# UNIVERSIDAD EUROPEA DE MADRID

## ESCUELA DE ARQUITECTURA, INGENIERÍA Y DISEÑO

## GRADO EN INGENIERÍA INFORMÁTICA

## ADMINISTRACIÓN DE SISTEMAS

# Procesamiento en la nube con Spark

**DANIEL GUTIÉRREZ TORRES 22060943**

**YUL CADARSO CARDENAS 211c1129**

## CURSO 2024-2025

Universidad Europea

# Índice

# 1. Introducción

En este trabajo, configuraremos y prepararemos un entorno de trabajo para realizar el procesamiento de un dataset utilizando PySpark, una herramienta que permite el procesamiento de grandes volúmenes de datos, pudiendo realizar operaciones de análisis y transformación de datos distribuidos de manera eficiente.

# 2. Desarrollo

## 2.1 Obtención del dataset

En primer lugar, hemos utilizado Kaggle para obtener un dataset de ejemplo para la práctica sobre las tasas de criminalidad en España entre los años 2019 y 2021 (https://www.kaggle.com/datasets/marshuu/crime-rate-in-spain-2019-2021). Este conjunto de datos proporciona información detallada sobre las tasas de criminalidad y los tipos de delitos cometidos en las diferentes ciudades de España.

El dataset recoge el número de veces que a lo largo del año se han cometido cada tipo de delito en cada ciudad de España. Paralelo el dataset contiene los siguientes campos:

- **Location**
- **Year**
- **Crime**
- **Total cases**

Un ejemplo del dataset una fila del dataset sería:

```
Barcelona, 2021, Homicidios intencionados e intentos de asesinato, 137
```

Una vez obtenido el dataset, lo hemos subido a Lorca con el objetivo de trabajar con él desde el centro de computación de la universidad.

## 2.2 Configuración del entorno de trabajo

Para la realización de la práctica hemos utilizado LORCA, el centro de computación de la universidad, en donde nos hemos conectado

Primero se realizar la conexión del el local al remoto con el siguiente código:

```
ssh 221c1129yulbrynner@195.235.211.197 -p 1863
```

Luego se crea la carpeta donde se almacenará el notebook y los demás archivos con el siguiente código:

```
mkdir Procesamiento_Spark
```

Luego se envia el CSV obtenido desde Kaggle desde el local al remoto con el siguiente código:

```
scp -P 1863
/Users/yulcardenas/Desktop/Tasa_de_crimen_España_2019_2021.csv
221c1129yulbrynner@195.235.211.197:/home/221c1129yulbrynner/Pr
ocesamiento_Spark
```

Finalmente, se crea un túnel por el puerto 8086 para poder usar jupyter, instalado en el remoto, desde local con el siguiente código:

```
ssh -f -N -L 8086:localhost:8085
221c1129yulbrynner@195.235.211.197 -p 1863
```

## 2.3 Configuración de Spark

A continuación, comprobamos que spark está instalado con el siguiente código:

```
import findspark
findspark.init()

from pyspark.sql import SparkSession

spark =
SparkSession.builder.master("local[*]").appName("GoogleColabSpark
").getOrCreate()
print(spark.version)
```

Después importamos las librerías con las que trabajaremos:

```
import os
import findspark
from pyspark.sql import SparkSession
from pyspark.sql.functions import col, avg, max, desc
```

Acto seguido, inicializamos Spark con el siguiente código:

```
spark=SparkSession.builder.master("local[*]").appName("ProcesamientoCrimenEspaña").getOrCreate()
```

## 2.4 Carga del dataset

Cargamos el dataset desde la ruta donde se encuentra el archivo

```
ruta_csv="Tasa_de_crimen_España_2019_2021.csv"
datos_crimenes=spark.read.csv(ruta_csv, header=True, inferSchema=True)
```

Podremos visualizar su estructura con el siguiente comando

```
datos_crimenes.printSchema()
```

## 2.5 Procesamiento de datos con PySpark

Una vez cargado el dataset, procedemos a utilizar PySpark para realizar un análisis y procesado de los datos, con el objetivo de hallar coincidencias, patrones y obtener métricas que nos permitan extraer conclusiones de los mismos.

### Tasa de crimen promedio por año

En primer lugar calcularemos el promedio anual de crímenes realizados cada año, con el objetivo de analizar las tendencias y patrones de los datos a lo largo de estos 3 años.

El código utilizado agrupa los datos por el campo “Year” y calcula el promedio de la columna “Total cases” mediante la función avg().

```
tasa_promedio_anual =
datos_crimenes.groupBy("Year").agg(avg(col("Total
cases")).alias("Tasa_promedio"))
tasa_promedio_anual.show()
```

El resultado de este, el cual podemos visionarlo con el comando show(), será el siguiente:

```
+----+----------------+
|Year|    Tasa_promedio|
+----+----------------+
|2019|6562.745535714285|
|2020|4991.915178571428|
|2021|5803.651785714285|
+----+----------------+
```

## Ciudad con mayor tasa de crimen en cada año

En segundo lugar, analizaremos las localidades más afectadas por la criminalidad en cada año. Para ello, agruparemos los datos por año y ciudad, calcular la tasa de crimen total para cada combinación y seleccionar aquella ciudad que registra el valor más alto en cada año.

El código utilizado agrupa los datos por los campos “Year” y “Location” mediante `groupBy()`, para crear combinaciones únicas de año y localidad. Luego, aplica la función max() sobre la columna “Total cases” para calcular el número máximo de casos registrados para cada grupo y asignando este valor a una nueva columna que he llamado “Tasa_maxima”. A continuación, ordena los resultados en orden descendente con orderBy(desc("Tasa_maxima")), mostrando las ciudades con las tasas más altas de criminalidad en la parte de arriba de la tabla.

```
ciudad_maxim_crimen = datos_crimenes.groupBy("Year",
"Location").agg(max(col("Total cases")).alias("Tasa_maxima"))
ciudad_maxim_crimen.orderBy(desc("Tasa_maxima")).show()
```

EL resultado de este, el cual podemos visionarlo con el comando show(), será el siguiente:

```
+----+---------+-----------+
|Year| Location|Tasa_maxima|
+----+---------+-----------+
|2019|   Madrid|     205312|
|2021|   Madrid|     197970|
|2020|   Madrid|     181917|
|2019|Barcelona|     165901|
|2021|Barcelona|     152565|
|2020|Barcelona|     135216|
|2021| Valencia|      74380|
|2019| Valencia|      68853|
|2020| Valencia|      64822|
|2021| Alicante|      49211|
|2021|  Seville|      48461|
|2019|  Seville|      48365|
|2019| Alicante|      48178|
|2021|   Malaga|      47803|
|2020|  Seville|      46704|
|2020| Alicante|      45607|
|2019|   Malaga|      44516|
|2020|   Malaga|      42372|
|2021|   Murcia|      37088|
|2020|   Murcia|      32819|
+----+---------+-----------+
```

## Tipo de crimen con tasa más baja

En tercer lugar, analizaremos los delitos menos frecuentes dentro del dataset, con el objetivo de obtener aquellos crímenes que tienen menor incidencia en términos de criminalidad.

El código utilizado agrupa los datos por el campo “Crime”, que contiene los diferentes tipos de crímenes registrados en el dataset. Para cada tipo de crimen calculará el promedio de los casos totales utilizando la función de agregación avg(). Esto permitirá obtener el dato de la frecuencia promedio con la que ocurre cada delito. La tasa calculada se almacena en una nueva columna que llamaremos “Tasa_promedio”.

EL resultado de este, el cual podemos visionarlo con el comando show(), será el siguiente:

```
tipo_crimen_minim=datos_crimenes.groupBy("Crime").agg(avg(col("Tot
al cases")).alias("Tasa_promedio"))
tipo_crimen_minim.orderBy("Tasa_promedio").show()
```

```
+--------------------+-----------------+
|               Crime|     Tasa_promedio|
+--------------------+-----------------+
|          Kidnapping|           3.1875|
|Intentional homic...|11.291666666666666|
|Intentional homic...|            34.25|
|Sexual assault wi...| 70.70833333333333|
|Other crimes agai...|           495.625|
|Crimes against fr...| 566.3541666666666|
|Serious and less ...| 663.4166666666666|
|    Drug trafficking| 681.3958333333334|
|       Vehicle theft|1345.6041666666667|
|Robbery with viol...|2801.8541666666665|
|Robberies with fo...|3143.9583333333335|
|Burglaries with f...|3535.2083333333335|
|               Theft|23808.645833333332|
|Other criminal of...|43843.958333333336|
+--------------------+-----------------+
```

# 3. Resultados obtenidos

## 3.1 Guardar los datos generados del procesamiento con Pyspark

Todos los datos obtenidos y mostrados en el apartado anterior con el comando show(), los almacenaremos en formato json con el objetivo de poder visualizarlos.

```
ruta_salida_tasa_promedio = "tasa_promedio_anual.csv"
tasa_promedio_anual.coalesce(1).write.csv(ruta_salida_tasa_promedio,
```

```
header=True, mode="overwrite")

ruta_salida_ciudad_max = "ciudad_maximo_crimen.csv"
ciudad_maxim_crimen.coalesce(1).write.csv(ruta_salida_ciudad_max,
header=True, mode="overwrite")

ruta_salida_tipo_crimen_min = "tipo_crimen_minimo.csv"
tipo_crimen_minim.coalesce(1).write.csv(ruta_salida_tipo_crimen_min,
header=True, mode="overwrite")
```

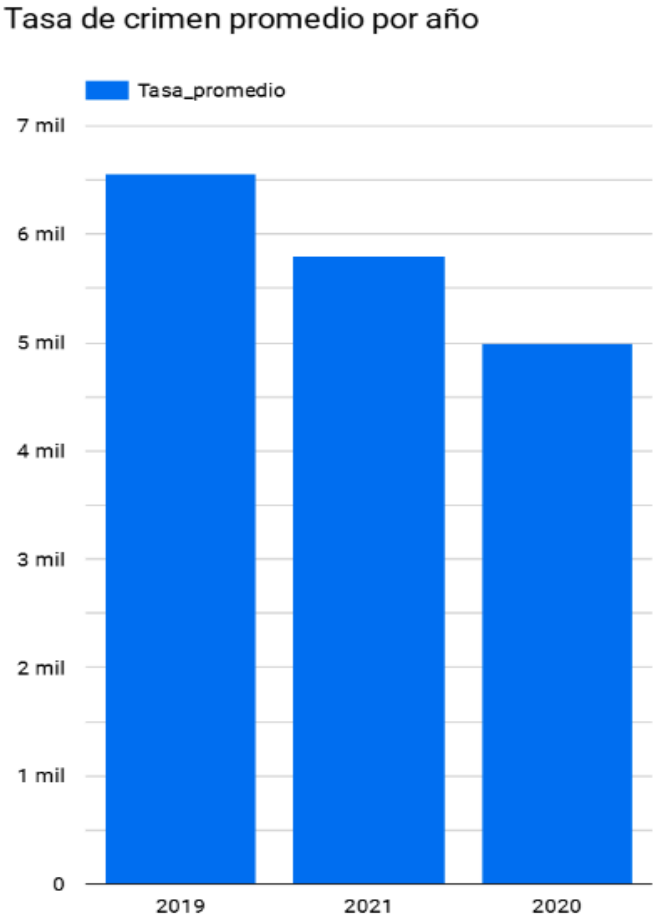
## 3.2 Procesamiento de datos con PySpark

Para visualizar los datos utilizaremos la herramienta de visualización Lookerstudio de Google, para mostrar los datos procesados en los puntos anteriores con PySpark

### Tasa de crimen promedio por año

La gráfica muestra la evolución de la criminalidad entre los años 2019, 2020 y 2021. Se observa que la tasa promedio en 2019 fue la más alta con un valor cercano a los 6 millones, mientras que en 2020 se redujo ligeramente con un valor por debajo de los 5 millones. En 2021, en cambio, la tasa promedio volvió a aumentar, alcanzando un valor ligeramente superior al de 2020 pero aún por debajo de los valores de 2019.

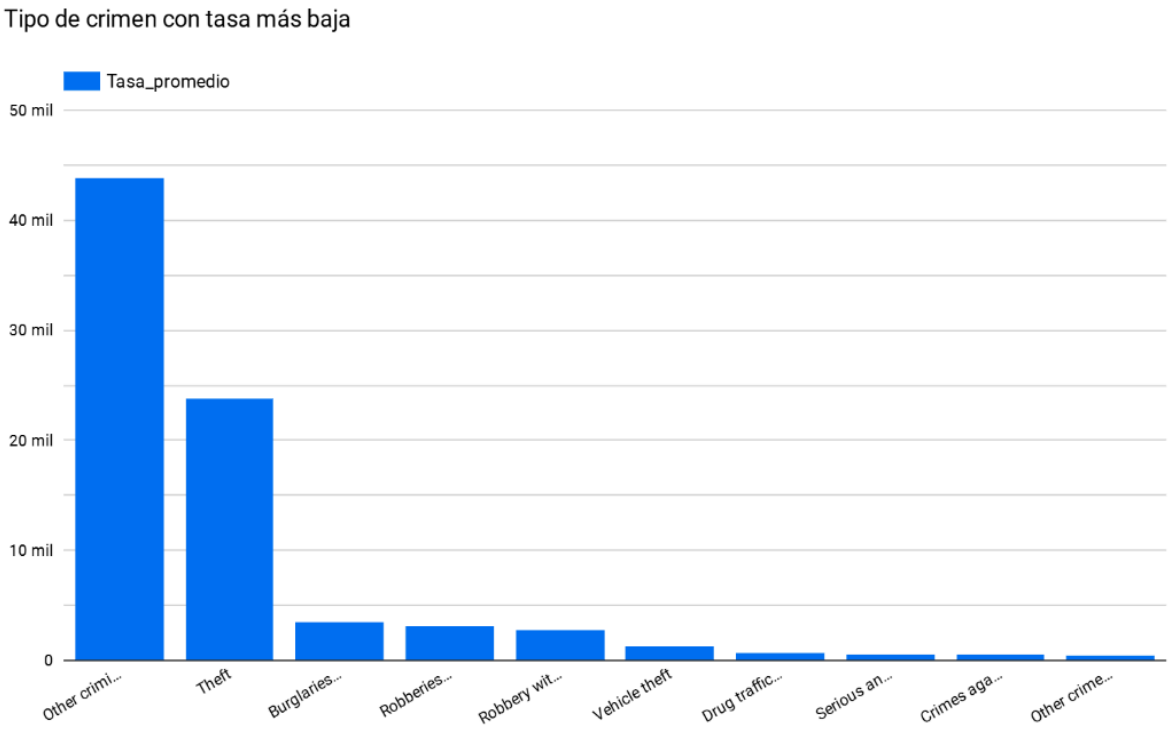
Este análisis puede reflejar cómo el número de crímenes cambió en el tiempo. Esto puede relacionarse con hechos históricos ocurridos en 2020 como la pandemia mundial que hizo que la gente no saliera de casa y en consecuencia no saliera de casa y cometiera crímenes.




### Tipo de crimen con tasa más baja

La gráfica muestra los tipos de crímenes clasificados por su tasa promedio a lo largo de estos 3 años analizados. Podemos observar que los delitos etiquetado como "Otros crimenes" tienen la mayor incidencia, en consecuencia probablemente al englobar variedad de delitos con poca incidencia, lo que eleva su representación en la gráfica.

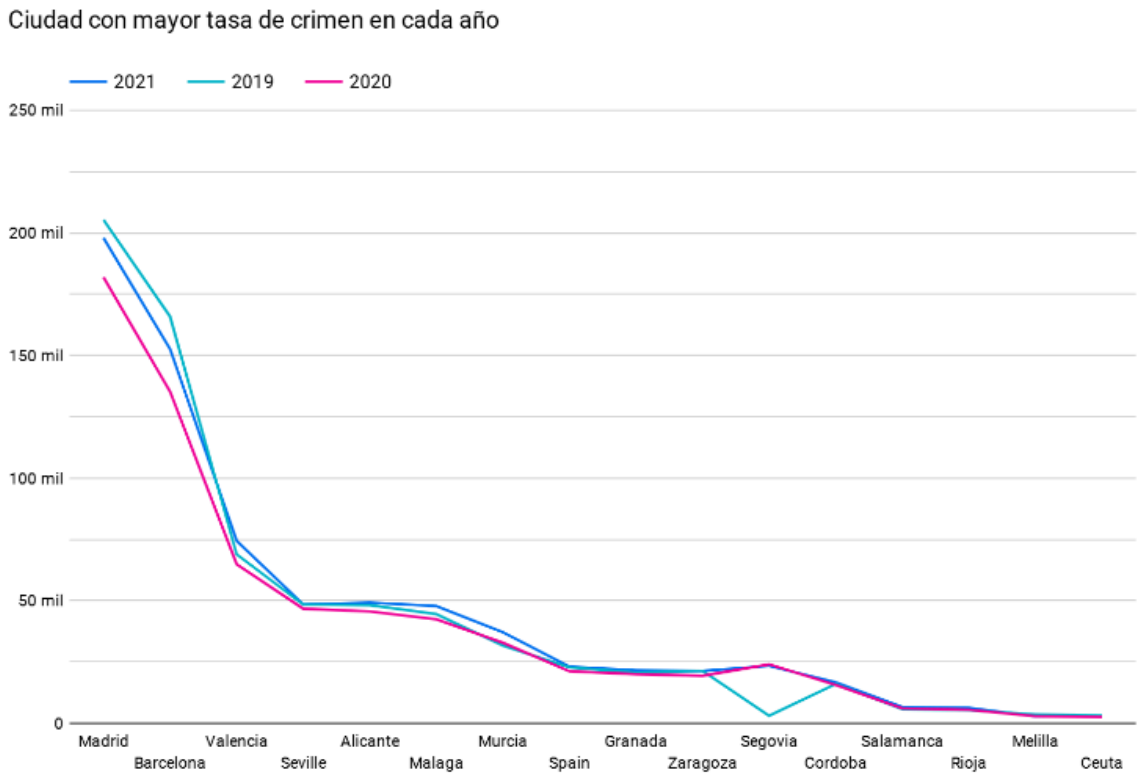
Por otro lado, los robos ocupan el segundo lugar, destacándose como un delito común con una tasa promedio alta. El resto de los tipos de crímenes, como "Robo en viviendas," "Robos," "Robo de vehículos," y "Tráfico de drogas" muestran tasas mucho más bajas, lo que refleja una menor incidencia de estos delitos en el conjunto de datos analizados.




## Ciudad con mayor tasa de crimen en cada año

La gráfica muestra las localidades más afectadas por la criminalidad durante los años 2019, 2020 y 2021, representando la tasa de criminalidad más alta registrada en cada ciudad. Los datos reflejan cómo la incidencia del crimen varía entre las principales ciudades españolas, destacando a Madrid y Barcelona como las localidades con mayores tasas de criminalidad en los tres años analizados. Por otro lado, Salamanca, La Rioja, Melilla y Ceuta presentan las tasas más bajas, indicando una criminalidad baja en estas áreas. Estos resultados obtenidos tienen sentido al analizar la tasa de población de cada localidad.

Por otro lado, podemos apreciar patrones como cuando el año 2020 se muestra una disminución en la criminalidad en la mayoría de las ciudades, posiblemente relacionada con factores externos como la pandemia acontecida en ese año. Además en 2021 podemos apreciar como los registros vuelven a incrementarse, una vez superado la pandemia.




# 4. Conclusiones

En conclusión PySpark es una buena opción para analizar y procesar grandes volúmenes de datos de manera eficiente y óptima. Con este análisis hemos podido analizar las tendencias de criminalidad a lo largo del tiempo y en diferentes localidades en un dataset relativamente grande con más de seis mil registros.

Tras analizar los resultados del procesamiento del dataset, podemos llegar a varias conclusiones. En primer lugar, observamos que algunos crímenes, como el robo, presentan tasas significativamente más altas en comparación con otros delitos, como el tráfico de drogas y el robo de vehículos. Además, el análisis de la evolución de las tasas de criminalidad en los años estudiados muestra cómo estos datos han podido verse alterados por la pandemia de 2020. Por último, recalcar el dato de que la mayoría de los crímenes se concentran en las grandes ciudades, que tienen una mayor tasa de población.

Con estas conclusiones del procesamiento y posterior análisis de los datos, se podrían tomar decisiones en políticas públicas o estrategias de prevención, con el objetivo de enfocar los recursos de manera más eficiente, priorizando los tipos de crímenes que requieren atención urgente.