Gian Tituaña, 325991.

```
!pip install snowflake-connector-python
```

Mostrar salida oculta

```
import snowflake.connector
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from datetime import datetime
import warnings

warnings.filterwarnings('ignore')
sns.set_style('whitegrid')
plt.rcParams['figure.figsize'] = (14, 6)

# %%
# Conexión a Snowflake
conn = snowflake.connector.connect(
    user='',
    password='',
    account='',
    warehouse='',
    database='',
    schema=''
)

print("✓ Conexión establecida con Snowflake")
```

```
✓ Conexión establecida con Snowflake
```

```
def execute_query(query, title):
    print(f"\n--- {title} ---")
    try:
        df = pd.read_sql(query, conn)
        print(f"Filas obtenidas: {len(df)}")
        display(df.head(10))
    except Exception as e:
        print(f"Error al ejecutar la consulta: {e}")
```

1. Demanda por zona y mes: ¿cuáles son las 10 zonas con más viajes por mes? (PU y DO por separado).

```
# Q1A. Top 10 Zonas de Recogida (PU)
QUERY_1A = """
SELECT
    D.year,
    D.month_name,
    Z.Borough AS pickup_borough,
    Z.Zone AS pickup_zone,
    COUNT(F.trip_id) AS total_trips
FROM GOLD.fct_trips F
JOIN GOLD.dim_zone Z ON F.pu_zone_sk = Z.zone_sk -- PU Zone
JOIN GOLD.dim_date D ON F.pickup_date_sk = D.date_sk
GROUP BY 1, 2, 3, 4
QUALIFY ROW_NUMBER() OVER (PARTITION BY D.year, D.month_name ORDER BY total_trips DESC) <= 10
ORDER BY D.year, D.month_name, total_trips DESC;
"""
execute_query(QUERY_1A, "1.A. TOP 10 ZONAS DE RECOGIDA (PU) POR MES")
```

```
--- 1.A. TOP 10 ZONAS DE RECOGIDA (PU) POR MES ---
Filas obtenidas: 1294
```

	YEAR	MONTH_NAME	PICKUP_BOROUGH	PICKUP_ZONE	TOTAL_TRIPS
0	2015	Apr	Manhattan	Upper East Side South	496506
1	2015	Apr	Manhattan	Midtown Center	460248
2	2015	Apr	Manhattan	Upper East Side North	457178
3	2015	Apr	Manhattan	Midtown East	436910
4	2015	Apr	Manhattan	Murray Hill	423564
5	2015	Apr	Manhattan	Union Sq	420559
6	2015	Apr	Manhattan	East Village	415857
7	2015	Apr	Manhattan	Times Sq/Theatre District	412655
8	2015	Apr	Manhattan	Penn Station/Madison Sq West	401296
9	2015	Apr	Manhattan	Clinton East	400316

```
# Q1B. Top 10 Zonas de Recogida (DO)
QUERY_1B = """
SELECT
    D.year,
    D.month_name,
    Z.Borough AS dropoff_borough,
    Z.Zone AS dropoff_zone,
    COUNT(F.trip_id) AS total_trips
FROM GOLD.fct_trips F
JOIN GOLD.dim_zone Z ON F.do_zone_sk = Z.zone_sk -- DO Zone
JOIN GOLD.dim_date D ON F.dropoff_date_sk = D.date_sk
GROUP BY 1, 2, 3, 4
-- Filtra para mantener solo las 10 mejores zonas de destino por mes
QUALIFY ROW_NUMBER() OVER (PARTITION BY D.year, D.month_name ORDER BY total_trips DESC) <= 10
ORDER BY D.year, D.month_name, total_trips DESC;
"""
execute_query(QUERY_1B, "1.B. TOP 10 ZONAS DE RECOGIDA (DO) POR MES")
```

```
--- 1.B. TOP 10 ZONAS DE RECOGIDA (DO) POR MES ---
Filas obtenidas: 1294
```

	YEAR	MONTH_NAME	DROPOFF_BOROUGH	DROPOFF_ZONE	TOTAL_TRIPS
0	2015	Apr	Manhattan	Midtown Center	501911
1	2015	Apr	Manhattan	Upper East Side North	468096
2	2015	Apr	Manhattan	Upper East Side South	439934
3	2015	Apr	Manhattan	Murray Hill	417886
4	2015	Apr	Manhattan	Times Sq/Theatre District	408092
5	2015	Apr	Manhattan	Midtown East	405979
6	2015	Apr	Manhattan	Union Sq	376837
7	2015	Apr	Manhattan	Penn Station/Madison Sq West	347802
8	2015	Apr	Manhattan	East Village	341253
9	2015	Apr	Manhattan	Clinton East	340091

2. Ingresos y propinas: ¿cómo varían los ingresos totales y el tip % por borough y mes?

```
# Q2. Ingresos y Propinas por Borough y Mes
QUERY_2 = """
SELECT
    D.year,
    D.month_name,
    Z.Borough AS pickup_borough,
    SUM(F.total_amount) AS total_revenue,
    SUM(F.tip_amount) AS total_tips,
    -- Cálculo del porcentaje de propina (usando NULLIF para evitar división por cero)
    ROUND( (SUM(F.tip_amount) / NULLIF(SUM(F.total_amount), 0)) * 100, 2) AS tip_percentage
FROM GOLD.fct_trips F
JOIN GOLD.dim_zone Z ON F.pu_zone_sk = Z.zone_sk
JOIN GOLD.dim_date D ON F.pickup_date_sk = D.date_sk
GROUP BY 1, 2, 3
ORDER BY 1, D.month_name, total_revenue DESC;
"""
execute_query(QUERY_2, "2. INGRESOS TOTALES Y PORCENTAJE DE PROPINAS")

```

```
--- 2. INGRESOS TOTALES Y PORCENTAJE DE PROPINAS ---
Filas obtenidas: 1034
```

	YEAR	MONTH_NAME	PICKUP_BOROUGH	TOTAL_REVENUE	TOTAL_TIPS	TIP_PERCENTAGE
0	2015	Apr	Manhattan	1.772094e+08	18512429.09	10.45
1	2015	Apr	Queens	3.574518e+07	3488979.45	9.76
2	2015	Apr	Brooklyn	1.467913e+07	1557768.72	10.61
3	2015	Apr	Unknown	3.673395e+06	393841.56	10.72
4	2015	Apr	Bronx	1.771535e+06	66009.49	3.73
5	2015	Apr	None	5.576637e+05	61795.63	11.08
6	2015	Apr	EWR	6.018036e+04	8031.24	13.35
7	2015	Apr	Staten Island	1.220357e+04	1223.12	10.02
8	2015	Aug	Manhattan	1.495241e+08	15342840.95	10.26
9	2015	Aug	Queens	3.582325e+07	3337728.24	9.32

3. Velocidad y congestión: promedio de mph por franja horaria y borough (viajes diurnos vs. nocturnos).

```
# Q3
QUERY_3 = """
SELECT
    Z.Borough AS pickup_borough,
    D.day_name,
    D.day_of_week, -- Para ordenar la semana
    -- Cálculo de la velocidad promedio en MPH (Millas por hora)
    AVG( (F.trip_distance / NULLIF(F.trip_duration_seconds, 0)) * 3600 ) AS avg_speed_mph
FROM GOLD.fct_trips F
JOIN GOLD.dim_zone Z
    ON F.pu_zone_sk = Z.zone_sk
JOIN GOLD.dim_date D
    ON F.pickup_date_sk = D.date_sk
WHERE
    F.trip_duration_seconds > 0 -- Excluir viajes con duración cero
    AND F.trip_distance > 0     -- Excluir viajes con distancia cero
GROUP BY 1, 2, 3
ORDER BY 3, avg_speed_mph DESC;
"""
execute_query(QUERY_3, "3. VELOCIDAD Y CONGESTION")

```

```
--- 3. VELOCIDAD Y CONGESTION ---
Filas obtenidas: 56
```

	PICKUP_BOROUGH	DAY_NAME	DAY_OF_WEEK	AVG_SPEED_MPH
0	EWR	Sun	0	740.053469
1	None	Sun	0	648.041180
2	Staten Island	Sun	0	92.410885
3	Queens	Sun	0	27.392559
4	Bronx	Sun	0	23.909200
5	Unknown	Sun	0	22.794440
6	Brooklyn	Sun	0	16.691321
7	Manhattan	Sun	0	13.074035
8	EWR	Mon	1	879.915000
9	None	Mon	1	685.977542

4. Duración del viaje: percentiles (p50/p90) de duración por PULocationID (pickup)

```
# Q4
QUERY_4 = """
SELECT
    Z.Zone AS pickup_zone,
    Z.Borough AS pickup_borough,
    -- P50: Mediana de la duración del viaje
    PERCENTILE_CONT(0.50) WITHIN GROUP (ORDER BY F.trip_duration_seconds) AS duration_p50_seconds,
    -- P90: 90% de los viajes son más cortos que esta duración
    PERCENTILE_CONT(0.90) WITHIN GROUP (ORDER BY F.trip_duration_seconds) AS duration_p90_seconds
FROM GOLD.fct_trips F
JOIN GOLD.dim_zone Z ON F.pu_zone_sk = Z.zone_sk
WHERE F.trip_duration_seconds > 0 -- Excluir viajes de duración cero
```

```
GROUP BY 1, 2
ORDER BY duration_p90_seconds DESC;
"""
execute_query(QUERY_4, "4. DURACION DEL VIAJE")
```

```
--- 4. DURACION DEL VIAJE ---
Filas obtenidas: 262
```

	PICKUP_ZONE	PICKUP_BOROUGH	DURATION_P50_SECONDS	DURATION_P90_SECONDS
0	Arden Heights	Staten Island	3682.5	6265.800
1	Far Rockaway	Queens	2820.0	5069.200
2	Heartland Village/Todt Hill	Staten Island	1711.5	4946.000
3	Hammels/Arverne	Queens	2682.5	4800.000
4	Charleston/Tottenville	Staten Island	2954.0	4760.600
5	Bloomfield/Emerson Hill	Staten Island	2562.0	4754.499
6	Great Kills	Staten Island	830.5	4662.899
7	Eltingville/Annadale/Prince's Bay	Staten Island	2042.5	4603.500
8	Rockaway Park	Queens	2160.0	4347.000
9	Mariners Harbor	Staten Island	1404.0	4296.600

5. Elasticidad temporal: distribución de viajes por día de semana y hora; ¿cuáles son las horas pico?

```
# Q5
QUERY_5 = """
SELECT
    D.day_name,
    D.day_of_week, -- 1=Lunes, 7=Domingo (Usado para el orden cronológico)
    COUNT(F.trip_id) AS total_trips
FROM GOLD.fct_trips F
JOIN GOLD.dim_date D ON F.pickup_date_sk = D.date_sk
GROUP BY 1, 2
-- Ordenamos para mostrar los días en orden cronológico, y luego el volumen de viajes
ORDER BY D.day_of_week, total_trips DESC;
"""
execute_query(QUERY_5, "5. Elasticidad Temporal")

```

```
--- 5. Elasticidad Temporal ---
Filas obtenidas: 7
```

	DAY_NAME	DAY_OF_WEEK	TOTAL_TRIPS
0	Sun	0	109748544
1	Mon	1	98291952
2	Tue	2	106511259
3	Wed	3	112913688
4	Thu	4	116910232
5	Fri	5	119392732
6	Sat	6	119018203

```
conn.close()
print("\nConexión a Snowflake cerrada.")
```

```
Conexión a Snowflake cerrada.
```