Fecha: Febrero 2026
Nombres y apellidos: Silky Féliz, Esteban Carmiol ,
Gabriela Amador, Jose Guerra C, Endry González
Asignatura: Sistema de de Gestión de Bases de Datos (SGBD)
Profesor/a: Sergio García y Juan Vidal
Máster: Máster Big Dta & Business Intelligence

ASIGNATURA 6: SISTEMAS DE GESTIÓN DE BASES DE DATS (SGBD)

1. Introducción

Previo al diseño y la caracterización de los datasets, se realizó un proceso de depuración y transformación estratégica sobre los conjuntos de datasets originales. Este procedimiento no consistió únicamente en una limpieza técnica, sino que se centró, en primera instancia, en establecer la relación de los datasets mediante la vinculación de coordenadas con las zonas geográficas de análisis. Asimismo, se llevó a cabo una selección y transformación de la data orientadas específicamente al caso de estudio, descartando aquellas que no aportan valor al análisis de ubicación. Como resultado, se obtuvo un conjunto de datos optimizado bajo un identificador geográfico único (ID_ZONA), lo que permite que la estructura de la base de datos sea coherente con la arquitectura del sistema que se describe en el presente trabajo.

2. Caracterización de cada dataset

A continuación, se caracterizan los dastasets identificando su origen, estructura, volumen, patrones de uso y limitaciones.

Ficha 1. Censo

👥 **Perfil Humano**	Descripción
Origen	U.S. Census Bureau (Decennial Census 2020 / American Community Survey ACS 5-Year Estimates). Datos oficiales con metodología estadística; incluyen márgenes de error (MOE) y están sujetos a error muestral.
Estructura	Tabular- Registros sociodemográficos. Variables:id_zona,poblacion_total,edad_mediana,ingreso_mediano_hogar,tasa_empleo,tamano_hogar_promedio,poblacion_hispana,tasa_ocupacion. En el proyecto se trabaja con los datos oficiales del censo 2020.
Volumen	31 registros finales (Un perfil de indicadores por zona relevante) y 7 variables.
Limitaciones	1-Al ser encuesta, existe muestreo y MOE; inferencias finas pueden ser inestables en zonas pequeñas.
Patrón de uso	1-Perfil socioeconómico por NTA (selección de zonas objetivo): seleccionar NTAs con características compatibles con el buyer persona (p. ej., ingreso alto, densidad, renta, etc.). 2-Realizar un filtrado relacional 1:1 para descartar zonas que no cumplan con el umbral de ingreso_medio_hogar.

Ficha 2. Movilidad

🚶 **Flujo Urbano**	Descripción
Origen	Metropolitan Transportation Authority (NYC Open Data).
Estructura	Eventos-Registros horarios por estación y línea de metro. Variables:: id_zona,tipo_dia,volumen_total_pasajeros,cantidad_estaciones. En el proyecto se trabaja por descarga con corte Oct-2024.
Volumen	Registros tras limpieza: 62 registros finales (Promedios por zona y tipo de día) y 4 variables.

Limitaciones	1-El ridership aproxima flujo de paso, pero no garantiza conversión a clientes. 2-La movilidad puede estar afectada por turismo, obras o incidencias; considerar estacionalidad.
Patrón de uso	1-Identificación de zonas de alto tránsito peatonal: Top estaciones, filtrar por franja comercial y sumar ridership por estación; luego asignar a NTA para scoring territorial. 2- Una vez asignada la estación a NTA, calcular perfil horario para comparar zonas.

Ficha 3. Restaurantes

🍽 **Mapa Rival**	Descripción
Origen	NYC Open Data – Department of Health and Mental Hygiene (DOHMH). Dataset: "DOHMH New York City Restaurant Inspection Results"
Estructura	Eventos- Inspecciones individuales por local. Variables: id_zona, tipo_cocina, clasificacion_competencia, cantidad_locales. En el proyecto se trabaja con datos actualizados al 2025.
Volumen	Registros tras limpieza: 618 (agrupados por zona y categoría) y 4 variables.
Limitaciones	1.Un restaurante puede estar activo, pero con inspección antigua; definir "recencia" mínima. 2. Múltiples violaciones por inspección inflan el número de filas; se debe agregar por ID Restaurante/inspección.
Patrón de uso	1-Competencia directa (Mexican/Hispanic) con última inspección: filtrar por cocina para "rivales directos". 2-Densidad de competencia por NTA: tras asignar nta_code (join espacial), contar locales por zona y comparar con demanda.

Ficha 4. Seguridad

🛡 **Zona Segura**	Descripción
Origen	NYPD Complaint Data (NYC Open Data). Datos públicos oficiales del Departamento de Policía de Nueva York.

Estructura	Eventos - Registros históricos de incidentes delictivos, niveles de gravedad y lugares de mayor incidencia. Variables: id_zona, tipo_delito, lugar_delito, cantidad_incidentes. En el proyecto se toman los reportes del año 2025.
Volumen	Registros tras limpieza: 527 registros (agrupados por Zona, Delito y Lugar) y 4 variables.
Limitaciones	1.Algunos delitos se geocodifican intencionalmente en ubicaciones protegidas (p. ej., comisaría) para proteger identidades. 2. Subreporte (propensión a denunciar)
Patrón de uso	1-Tasa de incidentes por NTA (delitos de alto impacto): filtrar categorías seleccionadas y contar incidentes por NTA. 2- Calcular la densidad de delitos por id_zona filtrando por delito (Felonies) y categorizando por lugar_delito (Comercial vs Vía Pública).

Ficha 5. Lugares de interés

📍 **Atractores**	Descripción
Origen	NYC Map of Common Places (NYC Planning). Datos oficiales de infraestructura y puntos estratégicos de la ciudad.
Estructura	Tabular y Geométrica- Ubicación exacta de puntos de interés como parques, escuelas y centros culturales. Variables:id_zona, categoria_lugar, cantidad_lugares En el proyecto se toman datos de infraestructura urbana actualizados al 2025.
Volumen	Registros tras limpieza: 232 registros finales (consolidados y corregidos por zona) y 3 variables.
Limitaciones	1-No todos los puntos de interés generan la misma atracción; se recomienda ponderar por tipo y, cuando exista, capacidad. 2- Algunos registros pueden representar sedes administrativas.
Patrón de uso	1-Filtrar los registros por el atributo categoría (por ejemplo, "Educación" o "Recreación") para cuantificar la presencia de imanes de público específicos en cada id_zona. 2-Utilizar la variable cantidad_lugares para asignar un peso numérico a cada zona. Las zonas con mayores valores en categorías clave para el *buyer persona* (como centros culturales o parques) recibirán una puntuación superior en el modelo de *scoring* final.

Ficha 6. Costos de alquiler

💰 **Viabilidad**	Descripción
Origen	Tabular- Recopilación multifuente de portales de Real Estate Comercial (LoopNet, CommercialCafe, StreetEasy).
Estructura	Registros sobre costos de alquiler. Variables: id_zona, precio_pies_cuadrados_anual
Volumen	38 registros y 2 variables. Datos actualizados y validados al 2025
Limitaciones	1-Datos no estandarizados entre plataformas 2-Sesgo por disponibilidad del inventario: un barrio con pocos anuncios puede parecer más barato o más caro artificialmente.
Patrón de uso	1-Filtro económico inicial ("knock-out"): el alquiler es un criterio clave para descartar zonas inviables 2-Penalización en el scoring: pesos más altos en escenarios conservadores donde se prioriza sostenibilidad financiera.

Ficha 7. Zonas

📍 Unidad Geográfica	Descripción
Origen	NYC Department of City Planning (DCP) - *Neighborhood Tabulation Areas (NTA)*.
Estructura	Documental geográfico – Datos geoespaciales. Variables: id_zona, nombre_zona, nombre_distrito Clave de integración: nta_code (o NTA2020) utilizado como llave común en el modelo de datos del proyecto.
Volumen	38 registros (Las 38 zonas oficiales que componen Manhattan) y 3 variables En el proyecto se toman datos estables del 2025.
Limitaciones	1-NTA no equivale exactamente a "barrios" percibidos; es una geografía estadística.
Tratamiento	Filtrado geográfico: Selección exclusiva de los registros pertenecientes a Manhattan. Verificación de codificación: Validación de que los códigos de zona coinciden exactamente con la codificación de la data del Censo.
Patrón de uso	1-Asignación de nta_code a puntos (join espacial): enriquecer MTA, DOHMH, NYPD y Facilities agregando nta_code para agregaciones por zona. 2-Construir una tabla por nta_code con métricas normalizadas para scoring.

3. Enfoque de modelado

El diseño lógico de este proyecto se fundamenta en un esquema de estrella orientado al análisis geoespacial de Manhattan. El enfoque del modelo sitúa una entidad maestra central ZONAS como la unidad principal de análisis y el núcleo del sistema, ya que todo el proyecto se estructura alrededor de esta demarcación geográfica.

Para lograr esta integración, cada uno de los datasets que conforman las diferentes entidades del modelo fueron procesados y transformados de sistema de coordenadas a la ubicación geográfica a nivel de Zona a la que pertenecen. Teniendo cada una de estas entidades una relación directa y/o espacial con la entidad maestra ZONAS, la relación que existe entre esta entidad maestra y las demás es un ID_ZONA único para cada demarcación.

Modelo Entidad - Relación TFM Sistema de Site Selection

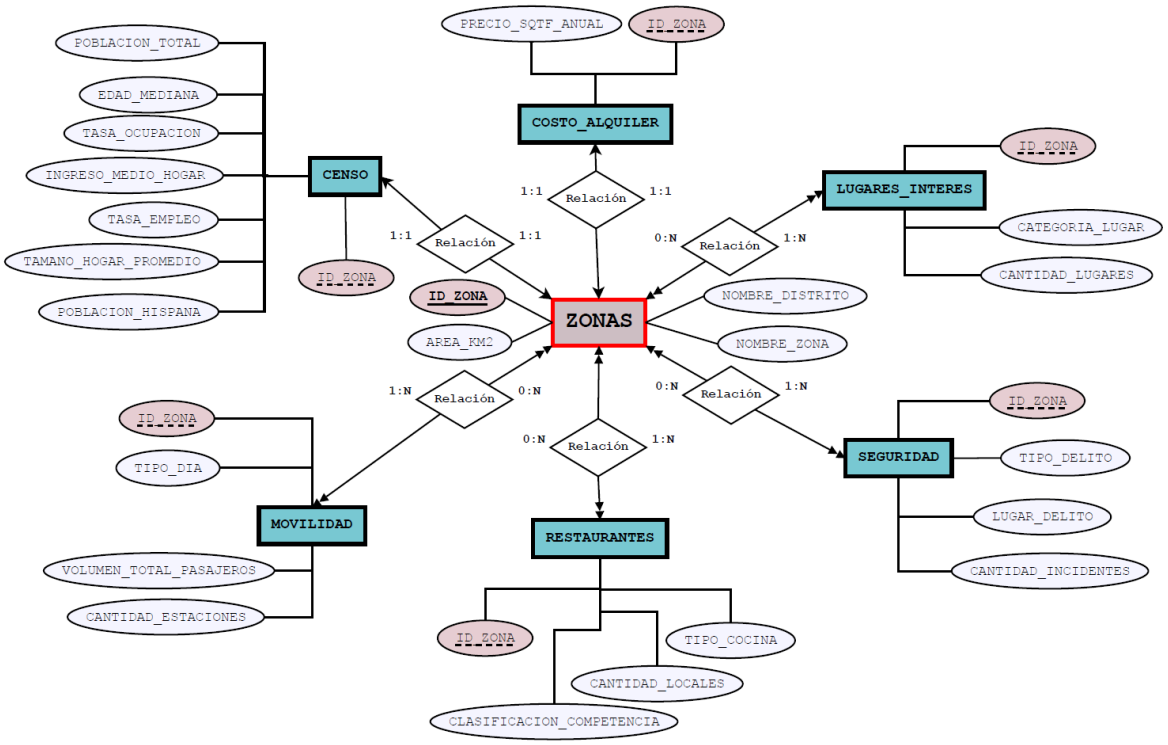


Ilustración 1 Diagrama del Modelo Entidad – Relación

En resumen, la arquitectura del modelo de datos está definida por:

- **Entidad central (Tabla ZONAS):** Funciona como el núcleo integrador del modelo. Contiene la Clave Primaria (ID_ZONA) que vincula todo el ecosistema de datos, además de atributos descriptivos esenciales como el NOMBRE_ZONA.

- **Entidades con relación directa (1:1):** Las tablas CENSO y COSTO_ALQUILER presentan una relación directa y única con la entidad central mediante el ID_ZONA. Esto significa que a cada zona le corresponde un único registro que consolida variables importantes seleccionadas como población, niveles de ingresos, empleo y el costo de alquiler anual.

- **Entidades con relación espacial y agregada (1:N):** Las entidades SEGURIDAD, MOVILIDAD, RESTAURANTES y LUGARES_INTERES se vinculan mediante el mismo ID_ZONA, actuando este como clave foránea o débil.

    - **Procesamiento espacial**: Originalmente, estos datos se encontraban georreferenciados como puntos; mediante un análisis espacial, se les asignó su zona correspondiente para permitir la unión con la entidad maestra o central.

    - **Lógica de agrupación**: Estas entidades no almacenan hechos aislados, sino métricas consolidadas por categorías. Por ejemplo, se agrupan por tipos de delito, clasificación de competencia o categorías de lugares de interés, lo que permite un análisis eficiente y orientado al objetivo del proyecto.

    - **Cardinalidad (1:N)**: La cardinalidad con estas entidades es de uno a muchos.

      Dado que durante el procesamiento de los datos al asignar la zona correspondiente se garantizó que cada registro de estas tablas contenga al menos una zona válida, asegurando la integridad referencial. Sin embargo, la cardinalidad inversa, es decir, desde la tabla maestra a las demás entidades, la relación es (0:N); esto con fines de escalabilidad del sistema, donde una zona puede que no tenga ningún valor registrado en los atributos de las demás dimensiones

Justificación de la selección del modelo

La elección de un esquema en estrella para el proyecto responde a una decisión de diseño estratégico basado en:

1. **Centralidad y escalabilidad:** Como el objetivo del proyecto es analizar las demarcaciones en Manhattan, el modelo estrella permite que la entidad creada denominada ZONAS actúe como el único punto de unión. De esta manera, se facilita que cualquier nueva variable que no se haya tomado en cuenta en la selección se pueda añadir al modelo sin alterar las relaciones existentes, permitiendo la escalabilidad del sistema.

2. **Eficiencia en consultas:** A diferencia de un modelo altamente normalizado como el esquema copo de nieve, el esquema en estrella reduce la complejidad de las consultas SQL. Esto agiliza el proceso de filtrado de zonas y minimiza la cantidad de relaciones necesarias para obtener los datos.

3. **Integración de datos heterogéneos:** El modelo nos permite integrar datos que vienen en formatos muy distintos; en el caso del proyecto, logramos unificar datos tabulares (Censo y Alquiler), georreferenciados (Puntos espaciales) y categorizados (Seguridad, Movilidad, etc.).

4. **Robustez del análisis:** Por último, la decisión de establecer la cardinalidad (0:N) desde la tabla maestra a las tablas en las dimensiones responde a una realidad de la unidad de análisis. Esto permite una arquitectura más flexible donde el sistema no se ve limitado por la falta de registros, permitiendo comparar los demás atributos en dado caso que una zona no posea información en alguna categoría específica.

A continuación, se presenta el diccionario de datos que muestra la arquitectura relacional del proyecto, con las especificaciones técnicas de cada entidad (atributos, tipo de dato, nualidad) y su función dentro del modelo.

Entidad	Atributo	Tipo de Dato	Descripción
ZONAS	ID_ZONA	VARCHAR (10) (PK)	**Clave Primaria**. Identificador único de la zona.
	NOMBRE_ ZONA	VARCHAR (60)	Nombre oficial de la zona
	NOMBRE_ DISTRITO	VARCHAR (30)	Nombre del distrito de la zona
	AREA_KM2	DECIMAL (10,4)	Superficie de la zona en km^2
CENSO	ID_ZONA	VARCHAR (10) (FK)	Clave Foránea. Vínculo con tabla ZONAS.
	POBLACION_ TOTAL	INTEGER	Habitantes totales de la zona.

Entidad	Atributo	Tipo de Dato	Descripción
	EDAD_MEDIANA	DECIMAL (5,2)	Edad mediana de la población.
	TASA_OCUPACION	DECIMAL (5,2)	Porcentaje de población ocupada.
	INGRESO_MEDIO_HOGAR	DECIMAL (12,2)	Ingreso promedio por hogar.
	TASA_EMPLEO	DECIMAL (5,2)	Tasa de empleo sobre población total.
	TAMANO_HOGAR_PROMEDIO	DECIMAL(4,2)	Promedio de personas por vivienda.
	POBLACION_HISPANA	INTEGER	Total de residentes de origen hispano.
COSTO_ALQUILER	ID_ZONA	VARCHAR (10) (FK)	Clave Foránea. Vínculo con tabla ZONAS.
	PRECIO_SQTF_ANUAL	DECIMAL (10,2)	Costo anual por pie cuadrado.
MOVILIDAD	ID_ZONA	VARCHAR (10) (FK)	Clave Foránea. Vínculo con tabla ZONAS.
	TIPO_DIA	VARCHAR (15)	Categoría: Laboral o Fin de semana.
	VOLUMEN_TOTAL_PASAJEROS	INTEGER	Flujo promedio diario de pasajeros.
	CANTIDAD_ESTACIONES	INTEGER	Total de estaciones de transporte.
SEGURIDAD	ID_ZONA	VARCHAR (10) (FK)	Clave Foránea. Vínculo con tabla ZONAS.
	TIPO_DELITO	VARCHAR (30)	Categoría del incidente de seguridad.
	LUGAR_DELITO	VARCHAR (20)	Tipo de ubicación del suceso.
	CANTIDAD_INCIDENTES	INTEGER	Total de incidentes registrados.
RESTAURANTES	ID_ZONA	VARCHAR (10) (FK)	Clave Foránea. Vínculo con tabla ZONAS.
	TIPO_COCINA	VARCHAR (20)	Especialidad gastronómica.
	CANTIDAD_LOCALES	INTEGER	Conteo de establecimientos del rubro.
	CLASIFICACION_COMPETENCIA	VARCHAR (20)	Nivel de rivalidad (Directa/Indirecta).
LUGARES_INTERES	ID_ZONA	VARCHAR (10) (FK)	Clave Foránea. Vínculo con tabla ZONAS.
	CATEGORIA_LUGAR	VARCHAR (30)	Clasificación del punto de interés.
	CANTIDAD_LUGARES	INTEGER	Conteo de sitios por categoría.

Debido al proceso previo de limpieza, agrupación y caracterización de las variables de las diferentes entidades, todos los atributos en el diccionario han sido establecidos bajo la restricción de no nulidad (NOT NULL). Por lo tanto, cada registro aporta información completa y válida para el análisis, garantizando que no existan valores incompletos y asegurando así la integridad del modelo y la eficiencia del sistema.

4. Justificación de la elección tecnológica

La selección del Sistema de Gestión de Bases de Datos (SGBD) constituye una decisión estructural dentro del TFM, ya que define el entorno donde se consolidará, integrará y explotará la información proveniente de múltiples fuentes heterogéneas. Dado que el modelo conceptual del proyecto se fundamenta en un esquema entidad–relación claramente definida —con ZONAS como entidad central y múltiples relaciones 1:1 y 0:N con CENSO, MOVILIDAD, RESTAURANTES, SEGURIDAD, LUGARES_INTERÉS y COSTO_ALQUILER— la tecnología seleccionada debe garantizar consistencia estructural, integridad referencial y eficiencia en consultas relacionales complejas.

En este contexto, se selecciona MySQL como Sistema de Gestión de Bases de Datos relacional, por su alineación técnica con el modelo lógico del proyecto, su robustez transaccional, su compatibilidad con herramientas analíticas y su amplia adopción en entornos académicos y profesionales.

A continuación se exponen una serie de criterios que se consideran relevantes para la selección de MySQL como SGBD del proyecto:

- **Coherencia con el modelo relacional del proyecto**

El modelo de datos del TFM responde a los principios clásicos del modelo relacional propuesto por Codd (1970)[1], donde la información se organiza en tablas interrelacionadas mediante claves primarias y foráneas. La arquitectura del sistema se basa en:

- ✔ Una entidad central (ZONAS) identificada por ID_ZONA.
- ✔ Relaciones estructuradas 1:1 (por ejemplo, CENSO – ZONAS).
- ✔ Relaciones 0:N (por ejemplo, ZONAS – RESTAURANTES, ZONAS – SEGURIDAD).
- ✔ Integración mediante joins directos y agregaciones por zona.

[1] Codd, E. F. (1970). A relational model of data for large shared data banks. *Communications of the ACM, 13*(6), 377–387. https://dl.acm.org/doi/10.1145/362384.362685

El modelo relacional continúa siendo el estándar dominante para sistemas que requieren integridad estructural y consistencia transaccional (Elmasri & Navathe, 2016)[2]. MySQL implementa plenamente este paradigma, ofreciendo soporte adecuado para:

- ✔ Definición de claves primarias y foráneas.
- ✔ Restricciones de integridad.
- ✔ Normalización de esquemas.
- ✔ Consultas SQL optimizadas.

En consecuencia, la elección de MySQL garantiza coherencia conceptual entre el modelo entidad–relación diseñado y la infraestructura tecnológica que lo soporta.

- **Garantía de integridad y propiedades**

El proyecto requiere asegurar que la consolidación de múltiples datasets no genere inconsistencias, duplicidades ni rupturas en la integridad referencial. Para ello, es fundamental operar bajo un sistema que garantice las propiedades ACID (Atomicidad, Consistencia, Aislamiento y Durabilidad), consideradas un estándar en bases de datos transaccionales (Silberschatz, Korth & Sudarshan, 2019)[3].

Mediante su motor de almacenamiento MySQL posee la capacidad de implementar estas propiedades, permitiendo:

- ✔ Transacciones seguras.
- ✔ Validación automática de claves foráneas.
- ✔ Recuperación ante fallos.
- ✔ Protección contra corrupción de datos.

Aunque el TFM no constituye un sistema transaccional intensivo, la consolidación de métricas y la construcción de la tabla maestra por Neighborhood Tabulation Area (NTA) requieren confiabilidad estructural para asegurar reproducibilidad metodológica.

---

[2] Elmasri, R., & Navathe, S. (2016). *Fundamentals of database systems* (7th ed.). Pearson. https://www.pearson.com/en-us/subject-catalog/p/fundamentals-of-database-systems/P200000003258

[3] Silberschatz, A., Korth, H. F., & Sudarshan, S. (2019). *Database system concepts* (7th ed.). McGraw-Hill. https://www.db-book.com/

- **Capacidad de consultas analíticas y agregaciones**

Los SGBD relacionales se consideran particularmente adecuados para operaciones analíticas estructuradas basadas en SQL. En este sentido la selección de MySQL tiene la capacidad de proporcionar los siguientes elementos:

- ✔ Optimización mediante índices
- ✔ Vistas y subconsultas
- ✔ Funciones de agregación
- ✔ Soporte para consultas complejas de diversas bases de datos o tablas.

- **Reproductibilidad y Estandarización**

Un criterio adicional en la selección tecnológica fue la reproducibilidad académica. MySQL es uno de los SGBD más utilizados a nivel global y cuenta con documentación extensa y soporte comunitario amplio (Oracle, 2024)[4]. La utilización de este sistema puede facilitar:

- ✔ Replicabilidad del sistema por terceros
- ✔ Implementación en entornos locales o en la nube
- ✔ Transferibilidad metodológica del modelo.

5. Descripción del MVP implementado

Debido a que la unidad de análisis de este proyecto es la demarcación geográfica, la base de datos se ha diseñado bajo un esquema de estrella donde la entidad ZONAS actúa como eje central.

Sin embargo, dado que existen entidades relacionadas con ZONAS que presentan múltiples categorías (como Movilidad, Seguridad, Restaurantes y Lugares de Interés), se ha implementado una regla de identificadores de registro (Primary Keys) compuestos para estas entidades. Esto garantiza que cada registro sea único dentro del modelo.

Tabla	Identificador de registro (PK)	Componentes de la Llave (PK)	Descripción
ZONAS	PRIMARY	ID_ZONA	Identificador de la demarcación.
CENSO	PRIMARY	ID_ZONA	Perfil socioeconómico único por zona.

[4] Oracle. (2024). *MySQL 8.0 Reference Manual*. https://dev.mysql.com/doc/refman/8.0/en/

ALQUILER	PRIMARY	ID_ZONA	Costo inmobiliario único por zona.
MOVILIDAD	PK_MOVILIDAD	ID_ZONA, TIPO_DIA	Registro único por zona según el flujo del día.
SEGURIDAD	PK_SEGURIDAD	ID_ZONA, TIPO_DELITO, LUGAR_DELITO	Incidencias agrupadas por tipo y ubicación.
RESTAURANTES	PK_RESTAURANTES	ID_ZONA, TIPO_COCINA, COMPETENCIA	Restaurantes por tipo de cocina y competencia.
LUGARES_ INTERES	PK_LUGARES_ INTERES	ID_ZONA, CATEGORIA_ LUGAR	Conteo de puntos de interés por clasificación.

Todas las tablas están vinculadas a la entidad maestra (ZONAS) mediante la clave foránea (FK) del ID_ZONA. Se ha definido una nomenclatura para las reglas de relación basada en la combinación de la tabla origen y la tabla maestra; por ejemplo: FK_MOVILIDAD_ZONAS.

En el proceso de creación de la base de datos se integra el uso de restricciones (CONSTRAINTS) como un mecanismo de control para asegurar la fiabilidad de la arquitectura. Esta función garantiza la integridad de las relaciones al vincular obligatoriamente cada dimensión con la entidad maestra mediante claves foráneas. Asimismo, su aplicación en conjunto con la restricción NOT NULL permite la consolidación de un modelo sin inconsistencias, asegurando que cada registro aporte información completa y válida para el análisis.

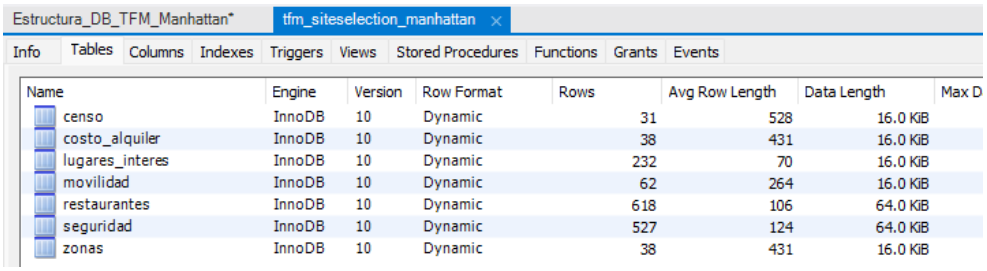
A continuación, se presenta una muestra de los comandos utilizados para la creación de la base de datos y la estructura de las tablas, basado en el diccionario de datos.

```sql
-- ESTRUCTURA DE BASE DE DATOS TFM SITE SELECTION MANHATTAN

-- 1. Creación de la base de datos del proyecto
CREATE DATABASE TFM_SiteSelection_Manhattan;
USE TFM_SiteSelection_Manhattan;

-- 2. Creación de la Entidad Maestra: ZONAS
CREATE TABLE ZONAS (
    ID_ZONA VARCHAR(10) NOT NULL,
    NOMBRE_ZONA VARCHAR(60) NOT NULL,
    NOMBRE_DISTRITO VARCHAR(30) NOT NULL,
    AREA_KM2 DECIMAL(10,4) NOT NULL,
    PRIMARY KEY (ID_ZONA)
);
```

```sql
-- 5. Creación de la Dimensión 1:N - MOVILIDAD (PK Compuesta)
-- Implementacion PK compuesta (ID_ZONA, TIPO_DIA) para garantizar registros únicos de movilidad por zona
CREATE TABLE MOVILIDAD (
    ID_ZONA VARCHAR(10) NOT NULL,
    TIPO_DIA VARCHAR(15) NOT NULL,
    VOLUMEN_TOTAL_PASAJEROS INT NOT NULL,
    CANTIDAD_ESTACIONES INT NOT NULL,
    CONSTRAINT PK_MOVILIDAD PRIMARY KEY (ID_ZONA, TIPO_DIA),
    CONSTRAINT FK_MOVILIDAD_ZONAS FOREIGN KEY (ID_ZONA) REFERENCES ZONAS(ID_ZONA)
);
```

Luego de creadas las tablas, se procedió a la importación de los datos, iniciando por la tabla maestra ZONAS. En este proceso se verificó que los atributos creados coincidieran con los de los datasets originales y que el tipo de dato asignado fuera el correcto. Por ejemplo, se utilizaron valores decimales (5,2) para atributos porcentuales y (12,2) para valores económicos.

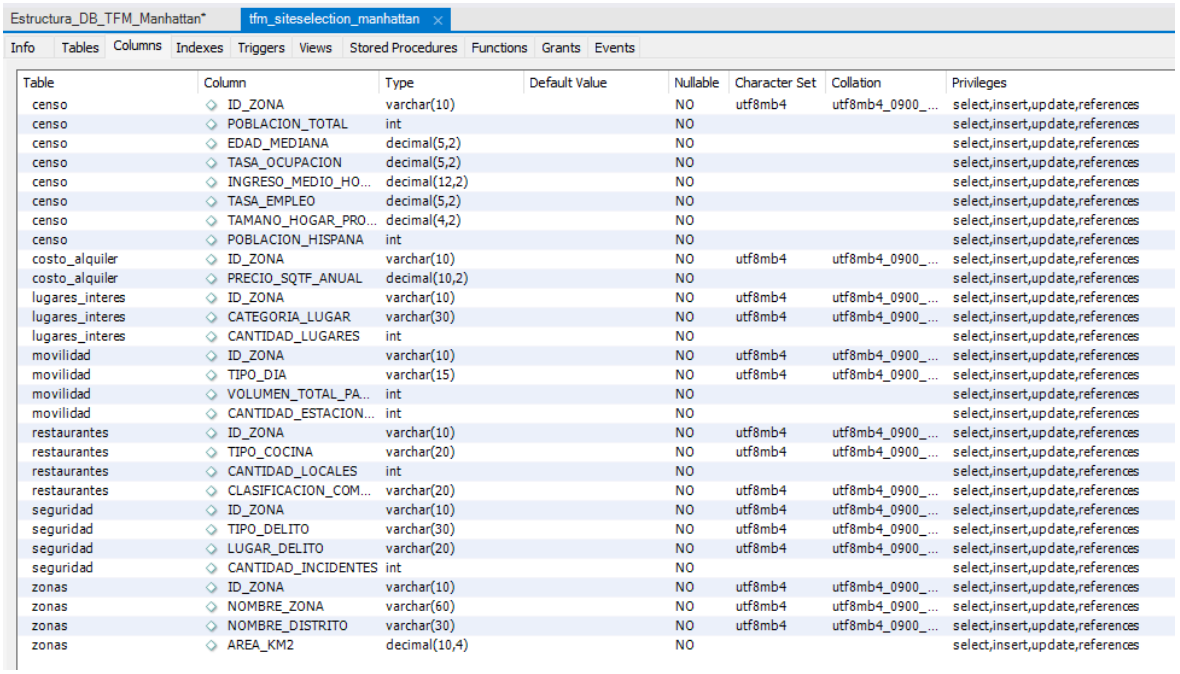
A continuación, se muestra el resultado de la implementación física en el SGBD; un resumen que evidencia el uso del motor de almacenamiento InnoDB, seleccionado por su capacidad para gestionar la integridad referencial y las restricciones del modelo. Asimismo, se detalla el volumen de datos procesado y la verificación del tamaño de las tablas, optimizado para mantener el sistema eficiente y con alto rendimiento mediante la limitación estratégica de caracteres en los atributos.



Name	Engine	Version	Row Format	Rows	Avg Row Length	Data Length	Max D
censo	InnoDB	10	Dynamic	31	528	16.0 KiB	
costo_alquiler	InnoDB	10	Dynamic	38	431	16.0 KiB	
lugares_interes	InnoDB	10	Dynamic	232	70	16.0 KiB	
movilidad	InnoDB	10	Dynamic	62	264	16.0 KiB	
restaurantes	InnoDB	10	Dynamic	618	106	64.0 KiB	
seguridad	InnoDB	10	Dynamic	527	124	64.0 KiB	
zonas	InnoDB	10	Dynamic	38	431	16.0 KiB	

Ilustración 2 Resumen de metadatos de las tablas del MVP

Por otro lado, se detalla la estructura de tablas y columnas, garantizando la uniformidad con el modelo lógico. La configuración 'Nullable: NO' en todos los atributos asegura la inexistencia de registros incompletos, validando la integridad de cada dimensión. Asimismo, la estandarización del ID_ZONA en todas las entidades permite realizar uniones (*joins*) de forma eficiente y precisa.



Estructura_DB_TFM_Manhattan* | tfm_siteselection_manhattan

Info | Tables | Columns | Indexes | Triggers | Views | Stored Procedures | Functions | Grants | Events

Table	Column	Type	Default Value	Nullable	Character Set	Collation	Privileges
censo	ID_ZONA	varchar(10)		NO	utf8mb4	utf8mb4_0900_...	select,insert,update,references
censo	POBLACION_TOTAL	int		NO			select,insert,update,references
censo	EDAD_MEDIANA	decimal(5,2)		NO			select,insert,update,references
censo	TASA_OCUPACION	decimal(5,2)		NO			select,insert,update,references
censo	INGRESO_MEDIO_HO...	decimal(12,2)		NO			select,insert,update,references
censo	TASA_EMPLEO	decimal(5,2)		NO			select,insert,update,references
censo	TAMANO_HOGAR_PRO...	decimal(4,2)		NO			select,insert,update,references
censo	POBLACION_HISPANA	int		NO			select,insert,update,references
costo_alquiler	ID_ZONA	varchar(10)		NO	utf8mb4	utf8mb4_0900_...	select,insert,update,references
costo_alquiler	PRECIO_SQTF_ANUAL	decimal(10,2)		NO			select,insert,update,references
lugares_interes	ID_ZONA	varchar(10)		NO	utf8mb4	utf8mb4_0900_...	select,insert,update,references
lugares_interes	CATEGORIA_LUGAR	varchar(30)		NO	utf8mb4	utf8mb4_0900_...	select,insert,update,references
lugares_interes	CANTIDAD_LUGARES	int		NO			select,insert,update,references
movilidad	ID_ZONA	varchar(10)		NO	utf8mb4	utf8mb4_0900_...	select,insert,update,references
movilidad	TIPO_DIA	varchar(15)		NO	utf8mb4	utf8mb4_0900_...	select,insert,update,references
movilidad	VOLUMEN_TOTAL_PA...	int		NO			select,insert,update,references
movilidad	CANTIDAD_ESTACION...	int		NO			select,insert,update,references
restaurantes	ID_ZONA	varchar(10)		NO	utf8mb4	utf8mb4_0900_...	select,insert,update,references
restaurantes	TIPO_COCINA	varchar(20)		NO	utf8mb4	utf8mb4_0900_...	select,insert,update,references
restaurantes	CANTIDAD_LOCALES	int		NO			select,insert,update,references
restaurantes	CLASIFICACION_COM...	varchar(20)		NO	utf8mb4	utf8mb4_0900_...	select,insert,update,references
seguridad	ID_ZONA	varchar(10)		NO	utf8mb4	utf8mb4_0900_...	select,insert,update,references
seguridad	TIPO_DELITO	varchar(30)		NO	utf8mb4	utf8mb4_0900_...	select,insert,update,references
seguridad	LUGAR_DELITO	varchar(20)		NO	utf8mb4	utf8mb4_0900_...	select,insert,update,references
seguridad	CANTIDAD_INCIDENTES	int		NO			select,insert,update,references
zonas	ID_ZONA	varchar(10)		NO	utf8mb4	utf8mb4_0900_...	select,insert,update,references
zonas	NOMBRE_ZONA	varchar(60)		NO	utf8mb4	utf8mb4_0900_...	select,insert,update,references
zonas	NOMBRE_DISTRITO	varchar(30)		NO	utf8mb4	utf8mb4_0900_...	select,insert,update,references
zonas	AREA_KM2	decimal(10,4)		NO			select,insert,update,references

Ilustración 3 Diccionario del MVP

Finalmente, se presenta el Diagrama Entidad-Relación (DER), que muestra visualmente la arquitectura en estrella del proyecto. Este gráfico confirma cómo las tablas de Censo y Costo de Alquiler se conectan con la tabla maestra ZONAS mediante claves foráneas (FK). Asimismo, detalla la unión de Lugares de Interés, Movilidad, Seguridad y Restaurantes a través de claves primarias compuestas. Esta estructura asegura que no existan errores en las relaciones y garantiza que cada dato sea único para el análisis.

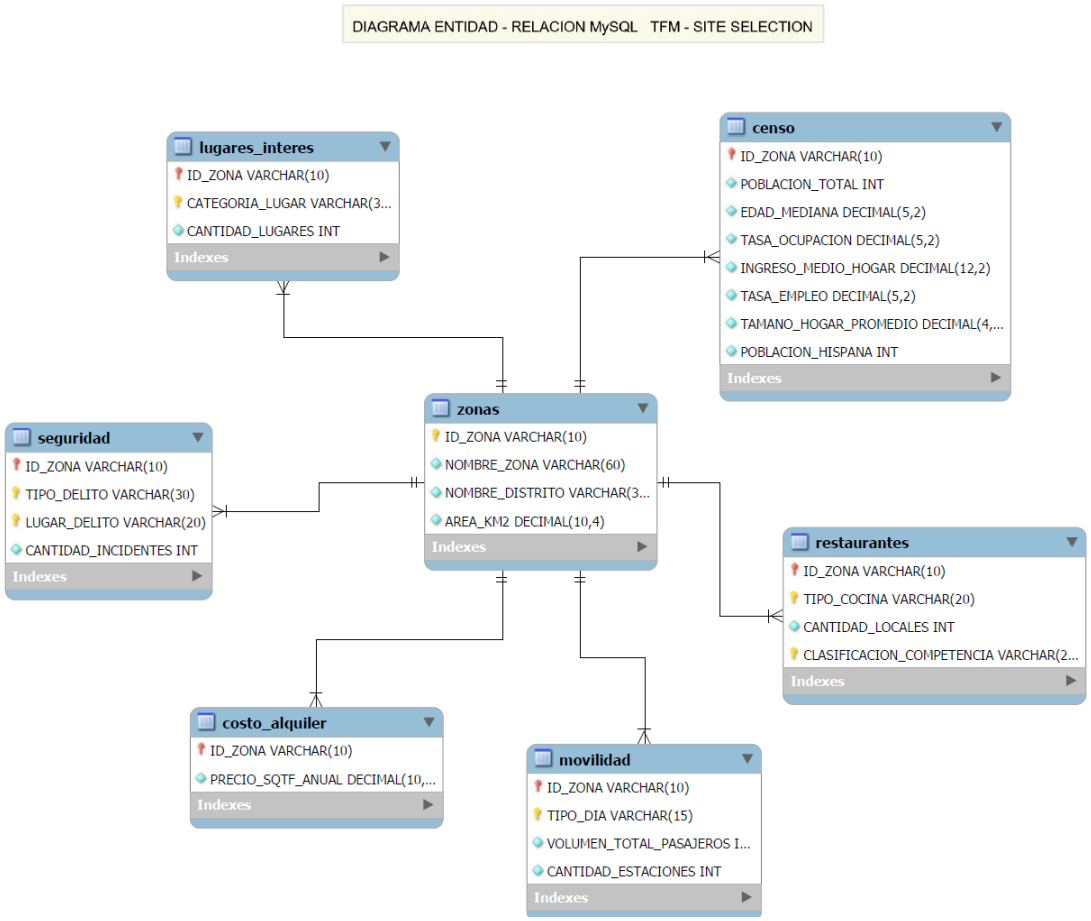


Ilustración 4 Diagrama Entidad–Relación del modelo Estrella en el SGBD

6. Consultas/operaciones representativas

Las consultas realizadas hasta el momento van en base a los objetivos planteados.
Primero, se separaron las zonas en cuartiles de acuerdo al costo de alquiler, otorgando un valor numérico para cada cuartil. Esto permite reducir casos atípicos al encontrarse agrupados.
Se utilizó la función 'NTILE(4)' para separar dichos cuartiles.

```
1 •  SELECT
2    ID_ZONA,
3    PRECIO_SQTF_ANUAL,
4    -- Se divide el mercado en 4 grupos: 1 (Barato) a 4 (Caro)
5    NTILE(4) OVER (ORDER BY PRECIO_SQTF_ANUAL ASC) AS CUARTIL_ALQUILER
6    FROM costo_alquiler
7    ORDER BY PRECIO_SQTF_ANUAL DESC;
```

ID_ZONA	PRECIO_SQTF_ANUAL	CUARTIL_ALQUILER
MN0502	199.67	4
MN0201	190.54	4
MN0603	190.00	4
MN0801	184.09	3
MN0803	170.00	3
MN0703	165.00	3
MN0601	140.00	3
MN0602	138.33	3
MN0303	127.58	3
MN0402	126.14	3
MN1201	124.13	3
MN0102	122.72	3
MN0701	110.00	2
MN1002	110.00	2
MN0302	107.90	2
MN0702	101.77	2
MN1102	100.00	2
MN0202	99.03	2
MN1202	95.00	2

Ilustración 5 - Código para cuartiles y tabla resultante

Una vez obtenida esta información, se procede a comparar los cuartiles con la tabla de restaurantes que son competencia directa para cada zona y cuartil. Se utiliza una expresión de tabla común (CTE) para que el código se mantenga organizado. La función 'COALESCE' cambia los valores 'NULL' por 0 en caso de que no exista competencia en dicha zona, mientras que utilizar 'SUM()' permite agruparlos por zona.

Con este cruce, existen diferentes variables que pueden observarse dependiendo de lo que se busque. Por ejemplo, un buen escenario podría estar en encontrar una zona entre cuartiles 1 y 2 con 0 rivales directos.

```
9   )
10  SELECT
11  z.NOMBRE_ZONA,
12  q.PRECIO_SQTF_ANUAL AS Renta_Anual,
13  q.CUARTIL_RENTA,
14  -- COALESCE asegura que si no hay rivales, aparezca un 0
15  COALESCE(SUM(r.CANTIDAD_LOCALES), 0) AS Rivales_Directos
16  FROM zonas z
17  JOIN Cuartiles q ON z.ID_ZONA = q.ID_ZONA
18  LEFT JOIN restaurantes r ON z.ID_ZONA = r.ID_ZONA
19  AND r.CLASIFICACION_COMPETENCIA = 'Directa (Rivales)'
20  GROUP BY
21  z.ID_ZONA,
22  z.NOMBRE_ZONA,
23  q.PRECIO_SQTF_ANUAL,
24  q.CUARTIL_RENTA
```

NOMBRE_ZONA	Renta_Anual	CUARTIL_RENTA	Rivales_Directos
Central Park	230.00	4	0
United Nations	210.00	4	0
Upper East Side-Carnegie Hill	200.00	4	4
Murray Hill-Kips Bay	190.00	4	9
SoHo-Little Italy-Hudson Square	190.54	4	12
East Midtown-Turtle Bay	220.00	4	17
West Village	220.81	4	18
East Harlem (South)	240.67	4	24
Midtown-Times Square	199.67	4	53
Stuyvesant Town-Peter Cooper Village	140.00	3	1
Upper West Side-Manhattan Valley	165.00	3	10
Tribeca-Civic Center	122.72	3	11
Upper East Side-Yorkville	170.00	3	13
Gramercy	138.33	3	13

Ilustración 6 - Código para cuartiles y rivales y tabla resultante

Siguiendo el ejemplo anterior, se seleccionan solo los cuartiles 1 y 2 y se añadió el factor de ingreso promedio de las zonas. En este caso, se procedió buscar una zona con un costo de renta bajo, alto ingreso y pocos o sin rivales directos. Se ordena por ingreso de forma descendente.

```
16  COALESCE(SUM(r.CANTIDAD_LOCALES), 0) AS Rivales_Directos
17  FROM zonas z
18  JOIN Cuartiles q ON z.ID_ZONA = q.ID_ZONA
19  JOIN censo c ON z.ID_ZONA = c.ID_ZONA
20  LEFT JOIN restaurantes r ON z.ID_ZONA = r.ID_ZONA
21  AND r.CLASIFICACION_COMPETENCIA = 'Directa (Rivales)'
22  -- Filtro para mostrar solo cuartiles 1 y 2
23  WHERE q.CUARTIL_RENTA IN (1, 2)
24  GROUP BY
25  z.NOMBRE_ZONA,
26  q.CUARTIL_RENTA,
27  q.PRECIO_SQTF_ANUAL,
28  c.INGRESO_MEDIO_HOGAR
29  ORDER BY
30  Ingreso_Anual DESC,      -- Mayor ingreso
31  Rivales_Directos ASC;    -- Sin competencia
```

NOMBRE_ZONA	CUARTIL_RENTA	Renta_Anual	Ingreso_Anual	Rivales_Directos
Financial District-Battery Park City	2	91.91	198961.00	24
Midtown South-Flatiron-Union Square	1	68.50	182280.00	26
Greenwich Village	2	99.03	174062.00	15
Upper West Side-Lincoln Square	2	110.00	164712.00	5
Upper West Side (Central)	2	101.77	159333.00	15
Chelsea-Hudson Yards	1	63.64	126272.00	24
Morningside Heights	1	76.36	87452.00	5
Washington Heights (North)	2	95.00	78465.00	38
Harlem (South)	1	70.00	74007.00	8
Hamilton Heights-Sugar Hill	2	95.00	68527.00	15
Inwood	1	49.27	68523.00	41
Lower East Side	2	107.90	58841.00	14
Harlem (North)	2	110.00	51386.00	4
Manhattanville-West Harlem	1	55.36	46819.00	7
East Harlem (North)	2	100.00	39957.00	32

Ilustración 7 - Código para cuartiles, competencia e ingresos y tabla resultante

7. Limitaciones y previsiones de evolución

El MVP implementado constituye una versión funcional y estructuralmente adecuada para el modelo relacional orientado al análisis territorial de Manhattan. Sin embargo, su alcance actual responde a un sprint enfocado en consolidación, limpieza, integración geoespacial y validación de integridad referencial. Bajo una lógica de desarrollo iterativo, el dataset y su arquitectura están diseñados para escalar progresivamente en sprints futuros.

Desde el punto de vista del crecimiento del sistema, las principales limitaciones identificadas no se consideran estructurales, sino estratégicas:

a) **Cobertura territorial limitada:** El modelo está circunscrito exclusivamente a Manhattan (38 zonas NTA). Aunque esta delimitación es coherente con el caso de estudio, restringe la aplicabilidad inmediata del sistema a otras ciudades.
b) **Corte temporal estático:** Los datasets utilizados corresponden a cortes específicos (2024–2025). El modelo no incorpora versionamiento temporal ni actualización automática, lo que impide análisis evolutivos o comparaciones interanuales.
c) **Nivel de agregación predefinido:** Las tablas de dimensiones (Seguridad, Movilidad, Restaurantes, Lugares de Interés) ya se encuentran agregadas por ID_ZONA y categoría. Esto mejora la eficiencia, pero limita el análisis a menor escala (calles específicas o direcciones exactas).
d) **Scoring no automatizado:** El modelo consolida métricas, pero el sistema de ponderación territorial aún depende de definiciones analíticas externas al SGBD.

Estas limitaciones no constituyen fallas del diseño, sino decisiones conscientes del alcance del proyecto.

7.1 Siguientes pasos en la evolución del dataset (futuros sprints)

El siguiente nivel de desarrollo consiste en transformar esta base en un dataset optimizado para análisis de Big Data mediante métodos de clusterización. Esta evolución implica pasar de una arquitectura orientada a integración y consulta relacional hacia una arquitectura orientada a segmentación territorial avanzada y detección de patrones latentes. La consolidación del dataset debe abordarse como un proceso progresivo de estructuración analítica, incremento de dimensionalidad y preparación para procesamiento escalable.

La evolución del dataset hacia un entorno preparado para clusterización con modelos de Big Data implica trascender de un sistema que integra datos para consulta y comparación, a una plataforma que identifica patrones emergentes en espacios multidimensionales complejos.

El modelo actual proporciona una base sólida para la evolución analítica del proyecto debido a cuatro elementos estructurales fundamentales. En primer lugar, la integridad referencial está garantizada mediante el uso consistente de claves primarias y foráneas, lo que asegura coherencia entre las dimensiones y evita inconsistencias en las relaciones entre entidades. En segundo lugar, la estructura en estrella facilita la eficiencia en consultas y permite una integración clara entre la entidad central y las tablas dimensionales, reduciendo complejidad operativa y favoreciendo la escalabilidad. En tercer lugar, la centralización mediante el identificador único ID_ZONA permite que todas las métricas converjan sobre una unidad territorial homogénea, condición indispensable para consolidar observaciones comparables en análisis multivariado. Finalmente, la cohesión entre fuentes heterogéneas —que integran datos sociodemográficos, movilidad, seguridad, competencia comercial, infraestructura urbana y costos inmobiliarios— proporciona una visión multidimensional del territorio, indispensable para procesos de segmentación avanzada.

A partir de esta base estructural, la consolidación futura del dataset permitirá que la arquitectura trascienda su función descriptiva inicial y se convierta en un entorno preparado para la segmentación territorial avanzada. Esta transición requerirá la construcción de una matriz analítica estandarizada en la que cada unidad territorial sea representada como un vector multidimensional de características cuantificables, derivadas y normalizadas a partir de las distintas dimensiones del modelo. La estandarización de variables, la ampliación de dimensionalidad y la integración temporal habilitarán la identificación de perfiles urbanos complejos, definidos no por un único indicador, sino por combinaciones estructurales de múltiples factores interrelacionados.

8. Herramienta de Repositorio

- **Repositorio GitHub:**

9. Conclusiones