

# Proyecto Final

SPARK JOB FINISHED

```
val df1 = spark.read
  .option("header", "true")
  .option("inferSchema", "true")
  .option("delimiter", ";")
  .csv("/workspace/Programacion-Avanzada/2_MINEDUC_RegistrosAdministrativos_2022-2023Fin.csv")

val df2 = spark.read
  .option("header", "true")
  .option("inferSchema", "true")
  .option("delimiter", ";")
  .csv("/workspace/Programacion-Avanzada/registro-administrativo-historico_2009-2024-inicio.csv")

val df3 = spark.read
  .option("header", "true")
  .option("inferSchema", "true")
  .option("delimiter", ";")
  .csv("/workspace/Programacion-Avanzada/ineval_sereestudiante2022_2023_2023diriamhne.csv")

df1: org.apache.spark.sql.DataFrame = [Año Lectivo: string, AMIE: string ... 248 more fields]
df2: org.apache.spark.sql.DataFrame = [Anio_lectivo: string, Zona: string ... 25 more fields]
df3: org.apache.spark.sql.DataFrame = [ciclo: string, grado: int ... 56 more fields]
```

Took 3 sec. Last updated by anonymous at July 24 2025, 4:41:23 PM.

FINISHED

```
val df1_clean = df1.select(
  "AMIE",
  "Año Lectivo",
  "Provincia",
  "Canton",
  "Total_Estudiantes",
  "Total_Docentes"
).withColumnRenamed("Año Lectivo", "anio_lectivo")

df1_clean: org.apache.spark.sql.DataFrame = [AMIE: string, anio_lectivo: string ... 4 more fields]
```

Took 0 sec. Last updated by anonymous at July 24 2025, 4:44:12 PM.

FINISHED

```
val df2_clean = df2.select(
  "AMIE",
  "Anio_lectivo",
  "Provincia",
  "Canton",
  "Total_Estudiantes",
  "Total_Docentes"
).withColumnRenamed("Anio_lectivo", "anio_lectivo")

df2_clean: org.apache.spark.sql.DataFrame = [AMIE: string, anio_lectivo: string ... 4 more fields]
```

Took 1 sec. Last updated by anonymous at July 24 2025, 4:44:45 PM.

FINISHED

```
val df3_clean = df3.select(
  "amie",
  "ciclo",
  "grado",
  "estado_eval",
  "isec", // Índice de desempeño escolar
  "imat", // Matemática
  "ilyl", // Lengua y Literatura
  "icn", // Ciencias Naturales
  "ies" // Estudios Sociales
).withColumnRenamed("amie", "AMIE")

df3_clean: org.apache.spark.sql.DataFrame = [AMIE: string, ciclo: string ... 7 more fields]
```

Took 0 sec. Last updated by anonymous at July 24 2025, 4:44:55 PM.

FINISHED

```
// Renombra en df1_clean
val df1_ready = df1_clean
  .withColumnRenamed("Canton", "Canton")
  .withColumnRenamed("Provincia", "Provincia") // (Solo si hay problemas con otros nombres)

val df2_ready = df2_clean

df1_ready: org.apache.spark.sql.DataFrame = [AMIE: string, anio_lectivo: string ... 4 more fields]
df2_ready: org.apache.spark.sql.DataFrame = [AMIE: string, anio_lectivo: string ... 4 more fields]
```

Took 0 sec. Last updated by anonymous at July 24 2025, 4:46:22 PM. (outdated)

FINISHED

```
val df12 = df1_ready.join(df2_ready, Seq("AMIE", "anio_lectivo", "Provincia", "Canton"), "outer")

df12: org.apache.spark.sql.DataFrame = [AMIE: string, anio_lectivo: string ... 6 more fields]
```

Took 0 sec. Last updated by anonymous at July 24 2025, 4:46:49 PM.

FINISHED

```
val df_final = df12.join(df3_clean, Seq("AMIE"), "outer")

df_final: org.apache.spark.sql.DataFrame = [AMIE: string, anio_lectivo: string ... 14 more fields]
```

Took 0 sec. Last updated by anonymous at July 24 2025, 4:47:13 PM.

```
df_final.show(10)
```

SPARK JOB FINISHED

# Proyecto Final

AMIE	anio_lectivo	Provincia	Canton	Total_Estudiantes	Total_Docentes	Total_Estudiantes	Total_Docentes	ciclo	grado	estado_eval	isec	imat	ily1	icn	ies
null	null	null	null	null	null	null	null	null	null	null	null	null	null	null	null
null	null	null	null	null	null	null	null	null	null	null	null	null	null	null	null
null	null	null	null	null	null	null	null	null	null	null	null	null	null	null	null
null	null	null	null	null	null	null	null	null	null	null	null	null	null	null	null
null	null	null	null	null	null	null	null	null	null	null	null	null	null	null	null
null	null	null	null	null	null	null	null	null	null	null	null	null	null	null	null
null	null	null	null	null	null	null	null	null	null	null	null	null	null	null	null
null	null	null	null	null	null	null	null	null	null	null	null	null	null	null	null
null	null	null	null	null	null	null	null	null	null	null	null	null	null	null	null
null	null	null	null	null	null	null	null	null	null	null	null	null	null	null	null
null	null	null	null	null	null	null	null	null	null	null	null	null	null	null	null
null	null	null	null	null	null	null	null	null	null	null	null	null	null	null	null

only showing top 10 rows

Took 4 sec. Last updated by anonymous at July 24 2025, 4:47:26 PM.

```
df1_ready.select("AMIE", "anio_lectivo", "Provincia", "Canton").show(10, false)
df2_ready.select("AMIE", "anio_lectivo", "Provincia", "Canton").show(10, false)
df3_clean.select("AMIE").show(10, false)
```

SPARK JOB FINISHED

AMIE	anio_lectivo	Provincia	Canton
01H01829	2022-2023 Fin	AZUAY	CUENCA
04H00462	2022-2023 Fin	CARCHI	ESPEJO
08H01180	2022-2023 Fin	SANTO DOMINGO DE LOS TSACHILAS	LA CONCORDIA
08H01524	2022-2023 Fin	ESMERALDAS	ATACAMES
08H02161	2022-2023 Fin	ESMERALDAS	QUININDE
09H06652	2022-2023 Fin	GUAYAS	GUAYAQUIL
10H00070	2022-2023 Fin	IMBABURA	IBARRA
11H02033	2022-2023 Fin	LOJA	SARAGURO
12H02246	2022-2023 Fin	LOS RIOS	PUEBLOVIEJO
17H04382	2022-2023 Fin	PICHINCHA	MEJIA

only showing top 10 rows

AMIE anio\_lectivo Provincia Canton

Took 0 sec. Last updated by anonymous at July 24 2025, 4:48:49 PM.

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

FINISHED

```
val df1_norm = df1_ready
  .withColumn("AMIE", trim(upper(col("AMIE"))))
  .withColumn("anio_lectivo", trim(upper(col("anio_lectivo"))))
  .withColumn("Provincia", trim(upper(col("Provincia"))))
  .withColumn("Canton", trim(upper(col("Canton"))))

val df2_norm = df2_ready
  .withColumn("AMIE", trim(upper(col("AMIE"))))
  .withColumn("anio_lectivo", trim(upper(col("anio_lectivo"))))
  .withColumn("Provincia", trim(upper(col("Provincia"))))
  .withColumn("Canton", trim(upper(col("Canton"))))

val df3_norm = df3_clean
  .withColumn("AMIE", trim(upper(col("AMIE"))))
```

```
import org.apache.spark.sql.functions._
df1_norm: org.apache.spark.sql.DataFrame = [AMIE: string, anio_lectivo: string ... 4 more fields]
df2_norm: org.apache.spark.sql.DataFrame = [AMIE: string, anio_lectivo: string ... 4 more fields]
df3_norm: org.apache.spark.sql.DataFrame = [AMIE: string, ciclo: string ... 7 more fields]
```

Took 0 sec. Last updated by anonymous at July 24 2025, 4:50:05 PM.

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

FINISHED

```
val cols_finales = Seq("AMIE", "anio_lectivo", "Provincia", "Canton", "Total_Estudiantes", "Total_Docentes")

val df3_union = df3_norm
  .withColumn("anio_lectivo", lit(null).cast("string"))
  .withColumn("Provincia", lit(null).cast("string"))
  .withColumn("Canton", lit(null).cast("string"))
  .withColumn("Total_Estudiantes", lit(null).cast("int"))
  .withColumn("Total_Docentes", lit(null).cast("int"))
  .select(cols_finales.head, cols_finales.tail: *)
```

```
import org.apache.spark.sql.functions._
cols_finales: Seq[String] = List(AMIE, anio_lectivo, Provincia, Canton, Total_Estudiantes, Total_Docentes)
df3_union: org.apache.spark.sql.DataFrame = [AMIE: string, anio_lectivo: string ... 4 more fields]
```

Took 0 sec. Last updated by anonymous at July 24 2025, 4:50:51 PM.

```
val df_final_union = df1_norm
  .select(cols_finales.head, cols_finales.tail: *)
  .unionByName(
    df2_norm.select(cols_finales.head, cols_finales.tail: *)
  )
  .unionByName(
    df3_union
  )
```

SPARK JOB (<http://10.0.5.2:4040/jobs/job?id=38>) FINISHED

df\_final\_union.show(10, false)

ProjectoFinal

AMIE	anio_lectivo	Provincia	Canton	Total_Estudiantes	Total_Docentes
01H01829	2022-2023	FIN AZUAY	CUENCA	7.0	3
04H00462	2022-2023	FIN CARCHI	ESPEJO	17.0	3
08H01180	2022-2023	FIN SANTO DOMINGO DE LOS TSACHILAS	LA CONCORDIA	79.0	15
08H01524	2022-2023	FIN ESMERALDAS	ATACAMES	131.0	25
08H02161	2022-2023	FIN ESMERALDAS	QUININDE	19.0	3
09H06652	2022-2023	FIN GUAYAS	GUAYAQUIL	86.0	5
10H00070	2022-2023	FIN IMBABURA	IBARRA	38.0	12
11H02033	2022-2023	FIN LOJA	SARAGURO	4.0	1
12H02246	2022-2023	FIN LOS RIOS	PUEBLOVIEJO	10.0	5
17H04382	2022-2023	FIN PICHINCHA	MEJIA	50.0	9

only showing top 10 rows

df\_final\_union: org.apache.spark.sql.Dataset[org.apache.spark.sql.Row] = [AMIE: string, anio\_lectivo: string ... 4 more fields]

Took 1 sec. Last updated by anonymous at July 24 2025, 4:51:23 PM.

df\_final\_union.write  
    .option("header", "true")  
    .mode("overwrite")  
    .csv("/workspace/Programacion-Avanzada/resultado\_final\_union.csv")

SPARK JOB (http://10.0.5.2:4040/jobs/job?id=39) FINISHED

Took 3 sec. Last updated by anonymous at July 24 2025, 4:51:41 PM.

df\_final\_union.createOrReplaceTempView("educacion\_union")

FINISHED

Took 0 sec. Last updated by anonymous at July 24 2025, 4:53:07 PM.

%sql  
SELECT Provincia, COUNT(\*) AS total\_registros, SUM(Total\_Estudiantes) AS suma\_estudiantes  
FROM educacion\_union  
GROUP BY Provincia  
ORDER BY suma\_estudiantes DESC

SPARK JOB FINISHED

settings

Provincia	total_registros	suma_e
PICHINCHA	34864	1.21681
MANABI	47680	685422
LOS RIOS	17888	390148
AZUAY	15887	347639
ESMERALDAS	17277	305954
EL ORO	11513	299858
TUNGURAHUA	6553	230865
SANTO DOMINGO DE LOS TSACHILAS	8062	218237

Took 1 sec. Last updated by anonymous at July 24 2025, 4:53:28 PM. (outdated)

Proceso completo de Integración, Limpieza y Carga de Datos  
Educativos a MySQL usando Apache Zeppelin, Spark y Bash

FINISHED

1. Carga de archivos en Spark

Se cargaron tres archivos CSV desde distintas fuentes oficiales:

- 2\_MINEDUC\_RegistrosAdministrativos\_2022-2023Fin.csv
- registro-administrativo-historico\_2009-2024-inicio.csv
- ineval\_serestudiante2022\_2023\_2023diciembre.csv

El objetivo era unificar la información relevante sobre instituciones educativas a nivel nacional.

## 2. Inspección y Estandarización de Columnas

Se identificaron y seleccionaron las columnas claves:

## ProyectoFinal

- AMIE (código de institución)
- anio\_lectivo
- Provincia
- Canton
- Total\_Estudiantes
- Total\_Docentes

Se estandarizaron nombres y se agregaron columnas nulas cuando fue necesario para compatibilidad.

## 3. Unión y Homologación de DataFrames

Se procesaron los tres DataFrames para garantizar que tuvieran la misma estructura.

Se usaron funciones como withColumnRenamed y withColumn en Spark/Scala.

Luego, se unificaron usando unionByName para obtener un solo DataFrame consolidado.

## 4. Exportación y Limpieza Final en Bash

El DataFrame final se exportó a un CSV único.

Se aplicaron scripts bash para:

- Reemplazar valores vacíos ( " " , , ) por NULL .
- Corregir líneas con problemas de formato.
- Eliminar filas basura o vacías.

Ejemplo de comandos utilizados:

```
sed -i 's/,"",/,NULL,/g; s/,"$/NULL/g; s/,,/,NULL,/g; s/,$/,NULL/' resultado_final_union_todo.csv
awk '!/^"/' resultado_final_union_todo.csv > resultado_final_union_todo_limpio.csv
sed -i 's/,""/$,NULL/' resultado_final_union_todo_limpio.csv
5. Carga a MySQL
El archivo limpio fue movido a /var/lib/mysql-files/ y cargado en una tabla SQL con:
```

```
sql
Copiar
Editar
LOAD DATA INFILE '/var/lib/mysql-files/resultado_final_union_todo_limpio.csv'
INTO TABLE union_educacion
FIELDS TERMINATED BY ','
OPTIONALLY ENCLOSED BY '"'
LINES TERMINATED BY '\n'
IGNORE 1 LINES
(AMIE, anio_lectivo, Provincia, Canton, Total_Estudiantes, Total_Docentes);
6. Validación y Consultas Iniciales
Se comprobó la carga correcta y la calidad de los datos con consultas SQL, por ejemplo:
SELECT COUNT(*) FROM union_educacion;
SELECT Provincia, COUNT(DISTINCT AMIE) as Instituciones FROM union_educacion GROUP BY Provincia;
SELECT * FROM union_educacion WHERE Total_Estudiantes IS NULL OR Total_Docentes IS NULL LIMIT 10;
```

Took 3 sec. Last updated by anonymous at July 24 2025, 5:18:36 PM.