1. Carga de datos

Para cargar los datos, debemos descargar en nuestra máquina local el dataframe mencionado en la guía del trabajo; esto, mediante el link que aparece en la misma.

```
[14] df=spark.read.csv('gdrive/MyDrive/st0263trabajo3/Casos_positivos_de_COVID-19_en_Colombia.csv',inferSchema=True,header=True)
```

2. Análisis exploratorio del dataframe

Columnas

```
[15] 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',
 'Pertenencia étnica',
 'Nombre del grupo étnico']
```

Tipos de datos

```
[16] df.printSchema()
```

```
root
 |-- fecha reporte web: string (nullable = true)
 |-- ID de caso: integer (nullable = true)
 |-- Fecha de notificación: string (nullable = true)
 |-- Código DIVIPOLA departamento: integer (nullable = true)
 |-- Nombre departamento: string (nullable = true)
 |-- Código DIVIPOLA municipio: integer (nullable = true)
 |-- Nombre municipio: string (nullable = true)
 |-- Edad: integer (nullable = true)
 |-- Unidad de medida de edad: integer (nullable = true)
 |-- Sexo: string (nullable = true)
 |-- Tipo de contagio: string (nullable = true)
 |-- Ubicación del caso: string (nullable = true)
 |-- Estado: string (nullable = true)
 |-- Código ISO del país: integer (nullable = true)
 |-- Nombre del país: string (nullable = true)
 |-- Recuperado: string (nullable = true)
 |-- Fecha de inicio de síntomas: string (nullable = true)
 |-- Fecha de muerte: string (nullable = true)
 |-- Fecha de diagnóstico: string (nullable = true)
 |-- Fecha de recuperación: string (nullable = true)
 |-- Tipo de recuperación: string (nullable = true)
 |-- Pertenencia étnica: integer (nullable = true)
 |-- Nombre del grupo étnico: string (nullable = true)
```

Seleccionar algunas columnas

```
df.select('ID de caso','Nombre departamento','Estado','Sexo').show(10)
```

```
+----------+-------------------+------+----+
|ID de caso|Nombre departamento|Estado|Sexo|
+----------+-------------------+------+----+
|   2265685|             BOGOTA|  Leve|   M|
|   2265686|             BOGOTA|  Leve|   M|
|   2265687|             BOGOTA|  Leve|   F|
|   2265688|             BOGOTA|  Leve|   F|
|   2265689|             BOGOTA|  Leve|   F|
|   2265690|             BOGOTA|  Leve|   F|
|   1851419|             NARIÑO|  Leve|   M|
|   1851420|             NARIÑO|  Leve|   M|
|   1851421|             NARIÑO|  Leve|   M|
|   1851422|             NARIÑO|  Leve|   F|
+----------+-------------------+------+----+
only showing top 10 rows
```

Renombrar columnas

```
df.withColumnRenamed("Nombre del grupo étnico","Grupo étnico").printSchema()
```

```
root
 |-- fecha reporte web: string (nullable = true)
 |-- ID de caso: integer (nullable = true)
 |-- Fecha de notificación: string (nullable = true)
 |-- Código DIVIPOLA departamento: integer (nullable = true)
 |-- Nombre departamento: string (nullable = true)
 |-- Código DIVIPOLA municipio: integer (nullable = true)
 |-- Nombre municipio: string (nullable = true)
 |-- Edad: integer (nullable = true)
 |-- Unidad de medida de edad: integer (nullable = true)
 |-- Sexo: string (nullable = true)
 |-- Tipo de contagio: string (nullable = true)
 |-- Ubicación del caso: string (nullable = true)
 |-- Estado: string (nullable = true)
 |-- Código ISO del país: integer (nullable = true)
 |-- Nombre del país: string (nullable = true)
 |-- Recuperado: string (nullable = true)
 |-- Fecha de inicio de síntomas: string (nullable = true)
 |-- Fecha de muerte: string (nullable = true)
 |-- Fecha de diagnóstico: string (nullable = true)
 |-- Fecha de recuperación: string (nullable = true)
 |-- Tipo de recuperación: string (nullable = true)
 |-- Pertenencia étnica: integer (nullable = true)
 |-- Grupo étnico: string (nullable = true)
```

Agregar columnas

```
from pyspark.sql.functions import lit
df.withColumn("newColumn", lit("valor")).show(10)
```

```
+------------------+----------+--------------------+---------------------------+--------------------+-------------------------+--------------------+--
|  fecha reporte web|ID de caso|Fecha de notificación|Código DIVIPOLA departamento|Nombre departamento|Código DIVIPOLA municipio|Nombre municipio|Ec
+------------------+----------+--------------------+---------------------------+--------------------+-------------------------+--------------------+--
|2021-03-04 00:00:00|   2265685|  2021-03-02 00:00:00|                         11|              BOGOTA|                    11001|              BOGOTA|
|2021-03-04 00:00:00|   2265686|  2021-03-02 00:00:00|                         11|              BOGOTA|                    11001|              BOGOTA|
|2021-03-04 00:00:00|   2265687|  2021-03-02 00:00:00|                         11|              BOGOTA|                    11001|              BOGOTA|
|2021-03-04 00:00:00|   2265688|  2021-03-02 00:00:00|                         11|              BOGOTA|                    11001|              BOGOTA|
|2021-03-04 00:00:00|   2265689|  2021-03-02 00:00:00|                         11|              BOGOTA|                    11001|              BOGOTA|
|2021-03-04 00:00:00|   2265690|  2021-03-02 00:00:00|                         11|              BOGOTA|                    11001|              BOGOTA|
|2021-01-15 00:00:00|   1851419|  2021-01-12 00:00:00|                         52|             NARIÑO|                    52227|              CUMBAL|
|2021-01-15 00:00:00|   1851420|  2021-01-11 00:00:00|                         52|             NARIÑO|                    52227|              CUMBAL|
|2021-01-15 00:00:00|   1851421|  2021-01-12 00:00:00|                         52|             NARIÑO|                    52317|           GUACHUCAL|
|2021-01-15 00:00:00|   1851422|  2021-01-12 00:00:00|                         52|             NARIÑO|                    52378|             LA CRUZ|
+------------------+----------+--------------------+---------------------------+--------------------+-------------------------+--------------------+--
only showing top 10 rows
```

Borrar columnas

```
df.drop("newColumn").printSchema()
```

```
root
 |-- fecha reporte web: string (nullable = true)
 |-- ID de caso: integer (nullable = true)
 |-- Fecha de notificación: string (nullable = true)
 |-- Código DIVIPOLA departamento: integer (nullable = true)
 |-- Nombre departamento: string (nullable = true)
 |-- Código DIVIPOLA municipio: integer (nullable = true)
 |-- Nombre municipio: string (nullable = true)
 |-- Edad: integer (nullable = true)
 |-- Unidad de medida de edad: integer (nullable = true)
 |-- Sexo: string (nullable = true)
 |-- Tipo de contagio: string (nullable = true)
 |-- Ubicación del caso: string (nullable = true)
 |-- Estado: string (nullable = true)
 |-- Código ISO del país: integer (nullable = true)
 |-- Nombre del país: string (nullable = true)
 |-- Recuperado: string (nullable = true)
 |-- Fecha de inicio de síntomas: string (nullable = true)
 |-- Fecha de muerte: string (nullable = true)
 |-- Fecha de diagnóstico: string (nullable = true)
 |-- Fecha de recuperación: string (nullable = true)
 |-- Tipo de recuperación: string (nullable = true)
 |-- Pertenencia étnica: integer (nullable = true)
 |-- Nombre del grupo étnico: string (nullable = true)
```

Filtrar datos

```
df.filter(df['Nombre municipio']=='MARINILLA').select('ID de caso','Nombre municipio','Edad','Sexo').show(10)
```

```
+----------+----------------+----+----+
|ID de caso|Nombre municipio|Edad|Sexo|
+----------+----------------+----+----+
|   1224866|       MARINILLA|  26|   F|
|   1280976|       MARINILLA|  25|   F|
|   2431680|       MARINILLA|  54|   F|
|   2084116|       MARINILLA|  20|   M|
|   2084130|       MARINILLA|  22|   M|
|   2432954|       MARINILLA|  58|   M|
|   2327572|       MARINILLA|  22|   F|
|   2327576|       MARINILLA|  27|   M|
|    845685|       MARINILLA|  21|   F|
|   2363832|       MARINILLA|  59|   F|
+----------+----------------+----+----+
only showing top 10 rows
```

Usar funciones UDF

```
from pyspark.sql.functions import udf
from pyspark.sql.types import StringType

def capitalino(nombre_municipio):
  if nombre_municipio == 'BOGOTA':
        return 'Si'
  else:
        return 'No'

esdecapital_udf=udf(capitalino,StringType())
df.withColumn('Es de la capital?',esdecapital_udf(df['Nombre municipio'])).select('ID de caso','Nombre municipio','Edad','Sexo', 'Es de la capital?
```

```
+----------+----------------+----+----+-----------------+
|ID de caso|Nombre municipio|Edad|Sexo|Es de la capital?|
+----------+----------------+----+----+-----------------+
|2265685   |BOGOTA          |49  |M   |Si               |
|2265686   |BOGOTA          |49  |M   |Si               |
|2265687   |BOGOTA          |51  |F   |Si               |
|2265688   |BOGOTA          |51  |F   |Si               |
|2265689   |BOGOTA          |51  |F   |Si               |
|2265690   |BOGOTA          |52  |F   |Si               |
|1851419   |CUMBAL          |24  |M   |No               |
|1851420   |CUMBAL          |24  |M   |No               |
|1851421   |GUACHUCAL       |33  |M   |No               |
|1851422   |LA CRUZ         |35  |F   |No               |
+----------+----------------+----+----+-----------------+
only showing top 10 rows
```

3

3.1

```
df.groupBy('Nombre departamento').count().orderBy('count',ascending=False).show(10,False)
```

```
+-------------------+-------+
|Nombre departamento|count  |
+-------------------+-------+
|BOGOTA             |1780254|
|ANTIOQUIA          |923106 |
|VALLE              |542476 |
|CUNDINAMARCA       |319162 |
|SANTANDER          |284709 |
|BARRANQUILLA       |266302 |
|CARTAGENA          |158558 |
|ATLANTICO          |138287 |
|BOYACA             |125438 |
|TOLIMA             |124690 |
+-------------------+-------+
only showing top 10 rows
```

3.2

```
df.groupBy('Nombre municipio').count().orderBy('count',ascending=False).show(10,False)
```

```
+----------------+-------+
|Nombre municipio|count  |
+----------------+-------+
|BOGOTA          |1780254|
|MEDELLIN        |527188 |
|CALI            |383637 |
|BARRANQUILLA    |266302 |
|CARTAGENA       |158558 |
|BUCARAMANGA     |135905 |
|IBAGUE          |89078  |
|SANTA MARTA     |83051  |
|MANIZALES       |82124  |
|VALLEDUPAR      |74318  |
+----------------+-------+
only showing top 10 rows
```

3.3

```
df.groupBy('Fecha de diagnóstico').count().orderBy('count',ascending=False).show(10,False)
```

```
+-------------------+-----+
|Fecha de diagnóstico|count|
+-------------------+-----+
|2022-01-07 00:00:00 |42072|
|2022-01-06 00:00:00 |40283|
|2022-01-12 00:00:00 |35532|
|2022-01-05 00:00:00 |35480|
|2022-01-11 00:00:00 |35410|
|2022-01-13 00:00:00 |34735|
|2021-06-25 00:00:00 |34173|
|2021-06-23 00:00:00 |33936|
|2021-06-15 00:00:00 |33826|
|2021-06-24 00:00:00 |33220|
+-------------------+-----+
only showing top 10 rows
```

3.4

```
df.groupBy('Edad').count().orderBy('count',ascending=False).show(10,False)
```

```
+----+------+
|Edad|count |
+----+------+
|30  |148550|
|29  |147213|
|28  |147068|
|31  |145520|
|27  |145236|
|26  |144526|
|32  |142215|
|25  |138975|
|33  |137084|
|35  |135806|
+----+------+
only showing top 10 rows
```

3.5

```
from pyspark.sql.functions import to_date, to_timestamp, month
df.withColumn("Fecha de diagnóstico",to_date(to_timestamp("Fecha de diagnóstico")))
data = df.groupBy(month('Fecha de diagnóstico').alias('mes')).count().orderBy('count',ascending=False).show(12,False)
```

```
+---+-------+
|mes|count  |
+---+-------+
|1  |1188196|
|6  |934429 |
|7  |688541 |
|5  |607510 |
|4  |494096 |
|12 |473736 |
|8  |412941 |
|11 |310573 |
|10 |294683 |
|9  |263213 |
|2  |253800 |
|3  |193374 |
+---+-------+
only showing top 12 rows
```

3.6

```
muertosAntioquia = df.filter((df['Fecha de muerte'] != 'null')).filter((df['Nombre Departamento'] == 'ANTIOQUIA')).count()
muertosAntioquia
```

```
21443
```