

Análisis de las transacciones de recarga del
sistema de transporte publico de Bogotá para la
mejora de los tiempos de espera de los usuarios
en la fila.

Cendy Lozano, César Pedreros, Fabian Vargas

Facultad de Ingeniería y Ciencias Básicas,

Universidad Central

Maestría en Analítica de Datos

Curso de Bases de Datos

Bogotá, Colombia

¹*clozanog1@ucentral.edu.co*, ²*cpedrerosc@ucentral.edu.co*, ³*hvargasg1@ucentral.edu.co*

November 12, 2022

Contents

| | | |
|----------|---|-----------|
| 1 | Introducción (Max 250 Palabras) - (<i>Primera entrega</i>) | 4 |
| 2 | Características del proyecto de investigación (Max 500 Palabras) - (<i>Primera entrega</i>) | 4 |
| 2.1 | Titulo del proyecto de investigación (Max 100 Palabras) - (<i>Primera entrega</i>) | 4 |
| 2.2 | Objetivo general (Max 100 Palabras) - (<i>Primera entrega</i>) | 5 |
| 2.2.1 | Objetivos especificos (Max 100 Palabras) - (<i>Primera entrega</i>) | 5 |
| 2.3 | Alcance (Max 200 Palabras) - (<i>Primera entrega</i>) | 5 |
| 2.4 | Pregunta de investigación (Max 100 Palabras) - (<i>Primera entrega</i>) | 5 |
| 2.5 | Hipotesis (Max 100 Palabras) - (<i>Primera entrega</i>) | 5 |
| 3 | Reflexiones sobre el origen de datos e información (Max 400 Palabras) - (<i>Primera entrega</i>) | 6 |
| 3.1 | ¿Cual es el origen de los datos e información ? (Max 100 Palabras) - (<i>Primera entrega</i>) | 6 |
| 3.2 | ¿Cuales son las consideraciones legales o eticas del uso de la información? (Max 100 Palabras) - (<i>Primera entrega</i>) | 6 |
| 3.3 | ¿Cuales son los retos de la información y los datos que utilizara en la base de datos en terminos de la calidad y la consolidación? (Max 100 Palabras) - (<i>Primera entrega</i>) | 6 |
| 3.4 | ¿Que espera de la utilización de un sistema de Bases de Datos para su proyecto? (Max 100 Palabras) - (<i>Primera entrega</i>) | 7 |
| 4 | Diseño del Modelo de Datos del SMBD (Sistema Manejador de Bases de Datos)(<i>Primera entrega</i>) | 8 |
| 4.1 | Características del SMBD (Sistema Manejador de Bases de Datos) para el proyecto (<i>Primera entrega</i>) | 8 |
| 4.2 | Diagrama modelo de datos (<i>Primera entrega</i>) | 9 |
| 4.3 | Imágenes de la Base de Datos (<i>Primera entrega</i>) | 10 |
| 4.4 | Código SQL - lenguaje de definición de datos (DDL) (<i>Primera entrega</i>) | 12 |
| 4.5 | Código SQL - Manipulación de datos (DML) (<i>Primera entrega</i>) | 13 |
| 4.6 | Código SQL + Resultados: Vistas (<i>Primera entrega</i>) | 14 |
| 4.7 | Código SQL + Resultados: Triggers (<i>Primera entrega</i>) | 16 |
| 4.8 | Código SQL + Resultados: Funciones (<i>Primera entrega</i>) | 19 |
| 4.9 | Código SQL + Resultados: procedimientos almacenados (<i>Primera entrega</i>) | 20 |
| 5 | Bases de Datos No-SQL (<i>Segunda entrega</i>) | 22 |
| 5.1 | Diagrama Bases de Datos No-SQL (<i>Segunda entrega</i>) | 22 |
| 5.2 | SMBD utilizado para la Base de Datos No-SQL (<i>Segunda entrega</i>) | 44 |

| | | |
|----------|--|-----------|
| 6 | Aplicación de ETL (Extract, Transform, Load) y Bodega de Datos (<i>Tercera entrega</i>) | 49 |
| 6.1 | Ejemplo de aplicación de ETL y Bodega de Datos (<i>Tercera entrega</i>) | 49 |
| 7 | Lecciones aprendidas (<i>Tercera entrega</i>) | 50 |
| 8 | Bibliografía | 51 |

1 Introducción (Max 250 Palabras) - (*Primera entrega*)

La movilidad de las personas en las grandes ciudades del mundo siempre ha sido un gran reto, en Bogotá de los años 60 a los 90 se contaba con un sistema de transporte privado que no cubría las necesidades de sus habitantes. Diferentes administraciones distritales propusieron planes para mejorar los aspectos de la movilidad en la ciudad, sin embargo, fue hasta el año 2000 cuando se logró implementar la primera troncal de sistema Transmilenio S.A.[Andrade01] que se extendía por la toda la avenida Caracas. El sistema Transmilenio ha tenido muchos contradictores sin embargo se mantiene en la actualidad con 11 troncales, 138 estaciones y 9 portales distribuidos por toda la ciudad y con una troncal adicional en pleno proceso de construcción como lo es la troncal de la calle 68.

Sin embargo, es evidente que en diferentes sectores de la ciudad el sistema se encuentra saturado, los usuarios deben realizar largas filas para poder ingresar a él, lo cual ha generado malestar, inconformismo y descontento con el servicio prestado. La ciudadana requiere con urgencia soluciones que mitiguen este tipo de situaciones en las que los ciudadanos invierten su tiempo en realizar largas filas para poder realizar la recarga de los pasajes en sus tarjetas.

La situación mencionada se ha visto reflejada en una reducción considerable en los ingresos del sistema por el aumento del fenómeno de evasión de pago de pasajes (colados) sumado al periodo de pandemia que incrementó el desplome de recursos con los que el sistema masivo de transporte puede operar.

Este proyecto tiene como objetivo realizar un análisis descriptivo de la cantidad y valor de las transacciones de recarga del sistema masivo de transporte Transmilenio buscando de determinar cuáles son los puntos de menos frecuencia para darle la mejor opción al usuario para que realice su recarga.

2 Características del proyecto de investigación (Max 500 Palabras) - (*Primera entrega*)

El presente proyecto ha sido una investigación estructurada siguiendo de forma paulatina lo aprendido en el curso de Bases de Datos y colaborativa (sin ánimo de lucro) entre un grupo de profesionales de diversas áreas y que a la vez son estudiantes de la maestría de Analítica de Datos y una compañía privada que de forma muy atenta nos permitió usar data clasificada para analizar un proceso que es de los más críticos dentro del sistema de transporte masivo Transmilenio, como lo es del de recaudo y recargas de pasajes. La investigación usó únicamente data asociada a las troncales más importantes del sistema, taquillas, dispositivos de recarga y cantidad de recargas sin incluir información sensible de usuarios del sistema.

2.1 Título del proyecto de investigación (Max 100 Palabras) - (*Primera entrega*)

Análisis de las transacciones de recarga del sistema de transporte publico de Bogotá para la mejora de los tiempos de espera de los usuarios en la fila.

2.2 Objetivo general (Max 100 Palabras) - (*Primera entrega*)

Realizar un análisis descriptivo y prospectivo de la cantidad y valor de las transacciones y recargas del sistema masivo de transporte Transmilenio en sus zonas y taquillas con el fin de determinar cuáles son los puntos de menos frecuencia para darle la mejor opción al usuario para que realice su recarga.

2.2.1 Objetivos especificos (Max 100 Palabras) - (*Primera entrega*)

- Recopilar la información de las transacciones de recarga realizadas en las diferentes taquillas del sistema Transmilenio de los ultimo 5 años.
- Realizar un análisis descriptivo y predictivo de la información de recargas realizado en las diferentes estaciones del sistema Transmilenio.
- Identificar el comportamiento de recarga de las diferentes estaciones del sistema por troncal.
- Realizar un pronóstico de recargas para el año 2023.

2.3 Alcance (Max 200 Palabras) - (*Primera entrega*)

El presente proyecto tiene como alcance en primera medida, realizar el análisis de la información disponible de las recargas que se realizaron en el sistema de transporte masivo Transmilenio entre los años 2017 y 2022 de forma que permita tener un diagnóstico de la situación y comportamiento que se ha venido presentando del proceso de recargas del sistema. Una vez terminada esta fase diagnóstica se busca realizar un modelo que permita generar un pronóstico de las recargas para el año 2023.

2.4 Pregunta de investigación (Max 100 Palabras) - (*Primera entrega*)

¿Es posible mejorar los tiempos de recarga presencial en las diferentes estaciones del sistema Transmilenio por medio de un análisis que le permita a los usuarios determinar a través de un modelo cuales son las estaciones menos congestionadas para realizar la recarga de los pasajes?

2.5 Hipotesis (Max 100 Palabras) - (*Primera entrega*)

El modelo tipo pronóstico que se desarrolle permitirá predecir con un 95 por ciento de certidumbre el comportamiento que tendrán las recargas en las diferentes estaciones del sistema de forma que permita al usuario mejorar su experiencia respecto al tiempo de recarga.

3 Reflexiones sobre el origen de datos e información (Max 400 Palabras) - (*Primera entrega*)

Como es sabido, gran parte de la información que se cuenta sobre el sistema de transporte masivo Transmilenio está disponible en la red y en particular en la página de corporativa de <https://www.transmilenio.gov.co/>. Sin embargo, la información puntual del proceso de recargas no es información pública y el acceso a ella la tienen privados que son los operadores del recaudo como lo es Recaudo Bogotá. La oportunidad de acceso a esta información fue posible ya que uno de los integrantes del grupo de trabajo es funcionario de esta última y a partir de un proceso interno le dieron aprobación para poder tener disponible la información estadística de las recargas entre los años 2017-2022. Tomando en cuenta lo anterior, se hizo un acuerdo de manejo de confidencialidad con la persona integrante del equipo de forma que la data base para el proyecto fuera filtrada inicialmente por ella y que nos fuera entregado para el objetivo del proyecto la información menos sensible y que nos sirva para la generación del modelo de pronóstico.

3.1 ¿Cual es el origen de los datos e información ? (Max 100 Palabras) - (*Primera entrega*)

El origen de la data utilizada en este proyecto son los sistemas de gestión y monitoreo del sistema de recaudo de la compañía Recaudo Bogotá.

3.2 ¿Cuales son las consideraciones legales o eticas del uso de la información? (Max 100 Palabras) - (*Primera entrega*)

Como se mencionó en uno de los apartados anteriores, fue posible el acceso a la información base para el correcto desarrollo del proyecto a partir de la aprobación de los líderes del servicio de Recaudo Bogotá, aprovechando que uno de los integrantes del grupo es funcionaria de la mencionada compañía. Sin embargo, es importante acotar que hubo un acuerdo de confidencialidad entre todos los estudiantes del equipo, de forma que la información tomada de la data original no tuviera datos sensible o clasificados y que se usaran únicamente campos que tuvieran información numérica que permitiera filtrar cálculos estadísticos para generar el futuro modelo sin afectar aspectos legales como el Habeas Data, infracciones a derechos de autor, copias ilegales o éticos como hacer un uso indebido de la información para autobeneficio económico.

3.3 ¿Cuales son los retos de la información y los datos que utilizara en la base de datos en terminos de la calidad y la consolidación? (Max 100 Palabras) - (*Primera entrega*)

A partir del desarrollo ejecutado sobre la data original es bueno tomar en cuenta que la misma correspondía a más de 12 millones de registros los cuales nos fueron entregados en archivos .csv y que tenían un peso aproximado a 1 Tb. Por tanto uno de los retos iniciales por parte de los dueños de la información

fue la generación de la misma, ya que de los 4 archivos .csv facilitados por los líderes de Recaudo Bogotá les tomó aproximadamente 8 horas cada archivo.

A continuación el paso a seguir (segundo reto) fue la revisión y selección de los campos que nos permitirá hacer el análisis y generar el futuro pronóstico de la amplia cantidad de información que se tenía originalmente.

El siguiente reto fue la carga de los archivos en el SGDB para continuar con la creación de tablas de forma tal que se pudiesen normalizar y definir relaciones optimas para la extracción de la data.

3.4 ¿Que espera de la utilización de un sistema de Bases de Datos para su proyecto? (Max 100 Palabras) - (*Primera entrega*)

Nuestro principal objetivo del uso de BD en el proyecto es la optimización en el manejo y acceso ordenado a la información clave para nuestro futuro modelo de pronóstico, asegurando que la información no va a estar duplicada de forma que se afecten nuestros resultados.

4 Diseño del Modelo de Datos del SMBD (Sistema Manejador de Bases de Datos) *(Primera entrega)*

4.1 Características del SMBD (Sistema Manejador de Bases de Datos) para el proyecto *(Primera entrega)*

El SMBD que usamos para el desarrollo de nuestro proyecto fue MySQL 8.0 cuyas características las podemos resumir de la siguiente manera:

- Rendimiento: MySQL 8.0 ofrece un mejor rendimiento en las siguientes áreas: cargas de trabajo de lectura / escritura, cargas de trabajo intensivas en E / S y cargas de trabajo de alta competencia.
- NoSQL: MySQL desde la versión 5.7 proporcionar la función de almacenamiento NoSQL. Esta función también se ha mejorado considerablemente en la versión 8.0, permitiendo eliminar la necesidad de una base de datos de documentos NoSQL separada.
- Funciones de Ventana: A partir de MySQL 8.0, se ha agregado un nuevo concepto llamado función de ventana, que se puede usar para implementar varios métodos de consulta nuevos. Las funciones de ventana son similares a las funciones agregadas como SUM () y COUNT (), pero no combina varias filas de resultados de consulta en una fila, sino que vuelve a colocar los resultados en varias filas. Es decir, la función de ventana no requiere GROUP BY.
- Fiabilidad: InnoDB ahora admite la atomicidad de la tabla DDL, es decir, el DDL en la tabla InnoDB también puede lograr la integridad de la transacción, ya sea fallando al retroceder o confirmando con éxito, de modo que no se produzca el éxito parcial de DDL. Además, también es compatible con la función a prueba de choques. Los metadatos se almacenan en un único diccionario de datos de transacciones.
- Seguridad: Mejoras en OpenSSL, nueva autenticación predeterminada, roles SQL, seguridad de contraseña, autorización.
- Requerimientos de instalación:
 - 512 Mb de memoria Ram.
 - 1024 Mb maquina virtual.
 - 1 GB de espacio de disco duro.
 - Sistema operativo: Windows, Linux y Unix.
 - Arquitectura del sistema 32/64 bit.

4.2 Diagrama modelo de datos (Primera entrega)

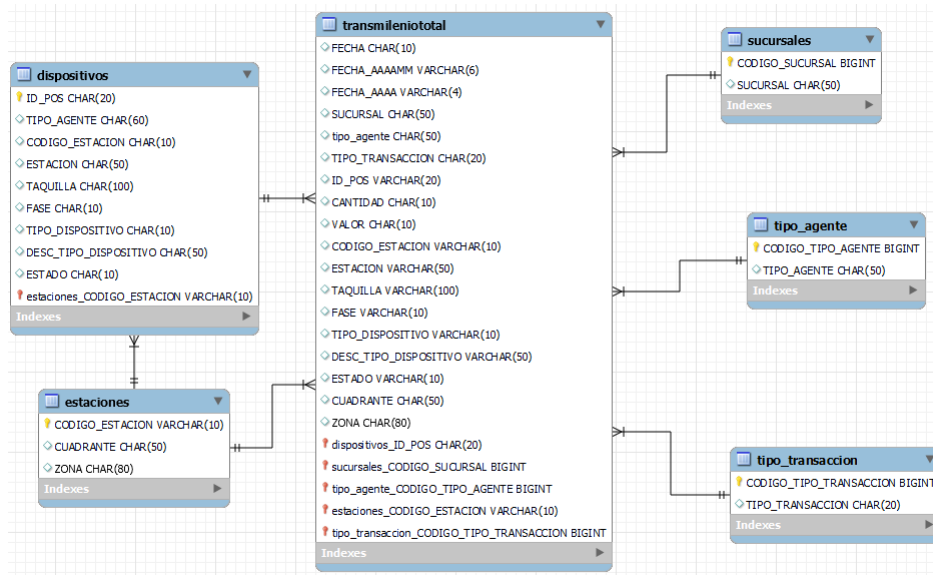


Figure 1: Modelo Base de Datos

4.3 Imágenes de la Base de Datos (*Primera entrega*)

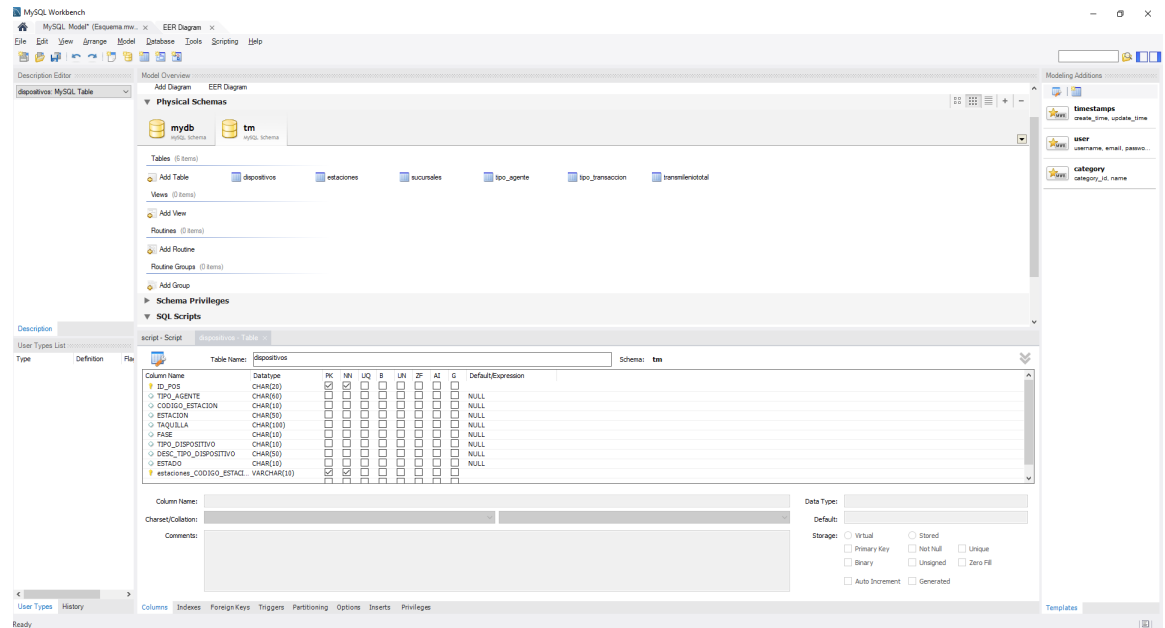


Figure 2: Tabla Dispositivos

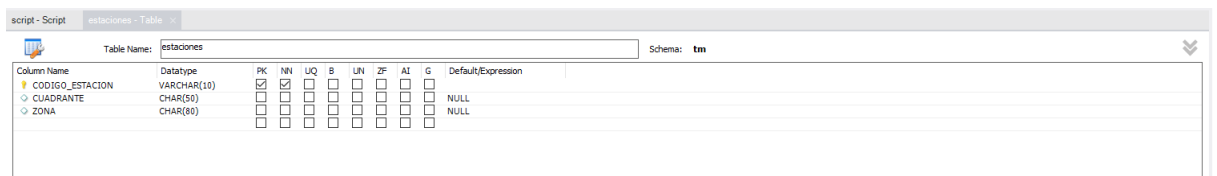


Figure 3: Tabla Estaciones

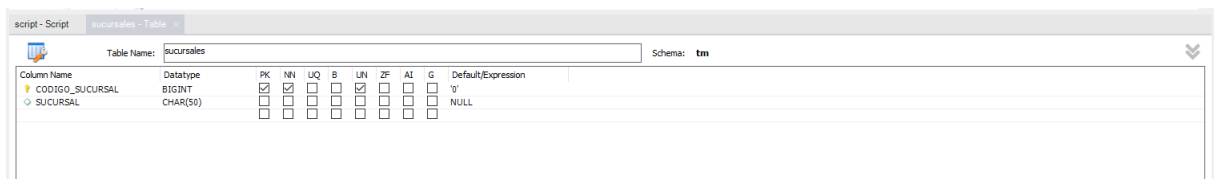


Figure 4: Tabla Sucursales

script - Script tipo_agente - Table

Table Name: Schema: **tm**

| Column Name | Datatype | PK | NN | UQ | B | UN | ZF | AI | G | Default/Expression |
|----------------------|----------|-------------------------------------|-------------------------------------|--------------------------|--------------------------|-------------------------------------|--------------------------|--------------------------|--------------------------|--------------------|
| ↳ CODIGO_TIPO_AGENTE | BIGINT | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input checked="" type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | 0' |
| ↳ TIPO_AGENTE | CHAR(50) | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | NULL |

Figure 5: Tabla Tipo Agente

script - Script tipo_transaccion - Table

Table Name: Schema: **tm**

| Column Name | Datatype | PK | NN | UQ | B | UN | ZF | AI | G | Default/Expression |
|---------------------------|----------|-------------------------------------|-------------------------------------|--------------------------|--------------------------|-------------------------------------|--------------------------|--------------------------|--------------------------|--------------------|
| ↳ CODIGO_TIPO_TRANSACCION | BIGINT | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input checked="" type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | 0' |
| ↳ TIPO_TRANSACCION | CHAR(20) | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | NULL |

Figure 6: Tabla Tipo Transacción

Physical Schemas

mydb tm

Tables (5 items)

dispositivos estaciones sucursales tipo_agente tipo_transaccion transmileniototal

Views (0 items)

script - Script transmileniototal - Table

Table Name: Schema: **tm**

| Column Name | Datatype | PK | NN | UQ | B | UN | ZF | AI | G | Default/Expression |
|--------------------------------|--------------|-------------------------------------|-------------------------------------|-------------------------------------|--------------------------|-------------------------------------|--------------------------|--------------------------|--------------------------|--------------------|
| ↳ fecha | CHAR(10) | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | NULL |
| ↳ FECHA_AAAA | VARCHAR(4) | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | NULL |
| ↳ SUCURSAL | CHAR(50) | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | NULL |
| ↳ tipo_agente | CHAR(50) | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | NULL |
| ↳ TIPO_TRANSACCION | CHAR(20) | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | NULL |
| ↳ ID_POS | VARCHAR(20) | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | NULL |
| ↳ CANTIDAD | CHAR(10) | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | NULL |
| ↳ VALOR | CHAR(10) | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | NULL |
| ↳ CODIGO_ESTACION | VARCHAR(10) | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | NULL |
| ↳ ESTACION | VARCHAR(30) | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | NULL |
| ↳ TAQUILLA | VARCHAR(100) | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | NULL |
| ↳ FASE | VARCHAR(10) | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | NULL |
| ↳ TIPO_DISPOSITIVO | VARCHAR(10) | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | NULL |
| ↳ DESC_TIPO_DISPOSITIVO | VARCHAR(50) | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | NULL |
| ↳ ESTADO | VARCHAR(10) | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | NULL |
| ↳ CUADRANTE | CHAR(50) | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | NULL |
| ↳ ZONA | CHAR(80) | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | NULL |
| ↳ dispositivos_ID_POS | CHAR(20) | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | |
| ↳ sucursales_CODIGO_SUCUR... | BIGINT | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input type="checkbox"/> | <input checked="" type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | |
| ↳ tipo_agente_CODIGO_TIPO... | BIGINT | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input type="checkbox"/> | <input checked="" type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | |
| ↳ estaciones_CODIGO_ESTACI... | VARCHAR(10) | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input type="checkbox"/> | <input checked="" type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | |
| ↳ tipo_transaccion_CODIGO_T... | BIGINT | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input type="checkbox"/> | <input checked="" type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | |

Figure 7: Tabla Transmilenio Total

4.4 Código SQL - lenguaje de definición de datos (DDL) (Primera entrega)

```
LOAD DATA INFILE
'C:\\ProgramData\\MySQL\\MySQL_Server_8.0\\Uploads\\
Estaciones.txt'
INTO TABLE ESTACIONES
FIELDS TERMINATED BY ',';
```

```
CREATE TABLE TRANSMILENIOTEMP (
FECHA                                CHAR(10),
SUCURSAL                            CHAR(50),
TIPO_AGENTE                         CHAR(50),
NOMBREAGENTE                        CHAR(80),
ESTACION                            CHAR(50),
ACCESO_ESTACION                     CHAR(100),
ID_POS                              CHAR(20),
TIPO_TRANSACCION                     CHAR(20),
CANTIDAD                            CHAR(10),
VALOR                               CHAR(20)
);
COMMIT;
```

```
CREATE TABLE DISPOSITIVOS (
ID_POS                              CHAR(20),
TIPO_AGENTE                         CHAR(60),
CODIGO_ESTACION                     CHAR(10),
ESTACION                            CHAR(50),
TAQUILLA                            CHAR(100),
FASE                                CHAR(10),
TIPO_DISPOSITIVO                     CHAR(10),
DESC_TIPO_DISPOSITIVO                CHAR(50),
ESTADO                              CHAR(10)
);
COMMIT;
```

```
CREATE TABLE TIPO_AGENTETEMP AS (
SELECT DISTINCT TIPO_AGENTE FROM TRANSMILENIODEF
);
COMMIT;
```

4.5 Código SQL - Manipulación de datos (DML) (Primera entrega)

```
SELECT * FROM TRANSMILENIOTOTAL;##10260365
SELECT COUNT(*) FROM TRANSMILENIOTOTAL;##10260365
```

```
SELECT SUM(CANTIDAD) CANTIDAD,SUM(VALOR) VALOR,
FECHA_AAAAMM, ESTACION, TAQUILLA,
DESC.TIPO_DISPOSITIVO ,CUADRANTE,ZONA
FROM TRANSMILENIOTOTAL
WHERE CUADRANTE IS NOT NULL
GROUP BY FECHA_AAAAMM, ESTACION, TAQUILLA,
DESC.TIPO_DISPOSITIVO ,CUADRANTE,ZONA;
```

```
SELECT
TRANSMILENIODEFNESTNDISP.FECHA, TRANSMILENIODEFNESTNDISP.FECHA_AAAAMM,
TRANSMILENIODEFNESTNDISP.FECHA_AAAA,
SUCURSALES.SUCURSAL, TIPO_AGENTE.tipo_agente ,
tipo_transaccion.TIPO_TRANSACCION,
TRIM(LEADING '0' FROM TRANSMILENIODEFNESTNDISP.ID_POS)
ID_POS, TRANSMILENIODEFNESTNDISP.CANTIDAD,
TRANSMILENIODEFNESTNDISP.VALOR,
TRIM(LEADING '0' FROM DISPOSITIVOS.CODIGO_ESTACION)
CODIGO_ESTACION, TRIM(DISPOSITIVOS.ESTACION)
ESTACION, TRIM(DISPOSITIVOS.TAQUILLA) TAQUILLA,
TRIM(DISPOSITIVOS.FASE)
FASE, TRIM(DISPOSITIVOS.TIPO_DISPOSITIVO)
TIPO_DISPOSITIVO, TRIM(DISPOSITIVOS.DESC_TIPO_DISPOSITIVO)
DESC_TIPO_DISPOSITIVO,
TRIM(DISPOSITIVOS.ESTADO) ESTADO
FROM
TRANSMILENIODEFNESTNDISP INNER JOIN SUCURSALES ON
TRIM(TRANSMILENIODEFNESTNDISP.CODIGO_SUCURSAL)=
TRIM(SUCURSALES.CODIGO_SUCURSAL)
INNER JOIN TIPO_AGENTE ON
TRIM(TRANSMILENIODEFNESTNDISP.CODIGO_TIPO_AGENTE)=
TRIM(TIPO_AGENTE.CODIGO_TIPO_AGENTE)
INNER JOIN TIPO_TRANSACCION ON
TRIM(TRANSMILENIODEFNESTNDISP.CODIGO_TIPO_TRANSACCION)=
TRIM(TIPO_TRANSACCION.CODIGO_TIPO_TRANSACCION)
LEFT JOIN DISPOSITIVOS ON TRIM(LEADING '0' FROM
TRANSMILENIODEFNESTNDISP.ID_POS)=TRIM(LEADING '0' FROM
DISPOSITIVOS.ID_POS)
);
COMMIT;
```

```

INSERT INTO TRANSMILENIODEF (
SELECT * FROM TRANSMILENIOTEMP
);
COMMIT;

```

```

UPDATE TRANSMILENIOTEMP SET
FECHA=REPLACE(FECHA, ' ' , ' ');
UPDATE TRANSMILENIOTEMP SET
SUCURSAL=REPLACE(SUCURSAL, ' ' , ' ');
UPDATE TRANSMILENIOTEMP SET
TIPO_AGENTE=REPLACE(TIPO_AGENTE, ' ' , ' ');
UPDATE TRANSMILENIOTEMP SET
NOMBREAGENTE=REPLACE(NOMBREAGENTE, ' ' , ' ');
UPDATE TRANSMILENIOTEMP SET
ESTACION=REPLACE(ESTACION, ' ' , ' ');
UPDATE TRANSMILENIOTEMP SET
ACCESO_ESTACION=REPLACE(ACCESO_ESTACION, ' ' , ' ');
UPDATE TRANSMILENIOTEMP SET
ID_POS=REPLACE(ID_POS, ' ' , ' ');
UPDATE TRANSMILENIOTEMP SET
TIPO_TRANSACCION=REPLACE(TIPO_TRANSACCION, ' ' , ' ');
UPDATE TRANSMILENIOTEMP SET
CANTIDAD=REPLACE(CANTIDAD, ' ' , ' ');
UPDATE TRANSMILENIOTEMP SET
VALOR=REPLACE(VALOR, ' ' , ' ');
COMMIT;

```

4.6 Código SQL + Resultados: Vistas *(Primera entrega)*

```

DROP VIEW VISTA_CODIFICADA;
CREATE VIEW VISTA_CODIFICADA AS (
SELECT * FROM TRANSMILENIODEFNESTNDISP
);
COMMIT;

```

```

DROP VIEW VISTA_ESTACIONES;
CREATE VIEW VISTA_ESTACIONES AS (
SELECT FECHA_AAAA_MM, SUCURSAL, tipo_agente ,
TIPO_TRANSACCION, CANTIDAD, VALOR, ESTACION,
TAQUILLA, FASE, DESC.TIPO_DISPOSITIVO,
ESTADO FROM TRANSMILENIOTOTAL
);
COMMIT;

```

```

DROP VIEW VISTA.CUADRANTES;
CREATE VIEW VISTA.CUADRANTES AS (
SELECT FECHA_AAAAMM, SUCURSAL, tipo_agente ,
TIPO_TRANSACCION, CANTIDAD, VALOR, CUADRANTE,
TAQUILLA, FASE, DESC.TIPO_DISPOSITIVO, ESTADO
FROM TRANSMILENIOTOTAL
);
COMMIT;

DROP VIEW VISTA.ZONAS;
CREATE VIEW VISTA.ZONAS AS (
SELECT FECHA_AAAAMM, SUCURSAL, tipo_agente ,
TIPO_TRANSACCION, CANTIDAD, VALOR, ZONA,
TAQUILLA, FASE, DESC.TIPO_DISPOSITIVO,
ESTADO FROM TRANSMILENIOTOTAL
);
COMMIT;

SELECT * FROM VISTA.CODIFICADA;
SELECT * FROM VISTA.ESTACIONES;
SELECT * FROM VISTA.ZONAS;
SELECT * FROM VISTA.CUADRANTES;

```

| Result Grid | | | | | | | | | |
|--|----------|--------------|------------|-----------------|--------------------|-------------------------|---------------------|------------------------|-----------------------------------|
| Filter Rows: | | | | | | | | | |
| Exports: Wrap Cell Contents: Fetch rows: | | | | | | | | | |
| | FECHA | FECHA_AAAAMM | FECHA_AAAA | CODIGO_SUCURSAL | CODIGO_TIPO_AGENTE | CODIGO_TIPO_TRANSACCION | NOMBRE_AGENTE | ESTACION | ACCESO_ESTACION |
| ▶ | 20170201 | 201702 | 2017 | 2 | 3 | 1 | Agente RB(00000001) | 21 ANGELES(03003) | Oriente |
| | 20170201 | 201702 | 2017 | 2 | 3 | 2 | Agente RB(00000001) | 21 ANGELES(03003) | Oriente |
| | 20170201 | 201702 | 2017 | 2 | 3 | 1 | Agente RB(00000001) | Alcalá(02200) | TAQUILLA UNO VAGON NORTE ALCALÁ |
| | 20170201 | 201702 | 2017 | 2 | 3 | 2 | Agente RB(00000001) | Alcalá(02200) | TAQUILLA UNO VAGON NORTE ALCALÁ |
| | 20170201 | 201702 | 2017 | 2 | 3 | 1 | Agente RB(00000001) | Alcalá(02200) | TAQUILLA UNO VAGON NORTE ALCALÁ |
| | 20170201 | 201702 | 2017 | 2 | 3 | 2 | Agente RB(00000001) | Alcalá(02200) | TAQUILLA UNO VAGON NORTE ALCALÁ |
| | 20170201 | 201702 | 2017 | 2 | 3 | 1 | Agente RB(00000001) | Alcalá(02200) | TAQUILLA UNO VAGON NORTE ALCALÁ |
| | 20170201 | 201702 | 2017 | 2 | 3 | 2 | Agente RB(00000001) | Alcalá(02200) | TAQUILLA UNO VAGON NORTE ALCALÁ |
| | 20170201 | 201702 | 2017 | 2 | 3 | 2 | Agente RB(00000001) | Alcalá(02200) | TAQUILLA UNO VAGON SUR ALCALÁ |
| | 20170201 | 201702 | 2017 | 2 | 3 | 1 | Agente RB(00000001) | Alcalá(02200) | TAQUILLA UNO VAGON SUR ALCALÁ |
| | 20170201 | 201702 | 2017 | 2 | 3 | 2 | Agente RB(00000001) | Alcalá(02200) | TAQUILLA UNO VAGON SUR ALCALÁ |
| | 20170201 | 201702 | 2017 | 2 | 3 | 2 | Agente RB(00000001) | Alcalá(02200) | TAQUILLA UNO VAGON SUR ALCALÁ |
| | 20170201 | 201702 | 2017 | 2 | 3 | 1 | Agente RB(00000001) | Alcalá(02200) | TAQUILLA UNO VAGON SUR ALCALÁ |
| | 20170201 | 201702 | 2017 | 2 | 3 | 2 | Agente RB(00000001) | Alcalá(02200) | TAQUILLA UNO VAGON SUR ALCALÁ |
| | 20170201 | 201702 | 2017 | 2 | 3 | 1 | Agente RB(00000001) | ALQUERIA(07005) | TAQUILLA UNO VAGON NORTE ALQUE... |
| | 20170201 | 201702 | 2017 | 2 | 3 | 2 | Agente RB(00000001) | ALQUERIA(07005) | TAQUILLA UNO VAGON NORTE ALQUE... |
| | 20170201 | 201702 | 2017 | 2 | 3 | 2 | Agente RB(00000001) | ALQUERIA(07005) | TAQUILLA UNO VAGON NORTE ALQUE... |
| | 20170201 | 201702 | 2017 | 2 | 3 | 1 | Agente RB(00000001) | Américas - Cr. 53(0... | TAQUILLA UNO VAGON OCCIDENTE A... |

Figure 8: Vista Codificada

| Result Grid | | | | | | | | | | |
|--------------|-----------|------------------|-------------------|----------|--------------------|----------|-------------------------|--------|---------------------------------|--|
| Filter Rows: | | | Export: | | Wrap Cell Content: | | Fetch rows: | | | |
| FECHA_AAAAMM | SUCURSAL | tipo_agente | TIPO_TRANSACCION | CANTIDAD | VALOR | ESTACION | TAQUILLA | FASE | DESC_TIPO_DISPOSITIVO | |
| 201702 | Agente RB | Estación Troncal | Anulacion Tarjeta | 2 | 21000 | Tintal | Sur | Fase 2 | Terminal de Carga Asistida | |
| 201702 | Agente RB | Estación Troncal | Carga de Tarjeta | 2420 | 17617300 | Tintal | Norte | Fase 2 | Terminal de Carga Asistida | |
| 201702 | Agente RB | Estación Troncal | Anulacion Tarjeta | 3 | 14600 | Tintal | Norte | Fase 2 | Terminal de Carga Asistida | |
| 201702 | Agente RB | Estación Troncal | Carga de Tarjeta | 2687 | 18733050 | Tintal | Norte | Fase 2 | Terminal de Carga Asistida | |
| 201702 | Agente RB | Estación Troncal | Anulacion Tarjeta | 6 | 37300 | Tintal | Norte | Fase 2 | Terminal de Carga Asistida | |
| 201702 | Agente RB | Estación Troncal | Carga de Tarjeta | 987 | 6272900 | Tintal | | Fase 2 | Dispositivo de Carga Móvil | |
| 201702 | Agente RB | Estación Troncal | Anulacion Tarjeta | 1 | 10000 | Tintal | | Fase 2 | Dispositivo de Carga Móvil | |
| 201702 | Agente RB | Estación Troncal | Carga de Tarjeta | 225 | 1025000 | Terreros | Taquilla Peatonal Norte | Fase 2 | Dispositivo de Carga Automática | |

Figure 9: Vista Estaciones

| Result Grid | | | | | | | | | | |
|--------------|-----------|------------------|-------------------|----------|--------------------|------------------------------|-------------------------|--------|---------------------------------|--|
| Filter Rows: | | | Export: | | Wrap Cell Content: | | Fetch rows: | | | |
| FECHA_AAAAMM | SUCURSAL | tipo_agente | TIPO_TRANSACCION | CANTIDAD | VALOR | ZONA | TAQUILLA | FASE | DESC_TIPO_DISPOSITIVO | |
| 201702 | Agente RB | Estación Troncal | Anulacion Tarjeta | 2 | 21000 | P. Americas - CDS Carrera 32 | Sur | Fase 2 | Terminal de Carga Asistida | |
| 201702 | Agente RB | Estación Troncal | Carga de Tarjeta | 2420 | 17617300 | P. Americas - CDS Carrera 32 | Norte | Fase 2 | Terminal de Carga Asistida | |
| 201702 | Agente RB | Estación Troncal | Anulacion Tarjeta | 3 | 14600 | P. Americas - CDS Carrera 32 | Norte | Fase 2 | Terminal de Carga Asistida | |
| 201702 | Agente RB | Estación Troncal | Carga de Tarjeta | 2687 | 18733050 | P. Americas - CDS Carrera 32 | Norte | Fase 2 | Terminal de Carga Asistida | |
| 201702 | Agente RB | Estación Troncal | Anulacion Tarjeta | 6 | 37300 | P. Americas - CDS Carrera 32 | Norte | Fase 2 | Terminal de Carga Asistida | |
| 201702 | Agente RB | Estación Troncal | Carga de Tarjeta | 987 | 6272900 | P. Americas - CDS Carrera 32 | | Fase 2 | Dispositivo de Carga Móvil | |
| 201702 | Agente RB | Estación Troncal | Anulacion Tarjeta | 1 | 10000 | P. Americas - CDS Carrera 32 | | Fase 2 | Dispositivo de Carga Móvil | |
| 201702 | Agente RB | Estación Troncal | Carga de Tarjeta | 225 | 1025000 | San Mateo - Tygua | Taquilla Peatonal Norte | Fase 2 | Dispositivo de Carga Automática | |
| 201702 | Agente RB | Estación Troncal | Carga de Tarjeta | 2746 | 20006450 | San Mateo - Tygua | Taquilla Peatonal Norte | Fase 2 | Terminal de Carga Asistida | |

Figure 10: Vista Zonas

| Result Grid | | | | | | | | | | |
|--------------|-----------|------------------|-------------------|----------|--------------------|----------------------------------|-------------------------|--------|---------------------------------|--|
| Filter Rows: | | | Export: | | Wrap Cell Content: | | Fetch rows: | | | |
| FECHA_AAAAMM | SUCURSAL | tipo_agente | TIPO_TRANSACCION | CANTIDAD | VALOR | CUADRANTE | TAQUILLA | FASE | DESC_TIPO_DISPOSITIVO | |
| 201702 | Agente RB | Estación Troncal | Anulacion Tarjeta | 2 | 21000 | Portal Americas - Transversal 86 | Sur | Fase 2 | Terminal de Carga Asistida | |
| 201702 | Agente RB | Estación Troncal | Carga de Tarjeta | 2420 | 17617300 | Portal Americas - Transversal 86 | Norte | Fase 2 | Terminal de Carga Asistida | |
| 201702 | Agente RB | Estación Troncal | Anulacion Tarjeta | 3 | 14600 | Portal Americas - Transversal 86 | Norte | Fase 2 | Terminal de Carga Asistida | |
| 201702 | Agente RB | Estación Troncal | Carga de Tarjeta | 2687 | 18733050 | Portal Americas - Transversal 86 | Norte | Fase 2 | Terminal de Carga Asistida | |
| 201702 | Agente RB | Estación Troncal | Anulacion Tarjeta | 6 | 37300 | Portal Americas - Transversal 86 | Norte | Fase 2 | Terminal de Carga Asistida | |
| 201702 | Agente RB | Estación Troncal | Carga de Tarjeta | 987 | 6272900 | Portal Americas - Transversal 86 | | Fase 2 | Dispositivo de Carga Móvil | |
| 201702 | Agente RB | Estación Troncal | Anulacion Tarjeta | 1 | 10000 | Portal Americas - Transversal 86 | | Fase 2 | Dispositivo de Carga Móvil | |
| 201702 | Agente RB | Estación Troncal | Carga de Tarjeta | 225 | 1025000 | San Mateo - Despensa | Taquilla Peatonal Norte | Fase 2 | Dispositivo de Carga Automática | |
| 201702 | Agente RB | Estación Troncal | Carga de Tarjeta | 2746 | 20006450 | San Mateo - Despensa | Taquilla Peatonal Norte | Fase 2 | Terminal de Carga Asistida | |
| 201702 | Agente RB | Estación Troncal | Carga de Tarjeta | 1994 | 13198200 | San Mateo - Despensa | Taquilla Peatonal Norte | Fase 2 | Terminal de Carga Asistida | |
| 201702 | Agente RB | Estación Troncal | Carga de Tarjeta | 507 | 3110900 | San Mateo - Despensa | | Fase 2 | Dispositivo de Carga Móvil | |
| 201702 | Agente RB | Estación Troncal | Anulacion Tarjeta | 1 | 3000 | San Mateo - Despensa | | Fase 2 | Dispositivo de Carga Móvil | |
| 201702 | Agente RB | Estación Troncal | Carga de Tarjeta | 2968 | 19591650 | San Mateo - Despensa | Taquilla Peatonal Sur | Fase 2 | Terminal de Carga Asistida | |
| 201702 | Agente RB | Estación Troncal | Anulacion Tarjeta | 8800 | 8800 | San Mateo - Despensa | Taquilla Peatonal Sur | Fase 2 | Terminal de Carga Asistida | |

Figure 11: Vista Cuadrantes

4.7 Código SQL + Resultados: Triggers (*Primera entrega*)

```
CREATE TRIGGER LOG_SUCURSALES AFTER INSERT ON SUCURSALES
FOR EACH ROW
    INSERT INTO SUCURSALES_LOG (ACCION)
    VALUE ( 'Se creó un Registro' );
```



```

CREATE TRIGGER LOG_SUCURSALES2 AFTER
DELETE ON SUCURSALES
FOR EACH ROW
    INSERT INTO SUCURSALES_LOG (ACCION)
    VALUE ( 'Se_borr_un_Registro' ) ;

CREATE TRIGGER LOG_SUCURSALES3 AFTER
UPDATE ON SUCURSALES
FOR EACH ROW
    INSERT INTO SUCURSALES_LOG (ACCION)
    VALUE ( 'Se_actualiz_un_Registro' ) ;

#####

CREATE TRIGGER LOG_TIPO_AGENTE AFTER
INSERT ON TIPO_AGENTE
FOR EACH ROW
    INSERT INTO TIPO_AGENTE_LOG (ACCION)
    VALUE ( 'Se_creo_un_Registro' ) ;

CREATE TRIGGER LOG_TIPO_AGENTE2 AFTER
DELETE ON TIPO_AGENTE
FOR EACH ROW
    INSERT INTO TIPO_AGENTE_LOG (ACCION)
    VALUE ( 'Se_borr_un_Registro' ) ;

CREATE TRIGGER LOG_TIPO_AGENTE3 AFTER
UPDATE ON TIPO_AGENTE
FOR EACH ROW
    INSERT INTO TIPO_AGENTE_LOG (ACCION)
    VALUE ( 'Se_actualiz_un_Registro' ) ;

#####

CREATE TRIGGER LOG_TIPO_TRANSACCION
AFTER INSERT ON TIPO_TRANSACCION
FOR EACH ROW
    INSERT INTO TIPO_TRANSACCION_LOG
    (ACCION) VALUE ( 'Se_creo_un_Registro' ) ;

CREATE TRIGGER LOG_TIPO_TRANSACCION2
AFTER DELETE ON TIPO_TRANSACCION
FOR EACH ROW
    INSERT INTO TIPO_TRANSACCION_LOG
    (ACCION) VALUE ( 'Se_borr_un_Registro' ) ;

```

```

CREATE TRIGGER LOG_TIPO_TRANSACCION3
AFTER UPDATE ON TIPO_TRANSACCION
FOR EACH ROW
INSERT INTO TIPO_TRANSACCION_LOG
(ACCION) VALUE ( 'Se actualiz _un_Registro' ) ;

```

A continuación se presenta el resultado de ejecutar el trigger:

| CODIGO_SUCURSAL | SUCURSAL |
|-----------------|----------------------------------|
| 9 | Convenio Universidad Andes |
| 10 | Convenio universidad del Rosario |
| 11 | Ecotrasmi - Tesirve |
| 12 | Instituto Distrital IDIPRON |
| 13 | MovilRed Digital |
| 14 | MovilRed Headquarter |
| 15 | Platik |
| 16 | RB Digital |
| 17 | Red Gelsa |
| 18 | Red RB |
| 19 | Servibanca |
| 20 | Soluciones en Red SAS |
| 21 | Soluciones en Red SAS Digital |
| 22 | Tuya Digital |
| 23 | Universidad Andes |
| 24 | Universidad de los Andes |
| 25 | Universidad Javerian |
| 26 | dd |
| NULL | NULL |

Figure 12: Ejecución Trigger

| CODIGO | ACCION | FECHA |
|--------|-----------------------|---------------------|
| 56 | Se creo un Registro | 2022-11-12 08:34:01 |
| 57 | Se actualizó un Re... | 2022-11-12 08:34:01 |
| 58 | Se actualizó un Re... | 2022-11-12 08:35:36 |
| 59 | Se actualizó un Re... | 2022-11-12 08:35:36 |
| 60 | Se creo un Registro | 2022-11-12 08:35:36 |

Figure 13: Ejecución Trigger

Review the SQL Script to be Applied on the Database

```

1 UPDATE `tm`.`SUCURSALES` SET `SUCURSAL` = 'd' WHERE (`CODIGO_SUCURSA
2 UPDATE `tm`.`SUCURSALES` SET `SUCURSAL` = 'Universidad Javeriana' WHERE (
3 INSERT INTO `tm`.`SUCURSALES` (`CODIGO_SUCURSAL`, `SUCURSAL`) VALUES
4

```

Figure 14: Ejecución Trigger

4.8 Código SQL + Resultados: Funciones (*Primera entrega*)

```
1 • CREATE DEFINER='root'@'localhost' FUNCTION `DESCRIPCIONCANTIDAD`(CANT INT) RETURNS char(15) CHARSET utf8mb3
2     DETERMINISTIC
3 BEGIN
4     DECLARE DESCRIP CHAR(15);
5     IF CANT<1000 THEN SET DESCRIP = 'CANTIDAD BAJA';
6     ELSEIF (CANT>=1000 AND CANT<=10000) THEN SET DESCRIP = 'CANTIDAD MEDIA';
7     ELSE SET DESCRIP = 'CANTIDAD ALTA';
8     END IF;
9     RETURN (DESCRIP);
10 END
```

Figure 15: Ejecución Funciones

```
1 • CREATE DEFINER='root'@'localhost' FUNCTION `DESCRIPCIONVALOR`(VAL INT) RETURNS char(15) CHARSET utf8m
2     DETERMINISTIC
3 BEGIN
4     DECLARE DESCRIP CHAR(15);
5     IF VAL<10000000 THEN SET DESCRIP = 'INGRESOS BAJOS';
6     ELSEIF (VAL>=10000000 AND VAL<=100000000) THEN SET DESCRIP = 'INGRESOS MEDIOS';
7     ELSE SET DESCRIP = 'INGRESOS ALTOS';
8     END IF;
9     RETURN (DESCRIP);
10 END
```

Figure 16: Ejecución Funciones

```
DSELECT FECHA_AAAAMM, ESTACION, DESC_TIPO_DISPOSITIVO ,
DESCRIPCIONCANTIDAD(SUM(CANTIDAD)) DESCRIP,
SUM(CANTIDAD)
FROM TRANSMILENIOTOTAL
WHERE FECHA_AAAAMM='202203' AND TRIM(ESTACION)=
'SUBA_ _AV. _BOYACA'
GROUP BY FECHA_AAAAMM, ESTACION,
DESC_TIPO_DISPOSITIVO ;

SELECT FECHA_AAAAMM, ESTACION, DESC_TIPO_DISPOSITIVO ,
DESCRIPCIONVALOR(SUM(VALOR)) DESCRIP, SUM(VALOR)
FROM TRANSMILENIOTOTAL
WHERE FECHA_AAAAMM='202203' AND TRIM(ESTACION)=
'SUBA_ _AV. _BOYACA'
GROUP BY FECHA_AAAAMM, ESTACION,
DESC_TIPO_DISPOSITIVO ;
```

| FECHA_AAAAMM | ESTACION | DESC_TIPO_DISPOSITIVO | DESCRIP | SUM(CANTIDAD) |
|--------------|-------------------|---------------------------------|----------------|---------------|
| 202203 | Suba - Av. Boyaca | Terminal de Carga Asistida | CANTIDAD ALTA | 43958 |
| 202203 | SUBA - AV. BOYACA | Dispositivo de Recarga Web (RB) | CANTIDAD BAJA | 97 |
| 202203 | SUBA - AV. BOYACA | Dispositivo de Carga Automática | CANTIDAD MEDIA | 4160 |
| 202203 | Suba - Av. Boyaca | Dispositivo de Carga Móvil | CANTIDAD MEDIA | 2621 |

Figure 17: Resultado Ejecución Funciones

4.9 Código SQL + Resultados: procedimientos almacenados (Primera entrega)

```

DROP PROCEDURE SALIDAZONAS;
CREATE PROCEDURE SALIDAZONAS (IN PERIODOINI CHAR(6) ,
IN PERIODOFIN CHAR(6) ,IN Z CHAR(80))
    SELECT
        FECHA_AAAAMM,ESTACION,CUADRANTE,ZONA,
        DESC_TIPO_DISPOSITIVO,tipo_agente ,
        SUM(CANTIDAD) CANTIDAD,SUM(VALOR) VALOR
    FROM TRANSMILENIOTOTAL
    WHERE FECHA_AAAAMM>=PERIODOINI AND
    FECHA_AAAAMM<=PERIODOFIN AND ZONA
    LIKE CONCAT( '%',Z, '%' )
    GROUP BY FECHA_AAAAMM,ESTACION,CUADRANTE,
    ZONA,DESC_TIPO_DISPOSITIVO,tipo_agente ;

call SALIDAZONAS( '202202' , '202203' ,
'Terminal--Heroes' );

DROP PROCEDURE SALIDAESTACIONES;
CREATE PROCEDURE SALIDAESTACIONES
(IN PERIODOINI CHAR(6) ,IN PERIODOFIN CHAR(6) ,IN E CHAR(80))
    SELECT
        FECHA_AAAAMM,ESTACION,CUADRANTE,ZONA,
        DESC_TIPO_DISPOSITIVO,tipo_agente ,
        SUM(CANTIDAD) CANTIDAD,
        SUM(VALOR) VALOR
    FROM TRANSMILENIOTOTAL
    WHERE FECHA_AAAAMM>=PERIODOINI AND
    FECHA_AAAAMM<=PERIODOFIN AND ESTACION LIKE
    CONCAT( '%',E, '%' )
    GROUP BY FECHA_AAAAMM,ESTACION,CUADRANTE,
    ZONA,DESC_TIPO_DISPOSITIVO,tipo_agente ;

call SALIDAESTACIONES( '202202' , '202203' ,
'Suba--Calle_95' );

DROP PROCEDURE SALIDACUADRANTES;

```

```

CREATE PROCEDURE SALIDACUADRANTES
(IN PERIODOINI CHAR(6),IN PERIODOFIN CHAR(6)
,IN C CHAR(80))
    SELECT
    FECHA_AAAAMM,ESTACION,CUADRANTE,ZONA,
    DESC_TIPO_DISPOSITIVO,tipo_agente ,
    SUM(CANTIDAD) CANTIDAD,SUM(VALOR)
    VALOR
    FROM TRANSMILENIOTOTAL
    WHERE FECHA_AAAAMM>=PERIODOINI AND
    FECHA_AAAAMM<=PERIODOFIN AND CUADRANTE
    LIKE CONCAT( '%',C, '%')
    GROUP BY FECHA_AAAAMM,ESTACION,CUADRANTE,
    ZONA,DESC_TIPO_DISPOSITIVO,tipo_agente ;

call SALIDACUADRANTES( '202202', '202203',
'Quinta_Paredes_-_Ricaurte');

```

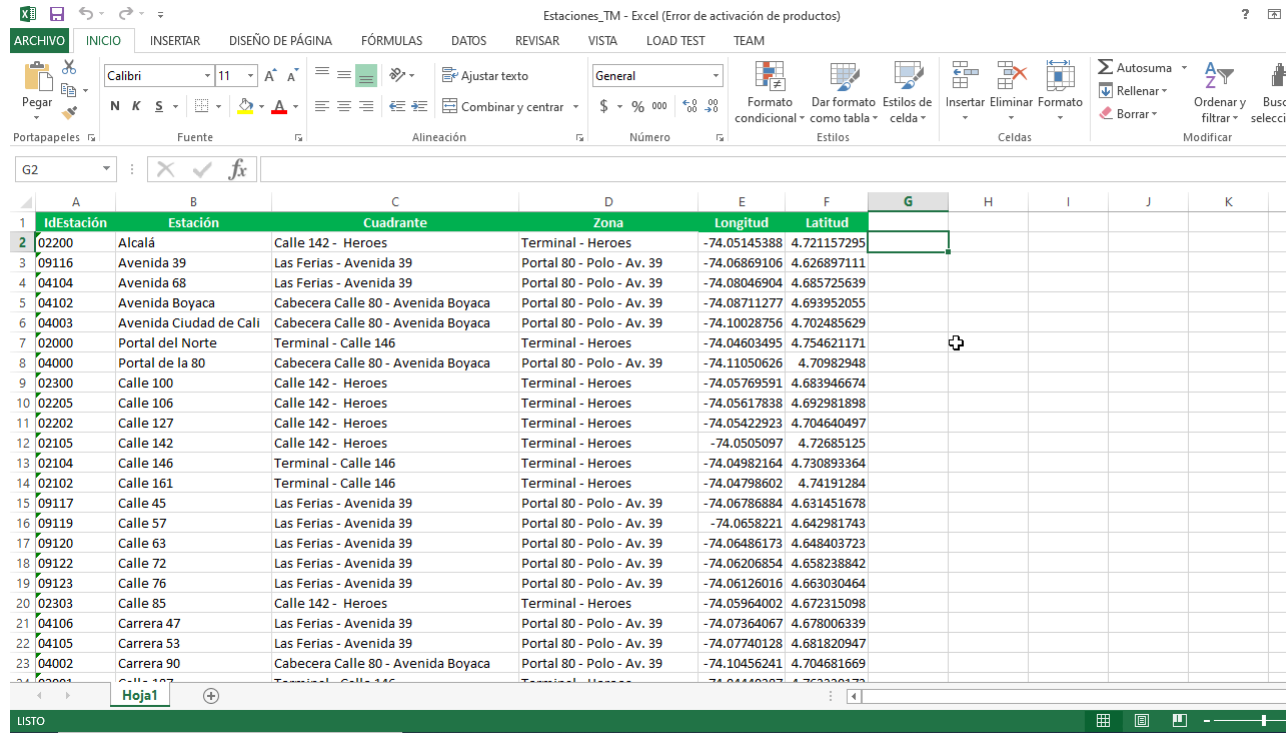
| FECHA_AAAAMM | ESTACION | CUADRANTE | ZONA | DESC_TIPO_DISPOSITIVO | tipo_agente | CANTIDAD | VALOR |
|--------------|--------------------------|----------------------|-------------------|---------------------------------|------------------|----------|------------|
| 202202 | Alcalá | Calle 142 - Heroes | Terminal - Heroes | Dispositivo de Carga Automática | Estación Troncal | 20964 | 150040225 |
| 202202 | Alcalá | Calle 142 - Heroes | Terminal - Heroes | Dispositivo de Carga Móvil | Estación Troncal | 577 | 4786450 |
| 202202 | Alcalá | Calle 142 - Heroes | Terminal - Heroes | Dispositivo de Recarga Web (RB) | Estación Troncal | 160 | 3910250 |
| 202202 | Alcalá | Calle 142 - Heroes | Terminal - Heroes | Terminal de Carga Asistida | Estación Troncal | 143003 | 1272685590 |
| 202202 | Cabecera Autopista Norte | Terminal - Calle 146 | Terminal - Heroes | Dispositivo de Carga Automática | Estación Troncal | 175068 | 970296780 |
| 202202 | Cabecera Autopista Norte | Terminal - Calle 146 | Terminal - Heroes | Dispositivo de Carga Móvil | Estación Troncal | 333 | 2469550 |
| 202202 | Cabecera Autopista Norte | Terminal - Calle 146 | Terminal - Heroes | Dispositivo de Recarga Web (RB) | Estación Troncal | 618 | 13924750 |
| 202202 | Cabecera Autopista Norte | Terminal - Calle 146 | Terminal - Heroes | Terminal de Carga Asistida | Estación Troncal | 461798 | 3609516530 |
| 202202 | Calle 100 | Calle 142 - Heroes | Terminal - Heroes | Dispositivo de Carga Automática | Estación Troncal | 14350 | 120986840 |
| 202202 | Calle 100 | Calle 142 - Heroes | Terminal - Heroes | Dispositivo de Carga Móvil | Estación Troncal | 137 | 1104850 |
| 202202 | Calle 100 | Calle 142 - Heroes | Terminal - Heroes | Dispositivo de Recarga Web (RB) | Estación Troncal | 477 | 10738050 |
| 202202 | Calle 100 | Calle 142 - Heroes | Terminal - Heroes | Terminal de Carga Asistida | Estación Troncal | 165464 | 1659436130 |
| 202202 | Calle 106 | Calle 142 - Heroes | Terminal - Heroes | Dispositivo de Carga Automática | Estación Troncal | 165464 | 86721255 |
| 202202 | Calle 106 | Calle 142 - Heroes | Terminal - Heroes | Dispositivo de Carga Móvil | Estación Troncal | 627 | 4921500 |
| 202202 | Calle 106 | Calle 142 - Heroes | Terminal - Heroes | Dispositivo de Recarga Web (RB) | Estación Troncal | 110 | 2792700 |

Figure 18: Resultado Ejecución Procedimiento Almacenado SALIDAZONAS

5 Bases de Datos No-SQL (Segunda entrega)

5.1 Diagrama Bases de Datos No-SQL (Segunda entrega)

Partimos de un archivo Excel que se generó desde la base de datos de MySQL sobre las estaciones de Transmilenio y que consideramos sería un buen punto de partida. Este archivo contiene la siguiente información:



| | A | B | C | D | E | F | G | H | I | J | K |
|----|------------|------------------------|------------------------------------|---------------------------|--------------|-------------|---|---|---|---|---|
| | IdEstación | Estación | Cuadrante | Zona | Longitud | Latitud | | | | | |
| 2 | 02200 | Alcalá | Calle 142 - Heroes | Terminal - Heroes | -74.05145388 | 4.721157295 | | | | | |
| 3 | 09116 | Avenida 39 | Las Ferias - Avenida 39 | Portal 80 - Polo - Av. 39 | -74.06869106 | 4.626897111 | | | | | |
| 4 | 04104 | Avenida 68 | Las Ferias - Avenida 39 | Portal 80 - Polo - Av. 39 | -74.08046904 | 4.685725639 | | | | | |
| 5 | 04102 | Avenida Boyaca | Cabecera Calle 80 - Avenida Boyaca | Portal 80 - Polo - Av. 39 | -74.08711277 | 4.693952055 | | | | | |
| 6 | 04003 | Avenida Ciudad de Cali | Cabecera Calle 80 - Avenida Boyaca | Portal 80 - Polo - Av. 39 | -74.10028756 | 4.702485629 | | | | | |
| 7 | 02000 | Portal del Norte | Terminal - Calle 146 | Terminal - Heroes | -74.04603495 | 4.754621171 | | | | | |
| 8 | 04000 | Portal de la 80 | Cabecera Calle 80 - Avenida Boyaca | Portal 80 - Polo - Av. 39 | -74.11050626 | 4.70982948 | | | | | |
| 9 | 02300 | Calle 100 | Calle 142 - Heroes | Terminal - Heroes | -74.05769591 | 4.683946674 | | | | | |
| 10 | 02205 | Calle 106 | Calle 142 - Heroes | Terminal - Heroes | -74.05617838 | 4.692981898 | | | | | |
| 11 | 02202 | Calle 127 | Calle 142 - Heroes | Terminal - Heroes | -74.05422923 | 4.704640497 | | | | | |
| 12 | 02105 | Calle 142 | Calle 142 - Heroes | Terminal - Heroes | -74.0505097 | 4.72685125 | | | | | |
| 13 | 02104 | Calle 146 | Terminal - Calle 146 | Terminal - Heroes | -74.04982164 | 4.730893364 | | | | | |
| 14 | 02102 | Calle 161 | Terminal - Calle 146 | Terminal - Heroes | -74.04798602 | 4.74191284 | | | | | |
| 15 | 09117 | Calle 45 | Las Ferias - Avenida 39 | Portal 80 - Polo - Av. 39 | -74.06786884 | 4.631451678 | | | | | |
| 16 | 09119 | Calle 57 | Las Ferias - Avenida 39 | Portal 80 - Polo - Av. 39 | -74.0658221 | 4.642981743 | | | | | |
| 17 | 09120 | Calle 63 | Las Ferias - Avenida 39 | Portal 80 - Polo - Av. 39 | -74.06486173 | 4.648403723 | | | | | |
| 18 | 09122 | Calle 72 | Las Ferias - Avenida 39 | Portal 80 - Polo - Av. 39 | -74.06206854 | 4.658238842 | | | | | |
| 19 | 09123 | Calle 76 | Las Ferias - Avenida 39 | Portal 80 - Polo - Av. 39 | -74.06126016 | 4.663030464 | | | | | |
| 20 | 02303 | Calle 85 | Calle 142 - Heroes | Terminal - Heroes | -74.05964002 | 4.672315098 | | | | | |
| 21 | 04106 | Carrera 47 | Las Ferias - Avenida 39 | Portal 80 - Polo - Av. 39 | -74.07364067 | 4.678006339 | | | | | |
| 22 | 04105 | Carrera 53 | Las Ferias - Avenida 39 | Portal 80 - Polo - Av. 39 | -74.07740128 | 4.681820947 | | | | | |
| 23 | 04002 | Carrera 90 | Cabecera Calle 80 - Avenida Boyaca | Portal 80 - Polo - Av. 39 | -74.10456241 | 4.704681669 | | | | | |

Figure 19: Archivo Base Original

Como paso adicional antes de cargar la data en MongoDB Atlas se procedió a crear un script en Nodejs para convertir el archivo Excel en JSON:

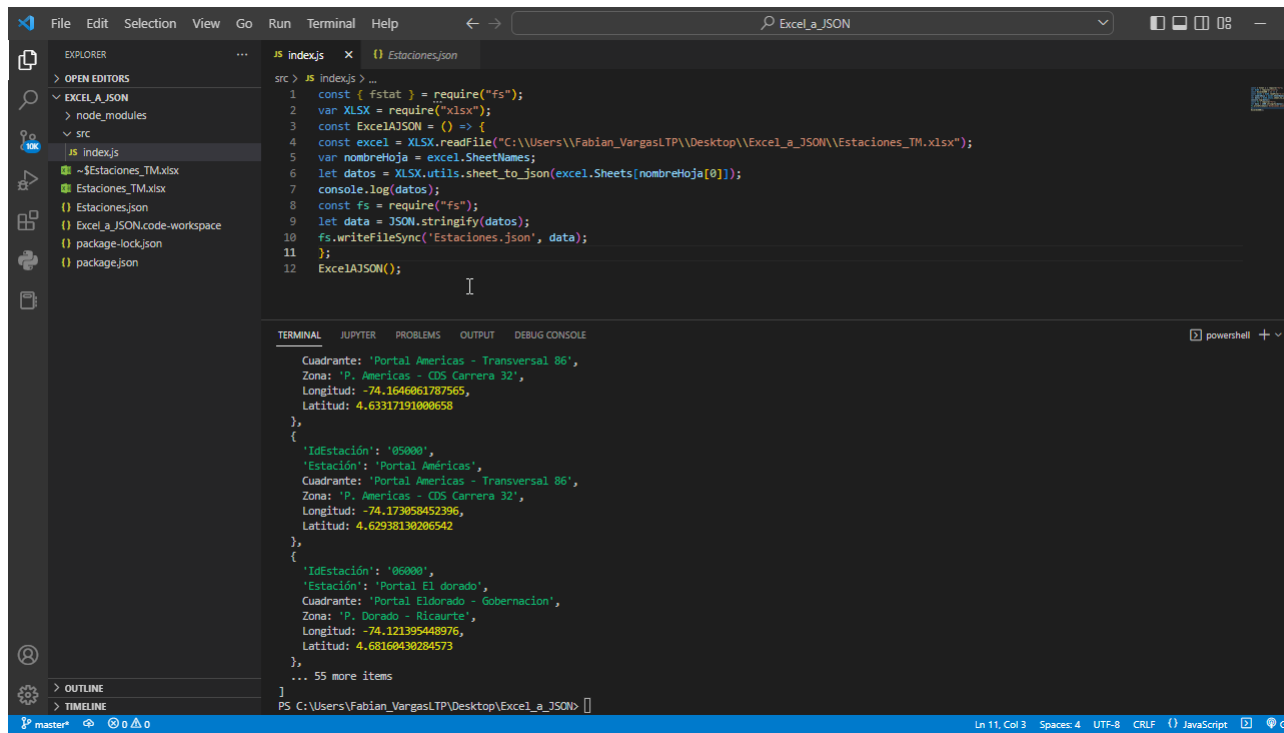


Figure 20: Código NodeJS para generar JSON

```

var XLSX = require("xlsx");
const ExcelAJSON = () => {
const excel = XLSX.readFile("C:\\Users\\Fabian_VargasLTP
\\Desktop\\Excel_a_JSON\\Estaciones_TM.xlsx");
var nombreHoja = excel.SheetNames;
let datos = XLSX.utils.sheet_to_json(excel.
[nombreHoja[0]]);
console.log(datos);
const fs = require("fs");
let data = JSON.stringify(datos);
fs.writeFileSync('Estaciones.json', data);
};
ExcelAJSON();

```

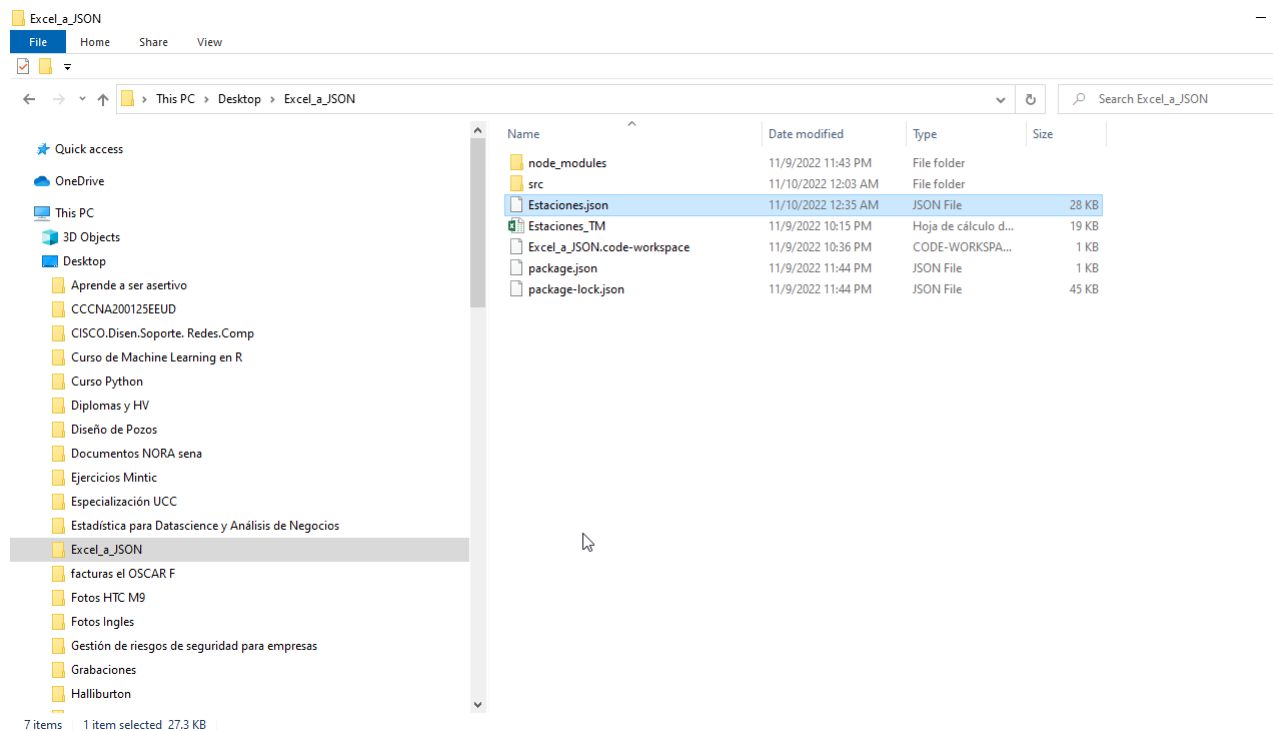


Figure 21: Archivo JSON generado por el código

El código JSON generado por el script fue (sólo se presenta una parte de 156 documentos):

```
[{"IdEstacion": "02200", "Estacion": "Alcal ",
"Cuadrante": "Calle_142_-_Heroes",
"Zona": "Terminal_-_Heroes",
"Longitud": -74.0514538784185,
"Latitud": 4.72115729506492},
{"IdEstacion": "09116", "Estacion": "Avenida_39",
"Cuadrante": "Las_Ferias_-_Avenida_39",
"Zona": "Portal_80_-_Polo_-_Av._39",
"Longitud": -74.068691056891,
"Latitud": 4.62689711127706},
{"IdEstacion": "04104", "Estacion": "Avenida_68",
"Cuadrante": "Las_Ferias_-_Avenida_39",
"Zona": "Portal_80_-_Polo_-_Av._39",
"Longitud": -74.0804690428983,
"Latitud": 4.68572563850324},
```

Luego de generar el JSON con la data a cargar en los documentos de

Mongodb Atlas, se procede a importar el archivo desde Compass:

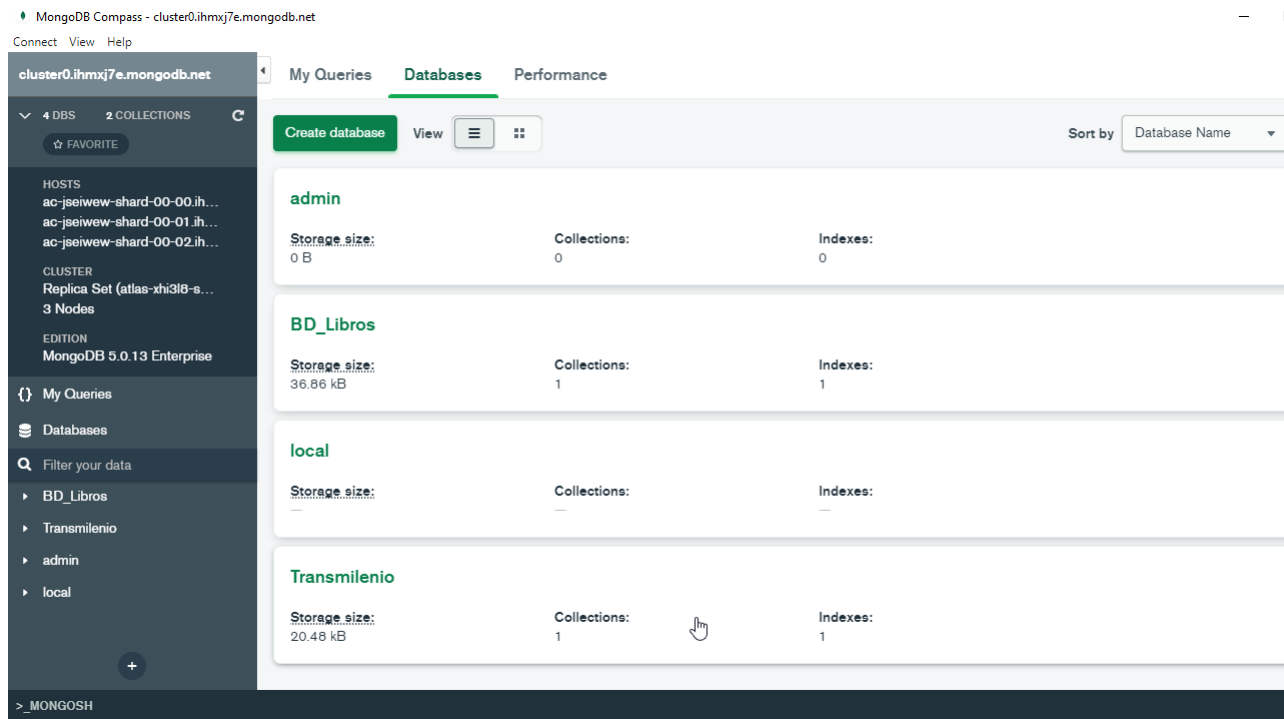


Figure 22: Interfaz de Compass instalado local

Se valida que no tenga información la base de datos:

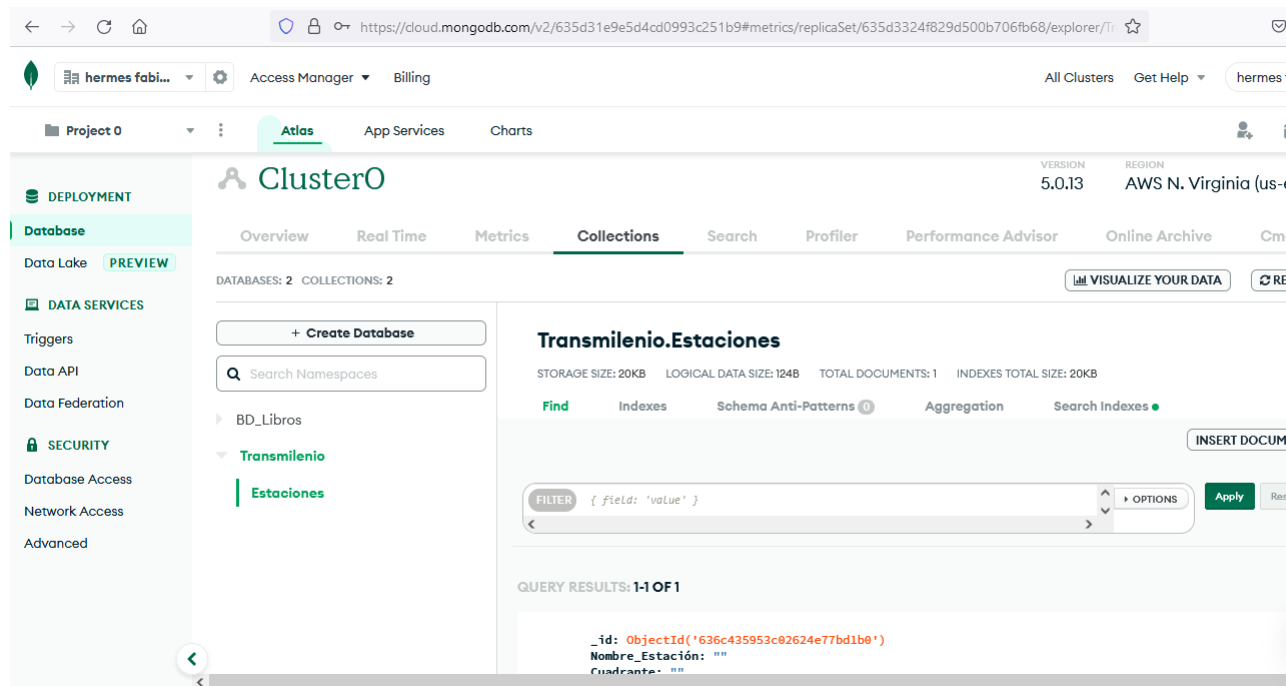


Figure 23: Vista en MongoDB Colección Estaciones

Se procede a cargar el archivo JSON desde COMPASS:

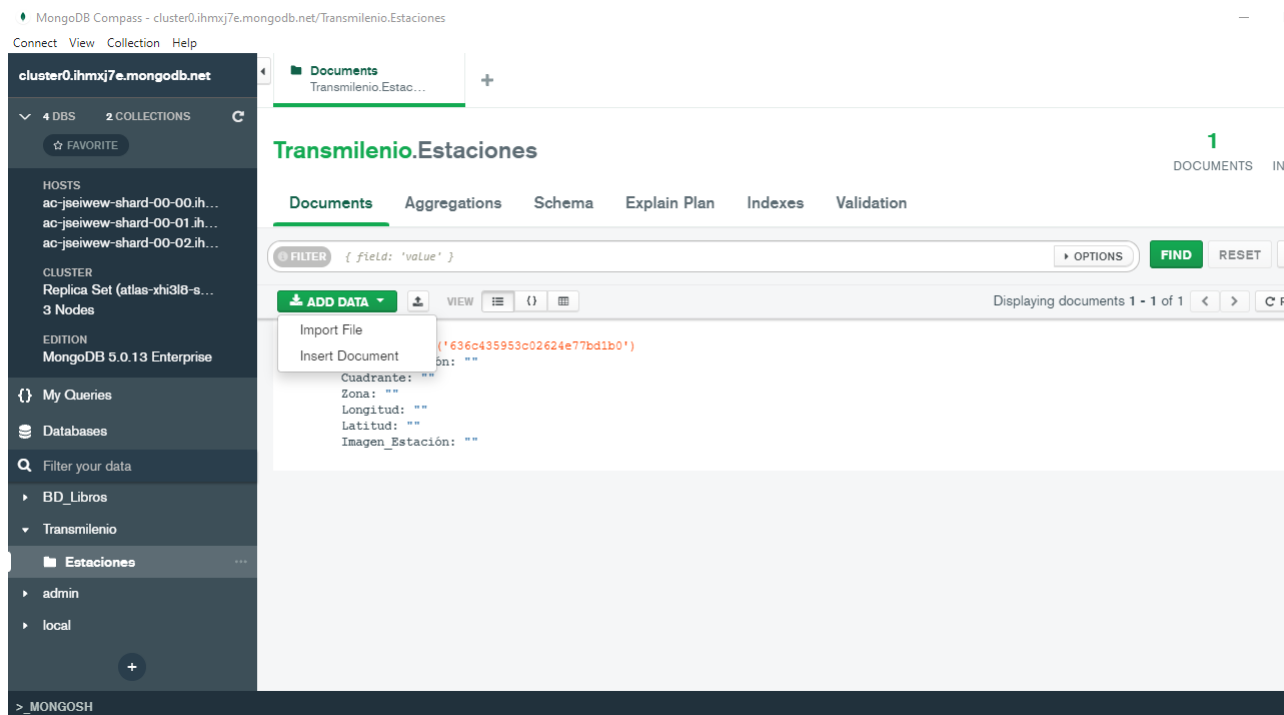


Figure 24: Carga de archivo JSON desde Compass

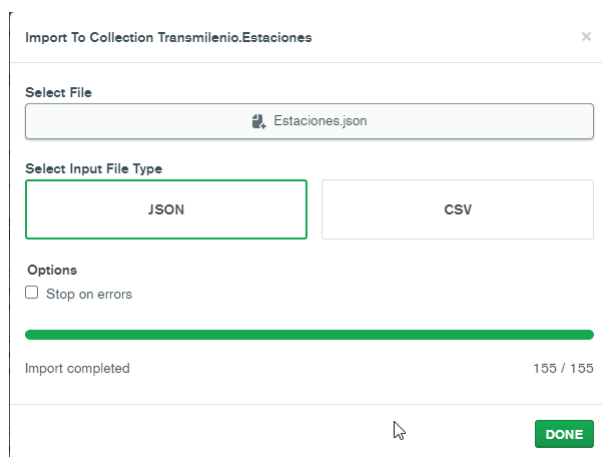


Figure 25: Carga de archivo JSON desde Compass

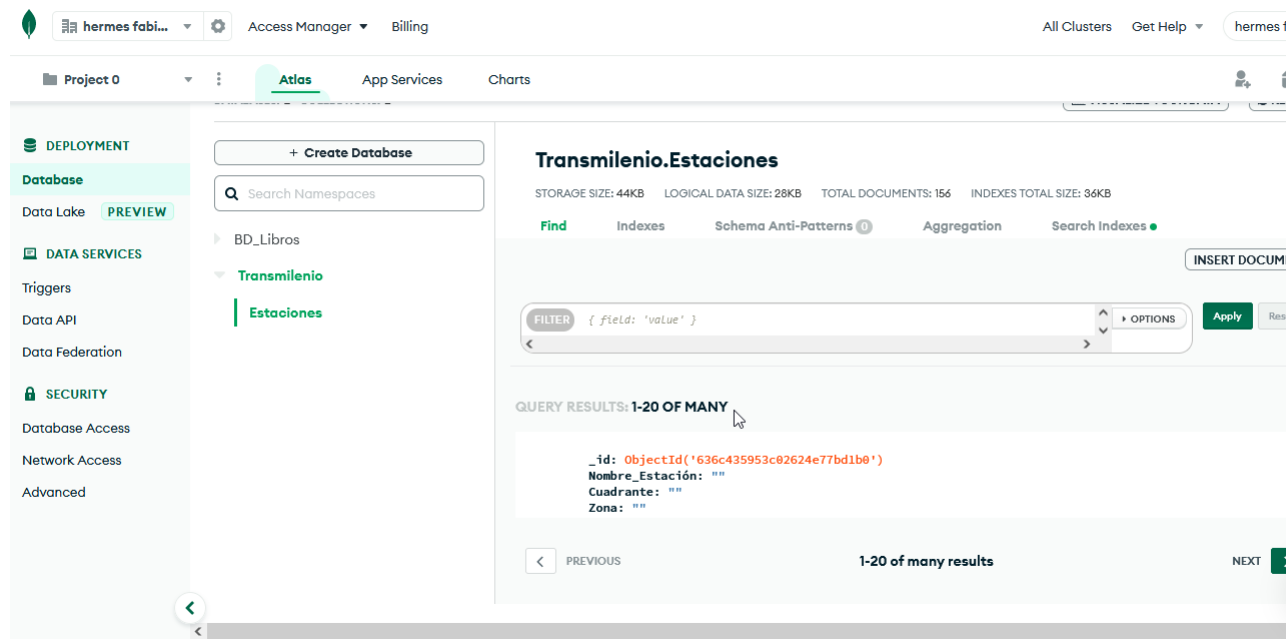


Figure 26: Confirmación de carga de Archivos sobre MongoDB Atlas

Para la carga de información que no sea tipo String o de tipo numérica sino Binaria como lo son imágenes se procede a crear una colección llamada Imágenes:

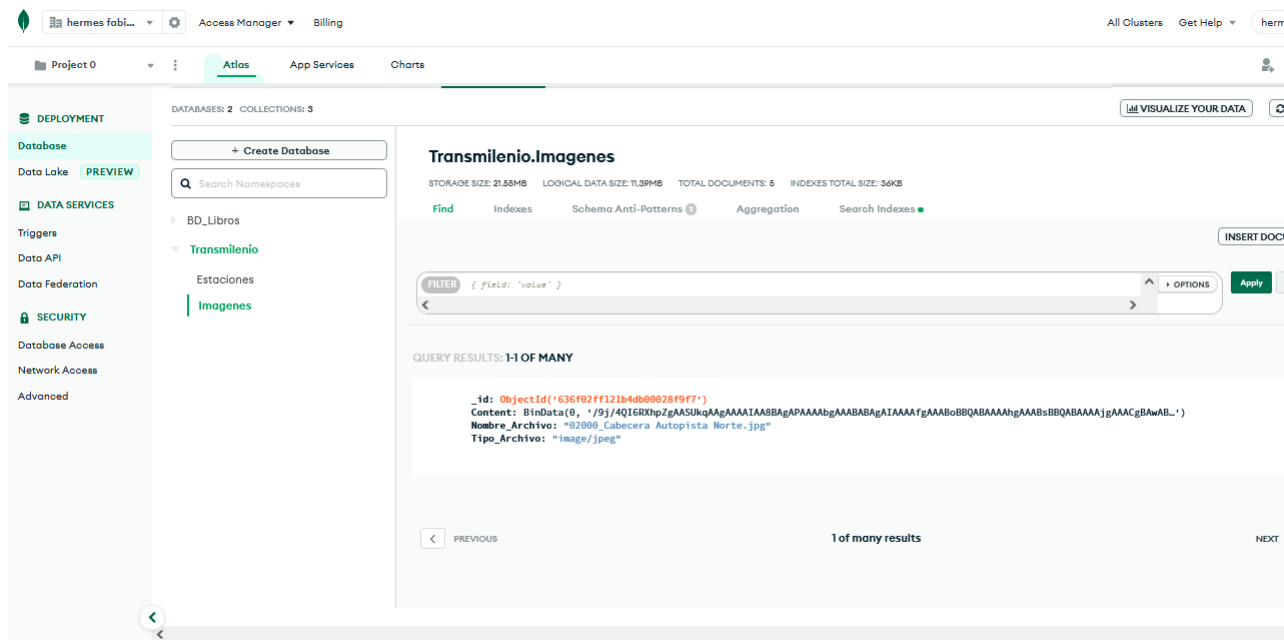


Figure 27: Visualización nueva Colección en MongoDB

Para ingresar y almacenar las imágenes dentro de la base de datos de MongoDB se usó un programa que se desarrolló en C a través del cual se estableció la conexión a la base de datos y a través de los métodos POST y GET se pudo almacenar o extraer la imagen hacia y desde Mongo:

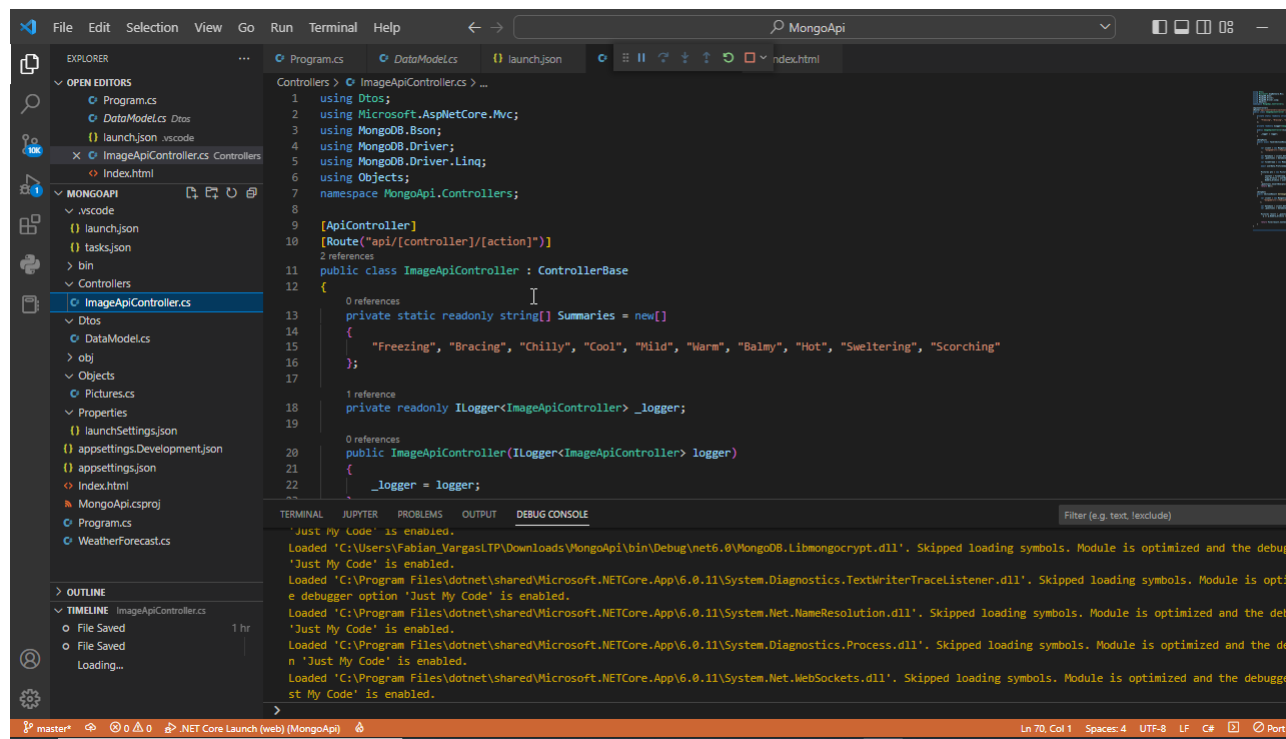


Figure 28: Código C Sharp para almacenamiento y consulta de imágenes

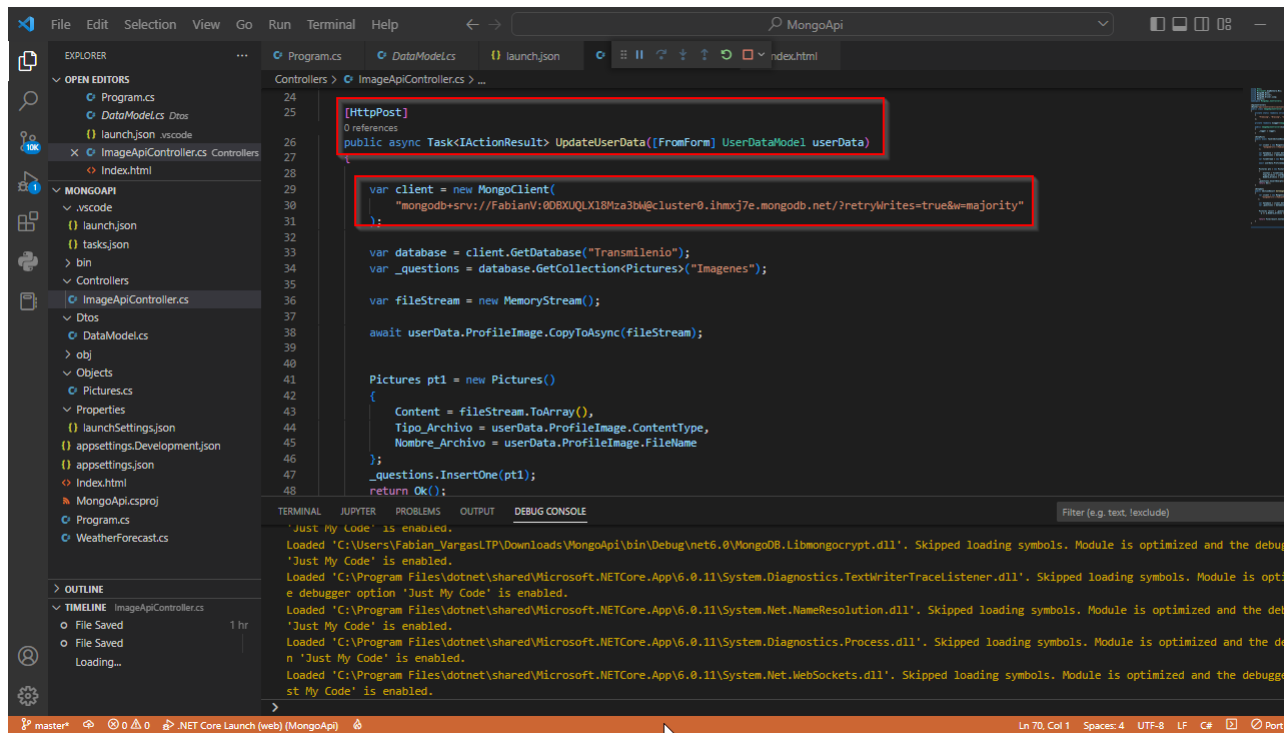


Figure 29: Método POST para almacenar

```

[HttpPost]
public async Task<IActionResult> UpdateUserData
([FromForm] UserDataModel userData)
{

    var client = new MongoClient(
        "mongodb+srv://FabianV:0DBXUQLXl8Mza3bW@
        cluster0.ihmxj7e.mongodb.net/
        ?retryWrites=true&w=majority"
    );

    var database = client.GetDatabase("Transmilenio");
    var _questions = database.GetCollection<Pictures>
("Imagenes");

    var fileStream = new MemoryStream();

    await userData.ProfileImage.CopyToAsync
(fileStream);
}

```

```
Pictures pt1 = new Pictures()
{
    Content = fileStream.ToArray(),
    Tipo_Archivo = userData.ProfileImage.
    ContentType,
    Nombre_Archivo = userData.ProfileImage.
    FileName
};
_questions.InsertOne(pt1);
return Ok();
}
```

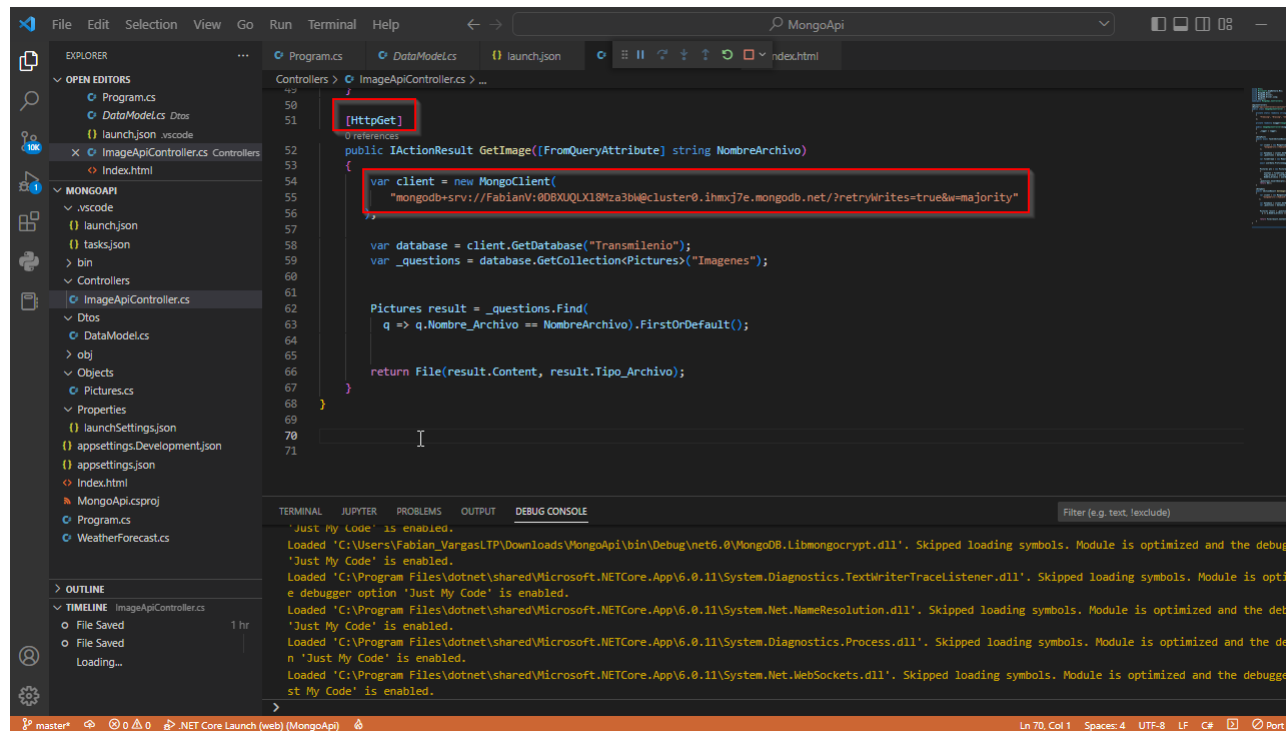


Figure 30: Método GET para obtener imagen

```
[HttpGet]
public IActionResult GetImage([FromQueryAttribute]
string NombreArchivo)
{
```



```

        var client = new MongoClient(
            "mongodb+srv://FabianV:0DBXUQLXl8Mza3bW@
            .....cluster0.ihmxj7e.mongodb.net/
            .....?retryWrites=true&w=majority"
        );

        var database = client.GetDatabase("Transmilenio");
        var _questions = database.GetCollection<Pictures>
            ("Imagenes");

        Pictures result = _questions.Find(
            q => q.Nombre_Archivo == NombreArchivo).
            FirstOrDefault();

        return File(result.Content, result.Tipo_Archivo);
    }

```

En la base de datos la información que se envió fue el nombre del archivo, el tipo de archivo y puntualmente el binario de la imagen generado a través del formato BSON tomando como base la clase que se configuró en el programa:

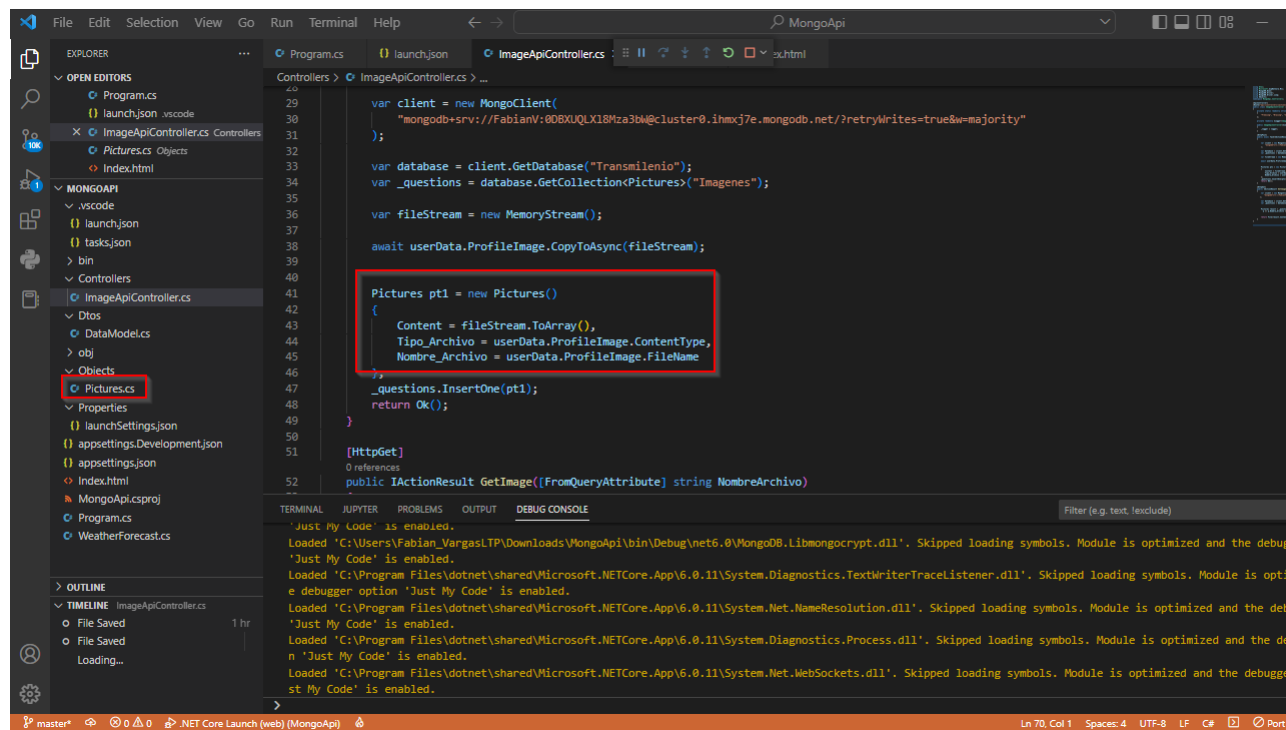


Figure 31: Estructura de la data a almacenar en la BD

ImageApiController.cs

```
using Dtos;
using Microsoft.AspNetCore.Mvc;
using MongoDB.Bson;
using MongoDB.Driver;
using MongoDB.Driver.Linq;
using Objects;
namespace MongoApi.Controllers;

[ApiController]
[Route("api/[controller]/[action]")]
public class ImageApiController : ControllerBase
{
    private static readonly string[] Summaries = new[]
    {
        "Freezing", "Bracing", "Chilly", "Cool", "Mild",
        "Wam", "Balmy", "Hot", "Sweltering", "Scorching"
    };
}
```

```

private readonly ILogger<ImageApiController> _logger;

public ImageApiController(ILogger<ImageApiController>
logger)
{
    _logger = logger;
}

[HttpPost]
public async Task<IActionResult> UpdateUserData
([FromForm] UserDataModel userData)
{
    var client = new MongoClient(
        "mongodb+srv://FabianV:0DBXUQLXl8Mza3bW@
.....cluster0.ihmxj7e.mongodb.net/
.....?retryWrites=true&w=majority"
    );

    var database = client.GetDatabase("Transmilenio");
    var _questions = database.GetCollection<Pictures>
("Imagenes");

    var fileStream = new MemoryStream();

    await userData.ProfileImage.CopyToAsync(fileStream);

    Pictures pt1 = new Pictures()
    {
        Content = fileStream.ToArray(),
        Tipo_Archivo = userData.ProfileImage.ContentType,
        Nombre_Archivo = userData.ProfileImage.FileName
    };
    _questions.InsertOne(pt1);
    return Ok();
}

[HttpGet]
public IActionResult GetImage([FromQueryAttribute]
string NombreArchivo)
{
    var client = new MongoClient(
        "mongodb+srv://FabianV:0DBXUQLXl8Mza3bW@
.....cluster0.ihmxj7e.mongodb.net/
.....?retryWrites=true&w=majority"
    );

```

```

    );

    var database = client.GetDatabase("Transmilenio");
    var _questions = database.GetCollection<Pictures>
("Imagenes");

    Pictures result = _questions.Find(
        q => q.Nombre_Archivo == NombreArchivo).
        FirstOrDefault();

    return File(result.Content, result.Tipo_Archivo);
}
}

```

Picture.cs

```

using MongoDB.Bson;
using MongoDB.Bson.Serialization.Attributes;

namespace Objects
{
    public class Pictures
    {
        [BsonId]
        [BsonRepresentation(BsonType.ObjectId)]
        public string Id { get; set; }
        public byte[] Content { get; set; }
        public string Nombre_Archivo { get; set; }
        public string Tipo_Archivo { get; set; }
    }
}

```

Para llamar los métodos de esta API se usó el programa POSTMAN de la siguiente manera:

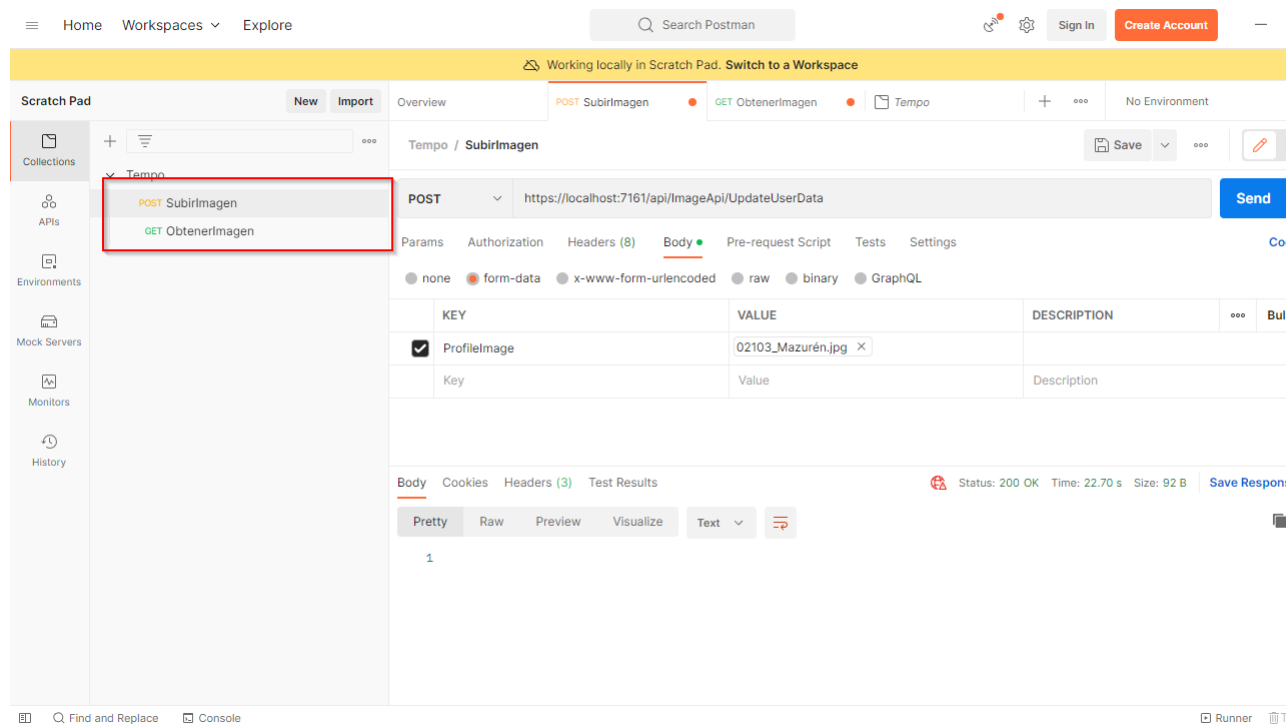


Figure 32: Interfaz POSTMAN

Para almacenar una imagen se cargó la respectiva imagen en el método Post llamado “SubirImagen” de la siguiente forma:

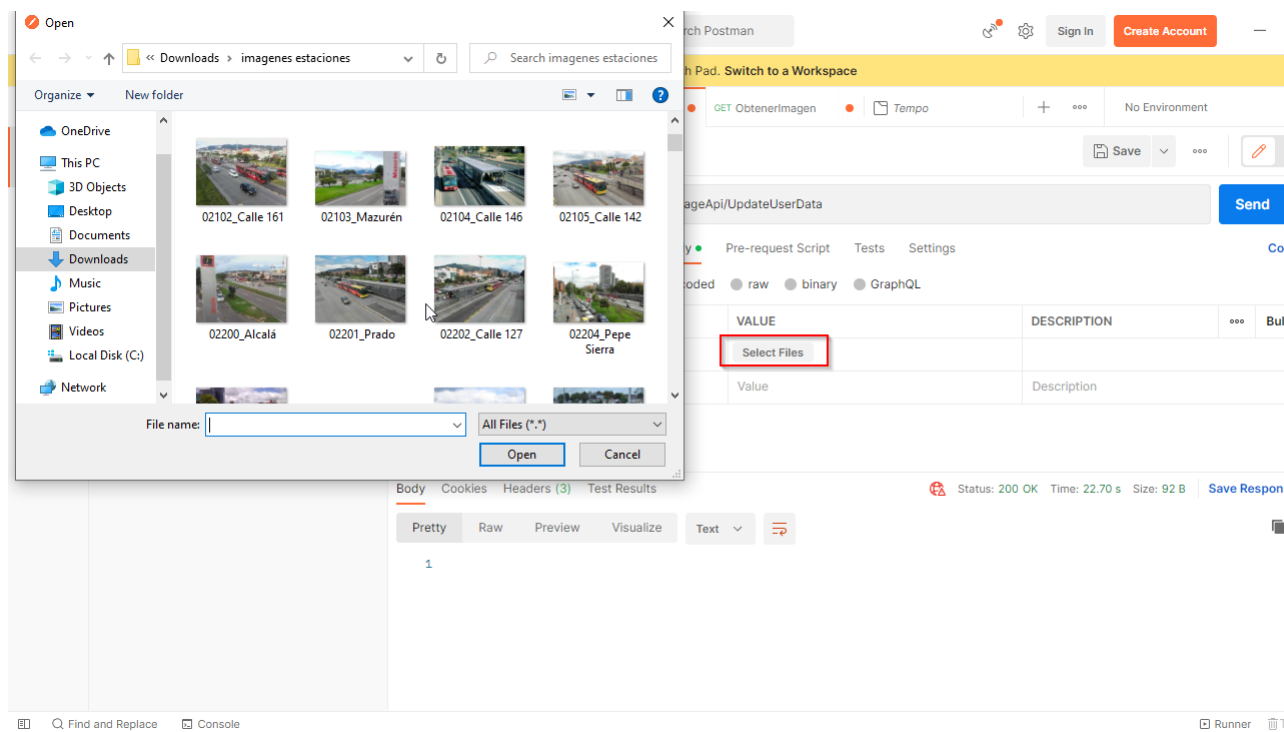


Figure 33: Seleccionar la imagen

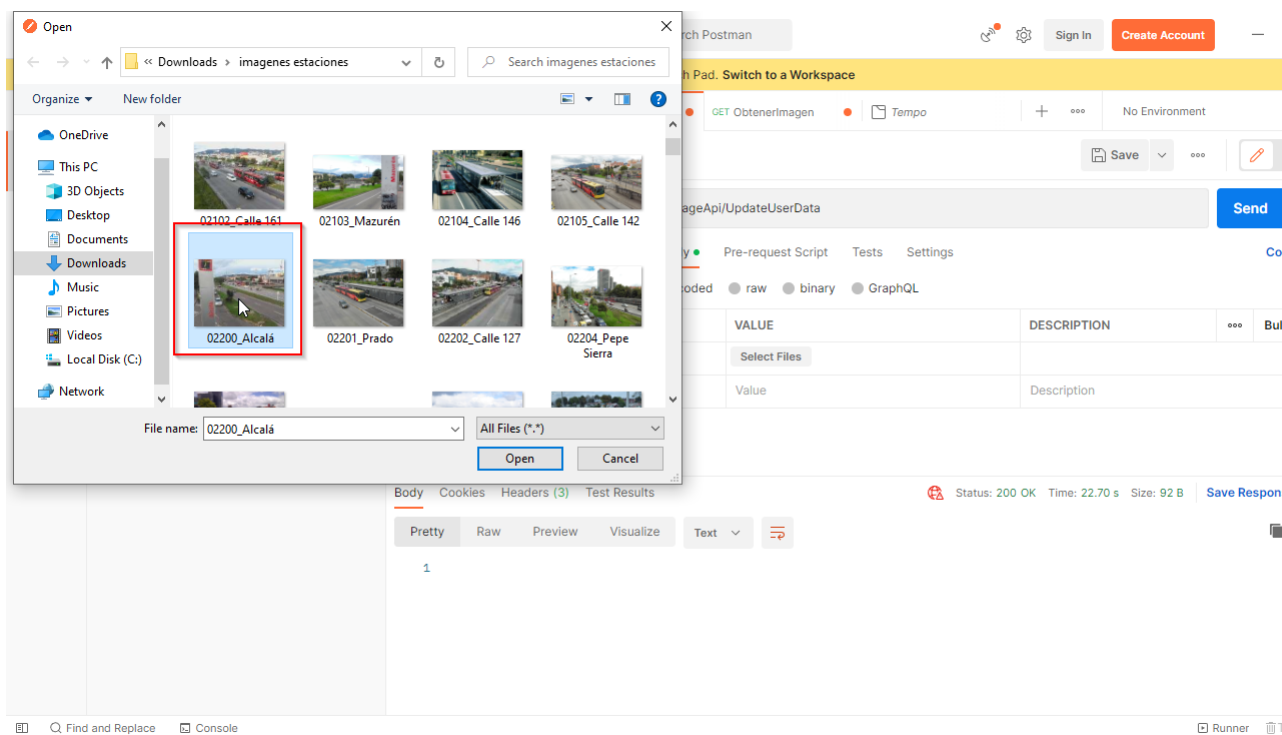


Figure 34: Seleccionar la imagen

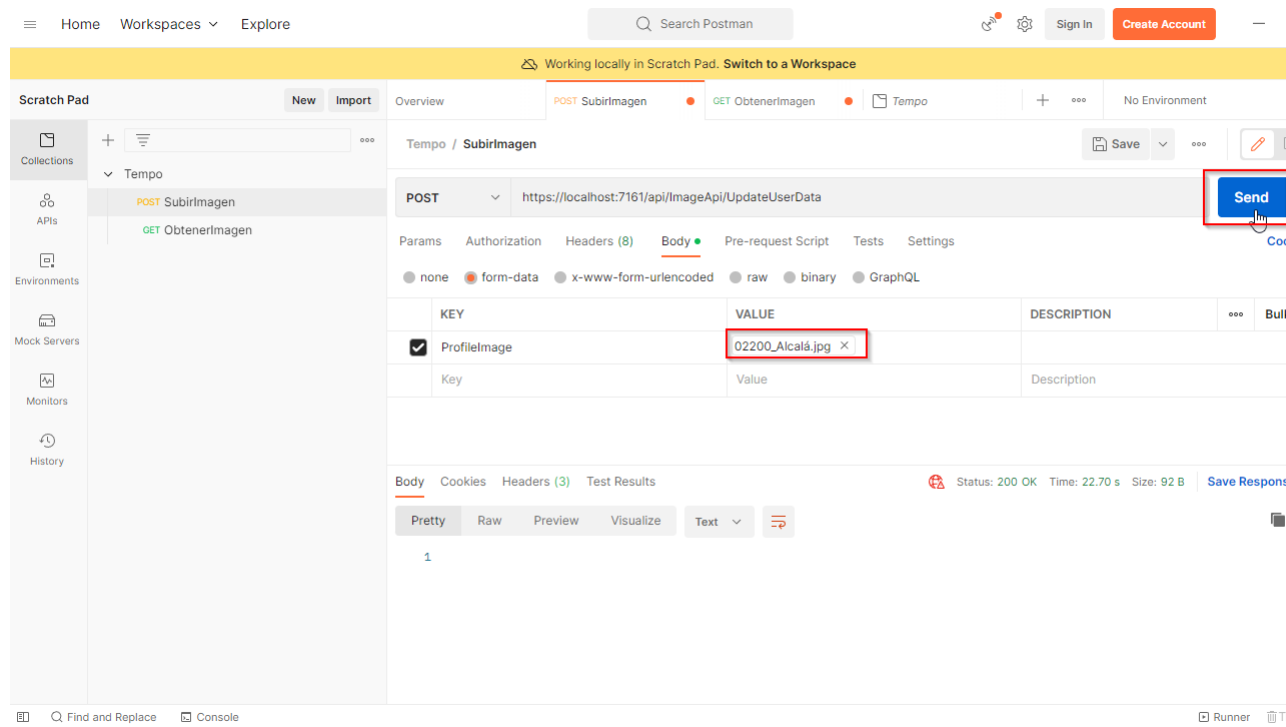


Figure 35: Ejecución método POST para envío de la imagen a MongoDB

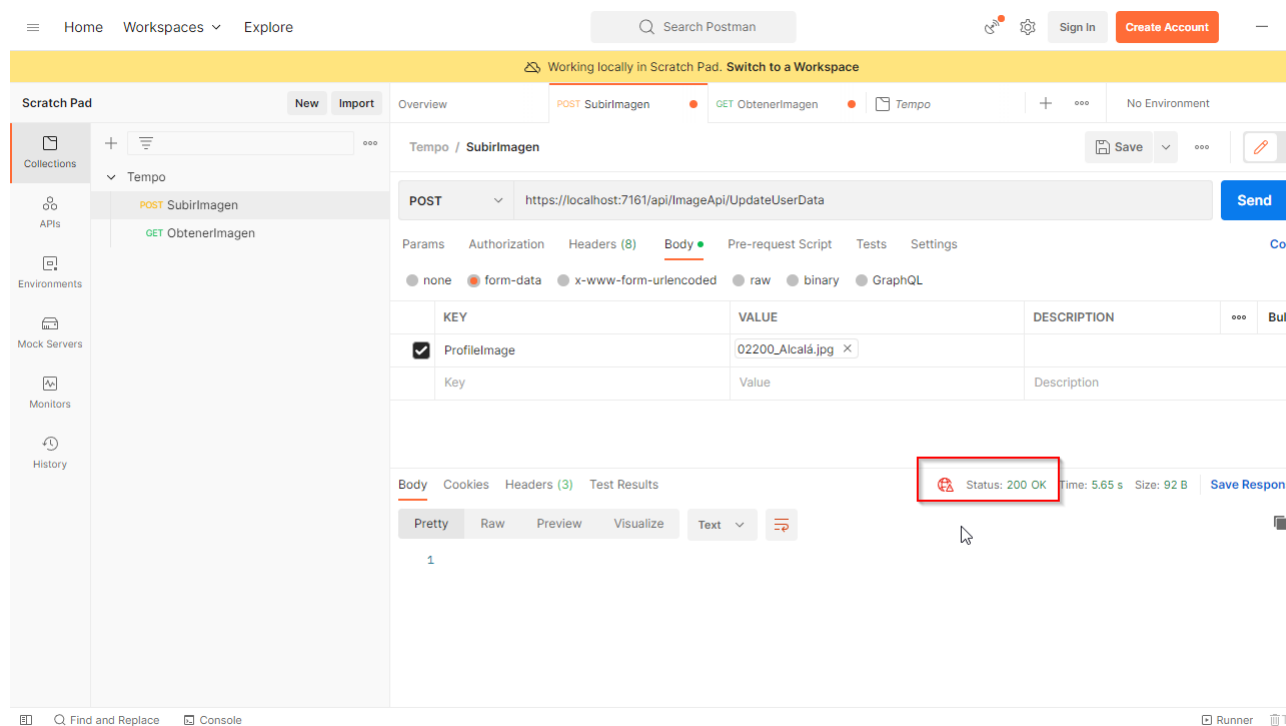


Figure 36: Respuesta Exitosa de la llamada del método

Luego se valida que haya quedado almacenada en la colección:

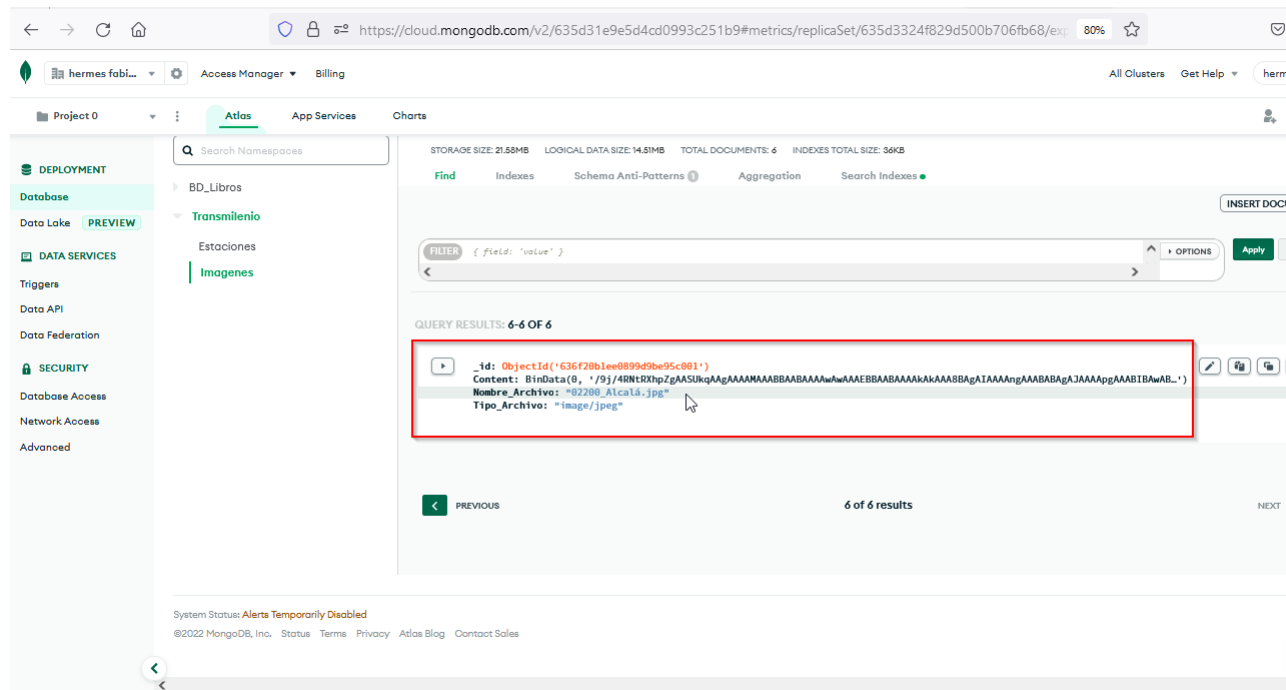


Figure 37: Validación del documento de la imagen cargada en Mongo

Finalmente, se procede a visualizar la imagen que quedó almacenada en la base de datos a través del método GET “ObtenerImagen”:

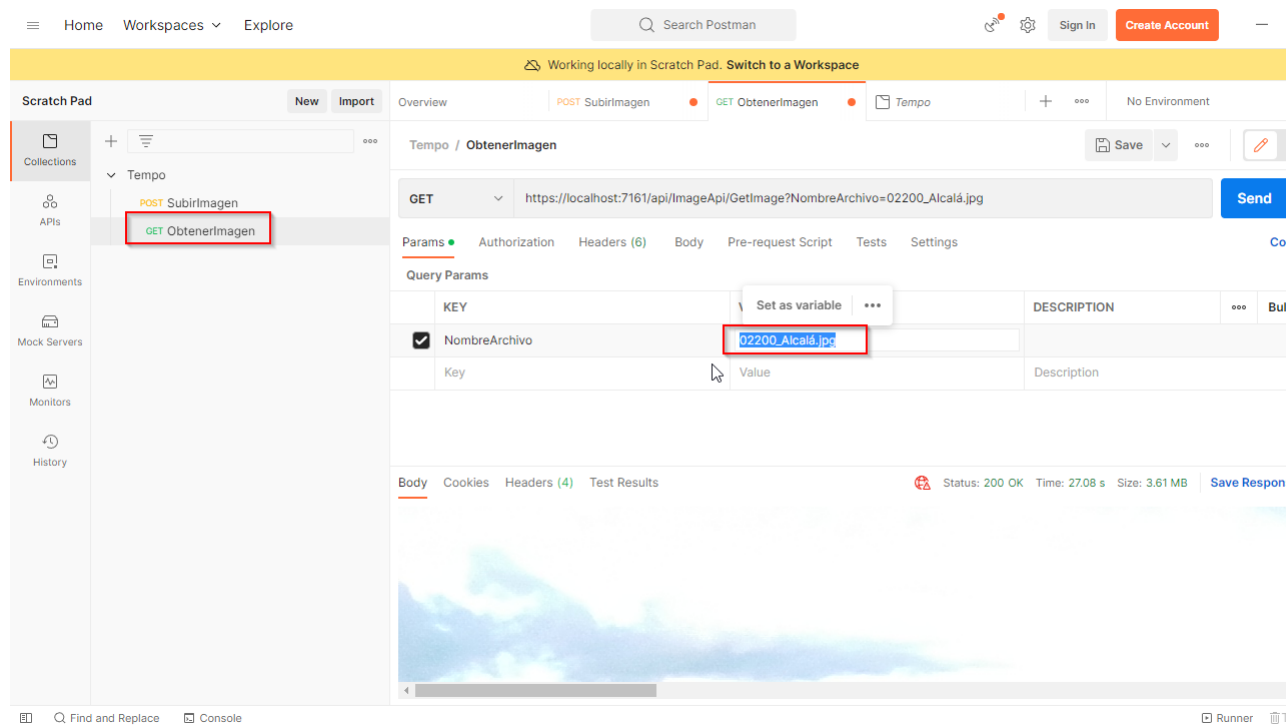


Figure 38: Llamado método GET en Postman para confirmar imagen cargada

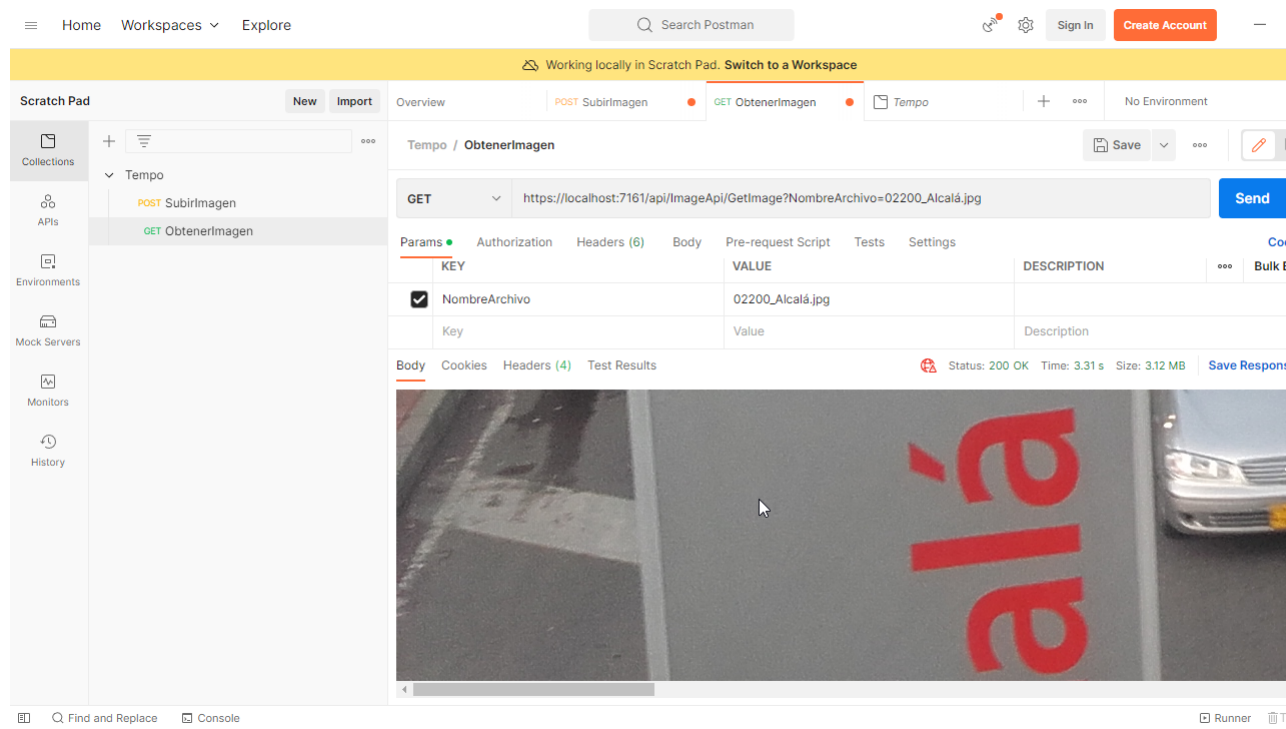


Figure 39: Llamado método GET en Postman para confirmar imagen cargada

5.2 SMBD utilizado para la Base de Datos No-SQL (*Segunda entrega*)

El sistema gestor de base de datos usado para ejecutar la parte de bases de datos no relacionales (NoSQL) fue MongoDB, más específicamente MongoDB Atlas. Este es un servicio de Cloud Database (o Base de Datos en la Nube), que te permite crear y administrar tu BBDD Mongo desde cualquier lugar del mundo, a través de su plataforma. Además, MongoDB Atlas no solo está orientado a ser accesible desde el navegador, sino que, fue desarrollado con el objetivo de aliviar el trabajo de los desarrolladores, al quitarles la necesidad de instalar y administrar entornos de BBDD, los que a veces pueden ser lentos y muy complejos en su configuración. Para nuestro trabajo se creó un Cluster y dentro del cluster se creó la BD llamada Transmilenio:

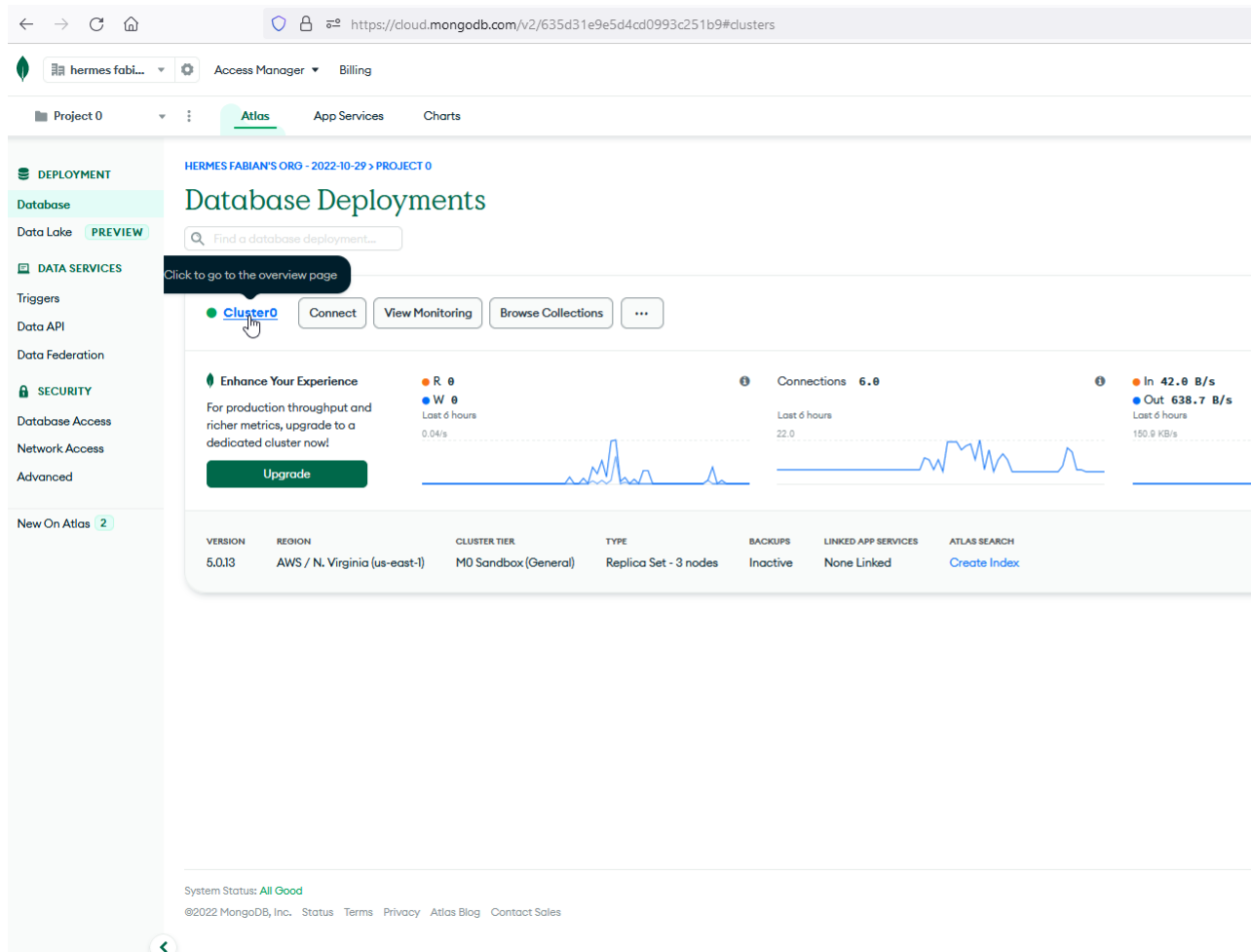


Figure 40: Página Principal MongoDB Atlas

← → ↻ 🏠 🔒 https://cloud.mongodb.com/v2/635d31e9e5d4cd0993c251b9#metrics/replicaSet/635d3324f829d500b706fb68/explorer/Transmilenio

hermes fabi... Access Manager Billing

Project 0 Atlas App Services Charts

DEPLOYMENT

Database

Data Lake PREVIEW

DATA SERVICES

Triggers

Data API

Data Federation

SECURITY

Database Access

Network Access

Advanced

HERMES FABIAN'S ORG - 2022-10-29 > PROJECT 0 > DATABASES

Cluster0

Overview Real Time Metrics Collections Search Profiler Performance Advisor Online Archive Cmd Line Tools

DATABASES: 2 COLLECTIONS: 3

+ Create Database

Search Namespaces

BD_Libros

Transmilenio

Estaciones

Imágenes

Transmilenio

LOGICAL DATA SIZE: 14.54MB STORAGE SIZE: 21.63MB INDEX SIZE: 72KB TOTAL COLLECTIONS: 2

| Collection Name | Documents | Logical Data Size | Avg Document Size |
|-----------------|-----------|-------------------|-------------------|
| Estaciones | 155 | 27.87KB | 185B |
| Imágenes | 6 | 14.51MB | 2.42MB |

Figure 41: Base de Datos y Colecciones

← → ↻ 🏠 🔒 https://cloud.mongodb.com/v2/635d31e9e5d4cd0993c251b9#metrics/replicaSet/635d3324f829d500b706fb68/explorer/Transmilenio/Estaciones/fin

hermes fabi... Access Manager Billing

Project 0 Atlas App Services Charts

HERMES FABIAN'S ORG - 2022-10-29 > PROJECT 0 > DATABASES

Cluster0

Overview Real Time Metrics Collections Search Profiler Performance Advisor Online Archive Cmd Line Tools

DATABASES: 2 COLLECTIONS: 3

+ Create Database

Search Namespaces

BD_Libros

Transmilenio

Estaciones

Imágenes

Transmilenio.Estaciones

STORAGE SIZE: 52KB LOGICAL DATA SIZE: 27.87KB TOTAL DOCUMENTS: 155 INDEXES TOTAL SIZE: 36KB

Find Indexes Schema Anti-Patterns Aggregation Search Indexes

FILTER { field: "value" }

QUERY RESULTS: 1-20 OF MANY

_id: ObjectId('636cb103cb312f76daa5bbe6')
IdEstación: "02200"
Estación: "Alcalá"
Cuadrante: "Calle 142 - Heroes"
Zona: "Terminal - Heroes"
Longitud: -74.0514538784185
Latitud: 4.72115729506492

_id: ObjectId('636cb103cb312f76daa5bbe7')
IdEstación: "09116"
Estación: "Avenida 39"
Cuadrante: "Las Forjas - Avenida 39"
Zona: "Portal 80 - Polo - Av. 39"
Longitud: -74.068691056891
Latitud: 4.62689711127706

_id: ObjectId('636cb103cb312f76daa5bbe8')
IdEstación: "04104"

Figure 42: Información Colección Estaciones

hermes fabi... Access Manager Billing

Project 0 Atlas App Services Charts

DEPLOYMENT

Database

Data Lake PREVIEW

DATA SERVICES

Triggers

Data API

Data Federation

SECURITY

Database Access

Network Access

Advanced

HERMES FABIAN'S ORG - 2022-10-29 > PROJECT 0 > DATABASES

ClusterO

Overview Real Time Metrics Collections Search Profiler Performance Advisor Online Archive Cmd Line Tools

DATABASES: 2 COLLECTIONS: 3

+ Create Database

Search Namespaces

BD_Libros

Transmilenio

Estaciones

Imágenes

Transmilenio.Imágenes

STORAGE SIZE: 21.58MB LOGICAL DATA SIZE: 14.51MB TOTAL DOCUMENTS: 6 INDEXES TOTAL SIZE: 36KB

Find Indexes Schema Anti-Patterns Aggregation Search Indexes

FILTER { field: "value" }

QUERY RESULTS: 1-1 OF MANY

```
{
  "_id": ObjectId('636f92ff121b4db08028f9f7'),
  "Content": BinData(0, '/9j/4QIGRXhpZgAASUkqAAgAAAAIAA8BAGAPAAAABgAAABABAgAIAAAAFgAAABoBBQABAAAAhgAAABsBBQABAAAAjgAAACB...
  "Nombre_Archivo": "02000_cabecera Autopista Norte.jpg",
  "Tipo_Archivo": "image/jpeg"
}
```

PREVIOUS 1 of many results

Figure 43: Información Colección Imágenes

6 Aplicación de ETL (Extract, Transform, Load) y Bodega de Datos *(Tercera entrega)*

6.1 Ejemplo de aplicación de ETL y Bodega de Datos *(Tercera entrega)*

7 Lecciones aprendidas (*Tercera entrega*)

8 Bibliografía

M. ANDRADE *El sistema integrado de transporte masivo Transmilenio: Recopilación, Estudio y Análisis de los componentes del sistema; Proyecto para la elaboración de unas memorias de la primera fase de operación del sistema.*, 2003

M. CORRAL *Plan de inversión para la disminución de evasores del pago en las estaciones del sistema de transporte masivo Transmilenio*, 2021

L. RÍOS *Análisis de datos de sentimientos enfocados al servicio de transporte masivo Transmilenio S.A aplicando tecnologías Big Data*, 2019