FINISHED

Took 3 sec. Last updated by anonymous at July 25 2025, 12:02:29 AM.

Estableciendo conexión con la base de datos

FINISHED

Took 0 sec. Last updated by anonymous at July 25 2025, 12:16:26 AM. (outdated)

SPARK JOB (http://10.0.2.15:4040/jobs/job?id=3) FINISHED

```
%spark
val url = "jdbc:mysql://localhost:3306/dataProyecto"
val table = "(SELECT * FROM MatrimonioFinal) AS tmp"

val properties = new java.util.Properties()
properties.setProperty("user", "root")
properties.setProperty("password", "Sirio1203")
properties.setProperty("driver", "com.mysql.cj.jdbc.Driver")

val df = spark.read.jdbc(url, table, properties)

// Muestra los primeros registros para verificar la conexión
df.show()
```

```
+------------+----------+---------+---------------+-----------+-----------+
|id_matrimonio|fecha_insc|hijos_rec|       mcap_bie|persona1_id|persona2_id|
+------------+----------+---------+---------------+-----------+-----------+
|           2|2024-02-01|        2|             No|          1|      65536|
|           3|2024-01-26|     null|             No|          2|      65537|
|           4|2024-02-06|        0|             No|          3|      65538|
|           5|2024-02-09|        0|             No|          4|      65539|
|           6|2024-01-31|        0|             No|          5|      65540|
|           7|2024-01-23|        1|             No|          6|      65541|
|           8|2024-02-16|        2|             No|          7|      65542|
|           9|2024-02-19|        0|             No|          8|      65543|
|          10|2024-01-29|     null|Sin información|          9|      65544|
|          11|2024-02-19|        0|             No|         10|      65545|
|           12|2024-01-26|     null|Sin información|         11|      65546|
|           13|2024-01-16|        1|             No|         12|      65547|
|           14|2024-02-02|        0|             No|         13|      65548|
|           15|2024-02-19|        0|             No|         14|      65549|
```

Took 1 sec. Last updated by anonymous at July 25 2025, 3:44:57 AM.

SPARK JOB (http://10.0.2.15:4040/jobs/job?id=4) FINISHED

```
%spark
// Configuración de conexión MySQL
val url = "jdbc:mysql://localhost:3306/dataProyecto"
val table = "(SELECT * FROM MatrimonioVistaCompleta) AS tmp"

val properties = new java.util.Properties()
properties.setProperty("user", "root")
properties.setProperty("password", "Sirio1203")
properties.setProperty("driver", "com.mysql.cj.jdbc.Driver")

// Crear DataFrame desde MySQL
val dfMatrimonioVista = spark.read.jdbc(url, table, properties)

// Mostrar resultados
dfMatrimonioVista.show()

// Registrar vista para usar en %sql
dfMatrimonioVista.createOrReplaceTempView("MatrimonioVistaCompleta")
```

```
+------------+----------+---------+--------------+----------+-----+----------+-----
+-------------+------------+--------+----------+-----------+----------+-----+----------
+-----+------------+------------+--------+----------+-----------+---------+------+---------
+----+-----------+
|id_matrimonio|fecha_insc|hijos_rec|       mcap_bie|persona1_id|sexo1|fecha_nac1|edad1|naciona
lidad1|estado_civil1|  etnia1|sabe_leer1|       tipo1|persona2_id|sexo2|fecha_nac2|edad2|nacio
nalidad2|estado_civil2|  etnia2|sabe_leer2|       tipo2|provincia|canton|parroquia|area|nombre
_pais|
+------------+----------+---------+--------------+----------+-----+----------+-----
+-------------+------------+--------+----------+-----------+----------+-----+----------
+-----+------------+------------+--------+----------+-----------+---------+------+---------
+----+-----------+
|            2|2024-02-01|        2|             No|          1|    M|1986-02-13|   37|      E
cuador|      Soltero| Mestizo|      true|CONTRAYENTE1|      65536|    M|1989-12-14|   34|
Ecuador|      Soltero| Mestizo|      true|CONTRAYENTE2|     null|  null|     null|null|
null|
|            3|2024-01-26|     null|             No|          2|    M|1989-07-01|   34|      E
```

Took 2 sec. Last updated by anonymous at July 25 2025, 3:49:08 AM.

FINISHED

```
// 1. Parámetros de conexión
val jdbcUrl = "jdbc:mysql://localhost:3306/dataProyecto"
val connectionProperties = new java.util.Properties()
connectionProperties.setProperty("user", "root")
connectionProperties.setProperty("password", "Sirio1203")

// 2. Carga la tabla Persona
val dfPersona = spark.read
  .jdbc(jdbcUrl, "Persona", connectionProperties)

// 3. Carga la tabla UbicacionResidencia
val dfUbicacion = spark.read
  .jdbc(jdbcUrl, "UbicacionResidencia", connectionProperties)

val dfIndicadores = spark.read
  .jdbc(jdbcUrl, "indicadores_provincia", connectionProperties)
```

```
jdbcUrl: String = jdbc:mysql://localhost:3306/dataProyecto
connectionProperties: java.util.Properties = {user=root, password=Sirio1203}
dfPersona: org.apache.spark.sql.DataFrame = [id_persona: int, sexo: string ... 15 more fields]
dfUbicacion: org.apache.spark.sql.DataFrame = [id: int, provincia: string ... 3 more fields]
dfIndicadores: org.apache.spark.sql.DataFrame = [provincia: string, pobreza_ingresos: double .
.. 2 more fields]
```

Took 0 sec. Last updated by anonymous at July 25 2025, 6:05:46 AM.

```
dfUbicacion.printSchema()
```

FINISHED

```
root
 |-- id: integer (nullable = true)
 |-- provincia: string (nullable = true)
 |-- canton: string (nullable = true)
 |-- parroquia: string (nullable = true)
 |-- area: string (nullable = true)
```

Took 0 sec. Last updated by anonymous at July 25 2025, 6:02:11 AM.

Consulta 1. Matrimonios durante meses específicos (Tendencias de temporada)

FINISHED

Took 0 sec. Last updated by anonymous at July 25 2025, 4:39:15 AM. (outdated)

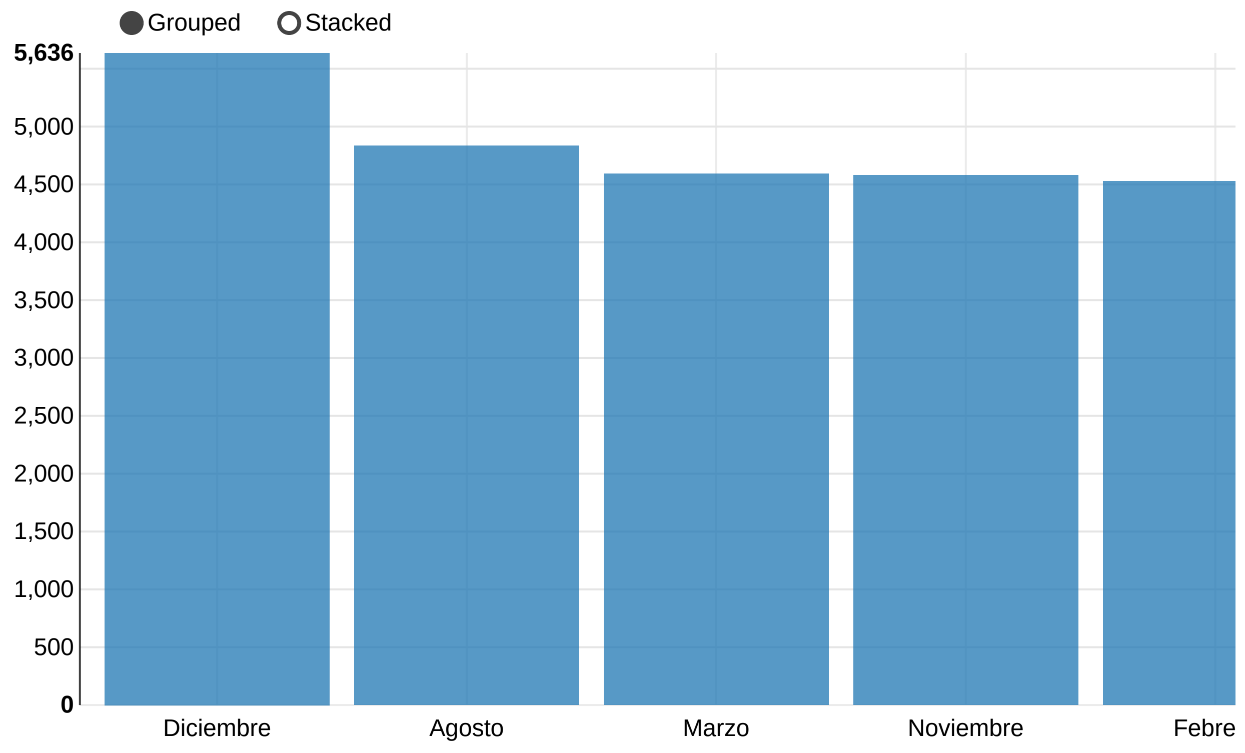
SPARK JOB FINISHED

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

val MatrimonioPorMes = dfMatrimonioVista
  .groupBy(month(col("fecha_insc")).alias("mes"))
  .agg(count("*").alias("total_matrimonios"))
  .withColumn("nombre_mes",
    when(col("mes") === 1, "Enero")
    .when(col("mes") === 2, "Febrero")
    .when(col("mes") === 3, "Marzo")
    .when(col("mes") === 4, "Abril")
    .when(col("mes") === 5, "Mayo")
    .when(col("mes") === 6, "Junio")
    .when(col("mes") === 7, "Julio")
    .when(col("mes") === 8, "Agosto")
    .when(col("mes") === 9, "Septiembre")
    .when(col("mes") === 10, "Octubre")
    .when(col("mes") === 11, "Noviembre")
    .when(col("mes") === 12, "Diciembre")
    .otherwise("Desconocido"))
  .orderBy(desc("total_matrimonios"))

z.show(MatrimonioPorMes.select("nombre_mes", "total_matrimonios"))
```

settings ▾



```
import org.apache.spark.sql.functions._
MatrimonioPorMes: org.apache.spark.sql.Dataset[org.apache.spark.sql.Row] = [mes: int, total_ma
trimonios: bigint ... 1 more field]
```

Took 1 sec. Last updated by anonymous at July 25 2025, 4:36:22 AM. (outdated)

Consulta 2: Matrimonios por etnia (etnia1, etnia2)

FINISHED

Took 0 sec. Last updated by anonymous at July 25 2025, 4:48:18 AM. (outdated)

SPARK JOB FINISHED

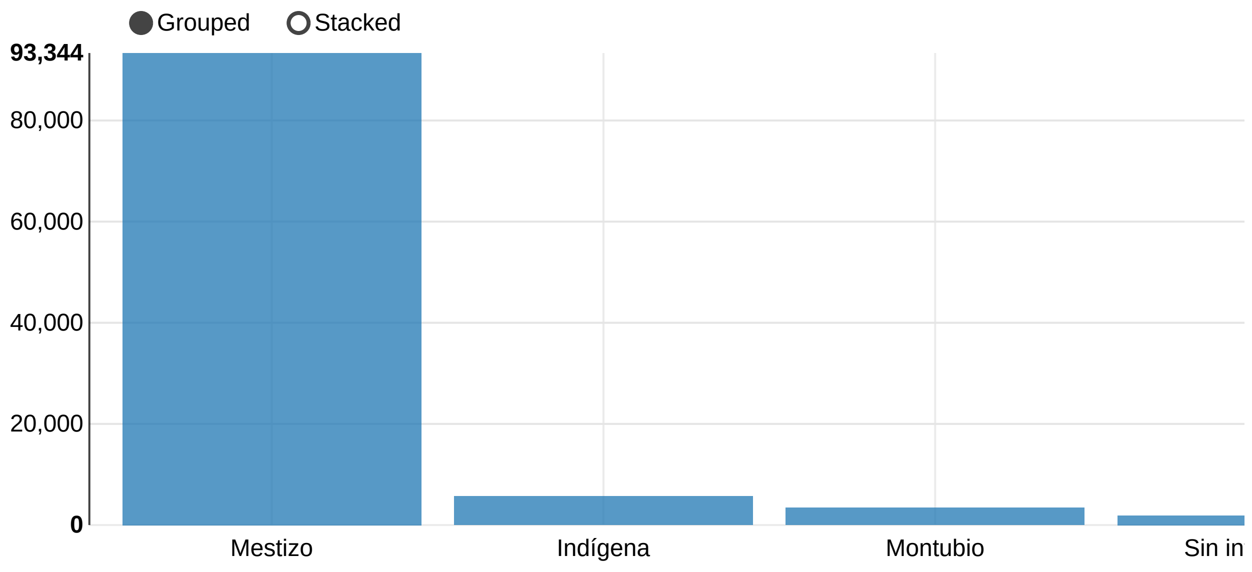
```
%spark
val etnia1 = dfMatrimonioVista
  .select(col("etnia1").alias("etnia"))
  .filter(col("etnia").isNotNull)

val etnia2 = dfMatrimonioVista
  .select(col("etnia2").alias("etnia"))
  .filter(col("etnia").isNotNull)

val matrimoniosPorEtnia = etnia1.union(etnia2)
  .groupBy("etnia")
  .count()
  .orderBy(desc("count"))

z.show(matrimoniosPorEtnia)
```

settings



```
etnia1: org.apache.spark.sql.Dataset[org.apache.spark.sql.Row] = [etnia: string]
etnia2: org.apache.spark.sql.Dataset[org.apache.spark.sql.Row] = [etnia: string]
matrimoniosPorEtnia: org.apache.spark.sql.Dataset[org.apache.spark.sql.Row] = [etnia: string,
count: bigint]
```

Took 1 sec. Last updated by anonymous at July 25 2025, 4:48:34 AM. (outdated)

Consulta 3: Matrimonios por provincia (provincia)

FINISHED

Took 0 sec. Last updated by anonymous at July 25 2025, 4:49:58 AM. (outdated)

SPARK JOB FINISHED

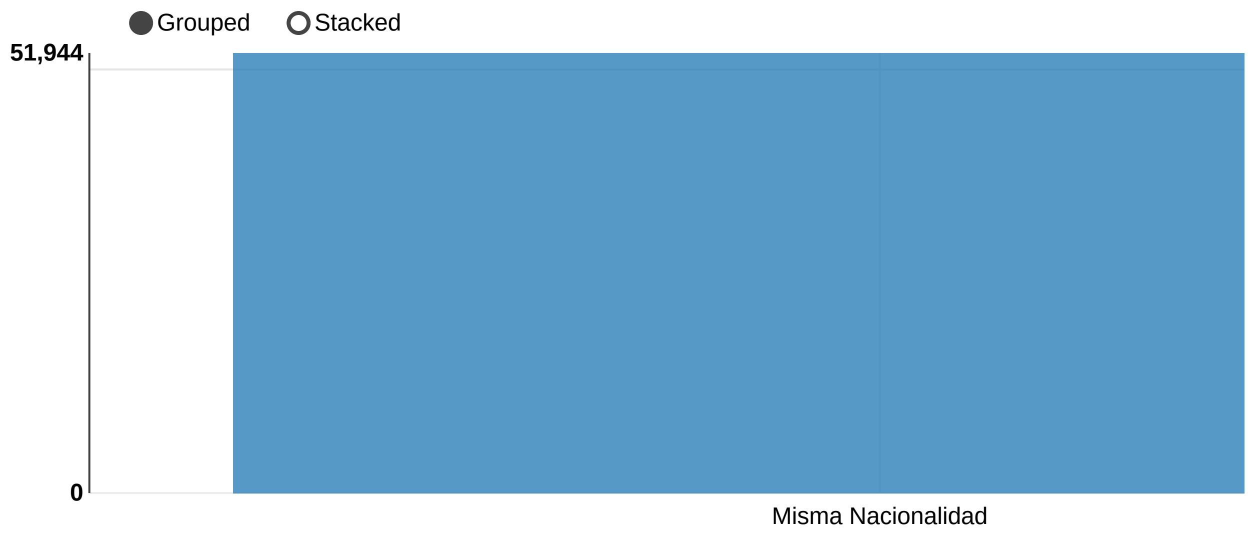
```
%spark
import org.apache.spark.sql.functions._


val NacionalidadMatrimonio = dfMatrimonioVista
  .withColumn("nationality_match",
    when(col("nacionalidad1") === col("nacionalidad2"), "Misma Nacionalidad")
     .otherwise("Diferente Nacionalidad"))
  .groupBy("nationality_match")
  .agg(count("*").alias("total_matrimonios"))
  .orderBy(desc("total_matrimonios"))

NacionalidadMatrimonio.show()
z.show(NacionalidadMatrimonio)
```

```
+--------------------+-----------------+
|   nationality_match|total_matrimonios|
+--------------------+-----------------+
|  Misma Nacionalidad|            51944|
|Diferente Naciona...|             1869|
+--------------------+-----------------+
```

settings ▾



```
import org.apache.spark.sql.functions._
NacionalidadMatrimonio: org.apache.spark.sql.Dataset[org.apache.spark.sql.Row] = [nationality_
match: string, total_matrimonios: bigint]
```

Took 2 sec. Last updated by anonymous at July 25 2025, 5:08:07 AM. (outdated)

Consulta 4: Top 10 provincias con más matrimonios

FINISHED

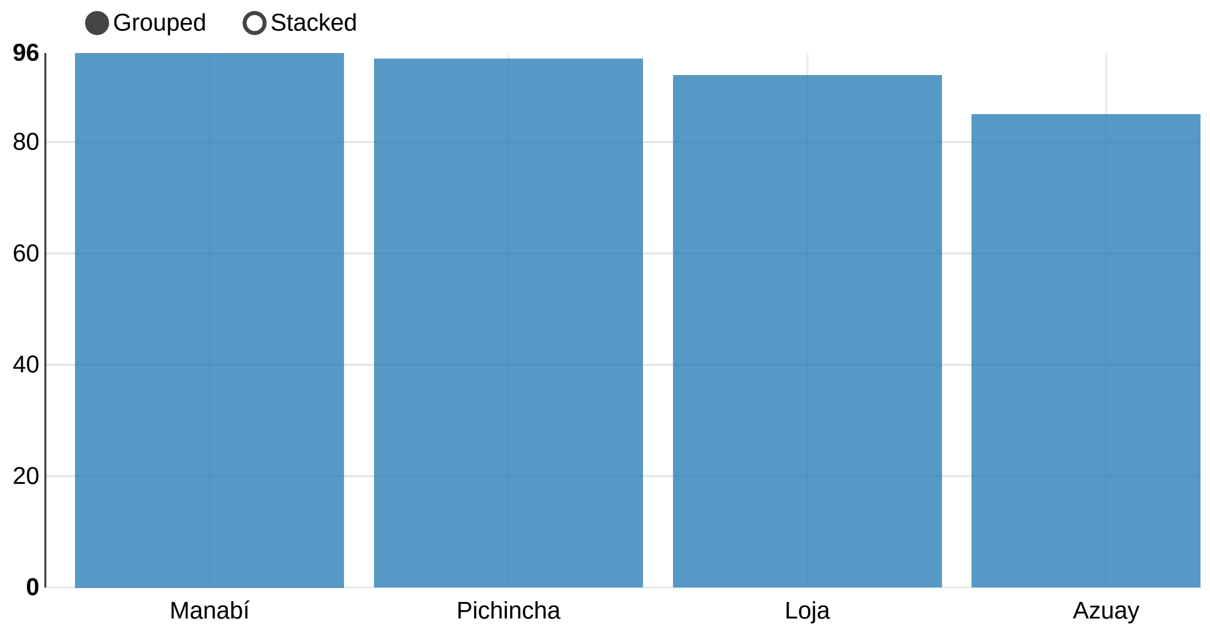
Took 0 sec. Last updated by anonymous at July 25 2025, 6:03:39 AM. (outdated)

SPARK JOB FINISHED

```
%spark
val dfProvinciasTop = dfUbicacion
  .groupBy("provincia")
  .agg(count("*").alias("total_matrimonios"))
  .orderBy(desc("total_matrimonios"))
  .limit(10)

z.show(dfProvinciasTop)
```

settings



```
dfProvinciasTop: org.apache.spark.sql.Dataset[org.apache.spark.sql.Row] = [provincia: string,
total_matrimonios: bigint]
```

Took 1 sec. Last updated by anonymous at July 25 2025, 6:02:38 AM. (outdated)

Consulta 5: Relación entre pobreza (NBI) e IDH por provincia

FINISHED

Took 0 sec. Last updated by anonymous at July 25 2025, 7:31:27 AM.

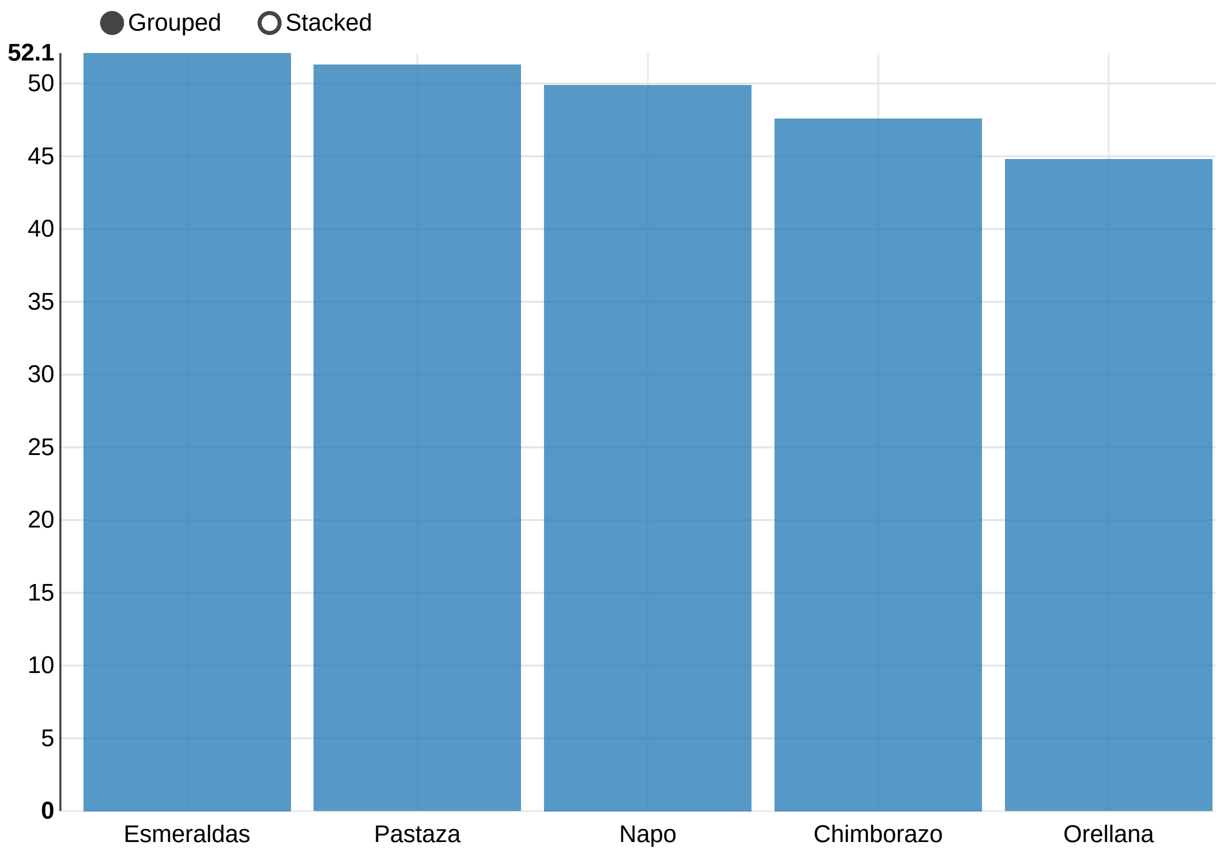
SPARK JOB FINISHED

```
%spark

val dfCruce = dfMatrimonioVista
  .select("provincia")
  .distinct()
  .join(dfIndicadores, Seq("provincia"))

z.show(dfCruce.select("provincia", "pobreza_nbi", "idh").orderBy(desc("pobreza_nbi")))
```

settings



```
dfCruce: org.apache.spark.sql.DataFrame = [provincia: string, pobreza_ingresos: double ... 2 m
ore fields]
```

Took 2 sec. Last updated by anonymous at July 25 2025, 7:26:43 AM. (outdated)