

Análisis de Multas de Tránsito – Barranquilla 2006 - 2020

Candy Manrique

Cindy Guevara

Gina Ortega

Karen Sanjuan

Lina Piocuda

Martha Coronel

Marcela Luna

Sharon Varelo

Stefany Lozano

Tina Varela

Vanessa Gonzalez

Análisis de Datos – Conexión Mujeres TIC

Tutor: Santiago Giraldo

Fecha de entrega: mayo de 2025

Barranquilla, Atlántico

Índice

- 1. Introducción**
- 2. Objetivos del análisis**
- 3. Hipótesis de trabajo**
- 4. Metodología**
 - 4.1. Fuentes de datos
 - 4.2. Herramientas utilizadas
 - 4.3. Técnicas de análisis aplicadas
- 5. Pasos preliminares realizados**
 - 5.1. Obtención de datos
 - 5.2. Carga de datos en MySQL
 - 5.3. Conexión con Visual Studio Code (Python)
 - 5.4. Exploración y limpieza de datos
- 6. Análisis de datos, resultados y hallazgos**
- 7. Interpretaciones de la evidencia y conclusiones**
 - 7.1. Hipótesis 1 (H1)
 - 7.2. Hipótesis 2 (H2)
 - 7.3. Hipótesis 3 (H3)
- 8. Referencias**

1. Introducción

En los últimos años, el aumento del número de vehículos y la dinámica del tránsito urbano han generado una creciente preocupación por las infracciones viales en ciudades como Barranquilla. Este fenómeno no solo refleja comportamientos individuales, sino posibles falencias en los mecanismos de control y sanción. A partir del análisis de información recopilada entre 2006 y 2020, este trabajo busca examinar cómo ha evolucionado la imposición y el pago de las multas de tránsito, prestando atención especial al tipo de vehículo involucrado, la frecuencia de las sanciones y la respuesta de los infractores ante el sistema de cobro.

En el presente trabajo se presentan los hallazgos obtenidos durante el análisis de datos, lo que permite una mejor comprensión del contexto vial y de la efectividad de las medidas implementadas en la ciudad.

2. Objetivo del análisis

Analizar el comportamiento de las multas de tránsito registradas en Barranquilla entre 2006 y 2020 utilizando datos provenientes del Sistema Integrado de Información sobre Multas y Sanciones por Infracciones de Tránsito (**SIMIT**), identificando tendencias en la cantidad de infracciones, tasa de pago, efectividad en el cobro, tipo de vehículo y reincidencia, con el fin de comprender el perfil del infractor y apoyar estrategias de control y recaudo.

3. Hipótesis

1. La cantidad de multas tiende a aumentar año tras año.
2. Los carros acumulan mayor número de multas que las motos.
3. La efectividad en el cobro de multas por parte de (**SIMIT**) es baja, especialmente en vehículos que presentan múltiples sanciones.

4. Metodología

4.1. Fuente de datos

La información utilizada para este análisis proviene de una fuente pública externa: el conjunto de datos titulado “*Multas **SIMIT** - Historial de Multas reportados en el Sistema Integrado de Información sobre Multas y Sanciones por Infracciones de Tránsito (**SIMIT**)*”, disponible en el portal oficial del gobierno colombiano: datos.gov.co. Se seleccionaron específicamente los registros que corresponden a la ciudad de Barranquilla por pertinencia, cubriendo el periodo comprendido entre los años 2006 a 2020.

4.2. Herramientas utilizadas

- a. **MySQL**: Utilizado para la carga y estructuración de la base de datos, permitiendo la importación masiva de datos y la organización adecuada de la información.
- b. **Python en Visual Studio Code (Pandas, Matplotlib, Seaborn)**: Empleado procesamiento, limpieza, transformación, análisis estadístico y visualización de datos, así como la creación de visualizaciones y gráficos.
- c. **Power BI**: Utilizado para la creación de dashboards interactivos y presentación de resultados del análisis.

4.3. Técnicas de análisis aplicadas

- d. **Análisis de datos descriptivo**: Se calcularon estadísticas básicas (media, mediana) para entender la distribución de los datos.
- e. **Análisis temporal**: Se analizó la evolución de las multas a lo largo del tiempo, realizando agrupaciones por año y mes.
- f. **Segmentación y agrupación**: Se agruparon datos por variables para identificar patrones y diferencias significativas entre años, además del tipo de vehículo y evaluar diferencias entre los distintos grupos.

5. Pasos realizados

5.1. Obtención de datos:

Se descargaron los datos en formato .csv desde el portal de datos.gov.co, específicamente la base de datos de multas del SIMIT correspondientes a la ciudad de Barranquilla, la información contenía variables como la vigencia, la placa del vehículo, el monto de la multa, la fecha de infracción y el estado de pago (Si - No). Total de registros (2.496.602).

5.2. Carga de datos en MySQL:

Para la carga y estructuración de los datos, se creó una base de datos en MySQL denominada **multas_simitbq**, la cual almacenará la información. Dentro de esta base de datos, se creó una tabla llamada **multas** con las columnas (*Vigencia*, *Placa*, *Fecha_Multa*, *Valor_Multa*, *Ciudad*, *Pagado_Si_No*). Ver **Figura 1**. para observar el código de creación de base de datos y tabla.

```
1 • CREATE DATABASE IF NOT EXISTS multas_simitbq;
2 • USE multas_simitbq;
3 • CREATE TABLE IF NOT EXISTS multas (
4     VIGENCIA VARCHAR(10),
5     PLACA VARCHAR(20),
6     FECHA_MULTA DATE,
7     VALOR_MULTA DECIMAL(10, 2),
8     CIUDAD VARCHAR(50),
9     PAGADO_SI_NO VARCHAR(5));
```

Figura 1. Código utilizado en MySQL para la creación de la base de datos y la tabla con las columnas.

Una vez creada la base de datos y la tabla, se utilizó **LOAD DATA INFILE** para cargar el archivo .csv en MySQL Workbench, esta decisión es debido a que la herramienta de importación estándar (Import Wizard) presentó fallas por el tamaño del archivo y el número de registros. El uso de **LOAD DATA INFILE** optimizó el servidor, permitiendo una importación más eficiente y facilitando la carga rápida de los datos. Se utilizó un solo comando en SQL y se dividió el archivo en 3 partes, además de realizar modificaciones en el formato de fecha. Ver **Figura 2** para observar el código utilizado.

```

10 • SHOW VARIABLES LIKE 'secure_file_priv';
11 • LOAD DATA INFILE 'C:/ProgramData/MySQL/MySQL Server 8.0/Uploads/multas_simitbq-1.csv'
12 INTO TABLE multas
13 FIELDS TERMINATED BY ';'
14 ENCLOSED BY '"'
15 LINES TERMINATED BY '\n'
16 IGNORE 1 LINES
17 (@vigencia, @placa, @fecha_multa, @valor_multa, @ciudad, @pagado_si_no)
18 SET
19     vigencia = TRIM(@vigencia),
20     placa = TRIM(@placa),
21     fecha_multa = STR_TO_DATE(TRIM(@fecha_multa), '%Y-%m-%d'),
22     valor_multa = NULLIF(TRIM(@valor_multa), ''),
23     ciudad = TRIM(@ciudad),
24     pagado_si_no = TRIM(@pagado_si_no);

```

Figura 2. Código utilizado para cargar archivo CSV a la base de datos y tabla.

A través de la siguiente tabla (**Figura 3.**), se verificó la carga de los datos, asegurando que todas las columnas y la información se pudiera visualizar y fuera coherente con el archivo original.

| | VIGENCIA | PLACA | FECHA_MULTA | VALOR_MULTA | CIUDAD | PAGADO_SI_NO |
|---|----------|--------|-------------|-------------|--------------|--------------|
| ▶ | 2007 | RBH343 | 2007-05-17 | 115800.00 | Barranquilla | SI |
| | 2007 | JAV837 | 2007-05-17 | 216900.00 | Barranquilla | NO |
| | 2007 | ZST89A | 2007-05-17 | 115800.00 | Barranquilla | NO |
| | 2007 | ZST89A | 2007-05-17 | 216900.00 | Barranquilla | NO |
| | 2007 | BJA333 | 2007-05-17 | 115800.00 | Barranquilla | SI |
| | 2007 | ZCA299 | 2007-05-17 | 216900.00 | Barranquilla | SI |
| | 2007 | BBB51B | 2007-05-17 | 216900.00 | Barranquilla | NO |
| | 2007 | GEZ83A | 2007-05-17 | 216900.00 | Barranquilla | NO |
| | 2007 | QHI767 | 2007-05-17 | 433800.00 | Barranquilla | NO |
| | 2007 | UYU015 | 2007-05-17 | 216900.00 | Barranquilla | NO |
| | 2007 | UYV689 | 2007-05-17 | 216900.00 | Barranquilla | NO |
| | 2007 | EUZ173 | 2007-05-17 | 216900.00 | Barranquilla | NO |
| | 2007 | QHC338 | 2007-05-17 | 433800.00 | Barranquilla | SI |
| | 2007 | SRL484 | 2007-05-17 | 216900.00 | Barranquilla | NO |
| | 2007 | GFP46A | 2007-05-17 | 115800.00 | Barranquilla | NO |
| | 2007 | RGA43 | 2007-05-17 | 216900.00 | Barranquilla | NO |
| | 2007 | SPO689 | 2007-05-17 | 216900.00 | Barranquilla | SI |
| | 2007 | QGU874 | 2007-05-17 | 216900.00 | Barranquilla | SI |

Figura 3. Verificación de la carga de datos en la tabla multas en MySQL.

Con la intención de confirmar que la totalidad de los registros fuera cargada completamente en la base de datos, se ejecutó la consulta *SELECT COUNT (*)FROM multas_simitbq.multas*. Ver **Figura 4**.

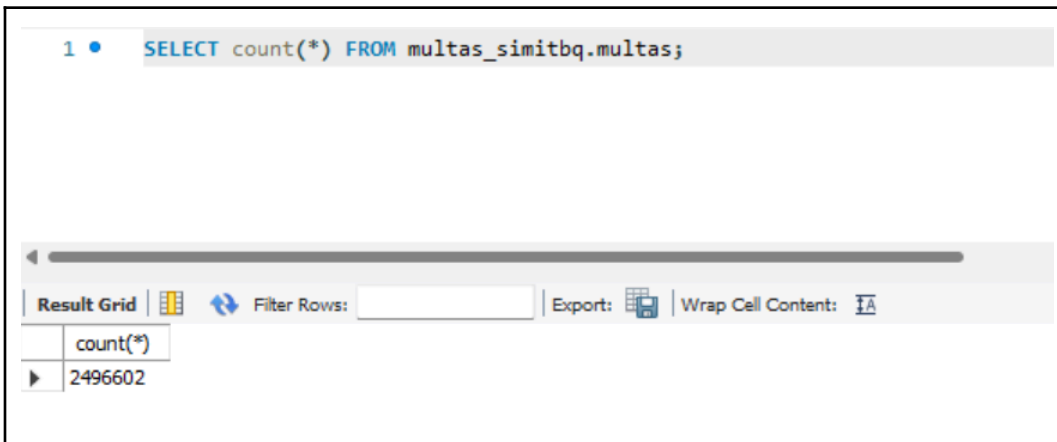


Figura 4. Consulta de validación del número total de registros en la tabla multas.

5.3. Conexión de base de datos desde MySQL con Visual Studio Code

Para iniciar en el entorno de desarrollo en Python, se realizó la importación de las principales librerías necesarias para los cálculos y las gráficas. Ver **Figura 5**.

```
1  import pymysql
2  import pandas as pd
3  import numpy as np
4  import matplotlib.pyplot as plt
5  import seaborn as sns
6
7
8  conn = pymysql.connect(
9      host="127.0.0.1",
10     port=3306,
11     user="root",
12     password="XXXXX",
13     database="multas_simitbq"
14 )
15
16 cursor = conn.cursor()
17 cursor.execute("SELECT * FROM multas;")
18 resultados = cursor.fetchall()
19
```

Figura 5. Conexión de la base de datos MySQL con Visual Studio Code mediante Python e importación de librerías.

5.4. Exploración y limpieza de datos con Python

Se llevó a cabo un proceso de exploración y limpieza de datos utilizando librerías **pandas** y **numpy**. Inicialmente se examinaron los tipos de datos, se actualizaron los tipos de datos adecuados para el tratamiento de la información. También se identificaron los registros duplicados y valores nulos, los cuales fueron tratados y eliminados para asegurar la calidad de los datos. Adicionalmente, se creó una columna para identificar el tipo de vehículo según su placa para la comprobación de las hipótesis. Al final de este paso, se guardó la información actualizada en un archivo .csv para continuar el tratamiento de datos. Ver código en la **Figura 6** y los resultados en la **Figura 7**.

```
20 print(" Análisis de Multas de Tránsito  Barranquilla 2006-2020")
21 #paso 1 Exploración de datos y Limpieza
22
23 columnas = ["VIGENCIA", "PLACA", "FECHA_MULTA", "VALOR_MULTA", "CIUDAD", "PAGADO_SI_NO"]
24 df = pd.DataFrame(resultados, columns=columnas)
25
26 print(df)
27 print(df.shape)
28 print(df.dtypes)
29
30 # Convertir la fecha a datetime
31 df['FECHA_MULTA'] = pd.to_datetime(df['FECHA_MULTA'], errors='coerce')
32
33 df = df.astype (
34     {
35         'VIGENCIA': 'int',
36         'PLACA': 'string',
37         'VALOR_MULTA': 'int',
38         'CIUDAD': 'string',
39         'PAGADO_SI_NO': 'string',})
40
41 print(df.info())
42
43 # Crear columna TIPO_VEHICULO según el último carácter de la placa
44 df['TIPO_VEHICULO'] = df['PLACA'].str.strip().str.upper().apply(
45     lambda x: 'MOTO' if x[-1].isalpha() else 'CARRO')
46 df['TIPO_VEHICULO'] = df['TIPO_VEHICULO'].astype('category')
47 print(df['TIPO_VEHICULO'].value_counts())
48
49 #Antes de eliminar
50 print(f"Duplicados antes de limpiar: {df.duplicated().sum()}")
51
52 print(df.duplicated())
53 # Eliminar duplicados
54 df = df.drop_duplicates()
55
56 # Después de eliminar
57 print(f"Duplicados después de limpiar: {df.duplicated().sum()}")
58 # Verificar valores nulos por columna
59 print(df.isnull().sum())
60 # Eliminar filas con valores nulos
61 df = df.dropna()
62
63 # Verificar que no haya valores nulos
64 print(df.isnull().sum())
65
66 #Guardar base de datos limpia con los cambios aplicados
67 df.to_csv('multas_simitbq_limpio.csv', index=False)
68
```

Figura 6. Proceso de exploración y limpieza de datos en Python.

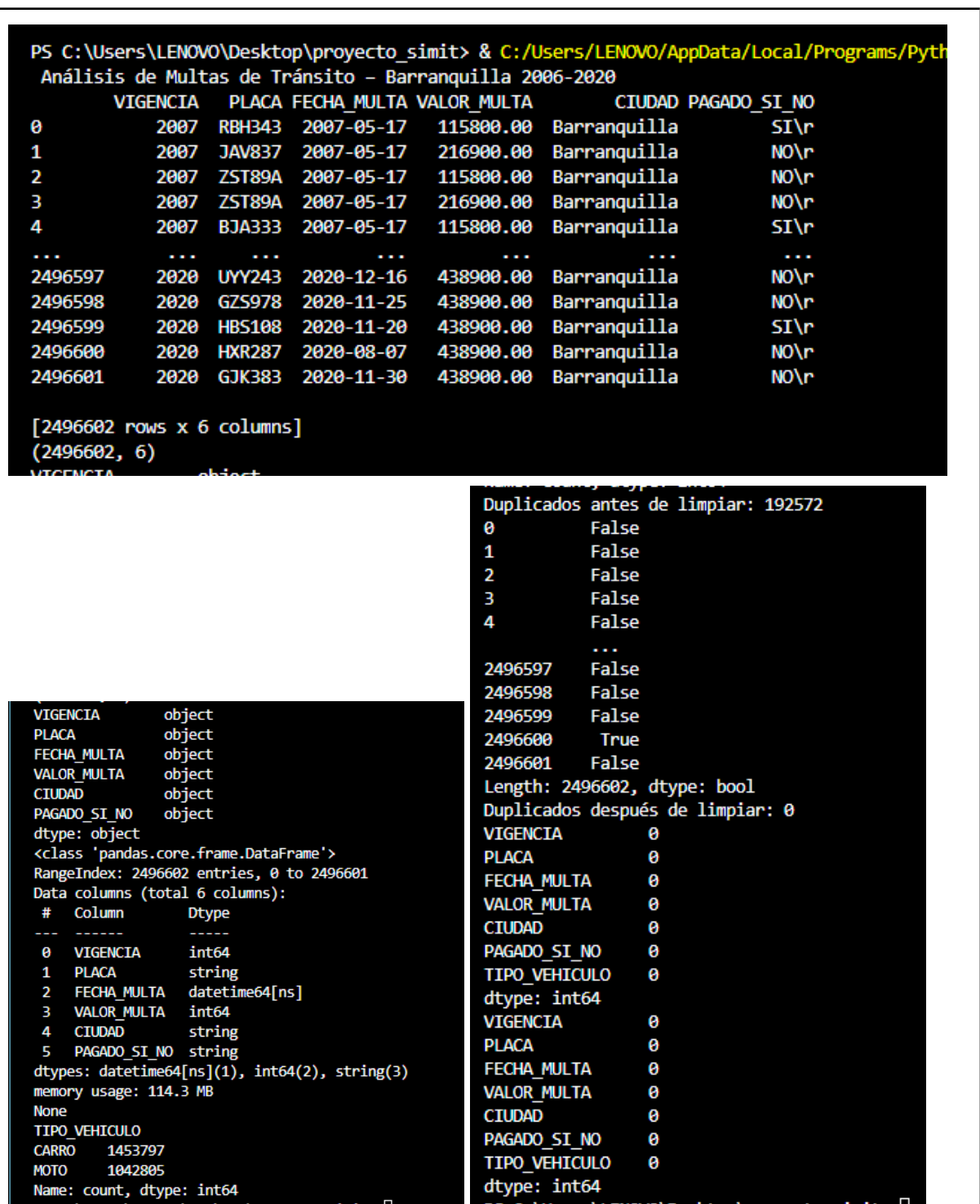


Figura 7. Resultados preliminares de la exploración y limpieza de datos: duplicados, nulos y tipos de datos corregidos.

6. Análisis de datos, resultados y hallazgos

Una vez finalizado el proceso de la limpieza, se procedió al análisis estadístico de los datos para responder a las preguntas planteadas y las hipótesis. Entre los cálculos realizados se encuentran: el número de multas registradas por año, el monto total recaudado por año, proporción de multas pagadas vs. no pagadas, efectividad, tipo de vehículo que acumula más multas, promedio multas, multa más baja, multa más alta, vehículos reincidentes. Ver código en **Figura 8**.

```
70 #Punto 2 - Analisis de datos Calculos
71
72 # 1 Cuantas multas se registraron por año?
73 multas_por_año=df.groupby(df["FECHA_MULTA"].dt.year).size()
74
75 print("\nMultas por año:", multas_por_año)
76
77 # 2 Cual es el monto total recaudado por año?
78 monto_por_año = df.groupby(df["FECHA_MULTA"].dt.year)["VALOR_MULTA"].sum()
79 print("\nMonto por año:",monto_por_año)
80
81 # 3 Cuál es la proporción de multas pagadas vs. no pagadas?
82 proporcion = df["PAGADO_SI_NO"].value_counts(normalize=True) *100
83 print(proporcion)
84
85 df['PAGADO_SI_NO'] = df['PAGADO_SI_NO'].str.strip().str.upper()
86 # 4 Calcular efectividad (%)
87 pagadas = (df['PAGADO_SI_NO'] == 'SI').sum()
88 total = len(df)
89 efectividad = (pagadas / total) * 100
90 print(f"\nEfectividad en el cobro de multas: {efectividad:.2f}%")
91
92 # 5 Tipo de vehículo que acumula más multa
93 multas_por_tipo = df.groupby("TIPO_VEHICULO", observed=True)["PLACA"].count()
94 tipo_con_mas_multas = multas_por_tipo.idxmax()
95 cantidad = multas_por_tipo.max()
96 print(f"\nEl tipo de vehículo que tiene más multas es {tipo_con_mas_multas}, con un total de {cantidad} multas.")
97
98 # 6 ¿Qué mes del año se imponen más multas?
99 df["mes"] = df["FECHA_MULTA"].dt.month
100 df["año"] = df["FECHA_MULTA"].dt.year
101 multas_por_año_mes = df.groupby(["año", "mes"]).size().reset_index(name="cantidad")
102 mes_max_por_año = multas_por_año_mes.loc[
103 multas_por_año_mes.groupby("año")["cantidad"].idxmax()]
104 print(f"\nEl mes por año que mas impone:{mes_max_por_año}")
105
106 # 7 Convertir valores a array de numpy
107 multas_array = multas_por_año.values
108
109 # Calcular diferencia entre años consecutivos
110 cambio_anual = np.diff(multas_array)
111 print("\nCambio de multas año tras año:")
112 print(cambio_anual)
```

```

114 # Calcular porcentaje de cambio año tras año
115 porcentaje_cambio = (cambio_anual / multas_array[:-1]) * 100
116 print("\nPorcentaje de cambio de multas año tras año:")
117 print(porcentaje_cambio)
118
119 # Contar cuántos años aumentaron
120 años_que_aumentaron = np.sum(cambio_anual > 0)
121 años_totales = len(cambio_anual)
122 print(f"\nAños que aumentó el número de multas: {años_que_aumentaron} de {años_totales}")
123
124 # 8 calculo de porcentaje de vehiculos reincidentes
125 # Contar cuántas veces aparece cada placa
126 vehiculos_reincidentes = df["PLACA"].value_counts()
127 # Total de vehículos únicos (placas distintas)
128 total_vehiculos_unicos = len(vehiculos_reincidentes)
129 # Contar cuántos vehículos tienen más de una multa (reincidentes)
130 cantidad_reincidentes = (vehiculos_reincidentes > 1).sum()
131 # Calcular el porcentaje de reincidencia
132 porcentaje_reincidencia = (cantidad_reincidentes / total_vehiculos_unicos) * 100
133 # Mostrar los resultados
134 print(f"\nTotal de vehículos únicos: {total_vehiculos_unicos}")
135 print(f"\nVehículos reincidentes (más de una multa): {cantidad_reincidentes}")
136 print(f"\nPorcentaje de vehículos reincidentes: {porcentaje_reincidencia:.2f}%")
137
138 #otros calculos
139
140 promedio_multas = np.mean(df["VALOR_MULTA"])
141 print(f"\nPromedio del valor de las multas: {promedio_multas:.2f}")
142
143 max_multa = np.max(df["VALOR_MULTA"])
144 min_multa = np.min(df["VALOR_MULTA"])
145 print(f"\nMulta más alta: {max_multa}")
146 print(f"\nMulta más baja: {min_multa}")
147
148 vehiculos_reincidentes = df["PLACA"].value_counts()
149 cantidad_reincidentes = (vehiculos_reincidentes > 1).sum()
150 print(f"\nVehículos reincidentes (más de una multa): {cantidad_reincidentes}")
151

```

Figura 8. Análisis de datos y cálculos estadísticos realizados en Python.

7. Interpretaciones de la evidencia y conclusiones:

Basándonos en los resultados obtenidos durante el proceso de investigación y análisis de los datos llegamos a las siguientes interpretaciones y conclusiones:

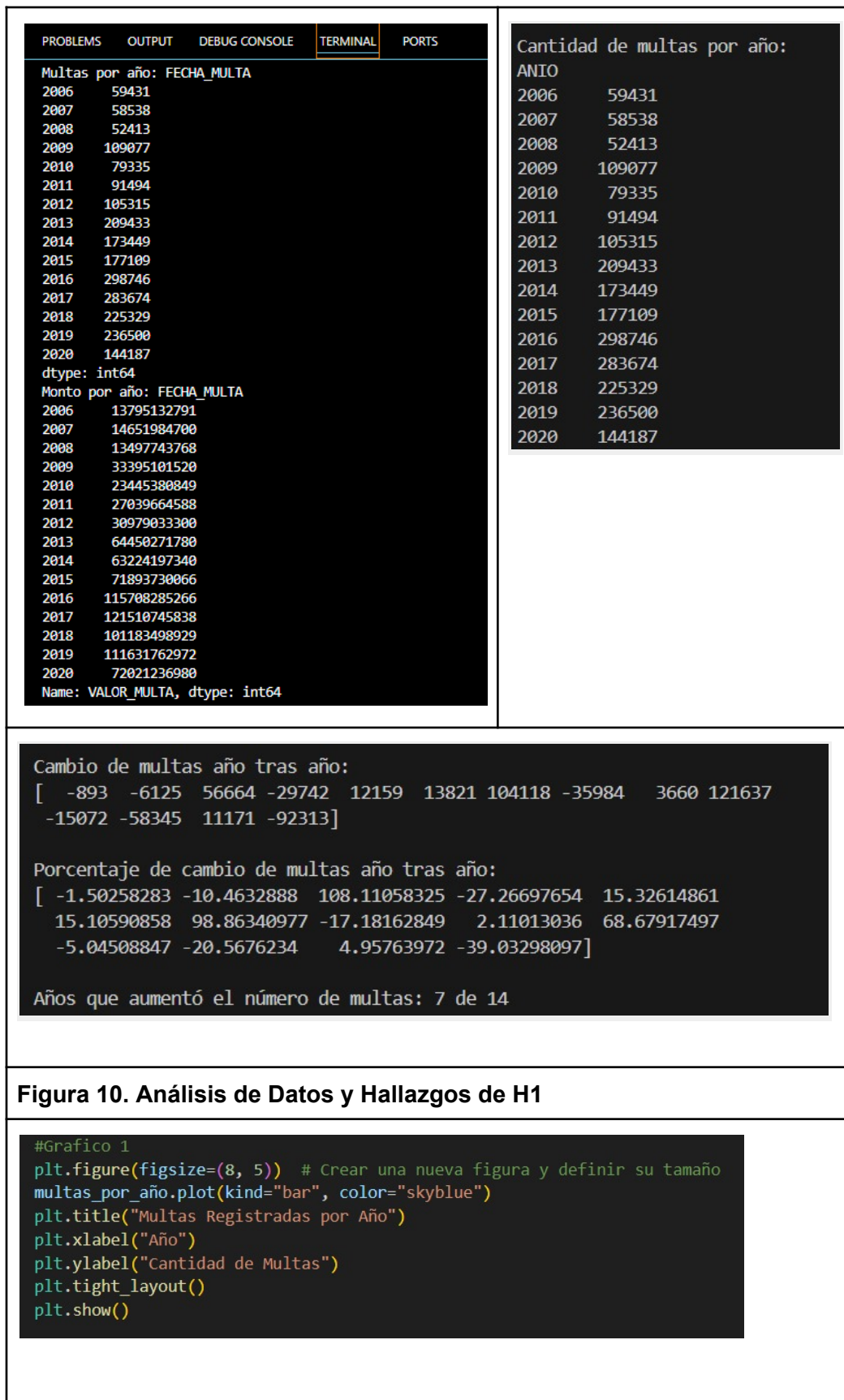
7.1. Hipótesis 1 (H1): “La cantidad de multas tiende a aumentar año tras año.”, para comprobar esta hipótesis, se formularon y respondieron las siguientes preguntas claves:

- ¿Cuántas multas se registran por año?
- ¿Cambio de multas año tras año?
- ¿Qué porcentaje de aumento o disminución se presenta cada año?
- ¿En cuáles años aumentó?,

El análisis fue desarrollado mediante scripts implementados en Visual Studio Code y la posterior visualización de los resultados en power bi, lo que permitió representar los siguientes códigos utilizados:

```
multas_por_año=df.groupby(df["FECHA_MULTA"].dt.year).size()
multas_array = multas_por_año.values
cambio_anual = np.diff(multas_array)
porcentaje_cambio = (cambio_anual / multas_array[:-1]) * 100
años_que_aumentaron = np.sum(cambio_anual > 0)
```

Figura 9. Código para visualización de la Hipótesis 1.



```

163 #Grafico 2
164 plt.figure()
165 monto_por_año.plot(kind="bar", color="green")
166 plt.title("Monto Total Recaudado por Año")
167 plt.xlabel("Año")
168 plt.ylabel("Monto")
169 plt.tight_layout()
170 plt.show()

```

Figura 11. Código de la gráfica Hipótesis H1.

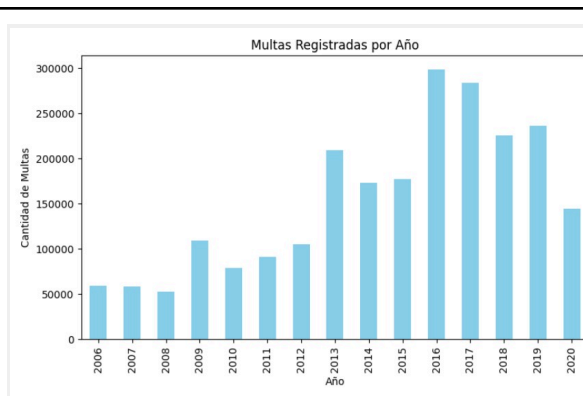


Figura 12. Gráfico de Multas Registradas por año.

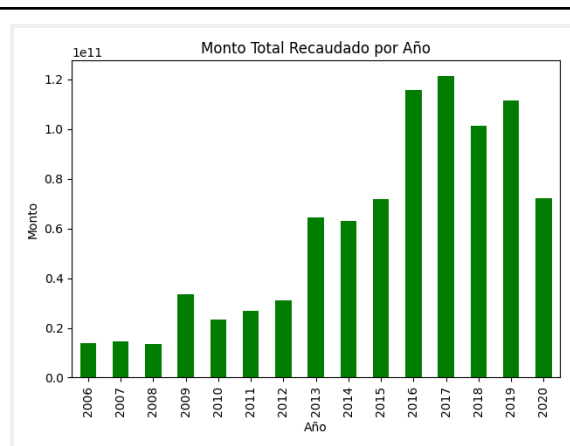


Figura 13. Gráfico de Monto Total Recaudado por año.

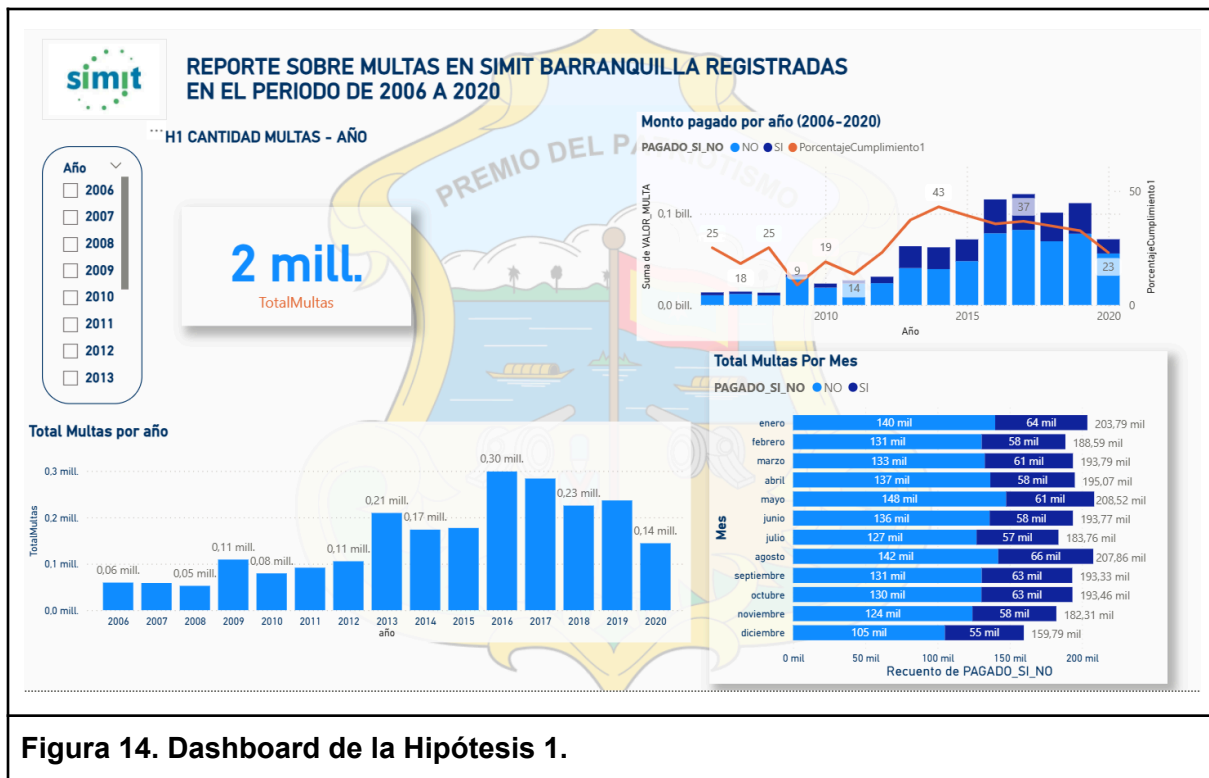


Figura 14. Dashboard de la Hipótesis 1.

Conclusión de la hipótesis 1 (H1): “La cantidad de multas tiende a aumentar año tras año.”

Con base en el análisis de los datos, se obtuvieron las siguientes evidencias:

- Se observaron incrementos de datos, significativos en los siguientes años (por ejemplo +108%, +98%, +68%).
- Se registraron caídas en otros periodos (por ejemplo. -27%, -39%).
- Solo 7 de 14 años se evidenció un aumento en la cantidad de multas superior al 50%.
- En general, el comportamiento no refleja tendencia de crecimiento constante, sino un patrón fluctuante e irregular

Este hallazgo justifica la pertenencia de la investigación, ya que permite comprender el comportamiento de las infracciones de tránsito en la ciudad de Barranquilla. lo cual estaría influenciado por los siguientes factores:

- Aumento o reducción de operaciones de control vial.
- Cambios en el comportamiento de los conductores.
- Variaciones en la normativa o mecanismos de transporte.

7.2. Hipótesis (H2) “Los carros acumulan mayor número de multas que las motos.”

Para investigar esta hipótesis, se planteó la siguiente pregunta:

- ¿Qué tipo de vehículo acumula más multas?

El análisis fue desarrollado mediante script implementado en Visual Studio Code, permitiendo procesar los datos de forma precisa y estructurada.

A continuación, se presenta el código utilizado para responder a esta pregunta:

```
multas_por_tipo = df.groupby("TIPO_VEHICULO", observed=True)["PLACA"].count()
tipo_con_mas_multas = multas_por_tipo.idxmax()
cantidad = multas_por_tipo.max()
```

Figura 15. Código para obtener el tipo de vehículo que acumula más multas.

Nota: La variable TIPO_VEHICULO fue correctamente creada con base en el último carácter de la placa (convención en Colombia).

Análisis de Datos y Hallazgos H2:

```
Cantidad de registros por tipo de vehículo:
TIPO_VEHICULO
CARRO      1370500
MOTO       933530
Name: count, dtype: int64
```

Figura 16. Resultado obtenido en Visual Studio Code para la H2.

```
#Grafico 5
sns.countplot(data=df, x="TIPO_VEHICULO", hue="TIPO_VEHICULO", palette="Set2", order=df["TIPO_VEHICULO"].value_counts().index)
plt.title("Multas por Tipo de Vehículo")
plt.xticks(rotation=45)
plt.ylabel("Cantidad")
plt.show()
```

Figura 17. Script para obtener gráficas de multas por tipo de vehículo.

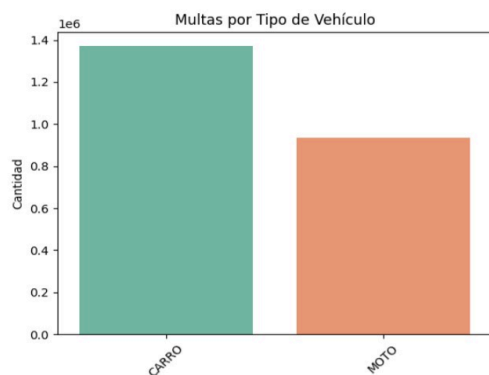


Figura 18. Análisis gráfico complementario H2.

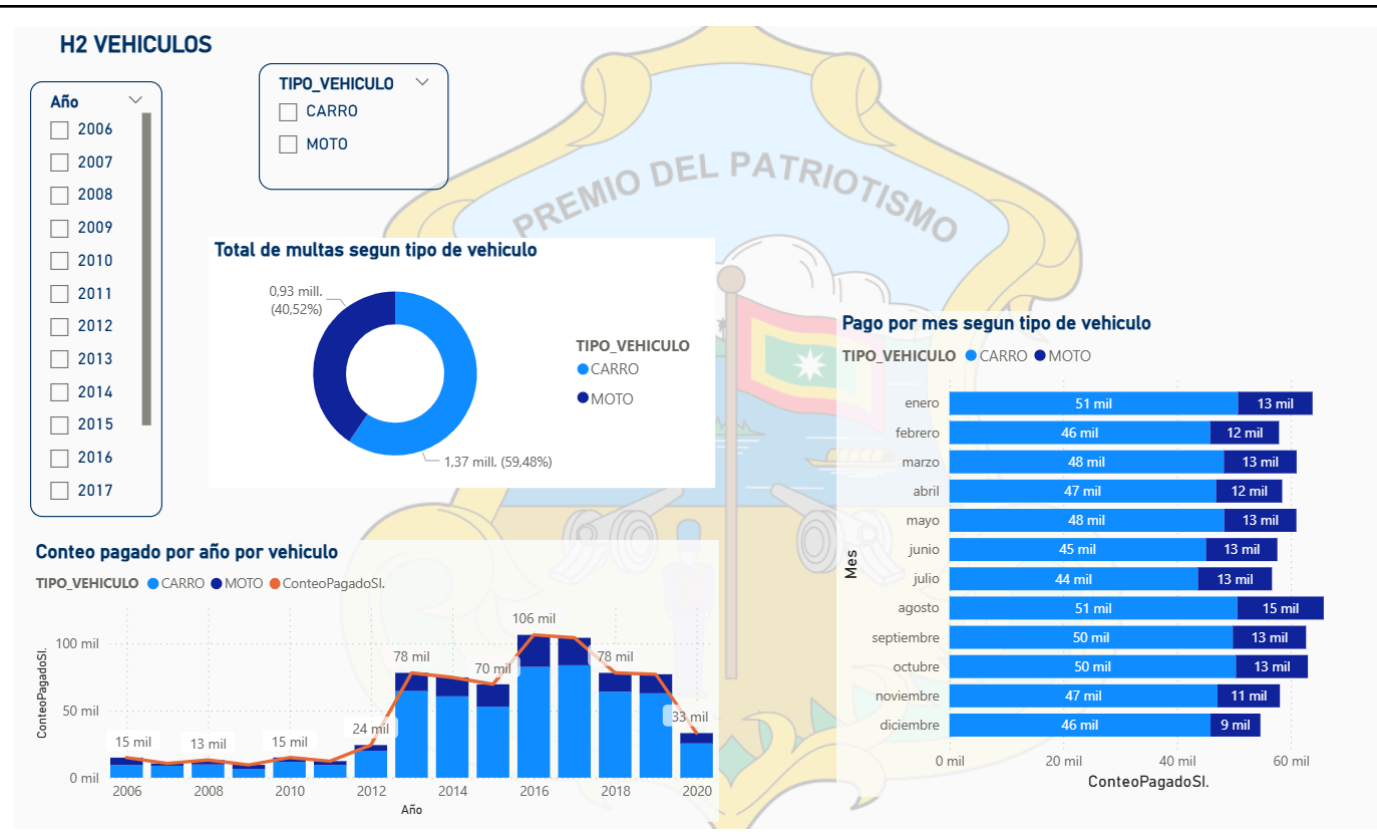


Figura 19. Dashboard de Hipótesis 2

Conclusión de la hipótesis 2 (H2):

El análisis de la hipótesis H2: “Los carros acumulan mayor número de multas que las motos.”

El análisis de los datos arrojó el siguiente resultado:

- Carros: 1,370,500 multas (59.5%).

- Motos: 933,530 multas (40%).

La hipótesis H2 se confirma con los datos obtenidos. Los carros representan la mayor proporción de infracciones registradas en comparación con otras motos. Esta información es significativa, ya que permite entender cómo se distribuyen las multas según el tipo de vehículo. Además, aporta elementos clave para el diseño de políticas de control diferenciadas por segmento vehicular, lo cual puede mejorar la eficiencia de la gestión del tránsito en la ciudad de Barranquilla.

7.3. Hipótesis 3 (H3): “La efectividad en el cobro de multas es baja, especialmente en reincidentes”,

Para abordar esta hipótesis, se formularon las siguientes preguntas:

- ¿Cuál es la proporción de multas pagadas vs.no pagadas?
- ¿Cuál es la efectividad general en el cobro?
- ¿Cuántos vehículos son reincidentes?
- ¿Qué porcentaje del parque sanciona a los reincidentes?,

El tratamiento y análisis de estas preguntas se realizó utilizando scripts desarrollados en Visual Studio Code.

A continuación en la **Figura 20.** presentamos el código utilizado para obtener estos resultados.

```
# paso 13 la efectividad de cobro es baja
# Contar cuántas multas fueron pagadas
pagadas = (df['PAGADO_SI_NO'] == 'SI').sum()
# Contar el total de multas
total = len(df)
# Calcular efectividad (%)
efectividad = (pagadas / total) * 100
# Mostrar resultado
print(f"\nEfectividad en el cobro de multas: {efectividad:.2f}%")

# Paso 14 calculo de porcentaje de vehículos reincidentes
# Contar cuántas veces aparece cada placa
vehiculos_reincidentes = df["PLACA"].value_counts()
# Total de vehículos únicos (placas distintas)
total_vehiculos_unicos = len(vehiculos_reincidentes)
# Contar cuántos vehículos tienen más de una multa (reincidentes)
cantidad_reincidentes = (vehiculos_reincidentes > 1).sum()
# Calcular el porcentaje de reincidencia
porcentaje_reincidencia = (cantidad_reincidentes / total_vehiculos_unicos) * 100
# Mostrar los resultados
print(f"Total de vehículos únicos: {total_vehiculos_unicos}")
print(f"Vehículos reincidentes (más de una multa): {cantidad_reincidentes}")
print(f"Porcentaje de vehículos reincidentes: {porcentaje_reincidencia:.2f}%")
```

Figura 20. Código hipótesis H3.

Análisis de Datos y Hallazgos para H3:

```
Name: VALOR_MULTA, dtype: int64
PAGADO_SI_NO
NO    68.715729
SI     31.284271
Name: proportion, dtype: float64

Efectividad en el cobro de multas: 31.28%
Total de vehículos únicos: 573986
Vehículos reincidentes (más de una multa): 363808
Porcentaje de vehículos reincidentes: 63.38%
```

Figura 21. Resultado de la Hipótesis 3.

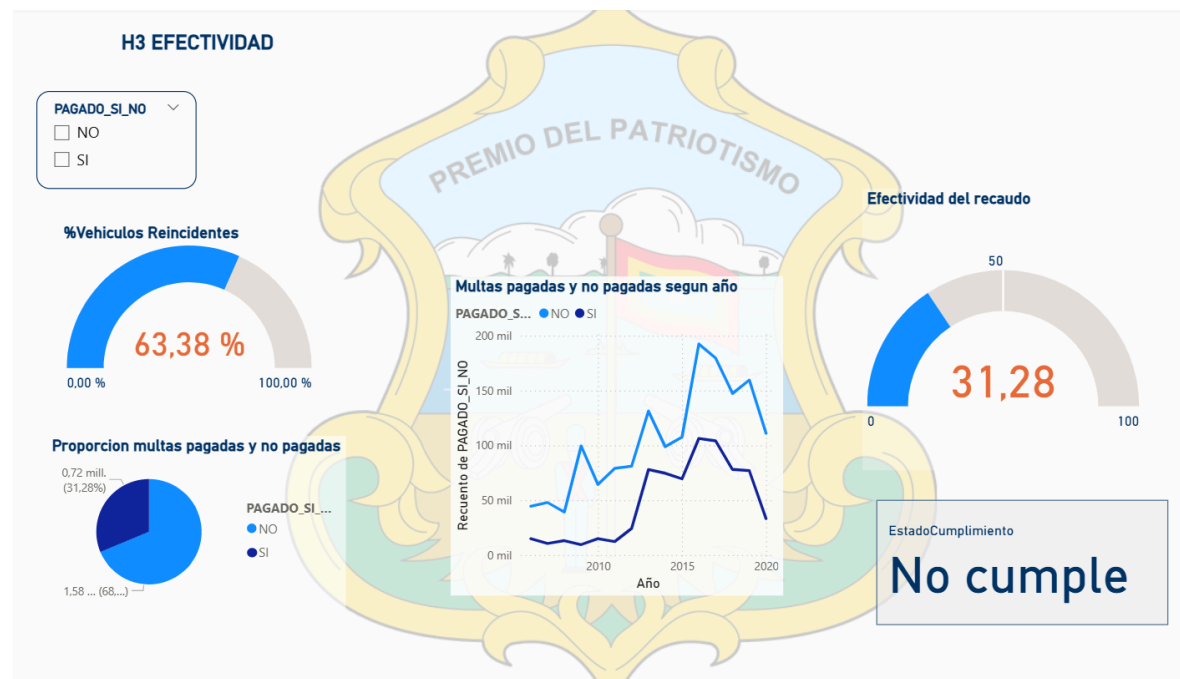


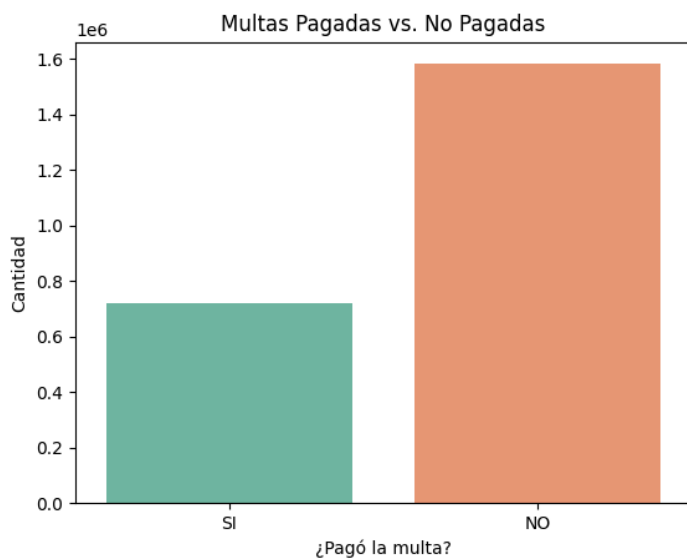
Figura 22. Dashboard de Hipótesis H3.

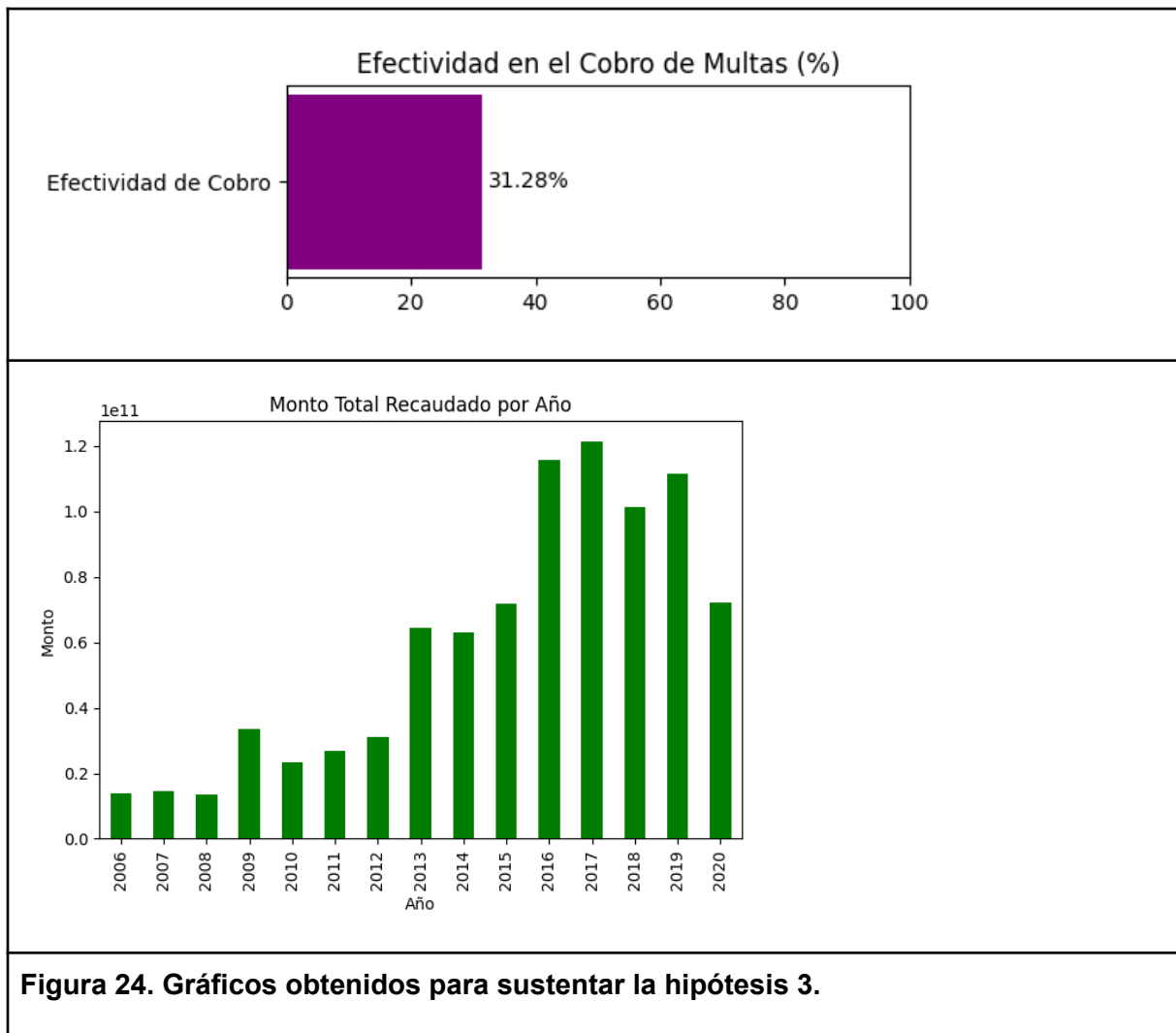
Análisis gráfico complementario de la H3:

```
sns.countplot(data=df, x="PAGADO_SI_NO", hue="PAGADO_SI_NO", palette="Set2")  
plt.title("Multas Pagadas vs. No Pagadas")  
plt.xlabel("¿Pagó la multa?")  
plt.ylabel("Cantidad")  
plt.show()
```

```
plt.figure(figsize=(6, 2))  
plt.barh(["Efectividad de Cobro"], [efectividad], color="purple")  
plt.xlim(0, 100)  
plt.title("Efectividad en el Cobro de Multas (%)")  
plt.text(efectividad + 1, 0, f"{efectividad:.2f}%", va='center') # Etiqueta del valor  
plt.tight_layout()  
plt.show()
```

Figura 23. Script para obtener gráficos de la H3 en Visual Code.





Conclusión de la hipótesis 3 (H3):

“La efectividad en el cobro de multas es baja, especialmente en reincidentes.”, El análisis realizado arrojó los siguientes resultados claves:

- Solo **31.28% fueron pagadas**, es decir 7 de cada 10 de las multas no fueron pagadas
- Existen **363,808** vehículos reincidentes, es decir, con más de una multa
- La efectividad de cobro no mejora significativamente en el periodo analizado.
- La alta reincidencia puede estar vinculada a **baja probabilidad de pago**.

Indicadores:

| Indicador | Valor obtenido | Umbral referencia | de | Evaluación |
|----------------------|----------------|-------------------|----|------------|
| Efectividad de cobro | 31.28% | < 50% | | Baja |

| | | | | |
|----------------------------|----|--------|-----------|------|
| Porcentaje reincidencia | de | 63.38% | > 25%-30% | Alta |
|----------------------------|----|--------|-----------|------|

Lo anterior podemos justificarlo basado en los **informes emitidos por la Contraloría General de la República (CGR)** sobre SIMIT (**ver referencias**) y los informes adicionales de otras auditorías a niveles departamentales por parte de la entidad, se estableció que la efectividad de cobro si es inferior al **50%, es baja** y hay debilidades en la gestión de cobro de multas de tránsito, a su vez establecen que un **porcentaje de reincidencia superior al 25%-30%** se considera **alto**, ya que indica que el sistema no está disuadiendo de manera efectiva a los infractores. Por lo tanto, una tasa de efectividad del 31% en el cobro de multas de tránsito en Barranquilla, y esto acompañado de una **alta reincidencia (63.38%)** es decir, más de la mitad de los infractores tienen más de una multa, esto demuestra que el cobro no está logrando disuadir a los infractores frecuentes. revelarían no sólo una ineficiencia administrativa en el recaudo, sino también una debilidad en la capacidad del sistema SIMIT para corregir conductas infractoras repetidas, afectando directamente la seguridad vial y la percepción de autoridad.

8. Referencias

Contraloría Distrital de Medellín. (2022). *Informe definitivo auditoría de cumplimiento: Gestión cobro de multas de tránsito*.
<https://www.cdm.gov.co/cgm/Paginaweb/IP/Informes%20de%20Auditora%20PVCFT%202022/Informe%20Definitivo%20Auditor%C3%ADa%20de%20Cumplimiento%20Gesti%C3%B3n%20Cobro%20de%20Multas%20de%20Tr%C3%A1nsito.pdf>

Contraloría General de la República. (2020). *Informe de auditoría de cumplimiento No. 013 de 2020: Federación Colombiana de Municipios – SIMIT*.
<https://www.fcm.org.co/wp-content/uploads/2022/02/Informe-CGR-CDSI-No.-013-de-2020-Auditoria-de-Cumplimiento-2019-FCM-SIMIT-1.pdf>

El Espectador. (2025, enero). *Hallazgos fiscales en Secretaría de Movilidad por ineficiencia en cobros de multas*.
<https://www.elespectador.com/bogota/hallazgos-fiscales-en-secretaria-de-movilidad-por-ineficiencia-en-cobros-de-multas/>

Federación Colombiana de Municipios. (s. f.). *Multas SIMIT – Historial de multas reportados en el Sistema Integrado de Información sobre Multas y Sanciones por Infracciones de Tránsito – SIMIT*. Datos.gov.co.
<https://www.datos.gov.co/Funci-n-p-blica/Multas-SIMIT/bgfy-53qq>