

# Proyecto final

## Profesión Ciencia de Datos

### Caso: Seguros de Automóviles

---

PAULO ANDRÉS OSSA BENITEZ

FEBRERO 15 DE 2025



# Introducción: Contexto del proyecto

---

Las compañías de seguros de automóviles enfrentan el reto constante de calcular tarifas justas y competitivas para sus clientes, mientras garantizan la rentabilidad del negocio. Uno de los factores clave en la determinación del precio de las pólizas es la incidencia de delitos relacionados con automóviles en cada municipio.

Este análisis tiene como objetivo identificar patrones de criminalidad en distintas regiones de México, proporcionando una base de datos confiable para evaluar el nivel de riesgo en cada zona. Al comprender la relación entre la delincuencia y la siniestralidad vehicular, la aseguradora podrá diseñar estrategias de fijación de precios más precisas, optimizando la oferta de seguros para distintos perfiles de clientes.

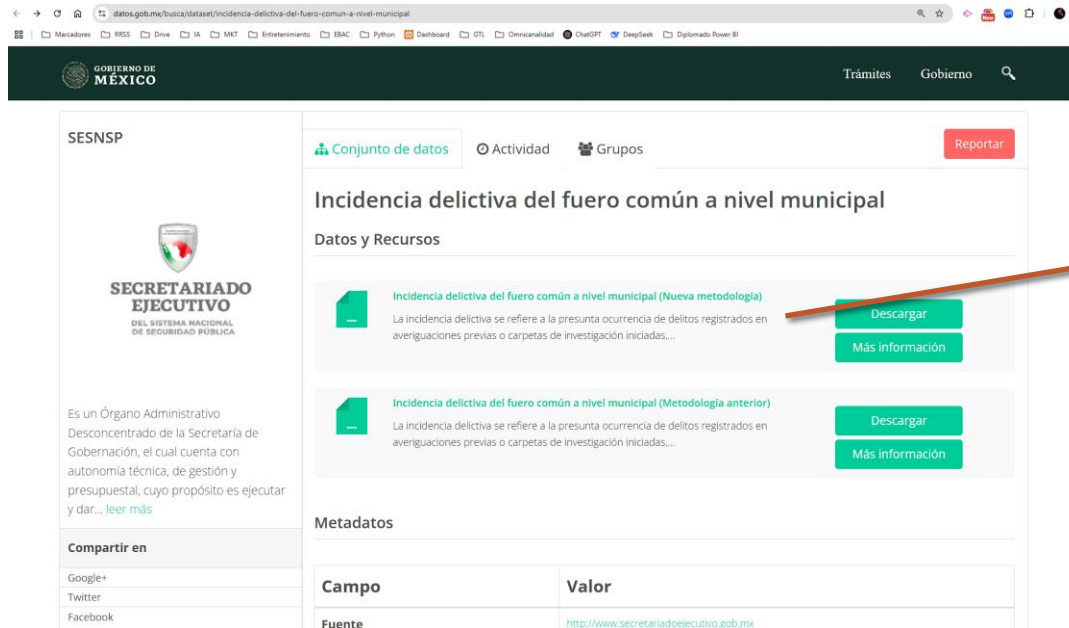
Además de la tarificación de pólizas, este estudio permitirá a la empresa identificar tendencias y prever riesgos emergentes, lo que contribuirá a mejorar la toma de decisiones y el desarrollo de estrategias de mitigación de pérdidas.



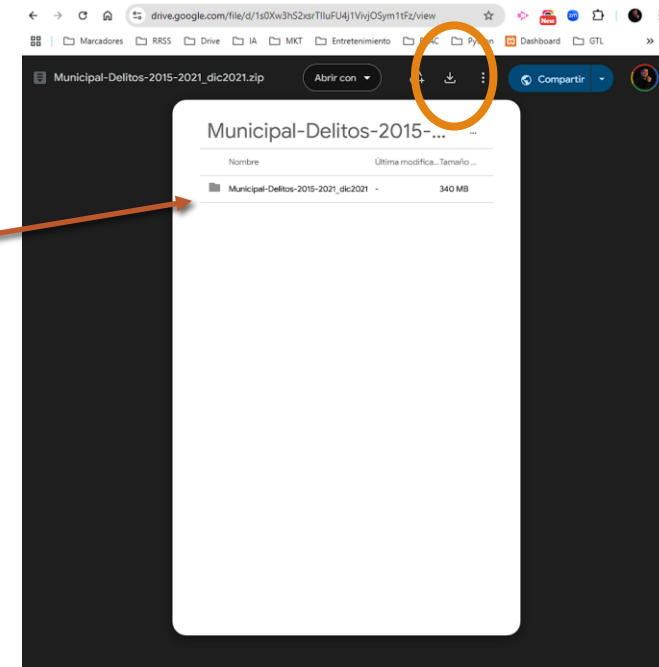
# Descarga y creación de la base de datos

---

# Descarga de los datos

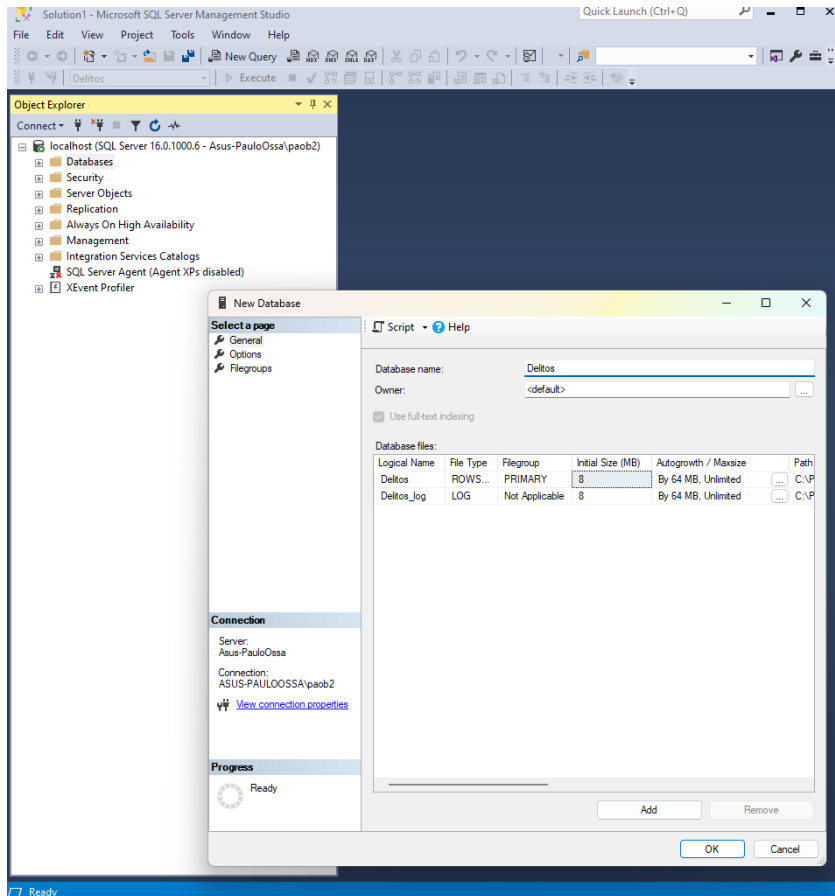


Del sitio oficial en la url: <https://datos.gob.mx/busca/dataset/incidencia-delictiva-del-fuero-comun-a-nivel-municipal> se hace la descarga del archivo con toda la data, este botón nos lleva a un Google Drive.



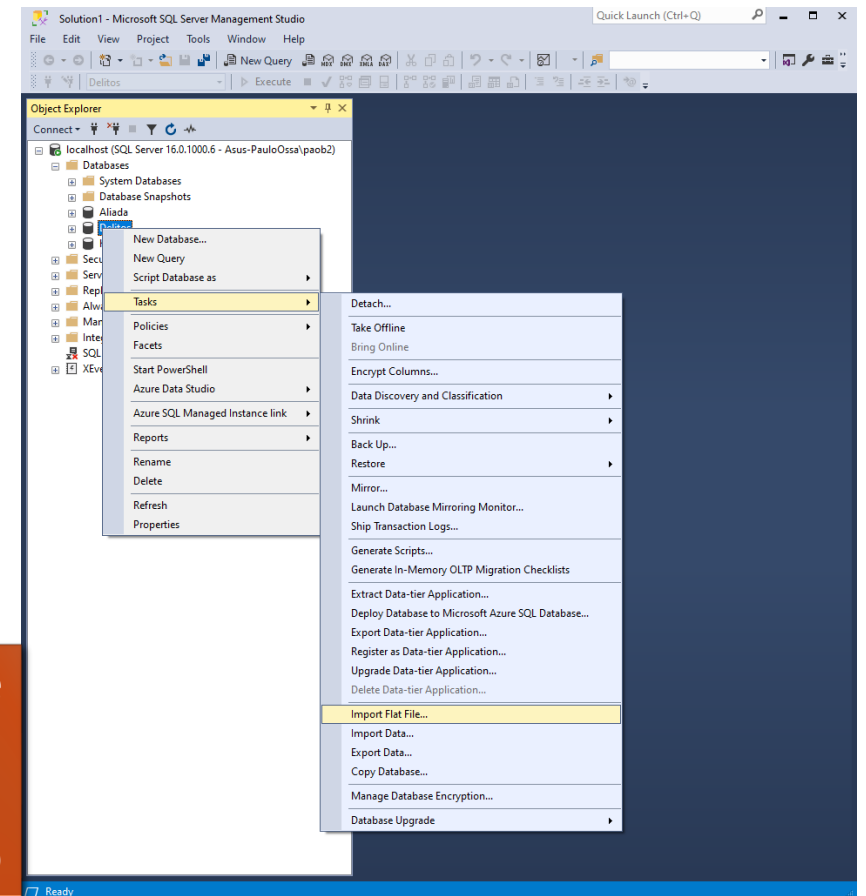
Desde este sitio se hace clic en el botón de descargar y se guarda el archivo en una carpeta local en la computadora.

# Creación de la base de datos



En el SSMS realizamos la creación de la base de datos, en esta caso la llamaremos “Delitos”

Luego de crear al base de datos hacemos la importación del archivo



# Visualización de datos en la BD

The screenshot displays the Microsoft SQL Server Management Studio interface. On the left, the Object Explorer shows the database structure, including tables and columns. The central pane shows a SQL query executed in the SQLQuery2.sql file. The query is a SELECT statement with a TOP (1000) clause, listing various columns from the [Delitos].[dbo].[delitos] table. The results pane at the bottom shows the output of the query, which is a table with 14 columns: Año, Clave\_Ent, Entidad, Cve\_Municipio, Municipio, Bien\_juridico\_afectado, Tipo\_de\_delito, Subtipo\_de\_delito, Modalidad, and 12 months (Enero through Agosto). The data shows various crimes reported in 2016, primarily in Michoacán de Ocampo, with the most common crime being Robo (Theft).

```
1 SELECT TOP (1000) [Año]
2 , [Clave_Ent]
3 , [Entidad]
4 , [Cve_Municipio]
5 , [Municipio]
6 , [Bien_juridico_afectado]
7 , [Tipo_de_delito]
8 , [Subtipo_de_delito]
9 , [Modalidad]
10 , [Enero]
11 , [Febrero]
12 , [Marzo]
13 , [Abril]
14 , [Mayo]
15 , [Junio]
16 , [Julio]
17 , [Agosto]
18 , [Septiembre]
19 , [Octubre]
20 , [Noviembre]
21 , [Diciembre]
22 FROM [Delitos].[dbo].[delitos]
23
```

Año	Clave_Ent	Entidad	Cve_Municipio	Municipio	Bien_juridico_afectado	Tipo_de_delito	Subtipo_de_delito	Modalidad	Enero	Febrero	Marzo	Abril	Mayo	Junio	Julio	Agosto
2016	16	Michoacán de Ocampo	16033	Gabriel Zamora	El patrimonio	Robo	Robo a casa habitación	Sin violencia	0	0	0	0	0	0	0	1
2016	16	Michoacán de Ocampo	16033	Gabriel Zamora	El patrimonio	Robo	Robo de vehículo automotor	Robo de coche de 4 ruedas Con violencia	2	0	3	2	2	8	3	5
2016	16	Michoacán de Ocampo	16033	Gabriel Zamora	El patrimonio	Robo	Robo de vehículo automotor	Robo de coche de 4 ruedas Sin violencia	2	0	1	2	0	0	3	0
2016	16	Michoacán de Ocampo	16033	Gabriel Zamora	El patrimonio	Robo	Robo de vehículo automotor	Robo de motocicleta Con violencia	0	1	0	0	0	0	1	0
2016	16	Michoacán de Ocampo	16033	Gabriel Zamora	El patrimonio	Robo	Robo de vehículo automotor	Robo de motocicleta Sin violencia	0	0	0	0	0	0	0	1
2016	16	Michoacán de Ocampo	16033	Gabriel Zamora	El patrimonio	Robo	Robo de vehículo automotor	Robo de embarcaciones pequeñas y grandes Sin viol...	0	0	0	0	0	0	0	0
2016	16	Michoacán de Ocampo	16033	Gabriel Zamora	El patrimonio	Robo	Robo de vehículo automotor	Robo de embarcaciones pequeñas y grandes Sin viol...	0	0	0	0	0	0	0	0
2016	16	Michoacán de Ocampo	16033	Gabriel Zamora	El patrimonio	Robo	Robo de autopartes	Con violencia	0	0	0	0	0	0	0	0
2016	16	Michoacán de Ocampo	16033	Gabriel Zamora	El patrimonio	Robo	Robo de autopartes	Sin violencia	0	0	0	0	0	0	0	0
2016	16	Michoacán de Ocampo	16033	Gabriel Zamora	El patrimonio	Robo	Robo a transportista	Con violencia	0	0	2	0	0	0	0	1
2016	16	Michoacán de Ocampo	16033	Gabriel Zamora	El patrimonio	Robo	Robo a transportista	Sin violencia	0	0	0	0	0	0	0	0
2016	16	Michoacán de Ocampo	16033	Gabriel Zamora	El patrimonio	Robo	Robo a transeúnte en vía pública	Con violencia	0	0	0	0	0	0	0	0
2016	16	Michoacán de Ocampo	16033	Gabriel Zamora	El patrimonio	Robo	Robo a transeúnte en espacio público	Sin violencia	0	0	0	0	0	0	0	0
2016	16	Michoacán de Ocampo	16033	Gabriel Zamora	El patrimonio	Robo	Robo a transeúnte en espacio abierto al público	Con violencia	0	0	0	0	0	0	0	1
2016	16	Michoacán de Ocampo	16033	Gabriel Zamora	El patrimonio	Robo	Robo a transeúnte en espacio abierto al público	Sin violencia	0	0	0	0	0	0	0	0
2016	16	Michoacán de Ocampo	16033	Gabriel Zamora	El patrimonio	Robo	Robo en transporte público individual	Con violencia	0	0	0	0	0	0	0	0
2016	16	Michoacán de Ocampo	16033	Gabriel Zamora	El patrimonio	Robo	Robo en transporte público individual	Con violencia	0	0	0	0	0	0	0	0
2016	16	Michoacán de Ocampo	16033	Gabriel Zamora	El patrimonio	Robo	Robo en transporte público colectivo	Con violencia	0	0	0	0	0	0	0	0
2016	16	Michoacán de Ocampo	16033	Gabriel Zamora	El patrimonio	Robo	Robo en transporte público colectivo	Sin violencia	0	0	0	0	0	0	0	0

Finalmente  
realizamos una  
consulta para  
visualizar los datos  
y que todo haya  
subido  
correctamente





# Conexión a la base de datos desde Python

---

```
+ Code + Markdown | ▶ Run All ⏹ Restart ⌵ Clear All Outputs | [x] Jupyter Variables ⌵ Outline ... base (Python 3.12.3)

import pyodbc
import pandas as pd

# Configuración de la conexión
server = 'localhost' # Nombre o dirección del servidor SQL Server
database = 'Delitos' # Nombre de tu base de datos
username = 'paob2' # Usuario de SQL Server
password = 'Paulo2024$' # Contraseña del usuario
driver = '{ODBC Driver 17 for SQL Server}' # Asegúrate de tener el driver adecuado instalado

# Conexión a la base de datos
try:

    ##connection = pyodbc.connect(
    ##    f'DRIVER={driver};SERVER={server};DATABASE={database};UID={username};PWD={password}'
    ##)

    connection = pyodbc.connect(
    f'DRIVER={driver};SERVER={server};DATABASE={database};Trusted_Connection=yes'
    )

    print("Conexión exitosa a la base de datos.")

    # Creación de un cursor
    cursor = connection.cursor()

    # Ejecución de una consulta
    query = "SELECT * FROM delitos" # Reemplaza 'TuTabla' con el nombre de tu tabla

    # Leer los datos en un DataFrame de Pandas
    df = pd.read_sql_query(query, connection)

except pyodbc.Error as e:
    print("Error al conectar a la base de datos:", e)
```

# Conexión a la base de datos desde Python

Los pasos en Python para hacer la conexión básicamente son los siguientes:

1. Usar la librería pyodbc para hacer la conexión, se debe instalar en caso de que antes no se haya hecho.
2. Definir los datos de conexión.
3. Bloque con un “try” en donde se crea la conexión y un cursor.
4. Finalmente se hace un query y de manera más practica se lleva a un dataframe con pandas.



# Visualización de datos con Pandas

```
df.sample(20)
```

[2]

Conexión exitosa a la base de datos.  
C:\Users\paob2\Anaconda3\envs\python38\python.exe: UserWarning: pandas only supports SQLAlchemy connectable (engine/connection) or database string URI or sqlite3 DBAPI2 connection. Other DBAPI2 objects are not tested. Please consider using SQLAlchemy.  
df = pd.read\_sql\_query(query, connection)  
Conexión cerrada.

Año	Clave_Ent	Entidad	Cve_Municipio	Municipio	Bien jurídico afectado	Tipo de delito	Subtipo de delito	Modalidad	Enero	...	Marzo	Abril	Mayo	Junio	Julio	Agosto	Septiembre	Octubre	Noviembre	Diciembre
312628	2017	29	29047	Lázaro Cárdenas	El patrimonio	Robo	Robo de autopartes	Sin violencia	0	...	0	0	0	0	0	0	0	0	0	0
1209282	2020	20	20065	Ixpantepec Nieves	El patrimonio	Robo	Robo en transporte individual	Sin violencia	0	...	0	0	0	0	0	0	0	0	0	0
809235	2018	26	26031	Huachinera	El patrimonio	Robo	Robo a transportista	Con violencia	0	...	0	0	0	0	0	0	0	0	0	0
145601	2017	12	12004	Alcozauca de Guerrero	La vida y la integridad corporal	Lesiones	Lesiones culposas	Con arma de fuego	0	...	0	0	0	0	0	0	0	0	0	0
1152406	2020	13	13037	Metztlitlán	Libertad personal	Secuestro	Secuestro	Secuestro extorsivo	0	...	0	0	0	0	0	0	0	0	0	0
1162917	2020	14	14060	Mexicacán	El patrimonio	Robo	Robo a transportista	Con violencia	0	...	0	0	0	0	0	0	0	0	0	0
120952	2017	7	7039	Huitiupán	El patrimonio	Robo	Robo en transporte individual	Sin violencia	0	...	0	0	0	0	0	0	0	0	0	0
1324174	2020	30	30149	Soteapan	Otros bienes jurídicos afectados (del fuero co...	Electorales	Electorales	Electorales	0	...	0	0	0	0	0	0	0	0	0	0
1370910	2021	8	8041	Maguarichi	La sociedad	Trata de personas	Trata de personas	Trata de personas	0	...	0	0	0	0	0	0	0	0	0	0
1345713	2020	32	32049	Valparaíso	El patrimonio	Fraude	Fraude	Fraude	1	...	1	0	0	5	1	1	1	0	2	0
798046	2018	23	23004	Othón P. Blanco	La libertad y la seguridad sexual	Abuso sexual	Abuso sexual	Abuso sexual	2	...	5	6	5	5	4	2	1	5	0	1
814343	2018	27	27009	Jalapa	El patrimonio	Robo	Robo a institución bancaria	Con violencia	0	...	0	0	0	0	0	0	0	0	0	0
1570611	2021	30	30186	Tomatlán	El patrimonio	Robo	Robo a negocio	Con violencia	0	...	0	0	0	0	0	0	0	1	0	0
1476267	2021	20	20311	San Pedro Jaltepetongo	Otros bienes jurídicos afectados (del fuero co...	Falsificación	Falsificación	Falsificación	0	...	0	0	0	0	0	0	0	0	0	0
1060040	2019	28	28028	Nuevo Morelos	El patrimonio	Robo	Robo de maquinaria	Robo de cables, tubos y otros objetos destinad...	0	...	0	0	0	0	0	0	0	0	0	0
1279841	2020	21	21214	Zinacatepec	El patrimonio	Robo	Robo en transporte individual	Con violencia	0	...	0	0	0	0	0	0	0	0	0	0
513880	2015	13	13017	Chapantongo	La vida y la integridad corporal	Homicidio	Homicidio culposo	En accidente de tránsito	0	...	0	0	0	0	0	0	0	0	0	0
273515	2017	21	21105	Ocoatepec	La libertad y la seguridad sexual	Incesto	Incesto	Incesto	0	...	0	0	0	0	0	0	0	0	0	0
134344	2017	8	8058	San Francisco de Conchos	Libertad personal	Secuestro	Secuestro	Secuestro para causar daño	0	...	0	0	0	0	0	0	0	0	0	0
1214176	2020	20	20115	San Bartolo Coyotepec	El patrimonio	Robo	Robo a transeúnte en espacio abierto al público	Sin violencia	0	...	0	0	0	0	0	0	0	0	0	0

20 rows x 21 columns

Luego con un sample verificamos que se haya ejecutado bien el proceso de conexión y carga de los datos del query.



# Limpieza y transformación de datos

---

# Información general del dataset

Esta información nos confirma que es un dataset completo, sin nulos en ningún campo, y que tenemos 1589462 registros.

Adicionalmente vemos que la mayoría de los datos tiene un tipo “object”, por lo que este es uno de los principales aspectos a transformar.

```
df.info()
```

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

```
RangeIndex: 1589462 entries, 0 to 1589461
```

```
Data columns (total 21 columns):
```

#	Column	Non-Null Count	Dtype
0	Año	1589462 non-null	int64
1	Clave_Ent	1589462 non-null	int64
2	Entidad	1589462 non-null	object
3	Cve_Municipio	1589462 non-null	int64
4	Municipio	1589462 non-null	object
5	Bien_jurídico_afectado	1589462 non-null	object
6	Tipo_de_delito	1589462 non-null	object
7	Subtipo_de_delito	1589462 non-null	object
8	Modalidad	1589462 non-null	object
9	Enero	1589462 non-null	object
10	Febrero	1589462 non-null	object
11	Marzo	1589462 non-null	object
12	Abril	1589462 non-null	object
13	Mayo	1589462 non-null	object
14	Junio	1589462 non-null	object
15	Julio	1589462 non-null	object
16	Agosto	1589462 non-null	object
17	Septiembre	1589462 non-null	object
18	Octubre	1589462 non-null	object
19	Noviembre	1589462 non-null	object
20	Diciembre	1589462 non-null	object

```
dtypes: int64(3), object(18)
```

```
memory usage: 254.7+ MB
```

```

df['Entidad'] = df['Entidad'].astype("string")
df['Municipio'] = df['Municipio'].astype("string")
df['Bien_jurídico_afectado'] = df['Bien_jurídico_afectado'].astype("string")
df['Tipo_de_delito'] = df['Tipo_de_delito'].astype("string")
df['Subtipo_de_delito'] = df['Subtipo_de_delito'].astype("string")
df['Modalidad'] = df['Modalidad'].astype("string")
df['Enero'] = df['Enero'].astype("int64")
df['Febrero'] = df['Febrero'].astype("int64")
df['Marzo'] = df['Marzo'].astype("int64")
df['Abril'] = df['Abril'].astype("int64")
df['Mayo'] = df['Mayo'].astype("int64")
df['Junio'] = df['Junio'].astype("int64")
df['Julio'] = df['Julio'].astype("int64")
df['Agosto'] = df['Agosto'].astype("int64")
df['Septiembre'] = df['Septiembre'].astype("int64")
df['Octubre'] = df['Octubre'].astype("int64")
df['Noviembre'] = df['Noviembre'].astype("int64")
df['Diciembre'] = df['Diciembre'].astype("int64")

df.info()

```

RangeIndex: 1589462 entries, 0 to 1589461

Data columns (total 21 columns):

#	Column	Non-Null Count	Dtype
0	Año	1589462 non-null	int64
1	Clave_Ent	1589462 non-null	int64
2	Entidad	1589462 non-null	string
3	Cve_Municipio	1589462 non-null	int64
4	Municipio	1589462 non-null	string
5	Bien_jurídico_afectado	1589462 non-null	string
6	Tipo_de_delito	1589462 non-null	string
7	Subtipo_de_delito	1589462 non-null	string
8	Modalidad	1589462 non-null	string
9	Enero	1589462 non-null	int64
10	Febrero	1589462 non-null	int64
11	Marzo	1589462 non-null	int64
12	Abril	1589462 non-null	int64
13	Mayo	1589462 non-null	int64
14	Junio	1589462 non-null	int64
15	Julio	1589462 non-null	int64
16	Agosto	1589462 non-null	int64
17	Septiembre	1589462 non-null	int64
18	Octubre	1589462 non-null	int64
19	Noviembre	1589462 non-null	int64
20	Diciembre	1589462 non-null	int64

dtypes: int64(15), string(6)

memory usage: 254.7 MB

## Tranformación de tipos de datos

# Datos únicos por variable

Esta información nos permite visualizar los campos candidatos a ser categorías como lo son:

- Año
- Entidad
- Bien jurídico afectado
- Tipo de delito
- Subtipo de delito
- Modalidad

```
df.nunique()
```

Año	7
Clave_Ent	32
Entidad	32
Cve_Municipio	2478
Municipio	2331
Bien_jurídico_afectado	7
Tipo_de_delito	40
Subtipo_de_delito	55
Modalidad	59
Enero	409
Febrero	418
Marzo	446
Abril	437
Mayo	458
Junio	446
Julio	446
Agosto	456
Septiembre	445
Octubre	459
Noviembre	432
Diciembre	424
dtype:	int64





# Análisis Bivariable

---

# Delitos por frecuencia y su porcentaje

```
# Delitos totales en orden de frecuencia y porcentaje frente al total

import pandas as pd

# Paso 1: Agrupar los datos por 'Tipo_de_delito' y calcular el total de delitos
delitos_totales = (
    df_melted.groupby('Tipo_de_delito')['Delitos']
    .sum() # Calcular el total de delitos por tipo de delito
    .reset_index() # Reiniciar el índice
)

# Paso 2: Ordenar los resultados de mayor a menor
delitos_totales = delitos_totales.sort_values(by='Delitos', ascending=False)

# Paso 1: Calcular el total global de delitos
total_delitos = delitos_totales['Delitos'].sum()

# Paso 2: Agregar una columna de porcentaje
delitos_totales['Porcentaje'] = (delitos_totales['Delitos'] / total_delitos) * 100

# Mostrar los resultados
delitos_totales
```

	Tipo_de_delito	Delitos	Porcentaje
32	Robo	4933311	37.077131
22	Lesiones	1401344	10.532037
39	Violencia familiar	1315041	9.883412
27	Otros delitos del Fuero Común	1249120	9.387972
8	Daño a la propiedad	895345	6.729116
5	Amenazas	650832	4.891438
17	Fraude	493028	3.705435
23	Narcomenudeo	395477	2.972274
18	Homicidio	284638	2.139245
10	Despojo	183328	1.377833
1	Abuso de confianza	172582	1.297069
21	Incumplimiento de obligaciones de asistencia f...	163909	1.231886
2	Abuso sexual	135592	1.019065
9	Delitos cometidos por servidores públicos	123704	0.929718
28	Otros delitos que atentan contra la libertad p...	119392	0.897311
15	Falsificación	117240	0.881137
4	Allanamiento de morada	92230	0.693170
37	Violación simple	85874	0.645401
25	Otros delitos contra la familia	80021	0.601411
24	Otros delitos contra el patrimonio	69810	0.524669
30	Otros delitos que atentan contra la vida y la ...	51052	0.383690
13	Extorsión	50125	0.376723
29	Otros delitos que atentan contra la libertad y...	44663	0.335672
26	Otros delitos contra la sociedad	39070	0.293637
36	Violación equiparada	24201	0.181887
3	Acoso sexual	23721	0.178279
14	Falsedad	21399	0.160828
38	Violencia de género en todas sus modalidades d...	19351	0.145436

# Tabla de frecuencia y porcentaje para tipos de robo

```
# Tabla de frecuencia y % de ocurrencia para los tipos de robos

# Paso 1: Filtrar los datos por Tipo_de_delito = "Robo"
df_robos = df_melted[df_melted['Tipo_de_delito'] == 'Robo']

# Paso 2: Agrupar por Subtipo_de_delito y calcular la frecuencia
frecuencia_robos = (
    df_robos.groupby('Subtipo_de_delito')['Delitos']
    .sum() # Sumar el número total de delitos por subtipo
    .reset_index() # Reiniciar el índice
)

# Paso 3: Calcular el total global de robos
total_robos = frecuencia_robos['Delitos'].sum()

# Paso 4: Agregar la columna de porcentaje
frecuencia_robos['Porcentaje'] = (frecuencia_robos['Delitos'] / total_robos) * 100

# Paso 5: Ordenar los resultados de mayor a menor
frecuencia_robos = frecuencia_robos.sort_values(by='Delitos', ascending=False)

# Mostrar los resultados
frecuencia_robos
```

	Subtipo_de_delito	Delitos	Porcentaje
0	Otros robos	1422907	28.842840
10	Robo de vehículo automotor	1234481	25.023377
3	Robo a negocio	684093	13.866813
1	Robo a casa habitación	562433	11.400721
5	Robo a transeúnte en vía pública	545073	11.048827
7	Robo de autopartes	111417	2.258463
12	Robo en transporte público colectivo	107575	2.180584
11	Robo en transporte individual	95505	1.935921
6	Robo a transportista	71077	1.440757
8	Robo de ganado	40539	0.821740
4	Robo a transeúnte en espacio abierto al público	27243	0.552225
13	Robo en transporte público individual	16944	0.343461
9	Robo de maquinaria	10937	0.221697
2	Robo a institución bancaria	3087	0.062575

# Tabla de frecuencia y porcentaje para tipos de lesiones

---

```
# Tabla de frecuencia y % de ocurrencia para los tipos de lesiones

# Paso 1: Filtrar los datos por Tipo_de_delito = "Robo"
df_robos = df_melted[df_melted['Tipo_de_delito'] == 'Lesiones']

# Paso 2: Agrupar por Subtipo_de_delito y calcular la frecuencia
frecuencia_robos = (
    df_robos.groupby('Subtipo_de_delito')['Delitos']
    .sum() # Sumar el número total de delitos por subtipo
    .reset_index() # Reiniciar el índice
)

# Paso 3: Calcular el total global de robos
total_robos = frecuencia_robos['Delitos'].sum()

# Paso 4: Agregar la columna de porcentaje
frecuencia_robos['Porcentaje'] = (frecuencia_robos['Delitos'] / total_robos) * 100

# Paso 5: Ordenar los resultados de mayor a menor
frecuencia_robos = frecuencia_robos.sort_values(by='Delitos', ascending=False)

# Mostrar los resultados
frecuencia_robos
```

	Subtipo_de_delito	Delitos	Porcentaje
1	Lesiones dolosas	1059632	75.615409
0	Lesiones culposas	341712	24.384591

# Tabla de frecuencia y porcentaje por tipo de Daño a la propiedad

---

```
# Tabla de frecuencia y % de ocurrencia para los tipos de Daño a la propiedad

# Paso 1: Filtrar los datos por Tipo_de_delito = "Robo"
df_robos = df_melted[df_melted['Tipo_de_delito'] == 'Daño a la propiedad']

# Paso 2: Agrupar por Subtipo_de_delito y calcular la frecuencia
frecuencia_robos = (
    df_robos.groupby('Subtipo_de_delito')['Delitos']
    .sum() # Sumar el número total de delitos por subtipo
    .reset_index() # Reiniciar el índice
)

# Paso 3: Calcular el total global de robos
total_robos = frecuencia_robos['Delitos'].sum()

# Paso 4: Agregar la columna de porcentaje
frecuencia_robos['Porcentaje'] = (frecuencia_robos['Delitos'] / total_robos) * 100

# Paso 5: Ordenar los resultados de mayor a menor
frecuencia_robos = frecuencia_robos.sort_values(by='Delitos', ascending=False)

# Mostrar los resultados
frecuencia_robos
```

	Subtipo_de_delito	Delitos	Porcentaje
0	Daño a la propiedad	895345	100.0





# Nuevo dataset para análisis de delitos relacionados a vehículos

---

# Nuevo dataset exportado a .csv

```
#Exporto de la BD solo lo que se necesita para el informe con delitos relacionados con vehiculos

# Configuración de la conexión
server = 'localhost' # Nombre o dirección del servidor SQL Server
database = 'Delitos' # Nombre de tu base de datos
username = 'paob2' # Usuario de SQL Server
password = 'Paulo2024$' # Contraseña del usuario
driver = '{ODBC Driver 17 for SQL Server}' # Asegúrate de tener el driver adecuado instalado

# Conexión a la base de datos
try:
    ##connection = pyodbc.connect(
    ##    f'DRIVER={driver};SERVER={server};DATABASE={database};UID={username};PWD={password}'
    ##)

    connection = pyodbc.connect(
        f'DRIVER={driver};SERVER={server};DATABASE={database};Trusted_Connection=yes'
    )
    print("Conexión exitosa a la base de datos.")

    # Creación de un cursor
    cursor = connection.cursor()

    # Ejecución de una consulta
    query = "SELECT *, CAST(Enero AS INT) + CAST(Febrero AS INT) + CAST(Marzo AS INT) + CAST(Abril AS INT) + CAST(Mayo AS INT) + CAST(Junio AS INT) + CAST(Julio AS INT) + CAST(Agosto AS INT) + CAST(Septiembre AS INT) + CAST(Octubre AS INT) + CAST(Noviembre AS INT) + CAST(Diciembre AS INT) AS Total_Meses"

    # Leer los datos en un DataFrame de Pandas
    df = pd.read_sql_query(query, connection)

except pyodbc.Error as e:
    print("Error al conectar a la base de datos:", e)

finally:
    # Cerrar la conexión
    if 'connection' in locals() and connection:
        connection.close()
        print("Conexión cerrada.")

df.to_csv("Delitos_sorted.csv", index=False)
```

Conexión exitosa a la base de datos.  
C:\Users\paob2\AppData\Local\Temp\ipykernel\_11292\3801578857.py:30: UserWarning: pandas only supports SQLAlchemy connectable (engine/connection) or database string URI or sqlite3 DBAPI2 connection. Other DBAPI2 objects are not tested. Please consider using SQLAlchemy.  
df = pd.read\_sql\_query(query, connection)  
Conexión cerrada.

Basados en el EDA realizado anteriormente generamos un nuevo archivo csv únicamente con la información relevante para los delitos relacionados con vehículos.



# Visualizaciones tablero en Looker Studio

---

Total de delitos analizados

1.140.973

Promedio Frecuencia x Delito

2.018,17

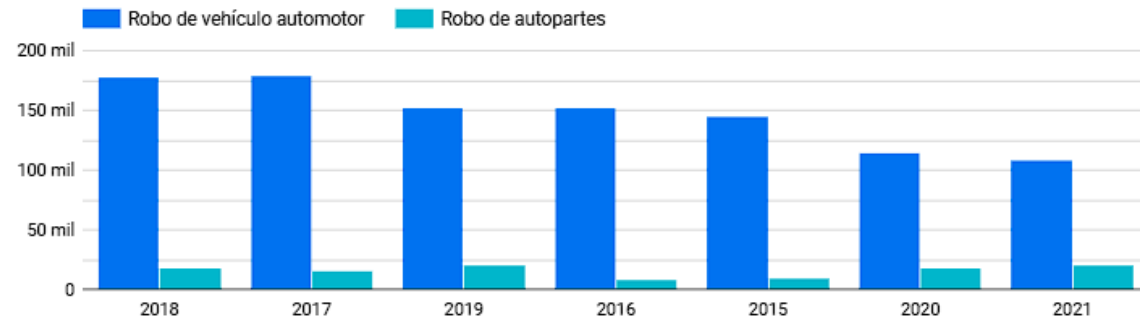
Cantidad Entidades

32

Subtipo\_de\_delito

Robo de vehículo automotor

Gráfico por Subtipos de Delito por Año



Año

Subtipo\_de\_delito

### El robo de vehículo automotor es el problema

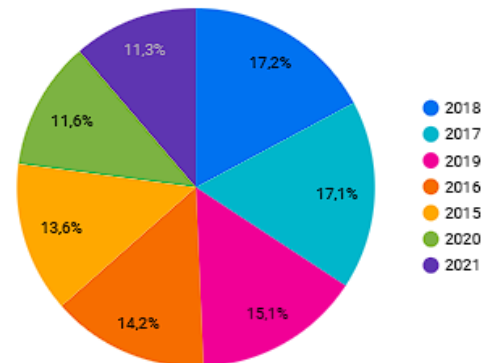
De todos los delitos que tienen que ver con vehículos, los que se han reportado todos tienen que ver con el delito de "Robo" y los dos Subtipos son "Robo de vehículo automotor" y "Robo de autopartes".

De inicio pensaríamos que es más fácil y común el robo de partes de vehículo, pero la data nos indica algo muy diferente, y es justo porque las piezas mas costosas de un vehículo no se pueden robar tan fácilmente, por lo que robar el vehículo para venderlos por partes es una gran modalidad que puede sustentar estas cifras.

El primer indicador de gran poder es la gran diferencia entre el robo de vehículos y el robo de partes, por lo que es muy importante centrarnos a partir de este momento en este tipo de delito, que además afecta en mayor proporción a la compañía de seguros de autos.

Finalmente no hay una varianza muy alta en el número de delitos por año, lo que indica a simple ojo que no va a haber un disminución notoria de delitos en los próximos años

Gráfico porcentaje anual de delitos



# Comportamiento Anual

Moda Subtipo de delito

## Robo de vehículo automotor

Moda Entidad

## Oaxaca

Moda Municipio

## General Bravo

Tabla comparativa de delitos por Año - Mes

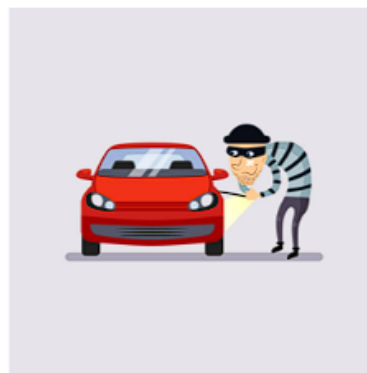
Año	Enero	Febrero	Marzo	Abril	Mayo	Junio	Julio	Agosto	Septiem...	Octubre	Noviem...	Diciembre	Total_ye...
2018	17.540	15.744	16.743	15.945	16.741	15.641	16.469	16.829	15.179	17.168	16.286	15.792	196.077
2017	16.079	14.782	16.232	14.975	16.171	15.934	16.295	17.341	16.419	17.430	16.832	16.067	194.557
2019	15.916	13.903	15.410	14.193	15.052	14.248	14.940	14.260	13.464	14.210	13.539	13.006	172.141
2020	13.695	12.688	12.671	9.722	9.813	10.187	10.781	10.389	10.516	11.160	10.617	10.544	132.783
2015	13.581	12.769	13.395	11.994	12.673	12.834	12.876	12.966	13.249	13.297	13.172	12.404	155.210
2016	12.121	12.592	12.978	12.434	12.697	13.133	13.556	13.850	14.291	14.575	14.534	14.872	161.633
2021	10.147	9.721	10.885	10.322	10.609	10.866	10.977	11.131	10.926	11.276	10.997	10.715	128.572
Total	99.079	92.199	98.314	89.585	93.756	92.843	95.894	96.766	94.044	99.116	95.977	93.400	1.140.973

### Enero y Octubre son meses de mucho cuidado

Analizando las modas de los meses por año, vemos como Enero tiene mas del 50% de frecuencia como el mes con mas delitos en el año.

Luego le sigue Octubre por lo que podemos notar una leve tendencia en ese par de meses donde aumenta el riesgo de reporte de incidencias por robo de vehículo.

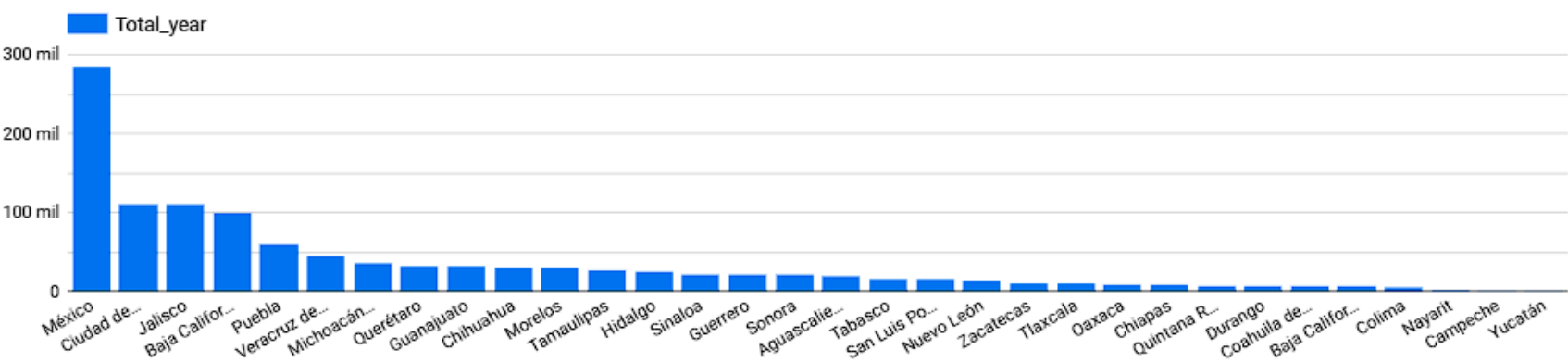
Por otro lado los meses con menos delitos suelen ser Febrero y Abril, por lo que pueden ser meses estratégicos en el tema financiero.



# Comportamiento Mensual



Gráfico por Subtipos de Delito por Entidad



Top 10 municipios con más delitos

	Municipio	Entidad	Total_year
1.	Ecatepec de Morelos	México	60.121
2.	Tijuana	Baja California	53.656
3.	Guadalajara	Jalisco	48.519
4.	Tlalnepantla de Baz	México	27.299
5.	Mexicali	Baja California	25.724
6.	Puebla	Puebla	23.382
7.	Zapopan	Jalisco	22.432
8.	Querétaro	Querétaro	21.178
9.	Toluca	México	18.846

Después del Cuando... el Donde

Ya descubrimos los meses con tendencia a más y menos delitos, lo que nos permite comenzar a tomar decisiones estratégicas basados en el cuando.

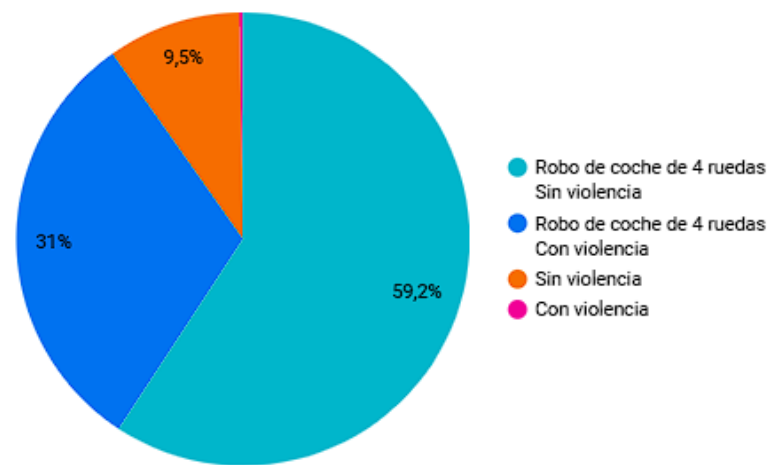
Ahora esta información nos deja bastante claro el nivel que pueden alcanzar en delitos en el Estado de México principalmente, CDMX, Jalisco, Baja California y Puebla. Estas son las entidades con mas riesgo de todo el país, pero cuando se piensa desde un panorama comercial podemos ver que pueden haber otro tipo de oportunidades que hablaremos en los insights.

Finalmente, si vamos más al detalle por Municipio. Ecatepec de Morelos, Tijuana, Guadalajara, Tlalnepantla de Baz y Mexicali se clasifican como los que más ocurrencia de delitos relacionados con vehículos tienen.

Al igual, esta es una oportunidad para generar estrategias comerciales mas allá de tener un valor elevado de seguros, para disminuir este flagelo.

# Comportamiento por entidad

Grafico de porcentaje por Modalidad



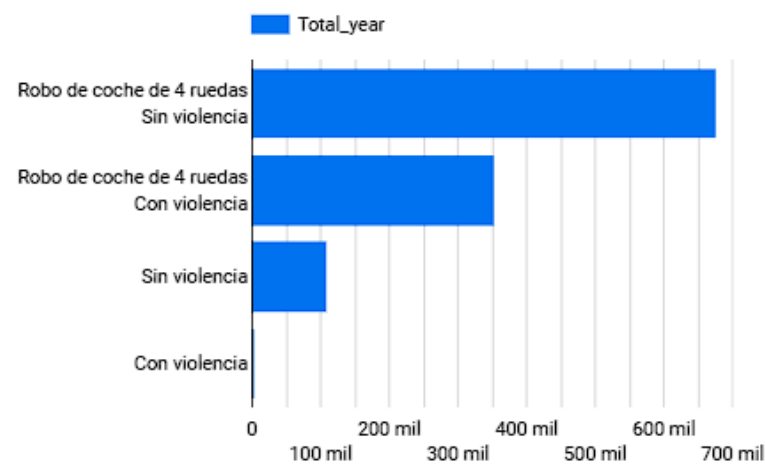
El Cómo segmenta mucho mejor una estrategia

Ya habíamos hecho el descubrimiento de que el robo de vehículos es mucho más común que el robo de repuestos. En este slide logramos visualizar mejor esa gran diferencia, pero con un ingrediente adicional y es el tema de la violencia.

Estamos seguros que cualquier delito que involucre el daño a personas suelen tener un peso y costo mucho mas grande, que simplemente daños materiales. Por tanto si sumamos los porcentajes de delitos sin violencia, donde presuntamente no hay una persona presente a la hora del robo, llegamos a un 68.7%, es decir una gran mayoría, y esta condición de violencia a la hora de vender seguros para autos, resulta de gran valor, porque baja el riesgo de tener que asumir valores por lesiones personales.

Este comportamiento general, tiene solo una excepción y es el Estado de México, que es el único que tiene una cantidad de robos de vehiculo con violencia mucho mas alto que los que no usan violencia.

Cantidad de delitos por modalidad



Top 10 Modalidad - Entidad

Modalidad	Entidad	Total_year
1.. Robo de coche de 4 ruedas Con violencia	México	147.052
2.. Robo de coche de 4 ruedas Sin violencia	México	126.027
3.. Robo de coche de 4 ruedas Sin violencia	Baja California	91.453
4.. Robo de coche de 4 ruedas Sin violencia	Jalisco	66.632
5.. Sin violencia	Ciudad de México	46.281
6.. Robo de coche de 4 ruedas Sin violencia	Ciudad de México	44.794
7.. Robo de coche de 4 ruedas Sin violencia	Puebla	31.928
8.. Robo de coche de 4 ruedas Con violencia	Jalisco	31.548
9.. Robo de coche de 4 ruedas Sin violencia	Veracruz de Ignacio de...	30.068

1 - 100 / 128 < >

Panorama de violencia

Subtipo_de_delito	Total_year ▾
1. Robo de vehículo automotor	90,23 %
1 - 2 / 2 < >	

Entidad	Total_year ▾
1. México	24,98 %
2. Ciudad de México	9,78 %
3. Jalisco	9,75 %
4. Baja California	8,79 %
5. Puebla	5,2 %
6. Veracruz de Ignacio de la Lla...	3,99 %
7. Michoacán de Ocampo	3,23 %
8. Querétaro	2,89 %
9. Guanajuato	2,83 %
10. Chihuahua	2,69 %
11. Morelos	2,66 %
12. Tamaulipas	2,31 %
13. Hidalgo	2,18 %
1 - 32 / 32 < >	

Modalidad	Total_year ▾
1. Robo de coche de 4 ruedas Sin violencia	59,23 %
2. Robo de coche de 4 ruedas Con violencia	31,01 %
1 - 4 / 4 < >	

Meses con mas delitos

Año	Enero	Octubre
2018	17.540	17.168
2017	16.079	17.430
2019	15.916	14.210
2020	13.695	11.160
2015	13.581	13.297
2016	12.121	14.575
2021	10.147	11.276
Total	99.079	99.116

Meses con menos delitos

Año	Febrero	Abril
2018	15.744	15.945
2017	14.782	14.975
2019	13.903	14.193
2015	12.769	11.994
2020	12.688	9.722
2016	12.592	12.434
2021	9.721	10.322
Total	92.199	89.585

**La ley de Pareto, también conocida como regla del 80/20, es un principio que establece que el 80% de los resultados provienen del 20% de las causas.**

En definitiva esta ley nos ayuda a centrarnos el 80/20 que realmente importa, y si miramos las tablas estamos viendo justamente ese 80/20 totalmente coherente con todas las graficas que nos vienen explicando en Cuándo, el Dónde y el Cómo de los delitos por robo de vehículos en México.

Así que para llegar a los insights y conclusiones este resumen es de gran valor, sabiendo que debemos prestar gran atención a:

- \* Los meses de enero y octubre con mayores delitos y a Febrero y Abril con menos delitos.
- \* El principal delito es el robo de vehículo
- \* Las principales entidades con robos son México, Ciudad de México, Baja Californio y Puebla.
- \* La gran mayoría de la modalidad es sin Violencia

# Pareto de la data

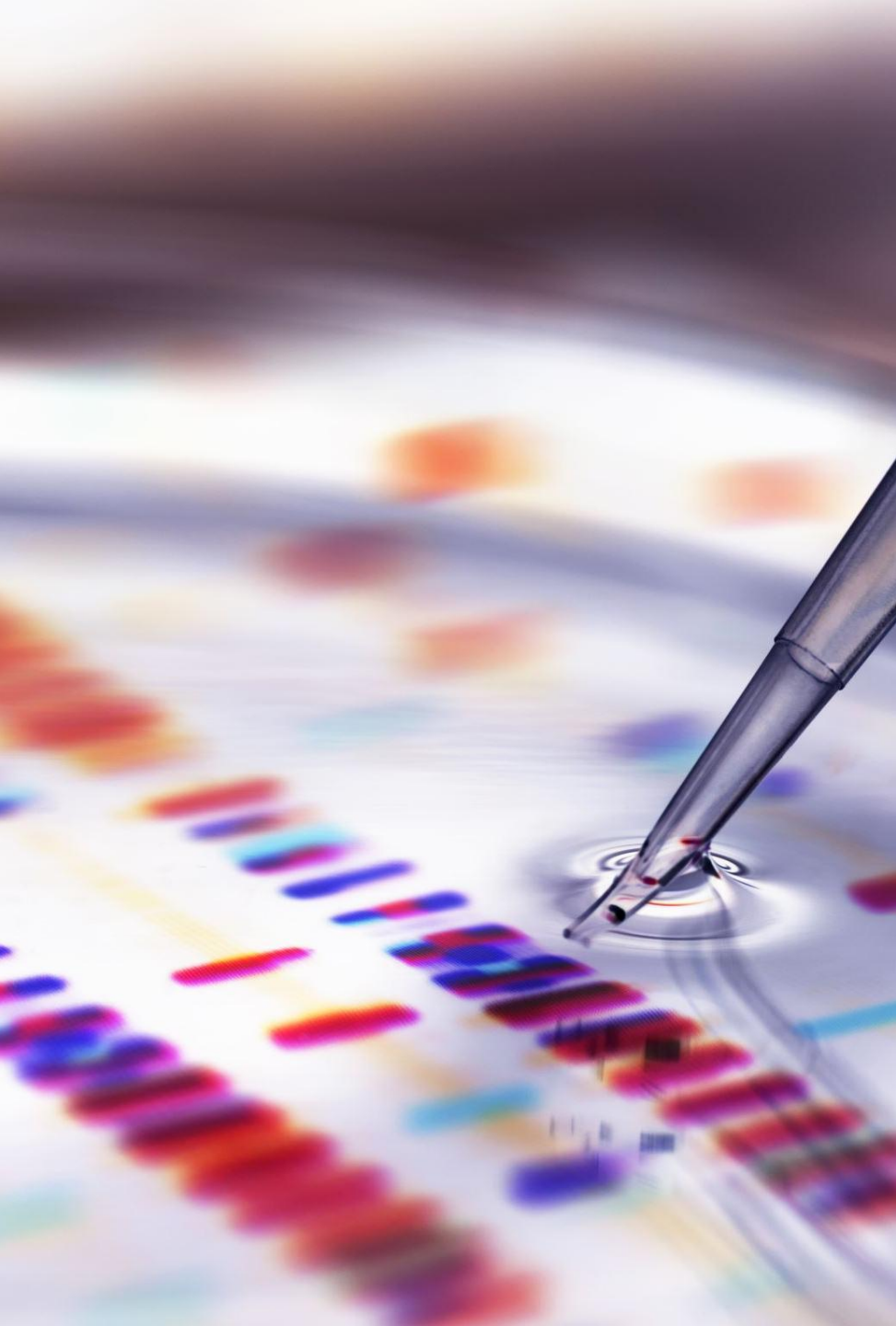
## HALLAZGOS

- No hay una varianza muy grande entre año y año por lo que no se puede determinar que los delitos vayan a subir o bajar en gran cantidad.
- Los meses donde suelen haber más delitos son Enero y Octubre
- Los meses donde suelen haber menos delitos son Febrero y Abril
- Los estados de México, CDMX, Jalisco, Baja California y Puebla con las entidades con la gran mayor representación de robo de vehículos en todo México.
- El 68% de los robos de vehículos o partes de vehículos se hace sin el uso de la violencia
- El Robo de vehículos es mucho mas común y usado que el robo de partes de vehículo.

## RECOMENDACIONES

- Si unimos todas las variables de riesgo, en efecto los valores de las pólizas pueden resultar mucho mas caras, sin embargo queda la oportunidad de prorratear con otros municipios que pueden pagar menos y el riesgo es mínimo.
- Adicional a apegarse al valor de una póliza de seguro de vehículo se puede ver la oportunidad de hacer campañas de prevención contra robos, en especial en los meses y lugares donde hay más vulnerabilidad.
- Igualmente se pueden hacer convenios con negocios que se encargan de la seguridad física de los vehículos o la inclusión de métodos de auto-apagado o uso de GPS para la localización, esto ayudaría a encontrar los vehículos y hacer operativos para la recuperación de los mismo disminuyendo los costes del seguro.
- Finalmente en su mayoría no se usa la violencia, lo que quiere decir que el rango de pago por lesiones personales baja.

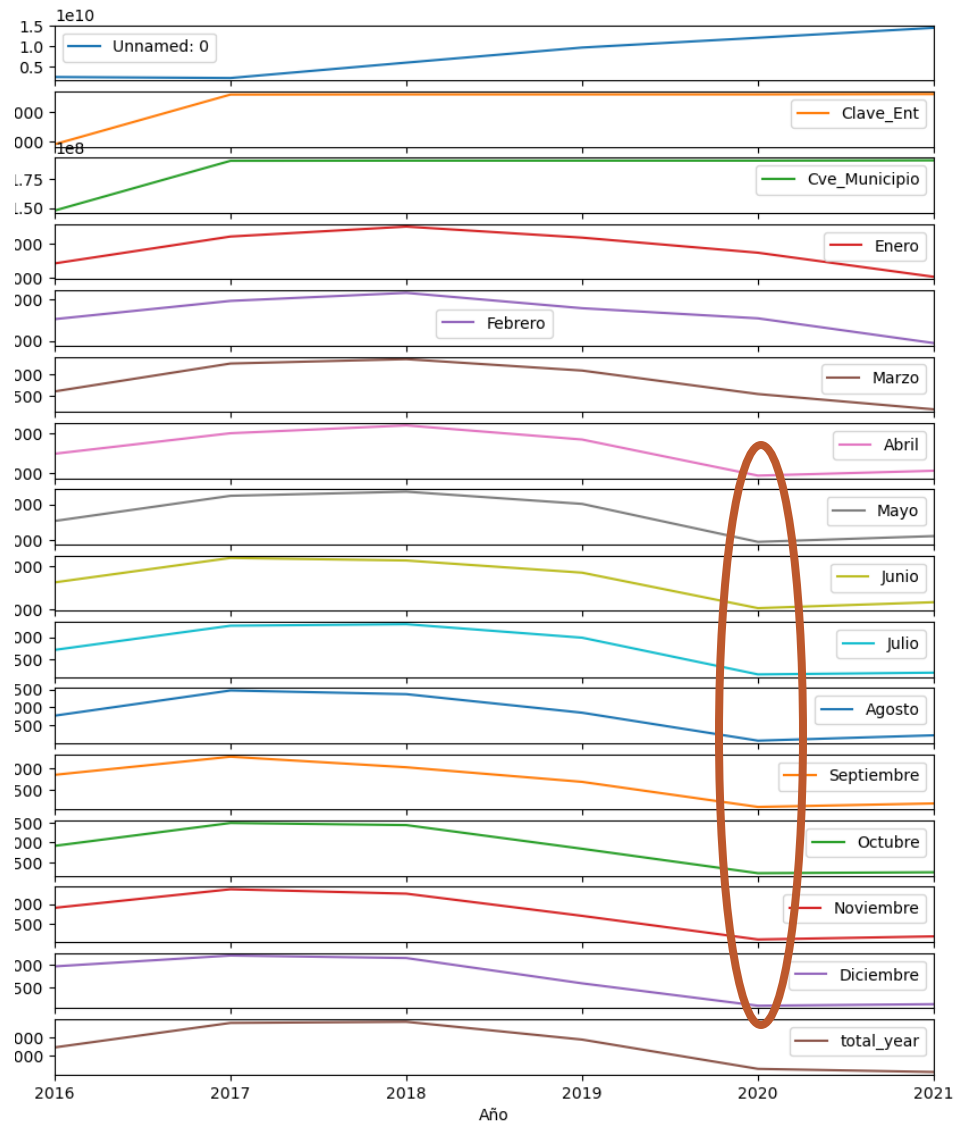
Insights  
comerciales y  
conclusiones



# Análisis de series de tiempo

---





# Visualización de series de tiempo

Estos datos corresponden al periodo 2016-2021

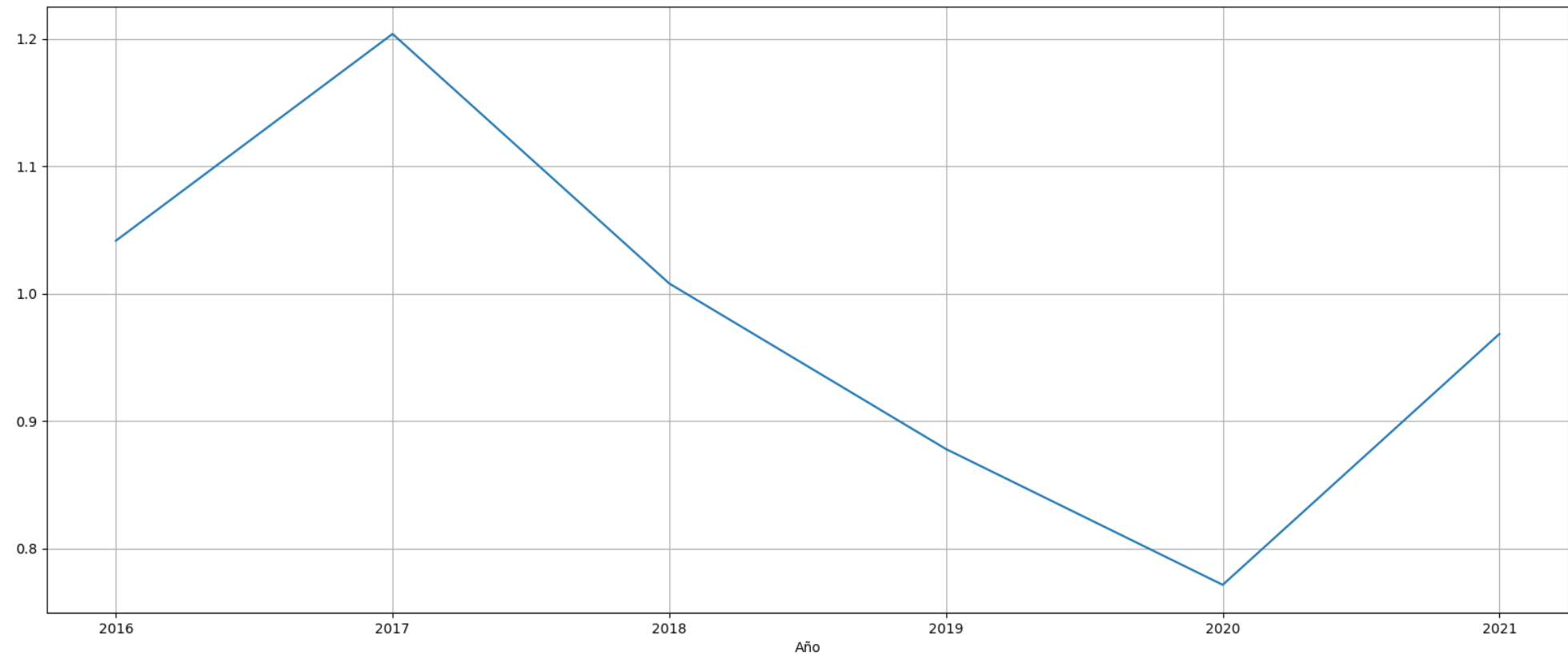
Se debe contemplar la situación de pandemia en 2020 para el análisis de los datos.

Se ve muy claramente el cambio en la tendencia de delitos a partir de abril de 2020 y su leve recuperación en 2021.

La tendencia de los delitos es congruente en las líneas de tiempo año a año, visualizando cada mes.

# Comportamiento anual del total de delitos

---



# Modelo de predicción simple

```
train_len = 5 # 70%

# Separar datos de entrenamiento y prueba correctamente
train = df_grouped.iloc[:train_len] # Primeras 45413 filas como entrenamiento
test = df_grouped.iloc[train_len:] # El resto como prueba

# Crear copia del DataFrame para predicciones
y_pred_sma = df_grouped.copy()

# Configurar la ventana del promedio móvil
ma_window = 2

# Calcular el promedio móvil (corregir error de escritura "rollling" -> "rolling")
y_pred_sma['sma_forecast'] = df_grouped['total_year'].rolling(ma_window).mean()

# Asignar el valor de la predicción en la posición correcta
y_pred_sma.loc[test.index, 'sma_forecast'] = y_pred_sma['sma_forecast'].iloc[train_len:]
```

Entrenamiento

Predicción

```
# Prediccion 2022

# Agregar el año 2022 al DataFrame
df_grouped.loc[2022] = None # Agregamos una fila vacía para 2022

# Calcular la predicción para 2022 usando la media de 2020 y 2021
df_grouped.loc[2022, 'sma_forecast'] = df_grouped['total_year'].iloc[-2:].mean()

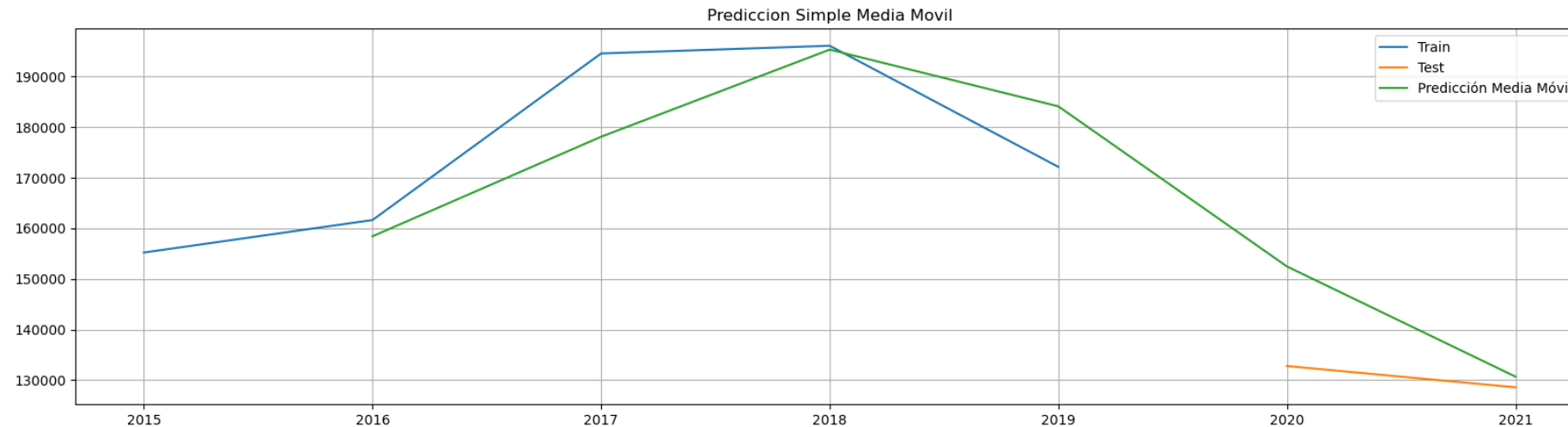
print(df_grouped[['total_year', 'sma_forecast']].tail(3)) # Ver los últimos 3 años
```

```
# Validacion de la predicción
rmse = np.sqrt(mean_squared_error(test['total_year'], y_pred_sma['sma_forecast'][train_len:])).round(2)
mape = np.round(np.mean(np.abs(test['total_year'] - y_pred_sma['sma_forecast'][train_len:])/test['total_year'])*100, 2)

results = pd.DataFrame({'Method': ['Simple moving average forecast'], 'MAPE': [mape], 'RMSE': [rmse]})
results = results[['Method', 'RMSE', 'MAPE']]
results
```

Validación

# Modelo de predicción simple



	Method	RMSE	MAPE
0	Simple moving average forecast	13994.5700	8.2300

En el modelo realizado se hicieron pruebas con diferentes ventanas de tiempo y el mejor MAPE lo tuvo una ventana = 2. Con esto vemos una tendencia a la reducción en el total de delitos relacionados con vehículos

# Forecast 2022

---

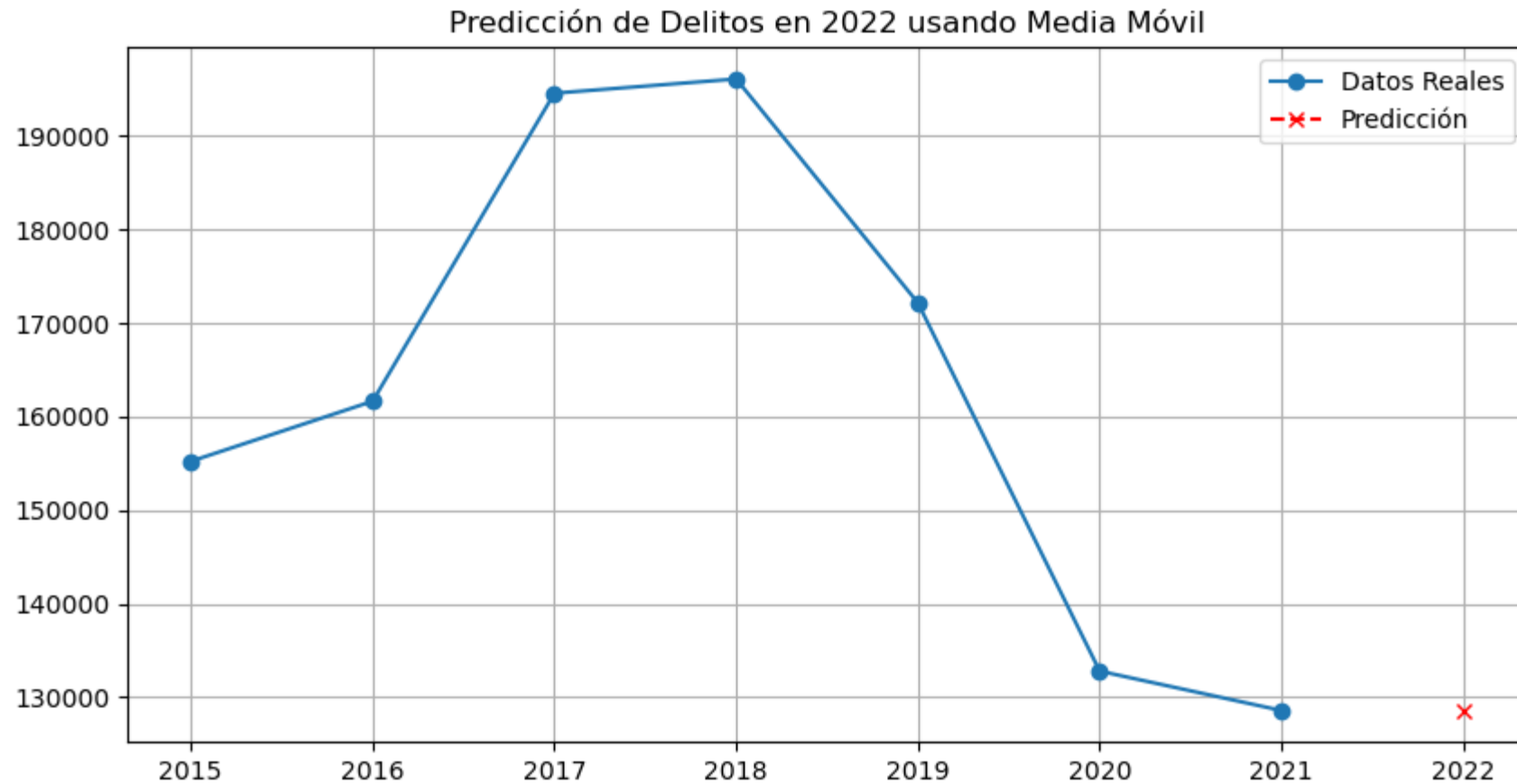
	forecast	observed
Año		
2015	NaN	155210
2016	158421.5000	161633
2017	178095.0000	194557
2018	195317.0000	196077
2019	184109.0000	172141
2020	152462.0000	132783
2021	130677.5000	128572

El comparativo de resultados entre el forecast y las observaciones tiende a ser consistente lo que nos confirma un buen porcentaje de confiabilidad en el modelo.

	total_year	sma_forecast
Año		
2020	132783.0000	NaN
2021	128572.0000	NaN
2022	NaN	128572.0000

Los resultados del forecast modelando 2022 basados en los últimos años, contemplando la variación especial de pandemia, nos arroja una estabilidad en el número de delitos para 2022, con un 8,23 de probabilidades de subir o bajar esas cifras.

# Gráfica predicción 2022





# Municipio estratégico

**Ecatepec de Morelos** tiene el índice mas alto de delitos relacionados con vehículos.

Municipio	total_year
Ecatepec de Morelos	60121
Tijuana	53656
Guadalajara	48519
Tlalnepantla de Baz	27299
Mexicali	25724
Puebla	23382
Zapopan	22432
Querétaro	21178
Benito Juárez	19290
Toluca	18846
Naucalpan de Juárez	18346
Juárez	18290
Aguascalientes	17141
Iztapalapa	16993
Cuautitlán Izcalli	16537
Gustavo A. Madero	14268
Nezahualcóyotl	14040
Culiacán	13764
Ensenada	12808
Cuernavaca	12755

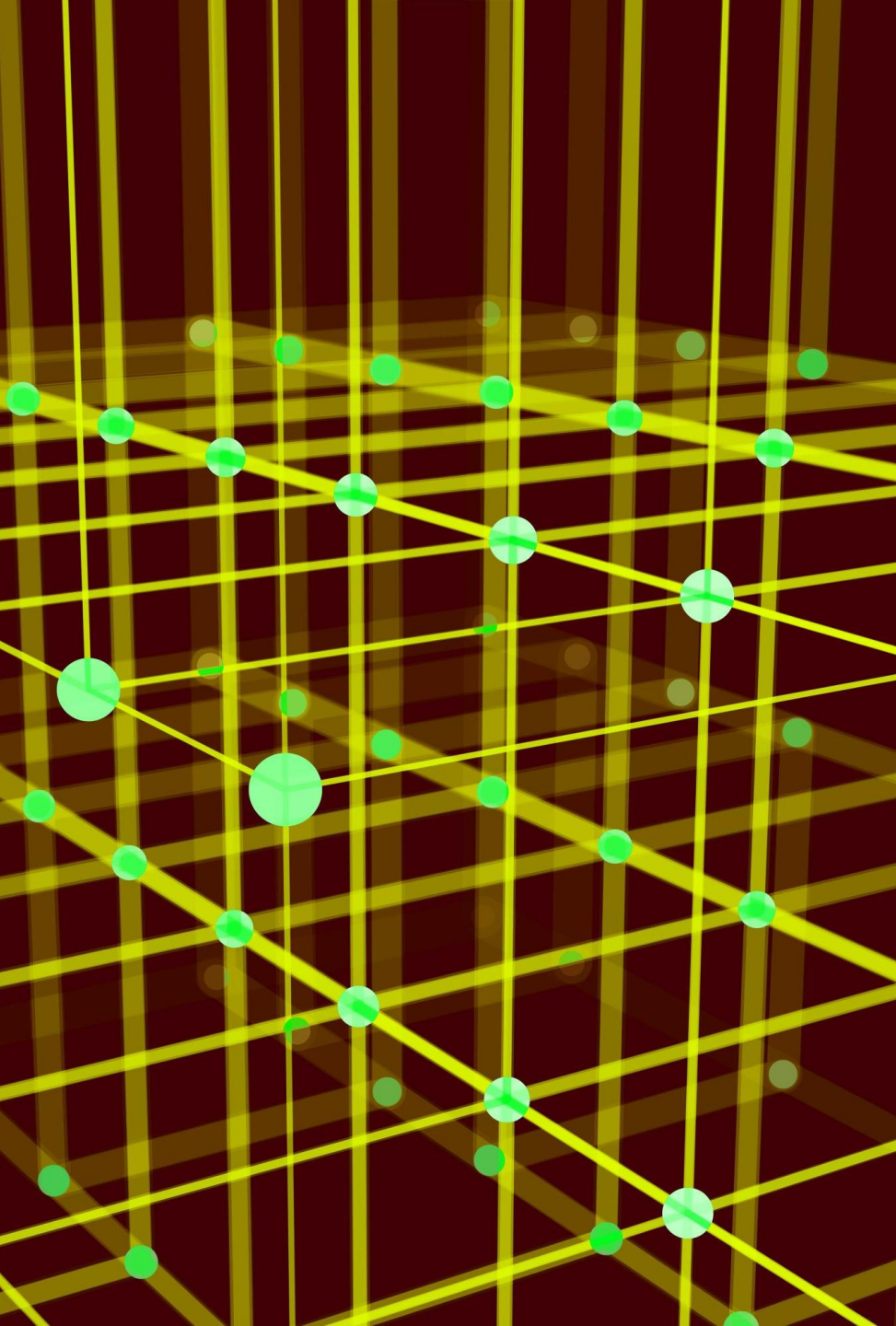
## ¿Por qué elegir este municipio estratégicamente?

**Alta demanda de seguros:** Un municipio con la mayor incidencia de delitos relacionados con vehículos implica un **mayor riesgo para los propietarios de autos**, lo que puede traducirse en una **alta demanda de pólizas de seguro**.

**Primas más altas y rentabilidad:** En zonas de alto riesgo, las aseguradoras pueden **ajustar las primas** de los seguros, lo que puede representar **una mayor rentabilidad** si se maneja correctamente.

**Estrategia de mitigación de riesgo:** Conociendo este municipio, la empresa puede implementar estrategias de **seguridad adicional**, descuentos en dispositivos de rastreo, o colaboraciones con autoridades locales para reducir el riesgo.

**Expansión estratégica:** Si la empresa busca **expansión geográfica**, empezar en un municipio con **alta incidencia de delitos** puede ser una oportunidad para captar clientes preocupados por la seguridad de sus vehículos.



# Clasificación de estados por peligrosidad (Clustering)

---

```
#Información de campos del dataset
df.info()
```

✓ 0.0s

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

```
RangeIndex: 64876 entries, 0 to 64875
```

```
Data columns (total 23 columns):
```

#	Column	Non-Null Count	Dtype
0	Unnamed: 0	64876 non-null	int64
1	Año	64876 non-null	int64
2	Clave_Ent	64876 non-null	int64
3	Entidad	64876 non-null	object
4	Cve_Municipio	64876 non-null	int64
5	Municipio	64876 non-null	object
6	Bien_jurídico_afectado	64876 non-null	object
7	Tipo_de_delito	64876 non-null	object
8	Subtipo_de_delito	64876 non-null	object
9	Modalidad	64876 non-null	object
10	Enero	64876 non-null	int64
11	Febrero	64876 non-null	int64
12	Marzo	64876 non-null	int64
13	Abril	64876 non-null	int64
14	Mayo	64876 non-null	int64
15	Junio	64876 non-null	int64
16	Julio	64876 non-null	int64
17	Agosto	64876 non-null	int64
18	Septiembre	64876 non-null	int64
19	Octubre	64876 non-null	int64
...			
21	Diciembre	64876 non-null	int64
22	total_year	64876 non-null	int64

# Normalización de datos

- Los campos de Entidad, Municipio, Bien\_jurídico\_afectado, Tipo\_de\_delito, Subtipo\_de\_delito, Modalidad están como tipo object. Ya que este dataset ya es un subconjunto de datos relacionado con delitos relacionados a vehículos, y que el objetivo esta relacionado con evaluar el número de delitos y no el tipo, solo para la variable "Entidad", no se hace necesario hacer un paso a datos numéricos, contemplando que la "Entidad" esta relacionada directamente con el campo Clave\_Ent que es de tipo "int", por tanto podemos realizar el modelo sin problemas.
- Se debe de eliminar el campo Unnamed: 0, qyue no tiene información de valor para el dataset
- La cantidad de datos en cada campo es consistente, por lo cual no hay datos faltantes

# Normalización de datos

```
# Datos básicos de estadística descriptiva
df.describe().T
```

✓ 0.0s

	count	mean	std	min	25%	50%	75%	max
Unnamed: 0	64876.0	794708.335162	458858.781721	1.0	397299.75	794726.5	1192113.25	1589413.0
Año	64876.0	2018.168074	1.951518	2015.0	2017.00	2018.0	2020.00	2021.0
Clave_Ent	64876.0	19.203095	7.589202	1.0	14.00	20.0	25.00	32.0
Cve_Municipio	64876.0	19303.128615	7605.000023	1001.0	14044.00	20210.0	25011.00	32058.0
Enero	64876.0	1.527206	14.043231	0.0	0.00	0.0	0.00	943.0
Febrero	64876.0	1.421157	12.995725	0.0	0.00	0.0	0.00	883.0
Marzo	64876.0	1.515414	13.698412	0.0	0.00	0.0	0.00	908.0
Abril	64876.0	1.380865	12.698194	0.0	0.00	0.0	0.00	932.0
Mayo	64876.0	1.445157	13.418357	-1.0	0.00	0.0	0.00	983.0
Junio	64876.0	1.431084	12.876541	0.0	0.00	0.0	0.00	718.0
Julio	64876.0	1.478112	13.279574	0.0	0.00	0.0	0.00	805.0
Agosto	64876.0	1.491553	13.589268	0.0	0.00	0.0	0.00	845.0
Septiembre	64876.0	1.449596	12.902367	0.0	0.00	0.0	0.00	740.0
Octubre	64876.0	1.527776	13.767299	0.0	0.00	0.0	0.00	770.0
Noviembre	64876.0	1.479391	13.423269	0.0	0.00	0.0	0.00	787.0
Diciembre	64876.0	1.439670	13.299013	0.0	0.00	0.0	0.00	945.0
total_year	64876.0	17.586981	157.060666	0.0	0.00	0.0	1.00	9917.0

Se ve un valor atípico en un conteo con un -1 en el mes de mayo. Se deben normalizar estos datos.

```
# conteo del numero de registros donde el total es cero  
df[df==0].count() / df.count()
```

✓ 0.0s

Unnamed: 0	0.000000
Año	0.000000
Clave_Ent	0.000000
Entidad	0.000000
Cve_Municipio	0.000000
Municipio	0.000000
Bien_jurídico_afectado	0.000000
Tipo_de_delito	0.000000
Subtipo_de_delito	0.000000
Modalidad	0.000000
Enero	0.867516
Febrero	0.869197
Marzo	0.866360
Abril	0.870137
Mayo	0.866638
Junio	0.866885
Julio	0.865821
Agosto	0.865682
Septiembre	0.864958
Octubre	0.865112
Noviembre	0.864526
Diciembre	0.866145
total_year	0.681531
dtype: float64	

# Normalización de datos

El porcentaje de ceros es bastante alto en los campos de meses, pero esto no es un problema debido a que es un conteo y casi todos los registros tienen un cero en algún mes.

# Normalización de datos

---

```
#Valores Faltantes
df.isnull().sum().sum()
✓ 0.0s

0

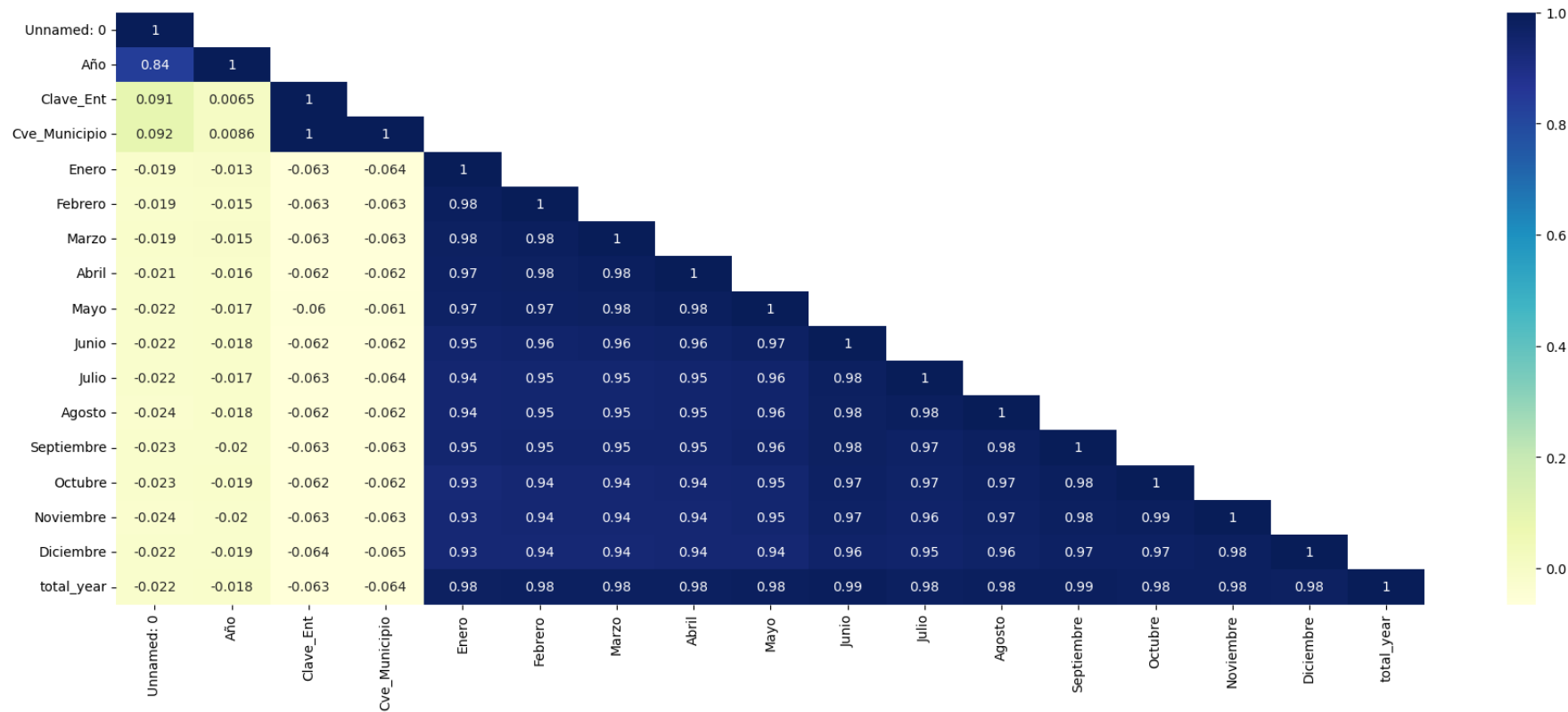
#Valores duplicados
df[df.duplicated()]
✓ 0.0s

Unnamed: 0  Año  Clave_Ent  Entidad  Cve_Mu
0 rows × 23 columns
```

No existen valores nulos o faltantes

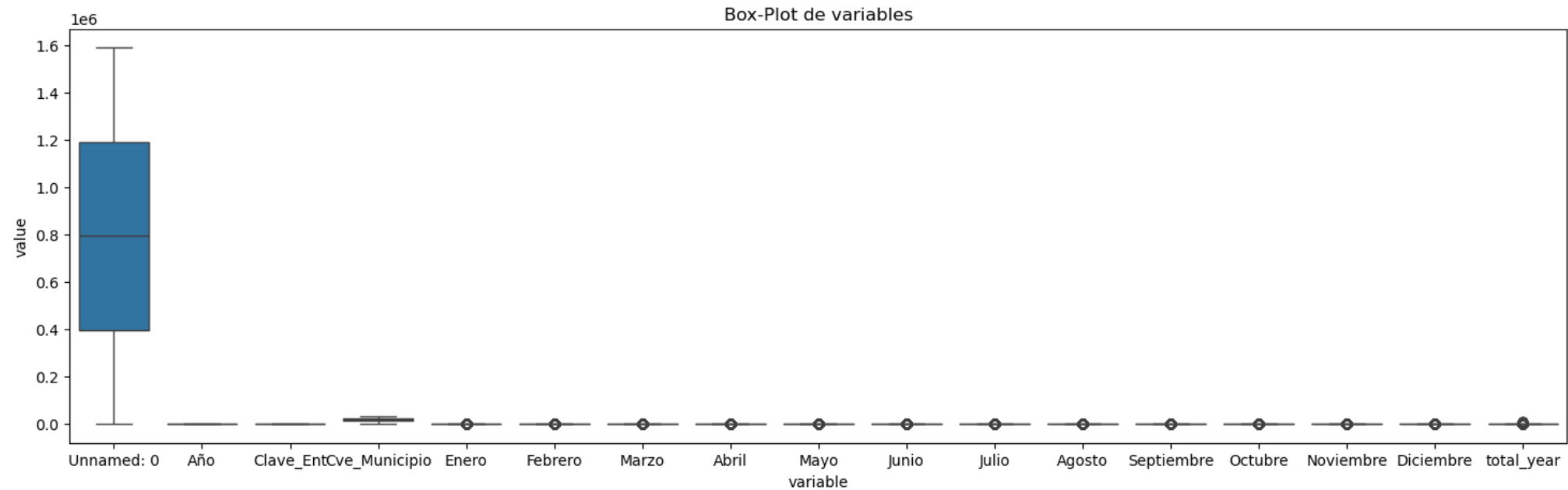


# Análisis de correlación



En el gráfico de correlación hay una muy alta correlación en los datos de los meses y total\_year, pero es totalmente normal, ya que se trata de un conteo, por lo que no se considera eliminar ninguna de las columnas, ya que son importantes para el análisis de clustering.

# Análisis de valores atípicos



No se visualizan outliers en el conjunto de datos, lo que sigue confirmando que se tiene un muy buen dataset para el desarrollo del modelo.

# Preparación data frame (Feature Engineering)

---

```
# Eliminar columna Unnamed: 0  
df2.drop("Unnamed: 0", axis=1, inplace=True)
```

[26] ✓ 0.0s

```
df2.head(3)
```

[27] ✓ 0.0s

	Año	Clave_Ent	Cve_Municipio	Enero	Febrero	Marzo	Abril	Mayo	Junio	Julio	Agosto	Septiembre	Octubre	Noviembre	Diciembre	total_year
0	2016	16	16033	2	0	3	2	2	8	3	5	0	4	2	6	37
1	2016	16	16033	2	0	1	2	0	0	3	0	0	0	0	1	9
2	2016	16	16033	0	0	0	0	0	0	0	0	0	0	0	0	0

Se elimina la columna Unnamed:0

# Preparación data frame (Feature Engineering)

```
# Ver los registros que contenga un -1
df2[df2['Mayo'] == -1]
✓ 0.0s
```

	Año	Clave_Ent	Cve_Municipio	Enero	Febrero	Marzo	Abril	Mayo	Junio	Julio	Agosto	Septiembre	Octubre	Noviembre	Diciembre	total_year	
0]	33245	2018	27	27011	0	0	0	0	-1	0	0	0	3	1	1	0	4

```
# Imputacion del valor -1 en un registro por el valor de 0
# No se eliminia el registro porque tiene datos de conteo de delitos en otros meses
# Se imputa con 0 ya que es un conteo que no puede tener valores negativos, y su valor mas cercano es 0. Nos se hace un mean o media, porque son valores elevados que
# pueden no ser congruentes con un registro en un municipio determinado
df2['Mayo'] = df2['Mayo'].replace(-1, 0)
print(df2['Mayo'].unique()) # Debería mostrar valores distintos a -1
✓ 0.0s
```

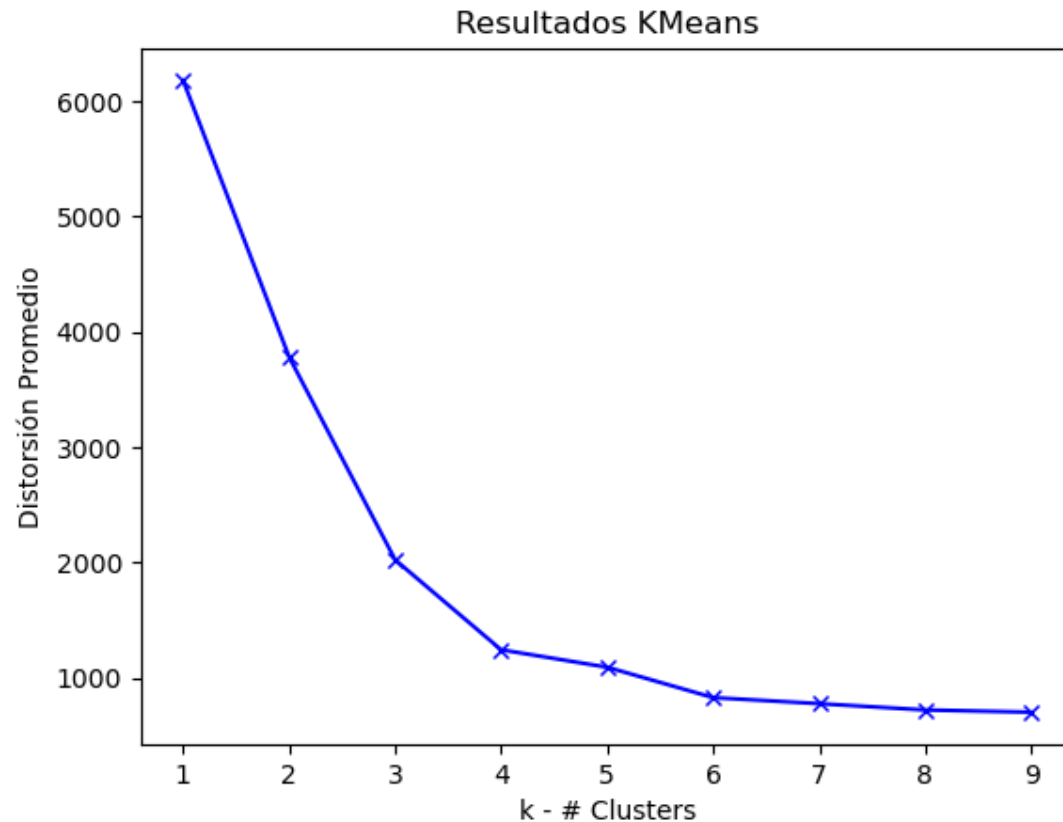
```
[ 2  0  1 11 10 42 158  4 12  3  9 18 46 23  6 27 44 75
85 17 19  5  7  8 33 53 14 35 22 128 13 20 26 15 223 32
60 71 38 16 24 55 167 36 120 25 47 66 39 69 65 31 21 34
40 43 200 111 209 385 72 983 61 59 51 64 80 135 45 57 82 104
30 37 28 149 29 50 92 93 193 462 106 89 101 119 274 58 56 54
434 129 41 137 98 322 160 204 150 67 48 273 70 297 232 208 155 81
189 122 196 248 680 52 251 74 103 84 90 102 182 134 63 284 451 96
118 231 73 77 715 203 49 179 62 336 125 83 86 76 117 138 400 488
126 79 100 68 124 140 97 619 177 141 94 121 139 185 174 112 113 178
123 460 545 105 187 78 626 157 114 115 148 153 110 271 262 88 159 540
162 256 402 87 127 235 116 176 143 131 91 172 506 154 107 163 454 144
170 240 415 132]
```

Imputación del valor -1  
en el mes de mayo.

Solo era un valor en un  
solo registro.

# Definición número de clusters

---



Se realiza un modelo de Kmeans con un rango de iteración de 1-10 para determinar una gráfica de distorsión promedio.

Vemos como el elbow (codo) se forma en un número de clusters igual a 4, por lo que de acuerdo a la teoría este es el valor más objetivo para realizar el modelo.

# Configuración del modelo

---

```
kmeans = KMeans(n_clusters=4, n_init = 15, random_state=1)  
kmeans.fit(df2)
```

✓ 0.2s

▼ KMeans ⓘ ?  
KMeans(n\_clusters=4, n\_init=15, random\_state=1)

Se genera el número de cluster que se definió que fue de 4.

También se define el parámetro de número de iteraciones que irá mejorando el modelo.



# Mapeo de centroides

```
#Se mapean los centroides con la lista de elementos y se crea un dataframe para los labels de los clusters
centroids = kmeans.cluster_centers_
df_centroids = pd.DataFrame(centroids, columns= list(df2))
df_centroids
```

✓ 0.0s

	Año	Clave_Ent	Cve_Municipio	Enero	Febrero	Marzo	Abril	Mayo	Junio	Julio	Agosto	Septiembre	Octubre	Noviembre	Diciembre	total_year
0	2018.018494	6.958168	6996.974461	2.969177	2.720828	2.889916	2.633311	2.667657	2.629568	2.737340	2.761559	2.670850	2.834654	2.767063	2.851277	33.133201
1	2018.470536	20.639050	20821.209850	0.616843	0.567018	0.609499	0.568734	0.586631	0.592964	0.620756	0.633861	0.583509	0.622559	0.614556	0.620800	7.237731
2	2018.000000	29.526224	29595.298951	0.733204	0.687188	0.733142	0.664023	0.695430	0.685252	0.671516	0.678384	0.700175	0.737450	0.686563	0.644605	8.316871
3	2018.002113	14.110589	14162.751585	2.719946	2.558288	2.727166	2.471003	2.644165	2.611939	2.709380	2.723703	2.659016	2.782226	2.692533	2.527471	31.826837

Luego de correr el modelo, se mapean los resultados de los centroides optimizados con el número de interacciones que se habían definido anteriormente. Con esto ya se han agrupado en cada cluster (por eso vemos 4 filas) los registros correspondientes que ha determinado el modelo.

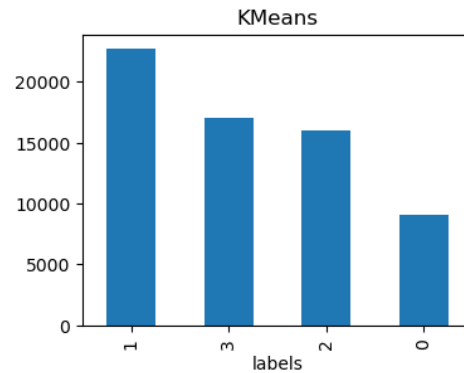
# Definición de labels

	Año	Clave_Ent	Cve_Municipio	Enero	Febrero	Marzo	Abril	Mayo	Junio	Julio	Agosto	Septiembre	Octubre	Noviembre	Diciembre	total_year	labels
35305	2019	4	4007	0	0	0	0	0	0	0	0	0	0	0	0	0	0
44474	2019	31	31013	0	0	0	0	0	0	0	0	0	0	0	0	0	2
31255	2018	20	20481	0	0	0	0	0	0	0	0	0	0	0	0	0	1
13572	2017	30	30190	0	0	0	0	0	0	0	0	0	0	0	0	0	2
5826	2017	11	11020	0	0	0	0	0	0	0	0	0	0	0	0	0	3
22416	2015	16	16042	0	0	0	1	0	0	0	0	1	1	0	0	3	3
54619	2020	31	31078	0	0	0	0	0	0	0	0	0	0	0	0	0	2
22691	2015	16	16110	0	0	0	0	0	0	0	0	0	0	0	0	0	3
35488	2019	6	6004	0	0	0	0	0	0	0	0	0	0	0	0	0	0
56728	2021	12	12066	1	1	2	1	1	1	2	2	0	1	0	3	15	3
53687	2020	30	30059	0	0	0	0	0	0	0	0	0	0	0	0	0	2
50222	2020	20	20281	0	0	0	0	0	0	0	0	0	0	0	0	0	1
14495	2018	5	5010	1	1	2	1	0	1	1	1	0	0	1	0	9	0
64009	2021	30	30162	0	0	0	0	1	0	0	0	0	0	0	0	1	2
5045	2017	7	7066	0	0	0	0	0	0	0	0	0	0	0	0	0	0
35569	2019	7	7014	0	0	0	0	0	0	0	0	0	0	0	0	0	0
62502	2021	24	24055	0	0	0	0	0	0	0	0	0	0	0	0	0	1

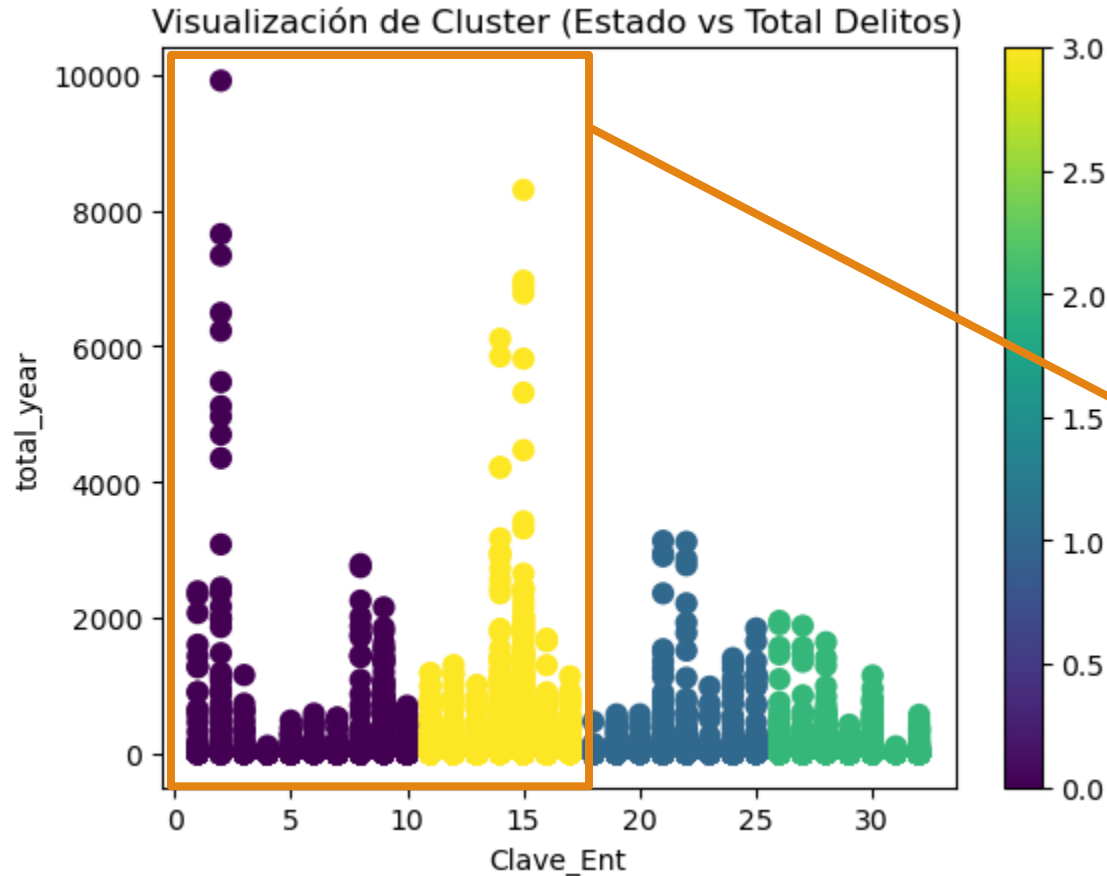
Se asignan los labels de cada clúster a los diferentes registros para identificarlos, con esto podemos agrupar los resultados de cada clúster que se ha formado.

# Resultados por clúster

```
labels
1    22740
3    17036
2    16016
0     9084
Name: count, dtype: int64
```



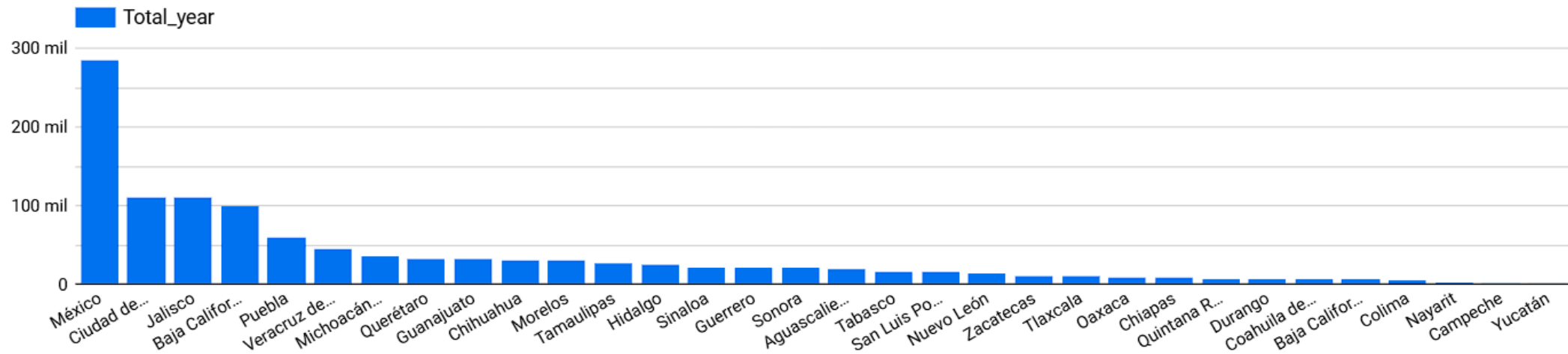
# Gráfico de dispersión



Este gráfico nos permite ver la distribución que hay de los 4 clústers (por color) y el comparativo entre “Clave\_Ent” es decir el Estado y el número total de delitos en el rango de años del dataset.

Como se puede apreciar los clúster 0 y 3 (los dos primeros) tienen los valores más altos en los totales de delitos, por lo que son en los que nos debemos enfocar para analizar el resultado, referente al objetivo que estamos desarrollando.

# Interpretación de datos



Como se puede ver en el gráfico de número de delitos totales por entidad (estado), vemos que justo el Estado de México, Ciudad de México, Jalisco y Baja California son los estados con mayores delitos relacionados con vehículos en el país.

Aun con estos datos seguramente podemos contar con el modelo una distribución diferente con respecto a la similitud que puede estar dada por coincidencias de datos por mes.

# Distribución para el clúster 0

	Clave_Ent	total_year	Entidad	labels
8	9	111587	Ciudad de México	0
1	2	100316	Baja California	0
7	8	30733	Chihuahua	0
0	1	20022	Aguascalientes	0
6	7	8963	Chiapas	0
9	10	7501	Durango	0
4	5	7424	Coahuila de Zaragoza	0
2	3	7409	Baja California Sur	0
5	6	6095	Colima	0
3	4	932	Campeche	0

Esta es la distribución del clúster 0 que tiene el estado con mayor número de delitos.

Con esta vista podemos determinar que la distribución de la data es muy similar por lo que se pueden hacer análisis independientes para generar estrategias segmentadas, probablemente por la data, la coincidencia esté en la temporalidad asociada a los delitos por mes.

Para este caso se sugiere trabajar estrategias comerciales especialmente para Ciudad de México, Baja California y Chihuahua.



# Distribución para el clúster 1

---

	Clave_Ent	total_year	Entidad	labels
3	21	59381	Puebla	1
4	22	32980	Querétaro	1
7	25	21871	Sinaloa	1
6	24	16359	San Luis Potosí	1
1	19	14224	Nuevo León	1
2	20	9819	Oaxaca	1
5	23	7660	Quintana Roo	1
0	18	2292	Nayarit	1

Esta es la distribución del clúster 1 que tiene el estado con mayor número de delitos.

Con esta vista podemos determinar que la distribución de la data es muy similar por lo que se pueden hacer análisis independientes para generar estrategias segmentadas, probablemente por la data, la coincidencia esté en la temporalidad asociada a los delitos por mes.

Para este caso se sugiere trabajar estrategias comerciales especialmente para Puebla, Querétaro y Sinaloa.



Google  
Data Studio

Importancia  
de Google  
Data Studio

---

# Importancia Google Data Studio

---

## 1. Transformación de Datos en Decisiones Estratégicas

Las empresas generan grandes volúmenes de datos provenientes de distintas fuentes (CRM, ERP, ecommerce, campañas de marketing, redes sociales, etc.).

Looker Studio permite:

- **Unificar y visualizar** datos de múltiples fuentes en un solo panel.
- **Identificar tendencias y patrones** de manera rápida.
- **Tomar decisiones informadas** basadas en métricas en tiempo real.

## 2. Fácil Acceso y Uso para Todos los Equipos

No todas las personas en la empresa tienen conocimientos técnicos en análisis de datos. Looker Studio facilita la democratización del acceso a la información con:

- **Dashboards interactivos y personalizables** sin necesidad de programación.
- **Interfaz intuitiva** para que cualquier usuario pueda generar reportes.
- **Acceso en la nube**, permitiendo consultar métricas desde cualquier dispositivo.

## 3. Automatización y Reducción del Trabajo Manual

Muchas empresas dependen de reportes manuales en Excel, lo que implica:

- ✗ **Errores humanos.**
  - ✗ **Tiempo perdido en consolidar datos.**
  - ✗ **Información desactualizada.**
- Con Looker Studio: ✓ **Reportes automatizados con datos en tiempo real.**
- ✓ **Integraciones con diversas plataformas sin necesidad de descargar archivos.**
  - ✓ **Mayor eficiencia operativa al eliminar procesos manuales.**

# Importancia Google Data Studio

---

## 4. Integración con Múltiples Fuentes de Datos

Looker Studio se conecta con herramientas clave como:

- **Google Analytics, Google Ads, Search Console** (para marketing digital).
- **BigQuery, MySQL, PostgreSQL** (para bases de datos empresariales).
- **Shopify, Facebook Ads, HubSpot, Salesforce, entre otros.** Esto permite analizar datos desde distintas áreas de la empresa en un solo lugar.

## 5. Monitoreo en Tiempo Real del Rendimiento del Negocio

Con Looker Studio, las empresas pueden:

- **Medir KPIs clave en tiempo real** (ventas, tráfico web, engagement, conversión, inventarios, etc.).
- **Detectar problemas y oportunidades de negocio** antes de que sea demasiado tarde.
- **Comparar períodos y analizar tendencias** para optimizar estrategias

## 6. Accesibilidad y Costo Cero

A diferencia de otras herramientas de BI costosas como Tableau o Power BI, **Looker Studio es gratuito** y no requiere licencias. Esto lo convierte en una **solución escalable y accesible** para empresas de cualquier tamaño.

## 7. Mejora de la Cultura Data-Driven

Implementar Looker Studio impulsa la **cultura de datos** dentro de la empresa, permitiendo que:

- Todos los equipos trabajen con **información basada en datos** en lugar de intuición.
- Se fomente la transparencia y alineación en objetivos empresariales.
- Se optimicen estrategias basadas en métricas reales.

Presentación

Comportamiento Anual

Comportamiento Mensual

Comportamiento por Entidad

Panorama de Violencia

El Pareto de la Data

Insights Comerciales

Conclusiones



## SEGUROS DE AUTOS

### ANALISIS DE DATOS DELITOS RELACIONADOS CON AUTOMOVILES EN MÉXICO 2015-2021

**El objetivo del análisis** es poder identificar los patrones de delitos, frecuencias, lugares, tipo de delito, con el fin de establecer parámetros estratégicos que permitan establecer rangos de precios de acuerdo al riesgo, así como entender las posibilidades de mejorar los servicios que ofrece la compañía de seguros para autos.



Por: Paulo Andrés Ossa Benítez  
Estudiante Profesión Ciencia de Datos EBAC  
16 de Enero de 2025, León, Guanajuato, México

# Visita el dashboard en tiempo real

[HTTPS://LOOKERSTUDIO.  
GOOGLE.COM/REPORTIN  
G/80821A5E-5BA7-45BA-  
986F-14FE014BB2B1](https://lookerstudio.google.com/reporting/80821a5e-5ba7-45ba-986f-14fe014bb2b1)



# Conclusiones y Recomendaciones

---

# Hallazgos

---

No hay una varianza muy grande entre año y año por lo que no se puede determinar que los delitos vayan a subir o bajar en gran cantidad.

---

Los meses donde suelen haber más delitos son Enero y Octubre

---

Los meses donde suelen haber menos delitos son Febrero y Abril

---

Los estados de México, CDMX, Jalisco, Baja California y Puebla con las entidades con la gran mayor representación de robo de vehículos en todo México.

---

El 68% de los robos de vehículos o partes de vehículos se hace sin el uso de la violencia

---

El Robo de vehículos es mucho mas común y usado que el robo de partes de vehículo.



# Recomendaciones



Si unimos todas las variables de riesgo, en efecto los valores de las pólizas pueden resultar mucho mas caras, sin embargo queda la oportunidad de prorratear con otros municipios que pueden pagar menos y el riesgo es mínimo.



Adicional a apegarse al valor de una póliza de seguro de vehículo se puede ver la oportunidad de hacer campañas de prevención contra robos, en especial en los meses y lugares donde hay más vulnerabilidad.



Igualmente se pueden hacer convenios con negocios que se encargan de la seguridad física de los vehículos o la inclusión de métodos de auto-apagado o uso de GPS para la localización, esto ayudaría a encontrar los vehículos y hacer operativos para la recuperación de los mismo disminuyendo los costes del seguro.



Finalmente en su mayoría no se usa la violencia, lo que quiere decir que el rango de pago por lesiones personales baja.