

# INFORME PRUEBA TÉCNICA: SISTEMA DE USO COMPARTIDO DE BICICLETAS, CIUDAD: LA.

Jacob Tinoco

ARKON DATA Prueba Técnica

## Contenido

Introducción.....	3
Estructura del Proyecto. ....	4
Descripción de la Estructura.....	5
0. Comprensión del problema.....	6
1. Análisis Exploratorio de datos.....	6
2. Visualización exploratoria de datos en el csv:.....	6
3. Preprocesamiento de datos. ....	7
4. Procesamiento de datos. ....	7
5. Análisis de datos. ....	7
5.1 Regresión Lineal. ....	7
5.2 Prophet.....	8
6. Resultados y Visualizaciones.....	8
Conclusiones.....	12
Recomendaciones .....	12
Anexos.....	13
Anexo 1: Script: visualización_general.py.....	13
Anexo 2: Script: histograma.py .....	14
Anexo 3: Script: estaciones_saturadas.py .....	15
Anexo 4: Script: horarios_saturados.py .....	16
Anexo 5: Script: modelo_LR.py.....	17
Anexo 6. Script modelo_prophet.py .....	18
Gráficos .....	19
Gráfico 1: Conteo de viajes al año a lo largo del tiempo dentro del data set. ....	19
Gráfico 2: Uso de regresión lineal para la predicción a 3 años. ....	19
Gráfico 3. Tendencias en el uso del servicio por: Año, mes, semana, y una predicción a 3 años en el uso del servicio. ....	20
Gráfico 4. Tendencia anual a lo largo del tiempo con predicción a 3 años. ....	21
Gráfico 5. Predicción de los siguientes 3 años, respecto a la información a lo largo del tiempo.....	21
Diagramas.....	22

Diagrama 1 Propuesta de producción del modelo analítico. ....	22
---	----

## Introducción.

La prueba técnica se centra en analizar datos anónimos del uso de un sistema compartido de bicicletas en Los Ángeles. La tabla proporcionada contiene un historial de viajes desde 2016 y se busca analizar la columna **Passholder\_type**.

En la Tabla 1, se describe el contenido del csv proporcionado para la prueba técnica de Arkon Data.

Columnas en csv	Descripción
<b>trip_id</b>	Identificador único para el viaje
<b>duration</b>	Duración del viaje en minutos
<b>start_time</b>	Día/hora donde el viaje inicia en formato ISO 8601 tiempo local
<b>end_time</b>	Día/hora donde el viaje termina en formato ISO 8601 tiempo local
<b>start_station</b>	La estación donde el viaje inició
<b>start_lat</b>	La latitud de la estación donde el viaje se originó
<b>start_lon</b>	La longitud de la estación donde el viaje se originó
<b>end_station</b>	La estación donde el viaje terminó
<b>end_lat</b>	La latitud de la estación donde terminó el viaje
<b>end_lon</b>	La longitud de la estación donde terminó el viaje
<b>bike_id</b>	Un entero único que identifica la bicicleta
<b>plan_duration</b>	Número de días que el usuario tendrá el pase. 0 significa un viaje único (Walk-up plan)
<b>trip_route_category</b>	“Round trip” son viajes que empiezan y terminan en la misma estación
<b>passholder_type</b>	El nombre del plan de passholder

*Tabla 1 Contenido de train\_set, csv proporcionado por Arkon Data*

# Estructura del Proyecto.

En la Figura 1, se ilustra como esta distribuida la estructura de este proyecto, mas adelante se describe brevemente cada sección.

```
Solución_prueba_tecnica_100225_V1.5.0_JT/
Data_science_data_analitics_example/
|
├─ data/
|   ├── train_set.csv
|   ├── test_set.csv
|   └─ sample_submission.csv
|
├─ data_processed/
|   ├── test_set_Processed_090225_V1.0.0_JT.csv
|   ├── train_set_Processed_070225_V1.1.0_JT.csv
|   └─ sample_submission.csv
|
├─ graphic/
|   ├── conteo_viajes_por_año.png
|   ├── distribucion_duracion_viajes.png
|   ├── distribucion_duracion_(año).png
|   ├── ...
|   ├── Modelo_analitico_Prophet_prediccion_3años.png
|   ├── Modelo_analitico_Prophet_Tendencias.png
|   ├── Modelo_analitico_Prophet_Tendencias_actuales.png
|   └─ Modelo_analitico_(LR).png
|
├─ notebook/
|   └─ notebook.ipynb
|
├─ src/
|   ├── estaciones_saturadas.py
|   ├── histograma.py
|   ├── horarios_saturados.py
|   ├── modelo_LR.py
|   ├── modelo_prophet.py
|   └─ visualizacion_general.py
|
├─ README.md
└─ requirements.txt
```

Figura 1 Estructura del proyecto.

## Descripción de la Estructura.

- **data/**: Contiene los datos originales utilizados en el análisis para el preprocesamiento.
- **data\_prosseced/**: Contiene los datos procesados.
- **grafic/**: Almacena las gráficas generadas a partir del análisis.
- **notebook/**: Incluye el Jupyter Notebooks con los códigos de prueba para el análisis exploratorio y los modelos analíticos.
- **notebook/grafic/**: Almacena las gráficas generadas a partir del análisis.
- **src/**: Contiene los scripts para visualizaciones del análisis exploratorio y modelos analíticos.
- **README.md**: Documento que proporciona una visión general del proyecto y sus objetivos.
- **requirements.txt**: Lista de dependencias necesarias para ejecutar el proyecto.

## 0. Comprensión del problema.

Para determinar la trayectoria del análisis primero se comprendió que se busca saber la saturación del servicio, y las tendencias para determinar los planes a futuro, previamente en el documento descriptivo .pdf de la prueba se especulaba una tendencia al alta en el uso del servicio, lo cual origino preguntas guías a responder:

- ¿Cuál es la tendencia de uso en el servicio a lo largo del tiempo?
- ¿Cuál es la saturación del servicio por estación, horarios, y días a lo largo del tiempo?
- ¿Qué datos pueden ayudar a responder las preguntas anteriores dentro del data set?
- Con los datos anteriores y conociendo el tipo de servicio, ¿Cuáles son los modelos predictivos adecuados para el análisis?
- ¿Cómo será la predicción de uso para el servicio a 3 años?

## 1. Análisis Exploratorio de datos.

Se realizó un análisis exploratorio de datos (EDA) para entender mejor la distribución y características del conjunto de datos:

## 2. Visualización exploratoria de datos en el csv:

- **Archivos de respaldo:** se creó una carpeta de respaldo para los datos originales nombrados como **train\_set\_original** y **test\_set\_original**
- **Verificación de datos:** limpieza de datos duplicados.
- **Identificación de datos relevantes:** **start\_time**, **end\_time**, **start\_sation**, **end\_satation**, **passholder\_type**
- **Identificación de formatos:** visualizar como era el formato en el **train\_set\_original**

Durante esta etapa no se encontraron datos duplicados, pero si se pudo identificar que el formato de fechas en **start\_time** y en **end\_time** eran inconsistentes.

### 3. Preprocesamiento de datos.

- **Formateo de datos:** Se formatearon los datos a la norma ISO 8601 y se creó un archivo nuevo llamado: **test\_set\_Preproceded\_090225\_V1.0.0\_JT**
- **Filtrado:** se filtraron los datos más relevantes en notas personales para su uso en el procesamiento y análisis.

### 4. Procesamiento de datos.

Para conocer la saturación del servicio Se generaron gráficos para visualizar:

- **Conteo de viajes a lo largo del tiempo:** Se utilizó un gráfico de líneas para mostrar la tendencia de viajes desde 2016 hasta 2021, pueden ver el código en Anexo 1.
- **Distribución de la duración de los viajes:** Se generó un histograma para visualizar la duración de los viajes, limitando a un rango de 0 a 60 minutos, pueden ver el código en Anexo 2.
- **Top 10 rutas más utilizadas:** Se identificaron las rutas más frecuentadas y se analizó la duración media de los viajes, pueden ver el código en Anexo 3.
- **Top 4 horarios de mayor saturación para las rutas más utilizadas:** Se identificaron los horarios más frecuentados y se analizó la duración media de los viajes, pueden ver el código en Anexo 4.

### 5. Análisis de datos.

En la fase previa se logró identificar una tendencia creciente en el uso del sistema desde 2016 a 2018, y justo después un pronunciamiento decreciente desde el 2019 al 2021, contrario a la especulación que teníamos al del proyecto, para analizar estas tendencias y comportamientos se compararon algunos modelos, ver Anexo 6, posteriormente se implementaron dos modelos analíticos regresión lineal, ver en Anexo 5, y uso de Prophet de Meta en Python, ver en Anexo 6.

#### 5.1 Regresión Lineal.

Se eligió la regresión lineal debido a su simplicidad y eficacia para predecir tendencias basadas en datos históricos. Este modelo permite comprender la relación entre variables, como el tiempo y el número de viajes, pero está limitado en su comprensión del entorno.

##### **Justificación:**

- **Interpretabilidad:** Los coeficientes del modelo son fáciles de interpretar, lo que ayuda a entender cómo cada variable impacta en la predicción.



- **Eficiencia:** Es más óptimo y usa menos recursos de hardware en comparación con modelos más complejos.
- **Adecuación:** La regresión lineal se adapta bien a los datos de viajes.

Pueden visualizar el código en el Anexo 5.0.

## 5.2 Prophet.

Prophet fue utilizado para realizar predicciones de series temporales, aprovechando su capacidad para manejar estacionalidades (Año, mes, días) y tendencias.

### Justificación:

- **Flexibilidad:** Prophet permite modelar datos con tendencias no lineales y cambios estacionales.
- **Interactividad:** Permite ajustes en los componentes del modelo, lo que facilita la personalización.
- **Predicciones a largo plazo:** Es ideal para pronósticos a largo plazo, como se requiere en este proyecto.

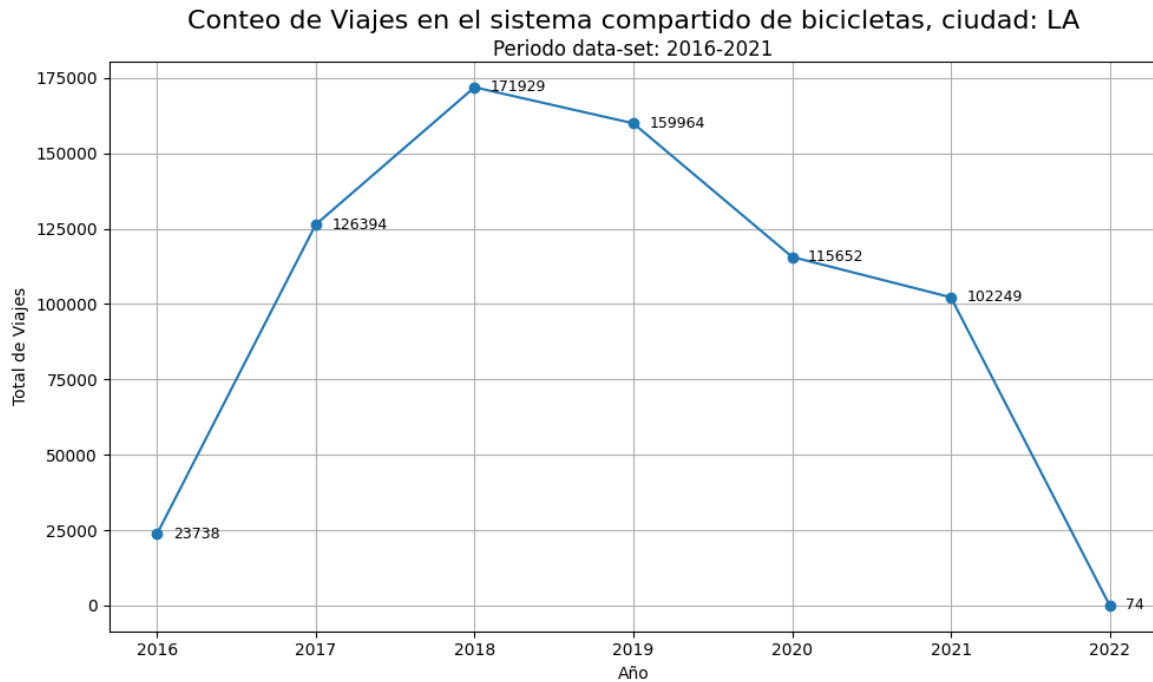
Pueden visualizar el código en el Anexo 5.1.

## 6. Resultados y Visualizaciones.

Se generaron visualizaciones que muestran las tendencias y proyecciones a futuro, incluyendo:

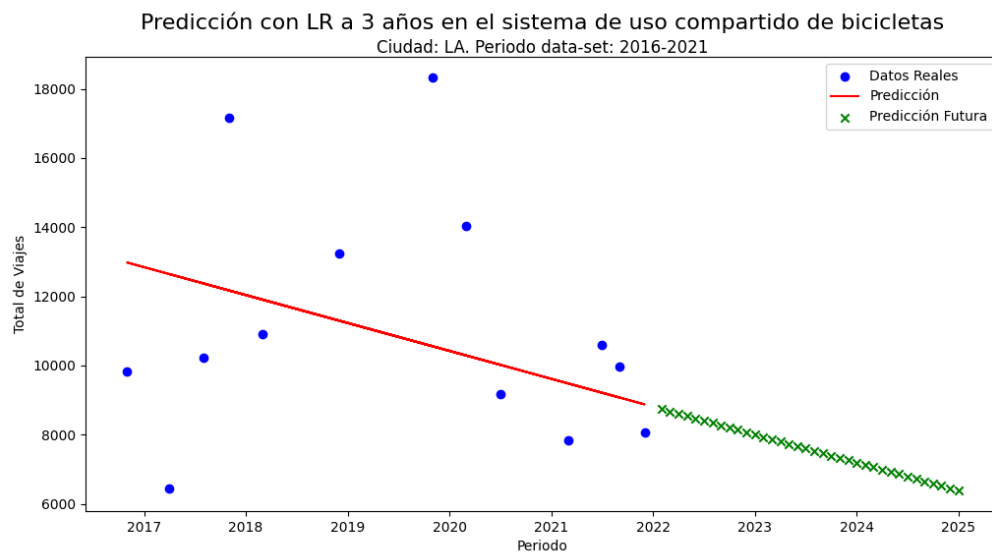
- **Gráficos de tendencias:** Para visualizar la evolución de los viajes.
- **Proyección a 3 años:** Utilizando gráficos de barras para mostrar las predicciones anuales.

En el Gráfico 1, puede verse que al iniciar el registro de viajes en 2016 se incrementó su demanda hasta el 2018, **fue durante y después de 2019 entrando la pandemia a los EU**, que la demanda baja significativamente hasta los últimos registros en **2021**, en dicho año fue donde la pandemia empezó a tener una mejor regularización de control sanitario, demostrando en 2021 no tener un déficit igual de pronunciado como de 2019 a 2020.



*Gráfico 1 Conteo de viajes al año a lo largo del tiempo dentro del data set.*

Al usar la regresión lineal del Gráfico 2, se pudo inferir, en caso de que la pandemia siguiera con sus restricciones de cuarentena que el déficit en demanda para el servicio pudo incrementarse, pero este modelo no es confiable debido a sus limitaciones en la comprensión del entorno, resultado en un MSE elevado.



*Gráfico 2 Uso de regresión lineal para la predicción a 3 años.*

Por otro lado, dentro de la versatilidad en Prophet para su personalización, se identificaron las tendencias del servicio en uso: anual, mensual, semanal. Además de una predicción a futuro más confiable. En la Gráfico 3, se visualizan las tendencias del servicio, destacando que los fines de semana son de mayor demanda, con un alza en el uso del servicio desde el día jueves y un decremento hacia el día lunes. Además, puede identificarse que la saturación del servicio se encuentra durante el otoño. Por último, se identificó que los horarios más concurridos para las estaciones con de mayor demanda son en primer lugar alrededor de la media noche (22 hr a 03 hrs) y en segundo lugar por la tarde (14 hrs a 16 hrs), ver Anexo 7.

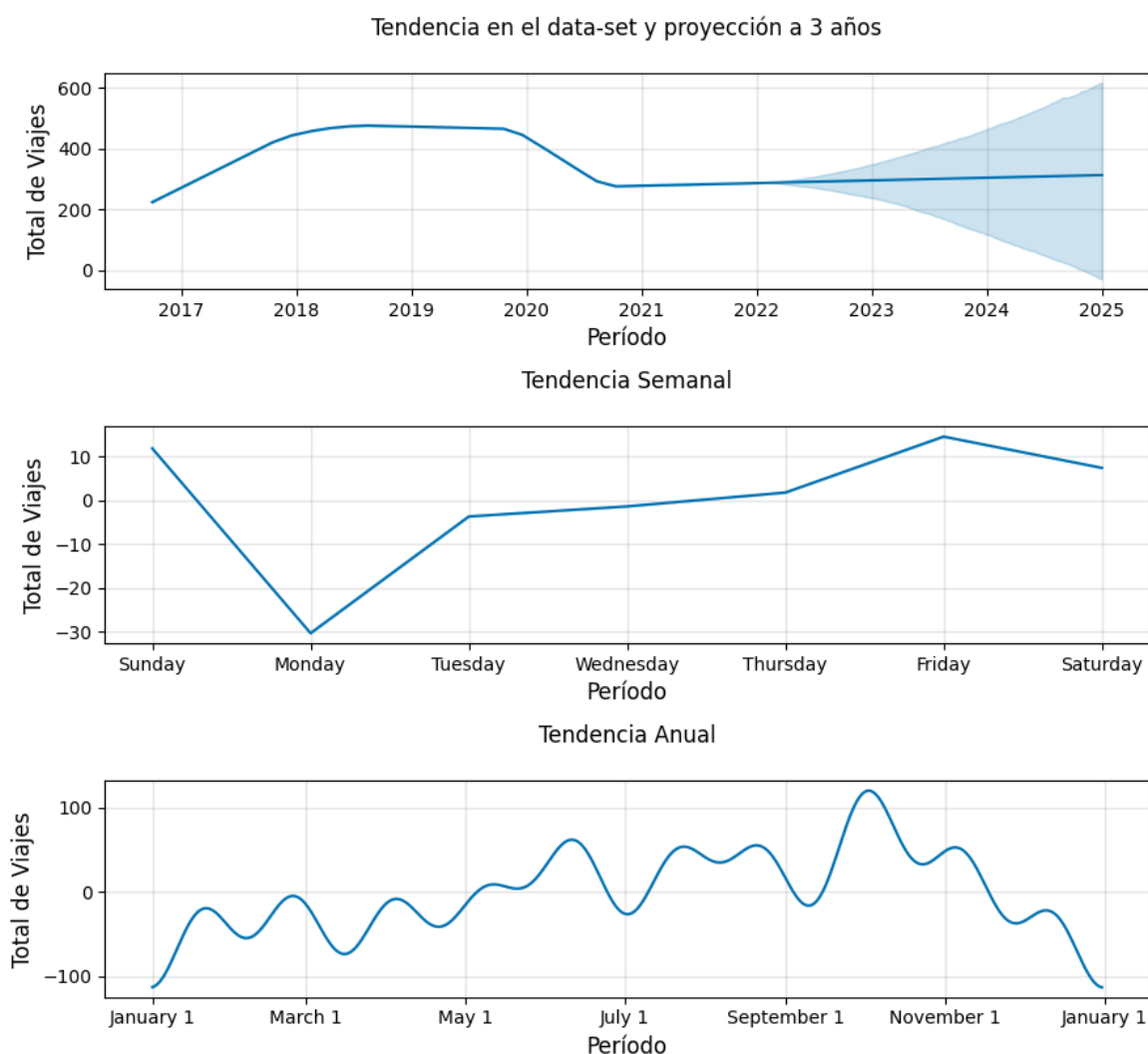


Gráfico 3 Tendencias en el uso del servicio por: Año, mes, semana, y una predicción a 3 años en el uso del servicio.

Por último, en el Gráfico 4, se muestran la tendencia en el uso del servicio del data set con una predicción a 3 años y en el Gráfico 5 se muestra el conteo de viajes del registro y una predicción a 3 años en la demanda.

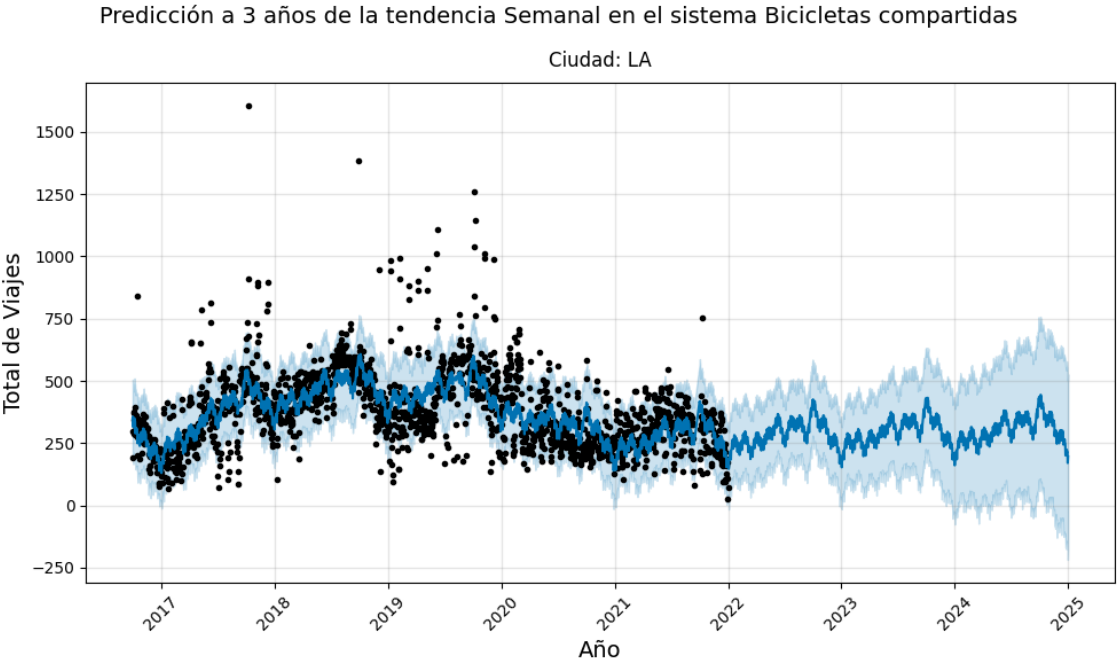


Gráfico 4 Tendencia anual a lo largo del tiempo con predicción a 3 años.

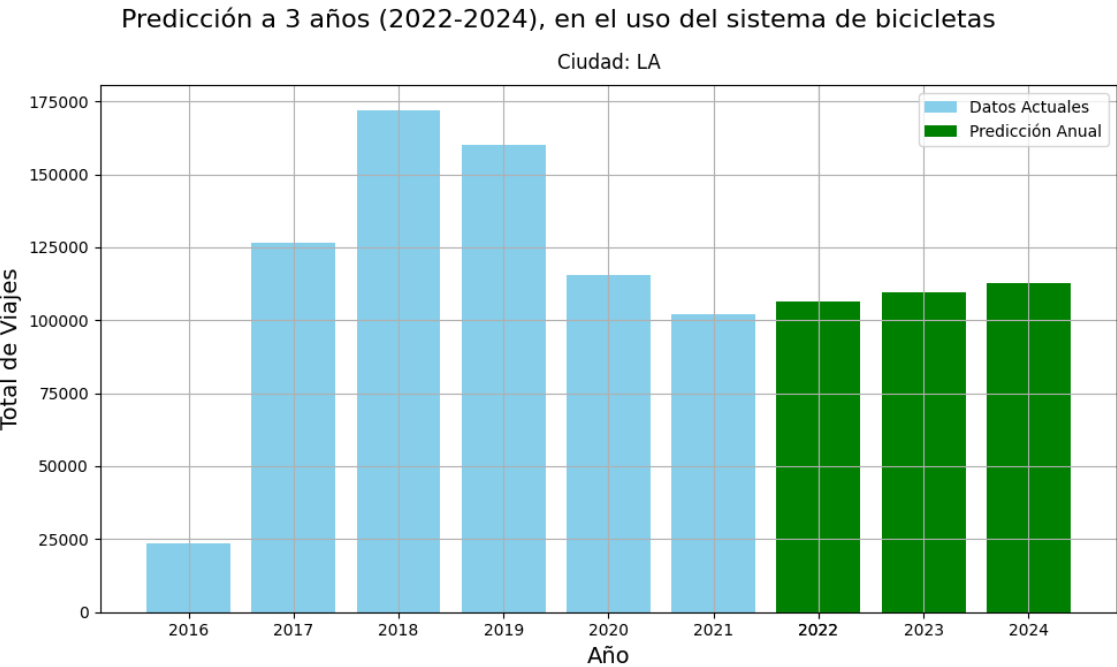
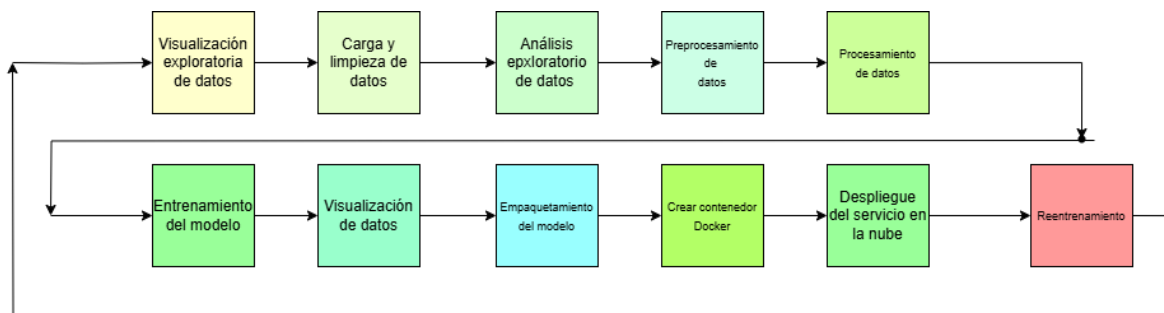


Gráfico 5 Predicción de los siguientes 3 años, respecto a la información a lo largo del tiempo.

Producción del modelo:

En el Diagrama 1 se propone como poner en producción el modelo analítico.



*Diagrama 1 Propuesta de producción del modelo analítico.*

## Conclusiones.

- **Prophet** predice un incremento en la demanda del servicio, lo cual tiene sentido al concluir los años de pandemia.
- Se identificaron patrones significativos en el uso del servicio: baja demanda iniciando la pandemia en 2019, y una regularización en 2021.
- Se recomienda considerar variables adicionales, como condiciones climáticas y eventos locales, para mejorar la precisión del modelo.
- La época con mayor demanda es otoño.
- Los días de mayor demanda son en fin de semana, iniciando el jueves y terminando el domingo
- Los horarios de mayor saturación son cercanos a la media noche, seguido de los horarios entre las 14 hrs y 16 hrs.

## Recomendaciones

- Continuar monitoreando los datos y ajustar los modelos según sea necesario.
- Explorar modelos más complejos si se dispone de más datos y recursos.
- Modularizar los Scripts para ejecutar en conjunto, y para crear un modelo escalable.

# Anexos.

## Anexo 1: Script: visualización\_general.py

Código para conteo de Viajes a lo largo del tiempo:

```
1  import pandas as pd
2  import matplotlib.pyplot as plt
3  import os
4
5  df = pd.read_csv(r"data_proscesed\test_set_Preproceced_090225_V1.0.0_JT.csv", low_memory=False)
6  df['start_date'] = pd.to_datetime(df['start_date'])
7  df['year'] = df['start_date'].dt.year
8  total_trips_per_year = df.groupby('year').size().reset_index(name='total_trips')
9
10 plt.figure(figsize=(10, 6))
11 plt.plot(total_trips_per_year['year'], total_trips_per_year['total_trips'], marker='o')
12 plt.title('Conteo de Viajes en el sistema compartido de bicicletas, ciudad: LA', fontsize=16, loc='center', pad=20)
13 plt.text(0.5, 1.02, 'Periodo data-set: 2016-2021', fontsize=12, ha='center', va='center', transform=plt.gca().transAxes)
14 plt.xlabel('Año')
15 plt.ylabel('Total de Viajes')
16
17 for i in range(len(total_trips_per_year)):
18     plt.text(total_trips_per_year['year'][i] + 0.1, total_trips_per_year['total_trips'][i],
19             str(total_trips_per_year['total_trips'][i]), ha='left', va='center', fontsize=9)
20
21 plt.xticks(total_trips_per_year['year'])
22 plt.grid()
23 plt.tight_layout()
24
25 if not os.path.exists('grafic'):
26     os.makedirs('grafic')
27
28 plt.savefig('grafic/conteo_viajes_por_año.png')
29 plt.show()
```

*Anexo 1 Código visualizacion\_general.py, este Script se encuentra en la carpeta src.*

## Anexo 2: Script: histograma.py

Código para obtener el Histograma de viajes de duración en minutos respecto a la frecuencia de uso.

```
1  import pandas as pd
2  import seaborn as sns
3  import matplotlib.pyplot as plt
4  import os
5
6  df = pd.read_csv(r"data_prosseced\test_set_Preproceced_090225_V1.0.0_JT.csv", low_memory=False)
7  df = df.dropna()
8  df_filtered = df[(df['duration'] >= 0) & (df['duration'] <= 60)]
9
10 if not os.path.exists('grafic'):
11     os.makedirs('grafic')
12
13 plt.figure(figsize=(10, 6))
14 sns.histplot(df_filtered['duration'], bins=30, kde=True)
15 plt.title('Distribución de la Duración de los Viajes (0 a 60 minutos)')
16 plt.xlabel('Duración (minutos)')
17 plt.ylabel('Frecuencia')
18 plt.xlim(0, 60)
19 plt.tight_layout()
20 plt.savefig('grafic/distribucion_duracion_viajes.png')
21 plt.close()
```

*Anexo 2 Código histograma.py, este script se encuentra en la carpeta src.*

## Anexo 3: Script: estaciones\_saturadas.py

Código para obtener la información Top 10 rutas más utilizadas periodo 2016 - 2021.

```
1 import pandas as pd
2 import numpy as np
3
4 df = pd.read_csv(r"data_prosseced\test_set_Preproceced_090225_V1.0.0_JT.csv", low_memory=False)
5
6 station_pairs = df.groupby(['start_station', 'end_station']).size().reset_index(name='count')
7 station_pairs = station_pairs.sort_values('count', ascending=False)
8
9 print('Rutas más utilizadas:')
10 print(station_pairs.head(10))
11
12 total_trips = len(df)
13 print(f'Número total de viajes: {total_trips}')
14
15 mean_duration = df['duration'].mean()
16 print(f'Duración media de los viajes: {mean_duration:.2f} minutos')
17
18 station_usage = df.groupby('start_station').size().reset_index(name='count')
19 station_usage = station_usage.sort_values('count', ascending=False)
20 print('Frecuencia de uso por estación:')
21 print(station_usage)
22
23 top_stations_by_trips = station_usage.sort_values('count', ascending=False)
24 print('Estaciones más concurridas por número de viajes:')
25 print(top_stations_by_trips.head(10))
26
27 station_durations = df.groupby('start_station')['duration'].mean().reset_index(name='mean_duration')
28 top_stations_by_duration = station_durations.sort_values('mean_duration', ascending=False)
29 print('Estaciones más concurridas por duración media de los viajes:')
30 print(top_stations_by_duration.head(10))
```

*Anexo 3 Código de estaciones\_saturadas.py, este Script se encuentra en la carpeta src*



## Anexo 4: Script: horarios\_saturados.py

Este código identifica las 10 estaciones más saturadas con sus 4 horarios mas concurridos para cada una de las estaciones antes mencionada

```
1 import pandas as pd
2
3 df = pd.read_csv(r"data_prosced\test_set_Preproceded_090225_V1.0.0_JT.csv", low_memory=False)
4
5 station_pairs = df.groupby(['start_station', 'end_station']).size().reset_index(name='count')
6 top_routes = station_pairs.sort_values('count', ascending=False).head(10)
7
8
9 print('Top 10 rutas más utilizadas:')
10 print(top_routes)
11
12 for idx, route in top_routes.iterrows():
13     start_station = route['start_station']
14     end_station = route['end_station']
15     filtered_data = df[(df['start_station'] == start_station) & (df['end_station'] == end_station)]
16     filtered_data['hour'] = pd.to_datetime(filtered_data['start_time']).dt.hour
17     hourly_counts = filtered_data.groupby(['start_station', 'end_station', 'hour', 'passholder_type']).size().reset_index(name='count')
18     top_hours = hourly_counts.sort_values('count', ascending=False).head(4)
19
20     print(f'\nTop 4 horarios más saturados para la ruta iniciando en: {start_station} -> concluyendo en: {end_station}:')
21     print(top_hours)
```

*Anexo 4 Código de horarios\_saturados.py, este código se encuentra en la carpeta src*

## Anexo 5: Script: modelo\_LR.py

Código de modelo analítico de Regresión Lineal (LR).

```
1 import pandas as pd
2 from sklearn.model_selection import train_test_split
3 from sklearn.linear_model import LinearRegression
4 from sklearn.metrics import mean_squared_error
5 import matplotlib.pyplot as plt
6 import numpy as np
7
8 df = pd.read_csv(r"data_prosced\test_set_Preproceded_090225_V1.0.0_JT.csv", low_memory=False)
9
10 df['start_datetime'] = pd.to_datetime(df['start_date'] + ' ' + df['start_hour'])
11 df['year'] = df['start_datetime'].dt.year
12 df['month'] = df['start_datetime'].dt.month
13 df['day'] = df['start_datetime'].dt.day
14 df['day_of_week'] = df['start_datetime'].dt.dayofweek
15 df['hour'] = df['start_datetime'].dt.hour
16
17 total_trips_per_month = df.groupby(['year', 'month']).size().reset_index(name='total_trips')
18 total_trips_per_month['time'] = total_trips_per_month['year'] + total_trips_per_month['month'] / 12
19
20 X = total_trips_per_month[['time']]
21 y = total_trips_per_month['total_trips']
22 X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
23
24 model = LinearRegression()
25 model.fit(X_train, y_train)
26
27 y_pred = model.predict(X_test)
28
29 mse = mean_squared_error(y_test, y_pred)
30 print(f'Error Cuadrático Medio: {mse}')
31
32 future_years_months = []
33 for year in range(total_trips_per_month['year'].max(), total_trips_per_month['year'].max() + 3):
34     for month in range(1, 13):
35         future_years_months.append(year + month / 12)
36
37 future_predictions = model.predict(np.array(future_years_months).reshape(-1, 1))
38
39 for year_month, prediction in zip(future_years_months, future_predictions):
40     year = int(year_month)
41     month = int((year_month - year) * 12) + 1
42     print(f'Predicción de viajes para {year}-{month:02d}: {prediction}')
43
44 plt.figure(figsize=(12, 6))
45 plt.scatter(X_test, y_test, color='blue', label='Datos Reales')
46 plt.plot(X_test, y_pred, color='red', label='Predicción')
47 plt.scatter(future_years_months, future_predictions, color='green', label='Predicción Futura', marker='x')
48 plt.title('Predicción con LR a 3 años en el sistema de uso compartido de bicicletas', fontsize=16, loc='center', pad=20)
49 plt.text(0.5, 1.02, 'Ciudad: LA. Periodo data-set: 2016-2021', fontsize=12, ha='center', va='center', transform=plt.gca().transAxes)
50 plt.xlabel('Periodo')
51 plt.ylabel('Total de Viajes')
52 plt.legend()
53 plt.savefig('grafic/Modelo_analitico_(LR).png')
54 plt.show()
```

*Anexo 5 Código de modelo\_LR.py, este código se encuentra en la carpeta src.*

## Anexo 6. Script modelo\_prophet.py

Código para el Modelo analítico usando Prophet de Meta.

```
1 import pandas as pd
2 import matplotlib.pyplot as plt
3 import os
4 from prophet import Prophet
5
6 df = pd.read_csv(r"data_procesado/test_set_Preprocesado_090225_V1.0.0_3T.csv", low_memory=False)
7
8 df['start_date'] = pd.to_datetime(df['start_date'])
9 df['year'] = df['start_date'].dt.year
10 df['month'] = df['start_date'].dt.month
11 df['day_of_week'] = df['start_date'].dt.day_name()
12
13 total_trips_per_year = df.groupby('year').size().reset_index(name='total_trips')
14 plt.figure(figsize=(10, 6))
15 plt.plot(total_trips_per_year['year'], total_trips_per_year['total_trips'], marker='o')
16 plt.title('Conteo de Viajes en el sistema compartido de bicicletas, ciudad: LA', fontsize=16, loc='center', pad=20)
17 plt.text(0.5, 1.02, 'Periodo data-set: 2016-2021', fontsize=12, ha='center', va='center', transform=plt.gca().transAxes)
18 plt.xlabel('Año')
19 plt.ylabel('Total de Viajes')
20
21 for i in range(len(total_trips_per_year)):
22     plt.text(total_trips_per_year['year'][i] + 0.1, total_trips_per_year['total_trips'][i],
23             str(total_trips_per_year['total_trips'][i]), ha='left', va='center', fontsize=9)
24
25 plt.xticks(total_trips_per_year['year'])
26 plt.grid()
27 plt.tight_layout()
28 plt.show()
29
30 total_trips_per_day = df.groupby('start_date').size().reset_index(name='total_trips')
31 total_trips_per_day.columns = ['ds', 'y']
32
33 model = Prophet(weekly_seasonality=True, yearly_seasonality=True)
34 model.fit(total_trips_per_day)
35
36 future = model.make_future_dataframe(periods=1095)
37 forecast = model.predict(future)
38 fig = model.plot_components(forecast)
39
40 for ax in fig.axes:
41     ax.set_xlabel('Periodo', fontsize=12)
42     ax.set_ylabel('Total de Viajes', fontsize=12)
43     ax.grid(True)
44
45 fig.axes[0].set_title('Tendencia en el data-set y proyección a 3 años\n', fontsize=12)
46 fig.axes[1].set_title('Tendencia Semanal\n', fontsize=12)
47 fig.axes[2].set_title('Tendencia Anual\n', fontsize=12)
48 plt.tight_layout(rect=[0, 0.03, 1, 0.95])
49 plt.savefig('grafic/Modelo_analitico_Prophet_Tendencias.png')
50 plt.show()
51
52 fig2 = model.plot(forecast)
53 plt.suptitle('Predicción a 3 años de la tendencia Semanal en el sistema Bicicletas compartidas', fontsize=14)
54 plt.text(0.5, 1.02, 'Ciudad: LA\n', fontsize=12, ha='center', va='center', transform=plt.gca().transAxes)
55 plt.xlabel('Año', fontsize=14)
56 plt.ylabel('Total de Viajes', fontsize=14)
57 plt.grid(True)
58 plt.xticks(rotation=45)
59 plt.tight_layout()
60 plt.savefig('grafic/Modelo_analitico_Prophet_Tendencias_actuales.png')
61 plt.show()
62
63 annual_forecast = forecast.resample('Y', on='ds').sum()
64 plt.figure(figsize=(10, 6))
65 years_actual = total_trips_per_year['year']
66 trips_actual = total_trips_per_year['total_trips']
67 years_forecast = annual_forecast.index.year
68 trips_forecast = annual_forecast['yhat']
69 plt.bar(years_actual, trips_actual, color='skyblue', label='Datos Actuales')
70 plt.bar(years_forecast[years_forecast >= 2022], trips_forecast[years_forecast >= 2022], color='green', label='Predicción Anual')
71 plt.suptitle('Predicción a 3 años (2022-2024), en el uso del sistema de bicicletas', fontsize=16)
72 plt.text(0.5, 1.02, 'Ciudad: LA\n', fontsize=12, ha='center', va='center', transform=plt.gca().transAxes)
73 plt.xlabel('Año', fontsize=14)
74 plt.ylabel('Total de Viajes', fontsize=14)
75 plt.xticks(years_actual.tolist() + years_forecast[years_forecast >= 2022].tolist())
76 plt.grid()
77 plt.tight_layout()
78 plt.legend()
79 plt.savefig('grafic/Modelo_analitico_Prophet_predicción_3años.png')
80 plt.show()
```

Anexo 6 Código de modelo\_prophet.py, este código se encuentra en la carpeta src.

## Gráficos

Gráfico 1: Conteo de viajes al año a lo largo del tiempo dentro del data set.

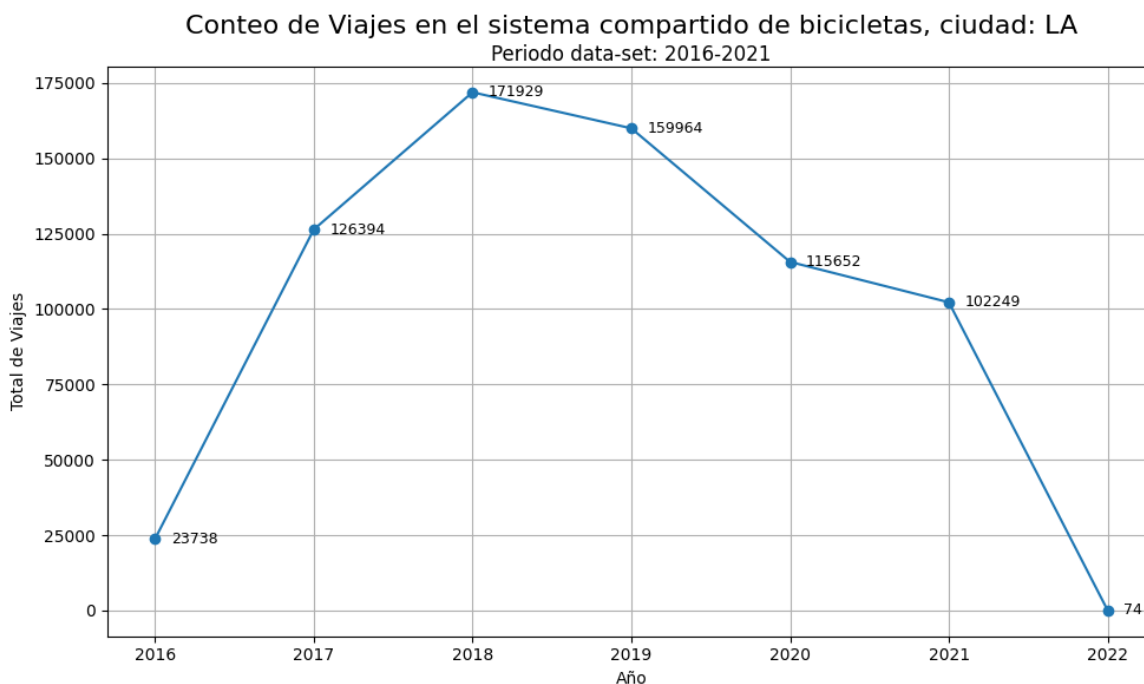


Gráfico 2: Uso de regresión lineal para la predicción a 3 años.

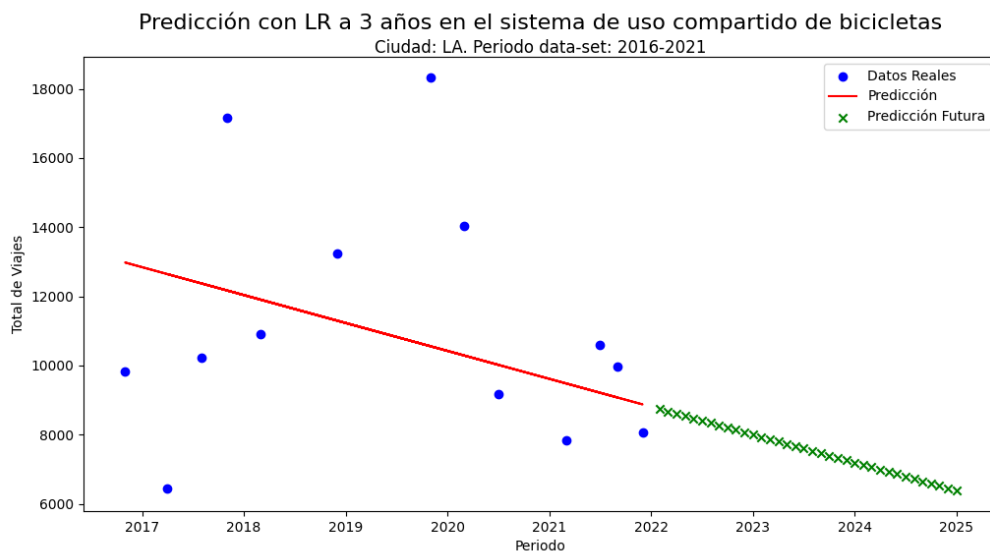


Gráfico 3. Tendencias en el uso del servicio por: Año, mes, semana, y una predicción a 3 años en el uso del servicio.

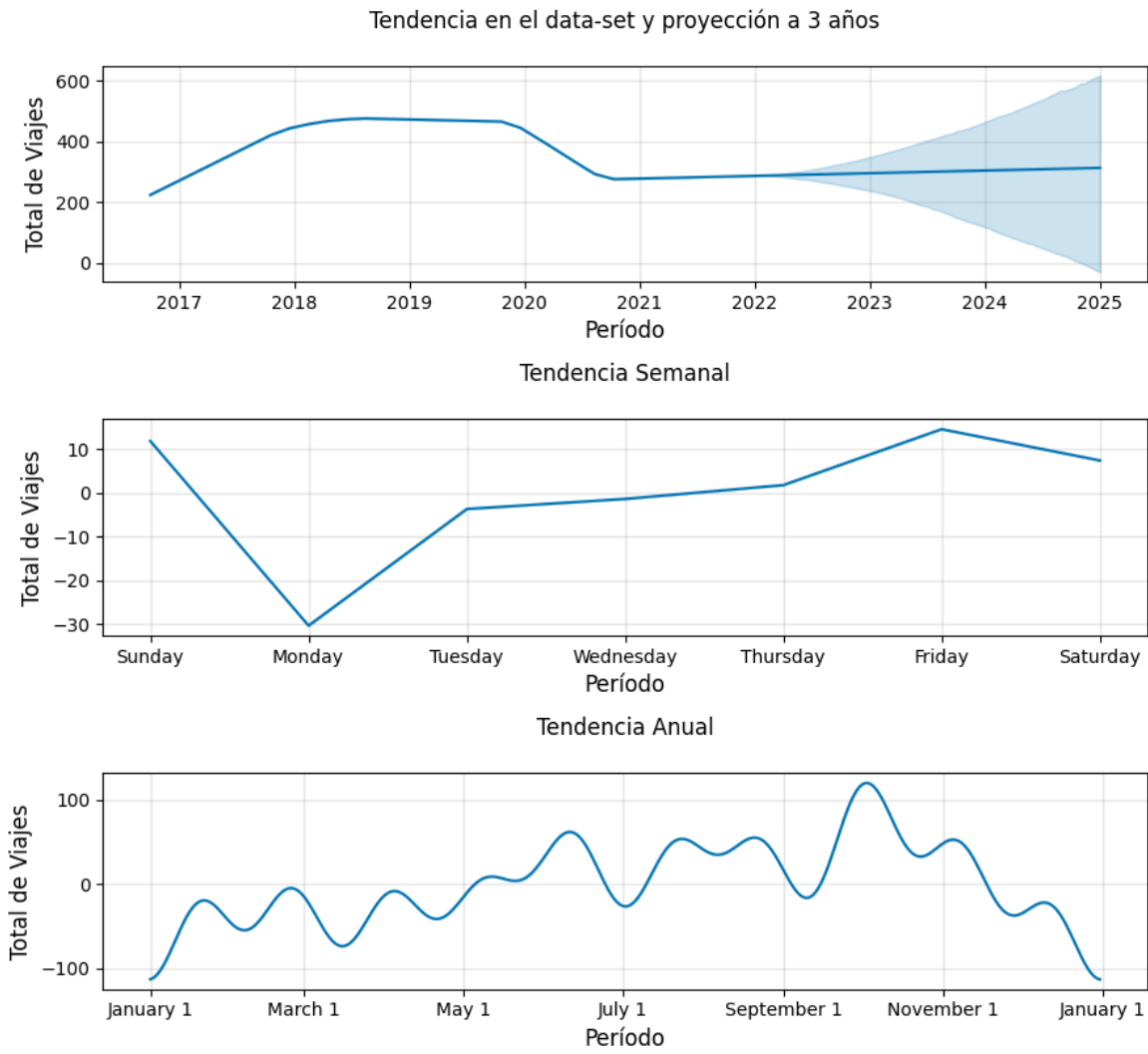


Gráfico 4. Tendencia anual a lo largo del tiempo con predicción a 3 años.

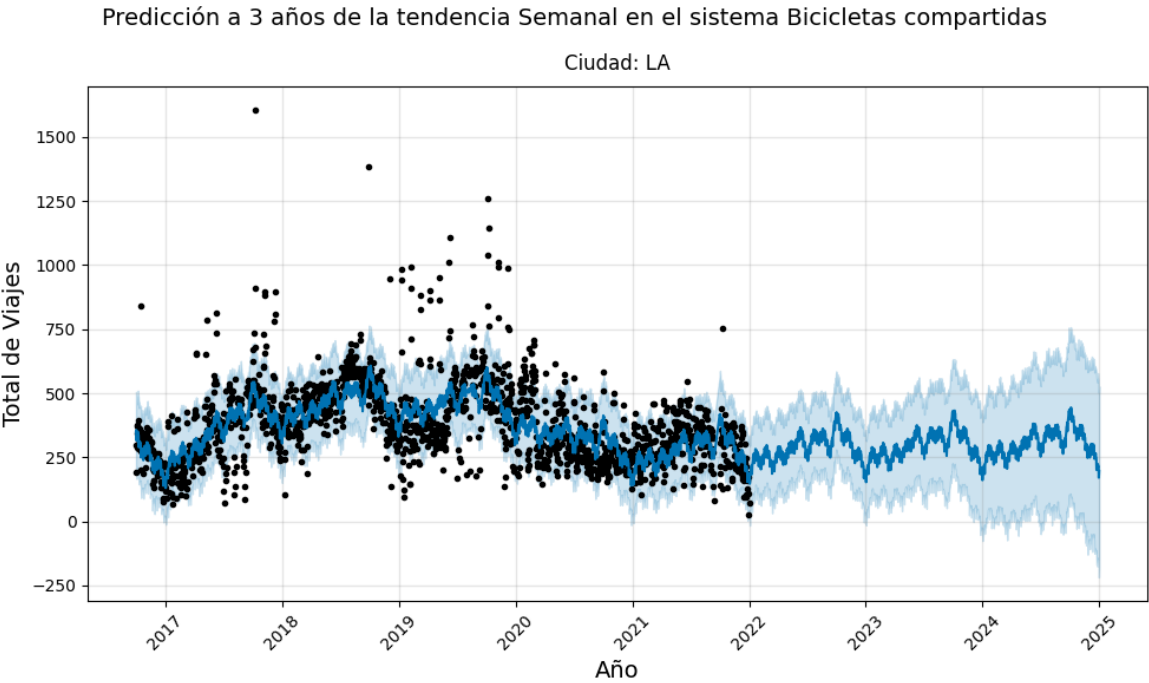
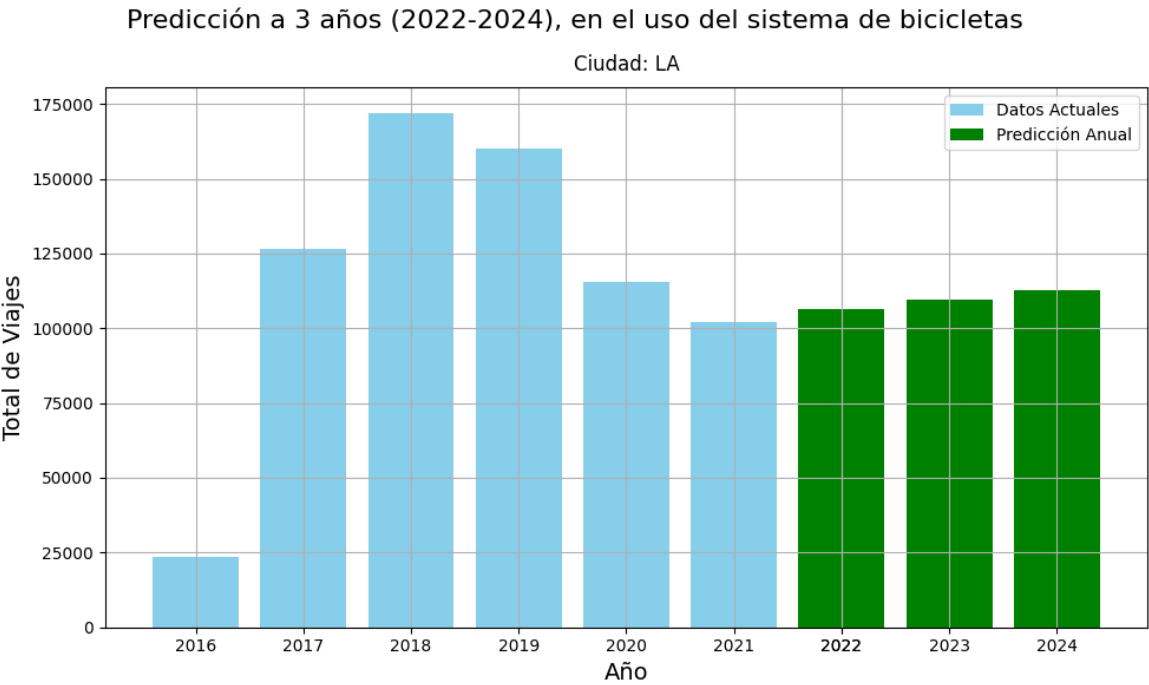


Gráfico 5. Predicción de los siguientes 3 años, respecto a la información a lo largo del tiempo.



## Diagramas.

Diagrama 1 Propuesta de producción del modelo analítico.

