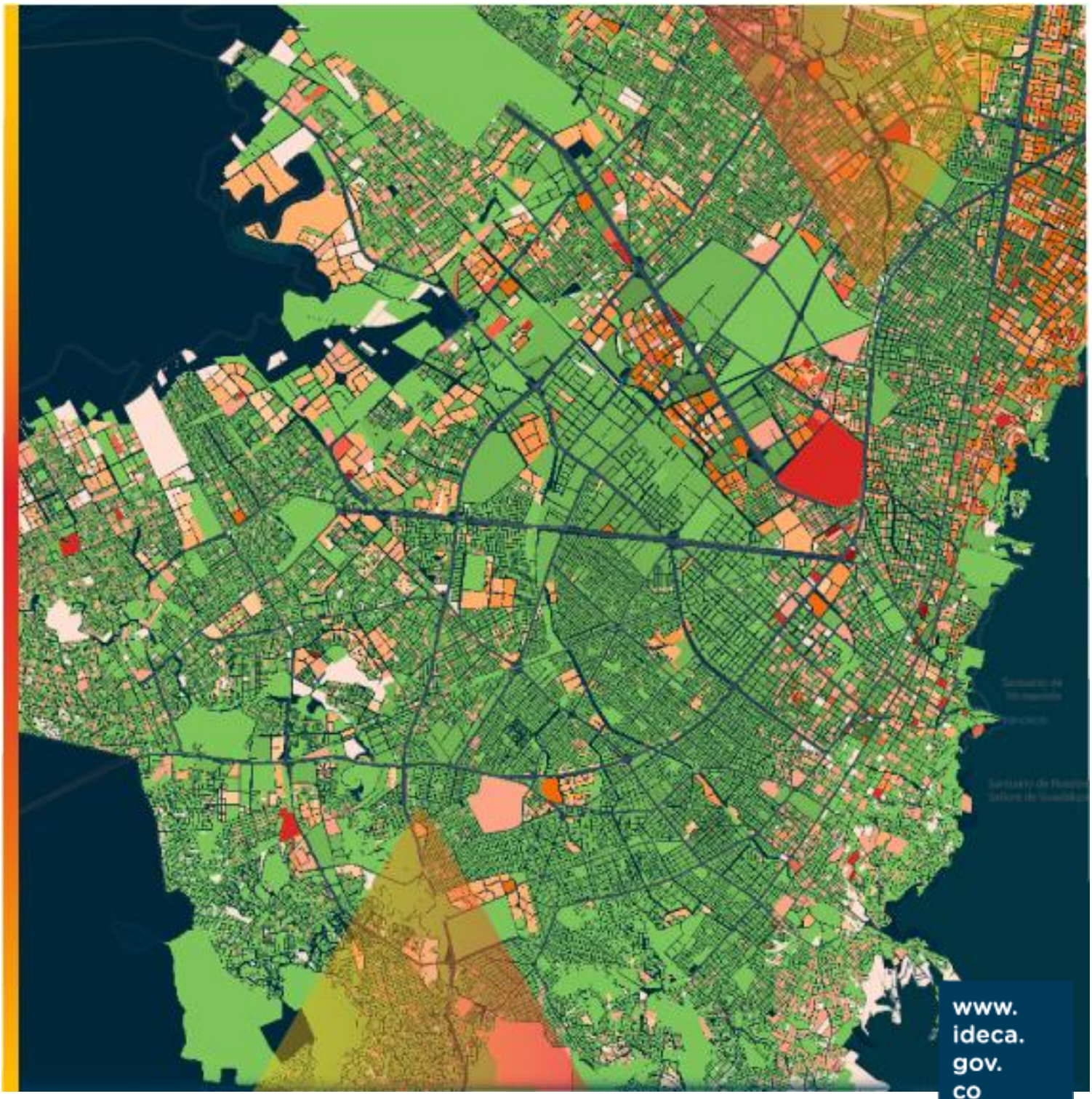
ideca



Localizador Predial FEN - Memoria Técnica

Bogotá, D.C., 30/12/2025/

ALCALDÍA MAYOR DE BOGOTÁ D.C.

ideca

Localizador Predial FEN

Gerente IDECA
Salomé Naranjo Luján

Subgerente de Analítica de Datos
Diego Ricardo Ibarra Rodríguez

Equipo de Trabajo

Profesional Especializado
Subgerencia de Analítica de Datos
Jack Hussein Yurgaky López

Pasante
Juan Diego Montero Murcia



Fecha de creación o actualización: 30/12/2025

Página web: www.ideca.gov.co

Correo electrónico: ideca@catastrobogota.gov.co

Licencia: Attribution 4.0 International (CC BY 4.0)

Control de Versiones

CAMBIOS

Fecha	Autor	Versión	Cambio efectuado
25/11/2025	Juan Diego Montero	1.0	Primera versión del documento. No hay cambios para registrar.
19/12/2025	Juan Diego Montero	1.1	Segunda versión del documento. Actualización de contenido.
30/12/2025	Juan Diego Montero	1.2	Tercera versión del documento. Actualización de contenido.

REVISORES

Nombre	Dependencia
Jack Yurgaky	Subgerencia de Analítica de Datos

Tabla de Contenido

Tabla de Contenido 4
Resumen ejecutivo 7
Introducción 8
Antecedentes 9
Alcance 10
Objetivos 11

1. Inicio 11
1.2. Requerimientos 12
1.3. Estudio de Mercado 13

2. Idear 15
2.1. Proceso de Ideación 15
2.2. Entendimiento de los Datos 15
2.3. Ingeniería de Características 17
2.4. Plataforma Tecnológica 17
2.5. Requisitos / Objetivos 18
2.6. Propuesta 19
2.7. Revisión con el cliente 20
2.8. Actividades 20

3. Prototipar 22
3.1. Prototipo / Modelado 23
3.2. Versión 0.1 24
3.3. Testeo 0.1 24
3.4. Incidencias 0.1 25
3.5. Ajustes 26
3.6. Revisión Cliente 26
3.7. Versión 0.2 27
3.8. Testeo 0.2 28
3.9. Incidencias 0.2 28
3.10. Versión 0.3 29
3.11. Testeo 0.3 29
3.12. Incidencias 0.3 30
3.13. Versión 0.4 30
3.14. Testeo 0.4 31
3.15. Incidencias 0.4 31
3.16. Versión 0.5 32

3.17. Testeo 0.5 32
3.18. Incidencias 0.5 33
3.19. Versión 0.6 33
3.20. Testeo 0.6 34
3.21. Incidencias 0.6 34

4. Operacionalizar - Industrializar 35
4.1. Transferencia 35
4.2. Retrospectiva 36
4.3. Finalización 37

5. Resultados 37

6. Conclusiones 38

7. Bibliografía 39

Lista de tablas

Tabla 1. Phasellus in efficitur neque. Donec elementum pretium rutrum..**¡Error! Marcador no definido.**
Tabla 2. Matriz de contenido Vs Tipos documentales .....**¡Error! Marcador no definido.**

Lista de gráficas

Gráfica 1. Phasellus in efficitur neque. Donec elementum pretium rutrum .....**¡Error! Marcador no definido.**
Gráfica 2. Phasellus in efficitur neque. Donec elementum pretium rutrum .....**¡Error! Marcador no definido.**

Resumen ejecutivo

El Localizador Predial FEN es una herramienta web pensada para hacer preguntas sobre los predios de la ciudad en español y recibir como respuesta una tabla y un mapa claros. En vez de revisar varias bases de datos y mapas por separado, la solución reúne la información catastral, urbanística y de mercado en un punto analítico unificado y agrega ayudas de uso para que cualquiera pueda consultar sin ser experto técnico.

El proyecto se construyó por versiones, mejorando poco a poco los datos, la velocidad de respuesta y la experiencia de uso, se añadieron más variables relevantes, se corrigieron errores en los listados y conteos, se incorporó una página inicial que explica cómo usar la herramienta y se ajustó el diseño a la imagen institucional. Hoy el Localizador Predial FEN es un producto mínimo viable en pruebas dentro de la entidad, que demuestra que es posible responder preguntas ejecutivas sobre el territorio de forma rápida, consistente y comprensible, y deja una base lista para llevarlo a producción y extenderlo a nuevos casos de uso.

Introducción

El caso de uso Localizador Predial FEN nace de una necesidad concreta identificada en la UAECD e IDECA, disponer de un punto analítico unificado que permita a diferentes perfiles de usuarios responder, en minutos y sin depender de consultas ad-hoc predefinidas, preguntas sobre los predios de la ciudad de forma tal que puedan responder a atributos físicos, económicos y normativos. Aunque IDECA ya articula la Infraestructura de Datos Espaciales del Distrito y concentra un volumen creciente de información predial y urbanística, estos datos se encuentran dispersos en múltiples repositorios y servicios, con esquemas y ritmos de actualización heterogéneos e incluso a diferentes geografias. Esta fragmentación incrementa la fricción para el análisis ejecutivo y dificulta que, en una sola vista, se puedan tomar decisiones apoyadas en evidencia geoespacial consistente.

Frente a este contexto, la meta académica y profesional del proyecto es diseñar y desarrollar un producto mínimo viable (PMV) que integre fuentes catastrales, normativas y de mercado inmobiliario en un cubo OLAP geoespacial y habilite consultas en lenguaje natural en español, de manera que los usuarios puedan identificar predios según criterios físicos, económicos, normativos y espaciales, obteniendo siempre que la consulta tenga correspondencia una tabla y un mapa con los predios que responden a la solicitud hecha. El alcance inicial se centra en la integración de la base catastral para Bogotá, la base norma urbanística y del ordenamiento territorial y la base de datos de ofertas inmobiliarias capturadas por el Observatorio Técnico Catastral, acompañadas de una interfaz ejecutiva con tableros de visualización, exportaciones y ayudas de uso, dejando como línea de evolución futura la incorporación de históricos, apertura pública, capacidades predictivas y automatizaciones adicionales. El impacto esperado es estratégico: reducir de horas o días a minutos el tiempo para responder preguntas recurrentes sobre el territorio, estandarizar las consultas y mejorar la coordinación entre áreas al trabajar sobre una misma versión del dato.

El ejercicio se apoya en dos corrientes técnicas complementarias, por un lado, la línea OLAP/SOLAP ha demostrado que los modelos dimensionales con medidas y extensiones espaciales permiten navegar el territorio por jerarquías, realizar agregaciones eficientes y soportar análisis multiescala sobre cubos geoespaciales. Por otro, los avances recientes en interfaces de lenguaje natural a SQL (NL→SQL), incluidos los desarrollos específicos para bases espaciales, han reducido la barrera para que usuarios no especialistas formulen consultas complejas sobre bases relacionales y geográficas. El Localizador Predial FEN se sitúa en la intersección de

ambas líneas al proponer un cubo geoespacial integrado, consultable mediante un módulo NL→SQL apoyado en modelos de lenguaje, y expuesto a través de una interfaz web orientada a las necesidades de los directivos de la entidad.

Metodológicamente el trabajo se estructura en fases que van desde el entendimiento del problema hasta la operacionalización del prototipo. En la etapa de inicio se caracterizaron las necesidades de la UAECD/IDECA y se definieron los requerimientos funcionales y no funcionales del caso de uso. Posteriormente, en la fase de ideación, se documentaron las fuentes de datos, se elaboró el Documento de Identificación de Variables y se diseñó el modelo analítico (cubo OLAP geoespacial, reglas de negocio y criterios de proximidad y pertenencia). A continuación, se construyó y refinó el prototipo mediante procesos ETL, modelado en PostgreSQL/PostGIS, desarrollo de la API en FastAPI, integración con el módulo NL→SQL y construcción de la interfaz web en React/Leaflet, avanzando por versiones sucesivas hasta el Producto Mínimo Viable -PMV actual. Finalmente, se abordaron actividades de prueba, transferencia al ambiente de servidor y sistematización de resultados y lecciones aprendidas. El documento recoge este recorrido, ofreciendo un marco completo para comprender el contexto, el diseño, la implementación y el potencial de evolución del Localizador Predial FEN.

Antecedentes

En Bogotá D.C. la IDECA actúa como articulador distrital de información geoespacial y concentra un volumen creciente de datos prediales y urbanísticos requeridos por la administración y la ciudadanía. A la par, la UAECD necesita responder con rapidez preguntas ejecutivas sobre el territorio para la toma de decisiones.

Históricamente, estas consultas han implicado cruces entre información de Catastro, Norma Urbana y capas temáticas publicadas en distintos repositorios y servicios, con esquemas y ritmos de actualización heterogéneos; esa dispersión eleva la fricción para obtener respuestas consistentes y oportunas.

En el marco de política pública, el Distrito impulsa decisiones basadas en datos, y la infraestructura geoespacial ya expone estándares de interoperabilidad OGC; sin embargo, estos mecanismos facilitan el acceso a datasets pero no resuelven por sí solos la necesidad de un punto analítico unificado para preguntas ejecutivas multifuente.

Desde la ingeniería de datos, la línea OLAP/SOLAP demostró que los modelos dimensionales con dimensiones y medidas espaciales habilitan navegación jerárquica, agregación y análisis territorial eficiente, sentando bases para un cubo geoespacial que integre fuentes y reglas de negocio.

Más recientemente, los avances en lenguaje natural a SQL (NL→SQL) (desde corpus de evaluación de alta complejidad hasta adaptaciones de LLM y aplicaciones espaciales) han reducido la barrera para que usuarios no especialistas formulen consultas complejas y las ejecuten contra bases relacionales y espaciales.

En este recorrido, el problema se formula con claridad, pues no existe un punto analítico unificado que sobre un cubo OLAP geoespacial, permita a directivos plantear preguntas en español y obtener resultados consistentes (tabla y mapa) combinando criterios espaciales (proximidad, pertenencia, intersección) y alfanuméricos (uso, estrato, avalúo, norma). De ahí surge la necesidad del presente anteproyecto.

Alcance

El proyecto busca alcanzar un producto mínimo viable – PMV (MVP por sus siglas en inglés). Integra en una primera versión la base catastral, la norma urbanística y el conjunto de ofertas prediales. Sobre esa integración se construye un cubo OLAP geoespacial el cual se accede mediante un módulo que reconoce lenguaje natural en español y lo convierte a consultas SQL permitiendo responder a búsquedas que contengan variables combinadas como el estrato, el avalúo catastral, el área de terreno o de construcción, el uso permitido para el suelo, las ofertas de mercado y el valor de referencia. El resultado cuenta con una interfaz que ofrece un tablero de control ejecutivo que contiene mapas, tablas y resumen de la consulta realizada. Se acompaña de guía de usuario y la documentación del procedimiento abordado. El producto inicial no incluye históricos, apertura pública, funcionalidades de predicción, ni automatizaciones; esas extensiones quedan planteadas como línea de evolución.

El impacto es directivo-estratégico. La herramienta reduce de horas o días a minutos el tiempo para responder preguntas recurrentes sobre el territorio y apoya decisiones de priorización (inversiones, localización de equipamientos, verificación de cumplimiento normativo). Al estandarizar las consultas y centralizar la vista del dato, mejora la coordinación entre áreas y disminuye la dependencia de análisis ad-hoc. En reuniones de seguimiento, los directivos podrán formular la necesidad

en lenguaje natural, ver el resultado en mapa y cifras, y sostener la discusión con evidencia consistente. El obtener el correcto desarrollo del PMV deja una base instalada para crecer en nuevas fuentes, más consultas y mayor profundidad analítica según la demanda institucional.

Objetivos

Objetivo General

Desarrollar una solución web para IDECA que integre datos prediales y urbanísticos en un cubo SOLAP (cubo OLAP con atributos geoespaciales) y habilite consultas en lenguaje natural para identificar predios según criterios físicos, económicos, normativos y espaciales, apoyando el análisis y la toma de decisiones directivas.

Objetivos Específicos

- Caracterizar y relacionar las fuentes prediales/urbanísticas y sus atributos clave, identificando medidas y dimensiones candidatas y mapeando las preguntas directivas a variables espaciales y alfanuméricas, para establecer el diccionario de datos, las reglas de negocio y los criterios (proximidad, pertenencia, rangos) que guiarán el cubo OLAP geoespacial.
- Producir el cubo OLAP geoespacial integrando las bases Catastral, de Norma Urbana, de Ofertas Inmobiliarias y capas complementarias, mediante procesos de extracción, transformación y carga (ETL), para habilitar consultas SQL sobre dimensiones como el área, uso, estrato y norma de los predios de la ciudad.
- Organizar una solución web que integre un tablero de control y un módulo de lenguaje natural, para convertir requerimientos en español a consultas SQL y presentar resultados en mapas y tablas.

1. Inicio

El caso de uso parte de una necesidad identificada por la UAECD/IDECA, que los directivos puedan responder, en minutos y sin depender de consultas ad-hoc, preguntas territoriales que combinan atributos físicos, económicos y normativos. Aunque IDECA concentra la información oficial, hoy está fragmentada en repositorios y servicios con estructuras heterogéneas, lo que eleva la fricción para el análisis ejecutivo.

1.2. Requerimientos

Se requiere una herramienta que reciba consultas en lenguaje natural y las ejecute sobre un cubo OLAP integrado con fuentes prediales y urbanísticas, entregando resultados en tabla y mapa para uso directivo. En síntesis, el sistema permitirá formular preguntas en español, consultar un repositorio consolidado y visualizar/descargar hallazgos en formatos abiertos con ayudas breves en la interfaz.

En primer lugar, el sistema admitirá consultas en lenguaje natural, aceptando entradas de hasta 512 caracteres y ejecutándolas sobre el cubo OLAP integrado, la respuesta mostrará simultáneamente tabla y mapa de los predios hallados, con paginación cuando existan más de 100 filas. Si la consulta no es entendible, se mostrará un mensaje con tres ejemplos válidos y, en todos los casos, la consulta aplicada quedará visible en una franja superior en modo solo lectura.

En segundo lugar, la consulta se realizará sobre fuentes unificadas, incluyendo como mínimo la base catastral, ofertas de predios y norma urbana; el modelo de datos subyacente seguirá un esquema en estrella o el que disponga la entidad, manteniendo coherencia para las consultas ejecutivas.

A continuación, los resultados se presentarán en tabla y mapa con campos esenciales: identificadores prediales, localidad, estrato, área y norma. El mapa utilizará simbolización simple para los lotes coincidentes; además, se mostrará un contador de resultados visibles y totales, y la tabla permitirá ordenar por cualquier columna. Para facilitar la exploración, al hacer clic en un predio del mapa se resaltará su fila, y al seleccionar una fila la vista del mapa enfocará el predio correspondiente; se incluirá también un botón “Copiar ID predial”.

Por otra parte, el sistema ofrecerá descarga de resultados filtrados en CSV (UTF-8, separador coma) y en Excel con geometría y los mismos atributos de la tabla; los archivos llevarán fecha y hora de generación y se descargarán directamente desde el navegador. Si no existen resultados, la exportación no se habilitará para evitar descargas vacías.

Finalmente, la interfaz incorporará ayudas de aprendizaje rápido: una sección “¿Cómo preguntar?” con cinco ejemplos ejecutables, mensajes de error simples y específicos cuando la consulta no sea válida, y un enlace “Volver a empezar” que limpia la búsqueda con un solo clic. Con ello se reduce la curva de aprendizaje y se favorece el uso autónomo por parte del equipo directivo.

1.3. Estudio de Mercado

En 2005, un estudio de la Université Laval (Québec, Canadá) realizado por Rivest, Bédard y coautores[1] integró sistemas de información geográfica (SIG) y procesamiento analítico en línea (OLAP) para la exploración espaciotemporal interactiva, aplicándolo a escenarios de planificación territorial y análisis de fenómenos geográficos a múltiples niveles (p. ej., navegación por áreas administrativas y períodos de tiempo). El trabajo concluyó que Spatial OLAP (SOLAP) acelera la exploración y el análisis geoespacial para el soporte a decisiones.

Posteriormente, en 2012, Bimonte, Bertolotto, Gensel y Boussaid[2] propusieron un modelo y un álgebra SOLAP que incorporan generalización cartográfica (map generalization) y jerarquías múltiples a distintas escalas; lo situaron en el contexto de productos cartográficos multirresolución y consultas que requieren pasar de nivel detalle a visión agregada en mapas. Demostraron que extender las jerarquías espaciales incrementa la utilidad del análisis territorial en cubos.

En la misma línea, ese mismo año en Corea del Sur, Kyung, Yom y Kim[3] plantearon criterios de actualización geoespacial y diseñaron un esquema estrella para Spatial Data Warehouse con implementación SOLAP, enfocado en procesos operativos de mantenimiento de bases cartográficas (p. ej., decidir qué capas actualizar y con qué prioridad). Evidenciaron que el modelado multidimensional espacial reduce costos y tiempos operativos.

Más adelante, en 2018, en la Universidad de Yale (New Haven, EE. UU.), Yu y colaboradores[4] presentaron Spider, un corpus de referencia para Text-to-SQL de alta complejidad y dominios cruzados, concebido para evaluar sistemas que convierten lenguaje natural en consultas SQL sobre múltiples esquemas. El estudio mostró la dificultad de generalizar a nuevos esquemas y fijó un estándar de evaluación para interfaces de lenguaje natural a SQL.

Luego, en 2023, Google Research (Mountain View, EE. UU.) con Sun y coautores[5] introdujo SQL-PaLM, adaptando LLMs a Text-to-SQL mediante few-shot e instruction fine-tuning, junto con selección de columnas y refinamiento por ejecución; lo validaron en benchmarks exigentes típicos del área (p. ej., colecciones de consultas complejas). Reportaron mejoras sustanciales en dichos benchmarks.

A continuación, en 2024, en ICEIS, Nascimento y colegas[6] analizaron Text-to-SQL en entornos industriales reales, con bases extensas y esquemas heterogéneos propios de organizaciones; compararon resultados con los leaderboards académicos y

observaron brechas, subrayando la necesidad de ajustar prompts, plantillas y esquemas al dominio concreto.

En 2025, en ISPRS IJGI, Wang y coautores[7] evaluaron Text-to-SQL para bases espaciales, considerando consultas con operadores geográficos (p. ej., cercanía e intersección) y proponiendo prompts que incorporan conocimiento geográfico del dominio. Mostraron ventajas para usuarios no expertos y señalaron retos particulares de las funciones espaciales.

Ese mismo año, en IEEE Transactions on Knowledge and Data Engineering, Liu y coautores[8] presentaron NALSpatial, una interfaz de lenguaje natural para bases espaciales que traduce preguntas a consultas ejecutables abarcando tipos como distancia, rango, k-NN y spatial join, esto lo aplicaron a escenarios típicos de SIG y servicios basados en ubicación. Concluyeron que su enfoque mejora la cobertura de consultas espaciales y la precisión frente a baselines previos.

Finalmente, en el International Journal of Geographical Information Science, Yu y colegas[9] presentaron Monkuu, una interfaz de NL a SQL para bases geoespaciales con mapeo dinámico de esquemas y desambiguación geográfica, orientada a que usuarios no expertos formulen consultas espaciales complejas sin conocer el SQL subyacente. Concluyeron que el sistema eleva la exactitud de generación y reduce la barrera de acceso a análisis territoriales.

2. Idear

En este capítulo se plasman todas las ideas de los actores interesados, se identifican necesidades y se definen los objetivos.

2.1. Proceso de Ideación

La ideación inició con espacios breves de contextualización y levantamiento de requerimientos con el equipo de la IDECA; de esas conversaciones salieron historias de usuario que fueron convertidas en requerimientos funcionales, consulta en lenguaje natural sobre un cubo OLAP geoespacial integrado (Catastro, Norma, Ofertas), respuesta en tabla con mapa, con paginación, exportación y ayudas (Sugerencias y ejemplos). Para acotar el alcance se definió un MVP web con límites claros (consulta hasta 512 caracteres) y presentación ejecutiva (contador de resultados, ordenamiento, sincronía mapa y tabla).

Como supuestos verificables, se plantearon hipótesis de datos: (i) el identificador de lote y las claves territoriales (localidad/UPL) permiten unificar fuentes; (ii) la norma urbana puede vincularse por la propiedad horizontal.

Con base en lo anterior, el proceso produjo un listado mínimo de tareas y datos para arrancar: inventariar fuentes y acordar campos clave; proponer el esquema en estrella; ejecutar un ETL inicial para Catastro/Norma/Ofertas y capas de apoyo; ensamblar un prototipo web (mapa con tabla, paginación, exportación, ayudas); y realizar una demostración guiada para recoger observaciones. Con esto, se puede iniciar la construcción del producto mínimo viable dentro del tiempo acordado.

2.2. Entendimiento de los Datos

El acceso a la información se realiza sobre las fuentes oficiales de IDECA, con IDECA–Subgerencia de Analítica de Datos (SAD) como entidad responsable de búsqueda y cargue en carpeta institucional.

Las fuentes primarias consolidadas para el MVP abarcan: Predio (tabla alfanumérica de referencia), Lote y Construcción (estructura predial),

Localidades, UPL y perímetro urbano (división administrativa y límites), Clasificación del Suelo, Reservas y Actuaciones Estratégicas (componente normativo y de ordenamiento), Tratamientos y Ofertas de predios (mercado). En conjunto, el inventario registra para cada conjunto de datos el listado de atributos, su descripción y si son necesarios (SI/NO) en el procesamiento.

En términos de tipos de datos, predominan: (i) geometrías vectoriales (polígonos/puntos) con campos de soporte como SHAPE, SHAPE_Length, SHAPE_Area (geom tipo geometry) en capas territoriales y (ii) tablas alfanuméricas con identificadores y descriptores. Las llaves de unión en el núcleo predial incluyen, Código Predio, Código Construcción y Código Resto (estructura del código predial), junto con CHIP; para división administrativa Localidad y UPL. En Ofertas, se incluyen dirección y claves para el emparejamiento con el predio. En Equipamientos se priorizan identificador único, nombre, dirección, CHIP, UPL y localidad.

Con base en el formato, se priorizan como atributos necesarios (ejemplos):

Predio: Código Manzana, Código Predio, Código Construcción, Código Resto, CHIP, NPN, Tipo de predio, Dirección, Área de terreno, Área construida, Destino económico, Vetustez, Código/Área de uso, Uso PH, entre otros.

Ofertas: Código/Tipo de oferta, dirección, identificadores prediales (estructura de código, CHIP), y campos operativos para análisis de mercado.

Localidades/UPL: Nombre, identificadores únicos y geometría (para pertenencia territorial).

Clasificación del suelo, Reservas, Actuaciones Estratégicas, Perímetro urbano y Centro poblado

Finalmente, se contemplan fuentes externas o secundarias cuando sean necesarias para enriquecer el análisis, en particular: Tablas de Zonas Homogéneas Físicas (tanto urbanas/expansión cómo rurales), sugerencias (Creada a partir de la información del cubo) y tablas con componentes económicos. Con esta especificación (estructurada exactamente como en el formato Excel) se habilita el ETL inicial y la configuración del cubo OLAP geoespacial para ejecutar las consultas del caso de uso.

https://catastrobogotacol.sharepoint.com/:x:/r/sites/ANALTICA_2025/Shared%20Documents/Proyectos/3_CU_Localizador%20Predial%20FEN/Documentos/5_20250915DocumentoIdentificaci%C3%B3nVariables.xlsx?d=wc40b6f7b50b747b0883d6885abdc829a&csf=1&web=1&e=9EOnA6

2.3. Ingeniería de Características

A partir del Documento de Identificación de Variables, Localizador Predial FEN, se realizó una ronda de aclaraciones con el cliente para resolver incoherencias iniciales del modelo. Primero, se confirmó el CHIP como llave principal (estructura de código y BARMANPRE como auxiliares). Asimismo, cuando falte, se permitirá emparejar por geocodificación de dirección y proximidad a la geometría del predio. También se estandarizaron unidades y dominios: área en m², precios en COP del corte y estrato como categoría numérica controlada.

Con estas definiciones, la preparación de datos se orientó a construir los insumos analíticos del MVP. Se ejecutarán limpiezas y estandarizaciones básicas: normalización de identificadores, tipificación de columnas, deduplicación, trim y mayúsculas en campos clave, verificación topológica de geometrías (cierre de polígonos, self-intersections), homologación de CRS, y geocodificación de direcciones de ofertas para su cruce espacial.

A partir de allí se derivan las características (features) necesarias para las consultas tipo y el tablero ejecutivo:

Medidas base: área terreno, área construcción.

Dimensiones/atributos: localidad, UPL, estrato.

Pertenencia/intersección: banderas para inclusión en reservas y clasificación del suelo.

Mercado: vínculo a ofertas y valor oferta.

El resultado de la ingeniería de características es un conjunto coherente y documentado de variables, relaciones y materializaciones que resuelven las ambigüedades iniciales y dejan listo el procesamiento/transformación (ETL) para poblar el cubo OLAP geoespacial y habilitar las consultas del caso de uso.

2.4. Plataforma Tecnológica

El repositorio de datos será PostgreSQL con extensión PostGIS, donde se modelará un esquema en estrella con índices espaciales y vistas/materializaciones para soportar agregaciones tipo OLAP. La capa de servicios se implementará con FastAPI, que recibirá la consulta en lenguaje SQL y ejecutará la consulta validada contra Postgres; para ello se impondrán barandas básicas (solo SELECT, lista blanca de tablas y funciones espaciales).

La interfaz de usuario se construirá en React, utilizando Leaflet para la visualización cartográfica y Proj4Leaflet cuando se requiera un sistema de referencia oficial, de modo que el mapa y la tabla se sincronicen y permitan paginación, ordenamiento y exportación. De forma complementaria, Power BI Desktop consumirá vistas del modelo para un tablero preliminar orientado a la demostración ejecutiva, mientras que el tablero final quedará integrado en el propio frontend. Para habilitar NL a SQL sin dependencias externas se empleará Ollama en localhost como motor de LLM: en el MVP podrá usarse un modelo del catálogo para generar SQL bajo control de la API. Esta plataforma posee compatibilidad con datos espaciales institucionales y una ruta clara para demostrar el caso de uso con tiempos de respuesta adecuados y control total del entorno.

2.5. Requisitos / Objetivos

Con base en la Ingeniería de Características y los acuerdos con los interesados, el alcance mínimo viable (MVP) se enfoca en permitir consultas ejecutivas sobre predios combinando criterios espaciales y alfanuméricos. En primer lugar, el comportamiento de consulta será vía lenguaje natural en el frontend (entrada de hasta 512 caracteres) que se traduce a SQL mediante Ollama; como salvaguarda, se habilita un modo avanzado para enviar SQL validada. Toda consulta se ejecuta exclusivamente sobre el cubo en PostgreSQL/PostGIS y retorna tabla y mapa sincronizados, ordenamiento por columna, contador de resultados y resaltado bidireccional mapa y tabla. Además, se mostrará la consulta aplicada en una franja superior (solo lectura). En segundo lugar, la solución permitirá exportar resultados filtrados en CSV (UTF-8) y Excel; si no hay resultados, la exportación se deshabilita.

En cuanto a datos y preparación, el MVP integra como mínimo Catastro, Norma Urbana, Ofertas y Equipamientos. Se adopta CHIP como llave principal (con BARMANPRE auxiliar).

Respecto a la plataforma, se usará PostgreSQL en conjunto con PostGIS como repositorio, FastAPI como API y React con Vite y Leaflet en el frontend. Ollama correrá en como motor LLM. El tablero preliminar se entrega en Power BI Desktop conectado a vistas del cubo, mientras que el tablero final residirá en el propio frontend.

Objetivos del caso de uso:

- Caracterizar y alinear las fuentes (Catastro, Norma, Ofertas y complementarias) y mapear las preguntas directivas a variables espaciales/alfanuméricas, consolidando diccionario de datos, reglas de negocio y sinónimos que soporten la traducción NL→SQL.
- Construir el cubo OLAP geoespacial en PostgreSQL/PostGIS mediante ETL inicial, materializando hechos/dimensiones y vistas de consulta para tiempo, ubicación, uso, estrato y norma, con índices espaciales y agregaciones básicas.
- Entregar el MVP web con consulta NL→SQL, tabla+mapa sincronizados, paginación y exportación (CSV/Excel), más un tablero preliminar en Power BI; todo validado con ejemplos representativos del uso directivo.

2.6. Propuesta

La solución se orienta a un punto analítico unificado que permita a los directivos formular preguntas en español y obtener, en segundos, tabla y mapa con los predios que cumplen criterios físicos, normativos y espaciales. Con base en el entendimiento de los datos, la propuesta integra tres piezas: (i) cubo OLAP geoespacial en PostgreSQL/PostGIS, (ii) servicio NL→SQL en FastAPI apoyado en Ollama, y (iii) una interfaz web en React + Leaflet con exportación y ayudas de uso.

El procesamiento parte de un ETL local que consolida Catastro, Norma Urbana, Ofertas y Equipamientos. Se adopta CHIP como llave principal (con BARMANPRE y NPN como auxiliares), se homologa el CRS oficial, se corrigen geometrías, se normalizan dominios (área en m²; valores en COP del corte; estrato numérico; uso según catálogo) y se habilita la geocodificación para ofertas sin vínculo directo. Con esos insumos se modela un esquema en estrella.

La traducción de lenguaje natural a SQL se realizará directamente en Ollama, inyectando al modelo un resumen del esquema (tablas, columnas y relaciones) y dos o tres ejemplos representativos del dominio. El system prompt obliga a devolver solo SQL en PostgreSQL/PostGIS). Este enfoque mantiene la simplicidad del MVP sin sacrificar control ni seguridad en el backend.

La interfaz web expone un buscador en español, muestra chips con los criterios detectados, sincroniza tabla y mapa (selección bidireccional), permite paginar y exportar los resultados en CSV y Excel con sello de fecha.

2.7. Revisión con el cliente

En la sesión de revisión, la directora solicitó incorporar un apartado específico sobre cómo utilizar la herramienta (flujo básico de consulta, interpretación de resultados y opciones de exportación) y otro dedicado a explicar con claridad las variables sobre las que se puede consultar (atributos físicos, económicos y normativos disponibles en el cubo). Estas observaciones orientan la documentación hacia las necesidades reales de los futuros usuarios, facilitando la apropiación de la solución y reduciendo la curva de aprendizaje.

2.8. Actividades

De forma general, el desarrollo del Localizador Predial FEN se organizará en un conjunto de actividades encadenadas que cubren desde el entendimiento del problema hasta la entrega del producto mínimo viable (MVP) y su documentación asociada.

En primer lugar, se realizará el inventario y caracterización de fuentes (Catastro, Norma Urbana, Ofertas y capas complementarias), identificando llaves de unión, atributos relevantes y criterios de calidad. A partir de ello, se definirá el modelo de datos analítico (esquema en estrella, hechos y dimensiones) y se detallarán las reglas de negocio y dominios necesarios para el cubo SOLAP.

En segundo lugar, se ejecutarán las actividades de extracción, transformación y carga (ETL) para construir el cubo OLAP geoespacial en PostgreSQL/PostGIS, limpieza y normalización de campos, homologación de unidades y CRS, verificación de geometrías, derivación de variables (áreas, banderas normativas, vínculos a ofertas) y materialización de vistas de consulta.

Así, se desarrollará la capa de servicios en FastAPI, encargada de recibir consultas SQL (generadas por el módulo de lenguaje natural, validarlas según las barandas definidas y ejecutar las consultas sobre el cubo. De manera complementaria, se configurará el módulo NL→SQL con Ollama (prompts,

ejemplos y reglas de salida) para traducir consultas en español a SQL ejecutable.

En paralelo, se implementará la interfaz web en React, integrando el buscador en lenguaje natural, la tabla de resultados con paginación y ordenamiento, el mapa sincronizado (Leaflet) y los mecanismos de exportación (CSV/Excel), así como las ayudas de uso.

3. Prototipar

En esta fase se materializó un prototipo funcional de alta fidelidad del Localizador Predial FEN que integra de forma operativa las tres capas definidas en la propuesta. En la capa de datos se consolidó el cubo OLAP geoespacial en PostgreSQL/PostGIS a partir del ETL local sobre Catastro, Norma Urbana, Ofertas y capas complementarias. Este cubo se expone mediante la vista materializada mv_predio_flat, con dominios normalizados, geometrías y campos derivados listos para consulta analítica.

Sobre esta base se implementó una API REST en FastAPI que actúa como puerta de entrada al cubo, pero sin incorporar el modelo de lenguaje. El backend ofrece endpoints para listar predios con filtros estructurados, ejecutar consultas SQL validadas sobre mv_predio_flat, obtener resultados geoespaciales optimizados para mapa, descargar los resultados en CSV/XLSX y consultar sugerencias desde la tabla de apoyo de sugerencias.

La traducción de lenguaje natural a SQL se prototipó mediante un servicio JavaScript dedicado que consume la API de Ollama expuesta en un puerto propio y accesible a través de un proxy de Vite. La SPA en React invoca la función chatToSQL, que envía al modelo un system prompt especializado en Text-to-SQL, junto con la pregunta del usuario, y recibe como respuesta únicamente la sentencia SQL. Esta cadena SQL se reenvía al backend a través del endpoint /predios/sql y se reutiliza para alimentar tanto la tabla como el mapa. De este modo, el modelo de lenguaje permanece desacoplado del backend y opera como un servicio independiente.

En la interfaz web se materializó el flujo completo "pregunta en español → SQL → tabla + mapa". El prototipo incluye un buscador en lenguaje natural, un panel de chat donde se muestran las interacciones usuario–modelo (incluida la consulta SQL generada, con opción de copiarla), chips con los criterios detectados, autocompletado de valores a partir del endpoint de sugerencias y una tabla de resultados con paginación, ordenamiento y exportación a CSV/Excel con sello de fecha. El mapa (Leaflet, cargado vía CDN) se sincroniza con la tabla, permitiendo la selección bidireccional de predios y la exploración espacial de los resultados. En conjunto, este prototipo ha permitido validar el flujo de extremo a extremo, identificar mejoras en desempeño y usabilidad, y recoger feedback temprano sobre la claridad de las variables y la experiencia de uso para los directivos.

3.1. Prototipo / Modelado

Describir lo relacionado con la materialización del diseño del proyecto se ve reflejada en el prototipo modelado a partir de la propuesta elaborada. Específicamente se ejemplifican los requisitos con la posible solución que guarda una parametrización ampliable y modificable.

La materialización del diseño del Localizador Predial FEN se refleja en un prototipo modelado que conecta de forma explícita los requisitos con una solución parametrizable y ampliable. A partir de la propuesta, se consolidó una arquitectura por capas: un cubo OLAP geoespacial en PostgreSQL/PostGIS como fuente única de verdad, una API en FastAPI que expone servicios genéricos de consulta y exportación sobre la vista mv_predio_flat, y una interfaz web en React que orquesta el flujo "pregunta en español → SQL → tabla + mapa". Esta separación permite que cada componente evolucione sin romper el conjunto.

En el nivel de datos, el modelado se concretó en un esquema en estrella donde los hechos prediales se describen mediante dimensiones físicas, económicas y normativas. El prototipo no "cablea" consultas fijas, sino que trabaja sobre una vista materializada única con dominios normalizados, de este modo, los filtros, listados y exportaciones se construyen en torno a nombres de campos y descripciones que pueden ampliarse sin rediseñar la aplicación. La lista de atributos visibles, sus etiquetas amigables y su agrupación temática se manejan como configuración, ejemplificando el requisito de contar con una parametrización modificable.

En la capa de interacción, el modelado se concreta en componentes reutilizables cómo un panel de chat que encapsula el diálogo con el modelo de lenguaje y la generación de SQL, un panel de mapa (Leaflet) que consume GeoJSON proveniente del backend, y una tabla de resultados con paginación, ordenamiento y exportación. El servicio NL→SQL se configuró como un recurso externo accesible vía proxy, de manera que el frontend puede cambiar de modelo o ajustar el prompt sin alterar la API ni el cubo. En conjunto, el prototipo modelado ejemplifica cómo los requisitos de consulta en español, combinación de criterios físicos/normativos/espaciales y visualización integrada tabla–mapa se resuelven mediante una solución modular, parametrizable y preparada para futuras ampliaciones.

3.2.Versión 0.1

La versión 0.1 corresponde al primer ensamblaje funcional del Localizador Predial FEN, centrado en comprobar la integración básica de los datos con la plataforma tecnológica propuesta. En esta iteración inicial se conectó el backend en FastAPI con una versión preliminar del cubo OLAP en PostgreSQL/PostGIS, construido solo con un subconjunto de campos prediales. El objetivo principal era verificar que la vista de consulta pudiera responder de forma consistente a las preguntas básicas y que la base de datos se comunicara correctamente con la capa de servicios.

Así mismo, se integró la IA usando el modelo Gemma3:4b (ejecutado localmente mediante Ollama) para traducir preguntas en español a sentencias SQL simples. La interfaz web, en esta versión, carecía de diseño visual elaborado pues no se habían incorporado estilos, componentes de experiencia de usuario ni funciones complementarias (no había exportación a CSV/Excel, ni ayudas de uso, ni sugerencias avanzadas). La tabla de resultados y el flujo de consulta eran muy básicos, pero suficientes para demostrar el ciclo extremo a extremo "pregunta → SQL → resultados" y evidenciar que la arquitectura propuesta (cubo + API + frontend + IA) era viable antes de avanzar hacia versiones más completas y usables.

3.3.Testeo 0.1

Describir la fiabilidad de la primera versión del prototipo; describir qué pruebas se realizaron; describir si la interoperabilidad de los datos entre las diversas fuentes vinculadas que hacen parte de la solución es continua.

En la versión 0.1 el testeo se enfocó principalmente en verificar la viabilidad técnica del flujo extremo a extremo más que en pruebas exhaustivas de calidad. Se realizaron pruebas manuales de consulta donde, a partir de preguntas sencillas en español, se validó que el modelo Gemma3:4b (ejecutado en Ollama) fuera capaz de interpretar la intención del usuario y devolver una sentencia SQL mínima pero ejecutable sobre la vista preliminar del cubo OLAP. Para cada consulta se midió de forma empírica el tiempo que tardaba la IA en generar el SQL y el tiempo total transcurrido hasta que

los resultados llegaban y se reflejaban en el frontend, con el objetivo de estimar la latencia del ciclo "pregunta → IA → backend → base de datos → frontend".

En cuanto a la interoperabilidad, la versión 0.1 únicamente permitía comprobar la continuidad básica entre capas esto debido a que el backend lograba conectarse a la base de datos PostgreSQL/PostGIS con el cubo preliminar y el frontend podía consumir las respuestas expuestas por la API. Aunque los datos todavía no incorporaban direcciones ni Zonas Homogéneas Físicas (ZHF), y los dominios no estaban completamente depurados, se confirmó que la cadena de integración tecnológica funcionaba sin rupturas (consulta desde el navegador, llamada a la IA, ejecución en el backend y retorno de resultados). No se desarrollaron en esta etapa baterías formales de pruebas ni escenarios complejos de uso; el énfasis estuvo en validar la fiabilidad mínima del flujo y en detectar de manera temprana tiempos de respuesta y posibles cuellos de botella para ser abordados en versiones posteriores.

3.4. Incidencias 0.1

Durante el testeo de la versión 0.1 la principal incidencia detectada estuvo relacionada con los tiempos de respuesta del flujo completo de consulta. En particular, se observó que el modelo Gemma3:4b, ejecutado localmente mediante Ollama, podía tardar entre uno y dos minutos en interpretar la consulta en lenguaje natural y devolver la sentencia SQL correspondiente. A esto se sumaba el tiempo adicional requerido para que el backend ejecutara la consulta sobre el cubo preliminar y enviara los resultados al frontend, lo que hacía que la experiencia percibida por el usuario fuera lenta y poco adecuada para escenarios de consulta ejecutiva.

Aunque la interoperabilidad técnica entre capas (frontend → IA → backend → base de datos) se comportó de manera continua y sin errores críticos, esta latencia elevada se identificó como una restricción importante para los requisitos no funcionales del sistema, especialmente en términos de desempeño y usabilidad. Como resultado, se reconoció la necesidad de: (i) explorar modelos de lenguaje más ligeros o configuraciones alternativas que reduzcan el tiempo de inferencia y (ii) optimizar la cadena de llamadas entre frontend, servicio de IA y API de datos.

3.5. Ajustes

En la versión 0.1 la evaluación del algoritmo/modelo se centró en verificar, de manera básica, si el flujo de lenguaje natural a SQL era funcional y compatible con el alcance propuesto. Para ello se realizaron pruebas manuales con consultas sencillas en español (filtrar predios por localidad, estrato o área, entre otros criterios básicos) y se revisó que el modelo Gemma3:4b, ejecutado localmente en Ollama, generara sentencias SQL sintácticamente válidas y ejecutables sobre la vista preliminar del cubo OLAP. La validación consistió en comprobar que las consultas no produjeran errores en el backend y que devolvieran resultados coherentes con la lógica esperada del dominio predial.

En esta etapa no se desarrolló una campaña formal de parametrización del modelo. Se trabajó con la configuración por defecto del despliegue local de Gemma3:4b y con un prompt de sistema mínimo orientado a que la respuesta fuese exclusivamente una sentencia SQL en PostgreSQL/PostGIS. Más allá de estas decisiones iniciales, no se aplicaron aún pruebas sistemáticas de ajuste de hiperparámetros (temperatura, top-p, longitud máxima, etc.), ni comparaciones cuantitativas con otros modelos, ya que el objetivo principal era validar el funcionamiento general del flujo NL→SQL.

Los resultados de esta validación fueron mixtos frente al alcance propuesto. Desde el punto de vista funcional, el modelo demostró ser capaz de producir consultas SQL ejecutables y de cerrar el ciclo "pregunta → SQL → resultados", por lo que puede considerarse aceptable como prueba de concepto. Sin embargo, los tiempos de respuesta observados (entre uno y dos minutos para obtener la sentencia SQL y completar el recorrido hasta el frontend) se alejaron de los requisitos de desempeño esperados para un uso directivo. En consecuencia, la aceptabilidad del modelo en 0.1 quedó condicionada: fue suficiente para demostrar la viabilidad técnica, pero no resultó adecuado como solución definitiva, lo que motivó la búsqueda de alternativas y ajustes en versiones posteriores.

3.6. Revisión Cliente

En la revisión de la versión 0.1 con la directora de la UAECD se mostró el flujo completo del prototipo, evidenciando que la solución ya es capaz de generar

información analítico–predial básica, listados de predios con sus principales atributos físicos, económicos y normativos, filtrados mediante consultas en lenguaje natural que se traducen a SQL y se reflejan en la tabla de resultados. Aunque el cubo utilizado aún era preliminar, la directora reconoció que la herramienta avanza en la dirección del objetivo planteado, un punto analítico unificado que permita responder, en un solo lugar, preguntas sobre los predios dirigidas a la toma de decisiones.

Su retroalimentación se centró en facilitar la apropiación por parte de los futuros usuarios más que en cambios funcionales de fondo. En concreto, solicitó incorporar una sección explícita sobre cómo usar la herramienta y otra en la que se detallen las variables sobre las que se puede consultar (atributos físicos, económicos y normativos disponibles). Estas observaciones orientan los ajustes posteriores hacia la documentación y la experiencia de uso, reforzando la alineación del Localizador Predial FEN con las expectativas de los directivos de la entidad.

Presentación.pptx

3.7. Versión 0.2

En la versión 0.2 se consolidó la integración de los datos con la plataforma tecnológica incorporando funciones complementarias sobre el mismo núcleo arquitectónico de la solución. Se mantuvo el flujo NL→SQL con el modelo de IA (aún con tiempos de respuesta elevados), pero se enriqueció la experiencia de uso, se implementó la selección sincronizada entre tabla y mapa, de modo que al hacer clic sobre un registro en la tabla se enfoca el predio correspondiente en el mapa y viceversa; se estabilizaron las exportaciones a CSV y Excel, asegurando que los filtros aplicados se reflejen correctamente en los archivos descargados; se agregó el sistema de sugerencias en el chat de la IA para guiar al usuario en la formulación de consultas; y se configuró la visualización diferenciada de lotes y construcciones en el mapa, junto con otras funcionalidades de apoyo al core del aplicativo.

Desde el punto de vista de datos, la versión 0.2 incorporó las Zonas Homogéneas Físicas tanto urbanas/expansión como rurales, incluyendo sus etiquetas descriptivas dentro del cubo y exponiéndolas en la interfaz como

variables consultables. En respuesta directa a la retroalimentación de la directora, se diseñó además una Landing Page que explica el paso a paso de uso del Localizador Predial FEN y presenta el conjunto inicial de variables con las que se puede consultar, mejorando la comprensión del alcance de la herramienta y su alineación con las necesidades de los usuarios directivos.

3.8. Testeo 0.2

Describir la fiabilidad de la segunda versión del prototipo; describir qué pruebas se realizaron; describir si la interoperabilidad de los datos entre las diversas fuentes vinculadas que hacen parte de la solución es continua.

En la versión 0.2 el testeo se centró en comprobar el funcionamiento básico y la estabilidad de las nuevas funcionalidades más que en aplicar pruebas formales exhaustivas. Se verificó de manera manual la sincronización tabla–mapa (comprobando que al seleccionar un registro en la tabla se enfocara el predio correspondiente en el mapa y que la interacción fuera fluida), el correcto funcionamiento de las exportaciones a CSV y Excel (revisando que los archivos descargados reflejaran los filtros aplicados), la aparición y uso de las sugerencias en el chat de la IA, así como la visualización de lotes y construcciones y la presencia de las Zonas Homogéneas Físicas urbanas/expansión y rurales con sus etiquetas correspondientes.

Las pruebas realizadas mostraron que el núcleo de la solución es más fiable que en la versión anterior y que la integración entre las diferentes capas tecnológicas (cubo OLAP, API en FastAPI, frontend en React y mapa en Leaflet) se mantiene continua, sin rupturas visibles en el flujo de datos. No obstante, se confirmó que persiste la incidencia ya identificada en la versión 0.1: los tiempos de respuesta del modelo de IA siguen siendo elevados, por lo que, aunque la interoperabilidad entre fuentes y componentes es estable, el desempeño percibido por el usuario aún no alcanza el nivel deseado para un contexto de consulta ejecutiva.

3.9. Incidencias 0.2

En el testeo de la versión 0.2 se confirmó que persiste la incidencia ya identificada en la 0.1 relacionada con los altos tiempos de respuesta del modelo de IA, que sigue afectando la fluidez percibida por el usuario. Además, se detectaron nuevos problemas funcionales en la capa de presentación, por un lado, un bug en el paginado de la tabla de resultados, que en ciertos escenarios no respeta correctamente el cambio de página o la cantidad de registros mostrados; por otro, inconsistencias en el conteo de predios, donde el número total reportado no siempre coincide con los registros realmente visualizados. Estas incidencias acotan los requisitos no funcionales de desempeño y los requisitos funcionales de presentación de resultados, y orientan las siguientes iteraciones hacia la corrección del paginado, la revisión de los cálculos de totales y la optimización del flujo NL→SQL para mejorar la experiencia de los usuarios.

3.10. Versión 0.3

En la versión 0.3 se realizó una depuración enfocada en la estabilidad del prototipo, se corrigieron los bugs identificados en iteraciones previas, incluyendo los problemas de paginado en la tabla de resultados y las inconsistencias en el conteo de predios, de modo que la cantidad total reportada, lo visualizado en pantalla y lo exportado quedaran alineados. Con esto, el flujo de consulta y navegación de resultados se volvió más consistente y fiable para el usuario.

El cambio más relevante fue la sustitución del modelo Gemma3:4b por qwen3-coder:480b-cloud, aprovechando su ejecución en la nube para mejorar los tiempos de respuesta en el contexto de pruebas. La integración se mantuvo mediante el mismo servicio de IA, pero se ajustaron las reglas y el prompt del modelo para ampliar las posibilidades de búsqueda (consultas más variadas, combinaciones de filtros físicos, económicos y normativos, y mejor manejo de condiciones complejas) garantizando que la salida siguiera siendo exclusivamente SQL válido sobre mv_predio_flat. En conjunto, la versión 0.3 consolida un prototipo más estable y con un módulo NL→SQL considerablemente más ágil y expresivo que en las versiones anteriores.

3.11. Testeo 0.3

En el testeo de la versión 0.3 se pudo percibir una mejora notable en el desempeño del módulo NL→SQL: el tiempo de respuesta de la IA pasó de rangos de minutos a tiempos del orden de segundos, situándose aproximadamente entre 5 y 10 segundos según la complejidad de la consulta. A través de pruebas manuales se verificó que las sentencias SQL generadas por el nuevo modelo qwen3-coder:480b-cloud se ejecutaran correctamente sobre mv_predio_flat y que los resultados llegaran al frontend sin errores, manteniendo la interoperabilidad continua entre la base de datos, la API y la interfaz web. Adicionalmente, se confirmó que las correcciones introducidas en el paginado y en el conteo de predios se comportan de forma estable, por lo que la versión 0.3 ofrece un flujo de consulta más ágil y consistente frente a las iteraciones anteriores.

3.12. Incidencias 0.3

No hubo incidencias debido a que esta versión trató principalmente de arreglar los problemas de la anterior.

3.13. Versión 0.4

La versión 0.4 se centró en el refinamiento visual y la mejora de la experiencia de uso del Localizador Predial FEN. Se ajustó el diseño de la interfaz incorporando la paleta de colores, símbolos y lineamientos institucionales de la UAECD, reorganizando los paneles para lograr una distribución más clara entre buscador, tabla y mapa. Esto permitió pasar de un prototipo funcional a una herramienta con aspecto más cercano a un producto listo para socialización interna, mejorando la legibilidad y la coherencia gráfica del aplicativo.

En el plano funcional, se añadió la interpretación de consultas espaciales del tipo “cerca de” o “alrededor de” un lugar o referencia. A partir de este cambio, la IA es capaz de traducir expresiones como “predios cerca de Teusaquillo” o “lotes alrededor de X barrio” en consultas SQL que incorporan criterios de proximidad, aprovechando las capacidades geoespaciales del cubo olap. Con ello, la versión 0.4 fortalece el objetivo analítico del

Localizador, al acercarse más a la forma en que los directivos formulan sus preguntas sobre el territorio.

3.14. Testeo 0.4

En el testeo de la versión 0.4 se validó principalmente el comportamiento de las nuevas consultas de proximidad ("cerca de", "alrededor de") y su integración con la interfaz ya ajustada al diseño institucional. Mediante pruebas manuales se comprobó que la IA genera sentencias SQL que incorporan correctamente criterios espaciales de cercanía y que la base de datos devuelve los resultados esperados, manteniendo la interoperabilidad continua entre cubo OLAP, API y frontend. No se observaron fallos funcionales en la visualización ni en la sincronización tabla–mapa.

La principal observación fue de desempeño, cuando se ejecutan consultas de proximidad de manera aislada, los tiempos de respuesta siguen siendo aceptables; sin embargo, al encadenar varios casos de "lugares cercanos a" o aplicarlos sobre zonas muy extensas se identificó un aumento significativo en los tiempos de respuesta de la base de datos, lo que sugiere la necesidad de optimizar las consultas espaciales y/o ajustar los parámetros de uso en escenarios intensivos.

3.15. Incidencias 0.4

En la versión 0.4 la incidencia más relevante está asociada al rendimiento de las consultas espaciales de proximidad. Se constató que, aunque funcionalmente correctas, las consultas del tipo "cerca de" degradan el tiempo de respuesta cuando se utilizan de forma repetida o sobre áreas con gran cantidad de predios, generando latencias mayores a las observadas en consultas más simples. Esta situación acota los requisitos no funcionales de desempeño para el módulo geoespacial y orienta las siguientes iteraciones hacia la optimización de índices, el refinamiento de la lógica de proximidad y/o la definición de límites y recomendaciones de uso para mantener una experiencia fluida en el contexto directivo.

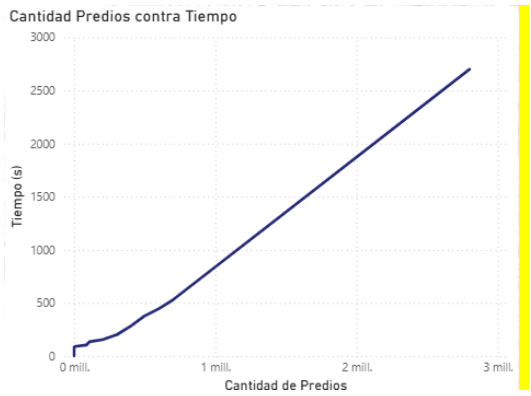
3.16. Versión 0.5

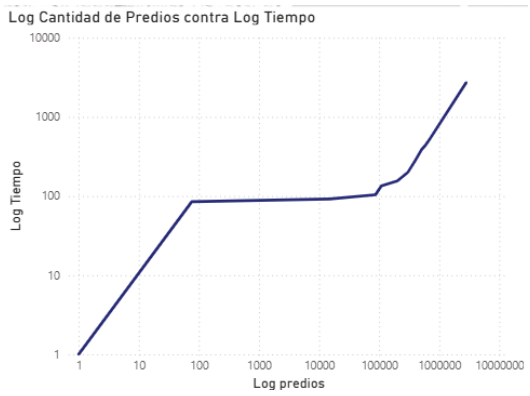
En la versión 0.5 se abordaron ajustes derivados tanto de los hallazgos técnicos identificados por el ingeniero de datos en el entorno de pruebas como de las observaciones funcionales realizadas por la directora en la última reunión de seguimiento. Estos ajustes se concentraron, por un lado, en corregir y afinar ciertas interpretaciones de la IA sobre los datos del cubo OLAP geoespacial, revisando reglas, sinónimos y patrones de consulta para que las sentencias SQL generadas se ajustaran mejor a la lógica predial y normativa definida por la entidad. Por otro lado, se implementó la limitación explícita del número de predios retornados por consulta, atendiendo la petición de los usuarios de evitar respuestas excesivamente voluminosas que afectaran el desempeño y la legibilidad de los resultados. Con esta iteración, el Localizador Predial FEN avanzó hacia un comportamiento más controlado y alineado con las expectativas institucionales, manteniendo el enfoque en la calidad de las respuestas y en la experiencia de uso en contextos ejecutivos.

3.17. Testeo 0.5

En la versión 0.5 el testeo se orientó a verificar que los ajustes introducidos se comportaran de forma estable tanto en el entorno desplegado como en el entorno local de desarrollo. En el servidor de pruebas se realizaron verificaciones manuales de conexión entre la base de datos y el backend, midiendo de manera empírica los tiempos de respuesta de las consultas ejecutadas directamente sobre la API para confirmar que no hubiera degradaciones en latencia ni errores de comunicación. Paralelamente, en ambiente local se probaron distintos escenarios de consulta en lenguaje natural, comprobando que la nueva limitación en el número de predios retornados por búsqueda se aplicara correctamente (evitando respuestas excesivamente voluminosas) y que las interpretaciones de la IA sobre ciertos campos y criterios del cubo reflejaran de forma más fiel la lógica esperada por la entidad. En conjunto, estas pruebas permitieron validar que los cambios de la 0.5 mejoran el control sobre el volumen de resultados y afinan

el comportamiento del módulo NL→SQL sin introducir incidencias adicionales en la integración entre cubo, API e interfaz.





3.18. Incidencias 0.5

En el testeo de la versión 0.5 se identificaron varias oportunidades de mejora relacionadas con los tiempos de carga tanto en el backend como en el frontend. Se observó que, a medida que aumenta el número de predios involucrados en una consulta, el tiempo total de respuesta crece de manera significativa, llegando en algunos casos a valores elevados que afectan la fluidez percibida por el usuario. Si bien las limitaciones de predios incorporadas en esta versión ayudan a contener parcialmente el problema, estos hallazgos ponen de manifiesto la necesidad de seguir optimizando las consultas al cubo, los mecanismos de paginación y la gestión de datos en la interfaz para garantizar un desempeño consistente incluso en escenarios de alta demanda.

3.19. Versión 0.6

La versión 0.6 se enfocó en mejorar el desempeño percibido por el usuario y en afinar la consistencia del cubo OLAP geoespacial. En la capa de presentación se reestructuró el flujo de carga de resultados para que la tabla y el mapa se procesen de forma paralela, permitiendo que la tabla muestre los predios encontrados sin tener que esperar a que finalice completamente la carga geoespacial del mapa. Este cambio reduce el tiempo de espera

visible y hace que la interacción inicial con los resultados sea más ágil en consultas con muchos registros.

En paralelo, se corrigieron incidencias detectadas en el cubo relacionadas con las Zonas Homogéneas Físicas (ZHF), ajustando etiquetas donde era necesario para asegurar que las consultas y los filtros sobre estas variables devolvieran información coherente y alineada con la definición institucional. Con estos ajustes, la versión 0.6 fortalece tanto la experiencia de uso como la calidad semántica de los datos expuestos por el Localizador Predial FEN.

3.20. Testeo 0.6

En la versión 0.6 el testeo se enfocó en analizar el comportamiento del sistema desde la perspectiva de consumo de memoria RAM. Se realizaron pruebas QA manuales sobre distintos escenarios de consulta, midiendo de forma separada el uso de memoria del backend (API y base de datos) y del frontend (aplicación React con mapa Leaflet) mientras se incrementaba la cantidad de predios cargados. Con estos datos se ajustó un modelo de regresión lineal que permitió proyectar el consumo esperado para volúmenes cercanos al total de predios.

Los resultados mostraron que, si se desea mostrar simultáneamente todos los predios en una sola consulta, el sistema requiere al menos 32 GB de RAM para funcionar de manera estable en condiciones controladas. No obstante, al considerar un entorno real con múltiples consultas concurrentes y otros procesos ejecutándose en el servidor, el análisis sugiere como recomendación una configuración de 64 GB de RAM, con el fin de contar con un margen de seguridad adecuado y reducir el riesgo de degradación en los tiempos de respuesta bajo carga elevada.

3.21. Incidencias 0.6

En la versión 0.6 las oportunidades de mejora se orientan principalmente a la forma de uso del aplicativo más que a fallos puntuales. A partir de los análisis de desempeño y consumo de recursos, se identificó que no resulta sostenible ni necesario mostrar todos los predios de la ciudad de manera simultánea en

una única consulta, por lo que se perfila como alternativa más adecuada un enfoque similar al de visores institucionales como Mapas Bogotá, donde la visualización y las consultas se acotan por niveles de zoom. Esta estrategia podría reducir de forma importante la carga sobre el servidor y mejorar la experiencia de uso; sin embargo, su implementación requiere una evaluación posterior más detallada, tanto desde el punto de vista técnico como funcional, antes de incorporarse a una nueva versión del Localizador Predial FEN.

4. Operacionalizar - Industrializar

En esta sección "Operacionalizar" escribir todo lo relacionado con la verificación del código, la aplicación, verificación de fórmulas, es decir, su validez, y que de paso para la implementación.

4.1. Transferencia

La transferencia de la solución al cliente se realizó mediante la entrega del paquete tecnológico completo al responsable del servidor de IDECA. Se envió el código fuente del frontend (aplicación web en React/Vite), del backend (API en FastAPI) y la configuración necesaria para el consumo del modelo de IA (qwen3-coder:480b-cloud a través del proxy de Vite), junto con el volcado de base de datos cuboolap.sql, que contiene el esquema analítico, la vista mv_predio_flat y los datos necesarios para las consultas.

Junto a estos artefactos se entregó un archivo README.md que actúa como guía técnica de instalación y ejecución en una máquina local o servidor de pruebas, allí se describen los requisitos de software (Python, Node.js, PostgreSQL, Ollama/servicio de IA), los pasos para restaurar la base de datos, configurar la conexión del backend y levantar tanto la API como el frontend. Desde el punto de vista del usuario final, el propio aplicativo incluye una Landing Page que funciona como manual de usuario básico, donde se explica el paso a paso para usar el Localizador Predial FEN y se listan las variables iniciales sobre las que se puede consultar.

Como propuesta inicial de mantenimiento, la solución queda preparada para que el cliente pueda: (i) actualizar la base de datos restaurando una nueva versión de cuboolap.sql cuando cambien los datos fuente, (ii) actualizar dependencias del frontend y backend mediante los gestores estándar (npm/pip), y (iii) sustituir o ajustar el modelo de IA sin modificar la lógica central de la API, aprovechando que el consumo del modelo está desacoplado y configurado vía proxy. Esto facilita la continuidad de la solución en ambiente de producción y abre la puerta a futuras mejoras sin reescribir el núcleo del aplicativo.

4.2. Retrospectiva

A lo largo del diseño e implementación del Localizador Predial FEN se consolidaron varias lecciones aprendidas que retroalimentan tanto la metodología de trabajo como la propia arquitectura de la solución. En primer lugar, se confirmó la importancia de partir de un modelo de datos sólido (cubo OLAP geoespacial con dominios normalizados y vista única de consulta) antes de invertir en funcionalidades avanzadas: muchas mejoras posteriores (ZHF, proximidad, exportaciones consistentes) fueron posibles precisamente porque existía una base analítica clara. También quedó evidente que el uso de modelos de lenguaje para NL→SQL requiere iteración y experimentación ya que la primera elección (Gemma3:4b local) fue útil como prueba de concepto, pero no cumplió los requisitos de desempeño, lo que llevó a explorar alternativas en la nube (qwen3-coder:480b-cloud) y a refinar los prompts y reglas de salida hasta alcanzar tiempos de respuesta aceptables.

En segundo lugar, la evolución por versiones mostró el valor de introducir mejoras graduales en la experiencia de usuario como lo son la sincronización tabla–mapa, exportaciones estables, sugerencias en el chat y la Landing Page surgieron como respuestas concretas a problemas detectados y a la retroalimentación de la directora, demostrando que la usabilidad y la documentación son tan críticas como el funcionamiento interno del sistema. Finalmente, las incidencias con el rendimiento de las consultas espaciales de proximidad subrayan la necesidad de considerar desde el inicio los requisitos no funcionales (tiempos de respuesta, carga sobre la base de datos, índices geoespaciales) cuando se diseñan casos de uso avanzados. Estas lecciones

sirven como guía para futuros proyectos: iterar rápido con prototipos funcionales, validar temprano con usuarios reales, desacoplar los componentes clave (datos, API, IA, interfaz) y planear desde el comienzo estrategias de optimización y mantenimiento que permitan escalar la solución y adaptarla a nuevos casos de uso.

4.3. Finalización

Al cierre del presente trabajo, el caso de uso Localizador Predial FEN se encuentra desarrollado en su versión de producto mínimo viable (MVP) y completamente documentado en cuanto a modelo de datos, arquitectura, evolución por versiones y uso de la herramienta. La solución fue transferida al ingeniero de datos responsable del servidor de IDECA y desplegada en un ambiente de pruebas accesible a través de la URL institucional configurada para tal fin (https://analitica.ideca.gov.co/IDECA/localizador), como paso previo a su puesta en producción.

No obstante, al momento de la finalización del documento, el entorno presenta una falla asociada al componente de IA (servicio NL→SQL), lo que impide el funcionamiento completo del flujo de consulta en lenguaje natural. La capa de datos, la API y la interfaz web se encuentran operativas bajo condiciones controladas, pero la estabilización definitiva del módulo de IA y su operación continua en el servidor quedan como actividades pendientes para culminar la fase de industrialización y habilitar el uso pleno del Localizador Predial FEN en contexto directivo.

5. Resultados

Al finalizar el desarrollo del Localizador Predial FEN se obtuvo una solución analítica integral que materializa los objetivos planteados, contar con un punto unificado en el que los directivos puedan formular preguntas en lenguaje natural sobre el territorio y recibir una respuesta estructurada en tabla y mapa. En términos técnicos, se construyó y documentó un cubo OLAP geoespacial en PostgreSQL/PostGIS con vista de consulta única, dominios normalizados y la incorporación de capas clave como Catastro, Norma Urbana, Ofertas y Zonas Homogéneas Físicas urbanas/expansión y rurales con sus etiquetas. Sobre este núcleo se dispuso una API en FastAPI que expone servicios de consulta, exportación y soporte a

funcionalidades avanzadas (GeoJSON, sugerencias, conteo y paginado estable), y un módulo NL→SQL basado en IA que, en su versión final, utiliza el modelo qwen3-coder:480b-cloud, reduciendo los tiempos de respuesta desde minutos en las primeras pruebas a rangos de aproximadamente 5–10 segundos dependiendo de la complejidad de la consulta.

En la capa de presentación se consolidó una interfaz web en React con mapa Leaflet que permite ejecutar el ciclo completo "pregunta en lenguaje natural → SQL → tabla + mapa", integrando selección bidireccional tabla–mapa, exportación a CSV/Excel, visualización diferenciada de lotes y construcciones, y consultas de proximidad del tipo "cerca de" aprovechando las capacidades geoespaciales del cubo. Adicionalmente, se diseñó una Landing Page que explica el paso a paso de uso y expone el conjunto inicial de variables consultables, alineando la herramienta con las necesidades reales de los directivos de la UAECD y facilitando su apropiación.

Más allá de los productos concretos (base de datos, API, interfaz e integración con IA), el proyecto aporta a la disciplina de la analítica geoespacial y a la gestión catastral una ruta replicable para combinar cubos OLAP, servicios web y modelos de lenguaje en soluciones de consulta ejecutiva. La experiencia acumulada en iterar sobre modelos de IA, ajustar prompts, controlar la generación de SQL y optimizar consultas espaciales deja insumos valiosos para futuros desarrollos que busquen habilitar búsquedas en lenguaje natural sobre información territorial compleja, tanto dentro de la UAECD como en otros contextos del sector público.

6. Conclusiones

El desarrollo del aplicativo web permitió comprobar que la combinación entre un cubo OLAP geoespacial, una API de servicios y una interfaz de lenguaje natural a SQL es una ruta viable para acercar la analítica territorial avanzada a los directivos de una entidad catastral. En la práctica, se evidenció que un esquema estrella espacial con dominios normalizados y jerarquías territoriales claras, concentrado en una única vista de consulta que integra información de Catastro, Norma Urbana, Ofertas y otras tablas complementarias, acelera la exploración geoespacial y facilita el soporte a decisiones sobre el territorio.

Por otra parte, la experiencia con Text-to-SQL confirma y aterriza en un contexto institucional muchas de las conclusiones reportadas en la literatura reciente, la

traducción de lenguaje natural a SQL es viable, pero exige adaptar prompts, plantillas y conocimiento del esquema al dominio concreto. La trayectoria desde Gemma3:4b local hasta qwen3-coder:480b-cloud mostró que no basta con "tener un LLM", sino que es necesario controlar tiempos de respuesta, restringir la salida a SQL seguro para soportar consultas típicas del análisis territorial. El resultado final es un flujo "pregunta en español → SQL → tabla + mapa" que, si bien no está exento de desafíos de desempeño en casos extremos, cumple el objetivo de reducir la barrera técnica para acceder a análisis prediales complejos.

A nivel de producto, se lograron resultados materiales concretos como lo son, un cubo OLAP geoespacial documentado, una API en FastAPI que expone servicios de consulta y exportación sobre dicho cubo, una SPA en React con mapa Leaflet que integra selección tabla–mapa, exportaciones, ayudas de uso y consultas de proximidad, y una Landing Page que explica el funcionamiento y las variables consultables. Sin embargo, el aporte del proyecto va más allá de estos artefactos, pues deja una metodología replicable para otros escenarios de gestión territorial en los que se quiera aprovechar SOLAP y NL→SQL sobre bases espaciales, y evidencia la necesidad de tratar la IA no como una tecnología desconocida, sino como un componente más dentro de una arquitectura de datos bien diseñada.

Finalmente, el proceso dejó hallazgos relevantes para futuras iniciativas. Entre los cuales se destacan, (i) el papel central de la experiencia de usuario y la documentación para la apropiación de soluciones analíticas por parte de directivos; y (ii) la conveniencia de desacoplar datos, API, IA e interfaz para poder evolucionar cada capa sin rehacer el sistema. Estas lecciones alimentan no solo la disciplina de la analítica geoespacial aplicada a la UAECD, sino también futuros proyectos que busquen integrar cubos espaciales, servicios web y modelos de lenguaje en el sector público.

7. Bibliografía

1. RIVEST, Sonia; BÉDARD, Yvan; PROULX, Marie-Josée; NADEAU, Martin; HUBERT, Frédéric; PASTOR, Julien. SOLAP technology: Merging business intelligence with geospatial technology for interactive spatio-temporal exploration and analysis of data. ISPRS Journal of Photogrammetry and Remote Sensing, 2005, vol. 60, n. 1, p. 17-33. DOI: 10.1016/j.isprsjprs.2005.10.002. Disponible en: https://www.sciencedirect.com/science/article/abs/pii/S0924271605000614

2. BIMONTE, Sandro; BERTOLOTTO, Michela; GENSEL, Jérôme; BOUSSAID, Omar. Spatial OLAP and Map Generalization: Model and Algebra. International Journal of

Data Warehousing and Mining, 2012, vol. 8, n. 1, p. 24-51. DOI: 10.4018/jdwm.2012010102. Disponible en: https://dl.acm.org/doi/abs/10.4018/jdwm.2012010102

3. KYUNG, Min-Ju; YOM, Jae-Hong; KIM, Seung-Yong. Spatial data warehouse design and spatial OLAP implementation for decision making of geospatial data update. KSCE Journal of Civil Engineering, 2012, vol. 16, p. 1023-1031. Disponible en: https://link.springer.com/article/10.1007/s12205-012-1410-2

4. YU, Tao; ZHANG, Rui; YANG, Kai; YASUNAGA, Michihiro; WANG, Dongxu; LI, Zifan; MA, James; LI, Irene; YAO, Qingning; ROMAN, Shanelle; ZHANG, Zilin; RADEV, Dragomir. Spider: A Large-Scale Human-Labeled Dataset for Complex and Cross-Domain Semantic Parsing and Text-to-SQL Task. Proceedings of EMNLP 2018, p. 3911-3921. DOI: 10.18653/v1/D18-1425. Disponible en: https://aclanthology.org/D18-1425/

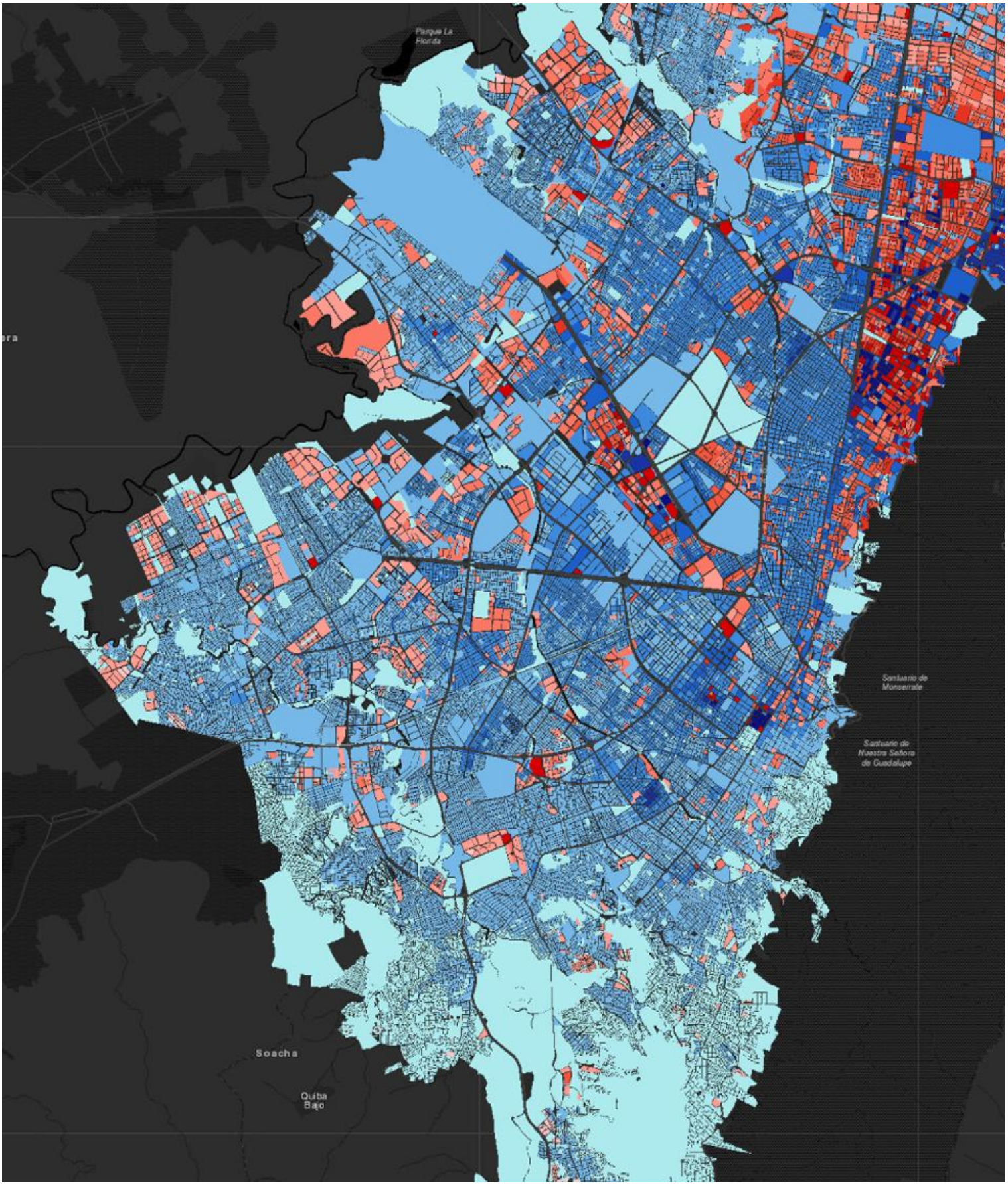
5. SUN, Ruoxi; ARIK, Sercan Ö.; MUZIO, Alex; MICULICICH, Lesly; GUNDABATHULA, Satya; YIN, Pengcheng; DAI, Hanjun; NAKHOST, Hootan; SINHA, Rajarishi; WANG, Zifeng; PFISTER, Tomas. SQL-PaLM: Improved Large Language Model Adaptation for Text-to-SQL (extended). arXiv, 2023. Disponible en: https://arxiv.org/pdf/2306.00739

6. NASCIMENTO, Eduardo R.; GARCÍA, Grettel M.; FEIJÓ, Lucas; VICTORIO, Wendy Z.; IZQUIERDO, Yenier T.; OLIVEIRA, Aiko R. de; COELHO, Gustavo M. C.; LEMOS, Melissa; GARCIA, Robinson L. S.; LEME, Luiz A. P.; CASANOVA, Marco A. Text-to-SQL Meets the Real-World. En: Proceedings of the 26th International Conference on Enterprise Information Systems (ICEIS 2024), 2024, p. 61-72. DOI: 10.5220/125552003006390. Disponible en: https://www.scitepress.org/Papers/2024/125552/125552.pdf

7. WANG, Hui; GUO, Li; LIANG, Yubin; LIU, Le; HUANG, Jiajin. GPT-Based Text-to-SQL for Spatial Databases. ISPRS International Journal of Geo-Information, 2025, 14(8):288. DOI: 10.3390/ijgi14080288. Disponible en: https://www.mdpi.com/2220-9964/14/8/288

8. LIU, Mengyi; WANG, Xieyang; XU, Jianqiu; LU, Hua; TONG, Yongxin. NALSpatial: A Natural Language Interface for Spatial Databases. IEEE Transactions on Knowledge and Data Engineering, 2025, 37(4): 2056–2070. Disponible en: https://dl.acm.org/doi/10.1109/TKDE.2025.3525587

9. YU, C.; (y col.) Monkuu: a LLM-powered natural language interface for geospatial databases with dynamic schema mapping. International Journal of Geographical Information Science, 2025. Disponible en: https://www.tandfonline.com/doi/full/10.1080/13658816.2025.2533322



ideca

ALCALDÍA MAYOR DE BOGOTÁ D.C.

ideca