UNAM

Análisis de Movilidad Urbana CDMX-ECOBICI con SPARK

Abraham Buenrostro Cruces

Profesor: Dr. JOSÉ GUSTAVO FUENTES CABRERA

UNAM

1.-Introducción.

La movilidad urbana es un desafío complejo en grandes metrópolis como la Ciudad de México. La congestión vehicular, la eficiencia del transporte público y la optimización de sistemas como las bicicletas impactan directamente la calidad de vida de los ciudadanos y la sostenibilidad ambiental. El análisis de datos juega un papel crucial en la comprensión de estos patrones de movilidad, permitiendo a planificadores urbanos y autoridades tomar decisiones informadas para mejorar la infraestructura y los servicios de transporte.

En esta práctica, se utiliza Spark como herramienta de procesamiento para analizar datos del sistema Ecobici de la Ciudad de México. Al aprovechar la capacidad de Spark para manejar grandes volúmenes de información de manera eficiente, se exploran aspectos como los patrones de uso de bicicletas, las estaciones más activas, los horarios pico y los principales usuarios de Ecobici. Este análisis permite identificar tendencias y oportunidades de mejora en el servicio de bicicletas compartidas, contribuyendo al desarrollo de soluciones de movilidad más sostenibles e inteligentes.

2.-Descripcion del Dataset.

Los datos que se ocuparon para realizar esta práctica fueron obtenidos de datos abiertos Ecobici, se recuperaron los datos del mes de marzo 2025.

Nombre del archivo: 2025-03.csv

Tamaño: 107 Mb

El dataset que se utilizo tiene 9 columnas * 1832349 registros

```
+---------------+
|Total_Registros|
+---------------+
|        1832349|
+---------------+
```

Información de los datos

```
root
 |-- Genero_Usuario: string (nullable = true)
 |-- Edad_Usuario: string (nullable = true)
 |-- Bici: integer (nullable = true)
 |-- Ciclo_Estacion_Retiro: string (nullable = true)
 |-- Fecha_Retiro: string (nullable = true)
 |-- Hora_Retiro: timestamp (nullable = true)
 |-- Ciclo_EstacionArribo: string (nullable = true)
 |-- Fecha_Arribo: string (nullable = true)
 |-- Hora_Arribo: timestamp (nullable = true)
```

Genero_Usuario (string): Género de los usuarios.

Edad_Usuario (String): Edad del usuario.

Bici (integer) el id de la bicicleta que se utilizó.
Ciclo_Estacion_Retiro (string): El id de la estación donde se solicito la ecobici.
Fecha_Retiro (String): El día en formato día/mes/año en que se retiro la ecobici de la estación.
Hora_Retiro (timestamp): La hora en que se retiró la bici de la estación.

Ciclo_EstacionArribo (String): El id de la estación en la que se entrego la ecobici.

Fecha_Arribo (String): El día en formato día/mes/año en que se llegó la ecobici a la nueva estación.

Hora_Arribo (timestamp) La hora en que arribo la bici a la estación.

Fuente: https://ecobici.cdmx.gob.mx/datos-abiertos/

3.-Explicacion de operaciones realizadas.

Carga y limpieza de datos

- Se leyó el archivo CSV desde una ruta local utilizando spark.read.csv() con header=True y inferSchema=True para identificar automáticamente los tipos de datos.
- Se eliminaron registros con fechas inconsistentes en Fecha_Retiro, manteniendo solo aquellos pertenecientes al mes 03/2025.
- Se eliminaron los registros con valores nulos en Edad_Usuario, ya que representaban una fracción menor del total (98 nulos).

Transformaciones

- Se registró el DataFrame como vista temporal (createOrReplaceTempView("Ecobicis")) para poder ejecutar consultas SQL.
- Se extrajeron componentes de fecha y hora (día, hora) usando funciones como hour() y group by, para analizar la afluencia horaria.
- Se agruparon edades por rangos de 10 años (buckets) para facilitar la visualización y comparación de uso por grupos de edad.
- Se realizaron ordenamientos (ORDER BY) para identificar los elementos más y menos frecuentes.

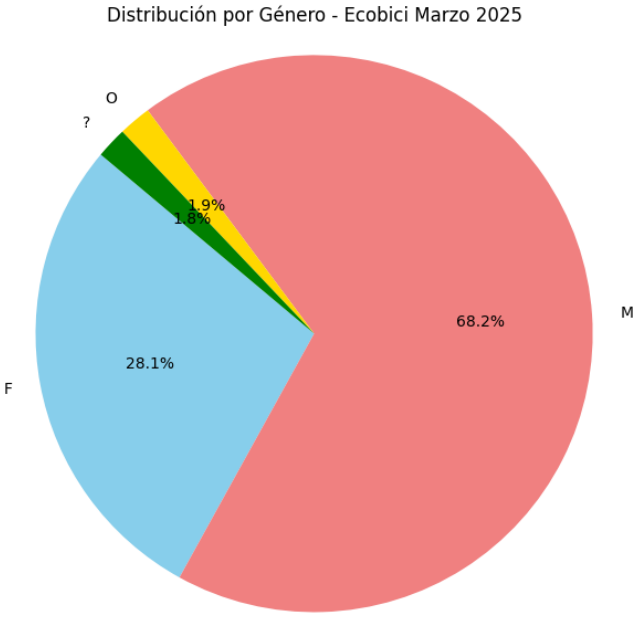
Acciones

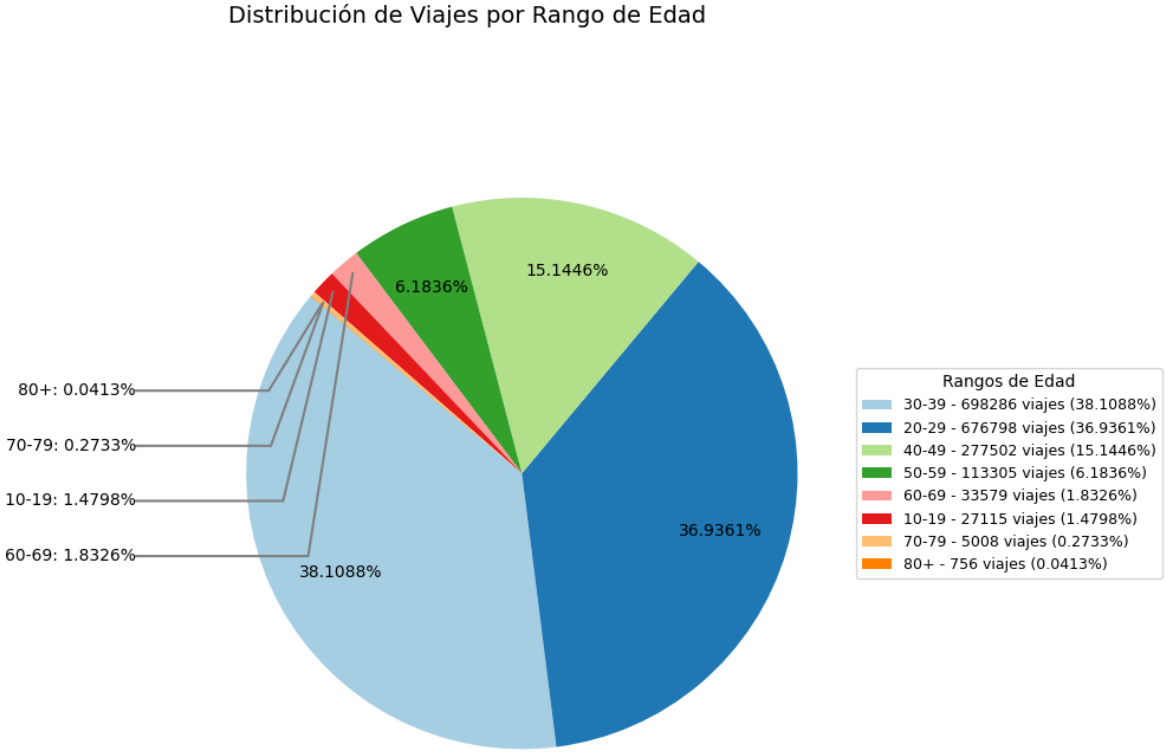
- Se realizaron consultas como df.count(), df.show(), y agregaciones con groupBy().count() para obtener valores y validar operaciones.
- Se exportaron resultados relevantes a CSV mediante DataFrame.write.csv().

4.- Análisis de los resultados

- Género: La mayoría de los usuarios fueron hombres, seguidos por mujeres, y un pequeño grupo sin especificar.

- Edad: Los grupos con mayor uso se concentraron entre los 20 y 40 años. Los rangos extremos (menores de 20 y mayores de 60) representaron una proporción significativamente menor.



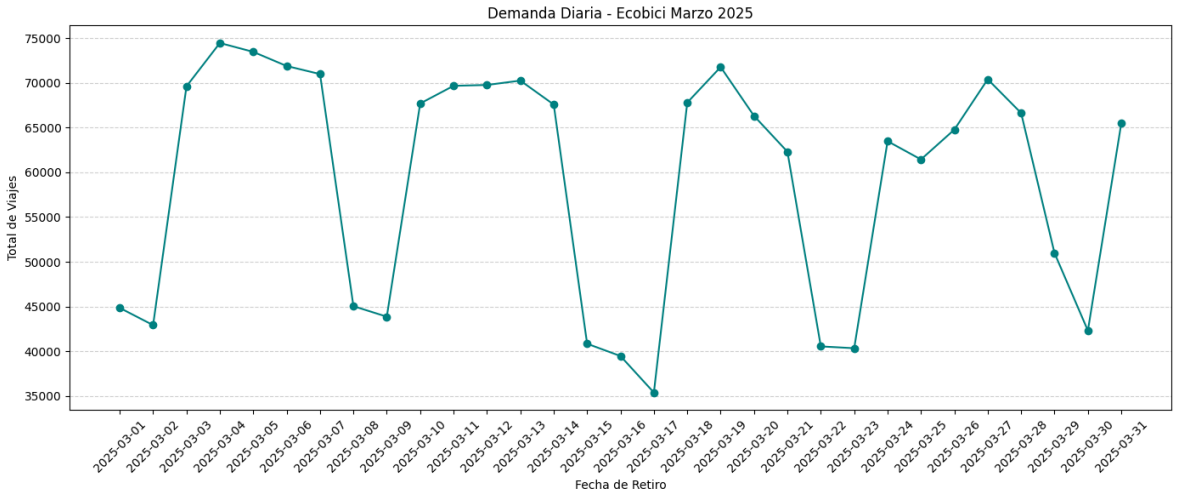


- Estaciones de retiro: Algunas estaciones mostraron alta concentración de viajes, lo que indica posibles zonas de alta demanda como centros laborales o de transporte.

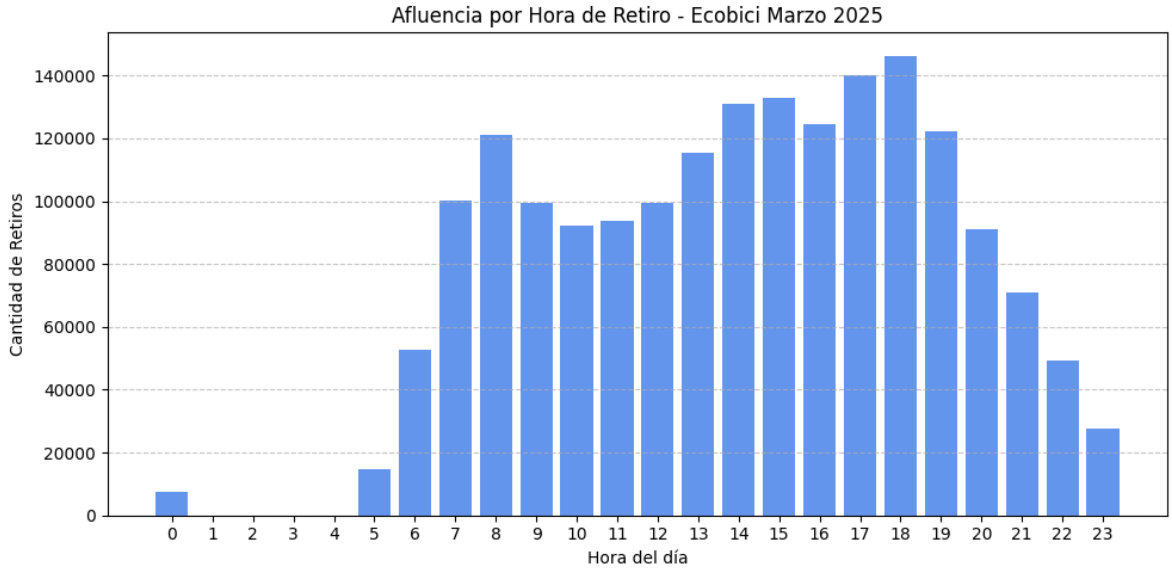
Ciclo_Estacion_Retiro	Total_Viajes
271-272	12521
548	10734
027	10715
107-108	10479
064	10037
237-238	9402
018	8872
273-274	8641
014	8637
192-193	8625

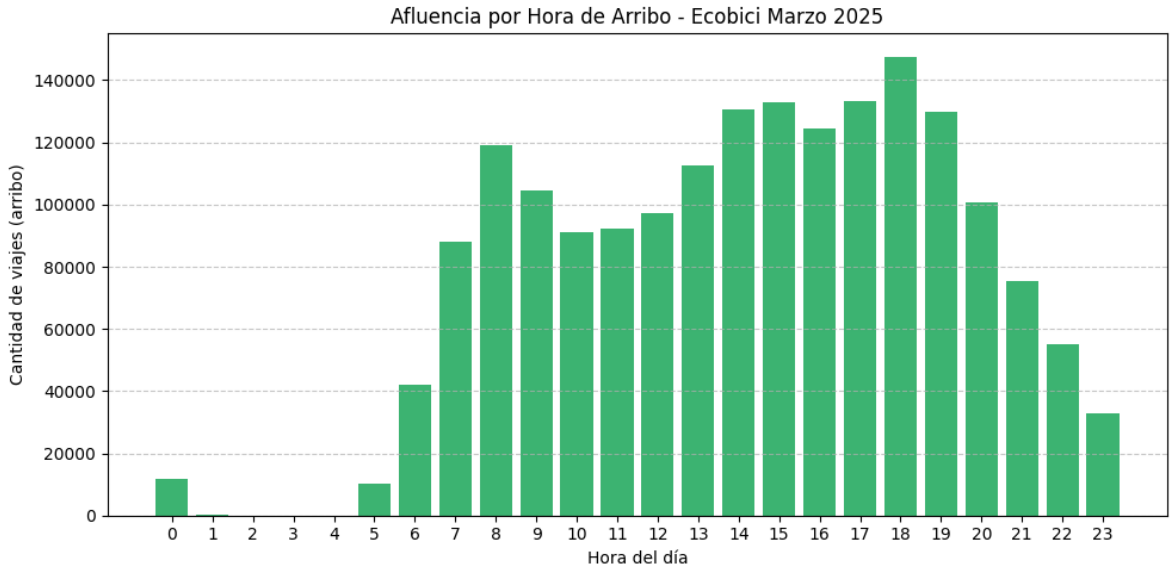
only showing top 10 rows

- Días de mayor uso: Se identificaron fechas con picos de demanda, lo que puede estar relacionado con eventos, clima o comportamiento laboral de los usuarios.



- Afluencia horaria: Las horas pico se ubicaron en la mañana (7-9 h) y tarde (17-20 h), coincidiendo con horarios de entrada y salida laboral.





5.-Fragmentos clave de código con comentarios

a) Cargar los datos

```
# Cargar el archivo CSV en un DataFrame
df = spark.read.csv("2025-03.csv", header=True, inferSchema=True)
# Mostrar los primeros
df.show(10)
# Mostrar el esquema del DataFrame
df.printSchema()
```

✓ 3.6s

b) Filtrar datos válidos del mes de marzo

```
df = df.filter(df.Fecha_Retiro.endswith("/03/2025"))
```
✓ 0.2s

c) Agrupar por hora de retiro

```
# Registrar el DataFrame como una tabla temporal
df.createOrReplaceTempView("Ecobicis")
# Realizar una consulta SQL para contar el número de viajes por estación de origen
df_estaciones_origen = spark.sql("""
    SELECT Ciclo_Estacion_Retiro, COUNT(*) AS Total_Viajes
    FROM Ecobicis
    GROUP BY Ciclo_Estacion_Retiro
    ORDER BY Total_Viajes DESC
""")

df_estaciones_origen.show(10)
```
✓ 16.2s

d) Agrupar edades por rango

```
# Consulta para agrupar por rangos de edad (por cada 10 años)
spark.sql("""
    SELECT
        CASE
            WHEN Edad_Usuario >= 0 AND Edad_Usuario <= 9 THEN '0-9'
            WHEN Edad_Usuario >= 10 AND Edad_Usuario <= 19 THEN '10-19'
            WHEN Edad_Usuario >= 20 AND Edad_Usuario <= 29 THEN '20-29'
            WHEN Edad_Usuario >= 30 AND Edad_Usuario <= 39 THEN '30-39'
            WHEN Edad_Usuario >= 40 AND Edad_Usuario <= 49 THEN '40-49'
            WHEN Edad_Usuario >= 50 AND Edad_Usuario <= 59 THEN '50-59'
            WHEN Edad_Usuario >= 60 AND Edad_Usuario <= 69 THEN '60-69'
            WHEN Edad_Usuario >= 70 AND Edad_Usuario <= 79 THEN '70-79'
            ELSE '80+'
        END AS Rango_Edad,
        COUNT(*) AS Total_Viajes
    FROM Ecobicis
    GROUP BY Rango_Edad
    ORDER BY Total_Viajes DESC
""").show()
```

6.- Documentación de IA

Durante el desarrollo de este análisis, se emplearon distintas herramientas de inteligencia artificial como asistentes auxiliares. A continuación, se detalla su uso:

6.1 Chatgpt

Se utilizó para:

- Asistencia técnica con PySpark y SQL.
- Generación y explicación de consultas para limpieza.
- Redacción estructurada de secciones del informe.

6.2 GitHub Copilot

Se empleó como asistente de autocompletado de código en Python dentro del entorno de desarrollo (VS). Su uso facilitó:

- Escritura más ágil de funciones auxiliares para agrupamiento y visualización.
- Estandarización de fragmentos repetitivos en la generación de operaciones.
- Verificación automática de sintaxis en scripts.

6.3 Google Gemini

Se utilizó al inicio del proyecto para investigación:

- Apoyar en la búsqueda inicial de referencias sobre movilidad urbana y el uso de datos abiertos de Ecobici CDMX.
- Sugerencias preliminares sobre cómo estructurar un análisis con Spark.
- Identificación de posibles variables relevantes para la exploración de afluencia por edad, género y fecha.

7.- Conclusión.

La práctica permitió aplicar conocimientos clave de análisis distribuido utilizando PySpark con datos de movilidad urbana. Se reforzo el uso de operaciones con limpieza, transformaciones y agrupaciones. Además, se comprobó la eficiencia de Spark para el procesamiento de grandes volúmenes de datos y se reforzó la importancia del análisis para la toma de decisiones basada en datos.