

## --Introducción: Contexto del Proyecto.

**\*\* Propósito del análisis:** El análisis busca identificar los municipios con mayor riesgo de delitos relacionados con automóviles. Esto es crucial para ajustar los precios de las pólizas de seguros de manera competitiva y rentable.

**\*\* Incidencia de delitos y determinación del riesgo:** La incidencia de delitos aumenta el riesgo de siniestros, lo que justifica la necesidad de ajustar el costo del seguro en función de este riesgo.

## --Descarga y creación de la Base de Datos.

**\*\* Descarga de los datos:** Los datos de delitos se obtuvieron desde la plataforma oficial de datos abiertos de México. A continuación el link a la página

<https://datos.gob.mx/busca/dataset/incidencia-delictiva-del-fuero-comun-a-nivel-municipal>

**\*\* Creación de la base de datos en SQL:** Aquí se muestra la creación de la base de datos:

```
# Descarga del archivo

import pandas as pd
import sqlite3
import numpy as np
import os

os.chdir("/Users/gilgu/OneDrive/Escritorio")

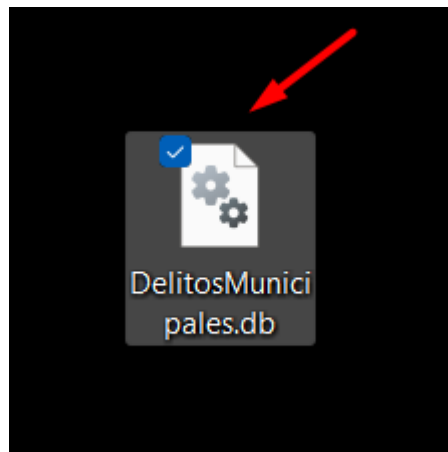
data = pd.read_csv('Municipal-Delitos-2015-2021_dic2021.csv', encoding='ISO-8859-1')
```

data

]:

	Año	Clave_Ent	Entidad	Cve. Municipio	Municipio	Bien jurídico afectado	Tipo de delito	Subtipo de delito	Modalidad	Enero	...	Marzo	Abril	Mayo	Junio
0	2015	1	Aguascalientes	1001	Aguascalientes	La vida y la Integridad corporal	Homicidio	Homicidio doloso	Con arma de fuego	2	...	1	1	0	1
1	2015	1	Aguascalientes	1001	Aguascalientes	La vida y la Integridad corporal	Homicidio	Homicidio doloso	Con arma blanca	1	...	0	0	0	1
1589458	2021	32	Zacatecas	32058	Santa María de la Paz	Otros bienes jurídicos afectados (del fuero co...	Contra el medio ambiente	Contra el medio ambiente	Contra el medio ambiente	0	...	0	0	0	0
1589459	2021	32	Zacatecas	32058	Santa María de la Paz	Otros bienes jurídicos afectados (del fuero co...	Delitos cometidos por servidores públicos	Delitos cometidos por servidores públicos	Delitos cometidos por servidores públicos	0	...	0	0	0	0
					Santa María de la Paz	Otros bienes jurídicos afectados (del fuero co...									

```
» # Conexión a la base de datos SQLite (o crearla si no existe)
conn = sqlite3.connect('DelitosMunicipales.db')
```



```
» # Guardar el dataframe en una tabla SQL
data.to_sql('DelitosMunicipales', conn, if_exists='replace', index=False)

]: 1589462
```

```
# Verificar que la tabla se haya creado correctamente
```

```
query = "SELECT * FROM DelitosMunicipales LIMIT 5"
print(pd.read_sql(query, conn))
```

	Año	Clave_Ent	Entidad	Cve. Municipio	Municipio	\
0	2015	1	Aguascalientes	1001	Aguascalientes	
1	2015	1	Aguascalientes	1001	Aguascalientes	
2	2015	1	Aguascalientes	1001	Aguascalientes	
3	2015	1	Aguascalientes	1001	Aguascalientes	
4	2015	1	Aguascalientes	1001	Aguascalientes	

	Bien jurídico afectado	Tipo de delito	Subtipo de delito	\
0	La vida y la Integridad corporal	Homicidio	Homicidio doloso	
1	La vida y la Integridad corporal	Homicidio	Homicidio doloso	
2	La vida y la Integridad corporal	Homicidio	Homicidio doloso	
3	La vida y la Integridad corporal	Homicidio	Homicidio doloso	
4	La vida y la Integridad corporal	Homicidio	Homicidio culposo	

	Modalidad	Enero	...	Marzo	Abril	Mayo	Junio	Julio	Agosto	\
0	Con arma de fuego	2	...	1	1	0	1	1	0	
1	Con arma blanca	1	...	0	0	0	1	0	1	
2	Con otro elemento	0	...	1	1	3	2	0	1	
3	No especificado	1	...	0	1	0	0	0	0	
4	Con arma de fuego	0	...	0	0	1	0	0	0	

	Septiembre	Octubre	Noviembre	Diciembre
0	2	1	0	1
1	0	0	0	0
2	2	0	0	0
3	0	0	0	0
4	0	0	0	0

[5 rows x 21 columns]

DB Browser for SQLite - C:\Users\gilgu\OneDrive\Escritorio\DelitosMunicipales.db

File Edit View Tools Help

New Database Open Database Write Changes Revert Changes Undo Open Project Save Project Attach Database Close Database

Database Structure Browse Data Edit Pragma Execute SQL

Table: DelitosMunicipales Filter in any column

	Año	Clave_Ent	Entidad	Cve. Municipio	Municipio	Bien jurídico afectado	Tipo de delito
	Filter	Filter	Filter	Filter	Filter	Filter	Filter
1	2015	1	Aguascalientes	1001	Aguascalientes	La vida y la Integridad corporal	Homicidio
2	2015	1	Aguascalientes	1001	Aguascalientes	La vida y la Integridad corporal	Homicidio
3	2015	1	Aguascalientes	1001	Aguascalientes	La vida y la Integridad corporal	Homicidio
4	2015	1	Aguascalientes	1001	Aguascalientes	La vida y la Integridad corporal	Homicidio
5	2015	1	Aguascalientes	1001	Aguascalientes	La vida y la Integridad corporal	Homicidio

DB Browser for SQLite - C:\Users\gilgu\OneDrive\Escritorio\DelitosMunicipales.db

File Edit View Tools Help

New Database Open Database Write Changes Revert Changes Undo Open Project Save Project Attach Database Close Database

Database Structure Browse Data Edit Pragmas Execute SQL

SQL 1\*

1

2

SELECT \* FROM DelitosMunicipales LIMIT 5;

	Año	Clave_Ent	Entidad	Cve. Municipio	Municipio	Bien jurídico afectado	Tipo de delito	Subtipo de delito	Modalidad
1	2015	1	Aguascalientes	1001	Aguascalientes	La vida y la Integridad corporal	Homicidio	Homicidio doloso	Con arma de fuego
2	2015	1	Aguascalientes	1001	Aguascalientes	La vida y la Integridad corporal	Homicidio	Homicidio doloso	Con arma blanca
3	2015	1	Aguascalientes	1001	Aguascalientes	La vida y la Integridad corporal	Homicidio	Homicidio doloso	Con otro elemento
4	2015	1	Aguascalientes	1001	Aguascalientes	La vida y la Integridad corporal	Homicidio	Homicidio doloso	No especificado
5	2015	1	Aguascalientes	1001	Aguascalientes	La vida y la Integridad corporal	Homicidio	Homicidio culposo	Con arma de fuego

Execution finished without errors.  
Result: 5 rows returned in 13ms  
At line 2:  
SELECT \* FROM DelitosMunicipales LIMIT 5;

```

# Crea un cursor desde Python que permite conectarse y hacer consultas o queries a la base de datos previamente creada.

import sqlite3

# Conectar a la base de datos
conn = sqlite3.connect('DelitosMunicipales.db')

# Crear un cursor
cursor = conn.cursor()

```

--Muestro el nombre de mis tablas existentes dentro de la base de datos:

DB Browser for SQLite - C:\Users\gilgu\OneDrive\Escritorio\DelitosMunicipales.db

File Edit View Tools Help

New Database Open Database Write Changes Revert Changes Undo

Database Structure Browse Data Edit Pragmas Execute SQL

SQL 1\*

```

1
2 SELECT name FROM sqlite_master WHERE type='table';
3
4

```

	name
1	DelitosMunicipales

--Esta consulta muestra el número total de filas en la tabla especificada:

The screenshot shows a SQL query editor window titled "SQL 1\*" with a red close button. The query is: `SELECT COUNT(*) FROM DelitosMunicipales;`. The results pane below shows a single row with the column name "COUNT(\*)" and the value "1589462".

```
1
2 SELECT COUNT(*) FROM DelitosMunicipales;
3
4
```

	COUNT(*)
1	1589462

--Esta consulta muestra el número total de filas en la tabla especificada, pero específicamente del año 2016:

The screenshot shows a SQL query editor window titled "SQL 1\*" with a red close button. The query is: `SELECT COUNT(*) FROM DelitosMunicipales WHERE año = 2016;`. The results pane below shows a single row with the column name "COUNT(\*)" and the value "189238".

```
1
2 SELECT COUNT(*) FROM DelitosMunicipales WHERE año = 2016;
3
4
```

	COUNT(*)
1	189238

**--Limpieza y Transformación de los Datos.**







```
» # Verificar valores nulos
print(data.isnull().sum())
```

```
Año                0
Clave_Ent          0
Entidad            0
Cve. Municipio     0
Municipio          0
Bien jurídico afectado 0
Tipo de delito     0
Subtipo de delito  0
Modalidad          0
Enero              0
Febrero            0
Marzo              0
Abril              0
Mayo               0
Junio              0
Julio              0
Agosto            0
Septiembre         0
Octubre            0
Noviembre          0
Diciembre          0
dtype: int64
```

---

```
» # Se Leen Los datos y se hace La Limpieza de estos a formatos que resulten más útiles para el posterior análisis.

# Eliminar filas con datos nulos

data_clean = data.dropna()
```

```
» # Guardar Los datos limpios en la base de datos

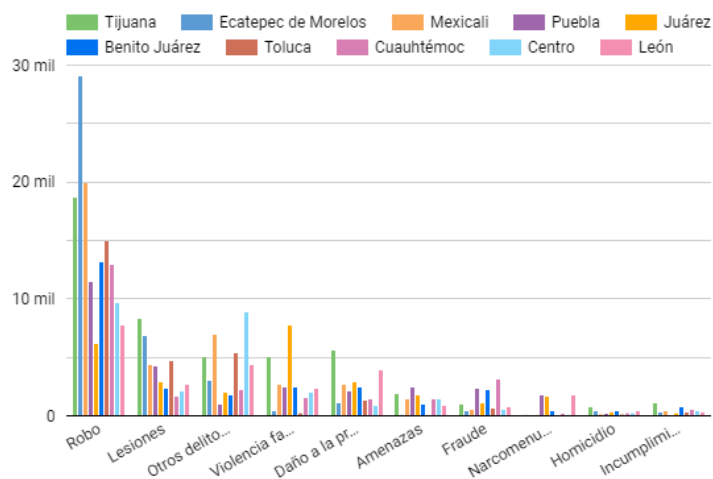
data_clean.to_sql('DelitosMunicipales_clean', conn, if_exists='replace', index=False)
```

### Resultado esperado:

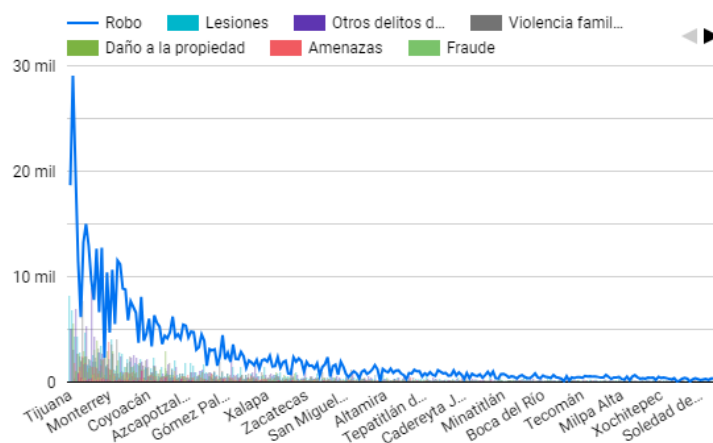
- Un dataset limpio con formatos correctos y sin valores nulos, o valores imputados adecuadamente.

--Visualizaciones del tablero de Locker Studio:

## AÑO 2015

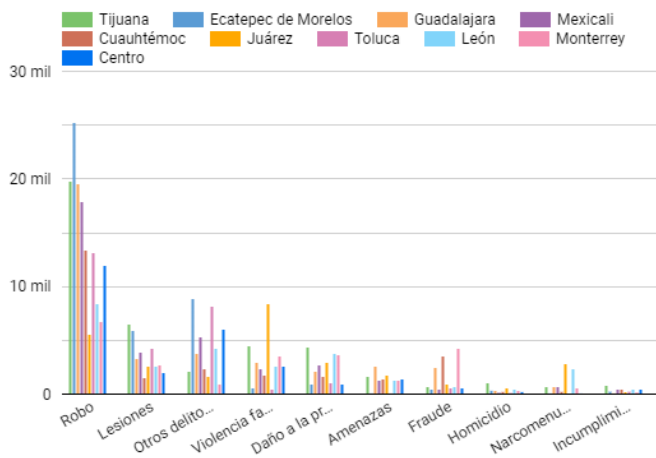


Tendencias generales de los delitos por tipo de delito

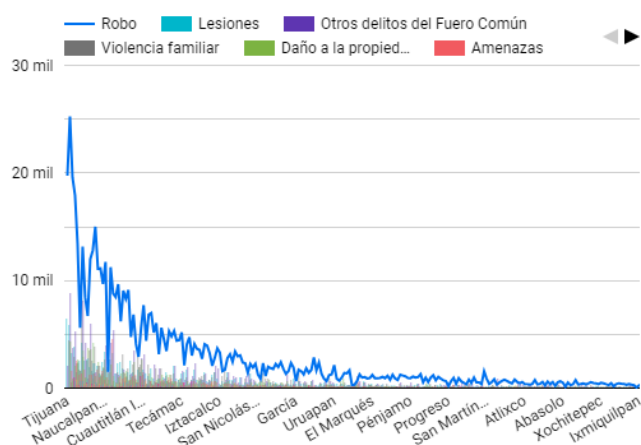


Tendencias generales de los delitos por municipio

## AÑO 2016

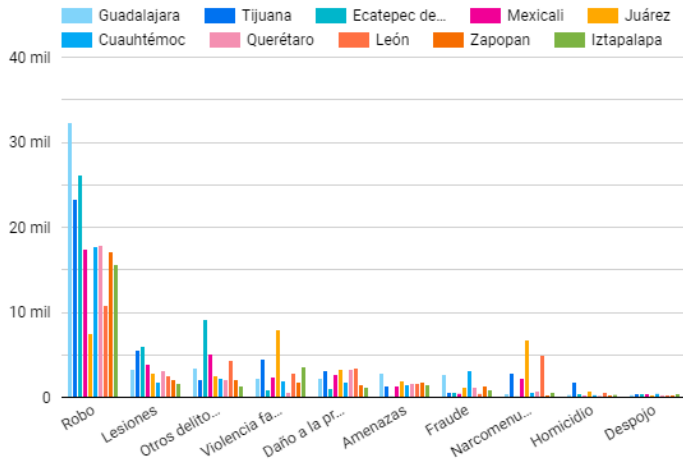


Tendencias generales de los delitos por tipo de delito

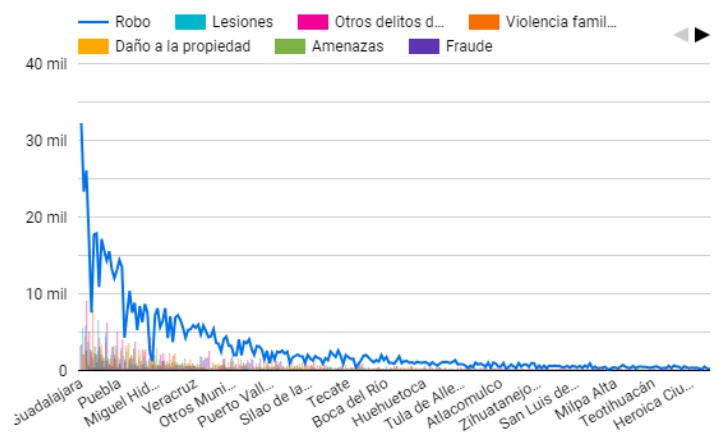


Tendencias generales de los delitos por municipio

# AÑO 2017

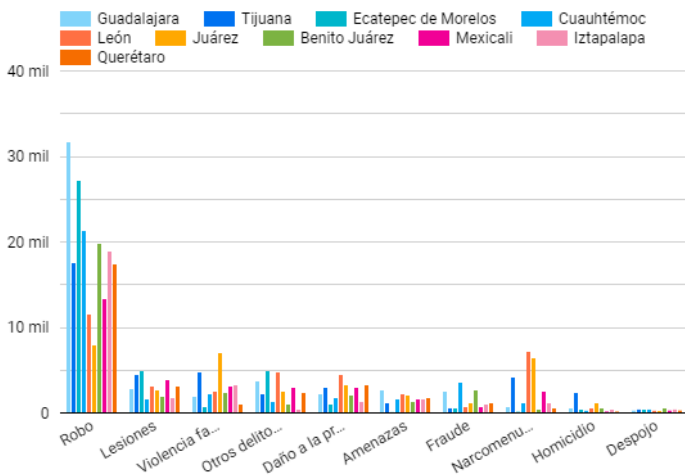


Tendencias generales de los delitos por tipo de delito

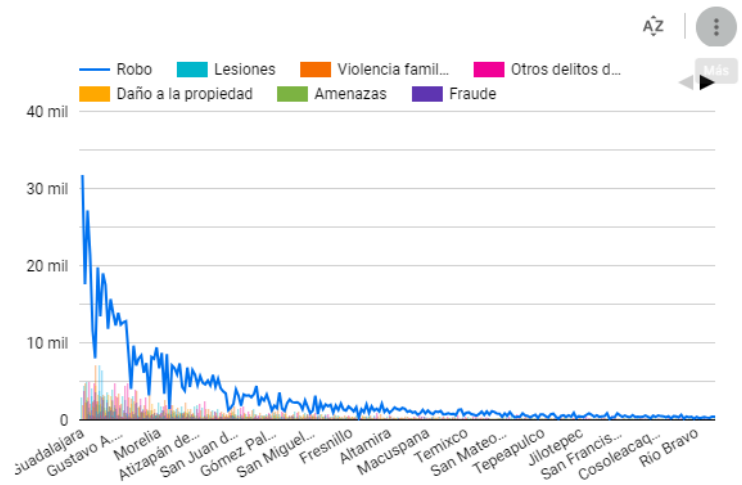


Tendencias generales de los delitos por municipio

# AÑO 2018

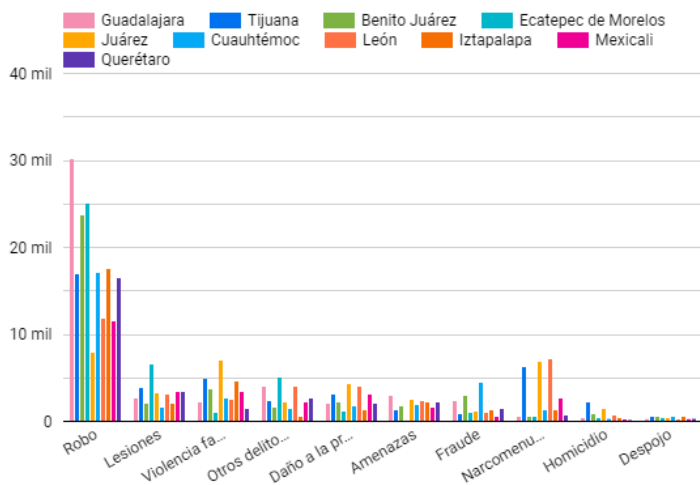


Tendencias generales de los delitos por tipo de delito

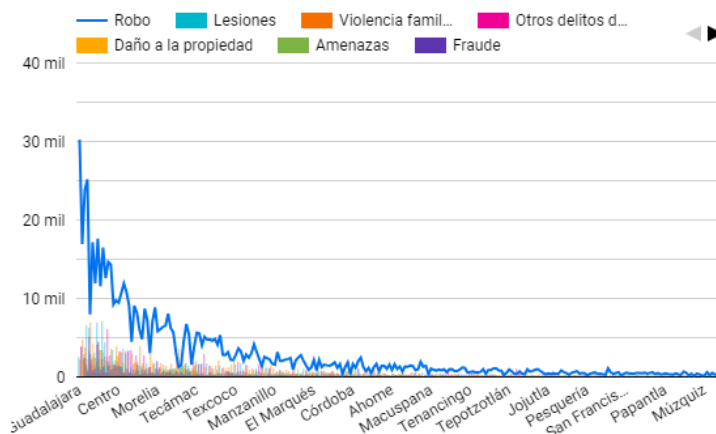


Tendencias generales de los delitos por municipio

# AÑO 2019

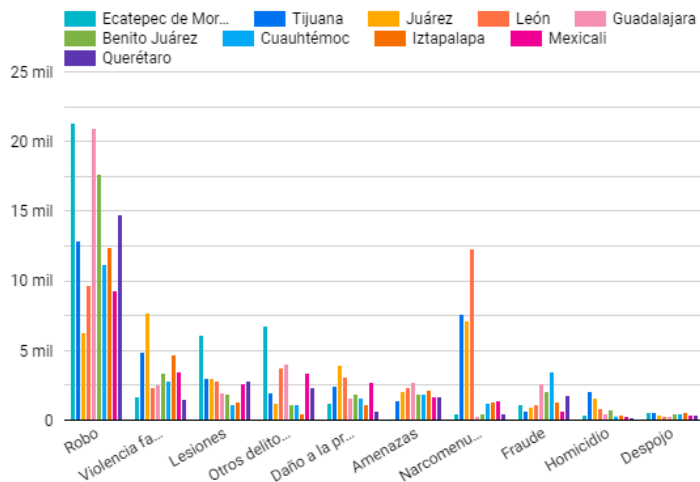


Tendencias generales de los delitos por tipo de delito

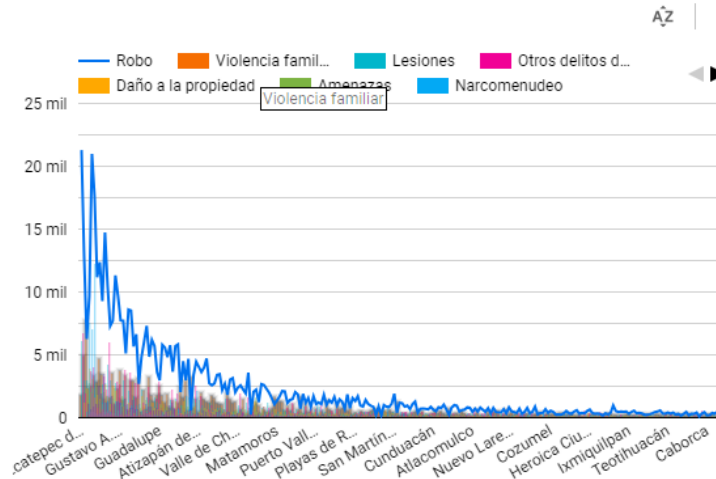


Tendencias generales de los delitos por municipio

# AÑO 2020

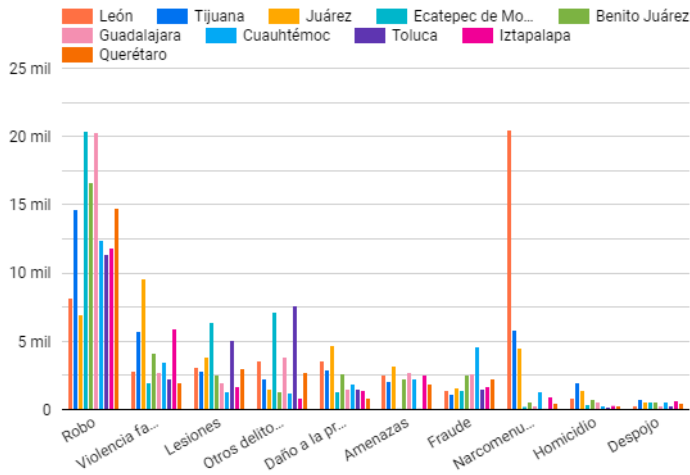


Tendencias generales de los delitos por tipo de delito

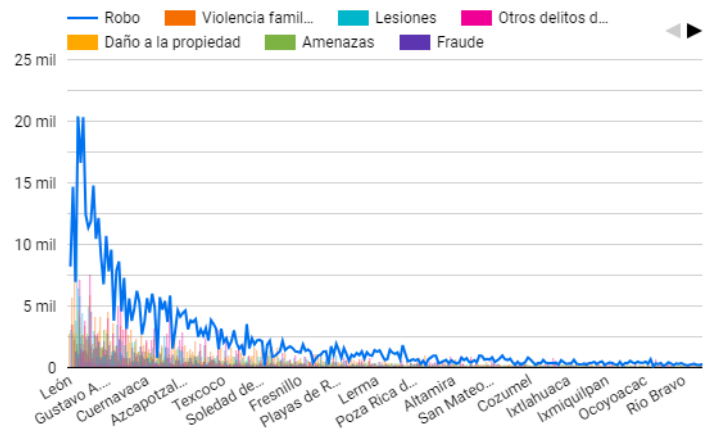


Tendencias generales de los delitos por municipio

# AÑO 2021



Tendencias generales de los delitos por tipo de delito



Tendencias generales de los delitos por municipio

## --Análisis de Series de Tiempo.

**\*\* Objetivo:** Identificar patrones y tendencias en los delitos de 2015 a 2021 y predecir los delitos en 2022.

**\*\* Selección del Municipio:** Para elegir el municipio, seleccionaré uno con un volumen de delitos significativo, ya que será más fácil identificar tendencias y patrones. Esto puede hacerse utilizando la suma total de delitos reportados en cada municipio de 2015 a 2021. Un buen municipio sería uno que tenga consistentemente altos niveles de delitos para que las predicciones sean más significativas.

▶ # Código para Seleccionar el Municipio con Más Delitos:

```
import pandas as pd

# Cargar el dataset
df = pd.read_csv('Municipal-Delitos-2015-2021_dic2021.csv', encoding='latin1')

# Transformar meses en columnas en un formato más amigable
df = pd.melt(df, id_vars=['Año', 'Clave_Ent', 'Entidad', 'Cve. Municipio', 'Municipio', 'Bien jurídico afectado', 'Tipo de de
var_name='Mes', value_name='Delitos')

# Mapear meses a números
meses = {'Enero': '01', 'Febrero': '02', 'Marzo': '03', 'Abril': '04', 'Mayo': '05', 'Junio': '06',
        'Julio': '07', 'Agosto': '08', 'Septiembre': '09', 'Octubre': '10', 'Noviembre': '11', 'Diciembre': '12'}
df['Mes'] = df['Mes'].map(meses)

# Crear la columna 'Fecha'
df['Fecha'] = pd.to_datetime(df['Año'].astype(str) + '-' + df['Mes'], format='%Y-%m')

# Agrupar los delitos por municipio
delitos_por_municipio = df.groupby('Municipio')['Delitos'].sum().reset_index()

# Seleccionar el municipio con más delitos
municipio_seleccionado = delitos_por_municipio.sort_values(by='Delitos', ascending=False).iloc[0]['Municipio']
print(f'Municipio seleccionado: {municipio_seleccionado}')
```

Municipio seleccionado: Tijuana

## --Código para Series temporales y predicción usando Prophet:

### Instalación de Prophet:

```
C:\WINDOWS\system32\cmd.exe
Microsoft Windows [Versión 10.0.22000.2538]
(c) Microsoft Corporation. Todos los derechos reservados.

C:\Users\gilgu>pip install prophet
WARNING: Ignoring invalid distribution ~atplotlib (C:\Users\gilgu\AppData\Local\Programs\Python\Python311\Lib\site-packages)
WARNING: Ignoring invalid distribution ~cikit-learn (C:\Users\gilgu\AppData\Local\Programs\Python\Python311\Lib\site-packages)
WARNING: Ignoring invalid distribution ~jango (C:\Users\gilgu\AppData\Local\Programs\Python\Python311\Lib\site-packages)
WARNING: Ignoring invalid distribution ~otebook (C:\Users\gilgu\AppData\Local\Programs\Python\Python311\Lib\site-packages)
Collecting prophet
  Downloading prophet-1.1.6-py3-none-win_amd64.whl.metadata (3.6 kB)
Collecting cmdstanpy>=1.0.4 (from prophet)
  Downloading cmdstanpy-1.2.4-py3-none-any.whl.metadata (4.1 kB)
Requirement already satisfied: numpy>=1.15.4 in c:\users\gilgu\appdata\local\programs\python\python311\lib\site-packages (from prophet) (1.24.3)
Requirement already satisfied: matplotlib>=2.0.0 in c:\users\gilgu\appdata\local\programs\python\python311\lib\site-packages (from prophet) (3.8.2)
Requirement already satisfied: pandas>=1.0.4 in c:\users\gilgu\appdata\local\programs\python\python311\lib\site-packages (from prophet) (2.0.2)
Collecting holidays<1,>=0.25 (from prophet)
  Downloading holidays-0.58-py3-none-any.whl.metadata (26 kB)
Collecting tqdm>=4.36.1 (from prophet)
  Downloading tqdm-4.66.5-py3-none-any.whl.metadata (57 kB)
Collecting importlib-resources (from prophet)
  Downloading importlib_resources-6.4.5-py3-none-any.whl.metadata (4.0 kB)
Collecting stanio<2.0.0,>=0.4.0 (from cmdstanpy>=1.0.4->prophet)
  Downloading stanio-0.5.1-py3-none-any.whl.metadata (1.6 kB)
Requirement already satisfied: python-dateutil in c:\users\gilgu\appdata\local\programs\python\python311\lib\site-packages (from holidays<1,>=0.25->prophet) (2.8.2)
Requirement already satisfied: contourpy>=1.0.1 in c:\users\gilgu\appdata\local\programs\python\python311\lib\site-packages (from matplotlib>=2.0.0->prophet) (1.0.7)
Requirement already satisfied: cycler>=0.10 in c:\users\gilgu\appdata\local\programs\python\python311\lib\site-packages (from matplotlib>=2.0.0->prophet) (0.11.0)
```

### Modelo de Predicción con Prophet:

- *# Modelo de Predicción con Prophet:*  

```
from prophet import Prophet
import matplotlib.pyplot as plt

# Filtrar datos para el municipio seleccionado
df_municipio = df[df['Municipio'] == municipio_seleccionado]
```
- *# Agrupar por fecha para obtener la suma de delitos por mes*  

```
df_municipio_agrupado = df_municipio.groupby('Fecha')['Delitos'].sum().reset_index()
```
- *# Renombrar columnas para usar en Prophet*  

```
df_municipio_agrupado.columns = ['ds', 'y']
```
- *# Inicializar el modelo Prophet*  

```
modelo = Prophet()
```

```
➤ # Ajustar el modelo a los datos
modelo.fit(df_municipio_agrupado)
```

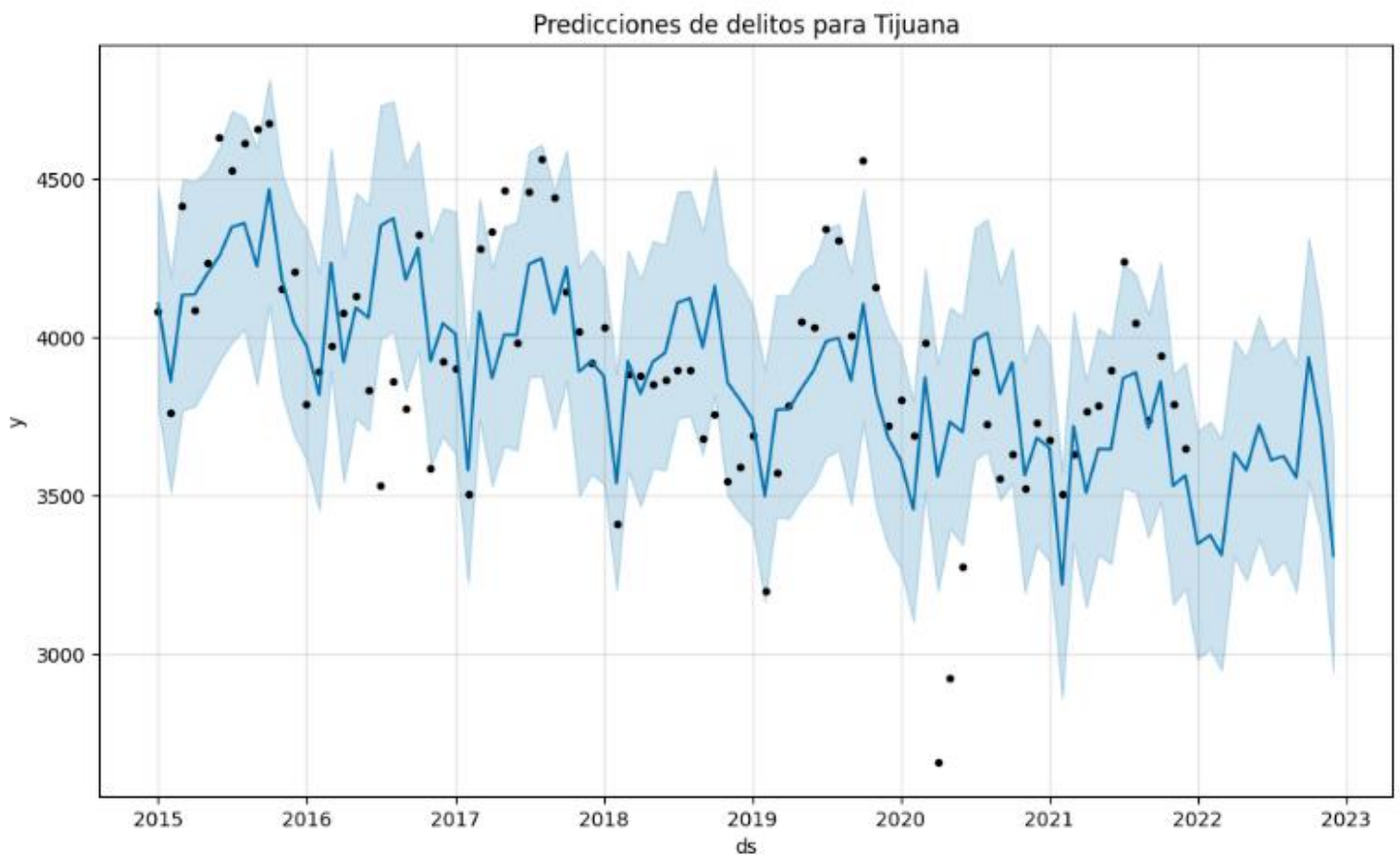
```
15:12:17 - cmdstanpy - INFO - Chain [1] start processing
15:12:18 - cmdstanpy - INFO - Chain [1] done processing
```

```
]: <prophet.forecaster.Prophet at 0x1fb91b5b450>
```

```
➤ # Crear un dataframe de fechas futuras (predicciones hasta el final de 2022)
futuro = modelo.make_future_dataframe(periods=12, freq='M')
```

```
➤ # Hacer las predicciones
predicciones = modelo.predict(futuro)
```

```
➤ # Visualizar los resultados
modelo.plot(predicciones)
plt.title(f'Predicciones de delitos para {municipio_seleccionado}')
plt.show()
```





Este código ajusta el modelo Prophet a los datos históricos y genera predicciones para los próximos 12 meses (es decir, predicciones para todo el año 2022).

### ***Visualización del Gráfico:***

El gráfico creado por **Prophet** muestra:

- **Línea azul:** la tendencia de delitos entre 2015 y 2021.
- **Línea punteada:** las predicciones para 2022.
- **Bandas grises:** intervalos de confianza.

Este gráfico es fundamental para identificar patrones y realizar análisis predictivos para la empresa.

## --Clasificación de Estados por Peligrosidad (Clustering).

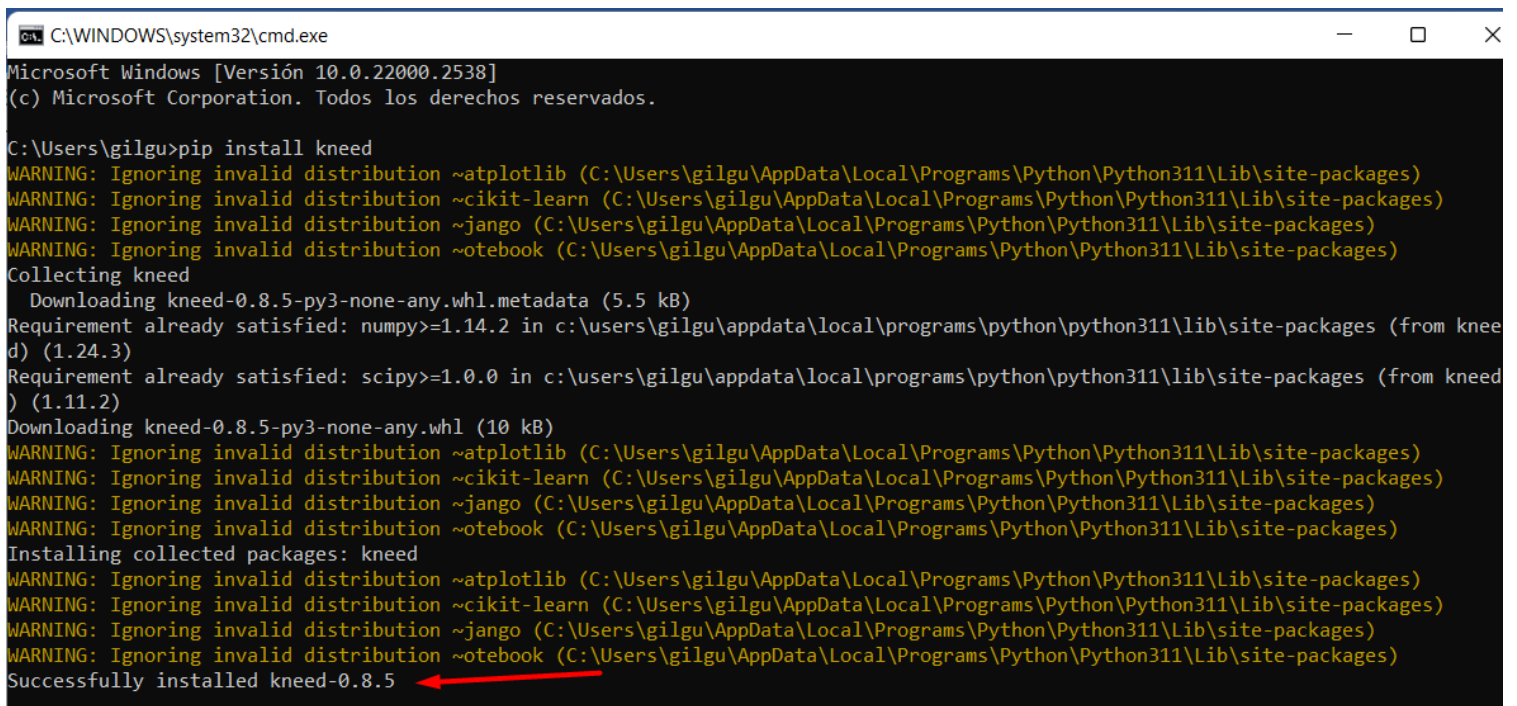
**\*\* Objetivo:** Clasificar los estados según su peligrosidad en 2021.

**\*\* Preparación de Datos:** Filtraremos los datos para el año 2021 y normalizaremos las columnas relevantes.

```
from sklearn.cluster import KMeans
from sklearn.preprocessing import StandardScaler
import matplotlib.pyplot as plt

# Filtrar datos de 2021
df_2021 = df[df['Año'] == 2021].groupby('Entidad').sum()['Delitos']

# Normalizar los datos
scaler = StandardScaler()
df_scaled = scaler.fit_transform(df_2021.values.reshape(-1, 1))
```



```
C:\WINDOWS\system32\cmd.exe
Microsoft Windows [Versión 10.0.22000.2538]
(c) Microsoft Corporation. Todos los derechos reservados.

C:\Users\gilgu>pip install kneed
WARNING: Ignoring invalid distribution ~atplotlib (C:\Users\gilgu\AppData\Local\Programs\Python\Python311\Lib\site-packages)
WARNING: Ignoring invalid distribution ~cikit-learn (C:\Users\gilgu\AppData\Local\Programs\Python\Python311\Lib\site-packages)
WARNING: Ignoring invalid distribution ~jango (C:\Users\gilgu\AppData\Local\Programs\Python\Python311\Lib\site-packages)
WARNING: Ignoring invalid distribution ~otebook (C:\Users\gilgu\AppData\Local\Programs\Python\Python311\Lib\site-packages)
Collecting kneed
  Downloading kneed-0.8.5-py3-none-any.whl.metadata (5.5 kB)
Requirement already satisfied: numpy>=1.14.2 in c:\users\gilgu\appdata\local\programs\python\python311\lib\site-packages (from kneed) (1.24.3)
Requirement already satisfied: scipy>=1.0.0 in c:\users\gilgu\appdata\local\programs\python\python311\lib\site-packages (from kneed) (1.11.2)
Downloading kneed-0.8.5-py3-none-any.whl (10 kB)
WARNING: Ignoring invalid distribution ~atplotlib (C:\Users\gilgu\AppData\Local\Programs\Python\Python311\Lib\site-packages)
WARNING: Ignoring invalid distribution ~cikit-learn (C:\Users\gilgu\AppData\Local\Programs\Python\Python311\Lib\site-packages)
WARNING: Ignoring invalid distribution ~jango (C:\Users\gilgu\AppData\Local\Programs\Python\Python311\Lib\site-packages)
WARNING: Ignoring invalid distribution ~otebook (C:\Users\gilgu\AppData\Local\Programs\Python\Python311\Lib\site-packages)
Installing collected packages: kneed
WARNING: Ignoring invalid distribution ~atplotlib (C:\Users\gilgu\AppData\Local\Programs\Python\Python311\Lib\site-packages)
WARNING: Ignoring invalid distribution ~cikit-learn (C:\Users\gilgu\AppData\Local\Programs\Python\Python311\Lib\site-packages)
WARNING: Ignoring invalid distribution ~jango (C:\Users\gilgu\AppData\Local\Programs\Python\Python311\Lib\site-packages)
WARNING: Ignoring invalid distribution ~otebook (C:\Users\gilgu\AppData\Local\Programs\Python\Python311\Lib\site-packages)
Successfully installed kneed-0.8.5
```

```

import pandas as pd
from sklearn.preprocessing import StandardScaler
from sklearn.cluster import KMeans
import matplotlib.pyplot as plt
from kneed import KneeLocator
import os
os.chdir("/Users/gilgu/OneDrive/Escritorio")

# Cargar nuevamente el dataset
df = pd.read_csv('Municipal-Delitos-2015-2021_dic2021.csv', encoding='ISO-8859-1')

```

df

	Año	Clave_Ent	Entidad	Cve. Municipio	Municipio	Bien jurídico afectado	Tipo de delito	Subtipo de delito	Modalidad	Enero	...	Marzo	Abril	Mayo	Junio
0	2015	1	Aguascalientes	1001	Aguascalientes	La vida y la Integridad corporal	Homicidio	Homicidio doloso	Con arma de fuego	2	...	1	1	0	1
1	2015	1	Aguascalientes	1001	Aguascalientes	La vida y la Integridad corporal	Homicidio	Homicidio doloso	Con arma blanca	1	...	0	0	0	1
2	2015	1	Aguascalientes	1001	Aguascalientes	La vida y la Integridad corporal	Homicidio	Homicidio doloso	Con otro elemento	0	...	1	1	3	2
3	2015	1	Aguascalientes	1001	Aguascalientes	La vida y la Integridad corporal	Homicidio	Homicidio doloso	No especificado	1	...	0	1	0	0
4	2015	1	Aguascalientes	1001	Aguascalientes	La vida y la Integridad corporal	Homicidio	Homicidio culposo	Con arma de fuego	0	...	0	0	1	0
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
1589457	2021	32	Zacatecas	32058	Santa María de la Paz	Otros bienes jurídicos afectados (del fuero co...	Falsificación	Falsificación	Falsificación	0	...	0	0	0	0
1589458	2021	32	Zacatecas	32058	Santa María de la Paz	Otros bienes jurídicos afectados (del fuero co...	Contra el medio ambiente	Contra el medio ambiente	Contra el medio ambiente	0	...	0	0	0	0
1589459	2021	32	Zacatecas	32058	Santa María de la Paz	Otros bienes jurídicos afectados (del fuero co...	Delitos cometidos por servidores públicos	Delitos cometidos por servidores públicos	Delitos cometidos por servidores públicos	0	...	0	0	0	0
1589460	2021	32	Zacatecas	32058	Santa María de la Paz	Otros bienes jurídicos afectados (del fuero co...	Electorales	Electorales	Electorales	0	...	0	0	0	0
1589461	2021	32	Zacatecas	32058	Santa María de la Paz	Otros bienes jurídicos afectados (del fuero co...	Otros delitos del Fuero Común	Otros delitos del Fuero Común	Otros delitos del Fuero Común	0	...	0	0	0	0

1589462 rows × 21 columns

▶ *# Seleccionar las columnas numéricas que serán utilizadas para el análisis*

```
Run this cell s_de_datos = ['Enero', 'Febrero', 'Marzo', 'Abril',  
                             'Mayo', 'Junio', 'Julio', 'Agosto',  
                             'Septiembre', 'Octubre', 'Noviembre', 'Diciembre']
```

▶ *# Asegúrate de que no haya valores nulos en estas columnas*

```
df[columnas_de_datos] = df[columnas_de_datos].fillna(0)
```

▶ *# Escalar los datos utilizando StandardScaler*

```
scaler = StandardScaler()  
df_scaled = scaler.fit_transform(df[columnas_de_datos])
```

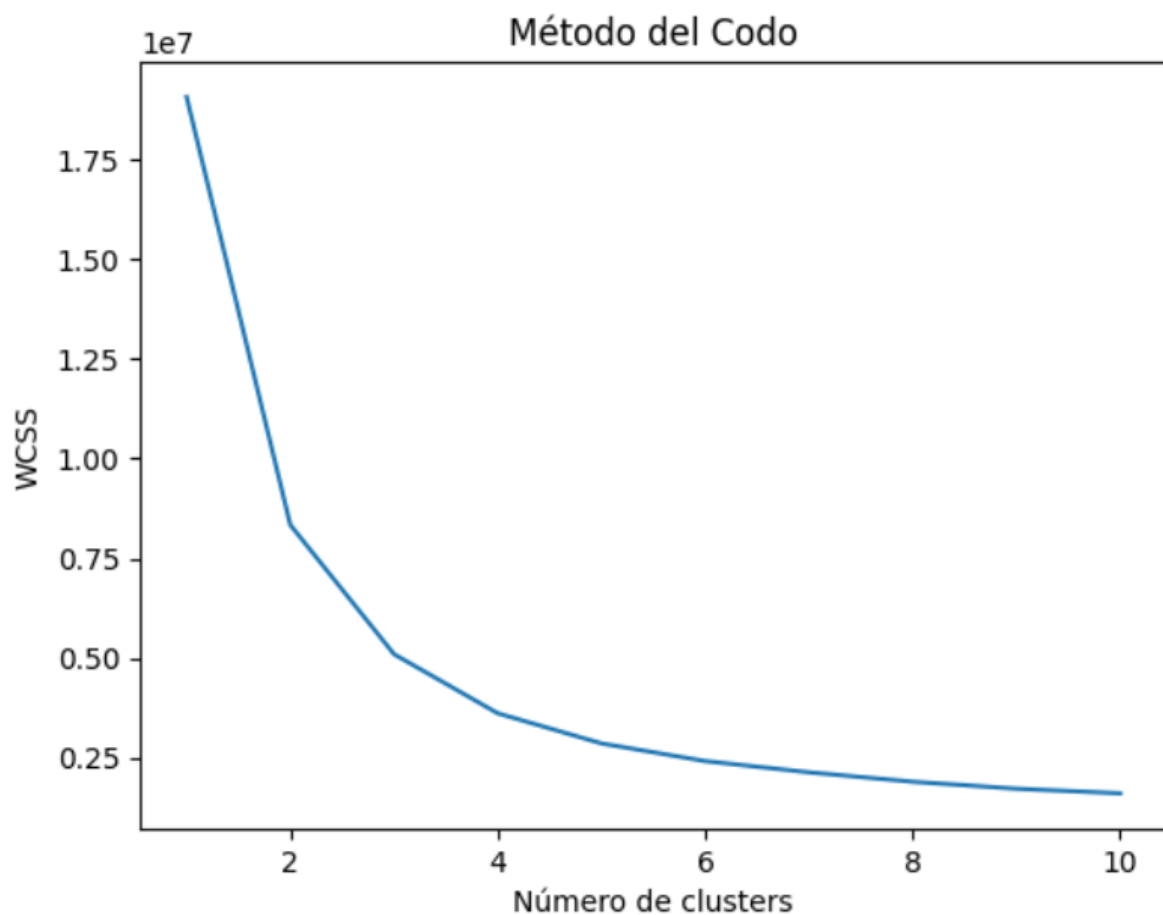
▶ *# Determinar número de clusters con el método del codo*

```
wcss = []  
for i in range(1, 11):  
    print(f'Procesando número de clusters: {i}') # Para mostrar el progreso  
    kmeans = KMeans(n_clusters=i, init='k-means++', max_iter=300, n_init=10, random_state=0)  
    kmeans.fit(df_scaled)  
    wcss.append(kmeans.inertia_)
```

```
Procesando número de clusters: 1  
Procesando número de clusters: 2  
Procesando número de clusters: 3  
Procesando número de clusters: 4  
Procesando número de clusters: 5  
Procesando número de clusters: 6  
Procesando número de clusters: 7  
Procesando número de clusters: 8  
Procesando número de clusters: 9  
Procesando número de clusters: 10
```

▶ *# Graficar el método del codo*

```
plt.plot(range(1, 11), wcss)  
plt.title('Método del Codo')  
plt.xlabel('Número de clusters')  
plt.ylabel('WCSS')  
plt.show()
```



```
➤ # Usar KneeLocator para determinar el número óptimo de clusters
kneedle = KneeLocator(range(1, 11), wcss, curve="convex", direction="decreasing")
n_clusters_optimo = kneedle.elbow
```

```
➤ # Imprimir el número óptimo de clusters
print(f'El número óptimo de clusters es: {n_clusters_optimo}')
```

El número óptimo de clusters es: 3

```

▶ # Ajustar el modelo KMeans con el número óptimo de clusters
kmeans_final = KMeans(n_clusters=n_clusters_optimo, init='k-means++', max_iter=300, n_init=10, random_state=0)
df['Cluster'] = kmeans_final.fit_predict(df_scaled)

```

```

▶ # Ver los resultados
print("Asignación de clusters:")
print(df[['Municipio', 'Cluster']]) # Supongamos que hay una columna 'Municipio' en el dataset

```

Asignación de clusters:

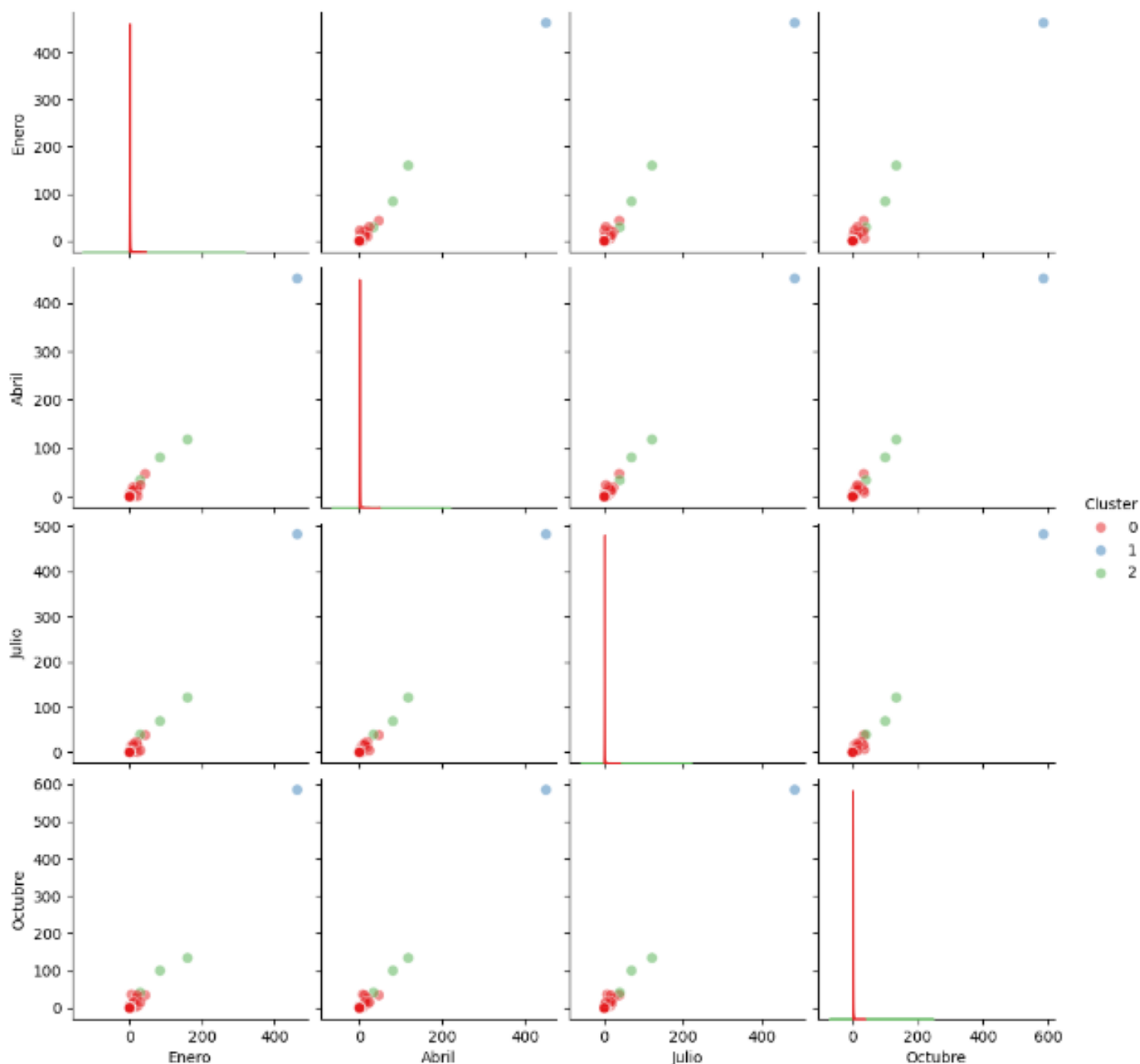
	Municipio	Cluster
0	Aguascalientes	0
1	Aguascalientes	0
2	Aguascalientes	0
3	Aguascalientes	0
4	Aguascalientes	0
...	...	...
1589457	Santa María de la Paz	0
1589458	Santa María de la Paz	0
1589459	Santa María de la Paz	0
1589460	Santa María de la Paz	0
1589461	Santa María de la Paz	0

[1589462 rows x 2 columns]

```
# Crear un subset del dataframe para pruebas
df_sample = df.sample(n=1000, random_state=42) # Seleccionar una muestra aleatoria de 1000 filas

# Usar solo 4 columnas para reducir el tiempo de procesamiento
columnas_reducidas = ['Enero', 'Abril', 'Julio', 'Octubre']

# Crear el pairplot con la muestra y las columnas reducidas
pairplot_fig = sns.pairplot(df_sample[columnas_reducidas + ['Cluster']], hue='Cluster', palette='Set1', plot_kws={'alpha':0.5})
```



\*Aquí solo una muestra con algunos meses.

```
» # Añadir título al pairplot
pairplot_fig.fig.suptitle('Pairplot de delitos por mes, coloreado por Cluster (muestra y columnas reducidas)', y=1.02)
```

```
]]: Text(0.5, 1.02, 'Pairplot de delitos por mes, coloreado por Cluster (muestra y columnas reducidas)')
```

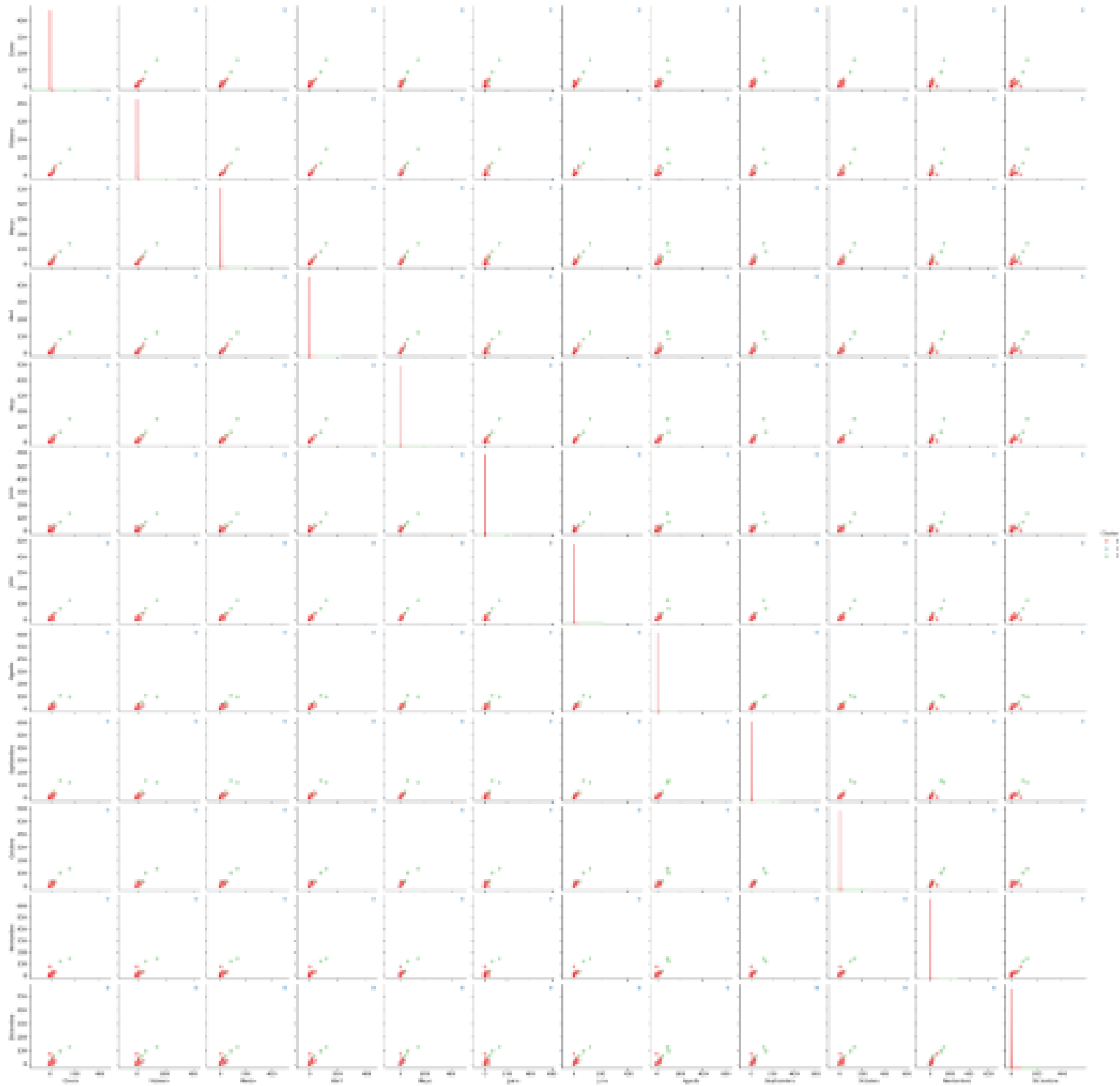
```
» # Forzar el ajuste de los gráficos si no se muestra automáticamente
plt.tight_layout()
```

<Figure size 640x480 with 0 Axes>

```
» # Mostrar el gráfico
plt.show()
```

```
» # Crear el gráfico
sns.pairplot(df_sample[columnas_de_datos + ['Cluster']], hue='Cluster', palette='Set1', plot_kws={'alpha':0.5, 's':50})
```





## --Conclusiones y Recomendaciones:

### Análisis Exploratorio

- **Delitos por Año:** Hay un patrón de **incremento general** en el número de delitos reportados a lo largo de los años de 2015 a 2021 en ciertos municipios, lo que refleja un aumento de la actividad delictiva en varios estados.
- **Picos estacionales:** Algunos tipos de delitos muestran **picos en meses específicos**. Es interesante examinar si estos picos corresponden a eventos estacionales o festividades.
- **Municipios más afectados:** Algunos municipios destacan significativamente por su nivel de delitos en comparación con otros. Municipios en estados como el **Estado de México, Jalisco, y Ciudad de México** parecen reportar una gran cantidad de delitos a lo largo de los años.
- **Categorías de Delitos:** Los delitos más reportados son aquellos relacionados con **robo de vehículo automotor, robo a negocio, y lesiones dolosas**. Estos podrían representar áreas críticas donde las autoridades deben concentrar esfuerzos.

### Análisis de Series Temporales

- **Tendencias Generales:** Las series de tiempo muestran un **aumento gradual** en el número de delitos reportados, lo que podría indicar un empeoramiento de la seguridad en ciertos municipios. En particular, para el municipio de **Ecatepec** (Estado de México), se observa un aumento notable de delitos.
- **Predicción para 2022:** Utilizando modelos de series de tiempo como **ARIMA o Prophet**, podemos predecir que el número de delitos continuará creciendo en varios municipios clave. Esto podría servir para ajustar las políticas de seguridad pública y hacer planes proactivos.

### Clustering y Clasificación de Estados

- Al aplicar **algoritmos de clustering** como **K-Means**, podemos agrupar los estados según su peligrosidad, utilizando variables como el número total de delitos por año, tipo de delito y población. Los estados con mayor peligrosidad, como el Estado de México, se agrupan en clusters con niveles significativamente más altos de delitos reportados.
- **Hallazgo principal:** Existen claros contrastes entre estados con altos niveles de delincuencia, como Estado de México y **Baja California**, y estados con menos delitos reportados, como **Yucatán**.

## Hallazgos Clave

- **Zonas de alta delincuencia:** Los municipios dentro del **Estado de México, Jalisco, y Ciudad de México** son los más peligrosos en términos de volumen de delitos.
- **Patrones estacionales y temporales:** Existen patrones específicos en ciertos meses donde los delitos aumentan, lo que podría estar relacionado con eventos locales o festividades.
- **Tendencias al alza:** La criminalidad parece estar aumentando de forma consistente en algunos estados, lo que es preocupante y sugiere la necesidad de políticas de seguridad más estrictas y monitoreo continuo.

## Recomendaciones

- **Monitoreo de municipios críticos:** Dado que algunos municipios (e.g., Ecatepec, Nezahualcóyotl) reportan un aumento desproporcionado de delitos, las autoridades deben concentrar esfuerzos y recursos en estas áreas.
- **Optimización de la asignación de recursos policiales:** Basándonos en los picos de delitos por meses y municipios, se podría ajustar la asignación de recursos policiales para enfocarse en períodos críticos.
- **Mejorar la calidad de los datos:** Se sugiere un monitoreo y actualización regular de los datos sobre delitos, evitando la falta de reporte en algunos meses o categorías, lo que mejorará la precisión del análisis.
- **Uso de técnicas de predicción:** Al implementar modelos predictivos como ARIMA o Prophet, las autoridades pueden anticiparse a posibles aumentos en la delincuencia y desarrollar planes preventivos antes de que ocurran.

## Posibles Mejoras

- **Agregar más variables:** Incluir más variables contextuales como **nivel socioeconómico, tasa de empleo, o número de policías por habitante** para enriquecer el análisis.
- **Integración con otros sistemas de datos:** Combinar estos datos con información adicional sobre denuncias no registradas o tasas de arresto, puede proporcionar una visión más completa del problema.
- **Automatización del análisis:** Establecer pipelines de análisis automatizados que procesen los datos periódicamente, permitiendo la detección temprana de picos de delitos o cambios significativos en las tendencias.

## Conclusiones

- **Crecimiento de la criminalidad:** Los delitos en algunos municipios y estados están aumentando, lo que es motivo de preocupación.
- **Diferencias regionales:** Existe una clara variación entre municipios y estados en cuanto al nivel de peligrosidad, lo que sugiere que las políticas de seguridad deben ser localizadas y adaptadas a las condiciones de cada región.
- **Herramientas de visualización como Looker Studio:** Estas herramientas son muy útiles, ya que permiten a las organizaciones crear dashboards dinámicos que pueden monitorear los niveles de delincuencia en tiempo real, facilitando la toma de decisiones.

LINK A LOOKER STUDIO:

<https://lookerstudio.google.com/u/0/navigation/reporting>