedan inspeccionarse antes de que finalicen las narrativas.

El mecanismo de diversidad mediante MMR contribuyó a evitar que el conjunto final quedara dominado por variaciones de una misma vista. En HR, por ejemplo, el top-6 incluyó cuatro tipos de gráfico y tres acciones analíticas, en lugar de concentrarse únicamente en boxplots con alto puntaje. La trazabilidad se verificó mediante la estructura de evidencia asociada a cada *insight*: métrica principal, valor calculado, puntaje compuesto, calidad del dato, transformaciones aplicadas y razones de descarte. En la muestra revisada de 18 narrativas, no se detectaron cifras inventadas ni afirmaciones fuera del paquete factual entregado al LLM.

Aprendizajes de diseño

El principal aprendizaje del trabajo es que el LLM aporta mayor valor como capa de interpretación y comunicación que como motor de decisión. En analítica de datos, una explicación útil no solo debe sonar razonable: debe poder rastrearse hasta cálculos, transformaciones y evidencias concretas. Por eso, el componente generativo resulta más seguro cuando redacta sobre hechos cerrados que cuando decide qué calcular o qué visualizar.

Un segundo aprendizaje es que la recomendación visual no puede evaluarse vista por vista de manera aislada. En EDA, el usuario construye entendimiento comparando perspectivas: distribución, relación, tendencia, segmentación y anomalías. La selección diversa es, por tanto, parte del valor de la solución, no un detalle estético. MMR permite balancear relevancia y cobertura, aunque su desempeño sigue dependiendo de heurísticas y parámetros que deben mantenerse visibles y ajustables.

Finalmente, la transparencia debe diseñarse como salida del sistema, no como explicación posterior. Registrar métricas, puntajes, transformaciones y descartes desde el inicio facilita depurar el motor, justificar recomendaciones y convertir una sesión de análisis en un caso auditable.

Contribuciones principales

La contribución central de esta memoria es una arquitectura neuro-simbólica aplicada a EDA tabular, donde el cálculo determinista y la generación de lenguaje cumplen roles separados y verificables. Esta arquitectura permite combinar recomendación visual, narrativa y auditoría sin convertir al LLM en fuente autónoma de verdad estadística.

El trabajo también entrega una implementación funcional del *pipeline*, con siete métricas de interesancia, ajuste por intención, selección diversa mediante MMR, panel de transparencia y exportación estructurada de evidencia. Además, documenta un patrón de control para sistemas con componentes generativos: limitar el LLM a metadatos y redacción, exigir salida estructurada y anclar toda narrativa a un paquete factual generado por el motor.

Limitaciones identificadas

Mientras que el capítulo de validación reporta limitaciones ligadas a condiciones experimentales concretas (disponibilidad del proveedor LLM, escalabilidad por columnas, calidad del catálogo), las limitaciones que se presentan a continuación tienen un alcance estructural, propio del diseño de la propuesta y no solo de su ejecución en las pruebas.

La aplicación web requiere un LLM para ejecutar el enriquecimiento semántico del catálogo y la narrativa explicativa. Este proveedor puede configurarse en nube o en local mediante Ollama; en ambos casos, el componente generativo forma parte del flujo funcional de la solución. Este requisito también introduce latencia variable según proveedor, modelo, hardware y disponibilidad del servicio.

Otra limitación es la escalabilidad frente a *datasets* anchos. La generación de combinaciones bivariadas crece rápidamente con el número de columnas, por lo que casos con más de 50 columnas podrían requerir prefiltrado, priorización semántica o estrategias de búsqueda más selectivas. Asimismo, la inferencia de intención depende de nombres de columnas y sinónimos suficientemente interpretables; columnas codificadas o abreviadas pueden dificultar el mapeo entre consulta y datos.

La validación realizada permite observar comportamiento sobre tres dominios sintéticos, pero no reemplaza una evaluación con usuarios reales ni con datos productivos de mayor variabilidad. Por ello, los resultados deben interpretarse como evidencia inicial de factibilidad y coherencia del enfoque, no como una medición definitiva de impacto organizacional.

Trabajo futuro

Una primera línea de trabajo futuro es mejorar la priorización de columnas en *datasets* anchos, incorporando prefiltrado basado en señales estadísticas, embeddings semánticos o selección adaptativa de pares. Esto permitiría reducir el costo cuadrático de las combinaciones sin perder cobertura analítica relevante.

También sería valioso fortalecer el mapeo entre lenguaje natural y columnas mediante embeddings o mecanismos de desambiguación asistida por el usuario, especialmente en *datasets* con nombres internos poco descriptivos. Desde el punto de vista de rendimiento, implementar caché del catálogo semántico enriquecido y ampliar las pruebas con modelos locales vía Ollama permitiría caracterizar mejor la relación entre calidad de narrativa, privacidad, latencia y requisitos de hardware.

Otras extensiones posibles incluyen incorporar nuevas familias de visualización, como mapas geográficos o diagramas de flujo, exportar resultados a notebooks o especificaciones Vega-Lite independientes, y agregar mecanismos de retroalimentación del usuario para ajustar pesos y reglas de selección.

Reflexión final

El Análisis Exploratorio de Datos sigue siendo una actividad guiada por juicio humano. La aplicación desarrollada no busca reemplazar ese juicio, sino reducir la fricción inicial: ayudar a decidir por dónde empezar, mostrar evidencia junto con cada recomendación y transformar hallazgos visuales en explicaciones verificables.

La experiencia de esta memoria refuerza una idea central: en sistemas de IA aplicados a datos, la transparencia no es un complemento deseable, sino una condición de uso responsable. La capacidad de generar texto fluido solo aporta valor si el sistema conserva una frontera clara entre lo que calcula, lo que decide y lo que redacta. Esa frontera es la que permite que una recomendación sea no solo útil, sino también revisable, reproducible y defendible.

REFERENCIAS BIBLIOGRÁFICAS

- [Carbonell y Goldstein, 1998] Carbonell, J. y Goldstein, J. (1998). The use of MMR, diversity-based reranking for reordering documents and producing summaries. En *Proceedings of the 21st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 335–336.
- [Cheng y Satyanarayan, 2023] Cheng, A. y Satyanarayan, A. (2023). Visualization authoring with large language models: Opportunities and challenges. *IEEE Computer Graphics and Applications*, 43(5):8–15.
- [Cleveland y McGill, 1984] Cleveland, W. S. y McGill, R. (1984). Graphical perception: Theory, experimentation, and application to the development of graphical methods. *Journal of the American Statistical Association*, 79(387):531–554.
- [Cohen, 1988] Cohen, J. (1988). *Statistical Power Analysis for the Behavioral Sciences*. Lawrence Erlbaum Associates, 2 edición.
- [Cramér, 1946] Cramér, H. (1946). *Mathematical Methods of Statistics*. Princeton University Press.
- [Freedman y Diaconis, 1981] Freedman, D. y Diaconis, P. (1981). On the histogram as a density estimator: l^2 theory. *Zeitschrift für Wahrscheinlichkeitstheorie und Verwandte Gebiete*, 57(4):453–476.
- [Gao et al., 2021] Gao, L., Ma, Y., Zhou, M., Heer, J., y Satyanarayan, A. (2021). Lida: A system for language-independent data visualization. En *Proceedings of the IEEE VIS Workshop on Natural Language Interfaces for Visualization*.
- [Heer y Bostock, 2010] Heer, J. y Bostock, M. (2010). Crowdsourcing graphical perception: Using mechanical turk to assess visualization design. En *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI)*, pp. 203–212.
- [Heer y Shneiderman, 2012] Heer, J. y Shneiderman, B. (2012). Interactive dynamics for visual analysis. *Communications of the ACM*, 55(4):45–54.
- [International Organization for Standardization (ISO), 2018] International Organization for Standardization (ISO) (2018). Iso 9241-11:2018 ergonomics of human-system interaction — part 11: Usability: Definitions and concepts. ISO Standard.
- [Jaccard, 1901] Jaccard, P. (1901). Étude comparative de la distribution florale dans une portion des alpes et des jura. *Bulletin de la Société Vaudoise des Sciences Naturelles*, 37:547–579.
- [Joanes y Gill, 1998] Joanes, D. N. y Gill, C. A. (1998). Comparing measures of sample skewness and kurtosis. *Journal of the Royal Statistical Society: Series D (The Statistician)*, 47(1):183–189.

- [Kautz, 2020] Kautz, H. A. (2020). The third ai summer. En *Proceedings of the AAAI Conference on Artificial Intelligence*, volumen 34, pp. 14637–14637. Issue 9.
- [Keeney y Raiffa, 1976] Keeney, R. L. y Raiffa, H. (1976). *Decisions with Multiple Objectives: Preferences and Value Tradeoffs*. Wiley.
- [Keim et al., 2008] Keim, D., Andrienko, G., Fekete, J.-D., Görg, C., Kohlhammer, J., y Melançon, G. (2008). Visual analytics: Definition, process, and challenges. *Information Visualization*.
- [Lee et al., 2022] Lee, D. J.-L., Tang, D., Agarwal, K., Boonmark, T., Chen, C., Kang, J., Mukhopadhyay, U., Song, J., Yong, M., Hearst, M. A., y Parameswaran, A. G. (2022). Lux: Always-on visualization recommendations for exploratory dataframe workflows. *Proceedings of the VLDB Endowment*, 15(3):727–738.
- [León, 2026] León, F. (2026). Datasets for visualizations system. [Data set]. Zenodo. <https://doi.org/10.5281/zenodo.20361432>.
- [Luo et al., 2018] Luo, Y., Qin, X., Tang, N., y Li, G. (2018). Deepeye: Towards automatic data visualization. En *Proceedings of the 34th IEEE International Conference on Data Engineering (ICDE)*, pp. 101–112.
- [Mackinlay, 1986] Mackinlay, J. D. (1986). Automating the design of graphical presentations of relational information. *ACM Transactions on Graphics*, 5(2):110–141.
- [Microsoft, 2025] Microsoft (2025). Copilot for power bi: Considerations and limitations. Microsoft Learn.
- [Moritz et al., 2019] Moritz, D., Wang, C., Nelson, G. L., Lin, H., Smith, A. M., Howe, B., y Heer, J. (2019). Formalizing visualization design knowledge as constraints: Actionable and extensible. En *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*, pp. 1–12.
- [Munzner, 2014] Munzner, T. (2014). *Visualization Analysis and Design*. CRC Press.
- [Narechania et al., 2020] Narechania, A., Kwon, B. C., Choi, H., Lee, S. C., Drosopoulos, C., y Shah, A. (2020). NI4dv: A toolkit for generating analytic specifications for data visualization from natural language queries. En *Proceedings of the 33rd Annual ACM Symposium on User Interface Software and Technology (UIST)*, pp. 369–380.
- [Nielsen, 2009a] Nielsen, J. (2009a). Powers of 10: Time scales in user experience. Nielsen Norman Group (NN/g), Article. Accessed: 2026-01-04.
- [Nielsen, 2009b] Nielsen, J. (2009b). Response times: The 3 important limits. Nielsen Norman Group.
- [NIST, 2023] NIST (2023). Artificial intelligence risk management framework (ai rmf 1.0). NIST AI 100-1.

- [NIST, 2024] NIST (2024). Artificial intelligence risk management framework: Generative artificial intelligence profile. NIST AI 600-1.
- [NIST/SEMATECH, 2023] NIST/SEMATECH (2023). e-handbook of statistical methods: Exploratory data analysis (eda). NIST/SEMATECH e-Handbook of Statistical Methods.
- [Phillips et al., 2021] Phillips, P. J., Hahn, C. A., Fontana, P. C., Yates, A. N., Greene, K., y Bro-niatowski, D. A. (2021). Four principles of explainable artificial intelligence. Technical Report NISTIR 8312, National Institute of Standards and Technology.
- [Robertson y Zaragoza, 2009] Robertson, S. y Zaragoza, H. (2009). The probabilistic relevance framework: BM25 and beyond. *Foundations and Trends in Information Retrieval*, 3(4):333–389.
- [Satyanarayan et al., 2017] Satyanarayan, A., Moritz, D., Wongsuphasawat, K., y Heer, J. (2017). Vega-lite: A grammar of interactive graphics. *IEEE Transactions on Visualization and Computer Graphics*, 23(1):341–350.
- [Scott, 1979] Scott, D. W. (1979). On optimal and data-based histograms. *Biometrika*, 66(3):605–610.
- [Setlur et al., 2023] Setlur, V., Borkin, M. A., Yalçın, N., y Gove, R. (2023). Evolving the role of natural language in visual data analysis: A survey. *IEEE Transactions on Visualization and Computer Graphics*, 29(1):1–11.
- [Spearman, 1904] Spearman, C. (1904). The proof and measurement of association between two things. *The American Journal of Psychology*, 15(1):72–101.
- [Tableau, 2024] Tableau (2024). Ask data (retired). Tableau Help.
- [Tukey, 1977] Tukey, J. W. (1977). *Exploratory Data Analysis*. Addison-Wesley.
- [Wang et al., 2023] Wang, C., Thompson, J., y Lee, B. (2023). Data formulator: Ai-powered concept-driven visualization authoring. *IEEE Transactions on Visualization and Computer Graphics*.
- [Ware, 2013] Ware, C. (2013). *Information Visualization: Perception for Design*. Morgan Kaufmann, 3 edición.
- [Wongsuphasawat et al., 2016] Wongsuphasawat, K., Moritz, D., Anand, A., Mackinlay, J., Howe, B., y Heer, J. (2016). Voyager: Exploratory analysis via faceted browsing of visualization recommendations. *IEEE Transactions on Visualization and Computer Graphics*, 22(1):649–658.
- [Wongsuphasawat et al., 2017] Wongsuphasawat, K., Qu, Z., Moritz, D., Riley, R., Anand, A., Dietze, F., Kulkarni, D., Mackinlay, J., y Heer, J. (2017). Voyager 2: Augmenting visual analysis with partial view specifications. *IEEE Transactions on Visualization and Computer Graphics*, 23(1):264–273.

ANEXOS

Anexo A: Justificación de constantes de semi-saturación

En este anexo se documenta la justificación de las constantes utilizadas por el *scoring*.

La Tabla 15 detalla la justificación de las constantes C utilizadas para normalizar las métricas principales mediante la función saturante definida en la propuesta de solución. Cada valor representa un punto de semi-saturación: cuando la métrica cruda alcanza C , su *score* normalizado alcanza 0,5.

Métrica	Vista	C	Justificación
skewness	histogram	1,5	$ \text{skew} \geq 1$ se considera distribución “altamente sesgada”; 1,5 sitúa la semi-saturación en una asimetría pronunciada pero no extrema.
outlier_ratio	boxplot univ.	0,05	Bajo una distribución normal, solo $\approx 0,7\%$ de observaciones caen fuera de las vallas de Tukey [Tukey, 1977]; un 5 % representa $\approx 7\times$ la tasa esperada, señal clara de colas pesadas.
median_spread	boxplot biv.	1,0	Rango de medianas grupales dividido por σ global; $1,0\sigma$ equivale a un tamaño de efecto grande según las convenciones de Cohen [Cohen, 1988].
unevenness	bar	0,3	Coefficiente de variación normalizado de frecuencias; 0,3 indica un desbalance moderado entre categorías.
trend_spearman	line	0,3	$ \rho = 0,3$ corresponde a un efecto “mediano” según Cohen [Cohen, 1988], umbral estándar de correlación moderada.
monotonicity	scatter	0,3	$\rho^2 = 0,3$ ($\rho \approx 0,55$) indica que la relación explica el 30 % de la varianza, efecto “grande” según Cohen [Cohen, 1988].
association	mapa de calor	0,3	V de Cramér = 0,3 corresponde a una asociación “mediana” según las convenciones de efecto para tablas de contingencia [Cohen, 1988].

Tabla 15: Justificación de constantes de semi-saturación por métrica.

Anexo B: Prompts utilizados por el componente generativo

Los siguientes *prompts* corresponden a las instrucciones usadas para las dos funciones acotadas del LLM dentro de la aplicación web: enriquecer el catálogo semántico y redactar narrativas sobre hechos ya calculados. Se incluyen porque forman parte del contrato de control del sistema: el modelo recibe metadatos y evidencia compacta, no el *dataset* completo, y debe responder en JSON para facilitar parseo, auditoría y trazabilidad.

B.1 Prompt de sistema para enriquecimiento de catálogo

Eres un analista de datos experto. Devuelve SOLO JSON válido, sin texto adicional.

B.2 Prompt de usuario para enriquecimiento de catálogo

Con base en este resumen de columnas, genera un catálogo semántico enriquecido. Para
→ cada columna entrega: description (1 frase), synonyms (lista corta), role
→ (measure/dimension), unit (cadena corta) y unit_confidence (0 a 1).
Puedes inferir unit SOLO si hay evidencia fuerte en el nombre, role_hint o los
→ rangos (ej: %/pct/porcentaje, usd/clp/eur, km/kg/hrs). Si no hay evidencia, usa
→ unit="" y unit_confidence=0. Si infieres unit, inclúyela en la description.
Responde JSON con el mismo set de columnas.

```
{columns_payload}
```

B.3 Prompt de sistema para generación de narrativas en lote

Eres un analista de datos. Escribe en español claro y natural (para personas no
→ técnicas). Devuelve SOLO JSON válido (sin Markdown). La salida debe ser un
→ objeto JSON donde cada key es un id y cada valor es un objeto con keys: summary,
→ evidence, next_action. No menciones nombres internos de variables/keys del
→ sistema (por ejemplo: metric_context, payload, metric_name, fields_human).

B.4 Prompt de usuario para generación de narrativas en lote

Usa SOLO los hechos entregados (no inventes números ni causalidad). Para cada id
→ entrega: summary (2-3 frases), evidence (4-6 frases con 1-2 cifras del payload y
→ la interpretación del nivel de la señal), y next_action (1-2 frases de siguiente
→ paso analítico).

Estilo: evita jerga de implementación. No uses comillas ni backticks para referirte
→ a variables/métricas. Evita frases del tipo "El <métrica> para <X> a través de
→ <Y>..."; prefiere construcciones naturales como "Al comparar <Y> por <X>..." o "En
→ la distribución de <X>...". No escribas códigos como 'unit_price_clp' o
→ 'product_subcategory' sin explicar su significado; prefiere etiquetas humanas
→ (p. ej. 'Precio unitario (CLP)') y úsalas de forma consistente. No uses nombres
→ de métricas en código (p. ej. 'median_spread'); usa su nombre en español.

Resumen base:
{dataset_summary}

Hechos por vista (lista de items con id + facts):
{items_json}

Los marcadores {columns_payload}, {dataset_summary} e {items_json} son reemplazados en tiempo de ejecución por artefactos generados por el motor determinista: resumen del *dataset*, catálogo base, metadatos por columna, métricas, transformaciones y evidencia agregada por vista.