

# INSTITUTO POLITÉCNICO NACIONAL ESCUELA SUPERIOR DE CÓMPUTO



DATA MINING

---

## Práctica 3:Mini ETL de ingresos del metro

---

Graciano Herrera Gabriel  
Meza Zamora Abraham Manuel

30 de marzo de 2022

# Índice

|   |           |
|---|-----------|
| <b>1. Introducción</b>  | <b>2</b>  |
| 1.1. Extraer . . . . .  | 2         |
| 1.2. Transformar . . . . .  | 2         |
| 1.3. Cargar . . . . .   | 2         |
| <b>2. Desarrollo</b>  | <b>3</b>  |
| 2.1. Preguntas . . . . .  | 3         |
| 2.2. Estructura de transformación para la tabla de hechos . . . . . | 3         |
| 2.3. Flujo de trabajo del ETL . . . . .                             | 3         |
| <b>3. Catálogos o tablas complementarias</b>                        | <b>4</b>  |
| <b>4. Exploración los datos integrados</b>                          | <b>5</b>  |
| <b>5. Pseudocódigo</b>  | <b>13</b> |
| <b>6. Conclusiones</b>  | <b>17</b> |

# 1. Introducción

Extract, Transform and Load (ETL) es el proceso que permite a las organizaciones mover datos desde múltiples fuentes, reformatearlos y limpiarlos, y cargarlos en otra base de datos, data mart, o data warehouse para analizar, o en otro sistema operacional para apoyar un proceso de negocio.

## 1.1. Extraer

La primera parte del proceso ETL consiste en extraer los datos desde los sistemas de origen. La mayoría de los proyectos de almacenamiento de datos funcionan con datos provenientes de diferentes sistemas de origen. Cada sistema separado puede usar una organización diferente de los datos o formatos distintos. Los formatos de las fuentes normalmente se encuentran en bases de datos relacionales o ficheros planos, pero pueden incluir bases de datos no relacionales u otras estructuras diferentes. La extracción convierte los datos a un formato preparado para iniciar el proceso de transformación.

Una parte intrínseca del proceso de extracción es la de analizar los datos extraídos, de lo que resulta un chequeo que verifica si los datos cumplen la pauta o estructura que se esperaba. De no ser así los datos son rechazados.

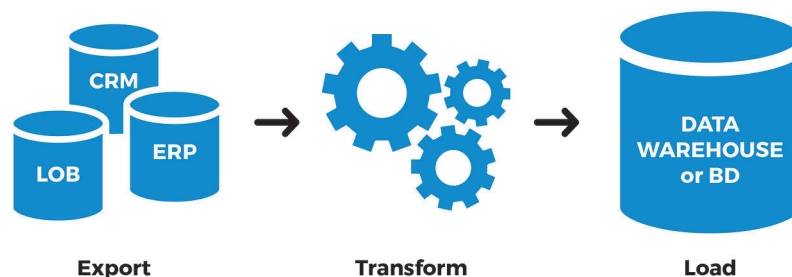
Un requisito importante que se debe exigir a la tarea de extracción es que ésta cause un impacto mínimo en el sistema origen. Si los datos a extraer son muchos, el sistema de origen se podría ralentizar e incluso colapsar, provocando que este no pueda utilizarse con normalidad para su uso cotidiano. Por esta razón, en sistemas grandes las operaciones de extracción suelen programarse en horarios o días donde este impacto sea nulo o mínimo.

## 1.2. Transformar

La fase de transformación aplica una serie de reglas de negocio o funciones sobre los datos extraídos para convertirlos en datos que serán cargados. Algunas fuentes de datos requerirán alguna pequeña manipulación de los datos.

## 1.3. Cargar

La fase de carga es el momento en el cual los datos de la fase anterior (transformación) son cargados en el sistema de destino. Dependiendo de los requisitos de la organización, este proceso puede abarcar una amplia variedad de acciones diferentes. En algunas bases de datos se sobrescribe la información antigua con nuevos datos. Los data warehouse mantienen un historial de los registros de manera que se pueda hacer una auditoría de los mismos y disponer de un rastro de toda la historia de un valor a lo largo del tiempo.



## 2. Desarrollo

### 2.1. Preguntas

1. ¿Cuál es el total de ingresos obtenidos por concepto de boletos en todas las líneas del metro? No, debido a que hay inconsistencias en los datos de la L12.
2. ¿Cuál es el total de ingresos promedio obtenidos que normalmente se obtienen durante el mes de diciembre en todas las líneas del metro? No, dado que hay inconsistencias en los datos de la L12.
3. ¿Cuál es el mes donde se obtiene el ingreso menor en todas las líneas del metro? No, dado que hay inconsistencias en los datos de la L12.
4. ¿Cuál es el día de la semana en el que históricamente se recibe la mayor cantidad de ingresos en cada línea del metro (de la 1 a la 12)? No, dado que hay inconsistencias en los datos de la L12.

### 2.2. Estructura de transformación para la tabla de hechos

A la tabla se le necesitan agregar las columnas de día, mes y año. Posteriormente se necesitan completar los datos de la línea 2.

La estructura de datos es la siguiente:

| fecha | tipoIngreso | linea | ingreso | dia | mes | anio | diaSemana |
|-------|-------------|-------|---------|-----|-----|------|-----------|
|       |             |       |         |     |     |      |           |

Cuadro 1: Estructura general de la BD.

### 2.3. Flujo de trabajo del ETL

1. Definimos la estructura de la base de datos en el gestor descrito en el Cuadro 1.
2. Definimos el catálogo para las líneas del metro descritas en el Cuadro 2.
3. Definimos el catálogo para los tipos de ingresos descritos en el Cuadro 3.
4. Exploramos las filas en el archivo de datos
  - a) Extraemos el día, mes y año del campo fecha separando los datos por el símbolo /.
  - b) El tipo de ingreso se obtiene de manera directa en la columna 1 de cada fila.
  - c) La línea del metro se calcula a partir del numero de la columna menos el desface del número de la columna.
  - d) El total de ingresos obtenidos por linea será calculado cuando ya se tengan los datos en la base de datos con la nueva estructura.
5. Ingresamos las queries generadas en un archivo `.sql` y ejecutamos en el gestor.
6. Generamos los queries para las vistas.
7. Hacemos las consultas en tableau.

### 3. Catálogos o tablas complementarias

#### a Catálogo de líneas de metro

| id | descripcion |
|----|-------------|
| 1  | 1           |
| 2  | 2           |
| 3  | 3           |
| 4  | 4           |
| 5  | 5           |
| 6  | 6           |
| 7  | 7           |
| 8  | 8           |
| 9  | 9           |
| 10 | A           |
| 11 | B           |
| 12 | 12          |

Cuadro 2: Catálogo de línea de metro.

#### b Catálogo de tipos de ingreso

| id | descripcion  |
|----|--------------|
| 1  | Boleto       |
| 2  | Tarjeta      |
| 3  | Recarga      |
| 4  | QR/Validador |

Cuadro 3: Catálogo de línea de tipo de ingreso.

#### c Otros que considere relevantes: Para esta práctica no ocupamos otra tabla complementaria.

## 4. Exploración los datos integrados

a Cantidad de registros totales (135,072) y por año

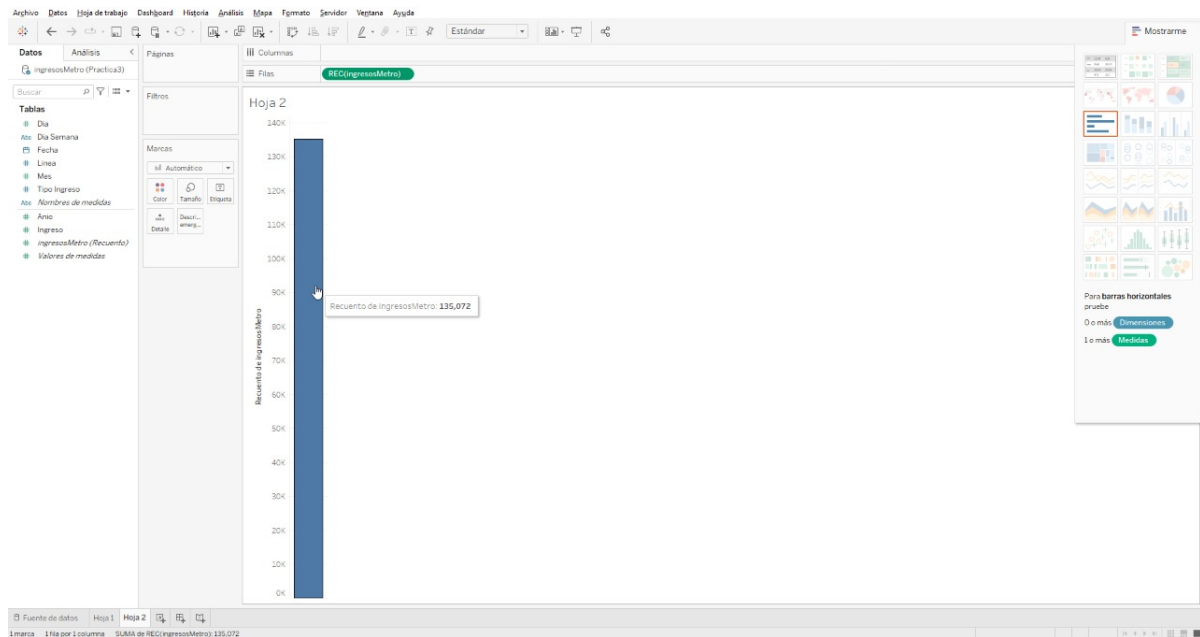


Figura 1: Resultado en Tableau de la cantidad de registros totales.

The screenshot shows a database query result window with a table containing one row and one column. The column is named 'registros' and the value is '135072'. The status bar at the bottom indicates 'Query executed successfully.' and '1 rows'.

|   | registros |
|---|-----------|
| 1 | 135072    |

Figura 2: Vista de la cantidad de registros totales.

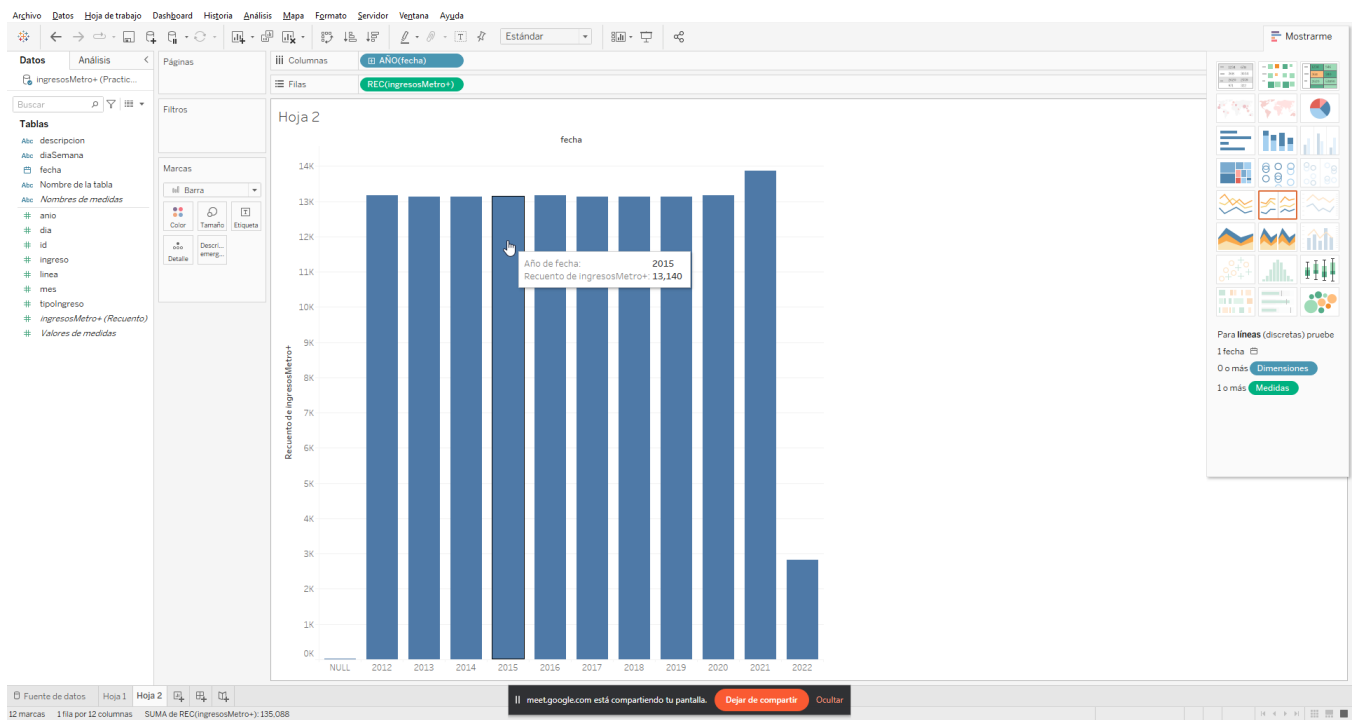


Figura 3: Resultado en Tableau de la cantidad de registros totales por año.

|    | registros | año  |
|----|-----------|------|
| 1  | 13140     | 2013 |
| 2  | 13140     | 2018 |
| 3  | 13140     | 2019 |
| 4  | 13176     | 2012 |
| 5  | 13176     | 2016 |
| 6  | 13140     | 2017 |
| 7  | 2832      | 2022 |
| 8  | 13140     | 2014 |
| 9  | 13140     | 2015 |
| 10 | 13872     | 2021 |
| 11 | 13176     | 2020 |

Figura 4: Vista de la cantidad de registros totales por año.

b Total de ingresos por línea en todo el periodo de tiempo

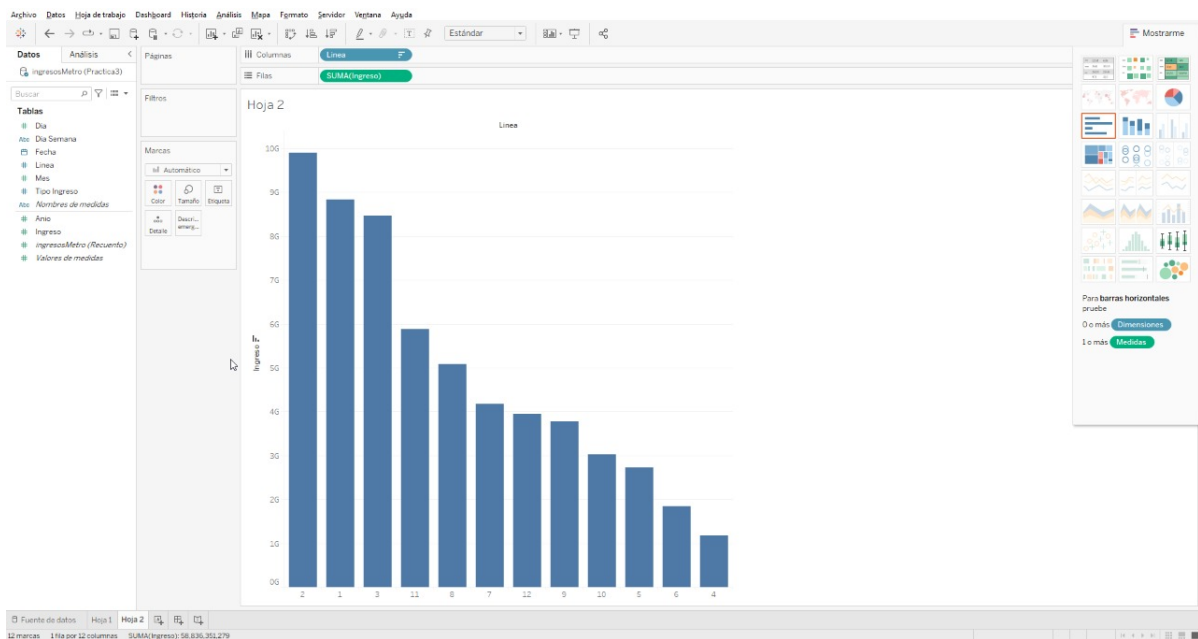


Figura 5: Resultado en Tableau de la cantidad de ingresos por línea en todo el periodo de tiempo.

|    | ingresos        | descripcion |
|----|-----------------|-------------|
| 1  | 8838648727.8951 | 1           |
| 2  | 3946654282.1502 | 12          |
| 3  | 9901695866.4175 | 2           |
| 4  | 8465108680.8136 | 3           |
| 5  | 1174194870.1118 | 4           |
| 6  | 2732934300.2163 | 5           |
| 7  | 1841744964.413  | 6           |
| 8  | 4176092969.5908 | 7           |
| 9  | 5080631230.3712 | 8           |
| 10 | 3778183613.8981 | 9           |
| 11 | 3020184075.1195 | A           |
| 12 | 5880277697.9153 | B           |

Figura 6: Vista de la cantidad de ingresos por línea en todo el periodo de tiempo.

c Total de ingresos por año



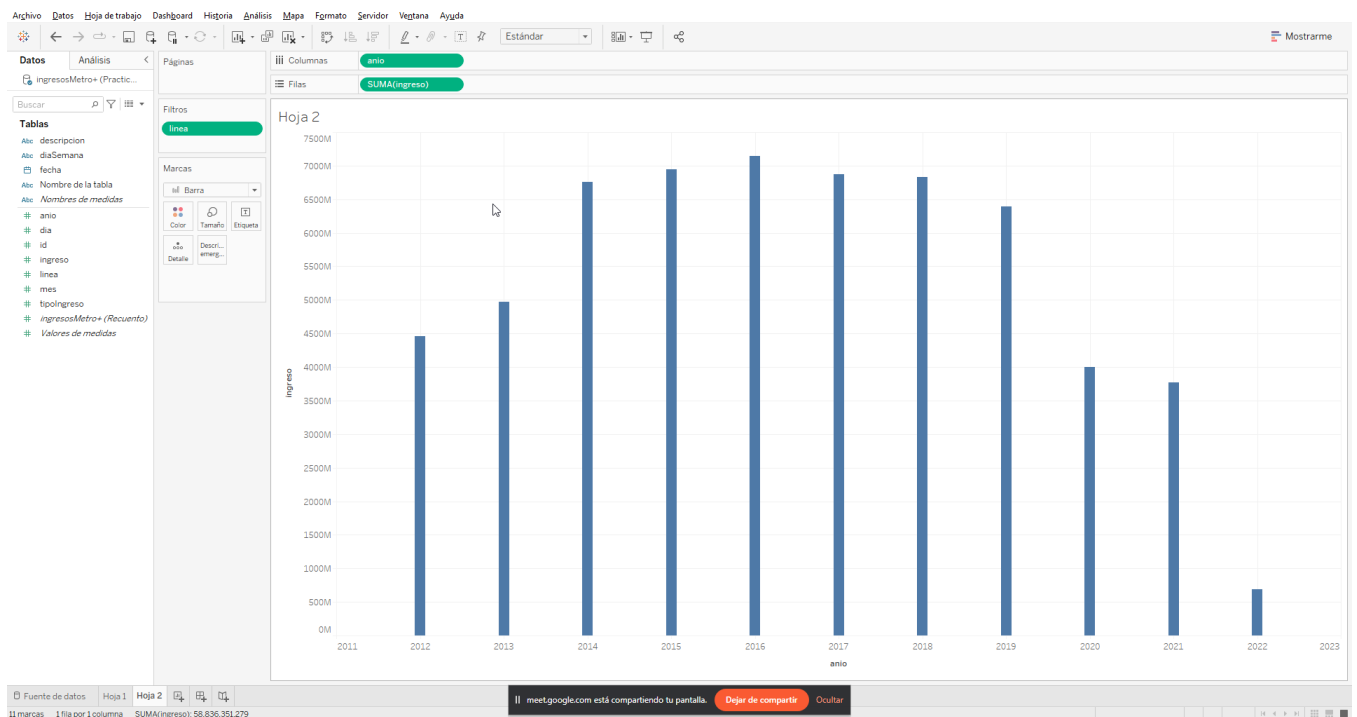


Figura 7: Resultado en Tableau del total de ingresos por año.

|    | ingresos        | año  |
|----|-----------------|------|
| 1  | 4463971819.97   | 2012 |
| 2  | 4968039307.77   | 2013 |
| 3  | 6763679176.35   | 2014 |
| 4  | 6941626497.21   | 2015 |
| 5  | 7143216975.34   | 2016 |
| 6  | 6877181523.90   | 2017 |
| 7  | 6835859192.00   | 2018 |
| 8  | 6388238731.00   | 2019 |
| 9  | 3999879929.00   | 2020 |
| 10 | 3768411465.3731 | 2021 |
| 11 | 686246660.9993  | 2022 |

Figura 8: Vista del total de ingresos por año.

d Total máximo (2016) y mínimo (2021 considerando que los datos del 2022 todavía no se terminan de procesar) de ingresos por año

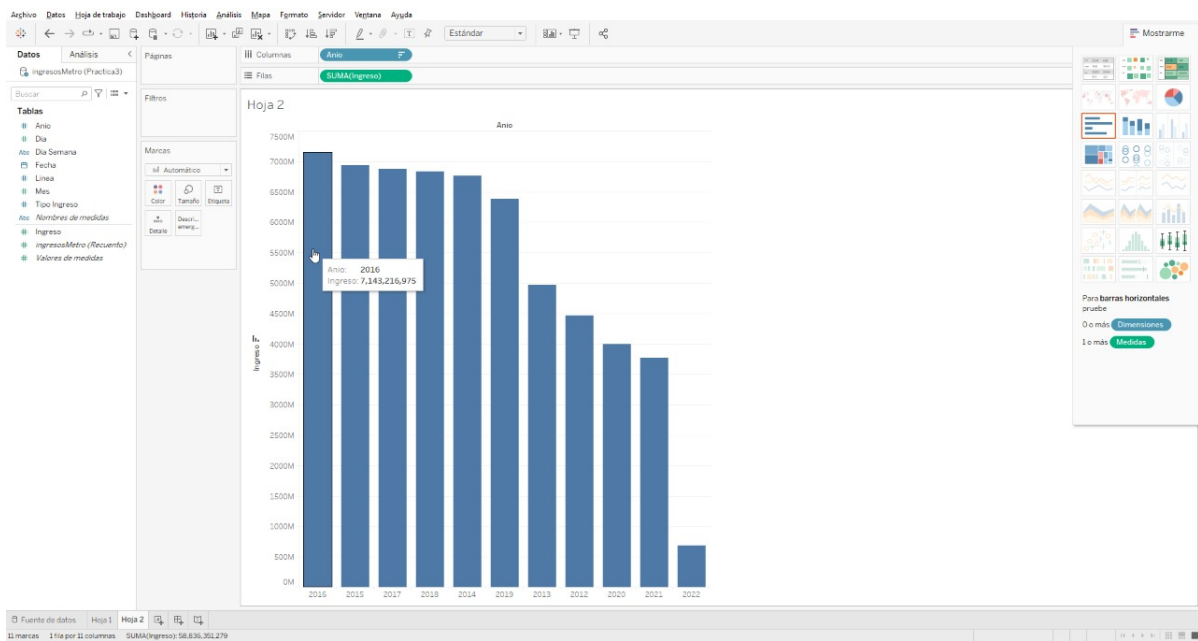


Figura 9: Resultado en Tableau máximo de ingresos por año.

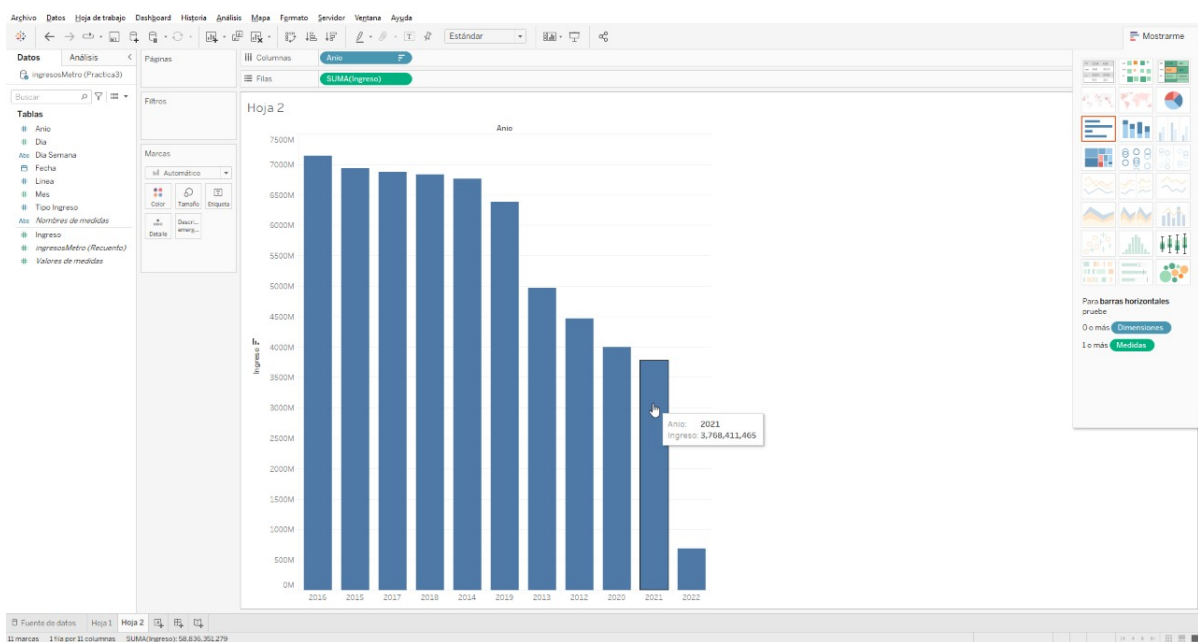


Figura 10: Resultado en Tableau del mínimo de ingresos por año.

|   | año  | máximo        |
|---|------|---------------|
| 1 | 2016 | 7143216975.34 |

Figura 11: Vista del máximo de ingresos por año.

|   | año  | mínimo         |
|---|------|----------------|
| 1 | 2022 | 686246660.9993 |

Figura 12: Vista del mínimo de ingresos por año.

### e Total de ingresos por tipo de ingreso

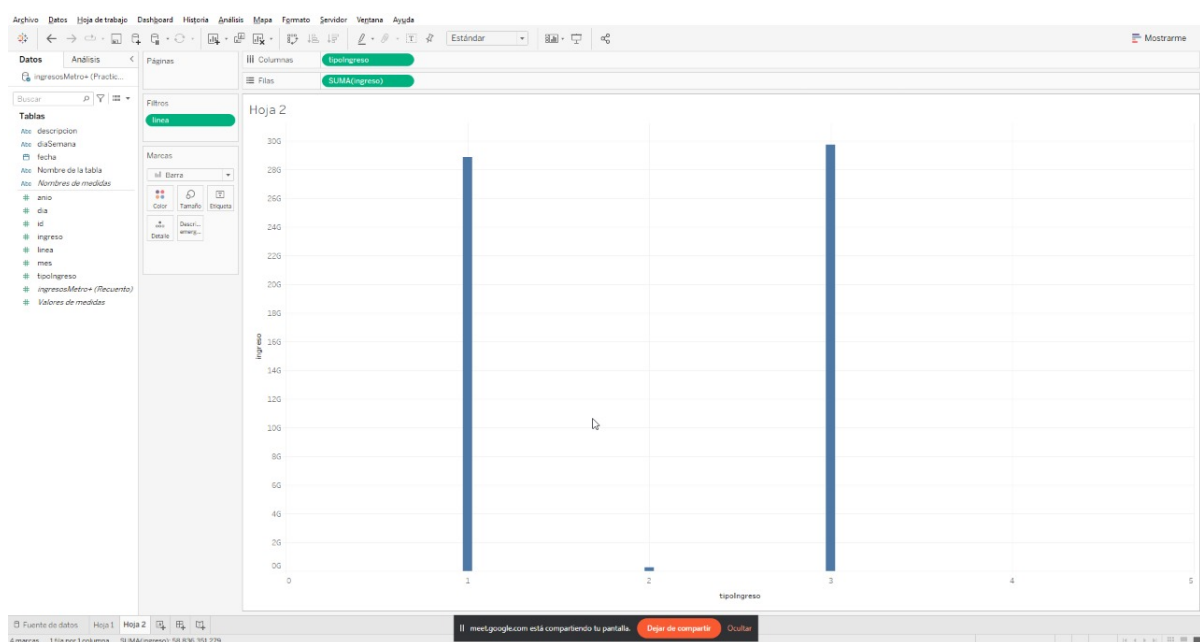


Figura 13: Resultado en Tableau de ingresos por tipo de ingreso.

|   | ingresos         | descripcion  |
|---|------------------|--------------|
| 1 | 1028658.00       | QR/Validador |
| 2 | 258344910.00     | Tarjeta      |
| 3 | 28865063678.00   | Boleto       |
| 4 | 29711914032.9124 | Recarga      |

Figura 14: Vista del total de ingresos por tipo de ingreso.

f Total de ingresos por día de la semana

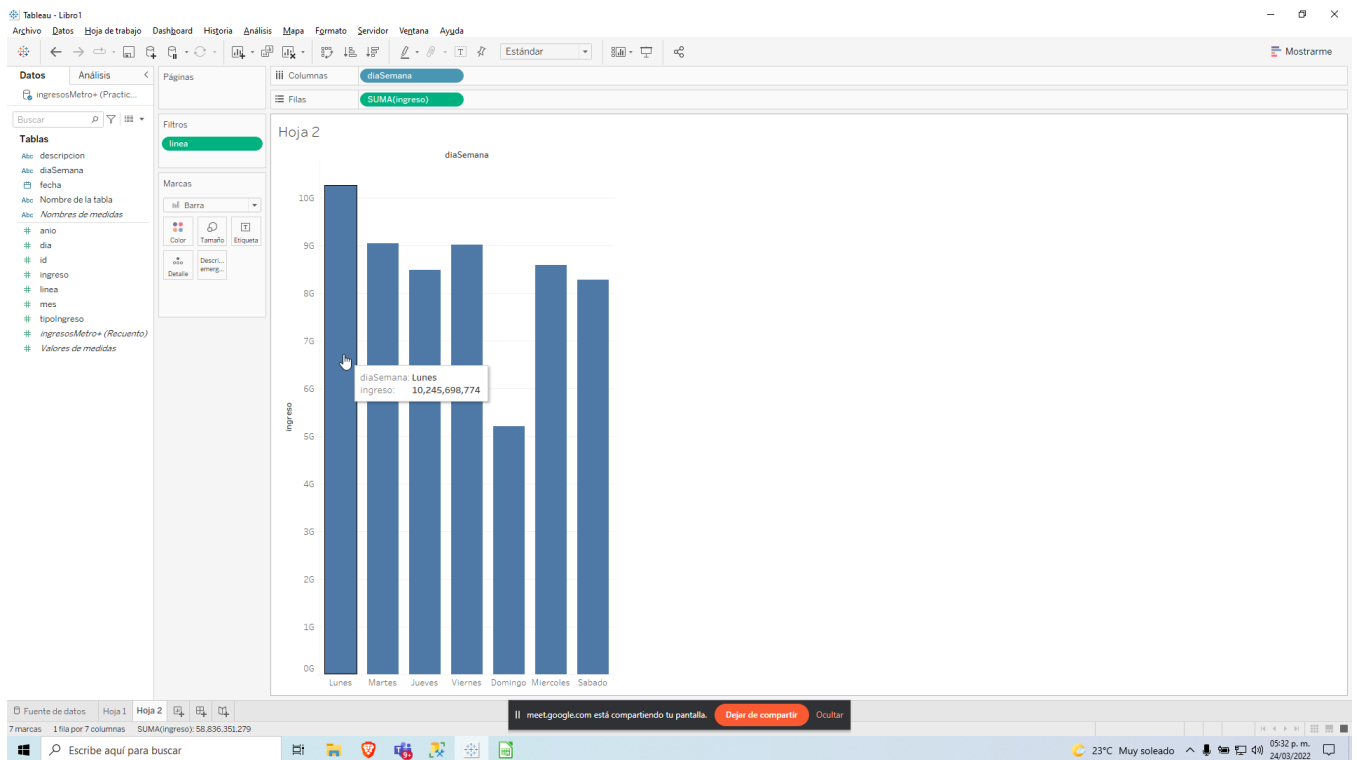


Figura 15: Resultado en Tableau del total de ingresos por tipo de ingreso.

|   | ingresos         | diaSemana |
|---|------------------|-----------|
| 1 | 5203659847.604   | Domingo   |
| 2 | 8271780660.4574  | Sabado    |
| 3 | 8479938992.3785  | Jueves    |
| 4 | 8588348250.1642  | Miercoles |
| 5 | 9012331043.7131  | Viernes   |
| 6 | 9034593711.0884  | Martes    |
| 7 | 10245698773.5068 | Lunes     |

Figura 16: Vista del total de ingresos por día de la semana.

g Total de ingresos por día de la semana y por año

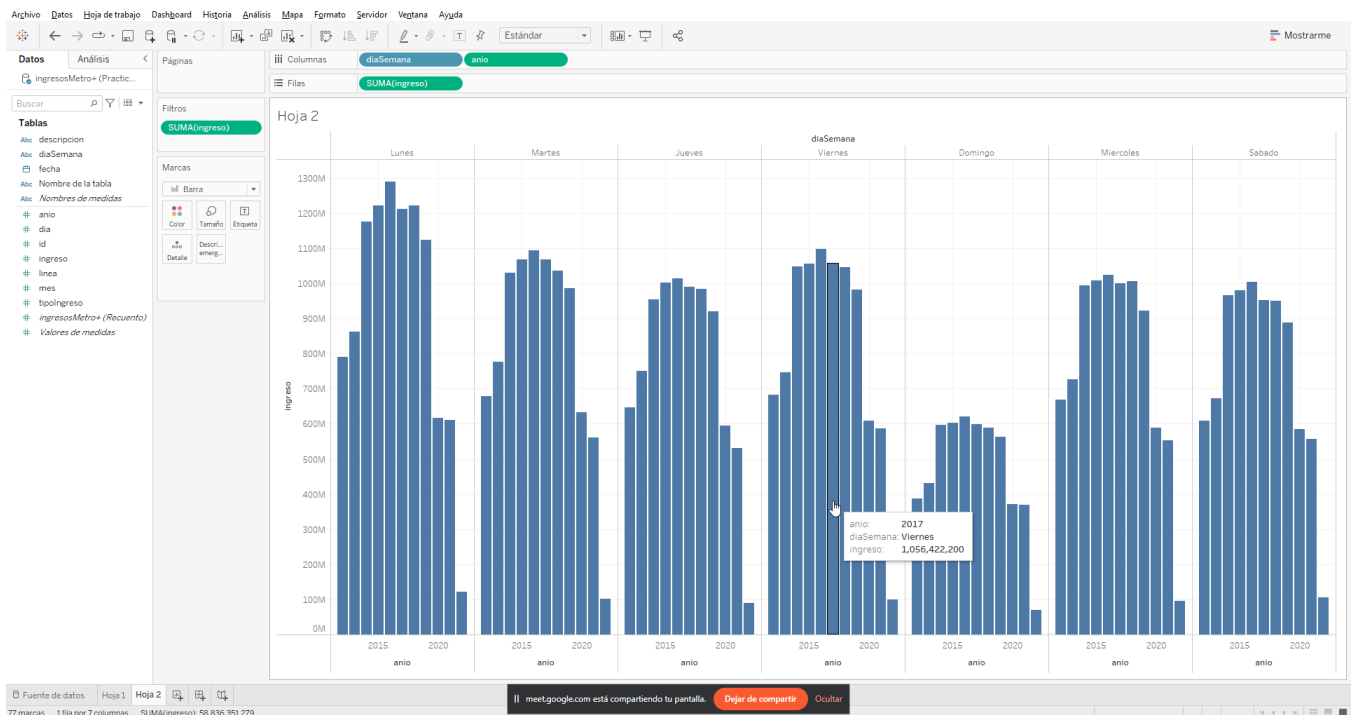


Figura 17: Resultado en Tableau del total de ingresos por día de la semana y por año.

|    | Ingresos      | diaSemana | anio |
|----|---------------|-----------|------|
| 1  | 790731115.70  | Lunes     | 2012 |
| 2  | 387932932.10  | Domingo   | 2012 |
| 3  | 679551663.60  | Martes    | 2012 |
| 4  | 681904501.75  | Viernes   | 2012 |
| 5  | 667853151.15  | Miercoles | 2012 |
| 6  | 647457760.57  | Jueves    | 2012 |
| 7  | 608540695.10  | Sabado    | 2012 |
| 8  | 751212153.11  | Jueves    | 2013 |
| 9  | 727543481.60  | Miercoles | 2013 |
| 10 | 861532627.36  | Lunes     | 2013 |
| 11 | 745883627.36  | Viernes   | 2013 |
| 12 | 431955288.54  | Domingo   | 2013 |
| 13 | 673683663.80  | Sabado    | 2013 |
| 14 | 776228466.00  | Martes    | 2013 |
| 15 | 1047638848.09 | Viernes   | 2014 |
| 16 | 1176847913.05 | Lunes     | 2014 |
| 17 | 953462702.36  | Jueves    | 2014 |
| 18 | 965605082.09  | Sabado    | 2014 |
| 19 | 596319089.09  | Domingo   | 2014 |
| 20 | 1029884300.07 | Martes    | 2014 |
| 21 | 993921241.60  | Miercoles | 2014 |
| 22 | 1067368350.72 | Martes    | 2015 |
| 23 | 1221845768.42 | Lunes     | 2015 |
| 24 | 1008886970.46 | Miercoles | 2015 |
| 25 | 980382673.60  | Sabado    | 2015 |
| 26 | 603840981.19  | Domingo   | 2015 |
| 27 | 1003101305.29 | Jueves    | 2015 |
| 28 | 1056200447.53 | Viernes   | 2015 |
| 29 | 1003870139.10 | Sabado    | 2016 |
| 30 | 1093128682.90 | Martes    | 2016 |
| 31 | 1013578910.10 | Jueves    | 2016 |
| 32 | 1289123863.99 | Lunes     | 2016 |
| 33 | 621694899.80  | Domingo   | 2016 |

Figura 18: Vista del total de ingresos por día de la semana y por año.

h Reportar valores atípicos y valores vacíos y su impacto, y otros errores o problemas al procesar los datos.

El principal valor atípico aparecía en los años anteriores a la construcción de la línea 12, en dónde no había nada, el ajuste que hicimos fue cambiar esos valores nulos por un ingreso igual a cero.

## 5. Pseudocódigo

**Algoritmo:** Extraer datos.

**Entrada:** Archivo de datos csv.

**Salida:** Querys para el sistema gestor de BD.

Tomamos los datos indexados en 0.

```

1: for  $i \leftarrow 1, numFilas$  do                                ▷ Ignoramos la primera fila del documento
2:    $dia, mes, anio \leftarrow datos[i][0]$                     ▷ La fecha siempre se encuentra en la primera columna
3:    $fecha \leftarrow anio + mes + dia$                         ▷ Armamos la fecha de acuerdo al formato de la BD
4:    $tipo \leftarrow datos[i][1]$                               ▷ El tipo siempre se encuentra en la segunda columna
5:   for  $j \leftarrow 2, numColumnas$  do                      ▷ Procesamos los datos de cada línea del metro
6:     if  $datos[i][j] \neq NULL$  then                          ▷ Si el dato es nulo, le asignamos un 0.
7:        $ingreso \leftarrow datos[i][j]$ 
8:     else
9:        $ingreso \leftarrow 0$ 
10:    end if
11:     $linea \leftarrow j - 1$                                 ▷ Calculamos la línea del metro en la que estamos y armamos el
query
12:     $query \leftarrow fecha + tipo + linea + ingreso + dia + mes + anio$ 
13:  end for
14: end for

```

a Código Utilizado

```

import csv
import datetime
import calendar

#D as de la semana
dias = { 'Monday': 'Lunes',
         'Tuesday': 'Martes',
         'Wednesday': 'Miercoles',
         'Thursday': 'Jueves',
         'Friday': 'Viernes',
         'Saturday': 'Sabado',
         'Sunday': 'Domingo' }

tipo_ingreso = { 'Boletos': 1,
                 'Tarjetas': 2,
                 'Recargas': 3,
                 'QR/Validador': 4 }

with open('datos.csv', newline='') as File:
    reader = csv.reader(File)
    first = True
    for row in reader:
        if not first:
            #FECHA
            dia, mes, ano = row[0].split('/')
            dia, mes, ano = int(dia), int(mes), int(ano)
            fecha = datetime.datetime(ano, mes, dia)
            dia_semana = calendar.day_name[fecha.weekday()]
            fecha = datetime.date(ano, mes, dia)

            #TIPO DE INGRESO
            tipo = tipo_ingreso[row[1]]

            for i in range(2, len(row)):
                ingreso = float(row[i]) if row[i] else 0
                linea = i - 1
                query = f"INSERT INTO ingresosMetro values ('{fecha}', {ti
                print(query)

        first = False

```

```

Prac03 > ls
creaciónBD.sql  datos.csv      extraer.py      vistas.sql
Prac03 > python3 extraer.py > datos.sql
Prac03 > ls
creaciónBD.sql  datos.sql      vistas.sql
datos.csv       extraer.py
Prac03 > █

```

Figura 19: Ejecución del programa, la salida del mismo es guardada en un archivo con las queries correspondientes.

```

1 INSERT INTO ingresosMetro values('2012-01-01', 1, 1, 648825.0, 1, 1, 2012, 'Domingo')
2 INSERT INTO ingresosMetro values('2012-01-01', 1, 2, 657120.0, 1, 1, 2012, 'Domingo')
3 INSERT INTO ingresosMetro values('2012-01-01', 1, 3, 466803.0, 1, 1, 2012, 'Domingo')
4 INSERT INTO ingresosMetro values('2012-01-01', 1, 4, 60120.0, 1, 1, 2012, 'Domingo')
5 INSERT INTO ingresosMetro values('2012-01-01', 1, 5, 242280.0, 1, 1, 2012, 'Domingo')
6 INSERT INTO ingresosMetro values('2012-01-01', 1, 6, 138525.0, 1, 1, 2012, 'Domingo')
7 INSERT INTO ingresosMetro values('2012-01-01', 1, 7, 138690.0, 1, 1, 2012, 'Domingo')
8 INSERT INTO ingresosMetro values('2012-01-01', 1, 8, 328383.0, 1, 1, 2012, 'Domingo')
9 INSERT INTO ingresosMetro values('2012-01-01', 1, 9, 176400.0, 1, 1, 2012, 'Domingo')
10 INSERT INTO ingresosMetro values('2012-01-01', 1, 10, 275370.0, 1, 1, 2012, 'Domingo')
11 INSERT INTO ingresosMetro values('2012-01-01', 1, 11, 472665.0, 1, 1, 2012, 'Domingo')
12 INSERT INTO ingresosMetro values('2012-01-01', 1, 12, 0, 1, 1, 2012, 'Domingo')
13 INSERT INTO ingresosMetro values('2012-01-02', 1, 1, 1532940.0, 2, 1, 2012, 'Lunes')
14 INSERT INTO ingresosMetro values('2012-01-02', 1, 2, 1491780.0, 2, 1, 2012, 'Lunes')
15 INSERT INTO ingresosMetro values('2012-01-02', 1, 3, 1168590.0, 2, 1, 2012, 'Lunes')
16 INSERT INTO ingresosMetro values('2012-01-02', 1, 4, 165690.0, 2, 1, 2012, 'Lunes')
17 INSERT INTO ingresosMetro values('2012-01-02', 1, 5, 437025.0, 2, 1, 2012, 'Lunes')
18 INSERT INTO ingresosMetro values('2012-01-02', 1, 6, 286620.0, 2, 1, 2012, 'Lunes')
19 INSERT INTO ingresosMetro values('2012-01-02', 1, 7, 433323.0, 2, 1, 2012, 'Lunes')
20 INSERT INTO ingresosMetro values('2012-01-02', 1, 8, 777951.0, 2, 1, 2012, 'Lunes')
21 INSERT INTO ingresosMetro values('2012-01-02', 1, 9, 504600.0, 2, 1, 2012, 'Lunes')
22 INSERT INTO ingresosMetro values('2012-01-02', 1, 10, 514455.0, 2, 1, 2012, 'Lunes')
23 INSERT INTO ingresosMetro values('2012-01-02', 1, 11, 965385.0, 2, 1, 2012, 'Lunes')
24 INSERT INTO ingresosMetro values('2012-01-02', 1, 12, 0, 2, 1, 2012, 'Lunes')
25 INSERT INTO ingresosMetro values('2012-01-03', 1, 1, 1398639.0, 3, 1, 2012, 'Martes')
26 INSERT INTO ingresosMetro values('2012-01-03', 1, 2, 1482255.0, 3, 1, 2012, 'Martes')
27 INSERT INTO ingresosMetro values('2012-01-03', 1, 3, 1143675.0, 3, 1, 2012, 'Martes')
28 INSERT INTO ingresosMetro values('2012-01-03', 1, 4, 162360.0, 3, 1, 2012, 'Martes')
29 INSERT INTO ingresosMetro values('2012-01-03', 1, 5, 388575.0, 3, 1, 2012, 'Martes')
"datos.sql" 135072L, 11582937B

```

Figura 20: Contenido de la parte inicial del archivo con las queries generadas.



```

135044 INSERT INTO ingresosMetro values('2022-02-26', 4, 8, 0.0, 26, 2, 2022, 'Sabado')
135045 INSERT INTO ingresosMetro values('2022-02-26', 4, 9, 12775.0, 26, 2, 2022, 'Sabado')
135046 INSERT INTO ingresosMetro values('2022-02-26', 4, 10, 0.0, 26, 2, 2022, 'Sabado')
135047 INSERT INTO ingresosMetro values('2022-02-26', 4, 11, 0.0, 26, 2, 2022, 'Sabado')
135048 INSERT INTO ingresosMetro values('2022-02-26', 4, 12, 0.0, 26, 2, 2022, 'Sabado')
135049 INSERT INTO ingresosMetro values('2022-02-27', 4, 1, 0.0, 27, 2, 2022, 'Domingo')
135050 INSERT INTO ingresosMetro values('2022-02-27', 4, 2, 0.0, 27, 2, 2022, 'Domingo')
135051 INSERT INTO ingresosMetro values('2022-02-27', 4, 3, 0.0, 27, 2, 2022, 'Domingo')
135052 INSERT INTO ingresosMetro values('2022-02-27', 4, 4, 0.0, 27, 2, 2022, 'Domingo')
135053 INSERT INTO ingresosMetro values('2022-02-27', 4, 5, 0.0, 27, 2, 2022, 'Domingo')
135054 INSERT INTO ingresosMetro values('2022-02-27', 4, 6, 0.0, 27, 2, 2022, 'Domingo')
135055 INSERT INTO ingresosMetro values('2022-02-27', 4, 7, 0.0, 27, 2, 2022, 'Domingo')
135056 INSERT INTO ingresosMetro values('2022-02-27', 4, 8, 0.0, 27, 2, 2022, 'Domingo')
135057 INSERT INTO ingresosMetro values('2022-02-27', 4, 9, 6145.0, 27, 2, 2022, 'Domingo')
135058 INSERT INTO ingresosMetro values('2022-02-27', 4, 10, 0.0, 27, 2, 2022, 'Domingo')
135059 INSERT INTO ingresosMetro values('2022-02-27', 4, 11, 0.0, 27, 2, 2022, 'Domingo')
135060 INSERT INTO ingresosMetro values('2022-02-27', 4, 12, 0.0, 27, 2, 2022, 'Domingo')
135061 INSERT INTO ingresosMetro values('2022-02-28', 4, 1, 0.0, 28, 2, 2022, 'Lunes')
135062 INSERT INTO ingresosMetro values('2022-02-28', 4, 2, 0.0, 28, 2, 2022, 'Lunes')
135063 INSERT INTO ingresosMetro values('2022-02-28', 4, 3, 0.0, 28, 2, 2022, 'Lunes')
135064 INSERT INTO ingresosMetro values('2022-02-28', 4, 4, 0.0, 28, 2, 2022, 'Lunes')
135065 INSERT INTO ingresosMetro values('2022-02-28', 4, 5, 0.0, 28, 2, 2022, 'Lunes')
135066 INSERT INTO ingresosMetro values('2022-02-28', 4, 6, 0.0, 28, 2, 2022, 'Lunes')
135067 INSERT INTO ingresosMetro values('2022-02-28', 4, 7, 0.0, 28, 2, 2022, 'Lunes')
135068 INSERT INTO ingresosMetro values('2022-02-28', 4, 8, 0.0, 28, 2, 2022, 'Lunes')
135069 INSERT INTO ingresosMetro values('2022-02-28', 4, 9, 11920.0, 28, 2, 2022, 'Lunes')
135070 INSERT INTO ingresosMetro values('2022-02-28', 4, 10, 0.0, 28, 2, 2022, 'Lunes')
135071 INSERT INTO ingresosMetro values('2022-02-28', 4, 11, 0.0, 28, 2, 2022, 'Lunes')
135072 INSERT INTO ingresosMetro values('2022-02-28', 4, 12, 0.0, 28, 2, 2022, 'Lunes')
"datos.sql" 135072L, 11582937B written

```

Figura 21: Contenido de la parte final del archivo con las queries generadas.

b Modelo de datos empleado para catálogos y tabla de hechos principal

```

CREATE TABLE tipoIngreso(
    id int NOT NULL IDENTITY,
    descripcion varchar(50) NOT NULL,
    PRIMARY KEY (id)
);

CREATE TABLE lineaMetro(
    id int NOT NULL IDENTITY,
    descripcion varchar(50) NOT NULL,
    PRIMARY KEY (id)
);

CREATE TABLE ingresosMetro(
    fecha date NOT NULL,
    tipoIngreso int NOT NULL,
    linea int NOT NULL,
    ingreso money NOT NULL,
    dia int NOT NULL,
    mes int NOT NULL,
    a o int NOT NULL,
    diaSemana varchar(10) NOT NULL,
    FOREIGN KEY(tipoIngreso) REFERENCES tipoIngreso(id),
    FOREIGN KEY(linea) REFERENCES lineaMetro(id),
    PRIMARY KEY(fecha , linea)
);

INSERT INTO tipoIngreso VALUES('Boleto');
INSERT INTO tipoIngreso VALUES('Tarjeta');

```

```

INSERT INTO tipoIngreso VALUES( 'Recarga' );
INSERT INTO tipoIngreso VALUES( 'QR/Validador' );

INSERT INTO lineaMetro VALUES( '1' );
INSERT INTO lineaMetro VALUES( '2' );
INSERT INTO lineaMetro VALUES( '3' );
INSERT INTO lineaMetro VALUES( '4' );
INSERT INTO lineaMetro VALUES( '5' );
INSERT INTO lineaMetro VALUES( '6' );
INSERT INTO lineaMetro VALUES( '7' );
INSERT INTO lineaMetro VALUES( '8' );
INSERT INTO lineaMetro VALUES( '9' );
INSERT INTO lineaMetro VALUES( 'A' );
INSERT INTO lineaMetro VALUES( 'B' );
INSERT INTO lineaMetro VALUES( '12' );

```

## 6. Conclusiones

### ■ Graciano Herrera Gabriel

En esta práctica pudimos realizar una conocer como es el proceso de armar una ETL de primera mano, considero que es de suma importancia realizar este proceso de manera muy cuidadosa y teniendo un análisis previo de los datos que tenemos para que al momento de realizar una exploración de los mismos y querer sacar conclusiones nuestros datos no estén de manera errónea o que causen alguna confusión a la interpretación final de éstos. El proceso más tardado fue el del análisis y transformación de los datos, buscar una nueva estructura para esos datos fue lo más complicado, la parte de la limpieza de datos no fue tan complicado ya que se podía hacer de manera relativamente fácil, pero seguramente habrá dataset en los que este proceso sea mucho más complejo. Por último, la carga e interpretación de los datos no fue tan difícil, ya que tableu ayuda mucho a realizar este proceso de manera intuitiva y visual.

### ■ Meza Zamora Abraham Manuel

Al realizar esta práctica nos pudimos dar cuenta de la complejidad que requiere armar un ETL, comenzando por la extracción de datos, que como en tareas anteriores es importante detectar anomalías en la información para no generar un problema para después. La transformación que propusimos consistía en reducir el número de columnas, asignar las líneas y los tipos de ingresos a catálogos, esto con la finalidad de mejorar el procesamiento de los datos, y en consecuencia la carga. Los pasos anteriores nos permitieron de manera muy general y resumida entender el comportamiento y preferencias de las personas que usan el sistema de metro en la ciudad de México, esta información podría ayudar al gobierno en determinadas situaciones, ya que se conocen datos como el día en que se tiene una mayor cantidad de ingresos, y también se conocen la tendencia que se tiene hacia el consumo de las recargas sobre los boletos en los últimos años.