Zeppelin Notebook Job Search anonymous

Proyecto Bimestral

Head default

FINISHED

📊 Análisis Exploratorio de Datos (EDA)

Objetivo:
Realizar un Análisis Exploratorio de Datos (EDA) completo al conjunto de datos que se utilizará en el proyecto bimestral.

🔍 Integrantes:
-Juan Pablo Landi
-María Valentina Samaniego
-María Paula Guallo

Took 0 seconds. Last updated by anonymous at June 18 2025, 7:18:44 PM.

FINISHED

1 Lectura del archivo

Took 0 seconds. Last updated by anonymous at June 18 2025, 8:15:06 PM.

SPARK JOB FINISHED

```
val dfNacidos = spark
    .read
    .option("header", true)
    .option("sep", ";")
    .option("inferSchema", true)
    .csv("/workspace/progava-s10/ENV_2023-copia.csv")
```

```
dfNacidos: org.apache.spark.sql.DataFrame = [prov_insc: string, cant_insc: string ... 45 more fields]
```

Took 22 seconds. Last updated by anonymous at June 25 2025, 8:34:16 PM.

FINISHED

2 Imprimir Cantidad de Filas y Columnas

🔢 *Propósito:*
Verificar que los datos se hayan cargado correctamente, comparando las dimensiones del dataset con lo especificado en la documentación oficial de origen.

✅ *Acción:*
Imprimir el número total de *filas* (registros) y *columnas* (variables).

📌 Esta validación inicial es esencial para confirmar que no hubo errores al momento de la lectura del archivo o la conexión con la fuente de datos.

Took 0 seconds. Last updated by anonymous at June 18 2025, 8:15:27 PM.

SPARK JOB FINISHED

```
print(s"Registros (filas); ${dfNacidos.count}, Variables(columnas): ${dfNacidos.columns.length}")
```

```
Registros (filas); 241295, Variables(columnas): 47
```

Took 2 seconds. Last updated by anonymous at June 18 2025, 7:00:34 PM. (outdated)

FINISHED

3 Imprimir el Esquema del Dataset

-*Propósito:*
Verificar si la interfaz de lectura ha procesado correctamente los tipos de datos de cada columna.

Referencia:
Según el archivo inec_nacidosvivos_dd_2023.ods, algunas columnas esperadas como *numéricas* son:

- anio_nac (Año de nacimiento)
- dia_nac (Día de nacimiento)
- anio_insc (¿Año de inscripción?)
- mes_insc (¿Mes de inscripción?)
- dia_insc (¿Día de inscripción?)
- talla (¿Talla del recién nacido?)
- peso (¿Peso del recién nacido?)

📌 Esta validación es clave para evitar errores posteriores durante el análisis estadístico o modelado.

Took 0 seconds. Last updated by anonymous at June 18 2025, 8:15:46 PM.

FINISHED

```
dfNacidos.printSchema
```

```
root
 |-- prov_insc: string (nullable = true)
 |-- cant_insc: string (nullable = true)
 |-- parr_insc: string (nullable = true)
 |-- fecha_insc: string (nullable = true)
 |-- anio_insc: string (nullable = true)
 |-- mes_insc: string (nullable = true)
 |-- dia_insc: string (nullable = true)
 |-- sexo: string (nullable = true)
 |-- fecha_nac: string (nullable = true)
 |-- anio_nac: integer (nullable = true)
 |-- mes_nac: string (nullable = true)
 |-- dia_nac: integer (nullable = true)
 |-- talla: string (nullable = true)
 |-- peso: string (nullable = true)
 |-- sem_gest: string (nullable = true)
```

```
|-- tipo_part: string (nullable = true)
|-- lugar_ocur: string (nullable = true)
|-- apgar1: string (nullable = true)
|-- apgar5: string (nullable = true)
|-- p_emb: string (nullable = true)
|-- prov_nac: string (nullable = true)
|-- cant_nac: string (nullable = true)
|-- parr_nac: string (nullable = true)
|-- area_nac: string (nullable = true)
|-- asis_por: string (nullable = true)
|-- nac_mad: string (nullable = true)
|-- cod_pais: string (nullable = true)
|-- fecha_mad: string (nullable = true)
|-- anio_mad: string (nullable = true)
|-- mes_mad: string (nullable = true)
|-- dia_mad: string (nullable = true)
|-- edad_mad: string (nullable = true)
|-- con_pren: string (nullable = true)
|-- num_emb: string (nullable = true)
|-- num_par: string (nullable = true)
|-- hij_viv: integer (nullable = true)
|-- hij_vivm: string (nullable = true)
|-- hij_nacm: string (nullable = true)
|-- etnia: string (nullable = true)
|-- est_civil: string (nullable = true)
|-- niv_inst: string (nullable = true)
|-- sabe_leer: string (nullable = true)
|-- prov_res: string (nullable = true)
|-- cant_res: string (nullable = true)
|-- parr_res: string (nullable = true)
|-- area_res: string (nullable = true)
|-- residente: string (nullable = true)
```

Took 0 seconds. Last updated by anonymous at June 18 2025, 7:08:12 PM.

FINISHED

4 Estadística Descriptiva

-Objetivo:
Resumir y describir las principales características de los datos mediante medidas estadísticas básicas.

-Enfoque 1: describe()
Utilizaremos el método describe() para obtener un resumen estadístico de las columnas numéricas (y, si es posible, también de algunas categóricas).

Took 0 seconds. Last updated by anonymous at June 18 2025, 8:18:17 PM.

SPARK JOB FINISHED

```
dfNacidos
    .describe()
    .show
```

```
+-------+---------------+---------+--------------------+----------+-----------------+----------+-----------------+------+----------+-----------------+----------+-----------------+-----------------+----------
|summary|      prov_insc|cant_insc|           parr_insc|fecha_insc|        anio_insc|  mes_insc|          dia_insc|  sexo| fecha_nac|         anio_nac|   mes_nac|          dia_nac|            talla|
+-------+---------------+---------+--------------------+----------+-----------------+----------+-----------------+------+----------+-----------------+----------+-----------------+-----------------+----------
|  count|         241295|   241295|              241295|    241295|           241295|    241295|           241295|241295|    241295|           241295|    241295|           241295|           241295|
|   mean|           null|     null|                null|      null|2023.0647378454437|      null|15.856224576692474|  null|      null|2022.9372345054808|      null|15.632362875318593| 48.54522348326832|3068.71532
| stddev|           null|     null|                null|      null|0.6836599754432079|      null| 8.685787511632697|  null|      null|1.5180530471163742|      null| 8.784520746500172|2.4648283275783904|505.607865
|    min|               |         |                    |          |                 |          |                 |Hombre|1900/01/01|             1900|     Abril|                1|               38|
|    max|Zamora Chinchipe|   Zaruma|Ángel Polibio Cháves|2024/04/30|             2024|Septiembre|                9| Mujer|2023/12/31|             2023|Septiembre|               31|   Sin información|   Sin inf
+-------+---------------+---------+--------------------+----------+-----------------+----------+-----------------+------+----------+-----------------+----------+-----------------+-----------------+----------
```

Took 26 seconds. Last updated by anonymous at June 18 2025, 7:14:16 PM. (outdated)

SPARK JOB FINISHED

```
dfNacidos
    .select("anio_insc", "dia_insc", "anio_nac", "dia_nac")
    .describe()
    .show
```

```
+-------+-----------------+-----------------+-----------------+-----------------+
|summary|        anio_insc|         dia_insc|         anio_nac|         dia_nac|
+-------+-----------------+-----------------+-----------------+-----------------+
|  count|           241295|           241295|           241295|           241295|
|   mean|2023.0647378454437|15.856224576692474|2022.9372345054808|15.632362875318593|
| stddev|0.6836599754432079| 8.685787511632697|1.5180530471163742| 8.784520746500172|
|    min|                |                |             1900|                1|
|    max|             2024|                9|             2023|               31|
+-------+-----------------+-----------------+-----------------+-----------------+
```

Took 2 seconds. Last updated by anonymous at June 18 2025, 7:36:08 PM. (outdated)

SPARK JOB FINISHED

```
import org.apache.spark.sql.types._
val numericCols = dfNacidos.schema.fields.filter {
    case StructField(_, IntegerType | LongType | FloatType | DoubleType | ShortType | DecimalType (),_ ,_) => true
    case _=> false
}.map(_.name)

dfNacidos.select(numericCols.map(col):_*).describe().show
```

```
+-------+-----------------+-----------------+-----------------+
|summary|         anio_nac|          dia_nac|          hij_viv|
+-------+-----------------+-----------------+-----------------+
|  count|           241295|           241295|           241295|
|   mean|2022.9372345054808|15.632362875318593|2.0577550301498166|
| stddev|1.5180530471163742| 8.784520746500172|1.2242374055754184|
|    min|             1900|                1|                1|
|    max|             2023|               31|               16|
+-------+-----------------+-----------------+-----------------+

import org.apache.spark.sql.types._
numericCols: Array[String] = Array(anio_nac, dia_nac, hij_viv)
```

Took 3 seconds. Last updated by anonymous at June 18 2025, 7:56:40 PM. (outdated)

SPARK JOB FINISHED

```
dfNacidos.select(numericCols.map(col): _*).summary().show
```

```
+-------+------------------+------------------+------------------+
|summary|          anio_nac|           dia_nac|           hij_viv|
+-------+------------------+------------------+------------------+
|  count|            241295|            241295|            241295|
|   mean|2022.9372345054808|15.632362875318593|2.0577550301498166|
| stddev|1.5180530471163742| 8.784520746500172|1.2242374055754184|
|    min|              1900|                 1|                 1|
|    25%|              2023|                 8|                 1|
|    50%|              2023|                16|                 2|
|    75%|              2023|                23|                 3|
|    max|              2023|                31|                16|
+-------+------------------+------------------+------------------+
```

Took 3 seconds. Last updated by anonymous at June 18 2025, 8:01:44 PM.

SPARK JOB FINISHED

```
dfNacidos.select(numericCols.map(col): _*).summary("stddev", "25%").show
```

```
+-------+------------------+-----------------+------------------+
|summary|          anio_nac|          dia_nac|           hij_viv|
+-------+------------------+-----------------+------------------+
| stddev|1.5180530471163742|8.784520746500172|1.2242374055754184|
|    25%|              2023|                8|                 1|
+-------+------------------+-----------------+------------------+
```

Took 1 second. Last updated by anonymous at June 18 2025, 8:07:55 PM.

SPARK JOB FINISHED

```
dfNacidos.select(numericCols.map(col): _*).summary("stddev", "91%").show
```

```
+-------+------------------+-----------------+------------------+
|summary|          anio_nac|          dia_nac|           hij_viv|
+-------+------------------+-----------------+------------------+
| stddev|1.5180530471163742|8.784520746500172|1.2242374055754184|
|    91%|              2023|               28|                 4|
+-------+------------------+-----------------+------------------+
```

Took 3 seconds. Last updated by anonymous at June 18 2025, 8:09:36 PM.

SPARK JOB FINISHED

```
dfNacidos.select(numericCols.map(col): _*).summary("count", "count_distinct").show()
```

```
+--------------+--------+-------+-------+
|       summary|anio_nac|dia_nac|hij_viv|
+--------------+--------+-------+-------+
|         count|  241295| 241295| 241295|
|count_distinct|      41|     31|     15|
+--------------+--------+-------+-------+
```

Took 2 seconds. Last updated by anonymous at June 18 2025, 8:11:41 PM.

FINISHED

5. Transformaciones

Took 3 seconds. Last updated by anonymous at June 18 2025, 8:13:31 PM.

FINISHED

```
val dfNacidosClean = dfNacidos.withColumn("fecha_insc_date", to_date(col("fecha_insc"), ""))
```

```
dfNacidosClean: org.apache.spark.sql.DataFrame = [prov_insc: string, cant_insc: string ... 46 more fields]
```

Took 1 second. Last updated by anonymous at June 25 2025, 7:40:29 PM.

FINISHED

```
dfNacidosClean.printSchema
```

```
root
 |-- prov_insc: string (nullable = true)
 |-- cant_insc: string (nullable = true)
 |-- parr_insc: string (nullable = true)
 |-- fecha_insc: string (nullable = true)
 |-- anio_insc: string (nullable = true)
 |-- mes_insc: string (nullable = true)
 |-- dia_insc: string (nullable = true)
 |-- sexo: string (nullable = true)
 |-- fecha_nac: string (nullable = true)
 |-- anio_nac: integer (nullable = true)
 |-- mes_nac: string (nullable = true)
 |-- dia_nac: integer (nullable = true)
 |-- talla: string (nullable = true)
 |-- peso: string (nullable = true)
 |-- sem_gest: string (nullable = true)
 |-- tipo_part: string (nullable = true)
 |-- lugar_ocur: string (nullable = true)
 |-- apgar1: string (nullable = true)
 |-- apgar5: string (nullable = true)
 |-- p_emb: string (nullable = true)
 |-- prov_nac: string (nullable = true)
 |-- cant_nac: string (nullable = true)
 |-- parr_nac: string (nullable = true)
 |-- area_nac: string (nullable = true)
 |-- asis_por: string (nullable = true)
 |-- nac_mad: string (nullable = true)
 |-- cod_pais: string (nullable = true)
 |-- fecha_mad: string (nullable = true)
```

```
|-- anio_mad: string (nullable = true)
|-- mes_mad: string (nullable = true)
|-- dia_mad: string (nullable = true)
|-- edad_mad: string (nullable = true)
|-- con_pren: string (nullable = true)
|-- num_emb: string (nullable = true)
|-- num_par: string (nullable = true)
|-- hij_viv: integer (nullable = true)
|-- hij_vivm: string (nullable = true)
|-- hij_nacm: string (nullable = true)
|-- etnia: string (nullable = true)
|-- est_civil: string (nullable = true)
|-- niv_inst: string (nullable = true)
|-- sabe_leer: string (nullable = true)
|-- prov_res: string (nullable = true)
|-- cant_res: string (nullable = true)
|-- parr_res: string (nullable = true)
|-- area_res: string (nullable = true)
|-- residente: string (nullable = true)
|-- fecha_insc_date: date (nullable = true)
```

Took 0 seconds. Last updated by anonymous at June 18 2025, 8:19:19 PM.

FINISHED

```
val dfNacidosClean = dfNacidos
    .withColumn("fecha_insc_date",to_date(col("fecha_insc"),"yyyy/MM/dd"))
    .withColumn("fecha_nac_date",to_date(col("fecha_nac"),"yyyy/MM/dd"))
    .withColumn("fecha_mad_date",to_date(col("fecha_mad"),"yyyy/MM/dd"))
```

```
dfNacidosClean: org.apache.spark.sql.DataFrame = [prov_insc: string, cant_insc: string ... 48 more fields]
```

Took 0 seconds. Last updated by anonymous at June 25 2025, 8:34:24 PM.

FINISHED

```
dfNacidosClean.printSchema
```

```
root
 |-- prov_insc: string (nullable = true)
 |-- cant_insc: string (nullable = true)
 |-- parr_insc: string (nullable = true)
 |-- fecha_insc: string (nullable = true)
 |-- anio_insc: string (nullable = true)
 |-- mes_insc: string (nullable = true)
 |-- dia_insc: string (nullable = true)
 |-- sexo: string (nullable = true)
 |-- fecha_nac: string (nullable = true)
 |-- anio_nac: integer (nullable = true)
 |-- mes_nac: string (nullable = true)
 |-- dia_nac: integer (nullable = true)
 |-- talla: string (nullable = true)
 |-- peso: string (nullable = true)
 |-- sem_gest: string (nullable = true)
 |-- tipo_part: string (nullable = true)
 |-- lugar_ocur: string (nullable = true)
 |-- apgar1: string (nullable = true)
 |-- apgar5: string (nullable = true)
 |-- p_emb: string (nullable = true)
 |-- prov_nac: string (nullable = true)
 |-- cant_nac: string (nullable = true)
 |-- parr_nac: string (nullable = true)
 |-- area_nac: string (nullable = true)
 |-- asis_por: string (nullable = true)
 |-- nac_mad: string (nullable = true)
 |-- cod_pais: string (nullable = true)
 |-- fecha_mad: string (nullable = true)
 |-- anio_mad: string (nullable = true)
 |-- mes_mad: string (nullable = true)
 |-- dia_mad: string (nullable = true)
 |-- edad_mad: string (nullable = true)
 |-- con_pren: string (nullable = true)
 |-- num_emb: string (nullable = true)
 |-- num_par: string (nullable = true)
 |-- hij_viv: integer (nullable = true)
 |-- hij_vivm: string (nullable = true)
 |-- hij_nacm: string (nullable = true)
 |-- etnia: string (nullable = true)
 |-- est_civil: string (nullable = true)
 |-- niv_inst: string (nullable = true)
 |-- sabe_leer: string (nullable = true)
 |-- prov_res: string (nullable = true)
 |-- cant_res: string (nullable = true)
 |-- parr_res: string (nullable = true)
 |-- area_res: string (nullable = true)
 |-- residente: string (nullable = true)
 |-- fecha_insc_date: date (nullable = true)
 |-- fecha_nac_date: date (nullable = true)
 |-- fecha_mad_date: date (nullable = true)
```

Took 1 second. Last updated by anonymous at June 18 2025, 8:29:15 PM.

SPARK JOB FINISHED

```
import org.apache.spark.sql.types._
import org.apache.spark.sql.functions._

val dateColumnNames = dfNacidosClean
    .schema
    .fields
    .filter {
        case StructField (_, TimestampType | DateType, _, _ ) => true
        case _=> false
    }
    .map(_.name)

// Ejemplo de calculo de estadisticas comunes
val statsExprs = dateColumnNames
    .flatMap { colName =>
        Seq(
            count(col(colName)).alias(s"${colName}_count"),
            min(col(colName)).alias(s"${colName}_min"),
            max(col(colName)).alias(s"${colName}_max"),
            countDistinct(col(colName)).alias(s"${colName}_countDistinct")
        )
    }
```

```
val dfStatsDateCols = dfNacidosClean
    .select(statsExprs:_*)

dfStatsDateCols.show()
```

```
+--------------------+-------------------+-------------------+----------------------------+-------------------+-------------------+-------------------+---------------------------+--------------------+-----------
|fecha_insc_date_count|fecha_insc_date_min|fecha_insc_date_max|fecha_insc_date_countDistinct|fecha_nac_date_count|fecha_nac_date_min|fecha_nac_date_max|fecha_nac_date_countDistinct|fecha_mad_date_count|fecha_mad_c
+--------------------+-------------------+-------------------+----------------------------+-------------------+-------------------+-------------------+---------------------------+--------------------+-----------
|              231333|         2001-01-01|         2024-04-30|                         452|             241295|         1900-01-01|         2023-12-31|                       2028|              239787|        192
+--------------------+-------------------+-------------------+----------------------------+-------------------+-------------------+-------------------+---------------------------+--------------------+-----------

import org.apache.spark.sql.types._
import org.apache.spark.sql.functions._
dateColumnNames: Array[String] = Array(fecha_insc_date, fecha_nac_date, fecha_mad_date)
statsExprs: Array[org.apache.spark.sql.Column] = Array(count(fecha_insc_date) AS fecha_insc_date_count, min(fecha_insc_date) AS fecha_insc_date_min, max(fecha_insc_date) AS fecha_insc_date_max, count(fecha_insc_da
```

Took 4 seconds. Last updated by anonymous at June 25 2025, 6:43:09 PM.

SPARK JOB FINISHED

```
dfNacidosClean
    .select(min($"fecha_insc_date"), max($"fecha_insc_date"), count($"fecha_insc_date"), countDistinct($"fecha_insc_date")).show
```

```
+-------------------+-------------------+--------------------+------------------------------+
|min(fecha_insc_date)|max(fecha_insc_date)|count(fecha_insc_date)|count(DISTINCT fecha_insc_date)|
+-------------------+-------------------+--------------------+------------------------------+
|         2001-01-01|         2024-04-30|              231333|                           452|
+-------------------+-------------------+--------------------+------------------------------+
```

Took 3 seconds. Last updated by anonymous at June 25 2025, 6:49:24 PM.

SPARK JOB FINISHED

```
dfNacidos.select(
    mode($"anio_nac"),
    median($"anio_nac"),
    stddev($"anio_nac"),
    stddev_pop($"anio_nac"), // divide N
    stddev_samp($"anio_nac")) // divide N - 1
    .show
```

```
+-------------+---------------+--------------------+--------------------+--------------------+
|mode(anio_nac)|median(anio_nac)|stddev_samp(anio_nac)|stddev_pop(anio_nac)|stddev_samp(anio_nac)|
+-------------+---------------+--------------------+--------------------+--------------------+
|         2023|         2023.0|   1.5180530471163742|   1.5180499014759343|   1.5180530471163742|
+-------------+---------------+--------------------+--------------------+--------------------+
```

Took 2 seconds. Last updated by anonymous at June 25 2025, 6:56:02 PM. (outdated)

FINISHED

1.clasificar las columnas enteras, seleccionar o ver
2.Ver columnas que se supone que deben ser enteras pero estan como String

QUE VAMOS A HACER
Obtener los valores distintos de una columna
LUEGO
agrupar y contar cuantos de esos valores distintos existen

Took 0 seconds. Last updated by anonymous at June 25 2025, 7:10:52 PM.

SPARK JOB FINISHED

```
import org.apache.spark.sql.types._
import org.apache.spark.sql.functions._

// 1. Filtrar columnas Integer y String
val columnasEnterasYTexto = dfNacidosClean.schema.fields
  .filter(f => f.dataType == IntegerType || f.dataType == StringType)
  .map(_.name)

// 2. Crear expresiones de agregación para cada columna
val statsExprs = columnasEnterasYTexto
  .flatMap { colName =>
    Seq(
      count(col(colName)).alias(s"${colName}_count"),
      countDistinct(col(colName)).alias(s"${colName}_countDistinct")
    )
  }

// 3. Aplicar select con las expresiones
val dfStatsIntStrCols = dfNacidosClean.select(statsExprs: _*)

// 4. Mostrar resultados
dfStatsIntStrCols.show
```

```
+--------------+----------------------+--------------+----------------------+--------------+----------------------+--------------+----------------------+--------------+----------------------+----------
|prov_insc_count|prov_insc_countDistinct|cant_insc_count|cant_insc_countDistinct|parr_insc_count|parr_insc_countDistinct|fecha_insc_count|fecha_insc_countDistinct|anio_insc_count|anio_insc_countDistinct|mes_insc_c
+--------------+----------------------+--------------+----------------------+--------------+----------------------+--------------+----------------------+--------------+----------------------+----------
|        241295|                    25|        241295|                   169|        241295|                   254|        241295|                   453|        241295|                     4|        24
+--------------+----------------------+--------------+----------------------+--------------+----------------------+--------------+----------------------+--------------+----------------------+----------

import org.apache.spark.sql.types._
import org.apache.spark.sql.functions._
columnasEnterasYTexto: Array[String] = Array(prov_insc, cant_insc, parr_insc, fecha_insc, anio_insc, mes_insc, dia_insc, sexo, fecha_nac, anio_nac, mes_nac, dia_nac, talla, peso, sem_gest, tipo_part, lugar_ocur, a
statsExprs: Array[org.apache.spark.sql.Column] = Array(count(prov_insc) AS prov_insc_count, count(prov_insc) AS prov_insc_countDistinct, count(cant_insc) AS cant_insc_co...
```

Took 52 seconds. Last updated by anonymous at June 25 2025, 8:35:32 PM.

SPARK JOB FINISHED

```
import org.apache.spark.sql.functions._
import org.apache.spark.sql.types._


val columnasInt = Seq("Anio_nac", "Dia_nac", "hij_viv")

val statsExprsInt = columnasInt.flatMap { colName =>
  Seq(
    count(col(colName)).alias(s"${colName}_count"),
    countDistinct(col(colName)).alias(s"${colName}_countDistinct")
```

```
  )
}

val dfStatsIntegerCols = dfNacidosClean.select(statsExprsInt: _*)
dfStatsIntegerCols.show
```

```
+--------------+----------------------+-------------+--------------------+-------------+--------------------+
|Anio_nac_count|Anio_nac_countDistinct|Dia_nac_count|Dia_nac_countDistinct|hij_viv_count|hij_viv_countDistinct|
+--------------+----------------------+-------------+--------------------+-------------+--------------------+
|        241295|                    41|       241295|                  31|       241295|                  15|
+--------------+----------------------+-------------+--------------------+-------------+--------------------+

import org.apache.spark.sql.functions._
import org.apache.spark.sql.types._
columnasInt: Seq[String] = List(Anio_nac, Dia_nac, hij_viv)
statsExprsInt: Seq[org.apache.spark.sql.Column] = List(count(Anio_nac) AS Anio_nac_count, count(Anio_nac) AS Anio_nac_countDistinct, count(Dia_nac) AS Dia_nac_count, count(Dia_nac) AS Dia_nac_countDistinct, count(
dfStatsIntegerCols: org.apache.spark.sql.DataFrame = [Anio_nac_count: bigint, Anio_nac_countDistinct: bigint ... 4 more fields]
```

Took 3 seconds. Last updated by anonymous at June 25 2025, 8:35:35 PM.

SPARK JOB FINISHED

```
// strig
val columnasString = Seq("Mes_nac", "Talla", "Peso", "Hij_Vivm", "Hij_Nacm","Anio_insc","Dia_insc","Sem_gest","Apgar1","Apgar5","Anio_mad","Dia_mad","Con_pren","Num_emb","Num_par","hij_nacm")

val statsExprsStringNum = columnasString.flatMap { colName =>
  Seq(
    count(col(colName)).alias(s"${colName}_count"),
    countDistinct(col(colName)).alias(s"${colName}_countDistinct")
  )
}

val dfSdfSnumCols = dfNacidosClean.select(statsExprsStringNum: _*)
dfSdfSnumCols.show
```

```
+-------------+---------------------+----------+-----------------+----------+-----------------+-------------+--------------------+-------------+--------------------+--------------+----------------------+
|Mes_nac_count|Mes_nac_countDistinct|Talla_count|Talla_countDistinct|Peso_count|Peso_countDistinct|Hij_Vivm_count|Hij_Vivm_countDistinct|Hij_Nacm_count|Hij_Nacm_countDistinct|Anio_insc_count|Anio_insc_countDistinc
+-------------+---------------------+----------+-----------------+----------+-----------------+-------------+--------------------+-------------+--------------------+--------------+----------------------
|       241295|                   12|    241295|               19|    241295|             2969|       241295|                   11|       241295|                  10|        241295|
+-------------+---------------------+----------+-----------------+----------+-----------------+-------------+--------------------+-------------+--------------------+--------------+----------------------

columnasString: Seq[String] = List(Mes_nac, Talla, Peso, Hij_Vivm, Hij_Nacm, Anio_insc, Dia_insc, Sem_gest, Apgar1, Apgar5, Anio_mad, Dia_mad, Con_pren, Num_emb, Num_par, hij_nacm)
statsExprsStringNum: Seq[org.apache.spark.sql.Column] = List(count(Mes_nac) AS Mes_nac_count, count(Mes_nac) AS Mes_nac_countDistinct, count(Talla) AS Talla_count, count(Talla) AS Talla_countDistinct, count(Peso)
```

Took 4 seconds. Last updated by anonymous at June 25 2025, 8:35:41 PM.

SPARK JOB FINISHED

```
// 3. Ver valores distintos y su frecuencia para columnas Integer
columnasInt.foreach { colName =>
  dfNacidosClean
    .groupBy(col(colName))
    .count()
    .orderBy(desc("count"))
    .show
}
```

```
+--------+------+
|Anio_nac| count|
+--------+------+
|    2023|238772|
|    2022|   757|
|    2021|   418|
|    2020|   320|
|    2019|   154|
|    2018|   123|
|    2013|   104|
|    2017|    97|
|    2014|    91|
|    2015|    82|
|    2016|    78|
|    2012|    62|
|    2011|    46|
|    2010|    35|
|    2006|    28|
|    2008|    26|
|    2009|    22|
|    2005|    22|
|    2007|    20|
|    1900|    15|
+--------+------+
only showing top 20 rows

+-------+-----+
|Dia_nac|count|
+-------+-----+
|      6| 8416|
|      7| 8243|
|     10| 8205|
|     27| 8167|
|     28| 8163|
|     20| 8131|
|      4| 8112|
|     17| 8101|
|     14| 8044|
|      5| 8041|
|     11| 8033|
|     13| 7997|
|      3| 7975|
|      8| 7960|
|      1| 7946|
|     18| 7915|
|     16| 7912|
|     23| 7908|
|     15| 7859|
|     24| 7838|
+-------+-----+
only showing top 20 rows

+-------+-----+
|hij_viv|count|
+-------+-----+
|      1|98215|
```

```
|      2|74535|
|      3|42847|
|      4|15666|
|      5| 5758|
|      6| 2341|
|      7| 1043|
|      8|  470|
|      9|  214|
|     10|  110|
|     11|   60|
|     12|   22|
|     13|    9|
|     14|    4|
|     16|    1|
+-------+-----+
```

Took 5 seconds. Last updated by anonymous at June 25 2025, 8:35:50 PM.

SPARK JOB FINISHED

```
// 4. Ver valores distintos String
columnasString.foreach { colName =>
  dfNacidosClean
    .groupBy(col(colName))
    .count()
    .orderBy(desc("count"))
    .show
}
```

```
+----------+-----+
|   Mes_nac|count|
+----------+-----+
|      Mayo|21344|
|     Marzo|21250|
|    Agosto|20774|
|     Julio|20755|
|     Abril|20630|
|     Junio|20576|
|     Enero|20572|
|Septiembre|19933|
|   Octubre|19502|
|   Febrero|18943|
| Diciembre|18558|
| Noviembre|18458|
+----------+-----+

+---------------+-----+
|          Talla|count|
+---------------+-----+
|             50|44547|
|             49|42690|
|             48|38292|
|             47|26371|
|             51|23937|
|             52|19596|
|             46|15009|
|Sin información| 8772|
|             45| 8337|
|             44| 4130|
|             43| 2441|
|             38| 1827|
|             42| 1669|
|             41| 1117|
|             40| 1090|
|             39|  584|
|             53|  497|
|             54|  206|
|             55|  183|
+---------------+-----+

+---------------+-----+
|           Peso|count|
+---------------+-----+
|           3000| 8786|
|Sin información| 8687|
|           3200| 6695|
|           3100| 5752|
|           3500| 5414|
|           3300| 4990|
|           2900| 4882|
|           3400| 4759|
|           2800| 4612|
|           2700| 3354|
|           3600| 3244|
|           2500| 2728|
|           2600| 2654|
|           3700| 2221|
|           3800| 1899|
|           3250| 1856|
|           3150| 1634|
|           3050| 1549|
|           2950| 1480|
|           2850| 1468|
+---------------+-----+
only showing top 20 rows

+---------------+------+
|       Hij_Vivm| count|
+---------------+------+
|              0|229313|
|Sin información|  7145|
|              1|  4282|
|              2|   424|
|              3|    78|
|              4|    38|
|              5|     5|
|              7|     4|
|              6|     4|
|              8|     1|
|             10|     1|
+---------------+------+

+---------------+------+
|       Hij_Nacm| count|
+---------------+------+
|              0|230153|
```

```
|Sin información|  7095|
|              1|  3498|
|              2|   438|
|              3|    85|
|              4|    18|
|              5|     4|
|              6|     2|
|              7|     1|
|             11|     1|
+---------------+------+

+---------+------+
|Anio_insc| count|
+---------+------+
|     2023|212079|
|     2024| 19068|
|         |  9962|
|     2001|   186|
+---------+------+

+--------+-----+
|Dia_insc|count|
+--------+-----+
|        | 9962|
|      27| 9007|
|      13| 8886|
|      17| 8614|
|      10| 8497|
|      22| 8399|
|      20| 8278|
|       6| 8032|
|      15| 7923|
|       8| 7893|
|       4| 7840|
|      19| 7838|
|      24| 7825|
|      16| 7819|
|       5| 7814|
|       7| 7780|
|      28| 7746|
|      14| 7746|
|      18| 7696|
|      23| 7537|
+--------+-----+
only showing top 20 rows

+--------------+-----+
|      Sem_gest|count|
+--------------+-----+
|            39|71046|
|            38|65833|
|            40|41479|
|            37|26020|
|Sin información| 9012|
|            41| 8912|
|            36| 7602|
|            35| 3372|
|            34| 2214|
|            33| 1325|
|            32| 1032|
|            42|  892|
|            31|  574|
|            30|  572|
|            29|  397|
|            28|  334|
|            27|  246|
|            26|  191|
|            25|  117|
|            24|   78|
+--------------+-----+
only showing top 20 rows

+--------------+------+
|         Apgar1| count|
+--------------+------+
|             8|146361|
|             9| 57316|
|             7| 19011|
|Sin información|  9471|
|             6|  4083|
|             5|  1676|
|            10|  1090|
|             4|   838|
|             3|   684|
|             1|   398|
|             2|   367|
+--------------+------+

+--------------+------+
|         Apgar5| count|
+--------------+------+
|             9|178774|
|            10| 33407|
|             8| 14944|
|Sin información|  9544|
|             7|  2673|
|             6|   848|
|             5|   476|
|             1|   209|
|             4|   198|
|             3|   141|
|             2|    81|
+--------------+------+

+--------+-----+
|Anio_mad|count|
+--------+-----+
|    1999|13106|
|    2000|12994|
|    2001|12607|
|    1997|12472|
|    2002|12318|
|    1996|12247|
|    1995|11891|
|    1994|11662|
|    1998|11611|
|    1993|11451|
|    2003|11184|
|    1992|10658|
```

```
|    2004| 9778|
|    1991| 9524|
|    1990| 8673|
|    1989| 7947|
|    2005| 7673|
|    1988| 7330|
|    2006| 6418|
|    1987| 6346|
+--------+-----+
only showing top 20 rows

+-------+-----+
|Dia_mad|count|
+-------+-----+
|     20| 9202|
|      1| 9143|
|     15| 8759|
|     10| 8683|
|     28| 8201|
|      5| 8193|
|     25| 8170|
|      8| 8041|
|     12| 8027|
|      2| 8007|
|     18| 7949|
|     24| 7916|
|      3| 7848|
|     17| 7804|
|      4| 7796|
|      6| 7700|
|     22| 7683|
|     14| 7680|
|     23| 7672|
|      7| 7658|
+-------+-----+
only showing top 20 rows

+---------------+-----+
|       Con_pren|count|
+---------------+-----+
|              5|50430|
|              8|34433|
|              6|33727|
|              7|29592|
|              9|19537|
|              4|18094|
|             10|14062|
|              3|10585|
|Sin información| 7251|
|              2| 6253|
|             12| 4374|
|              1| 3993|
|              0| 3049|
|             11| 2279|
|             15| 1349|
|             13|  739|
|             14|  636|
|             20|  311|
|             16|  192|
|             18|  113|
+---------------+-----+
only showing top 20 rows

+---------------+-----+
|        Num_emb|count|
+---------------+-----+
|              1|84118|
|              2|70144|
|              3|44323|
|              4|20196|
|              5| 8776|
|Sin información| 6693|
|              6| 3764|
|              7| 1698|
|              8|  783|
|              9|  399|
|             10|  191|
|             11|  108|
|             12|   55|
|             13|   30|
|             14|    9|
|             15|    3|
|             16|    2|
|             18|    1|
|             17|    1|
|             20|    1|
+---------------+-----+

+---------------+-----+
|        Num_par|count|
+---------------+-----+
|              1|91014|
|              2|74205|
|              3|42838|
|              4|16019|
|Sin información| 6693|
|              5| 5981|
|              6| 2429|
|              7| 1141|
|              8|  499|
|              9|  237|
|             10|  115|
|             11|   69|
|             12|   39|
|             13|   11|
|             14|    3|
|             15|    1|
|             16|    1|
+---------------+-----+

+---------------+------+
|       hij_nacm| count|
+---------------+------+
|              0|230153|
|Sin información|  7095|
|              1|  3498|
|              2|   438|
|              3|    85|
|              4|    18|
```

```
|              5|     4|
|              6|     2|
|              7|     1|
|             11|     1|
+---------------+------+
```

Took 21 seconds. Last updated by anonymous at June 25 2025, 8:36:16 PM.

READY