

TRABAJO GRUPAL Proyecto: FINISHED

ProyectoFinalGRupal

Importaciones Realizadas en el Ecuador

## PROYECTO BIMESTRAL

### INTEGRANTES

- DIANA ROCIO BERMEO
- CARLOS VALLADARES
- JORGE BELTRAN

Took 0 sec. Last updated by anonymous at July 25 2025, 1:01:47 PM.

### Objetivo del Proyecto

FINISHED

El objetivo general de este proyecto es **enriquecer y analizar** el conjunto de datos de importaciones del Ecuador, bajo el régimen general, correspondiente al **primer semestre de 2024 (enero - junio)**.

Se busca no solo comprender los datos internos del comercio registrado, sino también **incorporar variables externas** y de contexto económico internacional para obtener una visión más completa del fenómeno importador.

Took 0 sec. Last updated by anonymous at July 25 2025, 1:03:16 PM.

### Datos o Archivos Utilizados

FINISHED

Took 0 sec. Last updated by anonymous at July 25 2025, 1:04:27 PM.

### CARGA DEL DATA DESDE LA BASE DE DATOS

FINISHED

Took 0 sec. Last updated by anonymous at July 25 2025, 12:59:08 PM.

#### **importaciones**

FINISHED

Declaraciones aduaneras bajo régimen general (SENAE).

Archivo: `reducido_senae_importacion_general_enero_junio_2024.csv` (renombrado como `importaciones` ).

Took 0 sec. Last updated by anonymous at July 25 2025, 1:05:13 PM.

# ProyectoFinalGRupa

```
val url="jdbc:mysql://localhost:3306/importacionesZepelin"
val table="(SELECT * from importacionesData ) AS subconsulta1"
val propiedades = new java.util.Properties()
propiedades.setProperty("user","root")
propiedades.setProperty("password",".Alej1997")
propiedades.setProperty("driver","com.mysql.cj.jdbc.Driver")
val data1 = spark.read.jdbc(url,table,propiedades)
```

url: String = jdbc:mysql://localhost:3306/importacionesZepelin  
table: String = (SELECT \* from importacionesData ) AS subconsulta1  
propiedades: java.util.Properties = {user=root, password=.Alej1997, driver=com.mysql.cj.jdbc.D  
river}  
data1: org.apache.spark.sql.DataFrame = [TIPO\_IMPORTACION: string, ESTADO\_DECLARACION: string  
... 31 more fields]

Took 2 sec. Last updated by anonymous at July 25 2025, 12:59:10 PM.

val data = data1.drop("PIB\_PERCAPITA") FINISHED

data: org.apache.spark.sql.DataFrame = [TIPO\_IMPORTACION: string, ESTADO\_DECLARACION: string .  
.. 30 more fields]

Took 1 sec. Last updated by anonymous at July 25 2025, 12:59:11 PM.

z.show(data)

SPARK JOB (http://Ubuntu2204:4040/jobs/job?id=42) FINISHED



settings ▼

TIPO_IMPORTACION	ESTADO_DECLARACION	FEC_INGRESO	DISTRITO	
IMP.GRAL.	10-SALIDA AUTORIZADA	2024-06-12 00:00:00.0	019-GUAYAQUIL - AEREO	
IMP.GRAL.	10-SALIDA AUTORIZADA	2024-05-29 00:00:00.0	019-GUAYAQUIL - AEREO	
IMP.GRAL.	10-SALIDA AUTORIZADA	2024-05-15 00:00:00.0	019-GUAYAQUIL - AEREO	

**Output is truncated** to 102400 bytes. [Learn more about ZEPPELIN\\_INTERPRETER\\_OUTPUT\\_LIMIT](#)



Took 2 sec. Last updated by anonymous at July 25 2025, 12:59:13 PM.

## CSV COMPLEMENTARIOS

FINISHED

Took 0 sec. Last updated by anonymous at July 25 2025, 12:59:13 PM.

countries.csv

Archivo con los códigos de país ISO2 e ISO3, utilizado para normalizar la información y facilitar los filtros en los Grupos

# ProyectoFinalGRupa

Took 0 sec. Last updated by anonymous at July 25 2025, 1:06:16 PM.

## COUNTRIES

FINISHED

Took 0 sec. Last updated by anonymous at July 25 2025, 12:59:14 PM.

```
val countriesDF = spark
  .read
  .option("inferschema","true")
  .option("header", "true")
  .option("delimiter", ",")
  .csv("/home/vboxuser/zeppelin-0.11.1-bin-all/ProyectoFinal/countries.csv")
```

countriesDF: org.apache.spark.sql.DataFrame = [name: string, alpha2: string ... 8 more fields]

Took 2 sec. Last updated by anonymous at July 25 2025, 12:59:16 PM.

```
z.show(countriesDF)
```

SPARK JOB (http://Ubuntu2204:4040/jobs/job?id=45) FINISHED

settings ▼

name ▼	alpha2 ▼	alpha3 ▼	ccTLD ▼	continent ▼
Afghanistan	AF	AFG	null	+93
Albania	AL	ALB	null	+355
Algeria	DZ	DZA	null	+213
American Samoa	AS	ASM	null	+1 684
Andorra	AD	AND	null	+376
Angola	AO	AGO	null	+244
Anguilla	AI	AIA	null	+1 264
Antarctica	AQ	ATA	null	+672
Antigua and Barbuda	AG	ATG	null	+1 268

Took 1 sec. Last updated by anonymous at July 25 2025, 12:59:17 PM.

PIB\_per\_capita.csv

FINISHED

Datos del Banco Mundial sobre el PIB per cápita (PPP, US\$ constantes) del país

# ProyectoFinalGRupa

Took 0 sec. Last updated by anonymous at July 25 2025, 1:07:12 PM.

SPARK JOB (http://Ubuntu2204:4040/jobs/job?id=46) FINISHED

```
val pibDF= spark
  .read
  .option("header", "true")
  .option("inferSchema", "false")
  .option("delimiter", ",")
  .csv("/home/vboxuser/zeppelin-0.11.1-bin-all/ProyectoFInal/PIB_per_capita.csv")

// 2. Lista de columnas de años (de 1960 a 2024)
val yearColumns = (1960 to 2024).map(_.toString)

// 3. Convertir columnas de años a Double y reemplazar null por 0.0
val pibDFlimpio = yearColumns.foldLeft(pibDF) { (df, year) =>
  df.withColumn(year, coalesce(col(year).cast("double"), lit(0.0)))
}

pibDFlimpio.printSchema()
```

```
root
|-- Country Name: string (nullable = true)
|-- Country Code: string (nullable = true)
|-- Indicator Name: string (nullable = true)
|-- Indicator Code: string (nullable = true)
|-- 1960: double (nullable = false)
|-- 1961: double (nullable = false)
|-- 1962: double (nullable = false)
|-- 1963: double (nullable = false)
|-- 1964: double (nullable = false)
|-- 1965: double (nullable = false)
|-- 1966: double (nullable = false)
|-- 1967: double (nullable = false)
|-- 1968: double (nullable = false)
|-- 1969: double (nullable = false)
|-- 1970: double (nullable = false)
|-- 1971: double (nullable = false)
|-- 1972: double (nullable = false)
|-- 1973: double (nullable = false)
|-- 1974: double (nullable = false)
|-- 1975: double (nullable = false)
|-- 1976: double (nullable = false)
|-- 1977: double (nullable = false)
|-- 1978: double (nullable = false)
|-- 1979: double (nullable = false)
|-- 1980: double (nullable = false)
|-- 1981: double (nullable = false)
|-- 1982: double (nullable = false)
|-- 1983: double (nullable = false)
|-- 1984: double (nullable = false)
|-- 1985: double (nullable = false)
|-- 1986: double (nullable = false)
|-- 1987: double (nullable = false)
|-- 1988: double (nullable = false)
|-- 1989: double (nullable = false)
|-- 1990: double (nullable = false)
|-- 1991: double (nullable = false)
|-- 1992: double (nullable = false)
|-- 1993: double (nullable = false)
|-- 1994: double (nullable = false)
|-- 1995: double (nullable = false)
|-- 1996: double (nullable = false)
|-- 1997: double (nullable = false)
|-- 1998: double (nullable = false)
|-- 1999: double (nullable = false)
|-- 2000: double (nullable = false)
|-- 2001: double (nullable = false)
|-- 2002: double (nullable = false)
|-- 2003: double (nullable = false)
|-- 2004: double (nullable = false)
|-- 2005: double (nullable = false)
|-- 2006: double (nullable = false)
|-- 2007: double (nullable = false)
|-- 2008: double (nullable = false)
|-- 2009: double (nullable = false)
|-- 2010: double (nullable = false)
|-- 2011: double (nullable = false)
|-- 2012: double (nullable = false)
|-- 2013: double (nullable = false)
|-- 2014: double (nullable = false)
|-- 2015: double (nullable = false)
|-- 2016: double (nullable = false)
|-- 2017: double (nullable = false)
|-- 2018: double (nullable = false)
|-- 2019: double (nullable = false)
|-- 2020: double (nullable = false)
|-- 2021: double (nullable = false)
|-- 2022: double (nullable = false)
|-- 2023: double (nullable = false)
|-- 2024: double (nullable = false)
```

Took 9 sec. Last updated by anonymous at July 25 2025, 12:59:26 PM.

z.show(pibDFlimpio)


SPARK JOB (http://Ubuntu2204:4040/jobs/job?id=47) FINISHED



Country Name ▾	Country Code ▾	Indicator Name ▾	Indicator Code ▾	1960
Aruba	ABW	Inflation, consumer	FP.CPI.TOTL.ZG	0.0

# ProyectoFinalGRupa

		prices (annual %)		
Africa Eastern and Southern	AFE	Inflation, consumer prices (annual %)	FP.CPI.TOTL.ZG	0.0
Afghanistan	AFG	Inflation, consumer prices (annual %)	FP.CPI.TOTL.ZG	0.0
Africa Western and Central	AFW	Inflation, consumer prices (annual %)	FP.CPI.TOTL.ZG	0.0
Angola	AGO	Inflation, consumer prices (annual %)	FP.CPI.TOTL.ZG	0.0

**Output is truncated** to 102400 bytes. [Learn more about ZEPPELIN\\_INTERPRETER\\_OUTPUT\\_LIMIT](#) 

Took 2 sec. Last updated by anonymous at July 25 2025, 12:59:28 PM.

**inflacion\_mercancias\_pais.csv** FINISHED

Base global con las **tasas de inflación anual por país** del país exportador.

Took 0 sec. Last updated by anonymous at July 25 2025, 1:08:17 PM.

```
val inflacion= spark
  .read
  .option("header", "true")
  .option("inferSchema", "false")
  .option("delimiter", ",")
  .csv("/home/vboxuser/zeppelin-0.11.1-bin-all/ProyectoFinal/inflacion_mercancias_pais.csv")


// 2. Lista de columnas de años (de 1960 a 2024)
val yearColumns = (1960 to 2024).map(_.toString)

// 3. Aplicar transformación: convertir a Double y reemplazar null por 0.0
val inflacionDF = yearColumns.foldLeft(inflacion) { (df, year) =>
  df.withColumn(year, coalesce(col(year).cast("double"), lit(0.0)))
}

// 4. Eliminar columna innecesaria 'Column1' si existe
val inflacionDFlimpio = if (inflacionDF.columns.contains("Column1")) {
  inflacionDF.drop("Column1")
} else {
  inflacionDF
}

// 5. Verificar el resultado
inflacionDFlimpio.printSchema()

root
|-- Country Name: string (nullable = true)
|-- Country Code: string (nullable = true)
|-- Indicator Name: string (nullable = true)
|-- Indicator Code: string (nullable = true)
|-- 1960: double (nullable = false)
|-- 1961: double (nullable = false)
```

 SPARK JOB (http://Ubuntu2204:4040/jobs/job?id=48) FINISHED

# ProyectoFinalGRupa

```
|-- 1962: double (nullable = false)
|-- 1963: double (nullable = false)
|-- 1964: double (nullable = false)
|-- 1965: double (nullable = false)
|-- 1966: double (nullable = false)
|-- 1967: double (nullable = false)
|-- 1968: double (nullable = false)
|-- 1969: double (nullable = false)
|-- 1970: double (nullable = false)
|-- 1971: double (nullable = false)
```

Took 6 sec. Last updated by anonymous at July 25 2025, 12:59:34 PM.

z.show(inflacionDFlimpio)

SPARK JOB (http://Ubuntu2204:4040/jobs/job?id=49) FINISHED



settings ▼

Country Name ▼	Country Code ▼	Indicator Name ▼	Indicator Code ▼	1950
Aruba	ABW	PIB per c?pita, PPA (\$ a precios internacionales actuales)	NY.GDP.PCAP.PP.CD	0.0
null	AFE	PIB per c?pita, PPA (\$ a precios internacionales actuales)	NY.GDP.PCAP.PP.CD	0.0
Afganist?n	AFG	PIB per c?pita, PPA (\$ a precios internacionales actuales)	NY.GDP.PCAP.PP.CD	0.0

Output is truncated to 102400 bytes. Learn more about ZEPPELIN\_INTERPRETER\_OUTPUT\_LIMIT



Took 2 sec. Last updated by anonymous at July 25 2025, 12:59:36 PM.

## ENREQUICIMIENTO DE DATOS

FINISHED

### Reemplazar PAIS\_ORIGEN con su código ISO3

Took 0 sec. Last updated by anonymous at July 25 2025, 12:59:36 PM.

```
val df_joined1 = data
  .join(countriesDF, data("IS02") === countriesDF("alpha2"), "left")
  .withColumn("PAIS_ORIGEN", col("alpha3"))
  .drop("alpha2", "alpha3", "IS02")
```

FINISHED

df\_joined1: org.apache.spark.sql.DataFrame = [TIPO\_IMPORTACION: string, ESTADO\_DECLARACION: string ... 37 more fields]

Took 0 sec. Last updated by anonymous at July 25 2025, 12:59:37 PM.

z.show(df\_joined1)

SPARK JOB FINISHED

settings

ProyectoFinalGRupa

TIPO_IMPORTACION	ESTADO_DECLARACION	FEC_INGRESO	DISTRITO
IMP.GRAL.	10-SALIDA AUTORIZADA	2024-06-12 00:00:00.0	019-GUAYAQUIL - AEREO
IMP.GRAL.	10-SALIDA AUTORIZADA	2024-05-29 00:00:00.0	019-GUAYAQUIL - AEREO
IMP.GRAL.	10-SALIDA AUTORIZADA	2024-05-15 00:00:00.0	019-GUAYAQUIL - AEREO

Output is truncated to 102400 bytes. Learn more about ZEPPELIN\_INTERPRETER\_OUTPUT\_LIMIT

Took 1 sec. Last updated by anonymous at July 25 2025, 12:59:39 PM.

## Relación entre los Archivos

FINISHED

La conexión entre los archivos permite **enriquecer el dataset principal de importaciones** pasando de un análisis descriptivo a uno explicativo y comparativo. Las relaciones clave son:

### Integración del PIB per cápita

- Una vez normalizado el país de origen a ISO3, se realiza un **join** con `PIB_per_capita.csv`.
- Esto permite vincular las importaciones con el **nivel de desarrollo económico** del país exportador.

### Integración de la Inflación

- Se une la tasa de inflación anual de `inflacion_mercancias_pais.csv` al dataset principal usando ISO3.
- Permite analizar si los precios de importación están **influenciados por condiciones inflacionarias externas**.

Took 0 sec. Last updated by anonymous at July 25 2025, 1:09:46 PM.

ProyectoFinalGRupal

```
val pibresultante= pibDFlimpio.select(
  col("Country Code").as("PAIS_ORIGEN"),
  col("2023").as("PIB_PERCAPITA")
)
```

FINISHED

```
val dffinal= df_joined1.join(pibresultante, Seq("PAIS_ORIGEN"), "left")

pibresultante: org.apache.spark.sql.DataFrame = [PAIS_ORIGEN: string, PIB_PERCAPITA: double]
dffinal: org.apache.spark.sql.DataFrame = [PAIS_ORIGEN: string, TIPO_IMPORTACION: string ... 3
8 more fields]
```

Took 1 sec. Last updated by anonymous at July 25 2025, 12:59:41 PM.

z.show(dffinal)

SPARK JOB FINISHED



settings ▼

PAIS_ORIGEN	TIPO_IMPORTACION	ESTADO_DECLARACI..	FEC_INGRESO	
KOR	IMP.GRAL.	10-SALIDA AUTORIZADA	2024-06-12 00:00:00.0	
COL	IMP.GRAL.	10-SALIDA AUTORIZADA	2024-05-29 00:00:00.0	
USA	IMP.GRAL.	10-SALIDA AUTORIZADA	2024-05-15 00:00:00.0	

Output is truncated to 102400 bytes. Learn more about  
ZEPELIN\_INTERPRETER\_OUTPUT\_LIMIT



Took 3 sec. Last updated by anonymous at July 25 2025, 12:59:44 PM.

```
val inflacion= inflacionDFlimpio.select(
  col("Country Code").as("PAIS_ORIGEN"),
  col("2023").as("INFLACION")
)
```

FINISHED

```
val dfPuroo= dffinal.join(inflacion, Seq("PAIS_ORIGEN"), "left")

inflacion: org.apache.spark.sql.DataFrame = [PAIS_ORIGEN: string, INFLACION: double]
dfPuroo: org.apache.spark.sql.DataFrame = [PAIS_ORIGEN: string, TIPO_IMPORTACION: string ... 3
9 more fields]
```

Took 1 sec. Last updated by anonymous at July 25 2025, 12:59:45 PM.

z.show(dfPuroo)

SPARK JOB FINISHED



settings ▼

PAIS_ORIGEN ▼	TIPO_IMPORTACION ▼	ESTADO_DECLARACION ▼	FEC_INGRESO	≡
KOR	IMP.GRAL.	10-SALIDA AUTORIZADA	2024-06-12 00:00:00.0	
COL	IMP.GRAL.	10-SALIDA AUTORIZADA	2024-05-29 00:00:00.0	
USA	IMP.GRAL.	10-SALIDA AUTORIZADA	2024-05-15 00:00:00.0	

**Output is truncated** to 102400 bytes. [Learn more about ZEPPELIN\\_INTERPRETER\\_OUTPUT\\_LIMIT](#)

Took 4 sec. Last updated by anonymous at July 25 2025, 12:59:49 PM.

dfPuroo.printSchema

FINISHED

```
root
|-- PAIS_ORIGEN: string (nullable = true)
|-- TIPO_IMPORTACION: string (nullable = true)
|-- ESTADO_DECLARACION: string (nullable = true)
|-- FEC_INGRESO: timestamp (nullable = true)
|-- DISTRITO: string (nullable = true)
|-- REGIMEN: string (nullable = true)
|-- SUBPARTIDA: long (nullable = true)
|-- CODIGO_COMPLEMENTARIO: integer (nullable = true)
|-- CODIGO_SUPLEMENTARIO: integer (nullable = true)
|-- DESCRIPCION_ARANCELARIA: string (nullable = true)
|-- CODIGO_LIBERACION: string (nullable = true)
|-- TRATAMIENTO_PREFERENCIAL: string (nullable = true)
|-- CONVENIO_INTERNACIONAL: string (nullable = true)
|-- TIPO_UNIDAD_FISICA: string (nullable = true)
|-- TIPO_UNIDAD_COMERCIAL: string (nullable = true)
|-- PESO_NETO: double (nullable = true)
|-- ...
```

Took 1 sec. Last updated by anonymous at July 25 2025, 12:59:50 PM.

APARTADO DE CONSULTAS

FINISHED

Took 0 sec. Last updated by anonymous at July 25 2025, 1:10:34 PM.

FINISHED

# Consulta 1: Ranking de Socios Comerciales por Valor y Eficiencia Tributaria

## ProyectoFinalGRupa

¿Quiénes son nuestros principales socios comerciales y qué tan eficientes somos recaudando impuestos de ellos?

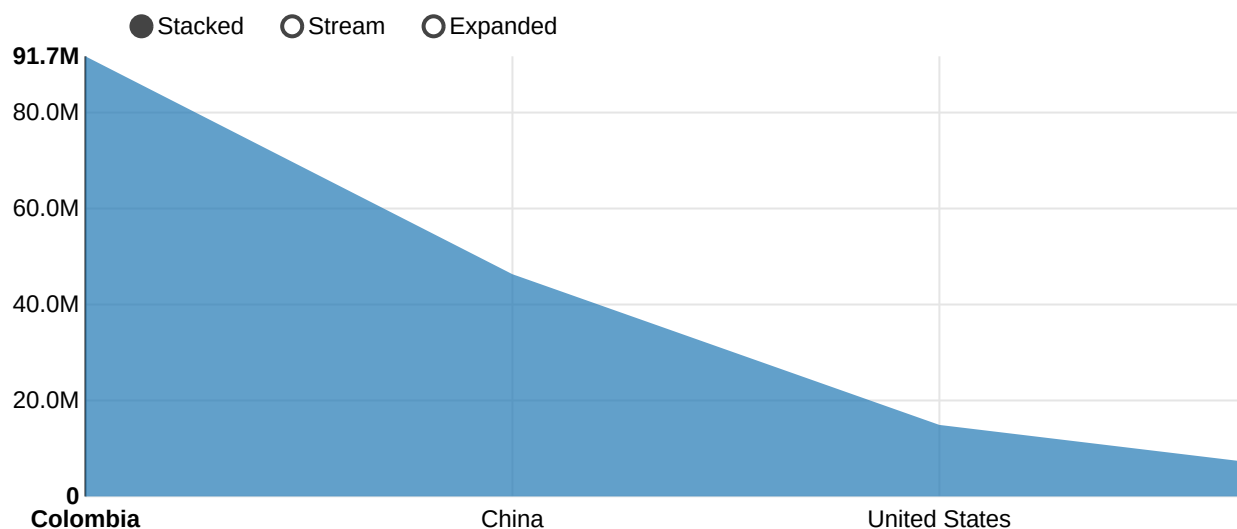
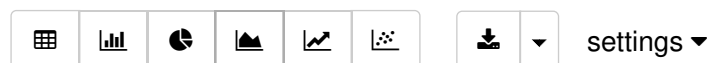
Took 0 sec. Last updated by anonymous at July 25 2025, 12:59:50 PM.

```
// Importa las funciones necesarias
import org.apache.spark.sql.functions.{sum, count, desc, col, round}

// Agrupamos por el nombre del país
val rankingPaíses = dfPuroo.groupBy("name")
  .agg(
    sum("CIF").as("Valor_CIF_Total"),
    sum("CARGA_TRIBUTARIA_TOTAL").as("Tributos_Totales"),
    count("*").as("Numero_Importaciones")
  )
  .withColumn("Porcentaje_Tributario", round(($"Tributos_Totales" / $"Valor_CIF_Total") * 100,
    .orderBy(desc("Valor_CIF_Total"))
    .limit(6)

// Muestra el resultado para visualización
z.show(rankingPaíses)
```

SPARK JOB FINISHED



```
import org.apache.spark.sql.functions.{sum, count, desc, col, round}
rankingPaíses: org.apache.spark.sql.Dataset[org.apache.spark.sql.Row] = [name: string, Valor_CIF_Total: double ... 3 more fields]
```

Took 6 sec. Last updated by anonymous at July 25 2025, 12:59:57 PM.

# Consulta 2: Evolución Mensual de las Importaciones y la Recaudación

FINISHED

¿Cómo varía el volumen de importaciones y la recaudación de impuestos mes a mes?

Took 0 sec. Last updated by anonymous at July 25 2025, 12:59:57 PM.

# ProyectoFinalGRupa

```
import org.apache.spark.sql.functions.{sum, window, date_format}
val evolutionMensual = dfPuroo.groupBy(date_format($"FEC_INGRESO", "yyyy-MM").as("Mes"))
  .agg(
    sum("CIF").as("Valor_CIF_Total"),
    sum("CARGA_TRIBUTARIA_TOTAL").as("Recaudacion_Total")
  )
  .orderBy("Mes")

z.show(evolutionMensual)
```

SPARK JOB FINISHED



settings ▲

## Available Fields

Mes

Valor\_CIF\_Total

Recaudacion\_Total

keys

Valor\_CIF\_Total ✕

groups

Mes ✕

values

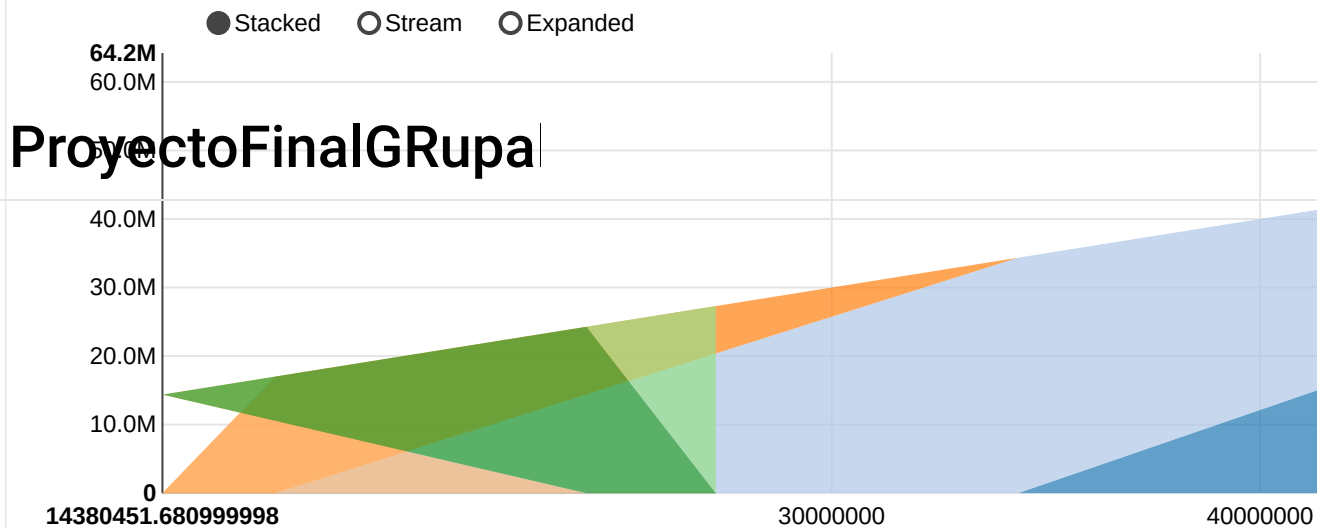
Valor\_CIF\_Total SUM ✕

xAxis :

Default

Rotate

Hide



```
import org.apache.spark.sql.functions.{sum, window, date_format}
evolucionMensual: org.apache.spark.sql.Dataset[org.apache.spark.sql.Row] = [Mes: string, Valor_CIF_Total: double ... 1 more field]
```

Took 4 sec. Last updated by anonymous at July 25 2025, 1:16:06 PM.

## Consulta 3: Desempeño y Especialización por Distrito Aduanero

¿Qué distritos aduaneros manejan más valor y qué tipo de carga procesan (pesada vs. ligera)?

Took 0 sec. Last updated by anonymous at July 25 2025, 1:00:04 PM.

```
import org.apache.spark.sql.functions.{sum, count, avg, desc}
```

SPARK JOB FINISHED

```
val analisisDistritos = dfPuroo.groupBy("DISTRITO")
  .agg(
    sum("CIF").as("Valor_Total_Importado"),
    count("*").as("Numero_Operaciones"),
    avg("PESO_NETO").as("Peso_Neto_Promedio_Kg")
  )
  .orderBy(desc("Valor_Total_Importado"))

z.show(analisisDistritos)
```



settings ▼

DISTRITO	Valor_Total_Importado	Numero_Operaciones
055-QUITO	9.121430947299995E7	1051
028-GUAYAQUIL - MARITIMO	7.916025707600029E7	2848
073-TULCAN	3787720.719000001	214
019-GUAYAQUIL - AEREO	3548960.773000003	984
037-MANTA	3336906.7980000004	99

091-CUENCA	331636.874	21
082-HUAQUILLAS	131579.061	20
093-CARACAS	4446.5289999999995	2

ProyectoFinalGRupa

```
import org.apache.spark.sql.functions.{sum, count, avg, desc}
 analisisDistritos: org.apache.spark.sql.Dataset[org.apache.spark.sql.Row] = [DISTRITO: string,
 Valor_Total_Importado: double ... 2 more fields]
```

Took 6 sec. Last updated by anonymous at July 25 2025, 1:00:10 PM.

Consulta 4: Impacto y Uso de los Convenios Internacionales

FINISHED

¿Qué volumen de comercio se realiza bajo convenios internacionales y cómo afecta esto a la carga tributaria promedio?

Took 0 sec. Last updated by anonymous at July 25 2025, 1:00:10 PM.

```
import org.apache.spark.sql.functions.{when, lit, sum, avg, count, desc}
// Creamos una columna para diferenciar si se usó un convenio o no
val dfConConvenio = dfPuroo.withColumn("Aplica_Convenio",
  when(col("CONVENIO_INTERNACIONAL") != "NO APLICA", "Con Convenio")
    .otherwise("Sin Convenio")
)

// Agrupamos para comparar las importaciones con y sin convenio
val impactoConvenios = dfConConvenio.groupBy("Aplica_Convenio")
  .agg(
    sum("CIF").as("Valor_CIF_Total"),
    avg("CARGA_TRIBUTARIA_TOTAL").as("Tributo_Promedio_por_Operacion"),
    count("*").as("Numero_de_Operaciones")
  )
  .orderBy(desc("Valor_CIF_Total"))

z.show(impactoConvenios)
```

SPARK JOB FINISHED

settings ▼

# ProyectoFinalGRupa

Aplica_Convenio	Valor_CIF_Total	Tributo
Sin Convenio	1.7181368712300035E8	3187.0624
Con Convenio	9704692.899999997	4501.4943

```
import org.apache.spark.sql.functions.{when, lit, sum, avg, count, desc}
dfConConvenio: org.apache.spark.sql.DataFrame = [PAIS_ORIGEN: string, TIPO_IMPORTACION: string
... 40 more fields]
impactoConvenios: org.apache.spark.sql.Dataset[org.apache.spark.sql.Row] = [Aplica_Convenio: s
tring, Valor_CIF_Total: double ... 2 more fields]
```

Took 4 sec. Last updated by anonymous at July 25 2025, 1:00:14 PM.

## Consulta 5: Productos Más Relevantes por Valor y su Carga Impositiva

FINISHED

¿En qué productos gastamos más dinero y cuál es la tasa impositiva real que se les aplica?

Took 0 sec. Last updated by anonymous at July 25 2025, 1:00:15 PM.

```
import org.apache.spark.sql.functions.{sum, desc, round}
val productosRelevantes = dfPuroo.groupBy("SUBPARTIDA", "DESCRIPCION_ARANCELARIA")
  .agg(
    sum("CIF").as("Valor_CIF_Total"),
    sum("CARGA_TRIBUTARIA_TOTAL").as("Tributos_Totales")
  )
  .withColumn("Tasa_Impositiva_Promedio", round(($"Tributos_Totales" / $"Valor_CIF_Total") * 100, 2))
  .orderBy(desc("Valor_CIF_Total"))
  .limit(20)

z.show(productosRelevantes)
```

SPARK JOB FINISHED

settings ▼

SUBPARTIDA	DESCRIPCION_ARANCELARIA	Valor_CIF_Total
2716000000	Energía eléctrica	8.785863512E7
8711200010	- - En CKD. Tarifa arancelaria variable para modelos y versiones autorizadas para el ensamblaje por	2.8244331965999994E7

# ProyectoFinalGRupa

	el Ministerio rector de la Política Industrial. MOE >= 18%	
5201003000	- De longitud de fibra superior a 1,22 mm (7/8 pulgada) pero inferior o igual a 28,57 mm (1 1/8 pulgada)	9290329.778999997
8429590000	- - Las demás	7463692.760999999

```
import org.apache.spark.sql.functions.{sum, desc, round}
productosRelevantes: org.apache.spark.sql.Dataset[org.apache.spark.sql.Row] = [SUBPARTIDA: big
int, DESCRIPCION_ARANCELARIA: string ... 3 more fields]
```

Took 6 sec. Last updated by anonymous at July 25 2025, 1:00:21 PM. (outdated)

%md

FINISHED

**\*\*El desarrollo de este proyecto nos ha permitido comprender de forma práctica cómo integrar y análisis más profundo y significativo. Usar herramientas como Spark, SQL, notebooks interactiv una visión más amplia sobre la realidad económica del país y el comercio internacional.\*\***

**El desarrollo de este proyecto nos ha permitido comprender de forma práctica cómo integrar y enriquecer datos provenientes de distintas fuentes para realizar un análisis más profundo y significativo. Usar herramientas como Spark, SQL, notebooks interactivos y unir datasets externos con variables macroeconómicas me ha brindado una visión más amplia sobre la realidad económica del país y el comercio internacional.**

Took 0 sec. Last updated by anonymous at July 25 2025, 1:15:34 PM.

FIN

FINISHED

Took 0 sec. Last updated by anonymous at July 25 2025, 1:10:51 PM.

%md

READY