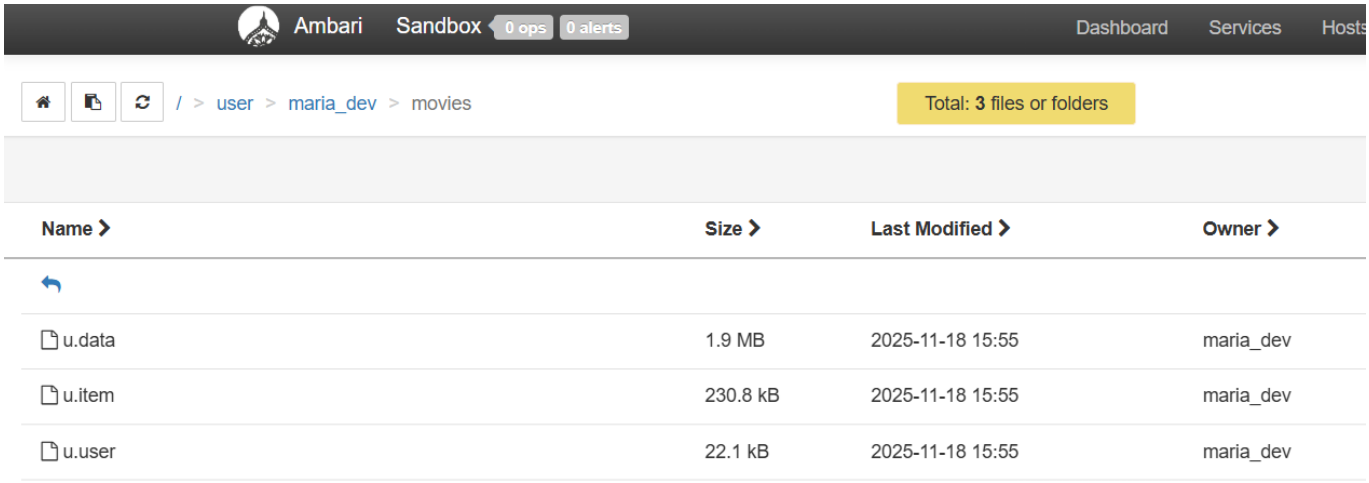
CONTENIDO
APARTADO A

Práctica con PIG

Mediante un script de PIG, encontrar las cinco películas (código, título y número de votos) más votadas (recuento de votos, no media).



1. Descripcion informal
    1. Cargamos **u.data** que es el archivo que contiene los votos y lo asignamos a la variable **data,** con la que vamos a trabajar todos los datos.
    2. Contamos cuántas veces aparece cada **movie_id** en los datos
    3. Cargar **u.item** que contiene los títulos de cada película
    4. Unimos con (**JOIN) movie_id** con los votos y con tirulos de cada película
    5. Ordenar por número de votos
    6. Sacar las 5 primeras

2. Implementa en PIG el script necesario para obtener la información deseada.

```
-- Cargar votos
data = LOAD '/user/maria_dev/movies/u.data' USING PigStorage('\t') AS (user_id:int, movie_id:int, rating:int, ts:int);

-- Contar votos por película
votos = GROUP data BY movie_id;
count_votos = FOREACH votos GENERATE group AS movie_id, COUNT(data) AS n_votos;

-- Cargar títulos
items = LOAD '/user/maria_dev/movies/u.item' USING PigStorage('|') AS (movie_id:int, title:chararray, release_date:chararray, url:chararray,
g1:int,g2:int,g3:int,g4:int,g5:int,g6:int,g7:int,g8:int,g9:int,g10:int,g11:int,g12:int,g13:int,g14:int,g15:int,g16:int,g17:int,g18:int,g19:int);

-- Hacer JOIN
joined = JOIN count_votos BY movie_id, items BY movie_id;

-- Seleccionar columnas
pelis = FOREACH joined GENERATE count_votos::movie_id AS movie_id, items::title AS title, count_votos::n_votos AS n_votos;

-- Ordenar y sacar top 5
ordenado = ORDER pelis BY n_votos DESC;
top5 = LIMIT ordenado 5;

DUMP top5;

STORE top5 INTO '/user/maria_dev/movies/resultados/top5_votadas' USING PigStorage(',');
```

3. Muestra la salida del mismo por pantalla y almacena también su resultado en un archivo en HDFS.

```
grunt> data = LOAD '/user/maria_dev/movies/u.data' USING PigStorage('\t') AS (user_id:int, movie_id:int, rating:int, ts:
int);
grunt> votos = GROUP data BY movie_id;
grunt> count_votos = FOREACH votos GENERATE group AS movie_id, COUNT(data) AS n_votos;
grunt> items = LOAD '/user/maria_dev/movies/u.item' USING PigStorage('|') AS (movie_id:int, title:chararray, release_dat
e:chararray, url:chararray, g1:int,g2:int,g3:int,g4:int,g5:int,g6:int,g7:int,g8:int,g9:int,g10:int,g11:int,g12:int,g13:i
nt,g14:int,g15:int,g16:int,g17:int,g18:int,g19:int);
grunt> joined = JOIN count_votos BY movie_id, items BY movie_id;
grunt> pelis = FOREACH joined GENERATE count_votos::movie_id AS movie_id, items::title AS title, count_votos::n_votos AS
 n_votos;
2025-11-18 15:16:09,765 [main] INFO  org.apache.pig.impl.util.SpillableMemoryManager - Selected heap (PS Old Gen) of siz
e 699400192 to monitor. collectionUsageThreshold = 489580128, usageThreshold = 489580128
grunt> ordenado = ORDER pelis BY n_votos DESC;
grunt> top5 = LIMIT ordenado 5;
grunt> DUMP top5;
```

```
Input(s):
Successfully read 100000 records (1979173 bytes) from: "/user/maria_dev/movies/u.data"
Successfully read 1682 records (236344 bytes) from: "/user/maria_dev/movies/u.item"

Output(s):
Successfully stored 5 records (146 bytes) in: "hdfs://sandbox-hdp.hortonworks.com:8020/tmp/temp9618564/tmp-313644938"

2025-11-18 15:17:42,479 [main] INFO  org.apache.hadoop.mapreduce.lib.input.FileInputFormat - Total input paths to proces
s : 1
2025-11-18 15:17:42,479 [main] INFO  org.apache.pig.backend.hadoop.executionengine.util.MapRedUtil - Total input paths t
o process : 1
(50,Star Wars (1977),583)
(258,Contact (1997),509)
(100,Fargo (1996),508)
(181,Return of the Jedi (1983),507)
(294,Liar Liar (1997),485)
grunt>
```

En la salida nos muestra movie Id, el titulo de la película, el año y nuero de votos totales

CONTENIDO
APARTADO B

Mediante un script de PIG, encontrar las diez películas mejor valoradas (código, título y media de puntuación) por los usuarios (ahora sí, media de todos los votos recibidos).

1. Describe informalmente los pasos que darás para llegar a la solución.
    1. Cargar votos
    2. Calcular media de rating por movie_id
    3. Cargar títulos
    4. Hacer JOIN
    5. Ordenar por media descendente
    6. Tomar las 10 mejores

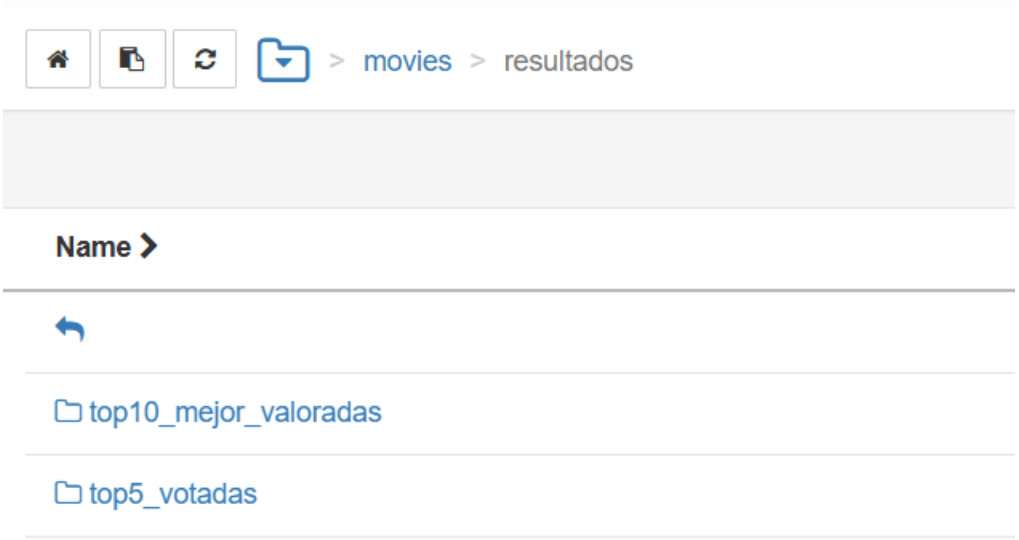
```
-- Cargar votos
data = LOAD '/user/maria_dev/movies/u.data' USING PigStorage('\t') AS (user_id:int, movie_id:int, rating:int, ts:int);

-- Media de rating
grp = GROUP data BY movie_id;
medias = FOREACH grp GENERATE group AS movie_id, AVG(data.rating) AS media;

-- Cargar títulos
items = LOAD '/user/maria_dev/movies/u.item' USING PigStorage('|') AS (movie_id:int, title:chararray, release_date:chararray, url:chararray,
g1:int,g2:int,g3:int,g4:int,g5:int,g6:int,g7:int,g8:int,g9:int,g10:int,g11:int,g12:int,g13:int,g14:int,g15:int,g16:int,g17:int,g18:int,g19:int);

-- Join
joined = JOIN medias BY movie_id, items BY movie_id;

result = FOREACH joined GENERATE medias::movie_id AS movie_id, items::title AS title, medias::media AS media;

orden = ORDER result BY media DESC;
top10 = LIMIT orden 10;

DUMP top10;
STORE top10 INTO '/user/maria_dev/movies/resultados/top10_mejor_valoradas' USING PigStorage(',');
```

2. Implementa en PIG el script necesario para obtener la información deseada.

```
grunt> data = LOAD '/user/maria_dev/movies/u.data' USING PigStorage('\t') AS
 (user_id:int, movie_id:int, rating:int, ts:int);
grunt> grp = GROUP data BY movie_id;
grunt> medias = FOREACH grp GENERATE group AS movie_id, AVG(data.rating) AS
media;
grunt> items = LOAD '/user/maria_dev/movies/u.item' USING PigStorage('|') AS
 (movie_id:int, title:chararray, release_date:chararray, url:chararray, g1:i
nt,g2:int,g3:int,g4:int,g5:int,g6:int,g7:int,g8:int,g9:int,g10:int,g11:int,g
12:int,g13:int,g14:int,g15:int,g16:int,g17:int,g18:int,g19:int);
grunt> joined = JOIN medias BY movie_id, items BY movie_id;
grunt> result = FOREACH joined GENERATE medias::movie_id AS movie_id, items:
:title AS title, medias::media AS media;
2025-11-18 19:08:55,664 [main] INFO  org.apache.pig.impl.util.SpillableMemor
yManager - Selected heap (PS Old Gen) of size 699400192 to monitor. collecti
onUsageThreshold = 489580128, usageThreshold = 489580128
grunt> orden = ORDER result BY media DESC;
grunt> top10 = LIMIT orden 10;
grunt> DUMP top10;
```

3. Muestra la salida del mismo por pantalla y almacena también su resultado en un archivo en HDFS.

```
Output(s):
Successfully stored 10 records (483 bytes) in: "hdfs://sandbox-hdp.hortonwor
ks.com:8020/tmp/temp1613408388/tmp317298689"

2025-11-18 19:09:55,087 [main] INFO  org.apache.hadoop.mapreduce.lib.input.F
ileInputFormat - Total input paths to process : 1
2025-11-18 19:09:55,087 [main] INFO  org.apache.pig.backend.hadoop.execution
engine.util.MapRedUtil - Total input paths to process : 1
(1653,Entertaining Angels: The Dorothy Day Story (1996),5.0)
(1293,Star Kid (1997),5.0)
(1467,Saint of Fort Washington, The (1993),5.0)
(814,Great Day in Harlem, A (1994),5.0)
(1500,Santa with Muscles (1996),5.0)
(1201,Marlene Dietrich: Shadow and Light (1996) ,5.0)
(1122,They Made Me a Criminal (1939),5.0)
(1189,Prefontaine (1997),5.0)
(1599,Someone Else's America (1995),5.0)
(1536,Aiqing wansui (1994),5.0)
grunt> STORE top10 INTO '/user/maria_dev/movies/resultados/top10_mejor_valor
adas' USING PigStorage(',');
```



CONTENIDO
APARTADO C

Práctica con PIG

Mediante un script de PIG, encontrar las cinco películas más antiguas con una valoración media por encima de 4 puntos.

1. Implementa en PIG el script necesario para obtener la información deseada.

```
-- Media rating
grp = GROUP data BY movie_id; medias = FOREACH grp GENERATE group AS movie_id, AVG(data.rating) AS media;

-- Cargar títulos con fechas
items = LOAD '/user/maria_dev/movies/u.item' USING PigStorage('|') AS (movie_id:int, title:chararray, release_date:chararray, url:chararray, ...);

-- Join
joined = JOIN medias BY movie_id, items BY movie_id;

-- Filtrar media > 4
buenas = FILTER joined BY medias::media > 4;

-- Ordenar por fecha ascendente
orden = ORDER buenas BY items::release_date ASC;

-- Quedarse solo las 5 primeras
top5_antiguas = LIMIT orden 5;

DUMP top5_antiguas;

STORE top5_antiguas INTO '/user/maria_dev/movies/resultados/top5_antiguas' USING PigStorage(',');
```

```
grunt> grp = GROUP data BY movie_id; medias = FOREACH grp GENERATE group AS
movie_id, AVG(data.rating) AS media;
grunt> items = LOAD '/user/maria_dev/movies/u.item' USING PigStorage('|') AS
 (movie_id:int, title:chararray, release_date:chararray, url:chararray, ...)
;
2025-11-18 19:14:08,643 [main] ERROR org.apache.pig.tools.grunt.Grunt - ERRO
R 1200: <line 12, column 141>  Syntax error, unexpected symbol at or near '.
.'
Details at logfile: /home/maria_dev/pig_1763492908637.log
grunt> joined = JOIN medias BY movie_id, items BY movie_id;
grunt> buenas = FILTER joined BY medias::media > 4;
2025-11-18 19:14:15,136 [main] WARN  org.apache.pig.newplan.BaseOperatorPlan
 - Encountered Warning IMPLICIT_CAST_TO_DOUBLE 1 time(s).
grunt> orden = ORDER buenas BY items::release_date ASC;
2025-11-18 19:14:18,867 [main] WARN  org.apache.pig.newplan.BaseOperatorPlan
 - Encountered Warning IMPLICIT_CAST_TO_DOUBLE 1 time(s).
grunt> top5_antiguas = LIMIT orden 5;
2025-11-18 19:14:22,778 [main] WARN  org.apache.pig.newplan.BaseOperatorPlan
 - Encountered Warning IMPLICIT_CAST_TO_DOUBLE 1 time(s).
grunt> DUMP top5_antiguas;
```

2. Muestra la salida del mismo por pantalla y almacena también su resultado en un archivo en HDFS.

```
Output(s):
Successfully stored 5 records (398 bytes) in: "hdfs://sandbox-hdp.hortonworks.com:8020/tmp/temp1613408388/tmp-1486663812"

2025-11-18 19:15:46,926 [main] INFO  org.apache.hadoop.mapreduce.lib.input.FileInputFormat - Total input paths to process : 1
2025-11-18 19:15:46,926 [main] INFO  org.apache.pig.backend.hadoop.executionengine.util.MapRedUtil - Total input paths to process : 1
(1191,4.333333333333333,1191,Letter From Death Row, A (1998),01-Feb-1998,,,0,0,0,0,0,0,1,0,1,0,0,0,0,0,0,0,0)
(604,4.012345679012346,604,It Happened One Night (1934),01-Jan-1934,,,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0)
(493,4.15,493,Thin Man, The (1934),01-Jan-1934,,,0,0,0,0,0,0,0,0,0,0,0,1,0,0,0,0)
(615,4.0508474576271185,615,39 Steps, The (1935),01-Jan-1935,,,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,0)
(1203,4.0476190476190474,1203,Top Hat (1935),01-Jan-1935,,,0,0,0,0,0,1,0,0,0,0,0,0,1,0,1,0,0,0)
grunt>
```

top10_mejor_valoradas	--
top5_antiguas	--
top5_votadas	--

CONTENIDO

APARTADO D

Práctica con PIG

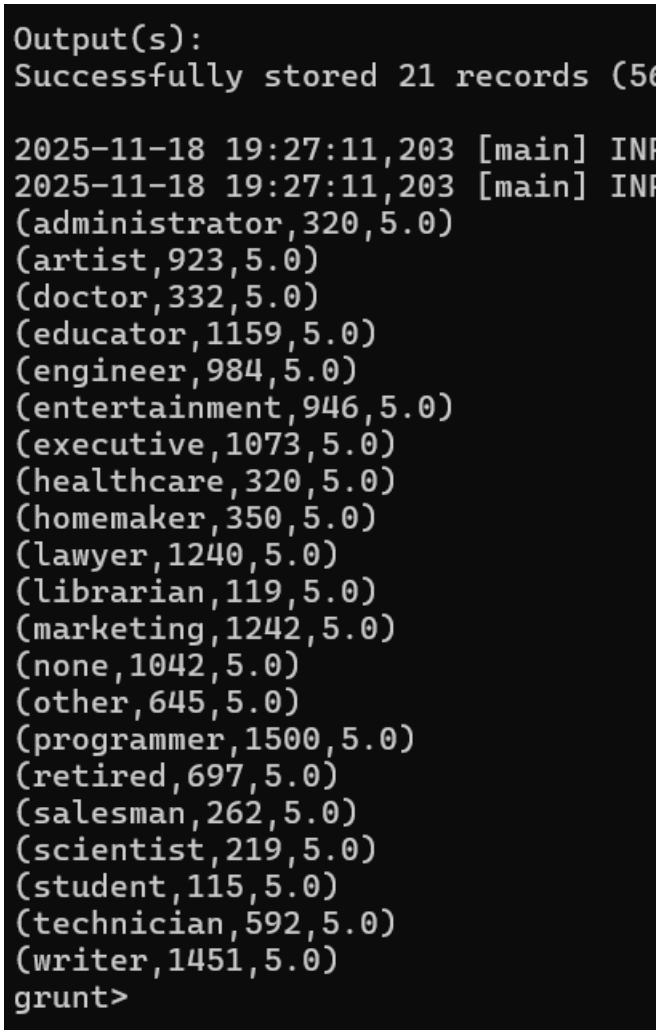
Mediante un script de PIG, encontrar la película mejor valorada por cada una de las ocupaciones (*student, writer, doctor,* etc.)

1. Implementa en PIG el script necesario para obtener la información deseada.

```
-- Cargar votos
data = LOAD '/user/maria_dev/movies/u.data' USING PigStorage('\t') AS (user_id:int, movie_id:int, rating:int, ts:int);

-- Cargar usuarios
users = LOAD '/user/maria_dev/movies/u.user' USING PigStorage('|') AS (user_id:int, age:int, gender:chararray, occupation:chararray, zip:chararray);

-- Join voto con usuario
join1 = JOIN data BY user_id, users BY user_id;

-- Calcular media por (ocupación, película)
grp = GROUP join1 BY (users::occupation, data::movie_id);

medias = FOREACH grp GENERATE group.occupation AS occupation, group.movie_id AS movie_id, AVG(join1.data::rating) AS media;

-- Película con mejor media por ocupación
by_occ = GROUP medias BY occupation;

mejor_por_ocupa = FOREACH by_occ {orden = ORDER medias BY media DESC;top1 = LIMIT orden 1;GENERATE FLATTEN(top1);}

DUMP mejor_por_ocupa;

STORE mejor_por_ocupa INTO '/user/maria_dev/movies/resultados/mejor_por_ocupacion' USING PigStorage(',');
```

```
grunt> data = LOAD '/user/maria_dev/movies/u.data' USING PigStorage('\t') AS
 (user_id:int, movie_id:int, rating:int, ts:int);
grunt> users = LOAD '/user/maria_dev/movies/u.user' USING PigStorage('|') AS
 (user_id:int, age:int, gender:chararray, occupation:chararray, zip:chararra
y);
grunt> join1 = JOIN data BY user_id, users BY user_id;
2025-11-18 19:23:03,581 [main] INFO  org.apache.pig.impl.util.SpillableMemor
yManager - Selected heap (PS Old Gen) of size 699400192 to monitor. collecti
onUsageThreshold = 489580128, usageThreshold = 489580128
grunt> grp = GROUP join1 BY (users::occupation, data::movie_id);
grunt> medias = FOREACH grp GENERATE group.occupation AS occupation, group.m
ovie_id AS movie_id, AVG(join1.data::rating) AS media;
grunt> by_occ = GROUP medias BY occupation;
grunt> mejor_por_ocupa = FOREACH by_occ {orden = ORDER medias BY media DESC;
top1 = LIMIT orden 1;GENERATE FLATTEN(top1);}
grunt> DUMP mejor_por_ocupa;
```

2. Muestra la salida del mismo por pantalla y almacena también su resultado en un archivo en HDFS.



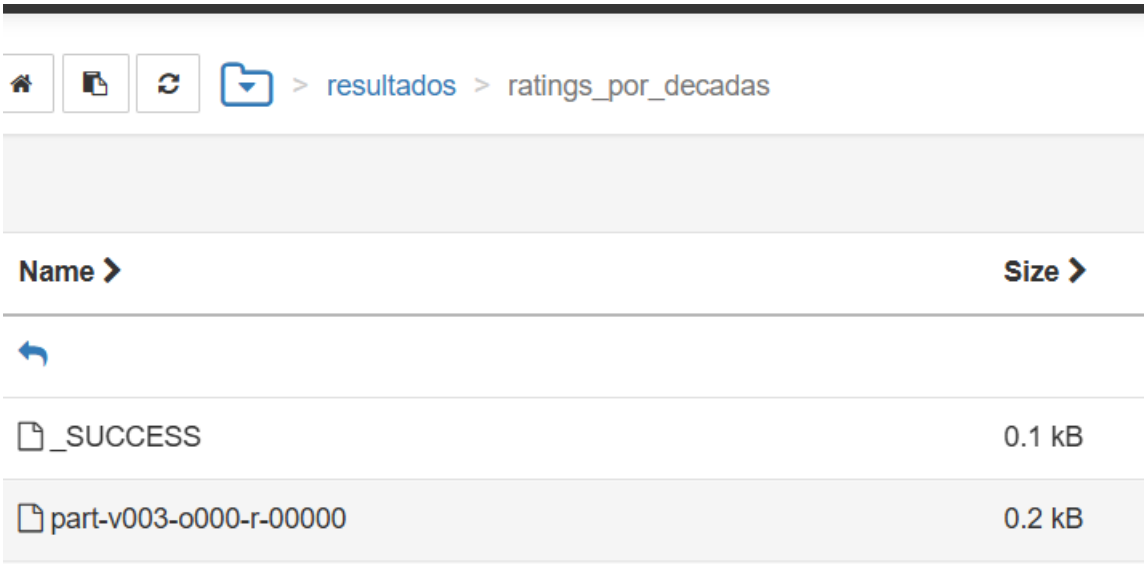
CONTENIDO
APARTADO E

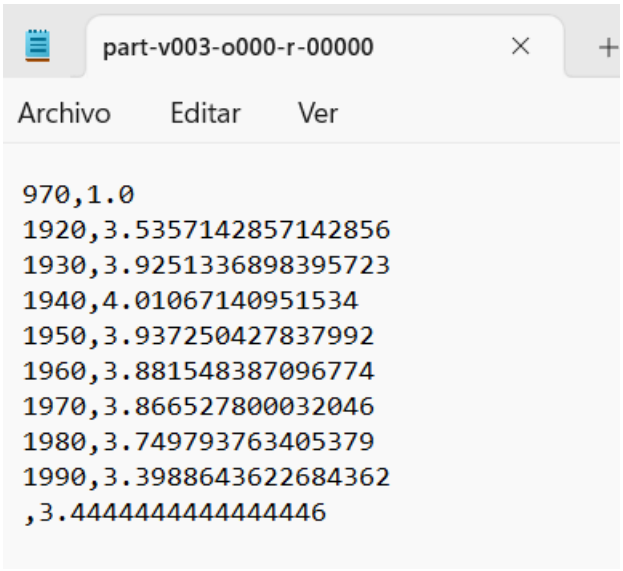
Mediante un script de PIG, encontrar el promedio de valoraciones por décadas, guardarlo en HDFS como un archivo `csv`. Posteriormente lo descargaremos a nuestro ordenador y con EXCEL hacer un gráfico de barras con los datos del ficchero

1. Implementa en PIG el script necesario para obtener la información deseada.
    - Analizar el archivo `u.item` (información de películas) y extraer la fecha de estreno utilizando el operador SUBSTRING.
    - Agrupar las películas por década (por ejemplo: 1970, 1980, 1990, etc.).
    - Calcular el promedio de `rating` por década.
    - Guardar los resultados en CSV para graficar posteriormente.

```
items = LOAD '/user/maria_dev/movies/u.item' USING PigStorage('|') AS (movie_id:int, title:chararray, release_date:chararray, video_release_date:chararray, imdb_url:chararray,
g1:int,g2:int,g3:int,g4:int,g5:int,g6:int,g7:int,g8:int,g9:int,g10:int,g11:int,g12:int,g13:int,g14:int,g15:int,g16:int,g17:int,g18:int,g19:int);


-- Extraer año
items_year = FOREACH items GENERATE movie_id, title, SUBSTRING(release_date, 7, 11) AS year;

-- Extraer década
items_decade = FOREACH items_year GENERATE movie_id, title, ( (int)year / 10 ) * 10 AS decade;

-- Cargar votos
data = LOAD '/user/maria_dev/movies/u.data' USING PigStorage('\t') AS (user_id:int, movie_id:int, rating:int, ts:int);

-- Join ratings + década
join1 = JOIN data BY movie_id, items_decade BY movie_id;

-- Agrupar por década
grp = GROUP join1 BY decade;

medias = FOREACH grp GENERATE group AS decade, AVG(join1.data::rating) AS media;

DUMP medias;

STORE medias INTO '/user/maria_dev/movies/resultados/ratings_por_decadas' USING PigStorage(',');
```


```
2025-11-18 20:02:10,459 [main] INFO  org.apac
ileInputFormat - Total input paths to process
2025-11-18 20:02:10,459 [main] INFO  org.apac
engine.util.MapRedUtil - Total input paths to
(970,1.0)
(1920,3.5357142857142856)
(1930,3.9251336898395723)
(1940,4.01067140951534)
(1950,3.937250427837992)
(1960,3.881548387096774)
(1970,3.866527800032046)
(1980,3.749793763405379)
(1990,3.3988643622684362)
```





2. Abrir el archive en Excel y generar un gráfico de barras con los datos.