Apartado A

```
[maria_dev@sandbox-hdp ~]$ pig
25/11/17 07:56:57 INFO pig.ExecTypeProvider: Trying ExecType : LOCAL
25/11/17 07:56:57 INFO pig.ExecTypeProvider: Trying ExecType : MAPREDUCE
25/11/17 07:56:57 INFO pig.ExecTypeProvider: Trying ExecType : TEZ_LOCAL
25/11/17 07:56:57 INFO pig.ExecTypeProvider: Trying ExecType : TEZ
25/11/17 07:56:57 INFO pig.ExecTypeProvider: Picked TEZ as the ExecType
2025-11-17 07:56:57,614 [main] INFO  org.apache.pig.Main - Apache Pig version 0.16.0.2.6.5.0-292 (rUnversioned directory) compiled May 11 2018, 07:56:28
2025-11-17 07:56:57,615 [main] INFO  org.apache.pig.Main - Logging error messages to: /home/maria_dev/pig_1763366217612.log
2025-11-17 07:56:57,645 [main] INFO  org.apache.pig.impl.util.Utils - Default bootup file /home/maria_dev/.pigbootup not found
2025-11-17 07:56:58,111 [main] INFO  org.apache.pig.backend.hadoop.executionengine.HExecutionEngine - Connecting to hadoop file system at: hdfs://sandbo
x-hdp.hortonworks.com:8020
2025-11-17 07:56:58,621 [main] INFO  org.apache.pig.PigServer - Pig Script ID for the session: PIG-default-b71962b2-2b2f-4b37-888e-901e08efdd91
2025-11-17 07:56:59,146 [main] INFO  org.apache.hadoop.yarn.client.api.impl.TimelineClientImpl - Timeline service address: http://sandbox-hdp.hortonwork
s.com:8188/ws/v1/timeline/
2025-11-17 07:56:59,240 [main] INFO  org.apache.pig.backend.hadoop.PigATSClient - Created ATS Hook
```

```
Details at logfile: /home/maria_dev/pig_1763366217612.log
grunt> usuarios = LOAD 'u.user' USING PigStorage('|') AS (user_id:int, age:int, gender:chararray, occupation:chararray, zip:chararray);
grunt>
```

1. Muestra el total de hombres y mujeres que hay en el archivo u.user.

```
grunt> grouped_by_gender = GROUP usuarios BY gender; cantidad_genero = FOREACH grouped_by_gender GENERATE group AS gender, COUNT(usuarios) AS total;
grunt> DUMP cantidad_genero;
```

```
Input(s):
Successfully read 943 records (22628 bytes) from: "hdfs://sandbox-hdp.hortonworks.com:8020/user/maria_dev/u.user"

Output(s):
Successfully stored 2 records (22 bytes) in: "hdfs://sandbox-hdp.hortonworks.com:8020/tmp/temp-1280575314/tmp1062316689"

2025-11-17 08:17:49,639 [main] INFO  org.apache.hadoop.mapreduce.lib.input.FileInputFormat - Total input paths to process : 1
2025-11-17 08:17:49,640 [main] INFO  org.apache.pig.backend.hadoop.executionengine.util.MapRedUtil - Total input paths to process : 1
(F,273)
(M,670)
```

2. Mediante instrucciones de PIG encontrar las 10 ocupaciones más frecuentes entre los usuarios.

```
grunt> grouped_by_occupation = GROUP usuarios BY occupation; occupation_10 = FOREACH grouped_by_occupation GENERATE group AS occupation, COUNT(usuarios) AS total;
grunt> ordered_occupation = ORDER occupation_10 BY total DESC;
grunt> top_10_ocupaciones = LIMIT ordered_occupation 10;
grunt> DUMP top_10_ocupaciones
```

```
Input(s):
Successfully read 943 records (22628 bytes) from: "hdfs://sandbox-hdp.hortonworks.com:8020/user/maria_dev/u.user"

Output(s):
Successfully stored 10 records (175 bytes) in: "hdfs://sandbox-hdp.hortonworks.com:8020/tmp/temp-1280575314/tmp-183076228"

2025-11-17 08:29:09,188 [main] INFO  org.apache.hadoop.mapreduce.lib.input.FileInputFormat - Total input paths to process : 1
2025-11-17 08:29:09,188 [main] INFO  org.apache.pig.backend.hadoop.executionengine.util.MapRedUtil - Total input paths to process : 1
(student,196)
(other,105)
(educator,95)
(administrator,79)
(engineer,67)
(programmer,66)
(librarian,51)
(writer,45)
(executive,32)
(scientist,31)
grunt>
```

3. Muestra la edad media por géneros.

```
grunt> grouped_by_gender = GROUP usuarios BY gender;
grunt> edad_media_genero = FOREACH grouped_by_gender GENERATE group AS gender, AVG(usuarios.age) as media_edad;
grunt> DUMP edad_media_genero
```

```
Input(s):
Successfully read 943 records (22628 bytes) from: "hdfs://sandbox-hdp.hortonworks.com:8020/user/maria_dev/u.user"

Output(s):
Successfully stored 2 records (34 bytes) in: "hdfs://sandbox-hdp.hortonworks.com:8020/tmp/temp-1280575314/tmp-1503947393"

2025-11-17 08:37:19,051 [main] INFO  org.apache.hadoop.mapreduce.lib.input.FileInputFormat - Total input paths to process : 1
2025-11-17 08:37:19,051 [main] INFO  org.apache.pig.backend.hadoop.executionengine.util.MapRedUtil - Total input paths to process : 1
(F,33.81318681318681)
(M,34.149253731343286)
```

4. Muestra la edad media por ocupaciones.

```
grunt> grouped_by_occupation = GROUP usuarios BY occupation;
grunt> edad_media_ocupacion = FOREACH grouped_by_occupation GENERATE group AS occupation, AVG(usuarios.age) as media_edad;
grunt> DUMP edad_media_ocupacion
```

```
Input(s):
Successfully read 943 records (22628 bytes) from: "hdfs://sandbox-hdp.hortonworks.com:8020/user/maria_dev/u.user"

Output(s):
Successfully stored 21 records (508 bytes) in: "hdfs://sandbox-hdp.hortonworks.com:8020/tmp/temp-1280575314/tmp-1295652924"

2025-11-17 08:39:44,789 [main] INFO  org.apache.hadoop.mapreduce.lib.input.FileInputFormat - Total input paths to process : 1
2025-11-17 08:39:44,789 [main] INFO  org.apache.pig.backend.hadoop.executionengine.util.MapRedUtil - Total input paths to process : 1
(none,26.555555555555557)
(other,34.523809523809526)
(artist,31.392857142857142)
(doctor,43.57142857142857)
(lawyer,36.75)
(writer,36.31111111111111)
(retired,63.07142857142857)
(student,22.081632653061224)
(educator,42.01052631578948)
(engineer,36.38805970149254)
(salesman,35.666666666666664)
(executive,38.71875)
(homemaker,32.57142857142857)
(librarian,40.0)
(marketing,37.61538461538461)
(scientist,35.54838709677419)
(healthcare,41.5625)
(programmer,33.121212121212125)
(technician,33.1481481481481 45)
(administrator,38.74683544303797)
(entertainment,29.22222222222222)
grunt>
```

5. Guarda el resultado de las cuatro consultas anteriores en un script de extensión ".pig". Ejecútalo. (recuerda, siempre en la carpeta /user/maría_dev)

6. Almacena la salida de las cuatro consultas anteriores en una carpeta de HDFS llamada pig_usuarios.

Apartado B

1. Carga y descripción del dataset

He hecho un LIMIT de las 10 primeras líneas para que no salgan todos los datos

```
grunt> ventas = LOAD 'retail_sales_dataset.csv' USING PigStorage(',') AS (trans_id:chararray, date:chararray, cust_id:chararray
, gender:chararray, age:int, category:chararray, quantity:int, price_unit:double, total:double);
grunt> DESCRIBE ventas;
ventas: {trans_id: chararray,date: chararray,cust_id: chararray,gender: chararray,age: int,category: chararray,quantity: int,pr
ice_unit: double,total: double}
grunt> lineas10 = LIMIT ventas 10;
grunt> DUMP lineas10
```

```
(Transaction ID,Date,Customer ID,Gender,,Product Category,,,)
(1,2023-11-24,CUST001,Male,34,Beauty,3,50.0,150.0)
(2,2023-02-27,CUST002,Female,26,Clothing,2,500.0,1000.0)
(3,2023-01-13,CUST003,Male,50,Electronics,1,30.0,30.0)
(4,2023-05-21,CUST004,Male,37,Clothing,1,500.0,500.0)
(5,2023-05-06,CUST005,Male,30,Beauty,2,50.0,100.0)
(6,2023-04-25,CUST006,Female,45,Beauty,1,30.0,30.0)
(7,2023-03-13,CUST007,Male,46,Clothing,2,25.0,50.0)
(8,2023-02-22,CUST008,Male,30,Electronics,4,25.0,100.0)
(9,2023-12-13,CUST009,Male,63,Electronics,2,300.0,600.0)
grunt>
```

```
grunt> agrupado = GROUP ventas ALL;
grunt> total = FOREACH agrupado GENERATE COUNT(ventas);
grunt> DUMP total
```

```
Input(s):
Successfully read 1001 records (51673 bytes) from: "hdfs://sandbox-hdp.hortonworks.com:8020/user/maria_dev/retail_sales_dataset
.csv"

Output(s):
Successfully stored 1 records (7 bytes) in: "hdfs://sandbox-hdp.hortonworks.com:8020/tmp/temp10355344498/tmp1766128918"

2025-11-18 09:18:17,564 [main] INFO  org.apache.hadoop.mapreduce.lib.input.FileInputFormat - Total input paths to process : 1
2025-11-18 09:18:17,564 [main] INFO  org.apache.pig.backend.hadoop.executionengine.util.MapRedUtil - Total input paths to proce
ss : 1
(1001)
```

La cantidad total es de 1001

2. Filtrado por rango de edad

creo mayores_top10 donde limito los clientes a 10 y así poder verlo.

```
grunt> clientes_mayores30 = FILTER ventas BY age > 30;
grunt> mayores_top10 = LIMIT clientes_mayores30 10;
grunt> DUMP mayores_top10;
```

```
otal input paths to process : 1
(1,2023-11-24,CUST001,Male,34,Beauty,3,50.0,150.0)
(3,2023-01-13,CUST003,Male,50,Electronics,1,30.0,30.0)
(4,2023-05-21,CUST004,Male,37,Clothing,1,500.0,500.0)
(6,2023-04-25,CUST006,Female,45,Beauty,1,30.0,30.0)
(7,2023-03-13,CUST007,Male,46,Clothing,2,25.0,50.0)
(9,2023-12-13,CUST009,Male,63,Electronics,2,300.0,600.0)
(10,2023-10-07,CUST010,Female,52,Clothing,4,50.0,200.0)
(12,2023-10-30,CUST012,Male,35,Beauty,3,25.0,75.0)
(14,2023-01-17,CUST014,Male,64,Clothing,4,30.0,120.0)
(15,2023-01-16,CUST015,Female,42,Electronics,4,500.0,2000.0)
grunt>
```

Primero cuento los clientes mayores de 30.

```
grunt> mayores_agrupados = GROUP clientes_mayores30 ALL;
grunt> contar_mayores = FOREACH mayores_agrupados GENERATE COUNT(clientes_mayores30);
grunt> DUMP contar_mayores;
```

```
Output(s):
Successfully stored 1 records (7 bytes) in: "hdfs://sandb
73/tmp-1507422037"

2025-11-19 10:27:13,400 [main] INFO  org.apache.hadoop.ma
ut paths to process : 1
2025-11-19 10:27:13,400 [main] INFO  org.apache.pig.backe
otal input paths to process : 1
(727)
```

La salida es 727 por lo que ya podemos calcular el porcentaje total de las transacciones de mayores de 30 años

727 (Mayores de 30) / 1001(Total de personas en el documento) * 100 = 72,627% es el porcentaje total.

Haciéndolo con comandos, seria así:

```
grunt> agrupado = GROUP ventas ALL;
grunt> total = FOREACH agrupado GENERATE COUNT(ventas) AS total_ventas;
grunt> clientes_mayores30 = FILTER ventas BY age > 30;
grunt> mayores_agrupados = GROUP clientes_mayores30 ALL;
grunt> contar_mayores = FOREACH mayores_agrupados GENERATE COUNT(clientes_mayores30) AS total_mayores;
grunt> valores_agrupados = CROSS contar_mayores, total;
grunt> calculo_total = FOREACH valores_agrupados GENERATE ((double)total_mayores/(double)total_ventas) * 100.0 AS porcentaje_total;
grunt> DUMP calculo_total;
```

```
Input(s):
Successfully read 1001 records (51673 by

Output(s):
Successfully stored 1 records (13 bytes)

2025-11-19 11:12:53,632 [main] INFO  org
2025-11-19 11:12:53,632 [main] INFO  org
(72.62737262737264)
```

3. Transformación de campos

```
grunt> genero_descuento = FOREACH ventas GENERATE trans_id, UPPER(gender), price_unit * quantity * 0.90;
2025-11-19 11:17:23,921 [main] WARN  org.apache.pig.newplan.BaseOperatorPlan - Encountered Warning IMPLICIT_CAST_TO_DOUBLE 1 time(s).
grunt> resultado_total = LIMIT genero_descuento 20;
2025-11-19 11:17:58,186 [main] WARN  org.apache.pig.newplan.BaseOperatorPlan - Encountered Warning IMPLICIT_CAST_TO_DOUBLE 1 time(s).
grunt> DUMP resultado_total;
```

Parece que cuenta como primera línea las categorías del archivo, asique para ver las 20 primeras hay que pedir que nos saque 21

```
(Transaction ID,GENDER,)
(1,MALE,135.0)
(2,FEMALE,900.0)
(3,MALE,27.0)
(4,MALE,450.0)
(5,MALE,90.0)
(6,FEMALE,27.0)
(7,MALE,45.0)
(8,MALE,90.0)
(9,MALE,540.0)
(10,FEMALE,180.0)
(11,MALE,90.0)
(12,MALE,67.5)
(13,MALE,1350.0)
(14,MALE,108.0)
(15,FEMALE,1800.0)
(16,MALE,1350.0)
(17,FEMALE,90.0)
(18,FEMALE,45.0)
(19,FEMALE,45.0)
```

```
(Transaction ID,GENDER,)
(1,MALE,135.0)
(2,FEMALE,900.0)
(3,MALE,27.0)
(4,MALE,450.0)
(5,MALE,90.0)
(6,FEMALE,27.0)
(7,MALE,45.0)
(8,MALE,90.0)
(9,MALE,540.0)
(10,FEMALE,180.0)
(11,MALE,90.0)
(12,MALE,67.5)
(13,MALE,1350.0)
(14,MALE,108.0)
(15,FEMALE,1800.0)
(16,MALE,1350.0)
(17,FEMALE,90.0)
(18,FEMALE,45.0)
(19,FEMALE,45.0)
(20,MALE,810.0)
```

4. Agrupación y agregación por categoría de producto

```
grunt> agrupados = GROUP ventas BY category;
grunt> calculo_categoria = FOREACH agrupados GENERATE group AS categoria, COUNT(ventas) AS num_transacciones
, SUM(ventas.total) AS ventas_total, AVG(ventas.age) AS promedio_edad;
grunt> result_ordenado = ORDER calculo_categoria BY ventas_total DESC;
grunt> DUMP result_ordenado;
```

```
2025-11-20 08:04:15,928 [main] INFO  org.apac
hs to process : 1
2025-11-20 08:04:15,932 [main] INFO  org.apac
nput paths to process : 1
(Electronics,342,156905.0,41.73684210526316)
(Clothing,351,155580.0,41.94871794871795)
(Beauty,307,143515.0,40.37133550488599)
```

5. Extracción de categorías distintas

```
grunt> product_category = FOREACH ventas GENERATE category;
grunt> categorias = DISTINCT product_category;
grunt> DUMP categorias;
```

```
nput paths to process : 1
(Beauty)
(Clothing)
(Electronics)
```

Hay 3 categorías de productos, belleza, ropa y electrónica.

6. Ordenación y obtención de top-transacciones

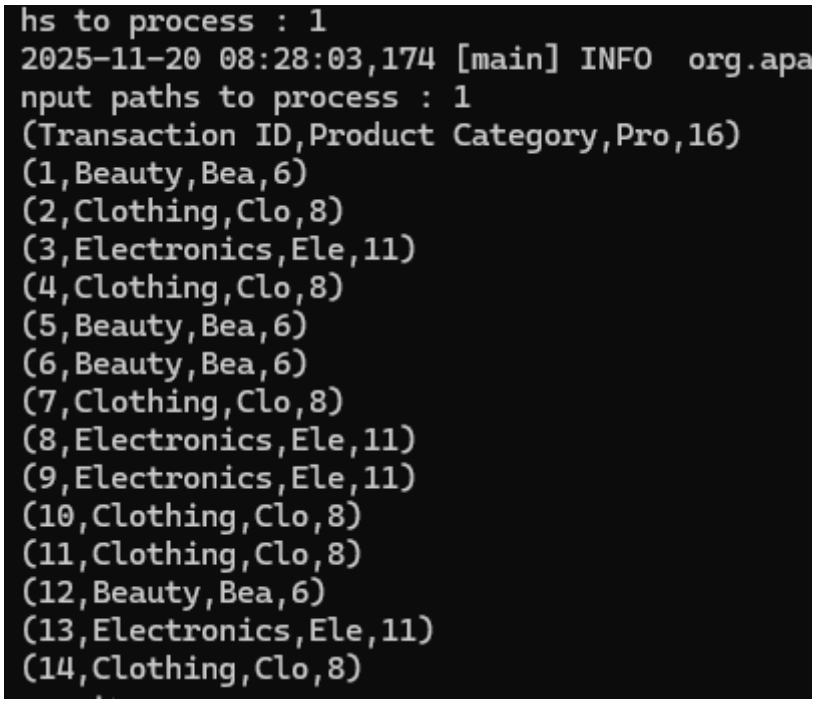
```
grunt> ordenado = ORDER ventas BY total DESC;
grunt> top5 = LIMIT ordenado 5;
grunt> resultado_final = FOREACH top5 GENERATE trans_id, cust_id, category, total;
grunt> DUMP resultado_final;
```

```
nput paths to process : 1
(664,CUST664,Clothing,2000.0)
(742,CUST742,Electronics,2000.0)
(155,CUST155,Electronics,2000.0)
(157,CUST157,Electronics,2000.0)
(572,CUST572,Clothing,2000.0)
```

7. Uso de funciones de cadena

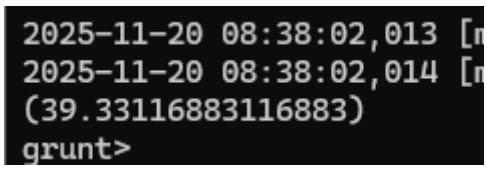
```
grunt> operaciones = FOREACH ventas GENERATE trans_id, category, SUBSTRING(category, 0, 3), SIZE(category);
grunt> result_registros = LIMIT operaciones 15;
grunt> DUMP result_registros;
```

Pasa lo mismo que en el ejercicio 3, para ver las 15 primeras, deberíamos sacar las 16 primeras ya que los datos ocupan una fila.



8. Filtrado por fecha y condiciones combinadas

```
grunt> ventas_filtradas = FILTER ventas BY date < '2023-07-01' AND total > (double)500;
grunt> agrupados = GROUP ventas_filtradas ALL;
grunt> promedio_edad = FOREACH agrupados GENERATE AVG(ventas_filtradas.age);
grunt> DUMP promedio_edad;
```



La edad promedio de los clientes de esa fecha es 39,33

9. Script completo + almacenamiento

```
-- Cargo los datos
ventas = LOAD 'retail_sales_dataset.csv' USING PigStorage(',') AS (trans_id:chararray, date:chararray,
cust_id:chararray, gender:chararray, age:int, category:chararray, quantity:int, price_unit:double, total:double);
DESCRIBE ventas;

-- Genero un descuento
datos_ventas = FOREACH sales GENERATE
    category,
    age,
    total,
    (totalt * 0.90) AS descuento;

-- Filtro las ventas por personas mayores de 30 que sus ventas sean mayor a 50
ventas_filtradas = FILTER datos_ventas BY age > 30 AND total > 50;

-- Agrupo por categoría de producto
agrupado_categoria = GROUP ventas_filtradas BY category;

-- Calculo número de ventas, suma total y edad promedio para cada categoría
final = FOREACH agrupado_categoria GENERATE
    group AS category,
    COUNT(ventas_filtradas) AS transacciones,
    SUM(ventas_filtradas.total) AS total_ingresos,
    AVG(ventas_filtradas.age) AS edad_promedio;

-- Ordeno para ver primero las categorías con más ingresos
result_ordenado = ORDER final BY total_ingresos DESC;

-- Guardamos el resultado en HDFS.
STORE result_ordenado INTO '/user/maría_dev/ejercicioB9' USING PigStorage(',');
```

Apartado B

1. Implementa un contador de palabras (cuantas veces aparece cada palabra en un texto)

```
text = LOAD '/user/maria_dev/ejercicio9/quijote.txt' USING PigStorage('\n') AS (line:chararray);
words = FOREACH text GENERATE FLATTEN(TOKENIZE(line)) AS word;
grouped_words = GROUP words BY word;
word_counts = FOREACH grouped_words GENERATE group AS word, COUNT(words) AS total;
ordered_counts = ORDER word_counts BY total DESC;

STORE ordered_counts INTO '/user/maria_dev/pig_quijote' USING PigStorage('\t');
```

```
[maria_dev@sandbox-hdp ~]$ wget https://www.gutenberg.org/cache/epub/2000/pg2000.txt -O quijote.txt
--2025-11-24 09:15:03--  https://www.gutenberg.org/cache/epub/2000/pg2000.txt
Resolving www.gutenberg.org (www.gutenberg.org)... 152.19.134.47, 2610:28:3090:3000:0:bad:cafe:47
Connecting to www.gutenberg.org (www.gutenberg.org)|152.19.134.47|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 2225845 (2.1M) [text/plain]
Saving to: ‘quijote.txt’

100%[======================================================>] 2,225,845    750KB/s   in 2.9s

2025-11-24 09:15:10 (750 KB/s) - ‘quijote.txt’ saved [2225845/2225845]

[maria_dev@sandbox-hdp ~]$
```

```
[maria_dev@sandbox-hdp ~]$ hdfs dfs -put quijote.txt /user/maria_dev/
[maria_dev@sandbox-hdp ~]$
```

```
Input(s):
Successfully read 38055 records (2226249 bytes) from: "/user/maria_dev/ejercicio9/quijote.txt"

Output(s):
Successfully stored 33248 records (370457 bytes) in: "/user/maria_dev/pig_quijote"

Counters:
Total records written : 33248
Total bytes written : 370457
Spillable Memory Manager spill count : 0
Total bags proactively spilled: 0
Total records proactively spilled: 0
```