

# Procesamiento de Datos de COVID-19 en Colombia usando PySpark en Google Colab

**1. Introducción** Este laboratorio se enfocó en el procesamiento de datos de casos positivos de COVID-19 en Colombia utilizando PySpark en Google Colab. El objetivo fue realizar diversas operaciones de manipulación y análisis de datos sobre el dataset proporcionado por el Ministerio de Salud.

## 2. Descripción del Dataset

- **Fuente de Datos:** Casos positivos de COVID-19 en Colombia
- **Origen:** Ministerio de Salud
- **Archivo:** CasosPositivosCovid19.csv

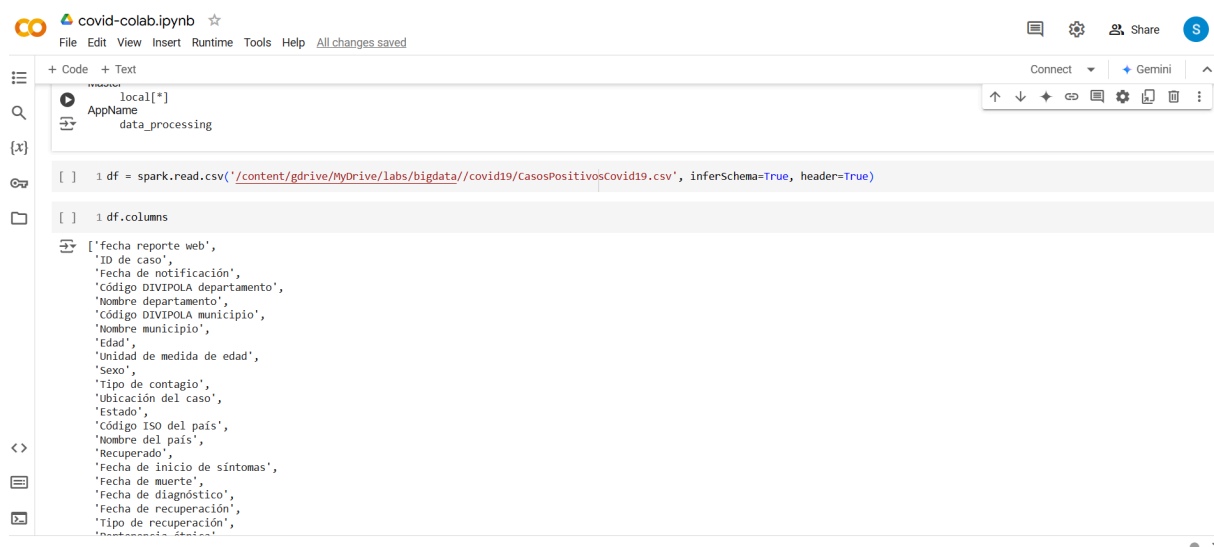
## 3. Procesamiento de Datos con PySpark Operaciones Realizadas:

### 1. Análisis de Columnas:

- Identificación de las columnas presentes en el dataset
- Evaluación de los tipos de datos de cada columna

### 2. Transformaciones de Datos:

- Selección de columnas específicas
- Renombrado de columnas
- Adición de nuevas columnas
- Eliminación de columnas no relevantes



```
File Edit View Insert Runtime Tools Help All changes saved
+ Code + Text
local[*]
AppName
data_processing

[ ] 1 df = spark.read.csv('/content/gdrive/MyDrive/labs/bigdata/covid19/casosPositivosCovid19.csv', inferSchema=True, header=True)

[ ] 1 df.columns

['fecha reporte web',
 'ID de caso',
 'Fecha de notificación',
 'código DIVIPOLA departamento',
 'Nombre departamento',
 'código DIVIPOLA municipio',
 'Nombre municipio',
 'Edad',
 'Unidad de medida de edad',
 'Sexo',
 'Tipo de contagio',
 'Ubicación del caso',
 'Estado',
 'código ISO del país',
 'Nombre del país',
 'Recuperado',
 'Fecha de inicio de síntomas',
 'Fecha de muerte',
 'Fecha de diagnóstico',
 'Fecha de recuperación',
 'Tipo de recuperación',
 'Tratamiento específico']
```

```
1 df.show(5)
```

fecha_reporte_web	id_caso	fecha_notificacion	codigo_divipola	departamento	nombre_departamento	codigo_divipola_municipio	nombre_municipio	edad	unidad_medida_edad	sexo	tipo_contagio	ubi
6/3/2020 0:00:00	1	2/3/2020 0:00:00	11	BOGOTA		11001	BOGOTA	19		1	F	Importado
9/3/2020 0:00:00	2	6/3/2020 0:00:00	76	VALLE		76111	BUGA	34		1	M	Importado
9/3/2020 0:00:00	3	7/3/2020 0:00:00	5	ANTIOQUIA		5001	MEDELLIN	50		1	F	Importado
11/3/2020 0:00:00	4	9/3/2020 0:00:00	5	ANTIOQUIA		5001	MEDELLIN	55		1	M	Relacionado
11/3/2020 0:00:00	5	9/3/2020 0:00:00	5	ANTIOQUIA		5001	MEDELLIN	25		1	M	Relacionado

only showing top 5 rows

```
1 df.select('edad', 'sexo').show(5)
```

edad	sexo
19	F
34	M
50	F
55	M
25	M

only showing top 5 rows

```
1 df.describe().show()
```

summary	fecha_reporte_web	id_caso	fecha_notificacion	codigo_divipola	departamento	nombre_departamento	codigo_divipola_municipio	nombre_municipio	edad	unidad
count	100000	100000	100000	100000	100000	100000	100000	100000	100000	100000
mean	NULL	50038.74855	NULL	2631.6288	NULL	25327.34487	NULL	39.30175	39.30175	0.08893
stddev	NULL	28870.559312724497	NULL	6172.66039906438	NULL	25830.580807180544	NULL	18.420127848324004	18.420127848324004	0.08893
min	1/4/2020 0:00:00	1	1/4/2020 0:00:00	5	AMAZONAS	5001	ABREGO	1	1	1
max	9/6/2020 0:00:00	100040	9/7/2020 0:00:00	47001	VICHADA	99001	puerto COLOMBIA	104	104	104

```
1 from pyspark.sql.functions import when
```

```
1 df = df.withColumn("es_español", when(df["nombre_pais"] == "ESPAÑA", True).otherwise(False))
```

```
2 df.show()
```

fecha_reporte_web	id_caso	fecha_notificacion	codigo_divipola	departamento	nombre_departamento	codigo_divipola_municipio	nombre_municipio	edad	unidad_medida_edad	sexo	tipo_contagio	ubi
6/3/2020 0:00:00	1	2/3/2020 0:00:00	11	BOGOTA		11001	BOGOTA	19		1	F	Importado
9/3/2020 0:00:00	2	6/3/2020 0:00:00	76	VALLE		76111	BUGA	34		1	M	Importado
9/3/2020 0:00:00	3	7/3/2020 0:00:00	5	ANTIOQUIA		5001	MEDELLIN	50		1	F	Importado
11/3/2020 0:00:00	4	9/3/2020 0:00:00	5	ANTIOQUIA		5001	MEDELLIN	55		1	M	Relacionado

```
1 df.filter(df['nombre_departamento'] == 'ANTIOQUIA').show()
```

fecha_reporte_web	id_caso	fecha_notificacion	codigo_divipola	departamento	nombre_departamento	codigo_divipola_municipio	nombre_municipio	edad	unidad_medida_edad	sexo	tipo_contagio	ubi
9/3/2020 0:00:00	3	7/3/2020 0:00:00	5	ANTIOQUIA		5001	MEDELLIN	50		1	F	Importado
11/3/2020 0:00:00	4	9/3/2020 0:00:00	5	ANTIOQUIA		5001	MEDELLIN	55		1	M	Relacionado
11/3/2020 0:00:00	5	9/3/2020 0:00:00	5	ANTIOQUIA		5001	MEDELLIN	25		1	M	Relacionado
11/3/2020 0:00:00	6	10/3/2020 0:00:00	5	ANTIOQUIA		5300	ITAGUI	27		1	F	Relacionado
14/3/2020 0:00:00	20	11/3/2020 0:00:00	5	ANTIOQUIA		5001	MEDELLIN	26		1	F	Relacionado
14/3/2020 0:00:00	21	11/3/2020 0:00:00	5	ANTIOQUIA		5001	MEDELLIN	28		1	M	Relacionado
14/3/2020 0:00:00	22	12/3/2020 0:00:00	5	ANTIOQUIA		5615	RIONEGRO	36		1	M	Importado
15/3/2020 0:00:00	32	11/3/2020 0:00:00	5	ANTIOQUIA		5001	MEDELLIN	55		1	M	Importado
19/3/2020 0:00:00	106	19/3/2020 0:00:00	5	ANTIOQUIA		5001	MEDELLIN	44		1	M	Importado
19/3/2020 0:00:00	107	12/3/2020 0:00:00	5	ANTIOQUIA		5001	MEDELLIN	56		1	M	Importado
19/3/2020 0:00:00	108	17/3/2020 0:00:00	5	ANTIOQUIA		5001	MEDELLIN	57		1	F	Importado
20/3/2020 0:00:00	131	15/3/2020 0:00:00	5	ANTIOQUIA		5001	MEDELLIN	22		1	F	Importado
20/3/2020 0:00:00	133	16/3/2020 0:00:00	5	ANTIOQUIA		5615	RIONEGRO	51		1	M	Relacionado
20/3/2020 0:00:00	134	17/3/2020 0:00:00	5	ANTIOQUIA		5300	LA ESTRELLA	28		1	M	Relacionado
20/3/2020 0:00:00	135	17/3/2020 0:00:00	5	ANTIOQUIA		5001	MEDELLIN	44		1	F	Importado
20/3/2020 0:00:00	136	17/3/2020 0:00:00	5	ANTIOQUIA		5001	MEDELLIN	37		1	M	Relacionado
20/3/2020 0:00:00	137	17/3/2020 0:00:00	5	ANTIOQUIA		5266	ENVIGADO	54		1	M	Importado
20/3/2020 0:00:00	141	17/3/2020 0:00:00	5	ANTIOQUIA		5001	MEDELLIN	62		1	F	Importado
20/3/2020 0:00:00	142	20/3/2020 0:00:00	5	ANTIOQUIA		5001	MEDELLIN	35		1	F	Importado
20/3/2020 0:00:00	143	14/3/2020 0:00:00	5	ANTIOQUIA		5266	ENVIGADO	46		1	M	Importado

only showing top 20 rows

```
1 df.groupby('codigo_iso_pais').mean().show(5, False)
```

codigo_iso_pais	avg(id_caso)	avg(codigo_divipola_departamento)	avg(codigo_divipola_municipio)	avg(edad)	avg(unidad_medida_edad)	avg(codigo_iso_pais)	avg(pertenencia_etnica)
858	2425.0	76.0	76113.0	24.0	1.0	858.0	6.0
530	782.75	10.25	10255.625	40.75	1.0	530.0	6.0
756	962.0	76.0	76001.0	68.0	1.0	756.0	6.0
300	341.25	52.25	52377.75	51.5	1.0	300.0	6.0
784	620.0	76.0	76520.0	45.0	1.0	784.0	6.0

only showing top 5 rows

```
1 df.groupby('codigo_iso_pais').sum().show(5, False)
```

codigo_iso_pais	sum(id_caso)	sum(codigo_divipola_departamento)	sum(codigo_divipola_municipio)	sum(edad)	sum(unidad_medida_edad)	sum(codigo_iso_pais)	sum(pertenencia_etnica)
858	2425	76	76113	24	1	858	6
530	6262	82	82045	326	8	4240	48
756	962	76	76001	68	1	756	6
300	1365	209	209511	206	4	1200	24
784	620	76	76520	45	1	784	6

only showing top 5 rows

```
1 write_uri='s3a://labs/bigdata/colab/csv'
```

```
1 df.coalesce(1).write.format("csv").option("header", "true").save(write_uri)
```

Buckets

Access Grants

Access Points

Object Lambda Access Points

Multi-Region Access Points

Batch Operations

IAM Access Analyzer for S3

Block Public Access settings for this account

Storage Lens

Dashboards

Storage Lens groups

AWS Organizations settings

Feature spotlight

AWS Marketplace for S3

csv/

Objects | Properties

Objects (2) Info

Copy S3 URI

Find objects by prefix

	Name	Type
	<a href="#">_SUCCESS</a>	-
	<a href="#">part-00000-ead3114b-92f0-462f-bf83-f88b35c88ae4-c000.csv</a>	csv