

# Universidad Nacional De La Matanza

## Especialización Ciencia de Datos

Captura y Almacenamiento de Datos

**Profesor: Marcelo Caiafa**

## **EDA de la Demanda de Viajes SUBE - Año 2024/25**

Una mirada a los patrones de movilidad en el  
transporte público argentino

### **Integrantes Grupo 4:**

Fica Millán, Yesica - DNI 27624.956

Petraroia, Franco - DNI 27.161.862

Miranda Charca, Florencia - DNI 41.398.768

De Los Ríos, Raúl - DNI 37.741.686

Contenido

1. Descripción del caso ..... 6

    1.1. Problemática planteada ..... 6

    1.2. Contexto del análisis..... 6

    1.3. Hipótesis: ..... 7

    1.4. Objetivo ..... 7

    1.5. Preguntas clave a responder: ..... 7

        Sobre el análisis del año 2024: ..... 7

        Sobre la comparación entre 2024 y 2025: ..... 7

2. Diccionario de Datos ..... 8

3. Metodología de trabajo..... 8

4. Clasificación de tipos de día y traducción de nombres de día de la semana ..... 9

    4.1. Descripción del Proceso ..... 9

5. Scraping de feriados desde La Nación ..... 10

    5.1. Verificación de permisos de scraping..... 10

    5.2. Proceso de extracción ..... 11

6. Análisis EDA ..... 12

    6.1. Limpieza y preparación de datos..... 12

    6.2. Análisis exploratorio del 2024 ..... 13

7. Resultados obtenidos EDA 2024 ..... 14

    7.1. Dimensiones del dataframe sube 2024..... 14

    7.2. Información general del dataframe ..... 14

    7.3. Rango de fechas de 'DIA\_TRANSPORTE' ..... 14

    7.4. Verificación de duplicados ..... 14

- 7.5. Estadísticas descriptivas ..... 14
  - 7.5.1. Detección y tratamiento de valores anómalos en la columna CANTIDAD..... 15
  - 7.5.2. Análisis de distribución de la columna CANTIDAD ..... 16
- 7.6. Columnas con desviación estándar igual a cero ..... 17
- 7.7. Valores faltantes..... 17
- 7.8. Análisis Descriptivo y Detección de Outliers ..... 19
  - 7.8.1. Estadísticas Descriptivas por AMBA ..... 19
  - 7.8.2. Identificación de Outliers ..... 20
  - 7.8.3. Consideración sobre los Outliers..... 20
- 7.9. Perfil temporal..... 21
- 7.10. Perfil por categoría..... 22
- 7.11. Cantidad de viajes ..... 25
  - 7.11.1. Por tipo de transporte ..... 25
  - 7.11.2. Por motivo de feriado ..... 26
  - 7.11.3. Por día de semana y tipo de transporte ..... 27
  - 7.11.4. Por tipo de día y tipo de transporte ..... 27
- 7.12. Primeras 10 provincias más demandante - clasificación por tipo de día ..... 28
- 8. Resultados obtenidos comparación Enero-Mayo 2024 vs 2025 ..... 29
  - 8.1. Análisis comparativo de las transacciones SUBE en el transporte público ..... 29
  - 8.2. Análisis de la evolución de los viajes en transporte público ..... 29
  - 8.3. Análisis de evolución de la cantidad de viajes por tipo de transporte..... 30
    - 8.3.1. Subte..... 30
    - 8.3.2. Tren ..... 31
    - 8.3.3. Colectivo ..... 32

- 8.4. Análisis de evolución de la cantidad de viajes por día y tipo de transporte ..... 32
  - 8.4.1. Subte..... 33
  - 8.4.2. Tren ..... 33
  - 8.4.3. Colectivo ..... 34
- 9. Conclusiones..... 35
- 10. Anexos: Scripts Ejecutados..... 38
  - 10.1. Anexo: api\_tipo\_dias2024.py..... 38
  - 10.2. Anexo: api\_tipo\_dias2025.py..... 39
  - 10.3. Anexo: scraping\_consulta\_robots.py..... 40
  - 10.4. Anexo: scraping\_feriados\_lanacion2024.py..... 40
  - 10.5. Anexo: scraping\_feriados\_lanacion2025.py..... 42
  - 10.6. Anexo: eda\_sube2024.py..... 44
  - 10.7. Anexo: eda\_sube2025.py..... 55
  - 10.8. Anexo: comparativa\_2025vs2024.py..... 58
- 11. Anexo: Resultados de ejecución de scripts ..... 69
  - 11.1. Anexo: api\_tipo\_dias2024.py..... 69
  - 11.2. Anexo: api\_tipo\_dias2025.py..... 69
  - 11.3. Anexo: scraping\_consulta\_robots.py..... 69
  - 11.4. Anexo: scraping\_feriados\_lanacion2024.py..... 71
  - 11.5. Anexo: scraping\_feriados\_lanacion2025.py..... 71
  - 11.6. Anexo: eda\_sube2024.py..... 72
  - 11.7. Anexo: eda\_sube2025.py..... 76
  - 11.8. Anexo: comparativa\_2025vs2024.py..... 80

## Índice de Gráficos

Ilustración 1 Histograma de la variable CANTIDAD (viajes por registro) .....	16
Ilustración 2 Histograma de la variable CANTIDAD_LOG (log de viajes por registro).....	16
Ilustración 3 Boxplot de la variable CANTIDAD según si pertenece a AMBA.....	20
Ilustración 4 Distribución mensual de la cantidad de viajes (2024).....	21
Ilustración 5 Cantidad de viajes por día de la semana (2024).....	22
Ilustración 6 Evolución mensual de viajes por tipo de transporte (2024) .....	23
Ilustración 7 Distribución porcentual de viajes por tipo de transporte (gráfico de torta).....	23
Ilustración 8 Promedio diario de viajes por tipo de día (hábil, fin de semana, feriado) .....	24
Ilustración 9 Distribución del total de viajes según días hábiles vs no hábiles .....	25
Ilustración 10 Boxplot de la cantidad de viajes por tipo de transporte (escala logarítmica).....	26
Ilustración 11 Cantidad de viajes por motivo de feriado (barras horizontales).....	26
Ilustración 12 Cantidad total de viajes por día y tipo de transporte (2024) .....	27
Ilustración 13 Distribución porcentual de viajes por tipo de transporte y tipo de día .....	28
Ilustración 14 Promedio de viajes por tipo de día (Top 10 provincias).....	29
Ilustración 15 Evolución acumulada de viajes enero–mayo 2024 vs 2025 (total nacional).....	30
Ilustración 16 Evolución mensual de viajes en SUBTE con SUBE (enero–mayo 2024 vs 2025) .....	31
Ilustración 17 Evolución mensual de viajes en TREN con SUBE (enero–mayo 2024 vs 2025) .....	31
Ilustración 18 Evolución mensual de viajes en COLECTIVO con SUBE (enero–mayo 2024 vs 2025) ....	32
Ilustración 19 Evolución diaria de viajes en SUBTE por tipo de día (enero 2024 a mayo 2025).....	33
Ilustración 20 Evolución diaria de viajes en TREN por tipo de día (enero 2024 a mayo 2025).....	34
Ilustración 21 Evolución diaria de viajes en COLECTIVO por tipo de día (enero 2024 a mayo 2025) ...	34

## 1. Descripción del caso

A partir de datos reales y públicos, se busca analizar la cantidad y distribución de viajes que se realizan en los diferentes tipos de transporte público para todo el territorio nacional. Para el análisis se utilizó el dataset “SUBE - Cantidad de transacciones (usos) por fecha”, correspondiente al año 2024 y lo transcurrido del año 2025. Disponibles en:

- [Año 2024](#)
- [Año 2025](#)

### 1.1. Problemática planteada

Dado que el uso del transporte público en Argentina representa un componente esencial en la movilidad diaria de millones de personas. Comprender cómo varía la demanda de viajes en función del tiempo y del tipo de transporte, es fundamental para la planificación del sistema de transporte público y la toma de decisiones en políticas públicas.

En este contexto, el presente trabajo se propone abordar los siguientes niveles de análisis:

- Realizar un análisis exploratorio de cómo fue la demanda de transporte público en el año 2024.
- Realizar una análisis comparativo entre el año 2024 y 2025 correspondiente a los meses de Enero a Mayo, para evaluar posibles cambios en el uso de la tarjeta SUBE, en un contexto donde se ha comenzado a incorporar otros medios de pago.

Al llevar a cabo este enfoque dual, se pretende arribar a una mejor comprensión del comportamiento de los usuarios del transporte público argentino.

### 1.2. Contexto del análisis

El análisis se centra en los datos públicos provistos por la Secretaría de Transporte de la Nación, la cual registra la cantidad de transacciones realizadas con la tarjeta SUBE a lo largo del año 2024 y de Enero a Mayo del año 2025.

Estos dataset reflejan el uso efectivo del transporte público en todo el territorio nacional, permitiendo estudiar variaciones en la demanda según factores temporales, como el día de la semana o la presencia de feriados, y los modos de transporte (colectivo, tren y subte principalmente).

La exploración se enmarca dentro de un enfoque de análisis exploratorio de datos (EDA) con el objetivo de extraer información útil, que sirva como base para futuros estudios comparativos.

### 1.3. Hipótesis:

Se espera que la demanda de viajes en el transporte público presente variaciones significativas según factores temporales, y que los días hábiles concentren el mayor volumen de viajes en comparación con fines de semana y feriados.

Asimismo, se considera que existe una distribución desigual en la cantidad de viajes entre los distintos tipos de transporte público, con predominancia del colectivo frente al subte, tren y lancha.

Por otro lado, se anticipa que la cantidad de viajes registrados por la tarjeta SUBE durante los primeros meses de 2025 podría mostrar una disminución respecto al mismo período de 2024, posiblemente asociada a la incorporación de nuevos medios de pago electrónicos en el sistema de transporte.

### 1.4. Objetivo

Analizar el comportamiento de la cantidad de viajes en el transporte público, registrados a través del uso de la tarjeta SUBE durante el año 2024, para identificar patrones temporales (mensualidad, tipo de día, día de la semana) y modales (colectivo, tren, subte, lancha).

Adicionalmente, comparar los datos correspondientes a los primeros cinco meses del año 2025 con los del mismo período de 2024, a fin de detectar posibles cambios en la demanda registrados por la tarjeta SUBE, asociados al impacto de la incorporación de nuevos medios de pago electrónicos.

### 1.5. Preguntas clave a responder:

Sobre el análisis del año 2024:

- ¿Cómo varía la cantidad total de viajes mes a mes en 2024?
- ¿Qué días de la semana presentan mayor demanda de viajes?
- ¿Cómo se distribuyen los viajes según el tipo de transporte utilizado?
- ¿Cómo afecta el tipo de día (día hábil, fin de semana, feriado) a la cantidad promedio de viajes?
- ¿Existe una diferencia significativa en la cantidad de viajes entre días feriados y no feriados?
- ¿Existe una diferencia significativa en la cantidad de viajes entre días hábiles y no hábiles?

Sobre la comparación entre 2024 y 2025:

- ¿Qué variaciones se observan en la cantidad de viajes registrados entre Enero y Mayo del año 2024 y el 2025?
- ¿La caída en los registros de SUBE podría estar relacionada con la incorporación de nuevos medios de pago?
- ¿Existen diferencias en la evolución interanual por tipo de transporte (colectivo, tren, subte)?
- ¿Qué diferencias se observan según el tipo de día (hábil, fin de semana, feriado) al comparar ambos años?

- ¿El impacto de los nuevos medios de pago afecta a todos los modos de transporte por igual?

## 2. Diccionario de Datos

COLUMNA	TIPO	DESCRIPCIÓN
DIA_TRANSPORTE	Fecha (date)	Día de transporte informado (formato ISO-8601)
NOMBRE_EMPRESA	Texto (string)	Nombre de la empresa de transporte
LINEA	Texto (string)	Descripción de la línea
AMBA	Texto (string)	Indica si es AMBA (SI/NO)
TIPO_TRANSPORTE	Texto (string)	Colectivo, tren, subte, lanchas
JURISDICCION	Texto (string)	Tipo de jurisdicción de la línea (NACIONAL, PROVINCIAL, MUNICIPAL); vacío para subte
PROVINCIA	Texto (string)	Nombre de la provincia; si jurisdicción nacional figura 'JN'; para subte vacío
MUNICIPIO	Texto (string)	Nombre del municipio; para jurisdicción nacional o provincial figura 'SD' o 'SN'; para subte vacío
CANTIDAD	Integer	Cantidad de transacciones de uso / check-in / checkout sin checkin / venta de boletos neteadas de reversas
DATO_PRELIMINAR	Texto (string)	Indica si el dato es preliminar (SI/NO)
DIA_SEMANA	Texto (string)	Nombre del día de la semana correspondiente a dia_transporte (LUNES, MARTES, MIÉRCOLES, JUEVES, VIERNES, SÁBADO y DOMINGO)
TIPO_DIA	Texto (string)	Clasificación del día: HÁBIL, FERIADO, FIN_DE_SEMANA
MOTIVO_FERIADO	Texto (string)	Motivo del feriado si corresponde (ej: AÑO NUEVO, NAVIDAD, etc.)
CANTIDAD_LOG	Float	Logaritmo natural de cantidad para análisis estadístico
MES	Entero (int)	Número del mes correspondiente a la fecha en DIA_TRANSPORTE (1 a 12)
MES_AÑO	Texto (string)	Representación del mes y año en formato MM-AAAA o similar
ES_HABIL	Booleano (bool)	Indica si el día es hábil (True) o no (False); útil para análisis binario

## 3. Metodología de trabajo

Para este análisis se trabajó con el lenguaje Python a través del IDE Visual Studio Code. El proceso comenzó con la descarga de los dataset, los cuales fueron complementados mediante el uso de una API y técnicas de *scraping*, con el objetivo de enriquecer el análisis.

Para facilitar el mantenimiento y la comprensión del código, se optó por dividir el desarrollo en distintos archivos `.py`, cada uno enfocado en una tarea específica del análisis (por ejemplo, clasificación de días, scraping de feriados, consulta del archivo robots.txt). Esta organización permitió un trabajo más ordenado y modular.

A continuación, se explicarán los pasos llevados a cabo durante el análisis EDA.



## 4. Clasificación de tipos de día y traducción de nombres de día de la semana

Como parte del EDA de los dataset de la tarjeta SUBE correspondiente al año 2024 y 2025, se desarrollaron dos script en Python (`api_tipo_dias2024.py` y `api_tipo_dias2025.py`) con el objetivo de enriquecer la información temporal de cada registro de uso del transporte. Este procesamiento consistió en agregar dos nuevas columnas a cada dataset original: `DIA_SEMANA` y `TIPO_DIA`.

### 4.1. Descripción del Proceso

#### 1) Lectura del dataset original:

Se leyó el archivo `dat-ab-usos-2024.csv` y `dat-ab-usos-2025.csv`, asegurando que la columna `DIA_TRANSPORTE` se interprete como una fecha. Dado que el dataset de 2025 aún no está completo, se hace referencia directamente a la web para obtener la versión más actualizada al momento de ejecutar el script, utilizando la siguiente línea de código:

```
df_sube = pd.read_csv("<URL del archivo CSV en línea>",  
parse_dates=["DIA_TRANSPORTE"])
```

#### 2) Obtención de feriados nacionales:

Se utilizó la API pública de Nager.Date, disponible en <https://date.nager.at/Api>, para obtener el listado oficial de feriados en Argentina durante el año 2024 y 2025. Esto permitió construir una lista de fechas consideradas feriados, sin necesidad de codificar esta información manualmente.

#### 3) Clasificación de tipos de día:

Se creó una función que clasifica cada fecha en una de las siguientes tres categorías:

- **FERIADO:** si la fecha corresponde a un feriado nacional según la API.
- **FIN\_DE\_SEMANA:** si el día corresponde a sábado o domingo.
- **HÁBIL:** cualquier otro día de la semana (de lunes a viernes que no sea feriado).

#### 4) Traducción del nombre del día:

Se generó una nueva columna `DIA_SEMANA`, en la cual se tradujo el nombre del día (por ejemplo, "Monday") al español (por ejemplo, "LUNES").

#### 5) Enriquecimiento y exportación del dataset:

Finalmente, se agregaron ambas columnas nuevas (`DIA_SEMANA` y `TIPO_DIA`) al dataframe original, y se exportó el resultado a un nuevo archivo llamado `df-sube-2024-tipo-dia.csv` y `df-sube-2025-tipo-dia.csv`. Esto facilitó posteriores análisis diferenciando el comportamiento de los usuarios según el tipo de día.

El código completo correspondiente a este procesamiento puede consultarse en los siguientes anexos:

- [Anexo: api\\_tipo\\_dias2024.py](#)
- [Anexo: api\\_tipo\\_dias2025.py](#)

La salida de consola generada durante su ejecución se encuentra documentada en los anexos:

- [Anexo: api\\_tipo\\_dias2024.py](#)
- [Anexo: api\\_tipo\\_dias2025.py](#)

## 5. Scraping de feriados desde La Nación

Con el objetivo de complementar la información temporal de los dataset y la clasificación de los días como *feriados*, se desarrolló un script en Python (`scraping_feriados2024.py` y `scraping_feriados2025.py`) para obtener el listado detallado de feriados nacionales del año 2024 y 2025, junto con su respectivo motivo. Esta información se extrajo del sitio web de La Nación, específicamente de la página, el acceso a dichas páginas son:

- <https://www.lanacion.com.ar/feriados/2024/>.
- <https://www.lanacion.com.ar/feriados/2025/>.

### 5.1. Verificación de permisos de scraping

Antes de iniciar con la extracción de datos mediante técnicas de *web scraping*, se consultó el archivo `robots.txt` del sitio web de La Nación para verificar si existían restricciones acerca del acceso automatizado a sus contenidos.

Este procedimiento se implementó en el script `scraping_consulta_robots.py`, utilizando la biblioteca `requests` de Python para acceder al archivo ubicado en <https://www.lanacion.com.ar/robots.txt>.

El archivo `robots.txt` no prohíbe de forma general el uso de *scraping* en el sitio. Solo restringe el acceso a ciertas rutas específicas, como por ejemplo:

- `/sinbarreras/,/newsletters/,/registracion/`
- URLs con parámetros `?utm_*`
- Algunas rutas como `/buscador/,/pf/api/...` y determinados artículos puntuales

Las páginas de nuestro interés no se encuentra listada dentro de las restricciones, por lo tanto acceder a las mismas está permitido.

El código correspondiente a este procesamiento puede consultarse en el anexo:

- [Anexo: scraping\\_consulta\\_robots.py](#)

La salida de consola generada durante su ejecución se encuentra documentada en el anexo:

- [Anexo: scraping\\_consulta\\_robots.py](#)

## 5.2. Proceso de extracción

El script realiza los siguientes pasos:

**1) Solicitud HTTP y parsing del HTML:**

Se utiliza la librería `requests` para descargar el contenido de la página y `BeautifulSoup` para interpretar su estructura HTML.

**2) Identificación de los feriados por mes:**

Tanto para 2024 como en 2025, la página organiza los feriados dentro de un bloque `<div class="holidays-card-calendar">`, y el nombre del mes se extrae explícitamente desde una etiqueta `<a class="com-link">`. Esto permitió hacer un mapeo directo entre nombre del mes y su número, mejorando la robustez del script.

**3) Extracción de la información relevante:**

De cada entrada de feriado se extrae:

- El día del mes, localizado en un elemento `<span>` cuya clase presenta un patrón común (detectado mediante expresiones regulares).
- El motivo del feriado, ubicado en un elemento `<h4>` con clase `com-text`.

Se construye la fecha completa en formato `YYYY-MM-DD` y se almacena en un diccionario Python junto con el motivo correspondiente.

**4) Manejo de errores:**

Se incluyen estructuras `try-except` para evitar que errores puntuales en el formato interrumpan el procesamiento del resto de los datos. También se imprimen advertencias si no se reconoce un mes.

**5) Verificación del scraping en consola:**

Al inicio se presentaron dificultades al extraer correctamente los datos del sitio web. Para validar que el scraping funcionaba como se esperaba, se incorporó una impresión por consola de las fechas y motivos de los feriados recolectados. Esto permitió detectar y corregir errores antes de aplicar el enriquecimiento al dataset.

**6) Enriquecimiento del dataset original:**

Luego de completar el scraping, se carga el archivo `df-sube-2024-tipo-dia.csv` o `df-sube-2025-tipo-dia.csv` según el año correspondiente, y se agrega una nueva columna llamada `MOTIVO_FERIADO`. Esta columna indica el motivo del feriado si la fecha coincide con una extraída del sitio; de lo contrario, se indica "NO FERIADO".

**7) Exportación del nuevo dataset:**

El dataset resultante se guarda como `df-sube-2024.csv` o `df-sube-2025.csv`, con las columnas originales más la nueva variable `MOTIVO_FERIADO`.

Este enriquecimiento permite realizar análisis diferenciado no solo por tipo de día, sino también según el motivo específico del feriado, habilitando estudios más detallados del comportamiento de los usuarios del transporte público en fechas especiales.

El código correspondiente a este procesamiento puede consultarse en los anexos:

- [Anexo: scraping\\_feriados\\_lanacion2024.py](#)
- [Anexo: scraping\\_feriados\\_lanacion2025.py](#)

La salida de consola generada durante su ejecución se encuentra documentada en los anexos:

- [Anexo: scraping\\_feriados\\_lanacion2024.py](#)
- [Anexo: scraping\\_feriados\\_lanacion2025.py](#)

## 6. Análisis EDA

En esta etapa se realizó un Análisis Exploratorio de Datos (EDA) con el objetivo de comprender las características principales del dataframe y detectar patrones relevantes en la demanda del transporte público durante el año 2024. Además, se preparó un segundo dataset correspondiente al año 2025 con el propósito de realizar comparaciones posteriores, sin un análisis exploratorio detallado.

### 6.1. Limpieza y preparación de datos

Se llevó a cabo un proceso de limpieza y estandarización sobre ambos datasets (2024 y 2025), con el fin de asegurar la calidad de los datos y su comparabilidad. Las principales acciones realizadas fueron:

- **Conversión de tipos de datos:** se aseguró que la columna `DIA_TRANSPORTE` fuera interpretada correctamente como tipo fecha (`datetime`), lo cual es fundamental para los análisis temporales.
- **Se analizaron y detectaron registros con valores faltantes en columnas clave:** Se identificaron valores nulos en columnas clave como `JURISDICCION`, `PROVINCIA` y `MUNICIPIO`, especialmente en registros del tipo de transporte `SUBTE`, los cuales fueron imputados con información conocida (por ejemplo, "CABA" o "CIUDAD AUTÓNOMA DE BUENOS AIRES").
- **Corrección y análisis de valores anómalos (outliers):** Se identificaron valores atípicos en la columna `CANTIDAD`, tanto negativos como extremadamente altos.
  - Los valores negativos fueron descartados, ya que no tienen justificación lógica en el contexto (cantidad de viajes no puede ser negativa).
  - Los valores extremadamente altos, aunque considerados outliers según el rango intercuartílico (IQR), no fueron eliminados, ya que podrían reflejar situaciones reales como eventos masivos, interrupciones o alteraciones en el servicio.
- **Estandarización de etiquetas:** Se unificaron valores con diferencias gramatical como "C.A.B.A" y "CABA", para evitar distorsiones en el análisis.
- **Creación de nuevas variables derivadas** (solo en el dataset 2024): se agregaron columnas como `DIA_SEMANA`, `TIPO_DIA` y `CANTIDAD_LOG` (transformación logarítmica de la cantidad de viajes) para facilitar análisis específicos y mejorar la interpretación de los resultados.

Estas tareas de preprocesamiento fueron fundamentales para asegurar que el análisis posterior se base en datos consistentes y representativos de la realidad.

El dataset limpio del año 2025 se almacenó como `dat-sube-2025.csv` y está preparado para su uso comparativo, sin haber sido sometido a un análisis exploratorio completo.

## 6.2. Análisis exploratorio del 2024

El análisis detallado se enfocó exclusivamente en el año 2024, e incluyó:

- **Evolución temporal de los viajes:** se analizaron las cantidades totales de viajes por mes y por día de la semana para identificar tendencias y variaciones estacionales.
- **Análisis por tipo de transporte:** se exploró cómo se distribuyen los viajes entre colectivos, trenes, subtes y lanchas, destacando los modos de transporte más utilizados.
- **Impacto del tipo de día:** se comparó la demanda entre días hábiles, fines de semana y feriados, identificando diferencias significativas en los patrones de uso.
- **Detección de anomalías y valores atípicos:** se evaluaron posibles outliers en los datos que pudieran afectar el análisis.
- **Visualizaciones clave generadas:** histogramas, gráficos de barras, boxplots y heatmaps que permiten observar la interacción entre día de la semana y tipo de transporte.

El análisis se complementó con la generación de archivos gráficos que facilitan la interpretación de los resultados y sirven como base para posteriores estudios comparativos, especialmente para evaluar el impacto de cambios futuros en el sistema de transporte.

El código correspondiente a este procesamiento puede consultarse en los anexos:

- [Anexo: eda\\_sube2024.py](#)
- [Anexo: eda\\_sube2025.py](#)

La salida de consola generada durante su ejecución se encuentra documentada en los anexos:

- [Anexo: eda\\_sube2024.py](#)
- [Anexo: eda\\_sube2025.py](#)

## 7. Resultados obtenidos EDA 2024

### 7.1. Dimensiones del dataframe sube 2024

El dataframe analizado contiene 504.676 filas y 13 columnas.

### 7.2. Información general del dataframe

A continuación, se presenta la estructura del dataframe utilizado:

Nº	Columna	Non-Null Count	Tipo de Dato
0	DIA_TRANSPORTE	504,676	datetime64[ns]
1	NOMBRE_EMPRESA	504,676	object
2	LINEA	504,676	object
3	AMBA	504,676	object
4	TIPO_TRANSPORTE	504,676	object
5	JURISDICCION	502,154	object
6	PROVINCIA	502,139	object
7	MUNICIPIO	502,139	object
8	CANTIDAD	504,676	int64
9	DATO_PRELIMINAR	504,676	object
10	DIA_SEMANA	504,676	object
11	TIPO_DIA	504,676	object
12	MOTIVO_FERIADO	504,676	object

De esta tabla se desprenden las siguientes observaciones:

- El dataframe contiene una columna de tipo `datetime64[ns]`, una de tipo `int64` y once columnas de tipo `object`.
- Las columnas `JURISDICCION`, `PROVINCIA` y `MUNICIPIO` presentan valores faltantes.

### 7.3. Rango de fechas de 'DIA\_TRANSPORTE'

El rango de fechas abarcado por el dataframe va desde el 1 de enero de 2024 hasta el 31 de diciembre de 2024.

### 7.4. Verificación de duplicados

Se realizó una verificación de registros duplicados, arrojando como resultado 0 (cero) duplicados.

### 7.5. Estadísticas descriptivas

Para obtener las estadísticas descriptivas, fue necesario especificar el parámetro `include=[np.number]`, con el objetivo de excluir columnas de tipo `datetime` cuya inclusión

generaba resultados no relevantes para el análisis. Además, se configuró la visualización con dos decimales para facilitar la interpretación de los datos.

En los resultados se observa un **valor mínimo de -105** en la columna `CANTIDAD`, lo cual es inconsistente y podría indicar un error de carga o procesamiento de datos.

	CANTIDAD
count	504676.00
mean	8194.59
std	18748.33
min	-105.00
25%	545.00
50%	2127.00
75%	7617.00
max	516002.00

7.5.1. Detección y tratamiento de valores anómalos en la columna `CANTIDAD`

Durante el análisis exploratorio, se identificaron 3 registros con valores negativos en la columna `CANTIDAD`, lo cual no resulta coherente con el significado de dicha variable (cantidad de viajes), ya que no tiene sentido registrar cantidades menores a cero. Los registros detectados fueron los siguientes:

DIA_TRANSPORTE	TIPO_TRANSPORTE	CANTIDAD
2024-02-20	TREN	-3
2024-03-05	TREN	-105
2024-04-20	TREN	-1

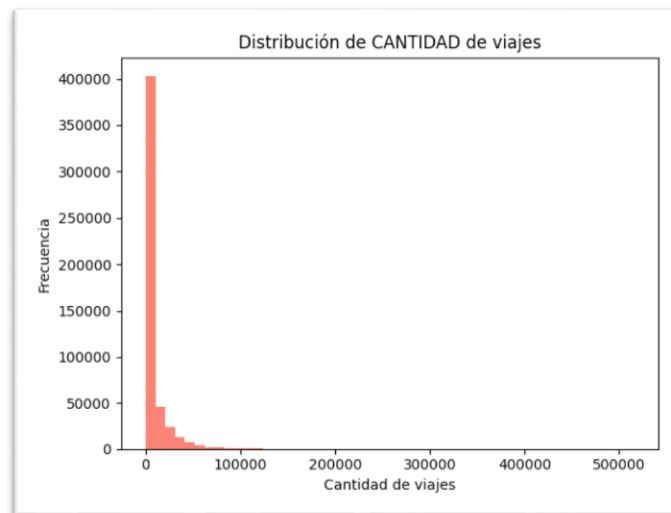
Estos valores se consideran datos anómalos, probablemente originados por errores de carga o ajustes no documentados. Como parte del proceso de limpieza de datos, se eliminaron estas filas para evitar distorsiones en el análisis estadístico posterior.

Una vez realizados los filtros correspondientes, se actualizó el resumen estadístico de la columna `CANTIDAD`, eliminando así la influencia de estos valores erróneos.

	CANTIDAD
count	504673.00
mean	8194.63
std	18748.37
min	1.00
25%	546.00
50%	2127.00
75%	7617.00
max	516002.00

### 7.5.2. Análisis de distribución de la columna CANTIDAD

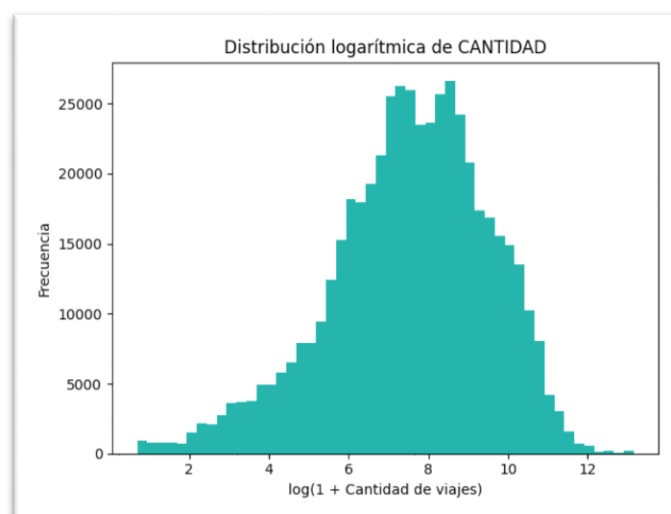
Se generó un histograma de la columna `CANTIDAD` para observar cuál es su distribución. En el siguiente gráfico se puede observar que existe una distribución muy marcada hacia la derecha, con una gran concentración de valores bajos y pocos valores altos, los cuales actúan como outliers.



*Ilustración 1 Histograma de la variable CANTIDAD (viajes por registro)*

Esta distribución altamente sesgada, con la mayoría de los valores concentrados en el rango bajo y unos pocos valores que alcanzan hasta más de 500.000 viajes en un día, puede dificultar el análisis e influir negativamente en ciertos modelos estadísticos.

Para mitigar esa asimetría, se aplicó una transformación logarítmica utilizando la función `np.log1p`. Esta función es útil en contextos donde puede haber valores iguales a cero, ya que el logaritmo natural de cero no está definido. De esta forma, se redujo el impacto de los valores extremos y se consigue una distribución más uniforme, como se puede observar en el siguiente gráfico.



*Ilustración 2 Histograma de la variable CANTIDAD\_LOG (log de viajes por registro)*



Luego de aplicar la transformación, la variable presenta una forma aproximadamente normal, lo que facilita tanto la interpretación como la aplicación de técnicas analíticas que suponen normalidad o baja asimetría.

Si bien se generó la columna transformada (`CANTIDAD_LOG`) esta no fue utilizada en el análisis, dado que el objetivo del estudio es estrictamente descriptivo y no requiere normalización para modelos predictivos.

## 7.6. Columnas con desviación estándar igual a cero

Se verificó que ninguna de las columnas numéricas presenta desviación estándar igual a cero, lo que indica que no existen variables constantes en el conjunto de datos.

## 7.7. Valores faltantes

Se decidió realizar un análisis detallado de los valores faltantes en lugar de eliminarlos directamente (`drop`). Como se muestra en la siguiente tabla, los valores nulos se encuentran exclusivamente en las columnas `JURISDICCION`, `PROVINCIA` y `MUNICIPIO`:

Columna	Valores Faltantes
DIA_TRANSPORTE	0
NOMBRE_EMPRESA	0
LINEA	0
AMBA	0
TIPO_TRANSPORTE	0
JURISDICCION	2522
PROVINCIA	2537
MUNICIPIO	2537
CANTIDAD	0
DATO_PRELIMINAR	0
DIA_SEMANA	0
TIPO_DIA	0
MOTIVO_FERIADO	0
CANTIDAD_LOG	0

Se utilizó el método `.isna()` para identificar a qué tipo de transporte corresponden los registros con valores faltantes:

Tipos de transporte con `JURISDICCION` nula:

```
TIPO_TRANSPORTE
SUBTE          2522
```

Tipos de transporte con `PROVINCIA` nula:

```
TIPO_TRANSPORTE
SUBTE          2522
```

COLECTIVO 11  
TREN 4

Tipos de transporte con MUNICIPIO nulo:

TIPO\_TRANSPORTE  
SUBTE 2522  
COLECTIVO 11  
TREN 4

Se observa que los 2522 valores nulos en las tres columnas corresponden al tipo de transporte SUBTE. Según el diccionario de datos, en estos casos dichas columnas se dejan vacías. Por lo tanto, se completaron con los siguientes valores:

- JURIDICCION: CABA
- PROVINCIA: CIUDAD AUTONOMA DE BUENOS AIRES
- MUNICIPIO: CABA

Después de esta imputación, se realizó una nueva verificación:

Columna	Valores Faltantes
JURISDICCION	0
PROVINCIA	15
MUNICIPIO	15

Los valores faltantes restantes corresponden a los siguientes casos:

**Tabla 1: Valores Nulos por Provincia**

TIPO_TRANSPORTE	NOMBRE_EMPRESA	LINEA	AMBA
COLECTIVO	EMPRESA 9 DE JULIO SRL	LINEA_500I_SFE	NO
COLECTIVO	EMPRESA 9 DE JULIO SRL	LINEA_500I_SFE	NO
COLECTIVO	EMPRESA 9 DE JULIO SRL	LINEA_500I_SFE	NO
COLECTIVO	EMPRESA 9 DE JULIO SRL	LINEA_500I_SFE	NO
COLECTIVO	EMPRESA 9 DE JULIO SRL	LINEA_500I_SFE	NO
COLECTIVO	EMPRESA 9 DE JULIO SRL	LINEA_500I_SFE	NO
COLECTIVO	EMPRESA 9 DE JULIO SRL	LINEA_500I_SFE	NO
COLECTIVO	EMPRESA 9 DE JULIO SRL	LINEA_500I_SFE	NO
COLECTIVO	EMPRESA 9 DE JULIO SRL	LINEA_500I_SFE	NO
COLECTIVO	EMPRESA 9 DE JULIO SRL	LINEA_500I_SFE	NO
TREN	SOFSE - TREN DEL VALLE FFCC	TREN DEL VALLE	NO
TREN	SOFSE - TREN DEL VALLE FFCC	TREN DEL VALLE	NO
TREN	SOFSE - TREN DEL VALLE FFCC	TREN DEL VALLE	NO
TREN	SOFSE - TREN DEL VALLE FFCC	TREN DEL VALLE	NO

**Tabla 2: Valores Nulos por Municipio**

TIPO_TRANSPORTE	NOMBRE_EMPRESA	LINEA	AMBA
COLECTIVO	EMPRESA 9 DE JULIO SRL	LINEA_500I_SFE	NO
COLECTIVO	EMPRESA 9 DE JULIO SRL	LINEA_500I_SFE	NO
COLECTIVO	EMPRESA 9 DE JULIO SRL	LINEA_500I_SFE	NO
COLECTIVO	EMPRESA 9 DE JULIO SRL	LINEA_500I_SFE	NO
COLECTIVO	EMPRESA 9 DE JULIO SRL	LINEA_500I_SFE	NO
COLECTIVO	EMPRESA 9 DE JULIO SRL	LINEA_500I_SFE	NO
COLECTIVO	EMPRESA 9 DE JULIO SRL	LINEA_500I_SFE	NO
COLECTIVO	EMPRESA 9 DE JULIO SRL	LINEA_500I_SFE	NO
COLECTIVO	EMPRESA 9 DE JULIO SRL	LINEA_500I_SFE	NO
COLECTIVO	EMPRESA 9 DE JULIO SRL	LINEA_500I_SFE	NO
COLECTIVO	EMPRESA 9 DE JULIO SRL	LINEA_500I_SFE	NO
TREN	SOFSE - TREN DEL VALLE FFCC	TREN DEL VALLE	NO
TREN	SOFSE - TREN DEL VALLE FFCC	TREN DEL VALLE	NO
TREN	SOFSE - TREN DEL VALLE FFCC	TREN DEL VALLE	NO
TREN	SOFSE - TREN DEL VALLE FFCC	TREN DEL VALLE	NO

Según esta información:

- A los registros correspondientes a LINEA\_500I\_SFE se les asignó la provincia y municipio: "SANTA FE".
- Para TREN DEL VALLE, se completaron los valores como:
  - PROVINCIA: JN (Jurisdicción Nacional)
  - MUNICIPIO: SD (Sin Dato), ya que el tren es de carácter nacional.

Finalmente, tras completar estos campos, se verifica que no quedan valores nulos en las columnas analizadas:

Columna	Valores Faltantes
JURISDICCION	0
PROVINCIA	0
MUNICIPIO	0

## 7.8. Análisis Descriptivo y Detección de Outliers

### 7.8.1. Estadísticas Descriptivas por AMBA

Se calcularon estadísticas descriptivas de la variable `CANTIDAD` diferenciando los registros según si corresponden a la región AMBA (SI) o no (NO):

AMBA	Count	Mean	Std. Dev.	Min	25%	50%	75%	Max
NO	354,058	2.439	3.324	1	350	1.178	3.294	34.994
SI	150,615	21.725	29.847	1	5.203	14.261	27.951	516.002

Esto permite observar que los registros en AMBA presentan en promedio valores mucho más altos de pasajeros, con una gran dispersión (desviación estándar), en comparación con las zonas fuera de AMBA.

### 7.8.2. Identificación de Outliers

Se realizó la detección de outliers utilizando el criterio del rango intercuartílico (IQR), separando por grupo AMBA:

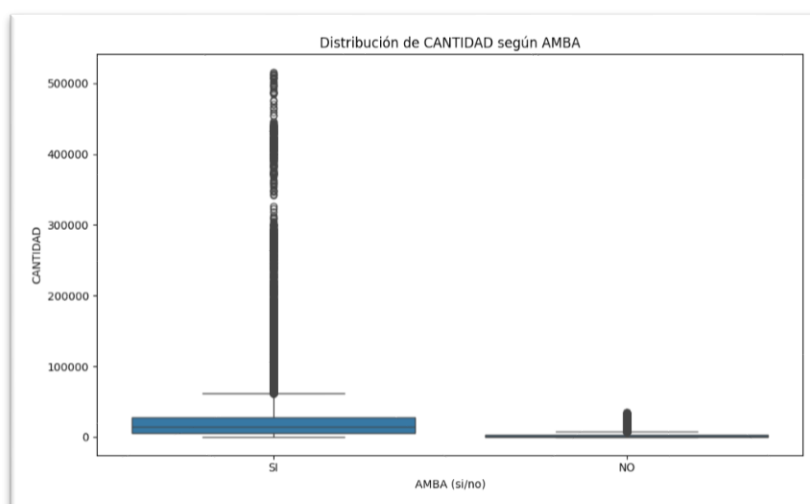
- **Para AMBA = NO:**
  - Q1: 350.0, Q3: 3294.0, IQR: 2944.0
  - Rango considerado normal: [0, 7710]
  - Total de filas: 354.058
  - Outliers detectados: 23.965
- **Para AMBA = SI:**
  - Q1: 5203.0, Q3: 27951.5, IQR: 22.748,5
  - Rango considerado normal: [0, 62.074,2]
  - Total de filas: 150.615
  - Outliers detectados: 8.112

### 7.8.3. Consideración sobre los Outliers

Si bien se identificaron outliers, no fueron eliminados del conjunto de datos. Esto se debe a que estos valores extremos podrían representar fenómenos reales como:

- AMBA concentra un 30% de la población nacional.
- A diferencia de otras provincias o regiones, AMBA tiene una alta frecuencia y gran volumen de transporte público.
- Hay días o casos específicos (días hábiles, eventos masivos, paros o piquetes, entre otros)

Por lo tanto, en lugar de descartarlos, se optó por marcarlos e incluirlos en el análisis, para evaluar su impacto y comprender mejor su contexto.



*Ilustración 3 Boxplot de la variable CANTIDAD según si pertenece a AMBA*

## 7.9. Perfil temporal

En el siguiente gráfico, donde se analiza la cantidad de viajes respecto a los meses del año, podemos observar una disminución en la cantidad de viajes durante los meses de enero, febrero, junio y diciembre, en comparación con el resto del año. Este mismo patrón se refleja en el gráfico Evolución mensual por tipo de transporte, donde la tendencia se presenta segmentada según el medio utilizado.

Esta caída en la cantidad de viajes podría deberse a que dichos meses coinciden con los períodos de vacaciones de verano e invierno, así como con el inicio y la finalización del ciclo escolar.

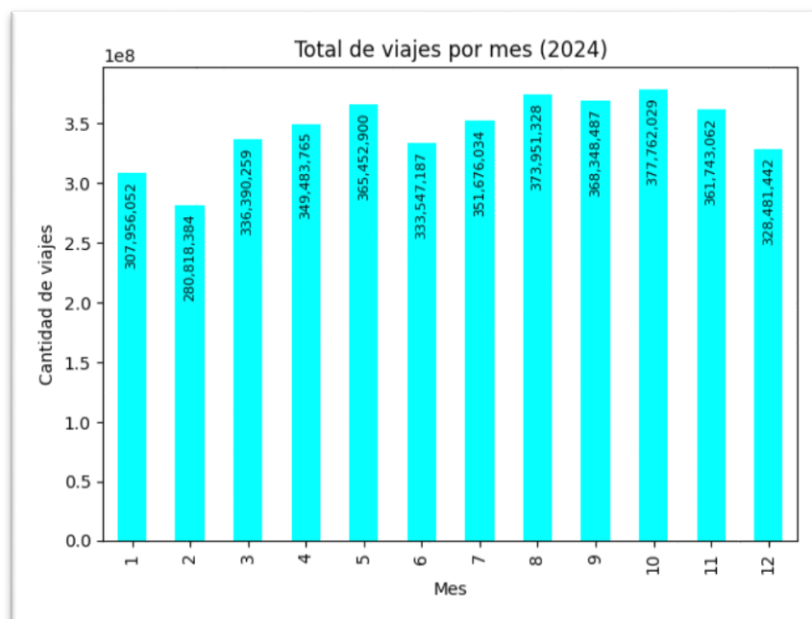


Ilustración 4 Distribución mensual de la cantidad de viajes (2024)

Por otro lado, si analizamos la cantidad de viajes respecto a los distintos días de la semana podemos ver que son bastante similares, con excepción de los sábados y domingos, donde se registra una disminución considerable. Entre los días laborables, se observa una leve caída de la cantidad de viajes el día lunes y un leve aumento el día viernes.

Estas variaciones pueden atribuirse al ritmo de la actividad laboral, que se concentra mayormente entre semana y se reduce durante el fin de semana. Por otro lado, no se detecta una influencia significativa del trabajo remoto en la distribución de los viajes.

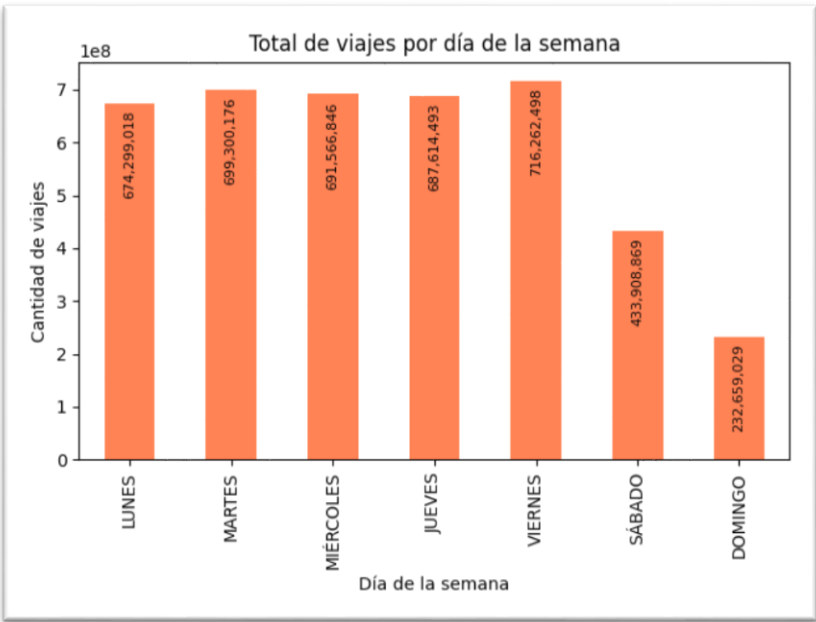


Ilustración 5 Cantidad de viajes por día de la semana (2024)

7.10. Perfil por categoría

En este caso, podemos observar la evolución mensual de la cantidad de viajes diferenciada por tipo de transporte. A simple vista, se nota que los viajes en colectivo superan ampliamente a las demás categorías, lo cual coincide con la variación analizada en el perfil temporal.

Además, se destaca el peso desproporcionado que tiene esta categoría en el total de viajes, en contraste con otras como el tren, subte o lancha, que muestran volúmenes significativamente menores. Esta diferencia en la cantidad contribuye a la dispersión y aparición de valores atípicos en los gráficos de boxplot.

Por otra parte, se podrían analizar patrones estacionales específicos para cada medio, como la caída general durante los períodos de vacaciones o el comportamiento más estable de subtes, que opera en zonas muy localizadas. Esta información permite comprender mejor la estructura del sistema de transporte y su uso a lo largo del año.

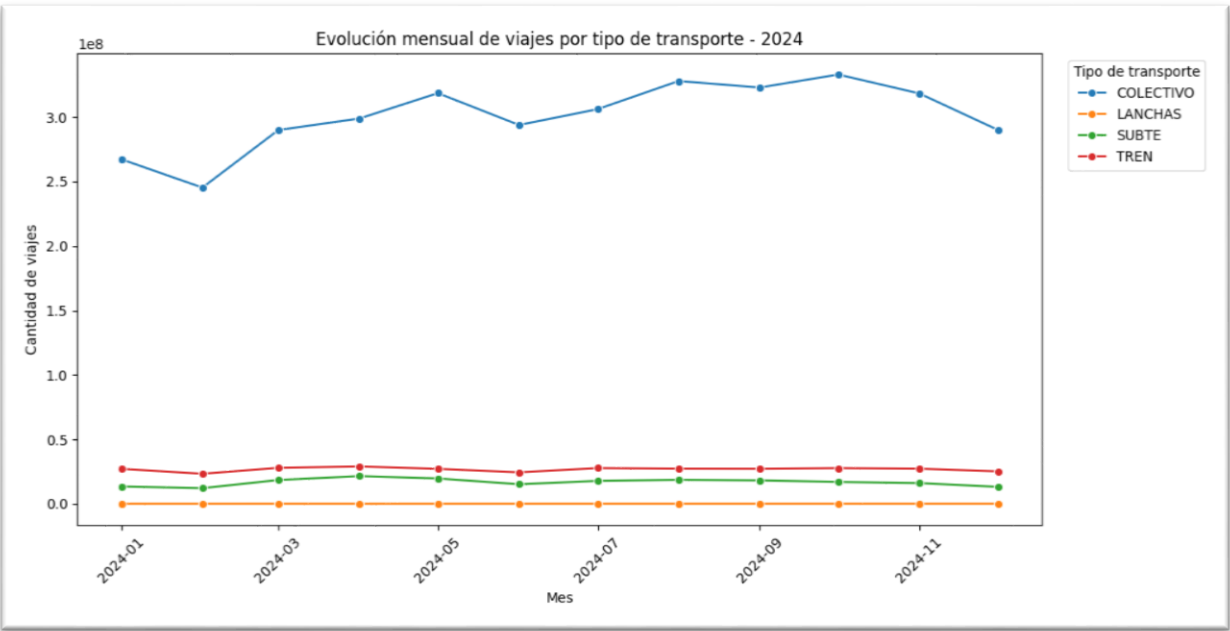


Ilustración 6 Evolución mensual de viajes por tipo de transporte (2024)

Como complemento se muestra el siguiente gráfico de torta, el cual muestra que los colectivos concentran la mayor parte de los viajes registrados, seguidos por el tren y en menor proporción el subte y lanchas. Esto indica que el colectivo sigue siendo el principal medio de transporte público a nivel nacional.

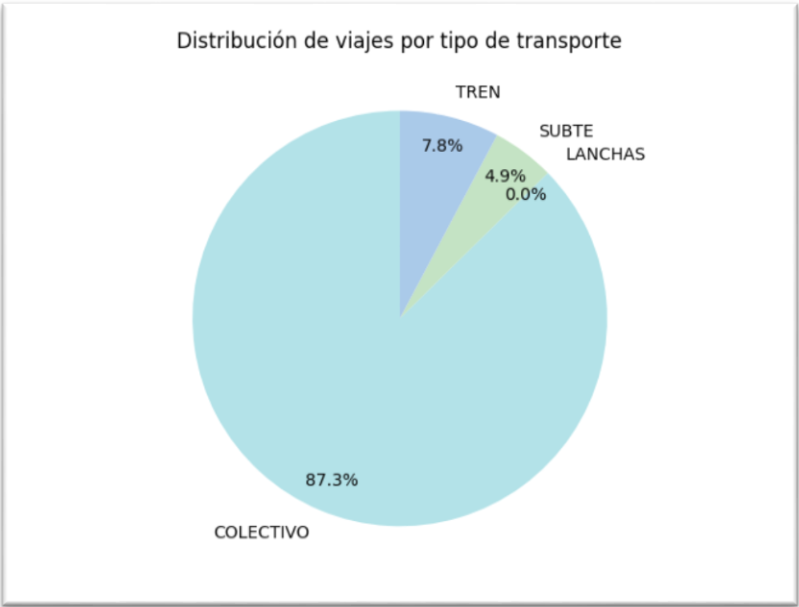
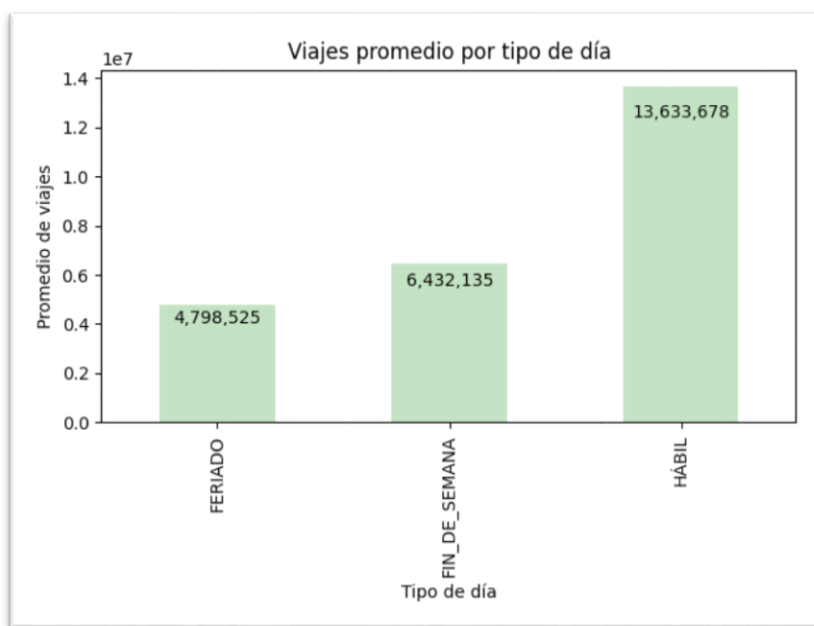


Ilustración 7 Distribución porcentual de viajes por tipo de transporte (gráfico de torta)

Si analizamos el promedio de viajes según el tipo de día, esto revela un patrón claro: los días hábiles concentran la mayor cantidad de viajes, con un promedio diario de 13.633.678 viajes, muy por encima de los fines de semana (6.432.135) y los feriados (4.798.525).

Esta diferencia sugiere que el transporte público en Argentina se utiliza principalmente con fines laborales y escolares. Los fines de semana y feriados, en cambio, presentan una disminución aproximada del 53% y 65% respectivamente, en comparación con un día hábil, lo que indica una caída significativa en la movilidad durante esos días.

Este patrón es consistente con una estructura de movilidad centrada en la actividad productiva.



*Ilustración 8 Promedio diario de viajes por tipo de día (hábil, fin de semana, feriado)*

En concordancia con lo analizado hasta ahora, y considerando únicamente la variable tipo de día, se observa que los días hábiles concentraron el 87,7% del total de viajes en el año, mientras que los días no hábiles (fines de semana y feriados) representaron solo el 17,3%.

Esta información refuerza la idea de que el transporte público está fuertemente ligado a las actividades laborales, educativas y administrativas, las cuales se desarrollan principalmente entre semana.

Además, la marcada diferencia entre ambos tipos de día evidencia una baja utilización del transporte público con fines recreativos o personales durante los días no hábiles, lo cual podría asociarse a una mayor permanencia en los hogares, el uso de medios de transporte alternativos o una menor oferta de servicios en esos días.



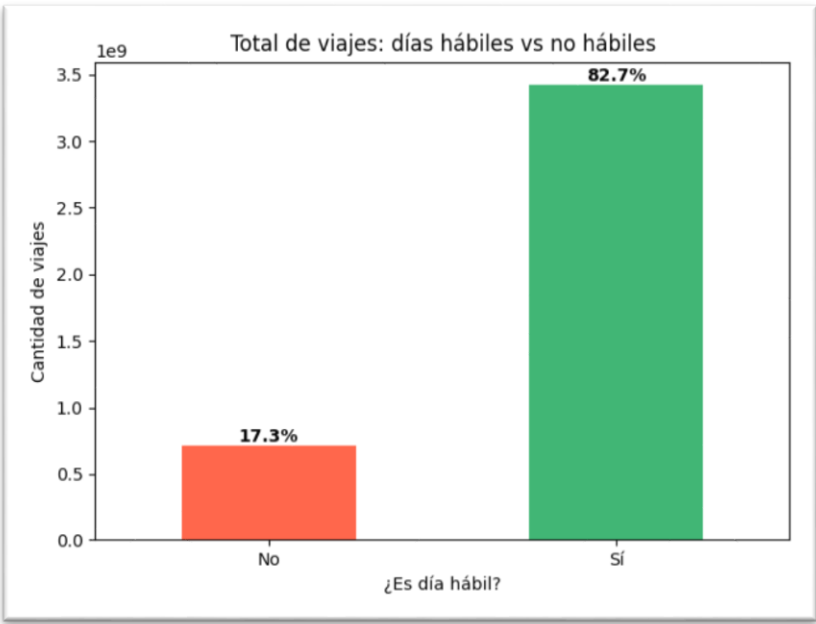


Ilustración 9 Distribución del total de viajes según días hábiles vs no hábiles

7.11. Cantidad de viajes

7.11.1. Por tipo de transporte

Este análisis muestra la distribución a nivel nacional de la cantidad de viajes por tipo de transporte. En la gráfica podemos observar una distribución de los datos con baja dispersión pero con la presencia de datos atípicos que reflejan las diferencias regionales. Grandes centros urbanos como Córdoba, Rosario o AMBA tienen una elevada cantidad de viajes que generan valores atípicos.

Estos valores atípicos pueden corresponder a que:

- Trenes y colectivos son los medios más utilizados y tienen mayor cobertura a nivel nacional.
- Líneas de colectivo o ramales de tren mucho más concurridos que otros.
- Paros o cortes de servicios que hacen que otro tipo de transporte se sature.
- Días hábiles o eventos masivos.

Para mejorar la información y evaluar mejor los outliers, se podría hacer un análisis más detallado teniendo en cuenta las distintas provincias y el tipo de transporte.

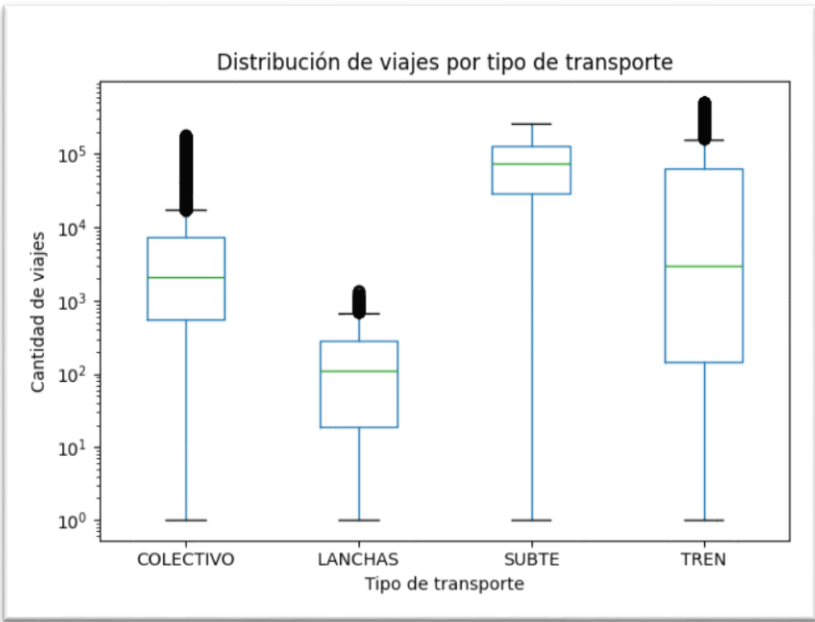


Ilustración 10 Boxplot de la cantidad de viajes por tipo de transporte (escala logarítmica)

7.11.2. Por motivo de feriado

Teniendo en cuenta esta variable, la cantidad de viajes no tiene gran variación, salvo año nuevo, donde la frecuencia del transporte es muy reducida. Para el caso de carnaval, este contempla dos días lo que daría en promedio una cantidad similar al resto.

Se destacan fechas como el 17 de Agosto (Paso a la inmortalidad de San Martin) y el 25 de mayo (Revolución de Mayo), lo cual puede explicarse por actividades conmemorativas, turismo interno o cambios en los patrones habituales de movilidad.

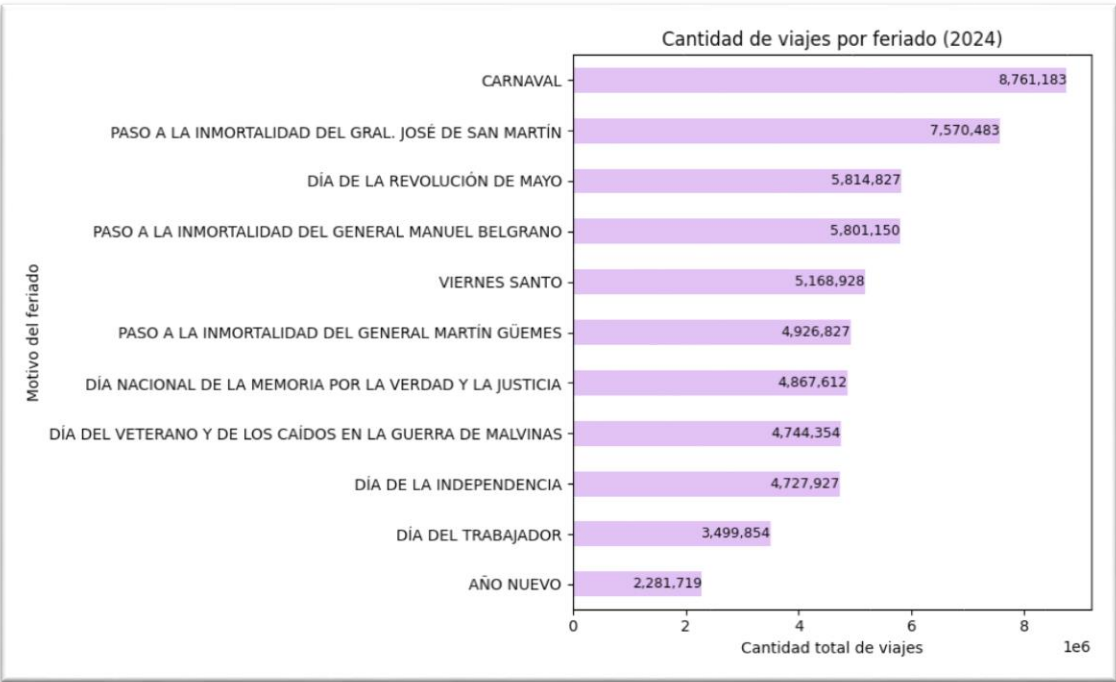


Ilustración 11 Cantidad de viajes por motivo de feriado (barras horizontales)

7.11.3. Por día de semana y tipo de transporte

Se realizó un gráfico heatmap para comprender como se relacionan los diferentes tipos de transporte a lo largo de la semana. Se observan las siguientes particularidades:

- El colectivo, claramente es el medio de transporte que concentra gran porcentaje de los viajes, convirtiéndose en el medio de transporte más utilizado durante toda la semana.
- Los días viernes son los días donde se registra la mayor cantidad de viajes, seguido por los días martes y miércoles.
- Los días domingo registran la menor cantidad de viajes en todos los medios de transporte, lo cual puede asociarse a menor movilidad por descanso, actividades familiares o reducción de frecuencias.
- Tanto el tren como el subte tienen su mayor volumen de viajes los días hábiles, con un notable descenso los fines de semana, esto sugiere un uso más vinculado a actividades laborales o escolares.
- Por último se observa que las lanchas poseen la menor cantidad de viajes, lo cual resulta esperable dado que se trata de un medio de transporte restringido a la zona del Delta.

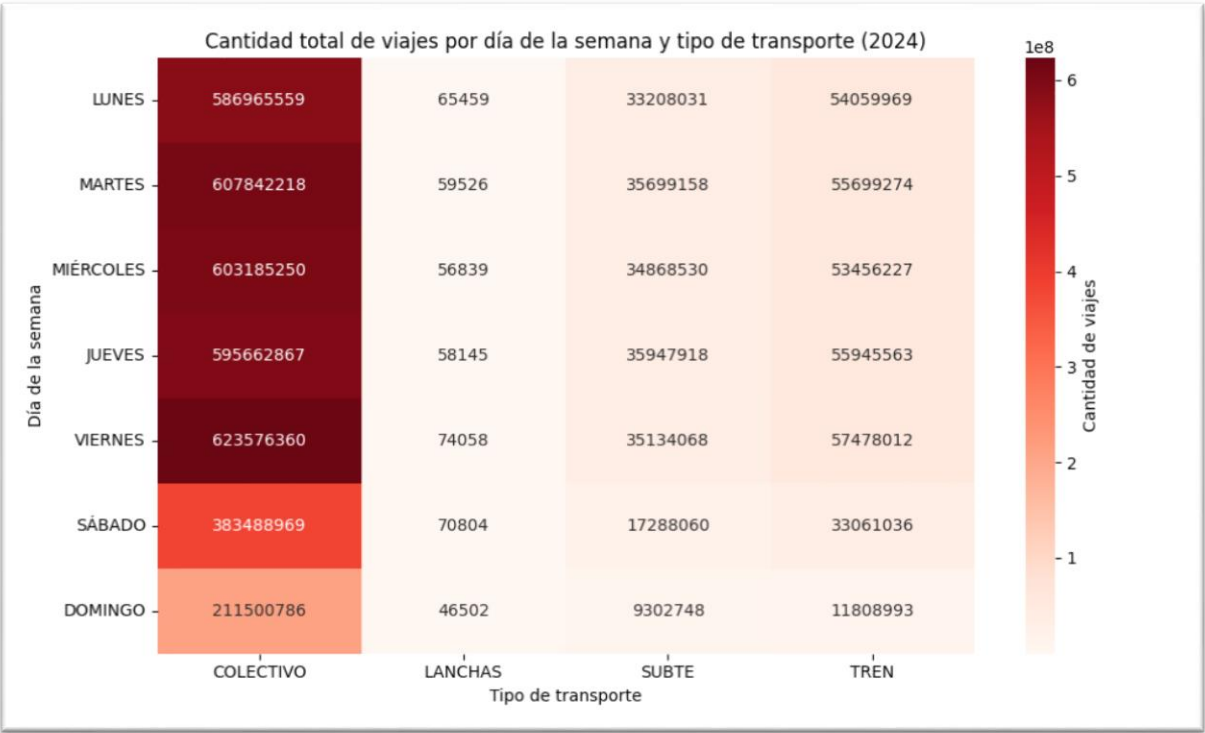


Ilustración 12 Cantidad total de viajes por día y tipo de transporte (2024)

7.11.4. Por tipo de día y tipo de transporte

En este análisis se observa la Distribución Modal del Transporte, donde se presentó un gráfico de torta (pie chart) que resume la participación relativa de cada modo de transporte.

Los resultados indican que:

- El colectivo es el medio más utilizado, con una amplia diferencia respecto a los demás.
- Le siguen en menor proporción el tren y el subte.
- Las lanchas tienen un uso marginal, concentrado en regiones específicas.

Esta distribución refleja un patrón de movilidad consistente a nivel nacional, alineado con la infraestructura y la accesibilidad de cada modo.

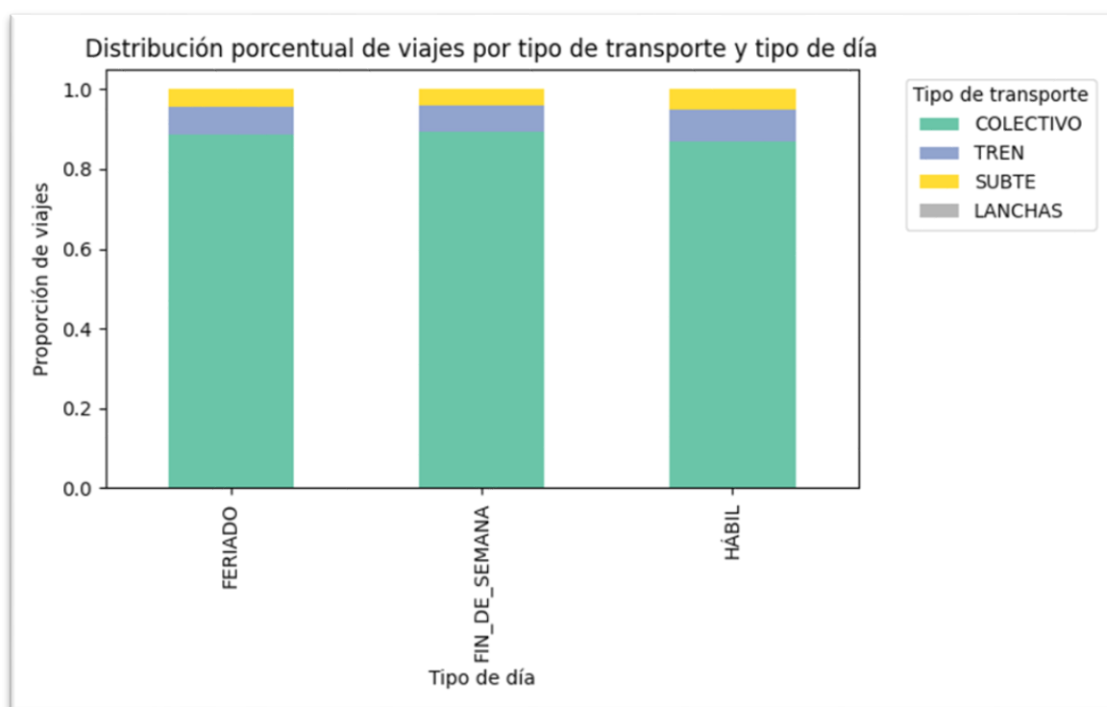


Ilustración 13 Distribución porcentual de viajes por tipo de transporte y tipo de día

## 7.12. Primeras 10 provincias más demandante - clasificación por tipo de día

Se observa un impacto claro del tipo de día sobre la demanda de transporte:

- Los días hábiles concentran la mayor cantidad de viajes diarios, seguidos por los fines de semana y finalmente los feriados.
- Esta tendencia es especialmente marcada en transportes bajo jurisdicción nacional, y en las provincias de Buenos Aires y Corrientes.

La caída en la demanda durante feriados se explica por la suspensión de actividades laborales y escolares, lo que demuestra la estrecha relación entre la movilidad y el ritmo de la vida productiva.

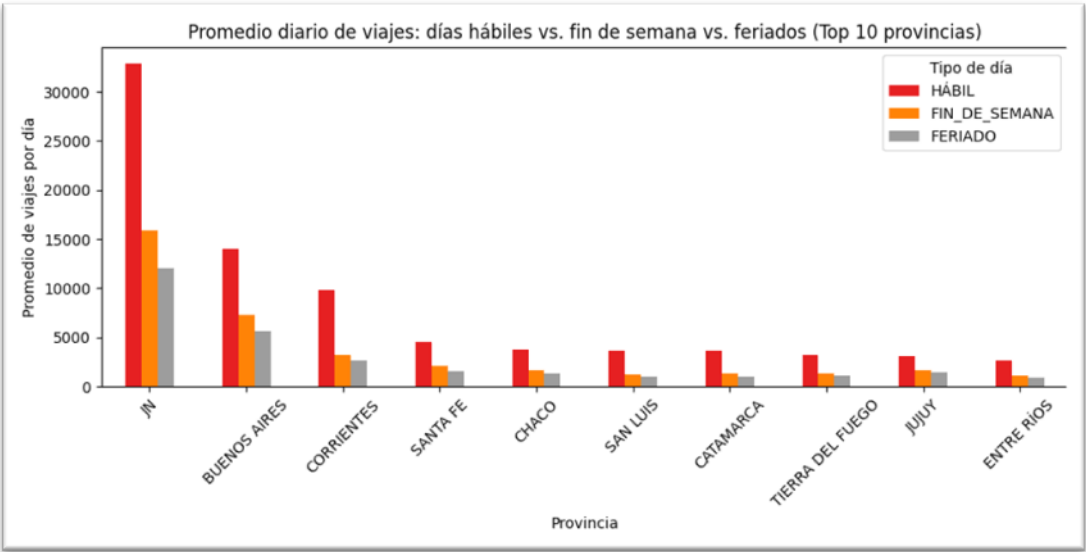


Ilustración 14 Promedio de viajes por tipo de día (Top 10 provincias)

## 8. Resultados obtenidos comparación Enero-Mayo 2024 vs 2025

### 8.1. Análisis comparativo de las transacciones SUBE en el transporte público

La tarjeta SUBE constituye el principal medio de pago del sistema de transporte público en Argentina, y sus registros permiten monitorear la evolución de la demanda en los distintos modos de transporte urbano e interurbano. En esta sección se analiza la evolución de la cantidad de viajes realizados utilizando SUBE, desagregada por tipo de transporte y tipo de día.

### 8.2. Análisis de la evolución de los viajes en transporte público

El siguiente gráfico presenta la cantidad total acumulada de viajes realizados en todo el país durante el período Enero–Mayo, comparando 2025 contra 2024. Se debe tener en cuenta que las transacciones registradas son exclusivamente con la tarjeta SUBE, que si bien es el principal medio de pago actualmente existen otros medios de pago.

Analizando el gráfico se observa una caída interanual en el volumen total de los viajes para los primeros cinco meses en los principales medios de transporte: subte, tren y colectivo. Esto puede explicarse a diferentes factores como en la modalidad de trabajo (remoto o híbrido), modificaciones en los patrones de modalidad y la adopción de medios de pagos alternativos como billeteras digitales, tarjetas bancarias contactless.

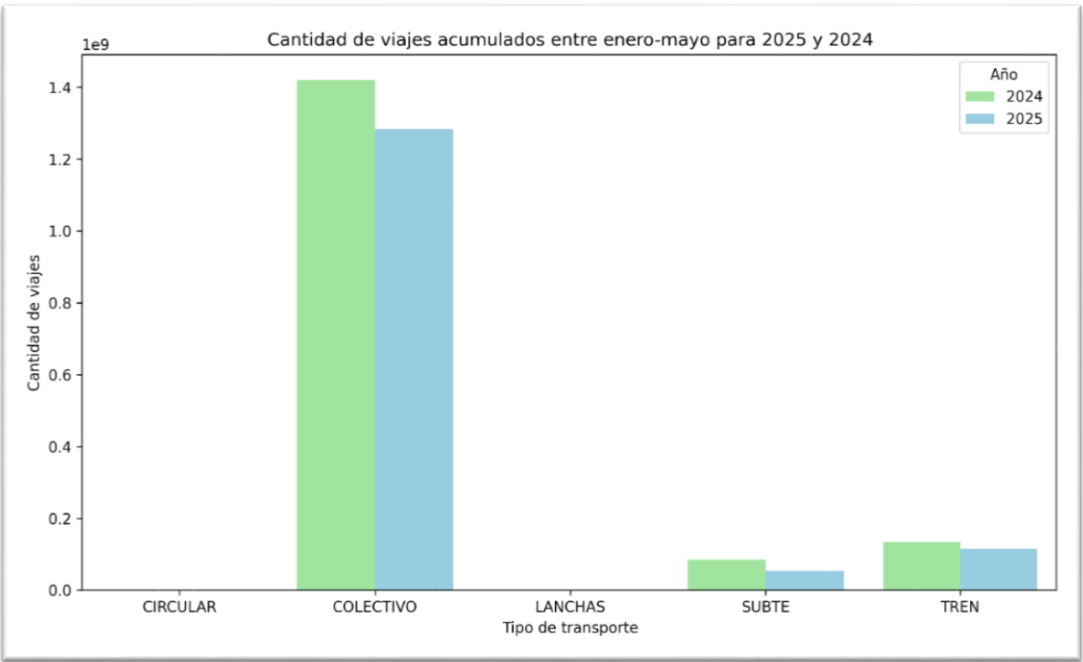


Ilustración 15 Evolución acumulada de viajes enero–mayo 2024 vs 2025 (total nacional)

8.3. Análisis de evolución de la cantidad de viajes por tipo de transporte

8.3.1. Subte

El siguiente gráfico muestra la evolución mensual de los viajes pagados con SUBE en el subte, comparando 2025 con el mismo período del año anterior. Se puede observar una caída interanual sostenida que se profundiza especialmente en el mes de marzo alcanzando una baja cercana al 50% en el mes de mayo. Se debe tener en cuenta que el subte fue uno de los primeros en implementar otras alternativas como medio de pago.

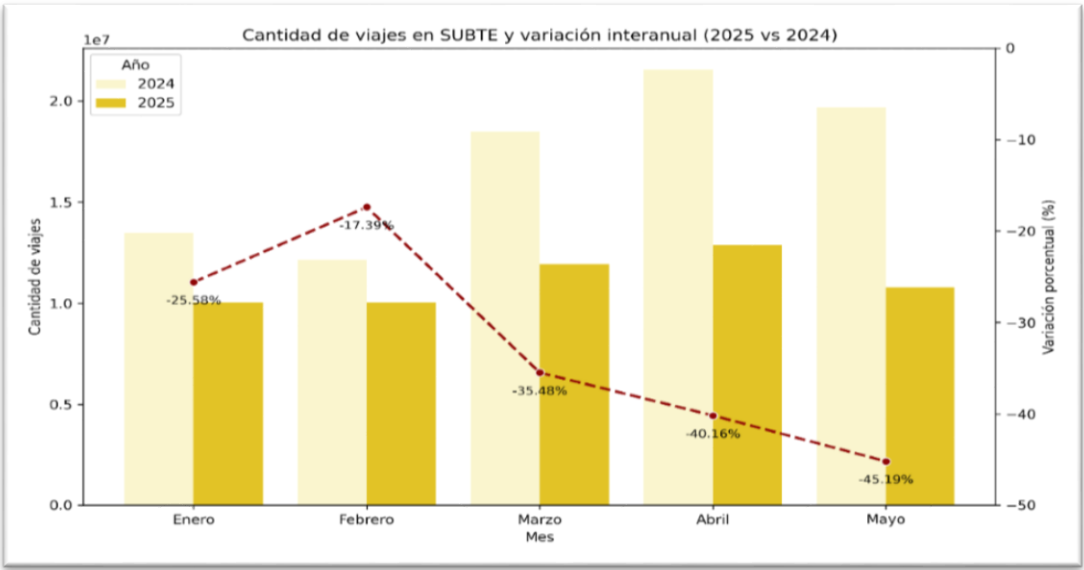


Ilustración 16 Evolución mensual de viajes en SUBTE con SUBE (enero–mayo 2024 vs 2025)

8.3.2. Tren

El siguiente gráfico muestra la comparación interanual de los viajes abonados con SUBE en los trenes. La caída es menos marcada que en el subte, destacándose una reducción de entre 3% y 15% entre enero y abril, pero profundizándose en mayo con una baja del 30%.

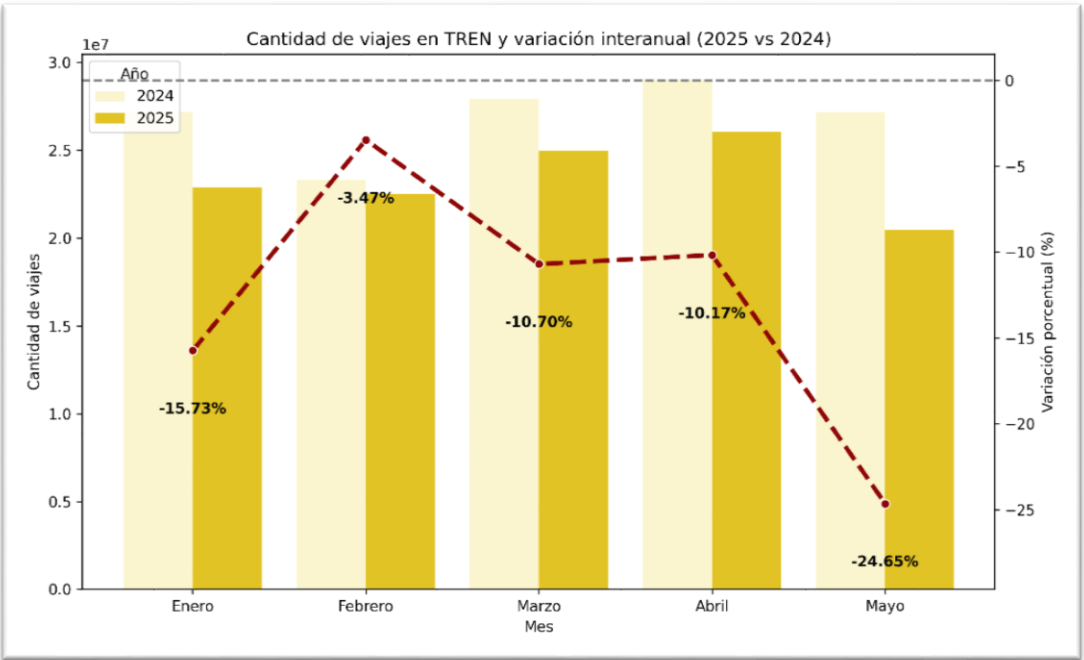


Ilustración 17 Evolución mensual de viajes en TREN con SUBE (enero–mayo 2024 vs 2025)

8.3.3. Colectivo

El gráfico analiza la evolución de los viajes en colectivo pagados con SUBE, mostrando un comportamiento mixto. Durante enero y febrero se observa una caída interanual moderada (alrededor del 10%), pero en marzo y abril las cifras incluso superan levemente las del año anterior, reflejando un pequeño repunte. Sin embargo, en mayo vuelve a registrarse una fuerte contracción, con una baja del 28%.

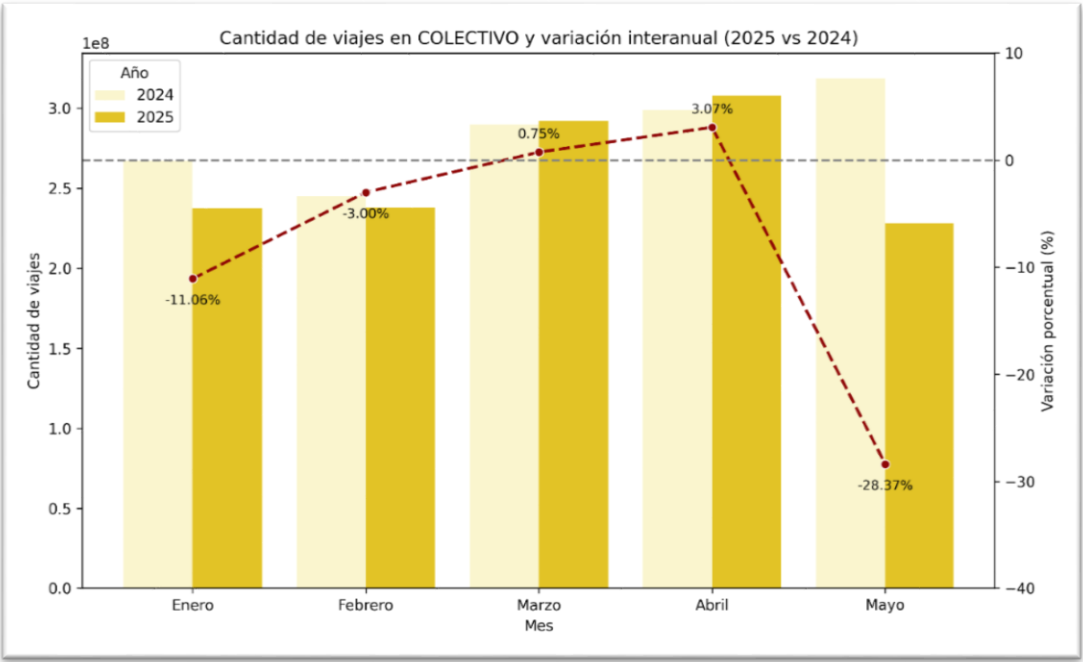


Ilustración 18 Evolución mensual de viajes en COLECTIVO con SUBE (enero–mayo 2024 vs 2025)

8.4. Análisis de evolución de la cantidad de viajes por día y tipo de transporte

A continuación, realizaremos un análisis de los gráficos de dispersión que presentan la evolución diaria de la cantidad de viajes registrados mediante la tarjeta SUBE entre enero de 2024 y mayo de 2025, discriminados según el tipo de día (hábil, fin de semana y feriado) y el modo de transporte utilizado: colectivo, subte y tren.

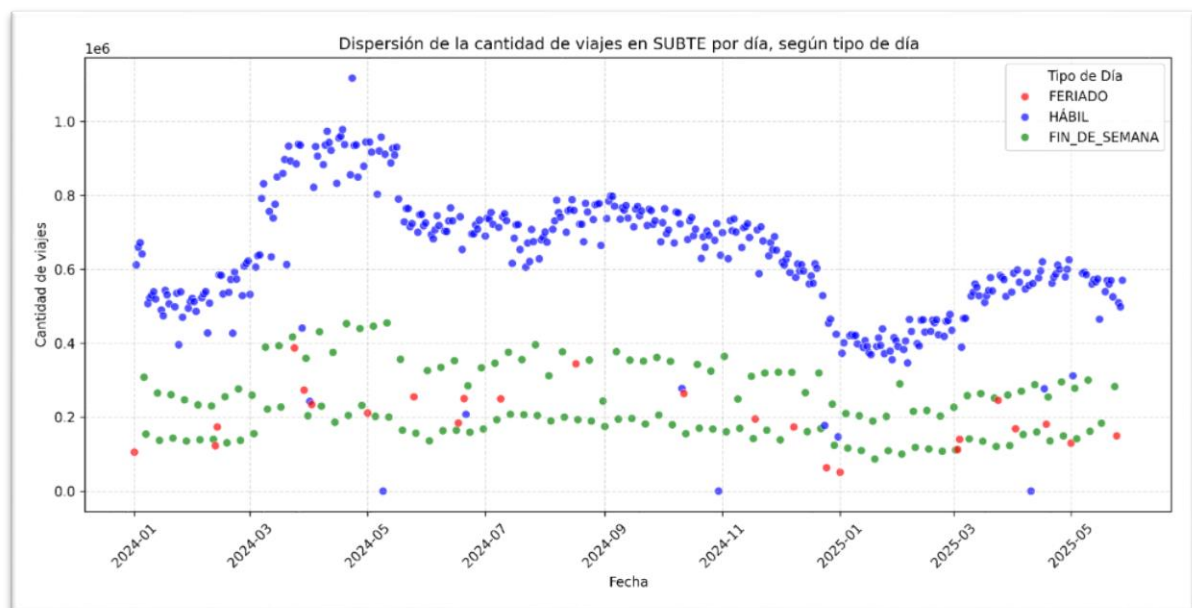
En términos generales, se observa una marcada estacionalidad semanal en los tres casos, con una clara diferencia entre la mayor cantidad de viajes durante los días hábiles y una caída significativa en fines de semana y feriados. Asimismo, se identifica cierta estacionalidad anual con caídas en períodos vacacionales como enero y repuntes hacia marzo, especialmente en los modos de transporte más vinculados a actividades laborales o educativas. A lo largo del período analizado, también se percibe una reducción interanual en los volúmenes de viajes, particularmente visible en los primeros meses de 2025, lo cual puede estar relacionado con factores como el crecimiento del trabajo remoto, cambios en los hábitos de movilidad y la adopción creciente de medios de pago alternativos no registrados por la tarjeta SUBE.



### 8.4.1. Subte

El siguiente gráfico se puede observar un fuerte componente estacional tanto semanal como anual. En particular, se destaca una caída marcada en la cantidad de viajes durante enero de cada año, seguida por una recuperación paulatina hacia marzo.

Esta estacionalidad es consistente con el uso intensivo del subte en actividades laborales y escolares, que disminuyen en el verano. Además, se observa una caída interanual en los días hábiles de 2025 respecto de 2024.



*Ilustración 19 Evolución diaria de viajes en SUBTE por tipo de día (enero 2024 a mayo 2025)*

### 8.4.2. Tren

En el caso del tren, el gráfico revela una evolución relativamente estable a lo largo del período, con una clara diferenciación entre los días hábiles y los fines de semana o feriados. Durante los días laborales, la cantidad de viajes se ubica en torno al millón, mientras que en los días no hábiles cae notablemente, a valores que oscilan entre los 200 mil y 700 mil viajes.

También se advierte una baja en los volúmenes durante los meses de verano, especialmente en enero, en línea con el receso escolar y las vacaciones. Si bien se percibe una leve disminución en los niveles diarios durante el primer tramo de 2025 en comparación con el mismo período del año anterior, esta caída no resulta tan pronunciada.

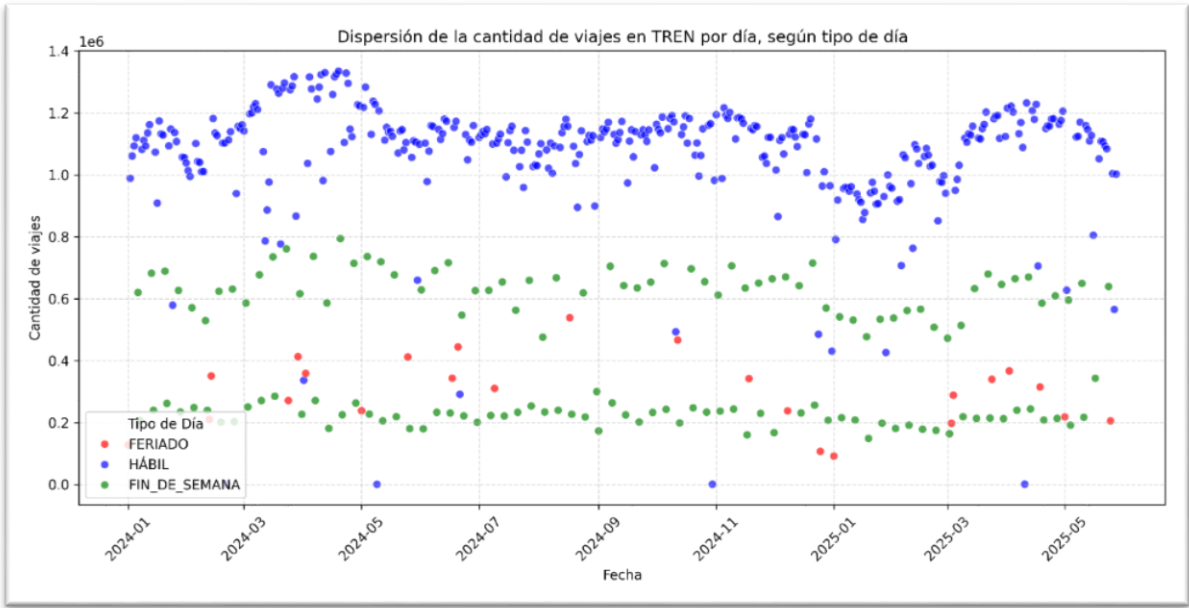


Ilustración 20 Evolución diaria de viajes en TREN por tipo de día (enero 2024 a mayo 2025)

8.4.3. Colectivo

Por su parte, el gráfico del colectivo presenta los mayores volúmenes diarios entre los tres modos de transporte analizados. En los días hábiles se superan con frecuencia los 10 millones de viajes, mientras que los fines de semana y feriados muestran una importante reducción, con valores que pueden ubicarse por debajo de los 4 millones. A lo largo del período considerado, se observa una cierta estabilidad en el patrón de uso, aunque con señales de descenso leve hacia comienzos de 2025. Esta caída, sin embargo, no es abrupta, lo que sugiere que, a diferencia de otros modos, el colectivo todavía conserva una alta proporción de usuarios que utilizan la tarjeta SUBE como medio principal de pago.

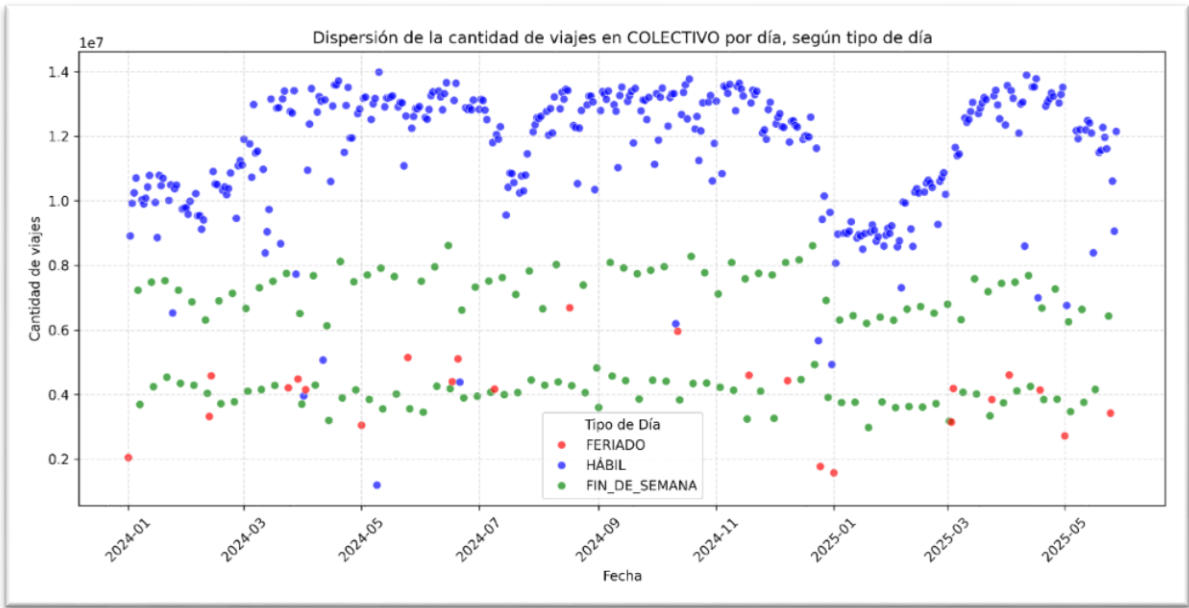


Ilustración 21 Evolución diaria de viajes en COLECTIVO por tipo de día (enero 2024 a mayo 2025)

## 9. Conclusiones

El presente trabajo, se lleva a cabo generando un análisis de la demanda del transporte público argentino mediante registros del sistema SUBE. A través de este análisis se ha permitido validar empíricamente la potencia metodológica del Análisis Exploratorio de Datos (EDA) como herramienta fundacional en procesos analíticos sobre sistemas complejos.

En el abordaje, se obtuvo un análisis profundo del año 2024 y, por otro, una primera comparación interanual con los meses de Enero a Mayo de 2025, en un contexto de cambios tecnológicos y nuevos hábitos de movilidad.

### 1. Valor del análisis exploratorio de datos en estudios de movilidad urbana

El propósito principal de EDA es analizar los datos antes de hacer suposiciones. Puede ayudar a identificar errores, así como comprender mejor los patrones dentro de los datos, detectar valores atípicos o eventos anómalos, o encontrar relaciones que puedan existir entre las variables.

El EDA no se limitó a una fase preliminar del análisis, sino que se constituyó como un proceso reflexivo y generador de conocimiento. A través de su aplicación se logró:

- Detectar valores erróneos y registros inconsistentes (por ejemplo, valores negativos de viajes o nulos).
- Identificar outliers estructurales en regiones como AMBA, cuya alta concentración de viajes refleja fenómenos urbanos y demográficos.
- Obtener una comprensión multivariable de los datos (temporal, modal, territorial), lo cual permitió generar hipótesis fundadas sobre el comportamiento de la demanda de transporte.

### 2. Enriquecimiento del dataset como pilar para análisis contextualmente significativos

El uso de técnicas de web scraping y APIs externas (Nager.Date) permitió dotar a los datos originales de contexto temporal asociado al comportamiento de los datos en función de los feriados y fines de semana. Gracias al uso combinado de estas técnicas se pudieron generar y obtener variables como TIPO\_DIA y MOTIVO\_FERIADO.

Este paso fue clave para interpretar con mayor precisión los comportamientos observados, en particular la vinculación de la movilidad con el calendario sociolaboral argentino.

### 3. Procesos de limpieza e imputación como etapas críticas en la fiabilidad analítica

El proceso de limpieza de datos realizado en este estudio fue una etapa crítica que permitió garantizar la validez, consistencia y utilidad analítica del dataset SUBE 2024/2025.

Esta fase se basó tomando decisiones metodológicas fundamentadas a preservar la representatividad del fenómeno observado y evitar sesgos en el análisis posterior.

Una de las primeras acciones implementadas fue la **conversión precisa de tipos de datos**. Se implementaron acciones clave como la conversión correcta de fechas para los análisis temporales (en particular la interpretación de la columna `DIA_TRANSPORTE` como variable de tipo fecha (*datetime*), lo que permitió la correcta implementación de los análisis temporales), la imputación contextualizada de valores faltantes en campos críticos (como *JURISDICCION* y *PROVINCIA*) —especialmente en registros del subte— y la estandarización de etiquetas que evitó distorsiones categóricas.

En cuanto a los **valores anómalos**, se descartaron los registros con cantidades negativas, al tiempo que se conservaron los outliers altos por su posible vínculo con situaciones reales de alta demanda.

Por último, en el dataset 2024 se incorporaron variables derivadas tales como *DIA\_SEMANA*, *TIPO\_DIA* y *CANTIDAD\_LOG*, que enriquecieron el análisis y facilitaron la interpretación.

#### 4. Evidencia empírica de la estructuración de la movilidad pública en torno a la actividad productiva

El análisis temporal mostró una concentración del 87,7% del volumen de viajes en días hábiles, en contraste con la significativa reducción observada durante fines de semana y feriados. Este hallazgo confirma la estrecha dependencia entre movilidad y ciclos laborales-escolares (en todos sus niveles), lo cual tiene implicancias directas para el diseño de políticas de transporte, asignación de recursos y planificación de frecuencias.

#### 5. Asimetría en la utilización del tipo de transporte público

Los datos evidencian una hegemonía del colectivo como medio de transporte, seguido a distancia por el tren, el subte y, marginalmente, las lanchas. Esta distribución refuerza la centralidad de los sistemas de transporte terrestre motorizado en la red nacional y exige un análisis crítico sobre su sustentabilidad, resiliencia y equidad territorial.

#### 6. Desigualdad espacial en la demanda y presencia de valores atípicos estructurales

Se detectaron valores extremos en los registros de grandes aglomerados urbanos como el AMBA, que reflejan fenómenos estructurales vinculados a la densidad poblacional, concentración de actividades económicas y centralización de infraestructuras tales como ubicación de zonas residenciales, departamentos, zonas comerciales, entre otros. Esto plantea la necesidad de redefinir los criterios de tratamiento de outliers en estudios empíricos con fuerte dependencia espacial.

#### 7. Comparación 2024–2025: el EDA como base para análisis longitudinales

La comparación entre los meses de Enero–Mayo de 2024 y Enero-Mayo de 2025 permitió observar una **reducción significativa en el volumen de viajes registrados con SUBE**, en particular:

- Subte: descenso de hasta el **50%** en mayo de 2025.
- Tren: caída progresiva, especialmente marcada en mayo (**30%**).
- Colectivo: comportamiento mixto, con repunte intermedio y caída posterior (**hasta 28% en mayo**).

Estas variaciones fueron interpretadas como si se tratasen de los nuevos hábitos de la población (trabajo remoto, cambios en el uso del transporte) y, especialmente, por la adopción creciente de medios de pago alternativos no capturados por la tarjeta SUBE, como billeteras virtuales y tarjetas contactless.

#### 8. **Proyección hacia análisis longitudinales y formulación de políticas públicas basadas en evidencia**

El desarrollo del presente informe ha establecido una línea base para la comparación interanual donde sirve como base para adoptar medios nuevos de pago.

En particular, la comparación entre ambos años (2024 y 2025) ya evidencia una **disminución en el volumen de transacciones registradas con SUBE**.

Esta observación inicial —fundamentada en evidencia empírica— sienta las bases para estudios futuros que analicen el impacto de nuevas modalidades sobre la trazabilidad de los datos, la inclusión digital y la planificación del transporte.

A partir de la incorporación de billeteras virtuales, tarjetas contactless y próximamente en todas las provincias sobre el uso de pagos por medio de QR, la trazabilidad del uso de transporte público dejará de tener una representación absoluta por medio del uso de la tarjeta SUBE.

Desde una perspectiva de política pública, el trabajo ofrece un **insumo valioso para la toma de decisiones informada**, permitiendo evaluar en el tiempo la efectividad de medidas adoptadas, anticipar demandas emergentes y diseñar intervenciones basadas en datos reales, contextualizados y comparables.

En este sentido, el informe no solo analiza el presente, sino que **proyecta una agenda de análisis longitudinal sostenido**, con potencial para contribuir a un sistema de movilidad más eficiente, inclusivo y adaptado a las dinámicas contemporáneas y sugiere incorporar a la base de datos del 2025 y venideros información sobre trazabilidad de transporte públicos clasificándolos por medios de pago (Billeteras virtuales, Tarjeta Contactless, QR y Tarjeta SUBE).

## 10. Anexos: Scripts Ejecutados

A continuación, se incluyen los scripts desarrollados y utilizados durante el proceso de análisis exploratorio de datos. Cada anexo contiene el código correspondiente, con comentarios explicativos para facilitar su comprensión.

- **Anexo 1:** `api_tipo_dias2024.py`  
Script utilizado para clasificar los días del dataframe como *HÁBIL*, *FERIADO* o *FIN\_DE\_SEMANA*, utilizando la API pública de Nager.Date.
- **Anexo 2:** `scraping_consulta_robots2024.py`  
Script que verifica las restricciones de scraping impuestas por el sitio web fuente mediante la lectura del archivo `robots.txt`.
- **Anexo 3:** `scraping_feriados_lanacion2024.py`  
Script de scraping para obtener la lista de feriados nacionales del año 2024 desde el sitio de La Nación, y enriquecer el dataframe con el motivo de cada feriado.

### 10.1. Anexo: `api_tipo_dias2024.py`

```
import requests          # librería para hacer peticiones HTTP
import pandas as pd      # librería estándar para trabajar con dataframes

# 1) Leer el dataset y convierte la columna DIA_TRANSPORTE a formato de fecha
df_sube = pd.read_csv("dat-ab-usos-2024.csv", parse_dates=["DIA_TRANSPORTE"])

# 2) Obtener feriados de Argentina en 2024
feriados_2024 =
requests.get("https://date.nager.at/api/v3/PublicHolidays/2024/AR").json()
fechas_feriados = set(item["date"] for item in feriados_2024) # El uso de
set(...) garantiza que las fechas estén sin duplicados

# 3) Función para clasificar cada fecha
def clasificar_fecha(fecha): # se define la funcion clasificar_fecha del
parametro fecha
    fecha_str = fecha.strftime("%Y-%m-%d")
    if fecha_str in fechas_feriados:
        return "FERIADO"
    elif fecha.weekday() >= 5: # .weekday() devuelve un número de 0 a 6
(5=sábado, 6=domingo)
        return "FIN_DE_SEMANA"
    else:
        return "HÁBIL"

# 4) Día de la semana traducido
dias_traduccion = {
    "Monday": "LUNES", "Tuesday": "MARTES", "Wednesday": "MIÉRCOLES",
```

```

    "Thursday": "JUEVES", "Friday": "VIERNES", "Saturday": "SÁBADO", "Sunday":
    "DOMINGO"
}
df_sube["DIA_SEMANA"] =
df_sube["DIA_TRANSPORTE"].dt.day_name().map(dias_traduccion)

# 5) Clasificación del tipo de día
df_sube["TIPO_DIA"] = df_sube["DIA_TRANSPORTE"].apply(clasificar_fecha)

# 6) Guardar el nuevo dataset
df_sube.to_csv("df-sube-2024-tipo-dia.csv", index=False, encoding="utf-8-sig")
# index=False → Le dice a pandas que no incluya la columna del índice
# encoding="utf-8-sig" → Especifica el tipo de codificación útil si después
usas Excel
print('✔ Proceso finalizado. Se han agregado las columnas "DIA_SEMANA" y
"TIPO_DIA" (desde API) al Dataset.')
print('📁 Archivo guardado como: df-sube-2024-tipo-dia.csv')

```

## 10.2. Anexo: api\_tipo\_dias2025.py

```

import requests          # librería para hacer peticiones HTTP
import pandas as pd      # librería estándar para trabajar con dataframes

# 1) Leer el dataset y convierte la columna DIA_TRANSPORTE a formato de fecha
df_sube = pd.read_csv("https://archivos-
datos.transporte.gob.ar/upload/Dat_Ab_Usos/dat-ab-usos-2025.csv",
parse_dates=["DIA_TRANSPORTE"])

# 2) Obtener feriados de Argentina en 2025
feriados_2025 =
requests.get("https://date.nager.at/api/v3/PublicHolidays/2025/AR").json()
fechas_feriados = set(item["date"] for item in feriados_2025) # El uso de
set(...) garantiza que las fechas estén sin duplicados

# 3) Función para clasificar cada fecha
def clasificar_fecha(fecha): # se define la funcion clasificar_fecha del
parametro fecha
    fecha_str = fecha.strftime("%Y-%m-%d")
    if fecha_str in fechas_feriados:
        return "FERIADO"
    elif fecha.weekday() >= 5: # .weekday() devuelve un número de 0 a 6
(5=sábado, 6=domingo)
        return "FIN_DE_SEMANA"
    else:
        return "HÁBIL"

```

```
# 4) Día de la semana traducido
dias_traduccion = {
    "Monday": "LUNES", "Tuesday": "MARTES", "Wednesday": "MIÉRCOLES",
    "Thursday": "JUEVES", "Friday": "VIERNES", "Saturday": "SÁBADO", "Sunday":
    "DOMINGO"
}
df_sube["DIA_SEMANA"] =
df_sube["DIA_TRANSPORTE"].dt.day_name().map(dias_traduccion)

# 5) Clasificación del tipo de día
df_sube["TIPO_DIA"] = df_sube["DIA_TRANSPORTE"].apply(clasificar_fecha)

# 6) Guardar el nuevo dataset
df_sube.to_csv("df-sube-2025-tipo-dia.csv", index=False, encoding="utf-8-sig")
# index=False → Le dice a pandas que no incluya la columna del índice
# encoding="utf-8-sig" → Especifica el tipo de codificación útil si después
usas Excel
print('✔ Proceso finalizado. Se han agregado las columnas "DIA_SEMANA" y
"TIPO_DIA" (desde API) al Dataset.')
print('📁 Archivo guardado como: df-sube-2025-tipo-dia.csv')
```

### 10.3. Anexo: scraping\_consulta\_robots.py

```
import requests

# Consulta el archivo robots.txt del sitio de Wikipedia
print(requests.get("https://www.lanacion.com.ar/robots.txt").text)

'''
El archivo nO prohíbe de forma general el scraping.
☹ Solo restringe el acceso a ciertas rutas, como:
- /sinbarreras, /newsletters/, /registracion, etc.
- URLs con ?utm_* en los parámetros.
- Algunas rutas específicas como /buscador, /pf/api/..., y ciertos artículos
puntuales.

# 🖱 La página de feriados https://www.lanacion.com.ar/feriados/2024/ no está
bloqueada
# ✔ Se puede hacer scraping de esa página.
'''
```

### 10.4. Anexo: scraping\_feriados\_lanacion2024.py

```
import requests
from bs4 import BeautifulSoup
```



```
import pandas as pd
from datetime import datetime
import re

# Mapeo de nombres de meses en español a número de mes
meses = {
    "enero": 1,
    "febrero": 2,
    "marzo": 3,
    "abril": 4,
    "mayo": 5,
    "junio": 6,
    "julio": 7,
    "agosto": 8,
    "septiembre": 9,
    "octubre": 10,
    "noviembre": 11,
    "diciembre": 12
}

feriados = {}

# 1. Descargar y parsear la página de feriados 2024
url = "https://www.lanacion.com.ar/feriados/2024/"
headers = {"User-Agent": "Mozilla/5.0"}
resp = requests.get(url, headers=headers, timeout=10)
soup = BeautifulSoup(resp.content, "html.parser")

# 2. Buscar cada bloque mensual
calendarios = soup.find_all("div", class_="holidays-card-calendar")

for calendario in calendarios:
    # 2.1 Extraer el nombre del mes desde el <a class="com-link">
    encabezado = calendario.find("div", class_="labeled-calendar")
    if not encabezado:
        continue

    link_mes = encabezado.find("a", class_="com-link")
    if not link_mes:
        continue

    nombre_mes = link_mes.text.strip().lower()
    numero_mes = meses.get(nombre_mes)
    if not numero_mes:
        print(f"⚠️ Mes no reconocido: {nombre_mes}")
        continue
```

```

# 2.2 Iterar la lista de feriados de ese mes
ul = calendario.find("ul", class_="holidays-list")
if not ul:
    continue

for li in ul.find_all("li"):
    dia_tag = li.find("span", class_=re.compile(r"--"))
    motivo_tag = li.find("h4", class_="com-text")
    if not dia_tag or not motivo_tag:
        continue

    try:
        dia = int(dia_tag.text.strip())
        motivo = motivo_tag.text.strip().upper()
        fecha = datetime(2024, numero_mes, dia).strftime("%Y-%m-%d")
        feriados[fecha] = motivo
    except Exception as e:
        print(f"✗ Error procesando {nombre_mes} {li}: {e}")

# 3. Mostrar resultados en consola
print("Feriados encontrados:")
for fecha, motivo in sorted(feriados.items()):
    print(f"{fecha} → {motivo}")

# 4. Leer CSV de SUBE y etiquetar feriados
df_sube = pd.read_csv("df-sube-2024-tipo-dia.csv",
    parse_dates=["DIA_TRANSPORTE"])
df_sube["MOTIVO_FERIADO"] = df_sube["DIA_TRANSPORTE"].dt.strftime("%Y-%m-%d").map(feriados).fillna("NO FERIADO")

# 5. Guardar resultado
df_sube.to_csv("df-sube-2024.csv", index=False, encoding="utf-8-sig")
print("✓ Scraping finalizado. Archivo guardado como df-sube-2024.csv")
print(f"✓ Total de feriados encontrados: {len(feriados)}")

```

### 10.5. Anexo: scraping\_feriados\_lanacion2025.py

```

import requests
from bs4 import BeautifulSoup
import pandas as pd
from datetime import datetime
import re

# Mapeo de nombres de meses en español a número de mes
meses = {
    "enero": 1,
    "febrero": 2,

```

```
"marzo": 3,
"abril": 4,
"mayo": 5,
"junio": 6,
"julio": 7,
"agosto": 8,
"septiembre": 9,
"octubre": 10,
"noviembre": 11,
"diciembre": 12
}

feriados = {}

# 1. Descargar y parsear la página de feriados 2025
url = "https://www.lanacion.com.ar/feriados/2025/"
headers = {"User-Agent": "Mozilla/5.0"}
resp = requests.get(url, headers=headers, timeout=10)
soup = BeautifulSoup(resp.content, "html.parser")

# 2. Buscar cada bloque mensual
calendarios = soup.find_all("div", class_="holidays-card-calendar")

for calendario in calendarios:
    # 2.1 Extraer el nombre del mes desde el <a class="com-link">
    encabezado = calendario.find("div", class_="labeled-calendar")
    if not encabezado:
        continue

    link_mes = encabezado.find("a", class_="com-link")
    if not link_mes:
        continue

    nombre_mes = link_mes.text.strip().lower()
    numero_mes = meses.get(nombre_mes)
    if not numero_mes:
        print(f"⚠ Mes no reconocido: {nombre_mes}")
        continue

    # 2.2 Iterar la lista de feriados de ese mes
    ul = calendario.find("ul", class_="holidays-list")
    if not ul:
        continue

    for li in ul.find_all("li"):
        dia_tag = li.find("span", class_=re.compile(r"--"))
        motivo_tag = li.find("h4", class_="com-text")
```

```

        if not dia_tag or not motivo_tag:
            continue

        try:
            dia = int(dia_tag.text.strip())
            motivo = motivo_tag.text.strip().upper()
            fecha = datetime(2025, numero_mes, dia).strftime("%Y-%m-%d")
            feriados[fecha] = motivo
        except Exception as e:
            print(f"✗ Error procesando {nombre_mes} {li}: {e}")

# 3. Mostrar resultados en consola
print("Feriados encontrados:")
for fecha, motivo in sorted(feriados.items()):
    print(f"{fecha} → {motivo}")

# 4. Leer CSV de SUBE y etiquetar feriados
df_sube = pd.read_csv("df-sube-2025-tipo-dia.csv",
                      parse_dates=["DIA_TRANSPORTE"])
df_sube["MOTIVO_FERIADO"] = df_sube["DIA_TRANSPORTE"].dt.strftime("%Y-%m-%d").map(feriados).fillna("NO FERIADO")

# 5. Guardar resultado
df_sube.to_csv("df-sube-2025.csv", index=False, encoding="utf-8-sig")
print("✓ Scraping finalizado. Archivo guardado como df-sube-2025.csv")
print(f"✓ Total de feriados encontrados: {len(feriados)}")

```

## 10.6. Anexo: eda\_sube2024.py

```

import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import numpy as np

# 1. CARGA DEL DATASET
df = pd.read_csv("df-sube-2024.csv", parse_dates=["DIA_TRANSPORTE"])

# 2. PRIMER VISTAZO
print("-" * 80 + "\nDIMENSIONES DEL DATASET SUBE 2024\n")
print(df.shape)

print("-" * 80 + "\nINFORMACION GENERAL DEL DATASET\n")
df.info()

print("-" * 80 + "\nRANGO DE FECHAS DE 'DIA_TRANSPORTE'\n")

```

```

print("Rango de fechas:", df["DIA_TRANSPORTE"].min(), "a",
df["DIA_TRANSPORTE"].max())

print("-" * 80 + "\nESTADÍSTICAS DESCRIPTIVAS (NUMÉRICAS)\n")
print(df.describe(include=[np.number]).round(2)) # filtra que solo se incluyan
columnas de tipo numérico y con 2 decimales

# Verificación de duplicados
print("-" * 80 + "\nVERIFICACIÓN DE DUPLICADOS\n")
print("Duplicados:", df.duplicated().sum())

# 3[ ]----- E S T A D Í S T I C A S   D E S C R I P T I V A S -----
# ♦ Detectar filas con valores negativos en la columna 'CANTIDAD'
negativos = df[df["CANTIDAD"] < 0]
print("\nCantidad de filas con valores negativos en CANTIDAD:",
len(negativos))
print(negativos[["DIA_TRANSPORTE", "TIPO_TRANSPORTE", "CANTIDAD"]].head(10))
print("\n⚠ Se encontraron 3 valores anómalos en CANTIDAD (valores
negativos). Estos se consideran errores o correcciones no documentadas.")

# ♦ Eliminar filas con valores negativos en 'CANTIDAD' para evitar problemas
en el análisis
df = df[df["CANTIDAD"] >= 0]
print("✓[ ] Filas con valores negativos en 'CANTIDAD' eliminadas
correctamente.")

# ♦ Mostrar resumen estadístico actualizado de la columna 'CANTIDAD'
print("-" * 50 + "\nESTADÍSTICAS ACTUALIZADAS\n" + "-" * 50)
print("Resumen estadístico de la columna CANTIDAD:")
print(df["CANTIDAD"].describe().round(2))

# ♦ Histograma de la variable 'CANTIDAD' (antes de la transformación)
plt.figure()
df["CANTIDAD"].hist(bins=50, color="salmon", grid=False)
plt.title("Distribución de CANTIDAD de viajes")
plt.xlabel("Cantidad de viajes")
plt.ylabel("Frecuencia")
plt.tight_layout()
plt.savefig("graficos/sube2024_histograma_cantidad.png")

# ♦ Aplicar transformación logarítmica para reducir la asimetría de la
distribución
df["CANTIDAD_LOG"] = np.log1p(df["CANTIDAD"]) # Se usa log1p para evitar
problemas con ceros (log(0) no está definido)

```

```

# 4. Histograma de la variable transformada 'CANTIDAD_LOG'
plt.figure()
df["CANTIDAD_LOG"].hist(bins=50, color="lightseagreen", grid=False)
plt.title("Distribución logarítmica de CANTIDAD")
plt.xlabel("log(1 + Cantidad de viajes)")
plt.ylabel("Frecuencia")
plt.tight_layout()
plt.savefig("graficos/sube2024_histograma_cantidad_log.png")

# 4. REVISAR COLUMNAS CONSTANTES (desvío estándar = 0)
print("-" * 50 + "\nCOLUMNAS CON DESVIACION ESTANDAR = 0\n")
stds = df.std(numeric_only=True) # Calcula la desviación estándar solo de las columnas numéricas
cero_std = stds[stds == 0.0] # Filtra las columnas cuya desviación estándar es exactamente cero (es decir, columnas constantes)
print(cero_std if not cero_std.empty else "Ninguna columna numérica es constante.")

# 5. VALORES FALTANTES
print("-" * 50 + "\nVALORES FALTANTES POR COLUMNA\n" + "-" * 50)
print(df.isna().sum()) # Devuelve un df del mismo tamaño con valores booleanos, luego suma los True por columna

print("\n----- ANÁLISIS DE VALORES NULOS -----")

# Filas con JURISDICCION nula
nulos_jur = df[df["JURISDICCION"].isna()]
print("\nTipos de transporte con JURISDICCION nula:")
print(nulos_jur["TIPO_TRANSPORTE"].value_counts())

# Filas con PROVINCIA nula
nulos_prov = df[df["PROVINCIA"].isna()]
print("\nTipos de transporte con PROVINCIA nula:")
print(nulos_prov["TIPO_TRANSPORTE"].value_counts())

# Filas con MUNICIPIO nulo
nulos_mun = df[df["MUNICIPIO"].isna()]
print("\nTipos de transporte con MUNICIPIO nulo:")
print(nulos_mun["TIPO_TRANSPORTE"].value_counts())

# Correccion datos nulos

```

```

es_subte = df["TIPO_TRANSPORTE"] == "SUBTE" # Filtrar filas donde
TIPO_TRANSPORTE es SUBTE
df.loc[es_subte, "JURISDICCION"] = df.loc[es_subte,
"JURISDICCION"].fillna("CABA") # Completar valores nulos con información
conocida
df.loc[es_subte, "PROVINCIA"] = df.loc[es_subte, "PROVINCIA"].fillna("CIUDAD
AUTÓNOMA DE BUENOS AIRES")
df.loc[es_subte, "MUNICIPIO"] = df.loc[es_subte, "MUNICIPIO"].fillna("CABA")

# Verificación final de nulos
print("\n----- Verificación de valores nulos tras corrección en SUBTE -----")
print(df[["JURISDICCION", "PROVINCIA", "MUNICIPIO"]].isna().sum())

# Filas con PROVINCIA nula
nulos_prov = df[df["PROVINCIA"].isna()]
print("\nValores Nulos por Provincia:")
print(nulos_prov[["TIPO_TRANSPORTE", "NOMBRE_EMPRESA", "LINEA",
"AMBA"]].to_string(index=False))

# Filas con MUNICIPIO nulo
nulos_mun = df[df["MUNICIPIO"].isna()]
print("\nValores Nulos por Municipio:")
print(nulos_mun[["TIPO_TRANSPORTE", "NOMBRE_EMPRESA", "LINEA",
"AMBA"]].to_string(index=False))

# COLECTIVO - Empresa 9 de Julio SRL - Línea 500 Santa Fe
colectivo_sfe = (df["TIPO_TRANSPORTE"] == "COLECTIVO") & (df["LINEA"] ==
"LINEA_500I_SFE")
df.loc[colectivo_sfe, "PROVINCIA"] = "SANTA FE"
df.loc[colectivo_sfe, "MUNICIPIO"] = "SANTA FE"

# TREN - Tren del Valle
tren_valle = (df["TIPO_TRANSPORTE"] == "TREN") & (df["LINEA"] == "FFCC TREN
DEL VALLE")
df.loc[tren_valle, "PROVINCIA"] = "JN"
df.loc[tren_valle, "MUNICIPIO"] = "SD"

print("\n----- Verificación final tras imputación específica: -----")
print(df[["JURISDICCION", "PROVINCIA", "MUNICIPIO"]].isna().sum())
print("\n✓ Filas con valores nulos corregidas correctamente.")

# 6----- O U T L I E R S -----
# Estadísticas descriptivas por AMBA (si/no)
print("\n----- Estadísticas descriptivas -----")
print(df.groupby('AMBA')['CANTIDAD'].describe())

```

```

# Boxplot para comparar distribuciones por AMBA
plt.figure(figsize=(10,6))
sns.boxplot(x='AMBA', y='CANTIDAD', data=df)
plt.title('Distribución de CANTIDAD según AMBA')
plt.xlabel('AMBA (si/no)')
plt.ylabel('CANTIDAD')
plt.tight_layout()
plt.savefig("graficos/sube2024_boxplot_amba.png")

print("\n----- Identificación de outliers por AMBA (sin eliminar) -----")
def identificar_outliers_por_grupo(df, columna_grupo, columna_valor):
    for grupo, subdf in df.groupby(columna_grupo):
        Q1 = subdf[columna_valor].quantile(0.25)
        Q3 = subdf[columna_valor].quantile(0.75)
        IQR = Q3 - Q1
        lower_bound = max(Q1 - 1.5 * IQR, 0)
        upper_bound = Q3 + 1.5 * IQR

        total = len(subdf)
        normales = subdf[(subdf[columna_valor] >= lower_bound) &
(subdf[columna_valor] <= upper_bound)]
        outliers = total - len(normales)

        print(f"{columna_grupo} = {grupo}")
        print(f"  Q1: {Q1:.1f}, Q3: {Q3:.1f}, IQR: {IQR:.1f}")
        print(f"  Rango normal: [{lower_bound:.1f}, {upper_bound:.1f}]")
        print(f"  Total filas: {total}, Outliers detectados: {outliers}\n")

# Llamamos a la función pero NO reasignamos df
identificar_outliers_por_grupo(df, 'AMBA', 'CANTIDAD')

print("----- Análisis de outliers -----")
print("⚠️ Los valores considerados outliers podrían corresponder a situaciones reales")
print("(eventos masivos, paros, problemas técnicos), por lo que se optó por mantenerlos.")
print("En lugar de eliminarlos, se los identificó y analizó por separado para entender su impacto.\n")

# 7️⃣----- P E R F I L   T E M P O R A L -----
df["MES"] = df["DIA_TRANSPORTE"].dt.month # Extrae el nº de mes de la columna "DIA_TRANSPORTE" y crea la columna "MES" con ese valor

# 💎Total de viajes por mes

```



```

viajes_mes = df.groupby("MES")["CANTIDAD"].sum() # Agrupa el df por columna
"MES", suma los valores de columna "CANTIDAD" y guarda resultado en viajes_mes
plt.figure()
viajes_mes.plot(kind="bar", color="cyan") # viajes_mes es una serie con
meses como índice y la suma de viajes como valores - gráfico de barras
plt.title("Total de viajes por mes (2024)")
plt.xlabel("Mes")
plt.ylabel("Cantidad de viajes")

# Agregar etiquetas a las barras con rotación y separación vertical
for i, v in enumerate(viajes_mes):
    plt.text(i, v - 80_000_000, f"{int(v):,}", ha='center', va='bottom',
fontsize=8, rotation=90)

plt.tight_layout()
plt.savefig("graficos/sube2024_viajes_por_mes.png")

# ♦Total de viajes por día de la semana
orden_dias = ["LUNES", "MARTES", "MIÉRCOLES", "JUEVES", "VIERNES", "SÁBADO",
"DOMINGO"]
viajes_dsem = df.groupby("DIA_SEMANA")["CANTIDAD"].sum().reindex(orden_dias) #
reindex cambia el orden del índice de la serie pq coincida con el orden
establecido
plt.figure()
viajes_dsem.plot(kind="bar", color="coral")
plt.title("Total de viajes por día de la semana")
plt.xlabel("Día de la semana")
plt.ylabel("Cantidad de viajes")

# Agregar etiquetas a las barras con rotación y separación vertical
for i, v in enumerate(viajes_dsem):
    plt.text(i, v - 180_000_000, f"{int(v):,}", ha='center', va='bottom',
fontsize=8, rotation=90)

plt.tight_layout()
plt.savefig("graficos/sube2024_viajes_por_dia_semana.png")

# ♦Evolucion mensual por tipo de transporte
df['MES_AÑO'] = df['DIA_TRANSPORTE'].dt.to_period('M').astype(str) # Crear
columna MES_AÑO

# Agrupar por MES_AÑO y TIPO_TRANSPORTE
df_mes = (
    df
    .groupby(['MES_AÑO', 'TIPO_TRANSPORTE'])['CANTIDAD']

```

```

        .sum()
        .reset_index()
    )

# Asegurar orden cronológico de los meses
df_mes['MES_ANO'] = pd.to_datetime(df_mes['MES_ANO'])
df_mes = df_mes.sort_values('MES_ANO')

plt.figure(figsize=(12,6))
sns.lineplot(
    data=df_mes,
    x='MES_ANO',
    y='CANTIDAD',
    hue='TIPO_TRANSPORTE',
    marker='o'
)

plt.title('Evolución mensual de viajes por tipo de transporte - 2024')
plt.xlabel('Mes')
plt.ylabel('Cantidad de viajes')
plt.xticks(rotation=45)
plt.legend(title='Tipo de transporte', bbox_to_anchor=(1.02,1), loc='upper
left')
plt.tight_layout()
plt.savefig("graficos/sube2024_evolucion_mensual.png")

# 8.----- PERFIL POR CATEGORIA -----
# ♦Por tipo de transporte
viajes_tipo = df.groupby("TIPO_TRANSPORTE")["CANTIDAD"].sum()
plt.figure()
colores = ["#B0E0E6", "#87CEEB", "#C1E1C1", "#A7C7E7", "#C6E2FF", "#98FB98"]
viajes_tipo.plot(
    # Graficar pie chart con etiquetas separadas y sin
    # decimales en porcentajes
    kind="pie",
    autopct='%1.1f%%',      # un decimal
    startangle=90,
    pctdistance=0.85,      # distancia del porcentaje al centro
    labeldistance=1.12,    # distancia de las etiquetas fuera de la torta
    colors=colores
)

plt.title("Distribución de viajes por tipo de transporte")
plt.ylabel("")
plt.tight_layout()
plt.savefig("graficos/sube2024_viajes_por_tipo_transporte.png")

```

```
# ◆Comparativa: HÁBIL / FERIADO / FIN DE SEMANA
# Agrupamos por día (fecha) y sumamos los viajes totales de ese día
viajes_diarios = df.groupby(["DIA_TRANSPORTE",
"TIPO_DIA"])[ "CANTIDAD"].sum().reset_index()

# Se calcula el promedio diario por tipo de día
promedios = viajes_diarios.groupby("TIPO_DIA")["CANTIDAD"].mean()

plt.figure()
ax = promedios.plot(kind="bar", color="#C1E1C1")

plt.title("Viajes promedio por tipo de día")
plt.xlabel("Tipo de día")
plt.ylabel("Promedio de viajes")
plt.tight_layout()

# Agregar valores dentro de las barras
for i, valor in enumerate(promedios):
    ax.text(i, valor * 0.95, f'{valor:,.0f}', ha='center', va='top',
color='black')

plt.savefig("graficos/sube2024_promedio_viajes_tipo_dia.png")

# ◆Días hábil vs no hábil
df["ES_HABIL"] = df["TIPO_DIA"] == "HÁBIL"
conteo_habiles = df.groupby("ES_HABIL")["CANTIDAD"].sum()

# Calcular porcentajes
total = conteo_habiles.sum()
porcentajes = (conteo_habiles / total * 100).round(1)

plt.figure()
bars = conteo_habiles.plot(kind="bar", color=["#FF6347", "#3CB371"])
plt.title("Total de viajes: días hábiles vs no hábiles")
plt.xlabel("¿Es día hábil?")
plt.ylabel("Cantidad de viajes")
plt.xticks([0, 1], labels=["No", "Sí"], rotation=0)

# Agregar etiquetas de porcentaje sobre cada barra
for i, (valor, porcentaje) in enumerate(zip(conteo_habiles, porcentajes)):
    plt.text(i, valor, f"{porcentaje}%", ha='center', va='bottom',
fontsize=10, fontweight='bold')

plt.tight_layout()
plt.savefig("graficos/sube2024_viajes_habil_vs_no.png")
```

```

# 9[ ]----- BOX PLOT DE CANTIDAD POR TIPO DE TR
A N S P O R T E -----
plt.figure()
df.boxplot(column="CANTIDAD", by="TIPO_TRANSPORTE", grid=False) # dibuja el
boxplot agrupado por tipo de transporte, sin mostrar la cuadrícula.
plt.yscale("log") # Mejora la visualización si hay outliers extremos
plt.title("Distribución de viajes por tipo de transporte")
plt.suptitle("")
plt.xlabel("Tipo de transporte")
plt.ylabel("Cantidad de viajes")
plt.tight_layout()
plt.savefig("graficos/sube2024_boxplot_cantidad_por_tipo.png")

# 10[ ]----- ANALISIS DE CANTIDAD DE VIAJES PO
R M O T I V O D E F E R I A D O -----
# Filtrar solo los días feriados reales con motivo válido
feriados_df = df[(df["TIPO_DIA"] == "FERIADO") & (df["MOTIVO_FERIADO"] != "NO
FERIADO")].copy()

# Agrupar por motivo del feriado y sumar cantidad de viajes
viajes_por_feriado =
feriados_df.groupby("MOTIVO_FERIADO")["CANTIDAD"].sum().sort_values(ascending=
True)

plt.figure(figsize=(10, 6))
viajes_por_feriado.plot(kind="barh", color="#DFBFF3")

# Agregar valores al final de cada barra
for i, (valor, label) in enumerate(zip(viajes_por_feriado,
viajes_por_feriado.index)):
    plt.text(valor - 1000, i, f"{int(valor):,}", va="center", ha="right",
color="black", fontsize=9)

plt.xlabel("Cantidad total de viajes")
plt.ylabel("Motivo del feriado")
plt.title("Cantidad de viajes por feriado (2024)")
plt.tight_layout()
plt.savefig("graficos/sube2024_cantidad_viajes_por_feriado.png")

# 11[ ]----- CANTIDAD TOTAL DE VIAJES POR DI
A D E L A S E M A N A Y T I P O D E T R A N S P O R T E -----
# Agrupar por día de la semana y tipo de transporte sumando la cantidad de
viajes
tabla_pivot = df.pivot_table(
    index="DIA_SEMANA",

```

```

        columns="TIPO_TRANSPORTE",
        values="CANTIDAD",
        aggfunc="sum"
    )

# Reordenar filas para que el orden de los días sea correcto
orden_dias = ["LUNES", "MARTES", "MIÉRCOLES", "JUEVES", "VIERNES", "SÁBADO",
              "DOMINGO"]
tabla_pivot = tabla_pivot.reindex(orden_dias)

# Graficar heatmap con seaborn
plt.figure(figsize=(10,6))
sns.heatmap(
    tabla_pivot,
    annot=True,          # Mostrar los valores dentro de cada celda
    fmt=".0f",           # Sin decimales
    cmap="Reds",         # Paleta de colores
    cbar_kws={'label': 'Cantidad de viajes'}
)

plt.title("Cantidad total de viajes por día de la semana y tipo de transporte
(2024)")
plt.xlabel("Tipo de transporte")
plt.ylabel("Día de la semana")
plt.tight_layout()
plt.savefig("graficos/sube2024_heatmap_dia_semana_tipo_transporte.png")

# 121---- CANTIDAD TOTAL DE VIAJES POR TIP
O DE DIA Y TIPO DE TRANSPORTE ----
#Distribución porcentual de viajes por tipo de transporte y día

# # Lectura
# nombre_archivo = "dat-sube-2024-tipo-dia.csv"
# # Leer el archivo CSV en un DataFrame llamado df
# df = pd.read_csv(nombre_archivo)

viajes_por_dia_tipo = (
    df.groupby(['TIPO_DIA', 'TIPO_TRANSPORTE'])['CANTIDAD']
      .sum()
      .reset_index()
)

# Normalizamos para obtener proporciones por tipo de día
total_por_tipo_dia =
viajes_por_dia_tipo.groupby('TIPO_DIA')['CANTIDAD'].transform('sum')
viajes_por_dia_tipo['PROPORCION'] = viajes_por_dia_tipo['CANTIDAD'] /
total_por_tipo_dia

```

```

# Pivot para graficar
tabla_prop = viajes_por_dia_tipo.pivot(index='TIPO_DIA',
columns='TIPO_TRANSPORTE', values='PROPORCION')
tabla_prop = tabla_prop[['COLECTIVO', 'TREN', 'SUBTE', 'LANCHAS']] # Orden
deseado

tabla_prop.plot(kind='bar', stacked=True, figsize=(8,5), colormap='Set2')
plt.title('Distribución porcentual de viajes por tipo de transporte y tipo de
día')
plt.ylabel('Proporción de viajes')
plt.xlabel('Tipo de día')
plt.legend(title='Tipo de transporte', bbox_to_anchor=(1.05, 1), loc='upper
left')
plt.tight_layout()
plt.savefig("graficos/sube2024_modal_por_tipo_dia.png")

# 131----- CANTIDAD TOTAL DE VIAJES POR PRIM
ERAS 10 PROVINCIA - CLASIFICACIÓN POR T
IPO DE DÍA -----

# Filtramos solo feriados y días hábiles
feriados = df[df["TIPO_DIA"] == "FERIADO"]
finde semana = df[df["TIPO_DIA"] == "FIN_DE_SEMANA"]
habiles = df[df["TIPO_DIA"] == "HÁBIL"]

# Promedio diario de viajes por provincia y tipo de día
prom_feriado =
feriados.groupby("PROVINCIA")["CANTIDAD"].mean().rename("FERIADO")
prom_habil = habiles.groupby("PROVINCIA")["CANTIDAD"].mean().rename("HÁBIL")
prom_fin_de_semana =
finde semana.groupby("PROVINCIA")["CANTIDAD"].mean().rename("FIN_DE_SEMANA")

comparativo = pd.concat([prom_habil, prom_fin_de_semana, prom_feriado],
axis=1).dropna().sort_values("HÁBIL", ascending=False).head(10)

comparativo.plot(kind="bar", figsize=(10,5), colormap="Set1")
plt.title("Promedio diario de viajes: días hábiles vs. fin de semana vs.
feriados (Top 10 provincias)")
plt.ylabel("Promedio de viajes por día")
plt.xlabel("Provincia")
plt.xticks(rotation=45)
plt.legend(title="Tipo de día", loc="upper right")
plt.tight_layout()
plt.savefig("graficos/sube2024_feriados_vs_habiles_por_provincia.png")

```

```
#1[4]---- MENSAJES FINALES DE CONFIRMACION ---
---
print("\n✓ El análisis EDA se completo correctamente.")
print("✓ El análisis fue realizado sin excluir registros del dataset
principal.\n")
print("📁 Se generaron los siguientes archivos:")
print(" - sube2024_boxplot_amba.png")
print(" - sube2024_boxplot_cantidad_por_tipo.png")
print(" - sube2024_cantidad_viajes_por_feriado.png")
print(" - sube2024_evolucion_mensual.png")
print(" - sube2024_feriados_vs_habiles_por_provincia.png")
print(" - sube2024_heatmap_dia_semana_tipo_transporte.png")
print(" - sube2024_histograma_cantidad.png")
print(" - sube2024_histograma_cantidad_log.png")
print(" - sube2024_modal_por_tipo_dia.png")
print(" - sube2024_promedio_viajes_tipo_dia.png")
print(" - sube2024_viajes_habil_vs_no_habil.png")
print(" - sube2024_viajes_por_dia_semana.png")
print(" - sube2024_viajes_por_mes.png")
print(" - sube2024_viajes_por_tipo_transporte.png")

# 5. Guardar resultado
df.to_csv("sube2024_eda.csv", index=False, encoding="utf-8-sig")
print("✓ Datos limpios y procesados son guardados en 'dat_sube2024_eda.csv'")

print("\nProceso terminado.\n\n")
```

### 10.7. Anexo: eda\_sube2025.py

```
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import numpy as np

# 1[4]---- CARGA DEL DATASET ----
df = pd.read_csv("df-sube-2025.csv", parse_dates=["DIA_TRANSPORTE"])

# 2[4]---- PRIMER VISTAZO ----
print("-" * 80 + "\nDIMENSIONES DEL DATASET SUBE 2025\n")
print(df.shape)

print("-" * 80 + "\nINFORMACION GENERAL DEL DATASET\n")
df.info()

print("-" * 80 + "\nRANGO DE FECHAS DE 'DIA_TRANSPORTE'\n")
```

```

print("Rango de fechas:", df["DIA_TRANSPORTE"].min(), "a",
df["DIA_TRANSPORTE"].max())

print("-" * 80 + "\nESTADÍSTICAS DESCRIPTIVAS (NUMÉRICAS)\n")
print(df.describe(include=[np.number]).round(2)) # filtra que solo se incluyan
columnas de tipo numérico y con 2 decimales

# Verificación de duplicados
print("-" * 80 + "\nVERIFICACIÓN DE DUPLICADOS\n")
print("Duplicados:", df.duplicated().sum())

# 3[ ]----- E S T A D Í S T I C A S   D E S C R I P T I V A S -----
# ♦ Detectar filas con valores negativos en la columna 'CANTIDAD'
negativos = df[df["CANTIDAD"] < 0]
print("\nCantidad de filas con valores negativos en CANTIDAD:",
len(negativos))
print(negativos[["DIA_TRANSPORTE", "TIPO_TRANSPORTE", "CANTIDAD"]].head(10))
print("\n⚠ Se encontró un valor anómalo en CANTIDAD (valor negativo). Estos
se considera un error o corrección no documentada.")

# ♦ Eliminar filas con valores negativos en 'CANTIDAD' para evitar problemas
en el análisis
df = df[df["CANTIDAD"] >= 0]
print("✓[ ] Filas con valores negativos en 'CANTIDAD' eliminadas
correctamente.")

# ♦ Mostrar resumen estadístico actualizado de la columna 'CANTIDAD'
print("-" * 50 + "\nESTADÍSTICAS ACTUALIZADAS\n" + "-" * 50)
print("Resumen estadístico de la columna CANTIDAD:")
print(df["CANTIDAD"].describe().round(2))

# ♦ Histograma de la variable 'CANTIDAD' (antes de la transformación)
plt.figure()
df["CANTIDAD"].hist(bins=50, color="salmon", grid=False)
plt.title("Distribución de CANTIDAD de viajes")
plt.xlabel("Cantidad de viajes")
plt.ylabel("Frecuencia")
plt.tight_layout()
plt.savefig("graficos/sube2025_histograma_cantidad.png")

# ♦ Aplicar transformación logarítmica para reducir la asimetría de la
distribución
df["CANTIDAD_LOG"] = np.log1p(df["CANTIDAD"]) # Se usa log1p para evitar
problemas con ceros (log(0) no está definido)

```



```

# 🎯 Histograma de la variable transformada 'CANTIDAD_LOG'
plt.figure()
df["CANTIDAD_LOG"].hist(bins=50, color="lightseagreen", grid=False)
plt.title("Distribución logarítmica de CANTIDAD")
plt.xlabel("log(1 + Cantidad de viajes)")
plt.ylabel("Frecuencia")
plt.tight_layout()
plt.savefig("graficos/sube2025_histograma_cantidad_log.png")

# 4.----- REVISAR COLUMNAS CONSTANTES (desvío estándar = 0) -----
print("-" * 50 + "\nCOLUMNAS CON DESVIACION ESTANDAR = 0\n")
stds = df.std(numeric_only=True) # Calcula la desviación estándar solo de las columnas numéricas
cero_std = stds[stds == 0.0] # Filtra las columnas cuya desviación estándar es exactamente cero (es decir, columnas constantes)
print(cero_std if not cero_std.empty else "Ninguna columna numérica es constante.")

# 5.----- VALORES FALTANTES -----
print("-" * 50 + "\nVALORES FALTANTES POR COLUMNA\n" + "-" * 50)
print(df.isna().sum()) # Devuelve un df del mismo tamaño con valores booleanos, luego suma los True por columna

print("\n----- ANÁLISIS DE VALORES NULOS -----")

# Filas con JURISDICCION nula
nulos_jur = df[df["JURISDICCION"].isna()]
print("\nTipos de transporte con JURISDICCION nula:")
print(nulos_jur["TIPO_TRANSPORTE"].value_counts())

# Filas con PROVINCIA nula
nulos_prov = df[df["PROVINCIA"].isna()]
print("\nTipos de transporte con PROVINCIA nula:")
print(nulos_prov["TIPO_TRANSPORTE"].value_counts())

# Filas con MUNICIPIO nulo
nulos_mun = df[df["MUNICIPIO"].isna()]
print("\nTipos de transporte con MUNICIPIO nulo:")
print(nulos_mun["TIPO_TRANSPORTE"].value_counts())

# Correccion datos nulos
es_subte = df["TIPO_TRANSPORTE"] == "SUBTE" # Filtrar filas donde TIPO_TRANSPORTE es SUBTE
df.loc[es_subte, "JURISDICCION"] = df.loc[es_subte, "JURISDICCION"].fillna("CABA") # Completar valores nulos con información conocida

```

```

df.loc[es_subte, "PROVINCIA"] = df.loc[es_subte, "PROVINCIA"].fillna("CIUDAD
AUTÓNOMA DE BUENOS AIRES")
df.loc[es_subte, "MUNICIPIO"] = df.loc[es_subte, "MUNICIPIO"].fillna("CABA")

# Verificación final de nulos
print("\n----- Verificación de valores nulos tras corrección en SUBTE -----")
print(df[["JURISDICCION", "PROVINCIA", "MUNICIPIO"]].isna().sum())

# Obtenemos los valores únicos para cada columna
print("\nValores únicos de Jurisdicción")
print(df['JURISDICCION'].unique().tolist())
print("\nValores únicos de Provincia")
print(df['PROVINCIA'].unique().tolist())
print("\nValores únicos de Municipio")
print(df['MUNICIPIO'].unique().tolist())

# Realizamos corrección de escritura
df['JURISDICCION'] = df['JURISDICCION'].replace("C.A.B.A", "CABA")
df['PROVINCIA'] = df['PROVINCIA'].replace("C.A.B.A", "CIUDAD AUTÓNOMA DE
BUENOS AIRES")

# Verificamos correcciones y que no esten mal escritos
print("\nValores únicos de Jurisdicción")
print(df['JURISDICCION'].unique().tolist())
print("\nValores únicos de Provincia")
print(df['PROVINCIA'].unique().tolist())

# Guardar dataset limpio en nuevo archivo CSV
df.to_csv("df-sube-2025.csv", index=False)

print("\n✓ Archivo 'df-sube-2025' guardado con los nulos corregidos y otras
correcciones.")

#6[ ]---- MENSAJES FINALES DE CONFIRMACION ----
print("\n✓ Se realiza limpieza y correccion del dataframe para una correcta
comparacion con el dataframe del año 2024.")

print("\nProceso terminado.\n\n")

```

## 10.8. Anexo: comparativa\_2025vs2024.py

```

import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns

# Load the cleaned datasets
df_2024 = pd.read_csv("df-sube-2024.csv")

```

```

df_2025 = pd.read_csv("df-sube-2025.csv")

# Add a 'ANIO' column to each DataFrame to distinguish the data
df_2024['ANIO'] = 2024
df_2025['ANIO'] = 2025

# Concatenate the two DataFrames
df_combined = pd.concat([df_2024, df_2025], ignore_index=True)

# Convert 'DIA_TRANSPORTE' to datetime if it's not already (important for consistency)
df_combined['DIA_TRANSPORTE'] = pd.to_datetime(df_combined['DIA_TRANSPORTE'])

# Save the combined dataset
df_combined.to_csv("dat-sube-2024-2025-combinado.csv", index=False,
encoding="utf-8-sig")

print("✔ Proceso finalizado. Se han unido los datasets de 2024 y 2025.")
print("📁 Archivo guardado como: dat-sube-2024-2025-combinado.csv")

#-----Análisis -----
---

# Carga de datos
df = pd.read_csv('dat-sube-2024-2025-combinado.csv')

# Convertimos DIA_TRANSPORTE en fecha
df["DIA_TRANSPORTE"] = pd.to_datetime(df["DIA_TRANSPORTE"], errors='coerce')
# Extraemos el mes
df["MES"] = df["DIA_TRANSPORTE"].dt.month

# Meses
meses_es = {
    1: 'Enero', 2: 'Febrero', 3: 'Marzo', 4: 'Abril', 5: 'Mayo', 6: 'Junio',
    7: 'Julio', 8: 'Agosto', 9: 'Septiembre', 10: 'Octubre', 11: 'Noviembre',
    12: 'Diciembre'
}

#-----PRIMER GRÁFICO-----
-----

viajes_por_tipo_transporte = (
    df[(df["MES"].isin([1, 2, 3, 4, 5]))]
    .groupby(["TIPO_TRANSPORTE", "ANIO"])["CANTIDAD"]
    .sum()
    .reset_index()
)

```

```

# Gráfico comparativo 2025 vs 2024 - cantidad de viajes por tipo de transporte
plt.figure(figsize=(10, 6))
sns.barplot(
    data=viajes_por_tipo_transporte,
    x="TIPO_TRANSPORTE",
    y="CANTIDAD",
    hue="ANIO",
    palette={2024: "lightgreen", 2025: "skyblue"}
)

plt.title("Cantidad de viajes acumulados entre enero-mayo para 2025 y 2024")
plt.xlabel("Tipo de transporte")
plt.ylabel("Cantidad de viajes")
plt.xticks(rotation=0)
plt.legend(title="Año")
plt.tight_layout()
plt.savefig("graficos/cantidad_de_viajes_acumulado_enero-abril2025vs2024.png",
    dpi=300)

print("Gráfico guardado como 'cantidad_de_viajes_acumulado_enero-
mayo2025vs2024.png'")

# ----- Segundo gráfico -----

# Filtramos TIPO_TRANSPORTE solo por SUBTE y MES de enero-mayo (son los únicos
meses completos de 2025)
viajes_subte_mensual = (
    df[(df["TIPO_TRANSPORTE"] == "SUBTE") & (df["MES"].isin([1, 2, 3, 4, 5]))]
    .groupby(["ANIO", "MES"])["CANTIDAD"]
    .sum()
    .reset_index()
)

# Agregamos los nombres del mes
viajes_subte_mensual["NOMBRE_MES"] = viajes_subte_mensual["MES"].map(meses_es)

# Orden correcto enero-mayo
orden_meses = [meses_es[i] for i in range(1, 6)]

# -----Tabla variación i.a. -----

var_2025vs2024 = viajes_subte_mensual.pivot(index="MES", columns="ANIO",
    values="CANTIDAD")
var_2025vs2024["VARIACION_%"] = ((var_2025vs2024[2025] - var_2025vs2024[2024])
    / var_2025vs2024[2024]) * 100

```

```

# Tabla limpia
tabla_variacion = var_2025vs2024[["VARIACION_%"]].reset_index()
tabla_variacion["NOMBRE_MES"] = tabla_variacion["MES"].map(meses_es)
tabla_variacion = tabla_variacion[["NOMBRE_MES", "VARIACION_%"]]

# Ordenar por MES antes de limpiar columnas
tabla_variacion = var_2025vs2024[["VARIACION_%"]].reset_index()
tabla_variacion["NOMBRE_MES"] = tabla_variacion["MES"].map(meses_es)

# Ordenar por MES (asegurado)
tabla_variacion = tabla_variacion.sort_values("MES")

# Redondear las variaciones
tabla_variacion["VARIACION_%"] = tabla_variacion["VARIACION_%"].round(2)

# Ahora sí, dejar solo nombre mes y variación
tabla_variacion = tabla_variacion[["NOMBRE_MES", "VARIACION_%"]]

#-----Armado de gráfico-----
-----

# Definir paleta de colores manual
colores = {2024: '#FFF9C4', 2025: '#FFD600'} # amarillo claro y amarillo
fuerte

fig, ax1 = plt.subplots(figsize=(10, 6))

# Gráfico de barras
sns.barplot(
    data=viajes_subte_mensual,
    x="NOMBRE_MES",
    y="CANTIDAD",
    hue="ANIO",
    order=orden_meses,
    ax=ax1,
    palette=colores
)

ax1.set_xlabel("Mes")
ax1.set_ylabel("Cantidad de viajes")
ax1.set_title("Cantidad de viajes en SUBTE y variación interanual (2025 vs
2024)")
ax1.legend(title="Año", loc="upper left")

# Crear segundo eje (para la variación porcentual)
ax2 = ax1.twinx()

```

```

sns.lineplot(
    data=tabla_variacion,
    x="NOMBRE_MES",
    y="VARIACION_%",
    marker='o',
    linewidth=2,
    color='darkred',
    linestyle='--',
    ax=ax2
)

ax2.set_ylabel("Variación porcentual (%)")
ax2.set_ylim(-50, 0) # Límite de eje secundario

# Añadir etiquetas encima de cada punto
for i, row in tabla_variacion.iterrows():
    ax2.text(
        row["NOMBRE_MES"],
        row["VARIACION_%"] + (1 if row["VARIACION_%"] >= 0 else -1.5),
        f"{row['VARIACION_%']:.2f}%",
        ha='center',
        va='bottom' if row["VARIACION_%"] >= 0 else 'top',
        fontsize=9,
        color='black'
    )

# Línea horizontal en 0%
ax2.axhline(0, color='gray', linestyle='--')

# Ajuste de leyendas
if ax2.get_legend() is not None:
    ax2.get_legend().remove()

plt.tight_layout()
plt.savefig("graficos/barras_linea_variacion_subte_colores.png", dpi=300)

print("✔ Gráfico combinado guardado como 'barras_linea_variacion_subte_colores.png'")

#-----Tercer gráfico TREN-----

# Filtrar TREN
viajes_tren_mensual = (
    df[(df["TIPO_TRANSPORTE"] == "TREN") & (df["MES"].isin([1, 2, 3, 4, 5]))]
    .groupby(["ANIO", "MES"])["CANTIDAD"]
    .sum()
    .reset_index()

```

```
)

# Nombres del mes
viajes_tren_mensual["NOMBRE_MES"] = viajes_tren_mensual["MES"].map(meses_es)

# Pivot para variación %
var_tren = viajes_tren_mensual.pivot(index="MES", columns="ANIO",
values="CANTIDAD")
var_tren["VARIACION_%"] = ((var_tren[2025] - var_tren[2024]) / var_tren[2024])
* 100

# Tabla limpia
tabla_var_tren = var_tren[["VARIACION_%"]].reset_index()
tabla_var_tren["NOMBRE_MES"] = tabla_var_tren["MES"].map(meses_es)
tabla_var_tren = tabla_var_tren.sort_values("MES")
tabla_var_tren["VARIACION_%"] = tabla_var_tren["VARIACION_%"].round(2)

# Gráfico Tren de barras y variación i.a.
fig, ax1 = plt.subplots(figsize=(10, 6))

# Barras
sns.barplot(
    data=viajes_tren_mensual,
    x="NOMBRE_MES",
    y="CANTIDAD",
    hue="ANIO",
    order=orden_meses,
    ax=ax1,
    palette=colores
)

ax1.set_xlabel("Mes")
ax1.set_ylabel("Cantidad de viajes")
ax1.set_title("Cantidad de viajes en TREN y variación interanual (2025 vs 2024)")
ax1.legend(title="Año", loc="upper left")

# Línea variación % en eje secundario
ax2 = ax1.twinx()

sns.lineplot(
    data=tabla_var_tren,
    x="NOMBRE_MES",
    y="VARIACION_%",
    marker='o',
    linewidth=3, # más gruesa
```

```

        color='darkred',
        linestyle='--',
        ax=ax2
    )

ax2.set_ylabel("Variación porcentual (%)")

# Ajustar límites para que se vea bien
y_min = min(0, tabla_var_tren["VARIACION_%"].min() - 5)
y_max = tabla_var_tren["VARIACION_%"].max() + 5
ax2.set_ylim(y_min, y_max)

# Etiquetas de % arriba
for i, row in tabla_var_tren.iterrows():
    ax2.text(
        row["NOMBRE_MES"],
        row["VARIACION_%"] + (3 if row["VARIACION_%"] >= 0 else -3),
        f"{row['VARIACION_%']:.2f}%",
        ha='center',
        va='bottom' if row["VARIACION_%"] >= 0 else 'top',
        fontsize=10,
        color='black',
        fontweight='bold'
    )

ax2.axhline(0, color='gray', linestyle='--')

if ax2.get_legend() is not None:
    ax2.get_legend().remove()

plt.tight_layout()
plt.savefig("graficos/barras_linea_variacion_tren_colores.png", dpi=300)

print("✔ Gráfico combinado MEJORADO guardado como  
'barras_linea_variacion_tren_colores.png'")

#-----Cuarto gráfico Colectivo-----

# Filtrar COLECTIVO
viajes_colectivo_mensual = (
    df[(df["TIPO_TRANSPORTE"] == "COLECTIVO") & (df["MES"].isin([1, 2, 3, 4, 5]))]
    .groupby(["ANIO", "MES"])["CANTIDAD"]
    .sum()
    .reset_index()
    .copy()

```



```
)

# Nombres del mes
viajes_colectivo_mensual["NOMBRE_MES"] =
viajes_colectivo_mensual["MES"].map(meses_es)

# Pivot para variación %
var_colectivo = viajes_colectivo_mensual.pivot(index="MES", columns="ANIO",
values="CANTIDAD")
var_colectivo["VARIACION_%"] = ((var_colectivo[2025] - var_colectivo[2024]) /
var_colectivo[2024]) * 100

# Tabla limpia
tabla_var_colectivo = var_colectivo[["VARIACION_%"]].reset_index().copy()
tabla_var_colectivo["NOMBRE_MES"] = tabla_var_colectivo["MES"].map(meses_es)
tabla_var_colectivo = tabla_var_colectivo.sort_values("MES")
tabla_var_colectivo["VARIACION_%"] =
tabla_var_colectivo["VARIACION_%"].round(2)

# Gráfico combinado
fig, ax1 = plt.subplots(figsize=(10, 6))

# Barras
sns.barplot(
    data=viajes_colectivo_mensual,
    x="NOMBRE_MES",
    y="CANTIDAD",
    hue="ANIO",
    order=orden_meses,
    ax=ax1,
    palette=colores
)

ax1.set_xlabel("Mes")
ax1.set_ylabel("Cantidad de viajes")
ax1.set_title("Cantidad de viajes en COLECTIVO y variación interanual (2025 vs
2024)")
ax1.legend(title="Año", loc="upper left")

# Línea variación %
ax2 = ax1.twinx()
ax2.set_ylim(-40, 10) # Fijar mínimo en -40

sns.lineplot(
    data=tabla_var_colectivo,
    x="NOMBRE_MES",
    y="VARIACION_%",
```

```

        marker='o',
        linewidth=2,
        color='darkred',
        linestyle='--',
        ax=ax2
    )

ax2.set_ylabel("Variación porcentual (%)")

# Etiquetas de % arriba
for _, row in tabla_var_colectivo.iterrows():
    ax2.text(
        row["NOMBRE_MES"],
        row["VARIACION_%"] + (1 if row["VARIACION_%"] >= 0 else -1.5),
        f"{row['VARIACION_%']:.2f}%",
        ha='center',
        va='bottom' if row["VARIACION_%"] >= 0 else 'top',
        fontsize=9,
        color='black'
    )

ax2.axhline(0, color='gray', linestyle='--')

if ax2.get_legend() is not None:
    ax2.get_legend().remove()

plt.tight_layout()
plt.savefig("graficos/barras_linea_variacion_colectivo_colores.png", dpi=300)

print("✔ Gráfico combinado guardado como
'barras_linea_variacion_colectivo_colores.png')

# -----GRÁFICOS DE DISPERSIÓN-----
#----- QUINTO GÁFICO-----

# Filtrar SUBTE y agrupar por día y tipo de día
subte_diario = (
    df[df["TIPO_TRANSPORTE"] == "SUBTE"]
    .groupby(["DIA_TRANSPORTE", "TIPO_DIA"])["CANTIDAD"]
    .sum()
    .reset_index()
    .sort_values("DIA_TRANSPORTE")
)

# Convertir a datetime

```

```
subte_diario["DIA_TRANSPORTE"] =
pd.to_datetime(subte_diario["DIA_TRANSPORTE"])

# Mapa de colores
colores = {"FERIADO": "red", "HÁBIL": "blue", "FIN_DE_SEMANA": "green"}

# Gráfico
plt.figure(figsize=(12, 6))
sns.scatterplot(
    data=subte_diario,
    x="DIA_TRANSPORTE",
    y="CANTIDAD",
    hue="TIPO_DIA",
    palette=colores,
    alpha=0.7
)

plt.xlabel("Fecha")
plt.ylabel("Cantidad de viajes")
plt.title("Dispersión de la cantidad de viajes en SUBTE por día, según tipo de
día")
plt.xticks(rotation=45)
plt.grid(True, linestyle='--', alpha=0.5)
plt.legend(title="Tipo de Día")

plt.tight_layout()
plt.savefig("graficos/dispersion_subte_por_dia.png", dpi=300)

print("✔ Gráfico SUBTE guardado como 'dispersion_subte_por_día.png'")

#----- SEXTO GÁFICO-----

# Filtrar TREN y agrupar por día y tipo de día
tren_diario = (
    df[df["TIPO_TRANSPORTE"] == "TREN"]
    .groupby(["DIA_TRANSPORTE", "TIPO_DIA"])["CANTIDAD"]
    .sum()
    .reset_index()
    .sort_values("DIA_TRANSPORTE")
)

# Convertir a datetime
tren_diario["DIA_TRANSPORTE"] = pd.to_datetime(tren_diario["DIA_TRANSPORTE"])

# Gráfico
plt.figure(figsize=(12, 6))
sns.scatterplot(
```

```

        data=tren_diario,
        x="DIA_TRANSPORTE",
        y="CANTIDAD",
        hue="TIPO_DIA",
        palette=colores,
        alpha=0.7
    )

plt.xlabel("Fecha")
plt.ylabel("Cantidad de viajes")
plt.title("Dispersión de la cantidad de viajes en TREN por día, según tipo de día")
plt.xticks(rotation=45)
plt.grid(True, linestyle='--', alpha=0.5)
plt.legend(title="Tipo de Día")

plt.tight_layout()
plt.savefig("graficos/dispersion_tren_por_dia.png", dpi=300)

print("✓ Gráfico TREN guardado como 'dispersión_tren_por_día.png'")

#-----SÉPTIMO GRÁFICO-----

# Filtrar COLECTIVO y agrupar por día y tipo de día
colectivo_diario = (
    df[df["TIPO_TRANSPORTE"] == "COLECTIVO"]
    .groupby(["DIA_TRANSPORTE", "TIPO_DIA"])["CANTIDAD"]
    .sum()
    .reset_index()
    .sort_values("DIA_TRANSPORTE")
)

# Convertir a datetime
colectivo_diario["DIA_TRANSPORTE"] =
pd.to_datetime(colectivo_diario["DIA_TRANSPORTE"])

# Gráfico
plt.figure(figsize=(12, 6))
sns.scatterplot(
    data=colectivo_diario,
    x="DIA_TRANSPORTE",
    y="CANTIDAD",
    hue="TIPO_DIA",
    palette=colores,
    alpha=0.7
)

```

```
plt.xlabel("Fecha")
plt.ylabel("Cantidad de viajes")
plt.title("Dispersión de la cantidad de viajes en COLECTIVO por día, según
tipo de día")
plt.xticks(rotation=45)
plt.grid(True, linestyle='--', alpha=0.5)
plt.legend(title="Tipo de Día")

plt.tight_layout()
plt.savefig("graficos/dispersion_colectivo_por_dia.png", dpi=300)

print("✓ Gráfico COLECTIVO guardado como 'dispersion_colectivo_por_día.png'")

print("Hemos finalizado el proceso de comparación")
```

## 11. Anexo: Resultados de ejecución de scripts

### 11.1. Anexo: `api_tipo_dias2024.py`

PS C:\Users\Yesica\Dropbox\CAD\TP CAD grupo04> &  
C:/Users/Yesica/AppData/Local/Programs/Python/Python313/python.exe  
"c:/Users/Yesica/Dropbox/CAD/TP CAD grupo04/api\_tipo\_dias2024.py"  
✓ Proceso finalizado. Se han agregado las columnas "DIA\_SEMANA" y "TIPO\_DIA" (desde API) al Dataset.  
📁 Archivo guardado como: df-sube-2024-tipo-dia.csv

### 11.2. Anexo: `api_tipo_dias2025.py`

PS C:\Users\Yesica\Dropbox\CAD\TP CAD grupo04> &  
C:/Users/Yesica/AppData/Local/Programs/Python/Python313/python.exe  
"c:/Users/Yesica/Dropbox/CAD/TP CAD grupo04/api\_tipo\_dias2025.py"  
✓ Proceso finalizado. Se han agregado las columnas "DIA\_SEMANA" y "TIPO\_DIA" (desde API) al Dataset.  
📁 Archivo guardado como: df-sube-2025-tipo-dia.csv

### 11.3. Anexo: `scraping_consulta_robots.py`

PS C:\Users\Yesica\Dropbox\CAD\TP CAD grupo04> &  
C:/Users/Yesica/AppData/Local/Programs/Python/Python313/python.exe  
"c:/Users/Yesica/Dropbox/CAD/TP CAD grupo04/scraping\_consulta\_robots.py"  
# Robots.txt (archivo)

User-agent: TestBot  
Disallow: /

User-agent: \*  
Disallow: /sinbarreras  
Disallow: /norefresh  
Disallow: /\*undefined\*  
Disallow: /debugsa  
Disallow: /debugsb  
Disallow: /preview  
Disallow: /herramientas  
Disallow: /Suscripcion  
Disallow: /NewslettersV1  
Disallow: /newsletters/  
Disallow: /1/SuscripcionV1  
Disallow: /registracion  
Disallow: /133919216/lanacion  
Disallow: /\*?utm\_\*  
Disallow: /weblogs/\*  
Disallow: /wsj/\*  
Disallow: \*/meteringamp  
Disallow: /buscador  
Disallow: /pf/api/v3/content/fetch/viafouraSource  
Disallow: /pf/api/v3/content/fetch/liftigniterSource  
Disallow: /opta-embed/  
Disallow: /widgets/  
Disallow: /pf/resources/  
Disallow: /cartelera-de-cine/Element  
Allow: /tema/\*  
Allow: /autor/\*

Sitemap: <https://www.lanacion.com.ar/sitemap-index.xml>  
Sitemap: <https://www.lanacion.com.ar/sitemap-news.xml>  
Sitemap: <https://www.lanacion.com.ar/sitemap-videos-jw.xml>  
Sitemap: <https://www.lanacion.com.ar/sitemap-index-historico.xml>  
Sitemap: <https://www.lanacion.com.ar/sitemap-articles-evergreen.xml>  
Sitemap: <https://www.lanacion.com.ar/sitemap-ampstories.xml>

Disallow: /el-mundo/un-argentino-fue-detenido-por-narcotrafico-en-indonesia-nid511832  
Disallow: /sociedad/acusan-a-un-argentino-de-narcotrafico-en-indonesia-nid512026  
Disallow: /politica/murio-el-senador-nacional-carlos-alberto-reutemann-nid06072021/  
Disallow: /economia/negocios/balance-nid04112021/

#### 11.4. Anexo: `scraping_feriados_lanacion2024.py`

```
PS C:\Users\Yesica\Dropbox\CAD\TP CAD grupo04> &
C:/Users/Yesica/AppData/Local/Programs/Python/Python313/python.exe
"c:/Users/Yesica/Dropbox/CAD/TP CAD grupo04/scraping.feriado2024.py"
Feriados encontrados:
2024-01-01 → AÑO NUEVO
2024-02-12 → CARNAVAL
2024-02-13 → CARNAVAL
2024-03-24 → DÍA NACIONAL DE LA MEMORIA POR LA VERDAD Y LA JUSTICIA
2024-03-29 → VIERNES SANTO
2024-04-01 → FERIADO PUENTE TURÍSTICO
2024-04-02 → DÍA DEL VETERANO Y DE LOS CAÍDOS EN LA GUERRA DE MALVINAS
2024-05-01 → DÍA DEL TRABAJADOR
2024-05-25 → DÍA DE LA REVOLUCIÓN DE MAYO
2024-06-17 → PASO A LA INMORTALIDAD DEL GENERAL MARTÍN GÜEMES
2024-06-20 → PASO A LA INMORTALIDAD DEL GENERAL MANUEL BELGRANO
2024-06-21 → FERIADO PUENTE TURÍSTICO
2024-07-09 → DÍA DE LA INDEPENDENCIA
2024-08-17 → PASO A LA INMORTALIDAD DEL GRAL. JOSÉ DE SAN MARTÍN
2024-10-11 → FERIADO PUENTE TURÍSTICO
2024-10-12 → DÍA DEL RESPETO A LA DIVERSIDAD CULTURAL
2024-11-18 → DÍA DE LA SOBERANÍA NACIONAL
2024-12-08 → DÍA DE LA INMACULADA CONCEPCIÓN DE MARÍA
2024-12-25 → NAVIDAD
✓ Scraping finalizado. Archivo guardado como df-sube-2024.csv
✓ Total de feriados encontrados: 19
```

#### 11.5. Anexo: `scraping_feriados_lanacion2025.py`

```
PS C:\Users\Yesica\Dropbox\CAD\TP CAD grupo04> &
C:/Users/Yesica/AppData/Local/Programs/Python/Python313/python.exe
"c:/Users/Yesica/Dropbox/CAD/TP CAD grupo04/scraping.feriado2025.py"
Feriados encontrados:
2025-01-01 → AÑO NUEVO
2025-03-03 → CARNAVAL
2025-03-04 → CARNAVAL
2025-03-24 → DÍA NACIONAL DE LA MEMORIA POR LA VERDAD Y LA JUSTICIA
2025-04-02 → DÍA DEL VETERANO Y DE LOS CAÍDOS EN LA GUERRA DE MALVINAS
2025-04-18 → VIERNES SANTO
2025-05-01 → DÍA DEL TRABAJADOR
2025-05-02 → PUENTE TURÍSTICO NO LABORABLE
2025-05-25 → DÍA DE LA REVOLUCIÓN DE MAYO
```

2025-06-16 → PASO A LA INMORTALIDAD DEL GENERAL MARTÍN GÜEMES  
2025-06-20 → PASO A LA INMORTALIDAD DEL GENERAL MANUEL BELGRANO  
2025-07-09 → DÍA DE LA INDEPENDENCIA  
2025-08-15 → PUENTE TURÍSTICO NO LABORABLE  
2025-08-17 → PASO A LA INMORTALIDAD DEL GRAL. JOSÉ DE SAN MARTÍN  
2025-10-12 → DÍA DEL RESPETO A LA DIVERSIDAD CULTURAL  
2025-11-21 → PUENTE TURÍSTICO NO LABORABLE  
2025-11-24 → DÍA DE LA SOBERANÍA NACIONAL  
2025-12-08 → DÍA DE LA INMACULADA CONCEPCIÓN DE MARÍA  
2025-12-25 → NAVIDAD

✓ Scraping finalizado. Archivo guardado como df-sube-2025.csv  
✓ Total de feriados encontrados: 19

11.6. Anexo: `eda_sube2024.py`

PS C:\Users\Yesica\Dropbox\CAD\TP CAD grupo04> &  
C:/Users/Yesica/AppData/Local/Programs/Python/Python313/python.exe  
"c:/Users/Yesica/Dropbox/CAD/TP CAD grupo04/eda\_sube2024.py"

DIMENSIONES DEL DATASET SUBE 2024

(504676, 13)

INFORMACION GENERAL DEL DATASET

```
<class 'pandas.core.frame.DataFrame'>  
RangeIndex: 504676 entries, 0 to 504675  
Data columns (total 13 columns):  
#   Column              Non-Null Count  Dtype  
---  ---  
0   DIA_TRANSPORTE      504676 non-null  datetime64[ns]  
1   NOMBRE_EMPRESA      504676 non-null  object  
2   LINEA               504676 non-null  object  
3   AMBA               504676 non-null  object  
4   TIPO_TRANSPORTE     504676 non-null  object  
5   JURISDICCION        502154 non-null  object  
6   PROVINCIA           502139 non-null  object  
7   MUNICIPIO           502139 non-null  object  
8   CANTIDAD            504676 non-null  int64  
9   DATO_PRELIMINAR     504676 non-null  object  
10  DIA_SEMANA          504676 non-null  object  
11  TIPO_DIA            504676 non-null  object  
12  MOTIVO_FERIADO       504676 non-null  object  
dtypes: datetime64[ns](1), int64(1), object(11)  
memory usage: 50.1+ MB
```

RANGO DE FECHAS DE 'DIA\_TRANSPORTE'



Rango de fechas: 2024-01-01 00:00:00 a 2024-12-31 00:00:00

ESTADÍSTICAS DESCRIPTIVAS (NUMÉRICAS)


CANTIDAD	
count	504676.00
mean	8194.59
std	18748.33
min	-105.00
25%	545.00
50%	2127.00
75%	7617.00
max	516002.00

VERIFICACIÓN DE DUPLICADOS

Duplicados: 0

Cantidad de filas con valores negativos en CANTIDAD: 3

	DIA_TRANSPORTE	TIPO_TRANSPORTE	CANTIDAD
65337	2024-02-20	TREN	-3
84741	2024-03-05	TREN	-105
147242	2024-04-20	TREN	-1

 Se encontraron 3 valores anómalos en CANTIDAD (valores negativos). Estos se consideran errores o correcciones no documentadas.

 Filas con valores negativos en 'CANTIDAD' eliminadas correctamente.

ESTADÍSTICAS ACTUALIZADAS

Resumen estadístico de la columna CANTIDAD:

count	504673.00
mean	8194.63
std	18748.37
min	1.00
25%	546.00
50%	2127.00
75%	7617.00
max	516002.00
Name: CANTIDAD, dtype: float64	

COLUMNAS CON DESVIACION ESTANDAR = 0

Ninguna columna numérica es constante.

VALORES FALTANTES POR COLUMNA

DIA_TRANSPORTE	0
----------------	---

```

NOMBRE_EMPRESA    0
LINEA              0
AMBA              0
TIPO_TRANSPORTE   0
JURISDICCION      2522
PROVINCIA          2537
MUNICIPIO          2537
CANTIDAD           0
DATO_PRELIMINAR   0
DIA_SEMANA         0
TIPO_DIA           0
MOTIVO_FERIADO     0
CANTIDAD_LOG       0
dtype: int64

```

----- ANÁLISIS DE VALORES NULOS -----

Tipos de transporte con JURISDICCION nula:

```

TIPO_TRANSPORTE
SUBTE          2522
Name: count, dtype: int64

```

Tipos de transporte con PROVINCIA nula:

```

TIPO_TRANSPORTE
SUBTE          2522
COLECTIVO      11
TREN            4
Name: count, dtype: int64

```

Tipos de transporte con MUNICIPIO nulo:

```

TIPO_TRANSPORTE
SUBTE          2522
COLECTIVO      11
TREN            4
Name: count, dtype: int64

```

----- Verificación de valores nulos tras corrección en SUBTE -----

```

JURISDICCION    0
PROVINCIA       15
MUNICIPIO       15
dtype: int64

```

Valores Nulos por Provincia:

TIPO_TRANSPORTE	NOMBRE_EMPRESA	LINEA	AMBA
COLECTIVO	EMPRESA 9 DE JULIO SRL	LINEA_500I_SFE	NO
COLECTIVO	EMPRESA 9 DE JULIO SRL	LINEA_500I_SFE	NO
COLECTIVO	EMPRESA 9 DE JULIO SRL	LINEA_500I_SFE	NO
COLECTIVO	EMPRESA 9 DE JULIO SRL	LINEA_500I_SFE	NO
COLECTIVO	EMPRESA 9 DE JULIO SRL	LINEA_500I_SFE	NO
COLECTIVO	EMPRESA 9 DE JULIO SRL	LINEA_500I_SFE	NO

```

COLECTIVO EMPRESA 9 DE JULIO SRL LINEA_500I_SFE NO
COLECTIVO EMPRESA 9 DE JULIO SRL LINEA_500I_SFE NO
COLECTIVO EMPRESA 9 DE JULIO SRL LINEA_500I_SFE NO
COLECTIVO EMPRESA 9 DE JULIO SRL LINEA_500I_SFE NO
COLECTIVO EMPRESA 9 DE JULIO SRL LINEA_500I_SFE NO
TREN SOFSE - TREN DEL VALLE FFCC TREN DEL VALLE NO
TREN SOFSE - TREN DEL VALLE FFCC TREN DEL VALLE NO
TREN SOFSE - TREN DEL VALLE FFCC TREN DEL VALLE NO
TREN SOFSE - TREN DEL VALLE FFCC TREN DEL VALLE NO

```

Valores Nulos por Municipio:

```

TIPO_TRANSPORTE      NOMBRE_EMPRESA      LINEA AMBA
COLECTIVO EMPRESA 9 DE JULIO SRL LINEA_500I_SFE NO
COLECTIVO EMPRESA 9 DE JULIO SRL LINEA_500I_SFE NO
COLECTIVO EMPRESA 9 DE JULIO SRL LINEA_500I_SFE NO
COLECTIVO EMPRESA 9 DE JULIO SRL LINEA_500I_SFE NO
COLECTIVO EMPRESA 9 DE JULIO SRL LINEA_500I_SFE NO
COLECTIVO EMPRESA 9 DE JULIO SRL LINEA_500I_SFE NO
COLECTIVO EMPRESA 9 DE JULIO SRL LINEA_500I_SFE NO
COLECTIVO EMPRESA 9 DE JULIO SRL LINEA_500I_SFE NO
COLECTIVO EMPRESA 9 DE JULIO SRL LINEA_500I_SFE NO
COLECTIVO EMPRESA 9 DE JULIO SRL LINEA_500I_SFE NO
COLECTIVO EMPRESA 9 DE JULIO SRL LINEA_500I_SFE NO
COLECTIVO EMPRESA 9 DE JULIO SRL LINEA_500I_SFE NO
TREN SOFSE - TREN DEL VALLE FFCC TREN DEL VALLE NO
TREN SOFSE - TREN DEL VALLE FFCC TREN DEL VALLE NO
TREN SOFSE - TREN DEL VALLE FFCC TREN DEL VALLE NO
TREN SOFSE - TREN DEL VALLE FFCC TREN DEL VALLE NO

```

----- Verificación final tras imputación específica: -----

```

JURISDICCION 0
PROVINCIA 0
MUNICIPIO 0
dtype: int64

```

✓ ☐ Filas con valores nulos corregidas correctamente.

----- Estadísticas descriptivas -----

```

count      mean      std min   25%   50%   75%   max
AMBA
NO  354058.0  2438.714389  3324.118086  1.0  350.0  1178.0  3294.0  34994.0
SI   150615.0  21725.356638  29847.271595  1.0  5203.0  14261.0  27951.5  516002.0

```

----- Identificación de outliers por AMBA (sin eliminar) -----

AMBA = NO

Q1: 350.0, Q3: 3294.0, IQR: 2944.0

Rango normal: [0.0, 7710.0]

Total filas: 354058, Outliers detectados: 23965

AMBA = SI

Q1: 5203.0, Q3: 27951.5, IQR: 22748.5

Rango normal: [0.0, 62074.2]

Total filas: 150615, Outliers detectados: 8112

----- Análisis de outliers -----

⚠ Los valores considerados outliers podrían corresponder a situaciones reales (eventos masivos, paros, problemas técnicos), por lo que se optó por mantenerlos. En lugar de eliminarlos, se los identificó y analizó por separado para entender su impacto.

✓ El análisis EDA se completo correctamente.

✓ El análisis fue realizado sin excluir registros del dataset principal.

📁 Se generaron los siguientes archivos:

- sube2024\_boxplot\_amba.png
- sube2024\_boxplot\_cantidad\_por\_tipo.png
- sube2024\_cantidad\_viajes\_por\_feriado.png
- sube2024\_evolucion\_mensual.png
- sube2024\_feriados\_vs\_habiles\_por\_provincia.png
- sube2024\_heatmap\_dia\_semana\_tipo\_transporte.png
- sube2024\_histograma\_cantidad.png
- sube2024\_histograma\_cantidad\_log.png
- sube2024\_modal\_por\_tipo\_dia.png
- sube2024\_promedio\_viajes\_tipo\_dia.png
- sube2024\_viajes\_habil\_vs\_no\_habil.png
- sube2024\_viajes\_por\_dia\_semana.png
- sube2024\_viajes\_por\_mes.png
- sube2024\_viajes\_por\_tipo\_transporte.png

✓ Datos limpios y procesados son guardados en 'dat\_sube2024\_eda.csv'

Proceso terminado.

## 11.7. Anexo: `eda_sube2025.py`

```
PS C:\Users\Yesica\Dropbox\CAD\TP CAD grupo04> &
C:/Users/Yesica/AppData/Local/Programs/Python/Python313/python.exe
"c:/Users/Yesica/Dropbox/CAD/TP CAD grupo04/eda_sube2025.py"
```

---

### DIMENSIONES DEL DATASET SUBE 2025

(205633, 13)

---

### INFORMACION GENERAL DEL DATASET

<class 'pandas.core.frame.DataFrame'>

RangeIndex: 205633 entries, 0 to 205632

Data columns (total 13 columns):

#	Column	Non-Null Count	Dtype
---	--------	----------------	-------

--- -----

0	DIA_TRANSPORTE	205633 non-null	datetime64[ns]
---	----------------	-----------------	----------------

---

```
1 NOMBRE_EMPRESA      205633 non-null object
2 LINEA                205633 non-null object
3 AMBA                 205633 non-null object
4 TIPO_TRANSPORTE      205633 non-null object
5 JURISDICCION         204612 non-null object
6 PROVINCIA            204612 non-null object
7 MUNICIPIO            204612 non-null object
8 CANTIDAD              205633 non-null int64
9 DATO_PRELIMINAR      205633 non-null object
10 DIA_SEMANA           205633 non-null object
11 TIPO_DIA             205633 non-null object
12 MOTIVO_FERIADO       205633 non-null object
dtypes: datetime64[ns](1), int64(1), object(11)
memory usage: 20.4+ MB
```

---

RANGO DE FECHAS DE 'DIA\_TRANSPORTE'

Rango de fechas: 2025-01-01 00:00:00 a 2025-05-29 00:00:00

---

ESTADÍSTICAS DESCRIPTIVAS (NUMÉRICAS)

```
      CANTIDAD
count  205633.00
mean   7247.62
std    16915.45
min     -1.00
25%    458.00
50%   1817.00
75%   6567.00
max   467163.00
```


---

VERIFICACIÓN DE DUPLICADOS

Duplicados: 0

Cantidad de filas con valores negativos en CANTIDAD: 1

	DIA_TRANSPORTE	TIPO_TRANSPORTE	CANTIDAD
170609	2025-05-03	TREN	-1

 ☐ Se encontró un valor anómalo en CANTIDAD (valor negativo). Estos se considera un error o corrección no documentada.

 ☒ Filas con valores negativos en 'CANTIDAD' eliminadas correctamente.

---

ESTADÍSTICAS ACTUALIZADAS

---

Resumen estadístico de la columna CANTIDAD:

```
count    205632.00
mean     7247.65
std      16915.48
```

---

```

min          1.00
25%          458.00
50%          1817.00
75%          6567.25
max          467163.00
Name: CANTIDAD, dtype: float64

```

COLUMNAS CON DESVIACION ESTANDAR = 0

Ninguna columna numérica es constante.

VALORES FALTANTES POR COLUMNA

```

DIA_TRANSPORTE    0
NOMBRE_EMPRESA    0
LINEA              0
AMBA              0
TIPO_TRANSPORTE    0
JURISDICCION      1021
PROVINCIA          1021
MUNICIPIO          1021
CANTIDAD           0
DATO_PRELIMINAR    0
DIA_SEMANA         0
TIPO_DIA           0
MOTIVO_FERIADO     0
CANTIDAD_LOG       0
dtype: int64

```

----- ANÁLISIS DE VALORES NULOS -----

Tipos de transporte con JURISDICCION nula:

TIPO\_TRANSPORTE

SUBTE 1021

Name: count, dtype: int64

Tipos de transporte con PROVINCIA nula:

TIPO\_TRANSPORTE

SUBTE 1021

Name: count, dtype: int64

Tipos de transporte con MUNICIPIO nulo:

TIPO\_TRANSPORTE

SUBTE 1021

Name: count, dtype: int64

----- Verificación de valores nulos tras corrección en SUBTE -----

```

JURISDICCION  0
PROVINCIA     0
MUNICIPIO     0

```

dtype: int64

Valores únicos de Jurisdicción

['MUNICIPAL', 'PROVINCIAL', 'NACIONAL', 'CABA', 'C.A.B.A']

Valores únicos de Provincia

['BUENOS AIRES', 'JN', 'CIUDAD AUTÓNOMA DE BUENOS AIRES', 'CHUBUT', 'ENTRE RÍOS', 'LA PAMPA', 'TUCUMAN', 'MENDOZA', 'SANTA FE', 'RÍO NEGRO', 'FORMOSA', 'SANTA CRUZ', 'TIERRA DEL FUEGO', 'JUJUY', 'NEUQUÉN', 'CHACO', 'CORRIENTES', 'CATAMARCA', 'SAN LUIS', 'CORDOBA', 'SANTIAGO DEL ESTERO', 'SAN JUAN', 'C.A.B.A']

Valores únicos de Municipio

['MERCEDES', 'GENERAL PUEYRREDON', 'SN', 'LANUS', 'SD', 'GENERAL RODRIGUEZ', 'BRANDSEN', 'LUJAN', 'MERLO', 'FLORENCIO VARELA', 'PILAR', 'ESTEBAN ECHEVERRIA', 'LOBOS', 'MORENO', 'EXALTACION DE LA CRUZ', 'MALVINAS ARGENTINAS', 'LA PLATA', 'SAN VICENTE', 'ESCOBAR', 'ALMIRANTE BROWN', 'EZEIZA', 'LOMAS DE ZAMORA', 'AVELLANEDA', 'QUILMES', 'BERAZATEGUI', 'LA MATANZA', 'MORON', 'GENERAL SAN MARTIN', 'SAN ISIDRO', 'SAN FERNANDO', 'TIGRE', 'SAN MIGUEL', 'JOSE C. PAZ', 'URBANO DE LA COSTA', 'CABA', 'PINAMAR', 'TORNQUIST', 'CONCORDIA', 'SANTA ROSA', 'SAN MIGUEL DE TUCUMAN', 'CIPOLLETTI', 'TANDIL', 'AZUL', 'VILLA GESELL', 'SAN PEDRO', 'FORMOSA', 'RIO GALLEGOS', 'RIO GRANDE', 'JUNIN', 'ZARATE', 'TRELEW', 'COMODORO RIVADAVIA', 'SAN SALVADOR DE JUJUY', 'SAN MARTIN DE LOS ANDES', 'PARANA', 'SANTA FE', 'SAN NICOLAS DE LOS ARROYOS', 'SAN CARLOS DE BARILOCHE', 'PLOTTIER', 'CORRIENTES', 'ROSARIO', 'NEUQUEN', 'RAWSON', 'BAHIA BLANCA', 'NECOCHEA', 'PRESIDENTE PERON', 'CORONEL ROSALES', 'SAN LUIS', 'USHUAIA', 'RECONQUISTA', 'RIO CUARTO', 'PRESIDENCIA ROQUE SAENZ PEÑA', 'GENERAL ROCA', 'CAMPANA', 'CAÑUELAS', 'ESQUEL', 'VILLA ALLENDE', 'GUALEGUAYCHU', 'RAFAELA', 'LA BANDA', 'VILLA MARIA', 'CONCEPCION DEL URUGUAY', 'OLAVARRIA', 'CHIVILCOY', 'PERGAMINO', 'GENERAL PICO', 'VENADO TUERTO', 'PALPALA', 'BALCARCE', 'ITUZAINGO', 'VIEDMA', 'PUERTO GENERAL SAN MARTIN', 'PUNTA INDIO', 'CORDOBA']

Valores únicos de Jurisdicción

['MUNICIPAL', 'PROVINCIAL', 'NACIONAL', 'CABA']

Valores únicos de Provincia

['BUENOS AIRES', 'JN', 'CIUDAD AUTÓNOMA DE BUENOS AIRES', 'CHUBUT', 'ENTRE RÍOS', 'LA PAMPA', 'TUCUMAN', 'MENDOZA', 'SANTA FE', 'RÍO NEGRO', 'FORMOSA', 'SANTA CRUZ', 'TIERRA DEL FUEGO', 'JUJUY', 'NEUQUÉN', 'CHACO', 'CORRIENTES', 'CATAMARCA', 'SAN LUIS', 'CORDOBA', 'SANTIAGO DEL ESTERO', 'SAN JUAN']

✓ Archivo 'df-sube-2025' guardado con los nulos corregidos y otras correcciones.

✓ Se realiza limpieza y correccion del dataframe para una correcta comparacion con el dataframe del año 2024.

Proceso terminado.

## 11.8. Anexo: `comparativa_2025vs2024.py`

```
PS C:\Users\flopy\Documents\UNLaM_esp cs datos\Tp_CAD> &  
C:/Users/flopy/AppData/Local/Programs/Python/Python313/python.exe  
"c:/Users/flopy/Documents/UNLaM_esp cs datos/Tp_CAD/Comparativa_2025vs2024.py"
```

✓ Proceso finalizado. Se han unido los datasets de 2024 y 2025.

📁 Archivo guardado como: dat-subte-2024-2025-combinado.csv

Gráfico guardado como 'cantidad\_de\_viajes\_acumulado\_enero-mayo2025vs2024.png'

✓ Gráfico combinado guardado como 'barras\_linea\_variacion\_subte\_colores.png'

✓ Gráfico combinado MEJORADO guardado como 'barras\_linea\_variacion\_tren\_colores.png'

✓ Gráfico combinado guardado como 'barras\_linea\_variacion\_colectivo\_colores.png'

✓ Gráfico SUBTE guardado como 'dispersión\_subte\_por\_día.png'

✓ Gráfico TREN guardado como 'dispersión\_tren\_por\_día.png'

✓ Gráfico COLECTIVO guardado como 'dispersión\_colectivo\_por\_día.png'

Hemos finalizado el proceso de comparación