Inicio

FINISHED

Extracción Inicial

Took 1 sec. Last updated by anonymous at October 12 2025, 4:48:21 PM.

FINISHED

```
%pyspark

df_CIE = spark.read.option("header", "true").csv("/data/raw/CIE10_2021.csv")
df_CIE.printSchema()
df_CIE.show(5)
```

FINISHED

```
%pyspark

df_Diabetes = spark.read.option("header", "true") \
    .option("sep", ";") \
    .csv("/data/raw/DF_ExLab_CExt_Diabetes.csv")
df_Diabetes.printSchema()
df_Diabetes.show(5)
```

```
root
 |-- FECHA_CORTE: string (nullable = true)
 |-- DEPARTAMENTO: string (nullable = true)
 |-- PROVINCIA: string (nullable = true)
 |-- DISTRITO: string (nullable = true)
 |-- UBIGEO: string (nullable = true)
 |-- RED: string (nullable = true)
 |-- IPRESS: string (nullable = true)
 |-- ID_PACIENTE: string (nullable = true)
 |-- EDAD_PACIENTE: string (nullable = true)
 |-- SEXO_PACIENTE: string (nullable = true)
 |-- EDAD_MEDICO: string (nullable = true)
 |-- ID_MEDICO: string (nullable = true)
 |-- COD_DIAG: string (nullable = true)
 |-- DIAGNOSTICO: string (nullable = true)
 |-- AREA_HOSPITALARIA: string (nullable = true)
 |-- SERVICIO_HOSPITALARIO: string (nullable = true)
 |-- ACTIVIDAD_HOSPITALARIA: string (nullable = true)
```

Took 4 sec. Last updated by anonymous at October 12 2025, 5:25:18 PM. (outdated)

FINISHED

```
%pyspark

df_Enf_Renal = spark.read.option("header", "true") \
    .option("sep", ";") \
    .csv("/data/raw/DF_ExLab_CExt_EnfermedadRenal.csv")
df_Enf_Renal.printSchema()
df_Enf_Renal.show(5)
```

FINISHED

```
%pyspark

df_Hiperlipidemia = spark.read.option("header", "true") \
    .option("sep", ";") \
    .csv("/data/raw/DF_ExLab_CExt_Hiperlipidemia.csv")
```

```
df_Hiperlipidemia.printSchema()
```

```
%pyspark

df_Plan_estrategico = spark.read.option("header", "true").csv("/data/raw/Datos_planeamiento_estrate
df_Plan_estrategico.printSchema()
df_Plan_estrategico.show(5)
```

FINISHED

```
root
 |-- Unnamed: 0: string (nullable = true)
 |-- Ubigeo: string (nullable = true)
 |-- Región / Provincia / Distrito: string (nullable = true)
 |-- Población total (2007): string (nullable = true)
 |-- Población total (2017) : string (nullable = true)
 |-- Población total (2020) : string (nullable = true)
 |-- Población de niños menores de un año (CENSO 2017) : string (nullable = true)
 |-- Población mayor de 80 años  (2020) : string (nullable = true)
 |-- Población con Discapacidad (2020) : string (nullable = true)
 |-- Superficie (km2)  : string (nullable = true)
 |-- Densidad (2020): string (nullable = true)
 |-- Capital legal  : string (nullable = true)
 |-- Altitud (msnm)  : string (nullable = true)
 |-- Latitud sur  : string (nullable = true)
 |-- Longitud oeste  : string (nullable = true)
 |-- Tipología de distrito según SDOT : string (nullable = true)
```

Took 0 sec. Last updated by anonymous at October 12 2025, 5:07:08 PM. (outdated)

```
%pyspark

df_Geodir = spark.read.option("header", "true").csv("/data/raw/geodir_ubigeo_inei_ubigeo_inei.csv")
df_Geodir.printSchema()
df_Geodir.show(5)
```

FINISHED

```
root
 |-- Ubigeo: string (nullable = true)
 |-- Distrito: string (nullable = true)
 |-- Provincia: string (nullable = true)
 |-- Departamento: string (nullable = true)
 |-- Poblacion: string (nullable = true)
 |-- Superficie: string (nullable = true)
 |-- Y: string (nullable = true)
 |-- X: string (nullable = true)
+------+-----------+-----------+------------+---------+----------+-------+--------+
|Ubigeo|   Distrito|  Provincia|Departamento|Poblacion|Superficie|      Y|       X|
+------+-----------+-----------+------------+---------+----------+-------+--------+
| 10101|Chachapoyas|Chachapoyas|    Amazonas|    29171|    153.78|-6.2294|-77.8714|
| 10102|   Asuncion|Chachapoyas|    Amazonas|      288|     25.71|-6.0317|-77.7122|
| 10103|     Balsas|Chachapoyas|    Amazonas|     1644|    357.09|-6.8375|-78.0214|
| 10104|      Cheto|Chachapoyas|    Amazonas|      591|     56.97|-6.2558|-77.7003|
| 10105|  Chiliquin|Chachapoyas|    Amazonas|      687|    143.43|-6.0778|-77.7392|
```

Took 0 sec. Last updated by anonymous at October 12 2025, 5:07:05 PM. (outdated)

```
%pyspark

df_Ipress = spark.read.option("header", "true").csv("/data/raw/ipress_Listado_de_Establecimientos.c
df_Ipress.printSchema()
df_Ipress.show(5)
```

FINISHED

```
root
 |-- Institución: string (nullable = true)
 |-- Código Único: string (nullable = true)
 |-- Nombre del establecimiento: string (nullable = true)
 |-- Clasificación: string (nullable = true)
 |-- Tipo: string (nullable = true)
 |-- Departamento: string (nullable = true)
 |-- Provincia: string (nullable = true)
 |-- Distrito: string (nullable = true)
 |-- UBIGEO: string (nullable = true)
 |-- Dirección: string (nullable = true)
 |-- Código DISA: string (nullable = true)
 |-- Código Red: string (nullable = true)
 |-- Código Microrred: string (nullable = true)
 |-- DISA: string (nullable = true)
 |-- Red: string (nullable = true)
 |-- Microrred: string (nullable = true)
 |-- Código UE: string (nullable = true)
```

Took 0 sec. Last updated by anonymous at October 12 2025, 5:07:12 PM. (outdated)

FINISHED

```
%pyspark

df_CIE.show(5)
```

```
+------+--------------------+
|CODIGO|         DESCRIPCION|
+------+--------------------+
| Y21.4|Ahogamiento y sum...|
| Y21.5|Ahogamiento y sum...|
| Y21.6|Ahogamiento y sum...|
| Y21.7|Ahogamiento y sum...|
| Y21.8|Ahogamiento y sum...|
+------+--------------------+
only showing top 5 rows
```

Took 0 sec. Last updated by anonymous at October 12 2025, 5:10:03 PM.

FINISHED

```
%pyspark

df_CIE = df_CIE.withColumnRenamed("CODIGO", "cod_enfermedad") \
               .withColumnRenamed("DESCRIPCION", "des_enfermedad")

df_CIE.show(5)
```

```
+--------------+--------------------+
|cod_enfermedad|      des_enfermedad|
+--------------+--------------------+
|         Y21.4|Ahogamiento y sum...|
|         Y21.5|Ahogamiento y sum...|
|         Y21.6|Ahogamiento y sum...|
|         Y21.7|Ahogamiento y sum...|
|         Y21.8|Ahogamiento y sum...|
+--------------+--------------------+
only showing top 5 rows
```

Took 0 sec. Last updated by anonymous at October 12 2025, 5:48:00 PM.

FINISHED

```
%pyspark
df_Diabetes.show(5)
```

```
+-----------+------------+----------------+-----------+------+--------------------+-------------
------+--------------------+-------------+-------------+-----------+--------------------+--------
+--------------------+-----------------+---------------------+----------------------+------------
-+---------------+--------------------+-----------+----------+--------------+-------------------
-+-----------+----------+
|FECHA_CORTE|DEPARTAMENTO|       PROVINCIA|   DISTRITO|UBIGEO|                 RED|
IPRESS|         ID_PACIENTE|EDAD_PACIENTE|SEXO_PACIENTE|EDAD_MEDICO|           ID_MEDICO|COD_DIAG
|         DIAGNOSTICO|AREA_HOSPITALARIA|SERVICIO_HOSPITALARIO|ACTIVIDAD_HOSPITALARIA|FECHA_MUESTR
A|FEC_RESULTADO_1|     PROCEDIMIENTO_1|RESULTADO_1|UNIDADES_1|FEC_RESULTADO_2|     PROCEDIMIENTO_
2|RESULTADO_2|UNIDADES_2|
+-----------+------------+----------------+-----------+------+--------------------+-------------
------+--------------------+-------------+-------------+-----------+--------------------+--------
+--------------------+-----------------+---------------------+----------------------+------------
-+---------------+--------------------+-----------+----------+--------------+-------------------
-+-----------+----------+
|   20240531|     UCAYALI|CORONEL PORTILLO|   MANANTAY|250107|RED ASISTENCIAL U...|      CAP I MA
NANTAY|eJwzNDAwtDC0NDMxN...|           51|    MASCULINO|         26|eJwzNjA2MzE2NLY0N...|   E11.9
|DIABETES MELLITUS   | CONSULTA EXTERNA|     MEDICINA GENERAL|  ATENCION  MEDICA   |     2020010
```

Took 2 sec. Last updated by anonymous at October 12 2025, 5:26:43 PM.

FINISHED

```
%pyspark

from pyspark.sql.functions import col, row_number, monotonically_increasing_id
from pyspark.sql.window import Window
from pyspark.sql.functions import to_date, col

# Creamos una ventana sin partición ni orden específico
windowSpec = Window.orderBy(monotonically_increasing_id())

df_Diabetes_main = df_Diabetes.select(
    col("COD_DIAG").alias("cod_enfermedad"),
    col("ID_PACIENTE").alias("id_paciente"),
    col("IPRESS").alias("cod_institucion"),
    col("ID_MEDICO").alias("id_medico"),
    col("SERVICIO_HOSPITALARIO").alias("servicio_hospitalario"),
    col("ACTIVIDAD_HOSPITALARIA").alias("actividad_hospitalaria"),
    col("FECHA_MUESTRA").alias("fecha_muestra"),
    col("UBIGEO").alias("ubigeo"),
)

# Añadimos cod_diagnostico como número incremental
df_Diabetes_main = df_Diabetes_main.withColumn(
    "cod_diagnostico", row_number().over(windowSpec)
)

# Corregimos la fecha
df_Diabetes_main = df_Diabetes_main.withColumn(
    "fecha_muestra",
    to_date(col("fecha_muestra").cast("string"), "yyyyMMdd")
)


df_Diabetes_main.show(5)
df_Diabetes_main.printSchema()
```

```
+-------------+--------------------+--------------------+--------------------+-----------------
---+----------------------+-------------+------+---------------+
|cod_enfermedad|         id_paciente|     cod_institucion|           id_medico|servicio_hospitala
rio|actividad_hospitalaria|fecha_muestra|ubigeo|cod_diagnostico|
+-------------+--------------------+--------------------+--------------------+-----------------
---+----------------------+-------------+------+---------------+
|         E11.9|eJwzNDAwtDC0NDMxN...|      CAP I MANANTAY|eJwzNjA2MzE2NLY0N...|     MEDICINA GENE
RAL|  ATENCION  MEDICA ...|   2020-01-02|250107|              1|
|         E13.9|eJwzNDAwMjAxNTUxs...|        P.M. ALAMEDA|eJwzsjS1NDI2MjE2N...|     MEDICINA GENE
RAL|  ATENCION  MEDICA ...|   2020-01-02|250105|              2|
```

```
|         E11.9|eJwzNDCwMLOwMDKzt...|H.III DANIEL ALCI...|eJwztLQwNjM1NTY3N...|       ENDOCRINOLO
GIA|  ATENCION  MEDICA ...|   2020-01-02|230103|              3|
|         E11.9|eJwzNDCwNDU3Mja1M...|CAP II OSCAR FERN...|eJwztLS0MDE1sjQ3N...| MEDICINA FAMILIA
R...|  ATENCION  MEDICA ...|   2020-01-02|230101|              4|
|         E11.9|eJwzNDA0szQCQgMDM...|       P.M. SAPOSOA|eJwzNDS1MDO0MDEyM...|     MEDICINA GENE
RAL|  ATENCION  MEDICA ...|   2020-01-02|220401|              5|
```

Took 10 sec. Last updated by anonymous at October 12 2025, 8:00:10 PM.

FINISHED

```
%pyspark

from pyspark.sql import Window
from pyspark.sql.functions import col, row_number, monotonically_increasing_id

# Tomamos ambos conjuntos de procedimientos
df_proc1 = df_Diabetes.select(col("PROCEDIMIENTO_1").alias("des_procedimiento")).distinct()
df_proc2 = df_Diabetes.select(col("PROCEDIMIENTO_2").alias("des_procedimiento")).distinct()

# Unimos, eliminamos duplicados y asignamos un código incremental
windowSpec = Window.orderBy(monotonically_increasing_id())

df_Procedimiento = (
    df_proc1
    .unionByName(df_proc2)
    .distinct()
    .withColumn("cod_procedimiento", row_number().over(windowSpec))
    .select("cod_procedimiento", "des_procedimiento")
)

df_Procedimiento.show(10)
```

```
+-----------------+--------------------+
|cod_procedimiento|   des_procedimiento|
+-----------------+--------------------+
|                1|DOSAJE DE GLUCOSA...|
|                2|DOSAJE DE COLESTE...|
+-----------------+--------------------+
```

Took 12 sec. Last updated by anonymous at October 12 2025, 5:36:01 PM.

FINISHED

```
%pyspark

from pyspark.sql.functions import col
from pyspark.sql.functions import to_date, col

# --- 1️⃣ Unimos df_Diabetes con df_Diabetes_main para obtener el cod_diagnostico ---
df_joined = df_Diabetes_main.join(
    df_Diabetes,
    ["id_paciente", "id_medico", "fecha_muestra"],
    "inner"
)

# --- 2️⃣ Creamos df_res1 y df_res2 con los datos de resultados ---
df_res1 = df_joined.select(
    col("cod_diagnostico"),
    col("PROCEDIMIENTO_1").alias("des_procedimiento"),
    col("RESULTADO_1").alias("resultado"),
    col("UNIDADES_1").alias("unidades"),
    col("FEC_RESULTADO_1").alias("fecha_resultado")
)

df_res2 = df_joined.select(
    col("cod_diagnostico"),
    col("PROCEDIMIENTO_2").alias("des_procedimiento"),
    col("RESULTADO_2").alias("resultado"),
    col("UNIDADES_2").alias("unidades"),
    col("FEC_RESULTADO_2").alias("fecha_resultado")
)
```

```
# --- 3️⃣ Unimos ambos conjuntos ---
df_Resultado_Procedimiento_temp = df_res1.unionByName(df_res2).distinct()

# --- 4️⃣ Asociamos el código de procedimiento según el nombre ---
df_Resultado_Procedimiento = df_Resultado_Procedimiento_temp.join(
    df_Procedimiento,
    on=["des_procedimiento"],
    how="left"
).select(
    "cod_diagnostico",
    "cod_procedimiento",
    "resultado",
    "unidades",
    "fecha_resultado"
)

# Corregimos la fecha
df_Resultado_Procedimiento = df_Resultado_Procedimiento.withColumn(
    "fecha_resultado",
    to_date(col("fecha_resultado").cast("string"), "yyyyMMdd")
)

df_Resultado_Procedimiento.show(10)
```

```
+---------------+-----------------+---------+--------+---------------+
|cod_diagnostico|cod_procedimiento|resultado|unidades|fecha_resultado|
+---------------+-----------------+---------+--------+---------------+
|         335027|                1|     99.0|   mg/dL|       20230518|
|         155766|                1|   114.98|   mg/dL|       20220402|
|          60665|                1|    144.0|   mg/dL|       20210426|
|         289348|                1|    114.2|   mg/dL|       20230214|
|          29517|                1|     91.0|   mg/dL|       20201104|
|          64224|                1|    149.0|   mg/dL|       20210517|
|         111879|                1|    250.0|   mg/dL|       20211111|
|          49653|                1|    186.0|   mg/dL|       20210215|
|         127578|                1|    97.45|   mg/dL|       20211229|
|         188596|                1|     89.0|   mg/dL|       20220621|
+---------------+-----------------+---------+--------+---------------+
only showing top 10 rows
```

Took 59 sec. Last updated by anonymous at October 12 2025, 5:38:13 PM. (outdated)

FINISHED

```
%pyspark

df_Medico = df_Diabetes.select(
    col("ID_MEDICO").alias("id_medico"),
    col("EDAD_MEDICO").alias("edad_medico")
).distinct()

df_Medico.show(10)
```

```
+--------------------+-----------+
|           id_medico|edad_medico|
+--------------------+-----------+
|eJwzNDQ1tzQxNLYwN...|         55|
|eJwzMjQwNTY0tbAwN...|         31|
|eJwzMjAzMLMwNTU0s...|         33|
|eJwzNDO3sDAyMTAwM...|         52|
|eJwzNDMwNbQ0NDK2N...|         46|
|eJwztDQzt7Q0MTY1N...|         41|
|eJwzNLYwNjM1M7Mwt...|         62|
|eJwzMjSwNDQwNzS0N...|         32|
|eJwzNDY1MzYzNjc0M...|         66|
|eJwzNDUztzQwMLIwN...|         53|
+--------------------+-----------+
only showing top 10 rows
```

Took 5 sec. Last updated by anonymous at October 12 2025, 5:42:05 PM.

FINISHED

```
%pyspark

df_Paciente = df_Diabetes.select(
    col("ID_PACIENTE").alias("id_paciente"),
    col("EDAD_PACIENTE").alias("edad_paciente"),
    col("SEXO_PACIENTE").alias("sexo_paciente"),
).distinct()

df_Paciente.show(10)
```

```
+--------------------+-------------+-------------+
|         id_paciente|edad_paciente|sexo_paciente|
+--------------------+-------------+-------------+
|eJwzNDY1NTQ0NDczN...|           64|     FEMENINO|
|eJwzNLQ0NbcwtbAwN...|           51|     FEMENINO|
|eJwzNDE2MzG0NLawM...|           66|     FEMENINO|
|eJwztDQ3NrS0BCIjC...|           44|    MASCULINO|
|eJwzNDYxMTGyMDI2N...|           76|     FEMENINO|
|eJwzNDY2tLAwNTE0M...|           55|     FEMENINO|
|eJwzNLY0MLAwNzA3N...|           54|     FEMENINO|
|eJwzNDIwMTM1sDQws...|           48|    MASCULINO|
|eJwzNLMwMLE0NDEwM...|           47|    MASCULINO|
|eJwzMjAyNLYwMjE0N...|           38|    MASCULINO|
+--------------------+-------------+-------------+
only showing top 10 rows
```

Took 4 sec. Last updated by anonymous at October 12 2025, 5:56:15 PM.

FINISHED

```
%pyspark
from pyspark.sql.functions import to_date, col

# Corregimos la fecha
df_Resultado_Procedimiento = df_Resultado_Procedimiento.withColumn(
    "fecha_resultado",
    to_date(col("fecha_resultado").cast("string"), "yyyyMMdd")
)

df_Resultado_Procedimiento.show(10)
```

```
+---------------+-----------------+---------+--------+---------------+
|cod_diagnostico|cod_procedimiento|resultado|unidades|fecha_resultado|
+---------------+-----------------+---------+--------+---------------+
|         335027|                1|     99.0|   mg/dL|     2023-05-18|
|         155766|                1|   114.98|   mg/dL|     2022-04-02|
|          60665|                1|    144.0|   mg/dL|     2021-04-26|
|         289348|                1|    114.2|   mg/dL|     2023-02-14|
|          29517|                1|     91.0|   mg/dL|     2020-11-04|
|          64224|                1|    149.0|   mg/dL|     2021-05-17|
|         111879|                1|    250.0|   mg/dL|     2021-11-11|
|          49653|                1|    186.0|   mg/dL|     2021-02-15|
|         127578|                1|    97.45|   mg/dL|     2021-12-29|
|         188596|                1|     89.0|   mg/dL|     2022-06-21|
+---------------+-----------------+---------+--------+---------------+
only showing top 10 rows
```

Took 54 sec. Last updated by anonymous at October 12 2025, 5:47:02 PM.

FINISHED

```
%pyspark

df_Enf_Renal.show(10)
```

```
+-----------+------------+---------+----------+------+--------------------+--------------------+-
------------------+-------------+-------------+-----------+--------------------+--------+-------
-------------+-----------------+--------------------+----------------------+------------+------
---------+--------------------+-----------+----------+---------------+--------------------+------
-----+----------+
|FECHA_CORTE|DEPARTAMENTO|PROVINCIA|  DISTRITO|UBIGEO|                 RED|              IPRESS|
ID_PACIENTE|EDAD_PACIENTE|SEXO_PACIENTE|EDAD_MEDICO|           ID_MEDICO|COD_DIAG|         DIAGNO
STICO|AREA_HOSPITALARIA|SERVICIO_HOSPITALARIO|ACTIVIDAD_HOSPITALARIA|FECHA_MUESTRA|FEC_RESULTADO_
1|     PROCEDIMIENTO_1|RESULTADO_1|UNIDADES_1|FEC_RESULTADO_2|     PROCEDIMIENTO_2|RESULTADO_2|UN
IDADES_2|
+-----------+------------+---------+----------+------+--------------------+--------------------+-
------------------+-------------+-------------+-----------+--------------------+--------+-------
-------------+-----------------+--------------------+----------------------+------------+------
---------+--------------------+-----------+----------+---------------+--------------------+------
-----+----------+
|   20240531|      CALLAO|   CALLAO|BELLAVISTA|070102|RED ASISTENCIAL S...|H.N. ALBERTO SABO...|e
JwzNDU3NzI2NDGwN...|           80|    MASCULINO|         45|eJwzNLQwNTM3Mzc3N...|   N18.3|ENFERME
DAD RENAL     | CONSULTA EXTERNA|           NEFROLOGIA|  ATENCION  MEDICA ...|  2020-01-02|
```

Took 1 sec. Last updated by anonymous at October 12 2025, 8:01:01 PM.

FINISHED

```
%pyspark

from pyspark.sql.functions import col, row_number, monotonically_increasing_id
from pyspark.sql.window import Window
from pyspark.sql.functions import to_date, col

# Creamos una ventana sin partición ni orden específico
windowSpec = Window.orderBy(monotonically_increasing_id())

last_cod_diag = df_Diabetes_main.agg({"cod_diagnostico": "max"}).collect()[0][0]

df_Renal_main = df_Enf_Renal.select(
    col("COD_DIAG").alias("cod_enfermedad"),
    col("ID_PACIENTE").alias("id_paciente"),
    col("IPRESS").alias("cod_institucion"),
    col("ID_MEDICO").alias("id_medico"),
    col("SERVICIO_HOSPITALARIO").alias("servicio_hospitalario"),
    col("ACTIVIDAD_HOSPITALARIA").alias("actividad_hospitalaria"),
    col("FECHA_MUESTRA").alias("fecha_muestra"),
    col("UBIGEO").alias("ubigeo")
)

# Añadimos cod_diagnostico como número incremental
df_Renal_main = df_Renal_main.withColumn(
    "cod_diagnostico", row_number().over(windowSpec) + last_cod_diag
)

df_Renal_main.show(5)
df_Renal_main.printSchema()
```

```
+-------------+--------------------+--------------------+--------------------+-----------------
---+---------------------+-------------+------+---------------+
|cod_enfermedad|         id_paciente|     cod_institucion|           id_medico|servicio_hospitala
rio|actividad_hospitalaria|fecha_muestra|ubigeo|cod_diagnostico|
+-------------+--------------------+--------------------+--------------------+-----------------
---+---------------------+-------------+------+---------------+
|         N18.3|eJwzNDU3NzI2NDGwN...|H.N. ALBERTO SABO...|eJwzNLQwNTM3Mzc3N...|           NEFROLO
GIA|  ATENCION  MEDICA ...|   2020-01-02|070102|         509717|
|         N18.9|eJwzNDUwMTQ2NjIzN...|        H.II HUANUCO|eJwzNLUwNDQxMTYzN...|           NEFROLO
GIA|  ATENCION  MEDICA ...|   2020-01-02|100102|         509718|
|         N18.9|eJwzNDUwMDYwMjYyt...|        H.II HUANUCO|eJwzNLUwNDQxMTYzN...|           NEFROLO
GIA|  ATENCION  MEDICA ...|   2020-01-02|100102|         509719|
|         N18.9|eJwzNLG0MDYzMzEzM...|        H.II HUANUCO|eJwzNLUwNDQxMTYzN...|           NEFROLO
GIA|  ATENCION  MEDICA ...|   2020-01-02|100102|         509720|
```

```
|         N18.6|eJwzNDUwMDY2MjAxN...|        H.II HUANUCO|eJwzNLUwNDQxMTYzN...|           NEFROLO
GIA|  ATENCION  MEDICA ...|   2020-01-02|100102|         509721|
```

Took 7 sec. Last updated by anonymous at October 12 2025, 8:01:22 PM.

FINISHED

```
%pyspark

from pyspark.sql import Window
from pyspark.sql.functions import col, row_number, monotonically_increasing_id

# Tomamos ambos conjuntos de procedimientos
df_proc1 = df_Enf_Renal.select(col("PROCEDIMIENTO_1").alias("des_procedimiento")).distinct()
df_proc2 = df_Enf_Renal.select(col("PROCEDIMIENTO_2").alias("des_procedimiento")).distinct()

# Unimos, eliminamos duplicados y asignamos un código incremental
windowSpec = Window.orderBy(monotonically_increasing_id())

last_cod_pro = df_Procedimiento.agg({"cod_procedimiento": "max"}).collect()[0][0]

df_Procedimiento_Renal = (
    df_proc1
    .unionByName(df_proc2)
    .distinct()
    .withColumn("cod_procedimiento", row_number().over(windowSpec) + last_cod_pro)
    .select("cod_procedimiento", "des_procedimiento")
)

df_Procedimiento_Renal.show(10)
```

```
+-----------------+--------------------+
|cod_procedimiento|   des_procedimiento|
+-----------------+--------------------+
|                3|DOSAJE DE GLUCOSA...|
|                4|DOSAJE DE CREATIN...|
+-----------------+--------------------+
```

Took 14 sec. Last updated by anonymous at October 12 2025, 6:29:40 PM.

FINISHED

```
%pyspark

from pyspark.sql.functions import col
from pyspark.sql.functions import to_date, col

df_Enf_Renal = df_Enf_Renal.withColumn(
    "FECHA_MUESTRA",
    to_date(col("FECHA_MUESTRA").cast("string"), "yyyyMMdd")
)

# --- 1️⃣ Unimos df_Enf_Renal con df_Renal_main para obtener el cod_diagnostico ---
df_joined = df_Renal_main.join(
    df_Enf_Renal,
    ["id_paciente", "id_medico", "fecha_muestra"],
    "inner"
)

# --- 2️⃣ Creamos df_res1 y df_res2 con los datos de resultados ---
df_res1 = df_joined.select(
    col("cod_diagnostico"),
    col("PROCEDIMIENTO_1").alias("des_procedimiento"),
    col("RESULTADO_1").alias("resultado"),
    col("UNIDADES_1").alias("unidades"),
    col("FEC_RESULTADO_1").alias("fecha_resultado")
)

df_res2 = df_joined.select(
    col("cod_diagnostico"),
    col("PROCEDIMIENTO_2").alias("des_procedimiento"),
    col("RESULTADO_2").alias("resultado"),
    col("UNIDADES_2").alias("unidades"),
```

```
    col("FEC_RESULTADO_2").alias("fecha_resultado")
)

# --- 3️⃣ Unimos ambos conjuntos ---
df_Resultado_Procedimiento_temp = df_res1.unionByName(df_res2).distinct()

# --- 4️⃣ Asociamos el código de procedimiento según el nombre ---
df_Resultado_Procedimiento_Renal = df_Resultado_Procedimiento_temp.join(
    df_Procedimiento_Renal,
    on=["des_procedimiento"],
    how="left"
).select(
    "cod_diagnostico",
    "cod_procedimiento",
    "resultado",
    "unidades",
    "fecha_resultado"
)

# Corregimos la fecha
df_Resultado_Procedimiento_Renal = df_Resultado_Procedimiento_Renal.withColumn(
    "fecha_resultado",
    to_date(col("fecha_resultado").cast("string"), "yyyyMMdd")
)

df_Resultado_Procedimiento_Renal.show(10)
```

```
+---------------+-----------------+---------+--------+---------------+
|cod_diagnostico|cod_procedimiento|resultado|unidades|fecha_resultado|
+---------------+-----------------+---------+--------+---------------+
|         509725|                3|    137.0|   mg/dL|     2020-01-02|
|         509805|                4|     0.67|   mg/dL|     2020-01-06|
|         510101|                4|      4.6|   mg/dL|     2020-01-15|
|         510247|                4|     1.06|   mg/dL|     2020-01-20|
|         510579|                4|     0.96|   mg/dL|     2020-01-25|
|         510903|                3|    245.0|   mg/dL|     2020-01-30|
|         510940|                4|     0.57|   mg/dL|     2020-01-31|
|         511016|                3|    98.42|   mg/dL|     2020-02-01|
|         511255|                3|    313.0|   mg/dL|     2020-02-05|
|         511339|                3|    100.0|   mg/dL|     2020-02-06|
+---------------+-----------------+---------+--------+---------------+
only showing top 10 rows
```

Took 9 sec. Last updated by anonymous at October 12 2025, 6:58:48 PM.

FINISHED

```
('N\xc3\xbamero de m\xc3\xa9dicos antes:', 13473)
('N\xc3\xbamero de m\xc3\xa9dicos despu\xc3\xa9s:', 5835)
+--------------------+-----------+
|           id_medico|edad_medico|
+--------------------+-----------+
|eJwzMjAyMzQ3Nzc2M...|         39|
|eJwzMjAzsbAwsjQyM...|         35|
|eJwzMjC2NDK2NDa3M...|         39|
|eJwzMjCxMDI3NrUwN...|         37|
|eJwzMjCxMDM3NzGyB...|         37|
|eJwzMjCxtDQ1MTQ2M...|         38|
|eJwzMjQxNTMzNzY0M...|         33|
|eJwzMjQxNjEyN7UwM...|         34|
|eJwzMjQyNzWxNLcwN...|         31|
|eJwzNDC1MLMwNjM1N...|         43|
+--------------------+-----------+
only showing top 10 rows
```

Took 30 sec. Last updated by anonymous at October 12 2025, 7:02:47 PM. (outdated)

FINISHED

```
%pyspark

from pyspark.sql import functions as F

# Seleccionamos las columnas relevantes
df_pacientes_extra = (
    df_Enf_Renal
    .select(
        F.col("ID_PACIENTE").alias("id_paciente"),
        F.col("EDAD_PACIENTE").alias("edad_paciente"),
        F.col("SEXO_PACIENTE").alias("sexo_paciente")
    )
    .dropna(subset=["id_paciente"])
    .dropDuplicates(["id_paciente"])
)


df_Paciente_final = (
    df_Paciente
    .unionByName(df_pacientes_extra)
    .dropDuplicates(["id_paciente"])
)

df_Paciente_final.show(10)
```

```
+--------------------+-------------+-------------+
|         id_paciente|edad_paciente|sexo_paciente|
+--------------------+-------------+-------------+
|eJwzMTAwN7cwNzM2M...|           68|     FEMENINO|
|eJwzMjA0MzU1NDQ3M...|           40|    MASCULINO|
|eJwzMjA0MzY2NzKwM...|           39|    MASCULINO|
|eJwzMjA0NLAwtjAxN...|           40|     FEMENINO|
|eJwzMjA0NjI2MLUwN...|           39|    MASCULINO|
|eJwzMjA0sDAyMjA0N...|           40|     FEMENINO|
|eJwzMjA0sLQwsDAwt...|           40|     FEMENINO|
|eJwzMjA1MDExMzE2t...|           34|     FEMENINO|
|eJwzMjA1MDIzszAxM...|           38|    MASCULINO|
|eJwzMjA1MrY0NzUxs...|           62|    MASCULINO|
+--------------------+-------------+-------------+
only showing top 10 rows
```

Took 17 sec. Last updated by anonymous at October 12 2025, 7:07:19 PM.

FINISHED

```
%pyspark

from pyspark.sql.functions import col, row_number, monotonically_increasing_id
from pyspark.sql.window import Window
from pyspark.sql.functions import to_date, col

# Creamos una ventana sin partición ni orden específico
windowSpec = Window.orderBy(monotonically_increasing_id())

last_cod_diag = df_Renal_main.agg({"cod_diagnostico": "max"}).collect()[0][0]

df_Hiperlipidemia_main = df_Hiperlipidemia.select(
    col("COD_DIAG").alias("cod_enfermedad"),
    col("ID_PACIENTE").alias("id_paciente"),
    col("IPRESS").alias("cod_institucion"),
    col("ID_MEDICO").alias("id_medico"),
    col("SERVICIO_HOSPITALARIO").alias("servicio_hospitalario"),
    col("ACTIVIDAD_HOSPITALARIA").alias("actividad_hospitalaria"),
    col("FECHA_MUESTRA").alias("fecha_muestra"),
    col("UBIGEO").alias("ubigeo")
)
```

```
# Añadimos cod_diagnostico como número incremental
df_Hiperlipidemia_main = df_Hiperlipidemia_main.withColumn(
    "cod_diagnostico", row_number().over(windowSpec) + last_cod_diag
)

df_Hiperlipidemia_main.show(5)
df_Hiperlipidemia_main.printSchema()
```

```
+--------------+--------------------+--------------------+--------------------+-----------------
---+----------------------+-------------+------+---------------+
|cod_enfermedad|         id_paciente|     cod_institucion|           id_medico|servicio_hospitala
rio|actividad_hospitalaria|fecha_muestra|ubigeo|cod_diagnostico|
+--------------+--------------------+--------------------+--------------------+-----------------
---+----------------------+-------------+------+---------------+
|         E78.2|eJwzNDQ2MzU2tLQwM...|       POL. SAN LUIS|eJwzNDCwNDG1NDYxM...|          CARDIOLO
GIA|  ATENCION  MEDICA ...|   2020-01-02|150134|         625066|
|         E78.1|eJwzNLY0tDC1sDQ1N...|    H.I LA ESPERANZA|eJwzNLYwNzcxNzQxN...|     MEDICINA INTE
RNA|  ATENCION  MEDICA ...|   2020-01-02|130105|         625067|
|         E78.2|eJwzNjAwNLc0MjI1N...|H.II GUSTAVO LANA...|eJwzNDU3NTGxMDczM...|            PEDIAT
RIA|  ATENCION  MEDICA ...|   2020-01-03|150801|         625068|
|         E78.5|eJwzsjQ3NrUwNTMxM...|H.II RENE TOCHE G...|eJwzMjAwNTYwNDU3M...|     MEDICINA GENE
RAL|  ATENCION  MEDICA ...|   2020-01-03|110201|         625069|
|         E78.2|eJwzMjQxMTQzN7IwN...|CAP III METROPOLI...|eJwzMjAwNTQwNLEwM...|     MEDICINA GENE
RAL|  ATENCION  MEDICA ...|   2020-01-03|100101|         625070|
+--------------+--------------------+--------------------+--------------------+-----------------
```

Took 3 sec. Last updated by anonymous at October 12 2025, 8:01:54 PM.

FINISHED

```
%pyspark

from pyspark.sql import Window
from pyspark.sql.functions import col, row_number, monotonically_increasing_id

# Tomamos ambos conjuntos de procedimientos
df_proc1 = df_Hiperlipidemia.select(col("PROCEDIMIENTO_1").alias("des_procedimiento")).distinct()
df_proc2 = df_Hiperlipidemia.select(col("PROCEDIMIENTO_2").alias("des_procedimiento")).distinct()

# Unimos, eliminamos duplicados y asignamos un código incremental
windowSpec = Window.orderBy(monotonically_increasing_id())

last_cod_pro = df_Procedimiento_Renal.agg({"cod_procedimiento": "max"}).collect()[0][0]

df_Procedimiento_Hiperlipidemia = (
    df_proc1
    .unionByName(df_proc2)
    .distinct()
    .withColumn("cod_procedimiento", row_number().over(windowSpec) + last_cod_pro)
    .select("cod_procedimiento", "des_procedimiento")
)

df_Procedimiento_Hiperlipidemia.show(10)
```

```
+-----------------+--------------------+
|cod_procedimiento|   des_procedimiento|
+-----------------+--------------------+
|                5|       TRIGLICERIDOS|
|                6|DOSAJE DE CREATIN...|
+-----------------+--------------------+
```

Took 10 sec. Last updated by anonymous at October 12 2025, 7:12:25 PM.

FINISHED

```
%pyspark

from pyspark.sql.functions import col
from pyspark.sql.functions import to_date, col

# --- 1️⃣ Unimos df_Hiperlipidemia con df_Hiperlipidemia_main para obtener el cod_diagnostico ---
```

```
df_joined = df_Hiperlipidemia_main.join(
    ,
    ["id_paciente", "id_medico", "fecha_muestra"],
    "inner"
)

# --- 2️⃣ Creamos df_res1 y df_res2 con los datos de resultados ---
df_res1 = df_joined.select(
    col("cod_diagnostico"),
    col("PROCEDIMIENTO_1").alias("des_procedimiento"),
    col("RESULTADO_1").alias("resultado"),
    col("UNIDADES_1").alias("unidades"),
    col("FEC_RESULTADO_1").alias("fecha_resultado")
)

df_res2 = df_joined.select(
    col("cod_diagnostico"),
    col("PROCEDIMIENTO_2").alias("des_procedimiento"),
    col("RESULTADO_2").alias("resultado"),
    col("UNIDADES_2").alias("unidades"),
    col("FEC_RESULTADO_2").alias("fecha_resultado")
)

# --- 3️⃣ Unimos ambos conjuntos ---
df_Resultado_Procedimiento_temp = df_res1.unionByName(df_res2).distinct()

# --- 4️⃣ Asociamos el código de procedimiento según el nombre ---
df_Resultado_Procedimiento_Hiperlipidemia = df_Resultado_Procedimiento_temp.join(
    df_Procedimiento_Hiperlipidemia,
    on=["des_procedimiento"],
    how="left"
).select(
    "cod_diagnostico",
    "cod_procedimiento",
    "resultado",
    "unidades",
    "fecha_resultado"
)

# Corregimos la fecha
df_Resultado_Procedimiento_Hiperlipidemia = df_Resultado_Procedimiento_Hiperlipidemia.withColumn(
    "fecha_resultado",
    to_date(col("fecha_resultado").cast("string"), "yyyyMMdd")
)

df_Resultado_Procedimiento_Hiperlipidemia.show(10)
```

```
+---------------+-----------------+---------+--------+---------------+
|cod_diagnostico|cod_procedimiento|resultado|unidades|fecha_resultado|
+---------------+-----------------+---------+--------+---------------+
|         625066|                6|     0.97|   mg/dL|     2020-01-02|
|         625747|                6|     0.82|   mg/dL|     2020-01-22|
|         625943|                5|    144.6|   mg/dL|     2020-01-21|
|         626171|                5|    119.0|   mg/dL|     2020-01-24|
|         626267|                6|     0.56|   mg/dL|     2020-02-04|
|         626380|                5|    149.4|   mg/dL|     2020-01-28|
|         626680|                5|    107.0|   mg/dL|     2020-01-31|
|         626767|                6|     0.56|   mg/dL|     2020-02-03|
|         626775|                5|    100.0|   mg/dL|     2020-02-06|
|         627535|                6|      0.6|   mg/dL|     2020-02-17|
+---------------+-----------------+---------+--------+---------------+
only showing top 10 rows
```

Took 11 sec. Last updated by anonymous at October 12 2025, 7:18:00 PM. (outdated)

```
%pyspark

from pyspark.sql import functions as F
```

FINISHED

```
# Seleccionamos las columnas relevantes
df_medicos_extra = (
    df_Hiperlipidemia
    .select(
        F.col("ID_MEDICO").alias("id_medico"),
        F.col("EDAD_MEDICO").alias("edad_medico")
    )
    .dropna(subset=["id_medico"])  # eliminamos filas sin id_medico
    .dropDuplicates(["id_medico"])  # eliminamos duplicados dentro del propio df_Enf_Renal
)


df_Medico_final = (
    df_Medico_final
    .unionByName(df_medicos_extra)
    .dropDuplicates(["id_medico"])
)

print("Número de médicos antes:", df_Medico.count())
print("Número de médicos después:", df_Medico_final.count())
df_Medico_final.show(10)
```

```
('N\xc3\xbamero de m\xc3\xa9dicos antes:', 13473)
('N\xc3\xbamero de m\xc3\xa9dicos despu\xc3\xa9s:', 6102)
+--------------------+-----------+
|           id_medico|edad_medico|
+--------------------+-----------+
|eJwzMjAyM7Y0MjM3N...|         35|
|eJwzMjAyMzQ3Nzc2M...|         39|
|eJwzMjAzsbAwsjQyM...|         35|
|eJwzMjC2NDK2NDa3M...|         39|
|eJwzMjCxMDI3NrUwN...|         37|
|eJwzMjCxMDM3NzGyB...|         37|
|eJwzMjCxtDQ1MTQ2M...|         38|
|eJwzMjQxNTMzNzY0M...|         33|
|eJwzMjQxNjEyN7UwM...|         34|
|eJwzMjQyNzWxNLcwN...|         31|
+--------------------+-----------+
only showing top 10 rows
```

Took 40 sec. Last updated by anonymous at October 12 2025, 7:19:28 PM.

FINISHED

```
%pyspark

from pyspark.sql import functions as F

# Seleccionamos las columnas relevantes
df_pacientes_extra = (
    df_Hiperlipidemia
    .select(
        F.col("ID_PACIENTE").alias("id_paciente"),
        F.col("EDAD_PACIENTE").alias("edad_paciente"),
        F.col("SEXO_PACIENTE").alias("sexo_paciente")
    )
    .dropna(subset=["id_paciente"])
    .dropDuplicates(["id_paciente"])
)


df_Paciente_final = (
    df_Paciente_final
    .unionByName(df_pacientes_extra)
    .dropDuplicates(["id_paciente"])
)

df_Paciente_final.show(10)
```

```
+--------------------+-------------+-------------+
|         id_paciente|edad_paciente|sexo_paciente|
+--------------------+-------------+-------------+
```

```
|eJwzMTAwN7cwNzM2M...|           68|     FEMENINO|
|eJwzMjA0MzU1NDQ3M...|           40|    MASCULINO|
|eJwzMjA0MzY2NzKwM...|           39|    MASCULINO|
|eJwzMjA0NLAwtjAxN...|           40|     FEMENINO|
|eJwzMjA0NjI2MLUwN...|           39|    MASCULINO|
|eJwzMjA0sDAyMjA0N...|           40|     FEMENINO|
|eJwzMjA0sLQwsDAwt...|           40|     FEMENINO|
|eJwzMjA1MDExMzE2t...|           34|     FEMENINO|
|eJwzMjA1MDIzszAxM...|           38|    MASCULINO|
|eJwzMjA1MrY0NzUxs...|           62|    MASCULINO|
+--------------------+-------------+-------------+
only showing top 10 rows
```

Took 27 sec. Last updated by anonymous at October 12 2025, 7:20:31 PM.

FINISHED

```
%pyspark

df_Geodir = df_Geodir.select(
    col("Ubigeo").alias("ubigeo"),
    col("Distrito").alias("distrito"),
    col("Provincia").alias("provincia"),
    col("Departamento").alias("departamento"),
    col("Poblacion").alias("poblacion")
)

df_Geodir.show(10)
```

```
+------+-----------+-----------+------------+---------+
|ubigeo|   distrito|  provincia|departamento|poblacion|
+------+-----------+-----------+------------+---------+
| 10101|Chachapoyas|Chachapoyas|    Amazonas|    29171|
| 10102|   Asuncion|Chachapoyas|    Amazonas|      288|
| 10103|     Balsas|Chachapoyas|    Amazonas|     1644|
| 10104|      Cheto|Chachapoyas|    Amazonas|      591|
| 10105|  Chiliquin|Chachapoyas|    Amazonas|      687|
| 10106|Chuquibamba|Chachapoyas|    Amazonas|     2064|
| 10107|    Granada|Chachapoyas|    Amazonas|      379|
| 10108|    Huancas|Chachapoyas|    Amazonas|     1329|
| 10109|   La Jalca|Chachapoyas|    Amazonas|     5513|
| 10110| Leimebamba|Chachapoyas|    Amazonas|     4206|
+------+-----------+-----------+------------+---------+
only showing top 10 rows
```

Took 0 sec. Last updated by anonymous at October 12 2025, 7:24:23 PM.

FINISHED

```
%pyspark

from pyspark.sql import functions as F

df_Ipress = (
    df_Ipress
    .select(
        F.col("Código Único").alias("cod_institucion"),
        F.col("UBIGEO").alias("ubigeo"),
        F.col("Tipo").alias("tipo"),
        F.col("Nombre del establecimiento").alias("nombre"),
        F.col("Clasificación").alias("clasificacion"),
        F.col("Institución").alias("institucion"),
        F.col("Red").alias("red")
    )
    .dropna(subset=["cod_institucion"])
    .dropDuplicates(["cod_institucion"])
)

df_Ipress.show(10)
```

```
+--------------+------+--------------------+--------------------+--------------------+---------
-------+--------------------+
|cod_institucion|ubigeo|                tipo|              nombre|       clasificacion|      inst
itucion|                 red|
+--------------+------+--------------------+--------------------+--------------------+---------
-------+--------------------+
|          10096|140308|ESTABLECIMIENTO D...|         EL PUEBLITO|PUESTOS DE SALUD ...|GOBIERNO R
EGIONAL|          LAMBAYEQUE|
|          10351|150116|ESTABLECIMIENTO D...|RENAN JAEL OTTA G...|CONSULTORIOS MEDI...|
PRIVADO|NO PERTENECE A NI...|
|           1090|190301|ESTABLECIMIENTO D...|            MEZAPATA|PUESTOS DE SALUD ...|GOBIERNO R
EGIONAL|            OXAPAMPA|
|          11078|140101|ESTABLECIMIENTO D...|SEGUNDO RAFAEL SA...|CONSULTORIOS MEDI...|
PRIVADO|NO PERTENECE A NI...|
|          11332|150132|ESTABLECIMIENTO D...|PACPAC ROMAN WILB...|CONSULTORIOS MEDI...|
PRIVADO|NO PERTENECE A NI...|
|           1159|190306|ESTABLECIMIENTO D...|    PUERTO AGUACHINI|PUESTOS DE SALUD ...|GOBIERNO R
```

Took 1 sec. Last updated by anonymous at October 12 2025, 7:29:37 PM.

```
%pyspark


df_Diabetes_main.show(3)
df_Renal_main.show(3)
df_Hiperlipidemia_main.show(3)
```

FINISHED

```
+--------------+--------------------+--------------------+--------------------+-----------------
---+----------------------+-------------+---------------+
|cod_enfermedad|         id_paciente|     cod_institucion|           id_medico|servicio_hospitala
rio|actividad_hospitalaria|fecha_muestra|cod_diagnostico|
+--------------+--------------------+--------------------+--------------------+-----------------
---+----------------------+-------------+---------------+
|         E11.9|eJwzNDAwtDC0NDMxN...|      CAP I MANANTAY|eJwzNjA2MzE2NLY0N...|     MEDICINA GENE
RAL|  ATENCION  MEDICA ...|   2020-01-02|              1|
|         E13.9|eJwzNDAwMjAxNTUxs...|        P.M. ALAMEDA|eJwzsjS1NDI2MjE2N...|     MEDICINA GENE
RAL|  ATENCION  MEDICA ...|   2020-01-02|              2|
|         E11.9|eJwzNDCwMLOwMDKzt...|H.III DANIEL ALCI...|eJwztLQwNjM1NTY3N...|       ENDOCRINOLO
GIA|  ATENCION  MEDICA ...|   2020-01-02|              3|
+--------------+--------------------+--------------------+--------------------+-----------------
---+----------------------+-------------+---------------+
only showing top 3 rows
```

Took 10 sec. Last updated by anonymous at October 12 2025, 7:31:57 PM.

```
%pyspark

df_diagnostico = (
    df_Diabetes_main
    .unionByName(df_Renal_main)
    .unionByName(df_Hiperlipidemia_main)
)
```

FINISHED

Took 0 sec. Last updated by anonymous at October 12 2025, 8:02:02 PM.

```
%pyspark
df_diagnostico.show(5)
df_Ipress.show(5)
```

FINISHED

```
+--------------+--------------------+--------------------+--------------------+-----------------
---+----------------------+-------------+------+---------------+
|cod_enfermedad|         id_paciente|     cod_institucion|           id_medico|servicio_hospitala
rio|actividad_hospitalaria|fecha_muestra|ubigeo|cod_diagnostico|
+--------------+--------------------+--------------------+--------------------+-----------------
---+----------------------+-------------+------+---------------+
|         E11.9|eJwzNDAwtDC0NDMxN...|      CAP I MANANTAY|eJwzNjA2MzE2NLY0N...|     MEDICINA GENE
RAL|  ATENCION  MEDICA ...|   2020-01-02|250107|              1|
|         E13.9|eJwzNDAwMjAxNTUxs...|        P.M. ALAMEDA|eJwzsjS1NDI2MjE2N...|     MEDICINA GENE
RAL|  ATENCION  MEDICA ...|   2020-01-02|250105|              2|
|         E11.9|eJwzNDCwMLOwMDKzt...|H.III DANIEL ALCI...|eJwztLQwNjM1NTY3N...|       ENDOCRINOLO
GIA|  ATENCION  MEDICA ...|   2020-01-02|230103|              3|
|         E11.9|eJwzNDCwNDU3Mja1M...|CAP II OSCAR FERN...|eJwztLS0MDE1sjQ3N...| MEDICINA FAMILIA
R...|  ATENCION  MEDICA ...|   2020-01-02|230101|              4|
|         E11.9|eJwzNDA0szQCQgMDM...|        P.M. SAPOSOA|eJwzNDS1MDO0MDEyM...|     MEDICINA GENE
RAL|  ATENCION  MEDICA ...|   2020-01-02|220401|              5|
+--------------+--------------------+--------------------+--------------------+-----------------
```

Took 10 sec. Last updated by anonymous at October 12 2025, 8:02:17 PM.

ERROR

```
%pyspark

from pyspark.sql import functions as F
from pyspark.sql.window import Window
from pyspark.sql import SparkSession

spark = SparkSession.builder.getOrCreate()

# 1️⃣ Normalizamos nombres para que coincidan mejor
def normalizar(df, col):
    return df.withColumn(
        col,
        F.lower(F.regexp_replace(F.col(col), "[^a-zA-Z0-9]+", ""))
    )

df_diag_norm = normalizar(df_diagnostico, "cod_institucion")
df_ipress_norm = normalizar(df_Ipress, "nombre")

# 2️⃣ Hacemos un broadcast del DataFrame pequeño
df_ipress_broadcast = F.broadcast(df_ipress_norm)

# 3️⃣ Hacemos join por ubigeo (reduce el universo de comparación)
df_join = (
    df_diag_norm.alias("d")
    .join(
        df_ipress_broadcast.alias("i"),
        on="ubigeo",  # mismo ubigeo → misma zona
        how="left"
    )
)

# 4️⃣ Calculamos similitud entre los nombres normalizados
df_join = df_join.withColumn(
    "distancia",
    F.levenshtein(F.col("d.cod_institucion"), F.col("i.nombre"))
)

# 5️⃣ Filtramos para quedarnos con los más parecidos dentro de cada fila
windowSpec = Window.partitionBy("d.id_paciente").orderBy("distancia")

df_best_match = (
    df_join
    .withColumn("rank", F.row_number().over(windowSpec))
    .filter(F.col("rank") == 1)
    .select(
        F.col("d.id_paciente"),
        F.col("i.cod_institucion").alias("cod_institucion"),
        F.col("d.id_medico"),
        F.col("d.cod_enfermedad"),
```

```
        F.col("d.servicio_hospitalario"),
        F.col("d.actividad_hospitalaria"),
        F.col("d.fecha_muestra"),
        F.col("d.cod_diagnostico"),
        F.col("d.ubigeo")
    )
)

df_best_match.show(10)
```

```
Traceback (most recent call last):
  File "/tmp/zeppelin_pyspark-2451224276107551257.py", line 367, in <module>
    raise Exception(traceback.format_exc())
Exception: Traceback (most recent call last):
  File "/tmp/zeppelin_pyspark-2451224276107551257.py", line 360, in <module>
    exec(code, _zcUserQueryNameSpace)
  File "<stdin>", line 42, in <module>
  File "/usr/hdp/current/spark2-client/python/pyspark/sql/dataframe.py", line 350, in show
    print(self._jdf.showString(n, 20, vertical))
  File "/usr/hdp/current/spark2-client/python/lib/py4j-0.10.6-src.zip/py4j/java_gateway.py", line
1158, in __call__
    answer = self.gateway_client.send_command(command)
  File "/usr/hdp/current/spark2-client/python/lib/py4j-0.10.6-src.zip/py4j/java_gateway.py", line
908, in send_command
    response = connection.send_command(command)
  File "/usr/hdp/current/spark2-client/python/lib/py4j-0.10.6-src.zip/py4j/java_gateway.py", line
1055, in send_command
```

Took 7 min 13 sec. Last updated by anonymous at October 12 2025, 8:13:00 PM.

FINISHED

```
%pyspark

from pyspark.sql import functions as F, Window

# 1️⃣ Normalizamos nombres (igual que antes)
def normalizar(df, col):
    return df.withColumn(
        col + "_norm",
        F.lower(F.regexp_replace(F.col(col), "[^a-zA-Z0-9]+", ""))
    )

df_diag_norm = normalizar(df_diagnostico, "cod_institucion")
df_ipress_norm = normalizar(df_Ipress, "nombre")

# 2️⃣ Broadcast del DataFrame pequeño
df_ipress_broadcast = F.broadcast(df_ipress_norm)

# 3️⃣ Join por ubigeo + primeras letras (reduce drásticamente comparaciones)
df_join = (
    df_diag_norm.alias("d")
    .join(
        df_ipress_broadcast.alias("i"),
        (F.col("d.ubigeo") == F.col("i.ubigeo")) &
        (F.substring("d.cod_institucion_norm", 1, 3) == F.substring("i.nombre_norm", 1, 3)),
        how="left"
    )
)

# 4️⃣ Calculamos distancia solo entre pares filtrados
df_join = df_join.withColumn(
    "distancia",
    F.levenshtein(F.col("d.cod_institucion_norm"), F.col("i.nombre_norm"))
)

# 5️⃣ Nos quedamos con el mejor match por paciente
windowSpec = Window.partitionBy("d.id_paciente").orderBy("distancia")

df_best_match = (
```

```
    df_join
    .withColumn("rank", F.row_number().over(windowSpec))
    .filter((F.col("rank") == 1) & (F.col("distancia") < 10))  # 🔥 ignora coincidencias malas
    .select(
        F.col("d.id_paciente"),
        F.col("i.cod_institucion").alias("cod_institucion"),
        F.col("d.id_medico"),
        F.col("d.cod_enfermedad"),
        F.col("d.servicio_hospitalario"),
        F.col("d.actividad_hospitalaria"),
        F.col("d.fecha_muestra"),
        F.col("d.cod_diagnostico"),
        F.col("d.ubigeo"),
        F.col("distancia")
    )
)
```

Took 1 sec. Last updated by anonymous at October 12 2025, 8:13:06 PM.

```
%pyspark

df_best_match.show(10)
```

FINISHED

```
+--------------------+--------------+--------------------+-------------+--------------------+-
--------------------+-------------+---------------+------+---------+
|         id_paciente|cod_institucion|           id_medico|cod_enfermedad|servicio_hospitalario|a
ctividad_hospitalaria|fecha_muestra|cod_diagnostico|ubigeo|distancia|
+--------------------+--------------+--------------------+-------------+--------------------+-
--------------------+-------------+---------------+------+---------+
|eJwzMjA0MzY2NzKwM...|          8378|eJwzNjKxNLMwNjMzs...|         N18.3|     MEDICINA INTERNA|
ATENCION  MEDICA ...|   2022-12-10|         562316|150140|        8|
|eJwzMjA1MDExMzE2t...|          8378|eJwzMjQ0MDIwMDMyM...|         E11.9|     MEDICINA GENERAL|
ATENCION  MEDICA ...|   2020-03-06|          15504|150140|        8|
|eJwzMjAwNLE0sDAyt...|          8378|eJwzMjQ1NjUxtjAzM...|         N18.9|     MEDICINA GENERAL|
ATENCION  MEDICA ...|   2024-01-18|         610258|150140|        8|
|eJwzMjAyNLc0NTYxM...|          8836|eJwzNLY0MDM0NjIwt...|         E78.2|     MEDICINA GENERAL|
ATENCION  MEDICA ...|   2023-09-08|         723637|140101|        8|
|eJwzMjIwNTUyMDSzM...|         12190|eJwzMjA0MTI0MzY3t...|         E78.5|     MEDICINA GENERAL|
ATENCION  MEDICA ...|   2022-06-25|         674267|120101|        0|
|eJwzMjQ2NDSwMDa1M...|          8836|eJwzMjAxsjS3MDA3M...|         E78.9|     MEDICINA GENERAL|
PROGRAMA MI SALUD ...|   2024-02-24|         744294|140101|        8|
```

Took 19 sec. Last updated by anonymous at October 12 2025, 8:13:39 PM.

```
%pyspark
df_diagnostico = df_best_match.select(
        F.col("cod_diagnostico"),
        F.col("cod_enfermedad"),
        F.col("id_paciente"),
        F.col("cod_institucion"),
        F.col("servicio_hospitalario"),
        F.col("actividad_hospitalaria"),
        F.col("fecha_muestra")
)

df_paciente = df_Paciente_final
df_medico = df_Medico_final
df_resultado_procedimiento = (
    df_Resultado_Procedimiento
    .unionByName(df_Resultado_Procedimiento_Renal)
    .unionByName(df_Resultado_Procedimiento_Hiperlipidemia)
)
df_procedimiento = (
    df_Procedimiento
    .unionByName(df_Procedimiento_Renal)
    .unionByName(df_Procedimiento_Hiperlipidemia)
)
```

FINISHED

```
df_ubigeo = df_Geodir
df_ipress = df_Ipress
```

Took 0 sec. Last updated by anonymous at October 12 2025, 8:28:45 PM.

```
%pyspark

df_diagnostico.show(2)
df_paciente.show(2)
df_medico.show(2)
df_resultado_procedimiento.show(2)
df_procedimiento.show(2)
df_ubigeo.show(2)
df_ipress.show(2)
df_CIE.show(2)
```

FINISHED

```
+---------------+--------------+--------------------+--------------+--------------------+------
----------------+-------------+
|cod_diagnostico|cod_enfermedad|         id_paciente|cod_institucion|servicio_hospitalario|activi
dad_hospitalaria|fecha_muestra|
+---------------+--------------+--------------------+--------------+--------------------+------
----------------+-------------+
|         562316|         N18.3|eJwzMjA0MzY2NzKwM...|           8378|     MEDICINA INTERNA|  ATEN
CION  MEDICA ...|   2022-12-10|
|          15504|         E11.9|eJwzMjA1MDExMzE2t...|           8378|     MEDICINA GENERAL|  ATEN
CION  MEDICA ...|   2020-03-06|
+---------------+--------------+--------------------+--------------+--------------------+------
----------------+-------------+
only showing top 2 rows
+--------------------+-------------+-------------+
|         id_paciente|edad_paciente|sexo_paciente|
+--------------------+-------------+-------------+
|eJwzMTAwN7cwNzM2M...|           68|     FEMENINO|
|eJwzMjA0MzU1NDQ3M...|           40|    MASCULINO|
```

Took 3 min 17 sec. Last updated by anonymous at October 12 2025, 8:32:55 PM.

```
%pyspark

from pyspark.sql import functions as F

df_paciente = df_paciente.withColumn(
    "grupo_etario",
    F.when(F.col("edad_paciente") < 5, "Menor de 5 años")
     .when((F.col("edad_paciente") >= 5) & (F.col("edad_paciente") <= 11), "Niñez (5-11)")
     .when((F.col("edad_paciente") >= 12) & (F.col("edad_paciente") <= 17), "Adolescencia (12-17)")
     .when((F.col("edad_paciente") >= 18) & (F.col("edad_paciente") <= 29), "Joven adulto (18-29)")
     .when((F.col("edad_paciente") >= 30) & (F.col("edad_paciente") <= 59), "Adulto (30-59)")
     .otherwise("Adulto mayor (60+)")  # >=60
)

df_paciente.show(10, truncate=False)
```

FINISHED

```
+-----------------------------------+-------------+-------------+------------------+
|id_paciente                        |edad_paciente|sexo_paciente|grupo_etario      |
+-----------------------------------+-------------+-------------+------------------+
|eJwzMTAwN7cwNzM2MDczMTI0NjUyBAAmyAPa|68           |FEMENINO     |Adulto mayor (60+)|
|eJwzMjA0MzU1NDQ3MzA2NjE2sgQAHo4Daw==|40           |MASCULINO    |Adulto (30-59)    |
|eJwzMjA0MzY2NzKwMLCwsDA1MgQAHr4DcQ==|39           |MASCULINO    |Adulto (30-59)    |
|eJwzMjA0NLAwtjAxNze3MDU2MQEAHtkDeQ==|40           |FEMENINO     |Adulto (30-59)    |
|eJwzMjA0NjI2MLUwNTI0tTQxsQQAHmoDcA==|39           |MASCULINO    |Adulto (30-59)    |
|eJwzMjA0sDAyMjA0NjY1MbY0NgMAHiMDZQ==|40           |FEMENINO     |Adulto (30-59)    |
|eJwzMjA0sLQwsDAwt7S0NDYwMgUAHwYDeQ==|40           |FEMENINO     |Adulto (30-59)    |
|eJwzMjA1MDExMzE2tTAxMQACAB6jA2I=    |34           |FEMENINO     |Adulto (30-59)    |
|eJwzMjA1MDIzszAxMzQ0MrUwMQQAHsIDbg==|38           |MASCULINO    |Adulto (30-59)    |
```

```
|eJwzMjA1MrY0NzUxsrSwNDU3MQEAH2EDhg==|62           |MASCULINO    |Adulto mayor (60+)|
+------------------------------------+-------------+-------------+------------------+
only showing top 10 rows
```

Took 33 sec. Last updated by anonymous at October 12 2025, 8:35:05 PM.

FINISHED

```
%pyspark

# 📦 Ruta base
output_path = "/data/final/"

# 🧩 Exportamos cada DataFrame
df_diagnostico.coalesce(1).write.mode("overwrite").option("header", "true").csv(output_path + "diag
df_paciente.coalesce(1).write.mode("overwrite").option("header", "true").csv(output_path + "pacient
df_medico.coalesce(1).write.mode("overwrite").option("header", "true").csv(output_path + "medico")
df_resultado_procedimiento.coalesce(1).write.mode("overwrite").option("header", "true").csv(output_
df_procedimiento.coalesce(1).write.mode("overwrite").option("header", "true").csv(output_path + "pr
df_ubigeo.coalesce(1).write.mode("overwrite").option("header", "true").csv(output_path + "ubigeo")
df_ipress.coalesce(1).write.mode("overwrite").option("header", "true").csv(output_path + "ipress")
df_CIE.coalesce(1).write.mode("overwrite").option("header", "true").csv(output_path + "cie")
```

Took 3 min 10 sec. Last updated by anonymous at October 12 2025, 8:42:05 PM.

FINISHED

```
%pyspark

from pyspark.sql import functions as F

# 1️⃣ Agregamos columnas de año y mes
df_parquet = df_diagnostico.withColumn("anio", F.year("fecha_muestra")) \
                           .withColumn("mes", F.month("fecha_muestra"))

# 2️⃣ Guardamos en Parquet particionado por año y mes
output_path = "/data/final/diagnostico_parquet/"

df_parquet.write.mode("overwrite") \
        .partitionBy("anio", "mes") \
        .parquet(output_path)
```

Took 21 min 49 sec. Last updated by anonymous at October 12 2025, 9:12:37 PM.

FINISHED

```
%pyspark

df_fact = (
    df_diagnostico
    .join(df_paciente, "id_paciente", "left")
    .join(df_resultado_procedimiento, "cod_diagnostico", "left")
    .join(df_procedimiento, "cod_procedimiento", "left")
    .join(df_ipress, "cod_institucion", "left")
    .join(df_ubigeo, "ubigeo", "left")
    .join(df_CIE, "cod_enfermedad", "left")
)
```

Took 1 sec. Last updated by anonymous at October 12 2025, 9:49:04 PM.

ERROR

```
%pyspark

df_fact = df_fact.select(
    "cod_diagnostico",
    "id_paciente",
    "sexo_paciente",
    "grupo_etario",
    "cod_institucion",
    "nombre",
    "departamento",
    "red",
    "cod_enfermedad",
```

```
    "des_enfermedad",
    "des_procedimiento",
    "resultado",
    "unidades",
    "fecha_muestra"
)
```

```
Traceback (most recent call last):
  File "/tmp/zeppelin_pyspark-2451224276107551257.py", line 367, in <module>
    raise Exception(traceback.format_exc())
Exception: Traceback (most recent call last):
  File "/tmp/zeppelin_pyspark-2451224276107551257.py", line 360, in <module>
    exec(code, _zcUserQueryNameSpace)
  File "<stdin>", line 15, in <module>
  File "/usr/hdp/current/spark2-client/python/pyspark/sql/dataframe.py", line 1202, in select
    jdf = self._jdf.select(self._jcols(*cols))
  File "/usr/hdp/current/spark2-client/python/lib/py4j-0.10.6-src.zip/py4j/java_gateway.py", line
1160, in __call__
    answer, self.gateway_client, self.target_id, self.name)
  File "/usr/hdp/current/spark2-client/python/pyspark/sql/utils.py", line 69, in deco
    raise AnalysisException(s.split(': ', 1)[1], stackTrace)
AnalysisException: u"cannot resolve ``fecha_muestra`` given input columns: [sexo_paciente, nombr
e, d.cod_enfermedad, d.cod_diagnostico, departamento, des_enfermedad, d.id_paciente, grupo_etari
o, resultado, fecha_resultado, unidades, cod_institucion, red, des_procedimiento];;\n'Project [co
d_diagnostico#7983, id_paciente#7967, sexo_paciente#2512, grupo_etario#9855, cod_institucion#910
```

Took 0 sec. Last updated by anonymous at October 12 2025, 10:53:36 PM.

```
%pyspark

df_fact.show(10)
```

FINISHED

```
+---------------+--------------------+-------------+------------------+---------------+----------
----------+------------+--------------------+--------------+--------------------+---------------
----+---------+--------+---------------+
|cod_diagnostico|         id_paciente|sexo_paciente|      grupo_etario|cod_institucion|
nombre|departamento|                 red|cod_enfermedad|      des_enfermedad|   des_procedimiento
|resultado|unidades|fecha_resultado|
+---------------+--------------------+-------------+------------------+---------------+----------
----------+------------+--------------------+--------------+--------------------+---------------
----+---------+--------+---------------+
|            833|eJwzNDQxNDYwNDEyM...|     FEMENINO|Adulto mayor (60+)|          26009|  CAP III
CARABAYLLO|        Lima|NO PERTENECE A NI...|         E11.9|Diabetes mellitus...|DOSAJE DE GLUCOS
A...|     92.0|   mg/dL|     2020-01-20|
|           1591|eJwzNLQ0NLOwMDM1N...|    MASCULINO|Adulto mayor (60+)|           8438|POLICLINIC
O JUAN ...|        Lima|BARRANCO - CHORRI...|         E11.9|Diabetes mellitus...|DOSAJE DE GLUCOS
A...|    253.0|   mg/dL|     2020-01-15|
|          11317|eJwzNLS0sDQxNjKxM...|    MASCULINO|    Adulto (30-59)|           9298|POLICLINIC
O CENTR...|        Lima|NO PERTENECE A NI...|         E14.9|Diabetes mellitus...|DOSAJE DE GLUCOS
A...|    133.51|   mg/dL|     2020-02-21|
```

Took 3 min 12 sec. Last updated by anonymous at October 12 2025, 10:09:49 PM.

```
%pyspark

# RAW
df_diagnostico.write.mode("overwrite").parquet("/user/hive/warehouse/raw/diagnostico")

# CURATED
df_fact.write.mode("overwrite").partitionBy("departamento").parquet("/user/hive/warehouse/curated/f
```

FINISHED

Took 4 min 56 sec. Last updated by anonymous at October 12 2025, 9:57:43 PM.

```
%pyspark

df_fact_olap = (
    df_fact
    .withColumn("resultado_num", F.col("resultado").cast("double"))
    .groupBy("grupo_etario", "departamento", "des_enfermedad", "mes")
    .agg(
        F.countDistinct("id_paciente").alias("num_pacientes"),
        F.avg("resultado_num").alias("promedio_resultado")
    )
)

df_fact_olap.show(10, truncate=False)
```

FINISHED

```
+--------------------+------------+------------------------------------------------------------------
------------------------+-------------+------------------+
|grupo_etario        |departamento|des_enfermedad
|num_pacientes|promedio_resultado|
+--------------------+------------+------------------------------------------------------------------
------------------------+-------------+------------------+
|Adulto mayor (60+)  |La Libertad |Diabetes mellitus especificada, con complicaciones no especifi
cadas                    |1            |98.0              |
|Adulto mayor (60+)  |Puno        |Diabetes mellitus no insulinodependiente, con complicaciones c
irculatorias perif�ricas|1            |148.5             |
|Adulto (30-59)      |La Libertad |Diabetes mellitus asociada con desnutrici�n, con complicacion
es neurol�gicas           |1            |194.5             |
|Adolescencia (12-17)|Loreto      |Otros trastornos especificados de la secreci�n interna del p
�ncreas                   |1            |129.0             |
|Adulto mayor (60+)  |Junin       |Insuficiencia Renal Cr�nica Estadio 5
|2            |43.06             |
|Adulto mayor (60+)  |Lambayeque  |Diabetes mellitus especificada, con complicaciones neurol�gic
```

Took 3 min 36 sec. Last updated by anonymous at October 12 2025, 10:13:30 PM. (outdated)

```
%pyspark

spark.sql("SHOW DATABASES").show()
```

FINISHED

```
+------------+
|databaseName|
+------------+
|     default|
|    foodmart|
+------------+
```

Took 3 sec. Last updated by anonymous at October 12 2025, 10:18:19 PM.

```
%pyspark

from pyspark.sql import SparkSession

spark = (
    SparkSession.builder
    .appName("Proyecto_OLAP")
    .config("spark.sql.warehouse.dir", "/tmp/hive_warehouse")  # ✅ ruta donde sí puedes escribir
    .enableHiveSupport()
    .getOrCreate()
)
```

FINISHED

Took 0 sec. Last updated by anonymous at October 12 2025, 10:20:09 PM.

```
%pyspark

spark.sql("CREATE DATABASE IF NOT EXISTS raw_db")
spark.sql("CREATE DATABASE IF NOT EXISTS curated_db")
```

FINISHED

```
DataFrame[]
```

Took 2 sec. Last updated by anonymous at October 12 2025, 10:20:24 PM.

```
%pyspark


spark.sql("SHOW DATABASES").show()
```

FINISHED

```
+------------+
|databaseName|
+------------+
|  curated_db|
|     default|
|    foodmart|
|      raw_db|
+------------+
```

Took 1 sec. Last updated by anonymous at October 12 2025, 10:20:49 PM.

```
%pyspark

df_diagnostico.write.mode("overwrite").saveAsTable("raw_db.diagnostico")
df_paciente.write.mode("overwrite").saveAsTable("raw_db.paciente")
df_medico.write.mode("overwrite").saveAsTable("raw_db.medico")
df_resultado_procedimiento.write.mode("overwrite").saveAsTable("raw_db.resultado_procedimiento")
df_procedimiento.write.mode("overwrite").saveAsTable("raw_db.procedimiento")
df_ipress.write.mode("overwrite").saveAsTable("raw_db.ipress")
df_CIE.write.mode("overwrite").saveAsTable("raw_db.cie")
df_ubigeo.write.mode("overwrite").saveAsTable("raw_db.ubigeo")
```

FINISHED

Took 8 min 11 sec. Last updated by anonymous at October 12 2025, 10:29:24 PM.

```
%pyspark

df_fact = df_fact.withColumn("anio", F.year("fecha_resultado"))
df_fact = df_fact.withColumn("mes", F.month("fecha_resultado"))
df_fact = df_fact.withColumn("resultado", F.col("resultado").cast("double"))

df_fact.write.mode("overwrite").saveAsTable("curated_db.fact_diagnostico")

from pyspark.sql import functions as F

df_cubo = (
    df_fact.groupBy("grupo_etario", "departamento", "des_enfermedad", "mes", "anio")
    .agg(
        F.avg("resultado").alias("promedio_resultado"),
        F.count("cod_diagnostico").alias("cantidad_diagnosticos")
    )
)

df_cubo.write.mode("overwrite").saveAsTable("curated_db.cubo_olap")
```

FINISHED

Took 7 min 59 sec. Last updated by anonymous at October 12 2025, 11:01:55 PM.

```
%pyspark
df_fact.printSchema()
```

FINISHED

```
root
 |-- cod_diagnostico: integer (nullable = true)
 |-- id_paciente: string (nullable = true)
 |-- sexo_paciente: string (nullable = true)
 |-- grupo_etario: string (nullable = true)
 |-- cod_institucion: string (nullable = true)
 |-- nombre: string (nullable = true)
 |-- departamento: string (nullable = true)
 |-- red: string (nullable = true)
 |-- cod_enfermedad: string (nullable = true)
 |-- des_enfermedad: string (nullable = true)
 |-- des_procedimiento: string (nullable = true)
 |-- resultado: string (nullable = true)
 |-- unidades: string (nullable = true)
 |-- fecha_resultado: date (nullable = true)
```

Took 0 sec. Last updated by anonymous at October 12 2025, 10:52:37 PM.

```
%pyspark

df_cubo.toPandas().to_csv("/home/zeppelin/cubo_olap.csv", index=False)
```

ERROR

```
Traceback (most recent call last):
  File "/tmp/zeppelin_pyspark-2451224276107551257.py", line 367, in <module>
    raise Exception(traceback.format_exc())
Exception: Traceback (most recent call last):
  File "/tmp/zeppelin_pyspark-2451224276107551257.py", line 360, in <module>
    exec(code, _zcUserQueryNameSpace)
  File "<stdin>", line 1, in <module>
  File "/usr/hdp/current/spark2-client/python/pyspark/sql/dataframe.py", line 1933, in toPandas
    require_minimum_pandas_version()
  File "/usr/hdp/current/spark2-client/python/pyspark/sql/utils.py", line 126, in require_minimum_p
andas_version
    "it was not found." % minimum_pandas_version)
ImportError: Pandas >= 0.19.2 must be installed; however, it was not found.
```

Took 0 sec. Last updated by anonymous at October 12 2025, 11:03:28 PM.

```
%pyspark

df_cubo.coalesce(1).write.mode("overwrite").option("header", "true").csv("/tmp/cubo_olap_export")
```

FINISHED

Took 3 min 42 sec. Last updated by anonymous at October 12 2025, 11:10:40 PM.

```
%pyspark

df_diagnostico.count()
```

FINISHED

```
40090
```

Took 51 sec. Last updated by anonymous at October 12 2025, 11:28:06 PM.

```
%pyspark
```

READY