

# Tarea evaluable

## CE\_5075 4.1

Big data aplicado



# Índice

<b>1 IMPORTAR DATOS</b>	<b>3</b>
<b>2 PREGUNTAS SOBRE LOS DATOS</b>	<b>6</b>
1. Llistat dels 10 programes de TV amb més d'una temporada que tenen millor valoració, ordenats per valoració en ordre decreixent.	6
2. Llistat dels 10 anys en què els seus programes de TV (segons l'any de llançament) han tengut més vots, ordenats per nombre de vots, en ordre decreixent.	7
3. Llistat dels 10 directors amb més pel·lícules, ordenats per nombre de pel·lícules en ordre decreixent.	8
4. Llistat dels 10 actors amb millor valoració mitjana de les seves pel·lícules, ordenats per valoració mitjana en ordre decreixent.	9

## APARTADO 1 - Películas i programas de Netflix

En la actividad de aprendizaje de la entrega 3 (y en algunos ejemplos de esta entrega), trabajamos con el conjunto de datos de las películas y programas de TV mejor valorados de Netflix, que se puede encontrar en Kaggle.

Recordemos que de los seis archivos que componen el conjunto de datos, solo nos interesan dos:

- **raw\_titles.csv**, que contiene la información de las películas (movies) y programas de TV (shows), incluyendo las series de la plataforma Netflix, con el número de votos y puntuación en IMDb.
- **raw\_credits.csv**, que contiene la información de los actores y directores de todas las películas y programas.

Nota: También puedes descargar los archivos desde el repositorio del curso, donde ya se han utilizado tabuladores como separadores de campos para evitar problemas con las importaciones: [raw\\_titles.csv](#) y [raw\\_credits.csv](#).

Como analistas de datos, nos han solicitado responder una serie de preguntas utilizando **Apache Impala**, ya sea desde **Hue** o desde el **Shell**.

## 1 IMPORTAR DATOS

Se han importado los datos desde el repositorio del curso ofrecido por el enunciado:

```
wget https://raw.githubusercontent.com/tnavarrete-iedib/bigdata-24-25/refs/heads/main/raw_titles.csv
```

```
wget https://raw.githubusercontent.com/tnavarrete-iedib/bigdata-24-25/refs/heads/main/raw_credits.csv
```

```
cloudera@quickstart.cloudera ~/tarea_4 (1.057s)
wget https://raw.githubusercontent.com/tnavarrete-iedib/bigdata-24-25/refs/heads/main/raw_titles.csv
--2025-01-22 08:09:43-- https://raw.githubusercontent.com/tnavarrete-iedib/bigdata-24-25/refs/heads/main/raw_titles.csv
Resolving raw.githubusercontent.com... 185.199.109.133, 185.199.110.133, 185.199.108.133, ...
Connecting to raw.githubusercontent.com|185.199.109.133|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 624984 (610K) [text/plain]
Saving to: `raw_titles.csv'

100%[=====>] 624,984
2025-01-22 08:09:44 (1.90 MB/s) - `raw_titles.csv' saved [624984/624984]

cloudera@quickstart.cloudera ~/tarea_4 (2.681s)
wget https://raw.githubusercontent.com/tnavarrete-iedib/bigdata-24-25/refs/heads/main/raw_credits.csv
--2025-01-22 08:09:51-- https://raw.githubusercontent.com/tnavarrete-iedib/bigdata-24-25/refs/heads/main/raw_credits.csv
Resolving raw.githubusercontent.com... 185.199.111.133, 185.199.109.133, 185.199.110.133, ...
Connecting to raw.githubusercontent.com|185.199.111.133|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 4238497 (4.0M) [text/plain]
Saving to: `raw_credits.csv'

100%[=====>] 4,238,497
2025-01-22 08:09:53 (2.21 MB/s) - `raw_credits.csv' saved [4238497/4238497]
```

Lo siguiente ha sido compartir los archivos en Hadoop:

```
hdfs dfs -mkdir netflix_movies
hdfs dfs -put raw_titles.csv netflix_movies/
hdfs dfs -put raw_credits.csv netflix_movies/
```

```
cloudera@quickstart.cloudera ~/tarea_4 (11.424s)
hdfs dfs -mkdir netflix_movies

cloudera@quickstart.cloudera ~/tarea_4 (13.771s)
hdfs dfs -put raw_titles.csv netflix_movies/

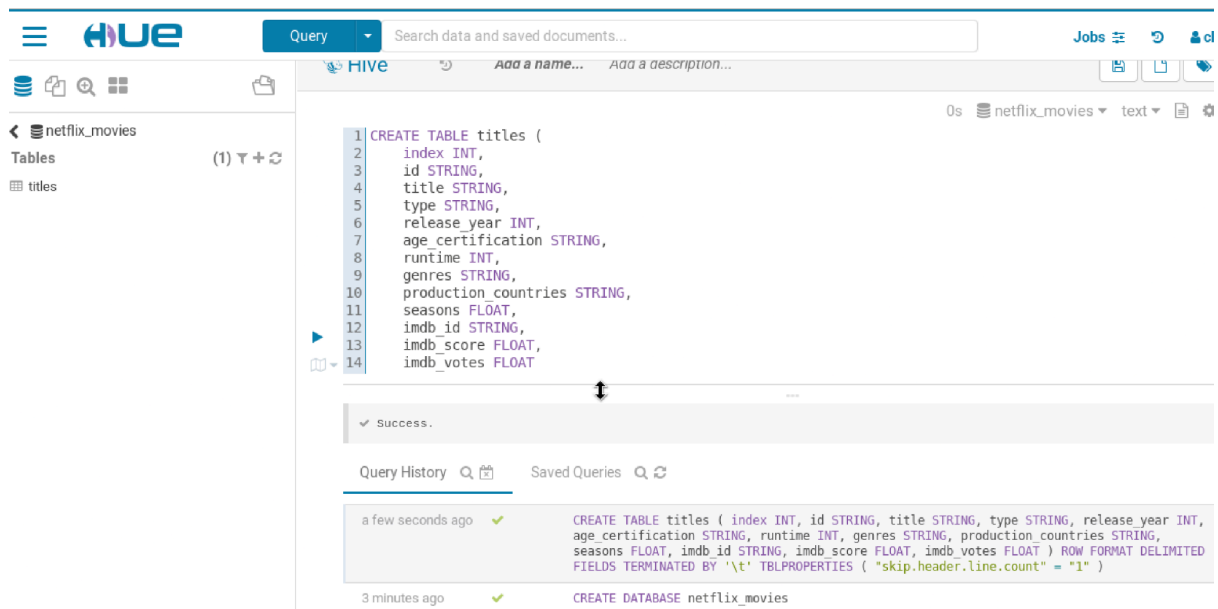
cloudera@quickstart.cloudera ~/tarea_4 (14.343s)
hdfs dfs -put raw_credits.csv netflix_movies/

cloudera@quickstart.cloudera ~/tarea_4 (11.714s)
hdfs dfs -ls netflix_movies/
Found 2 items
-rw-r--r-- 1 cloudera cloudera 4238497 2025-01-22 08:22 netflix_movies/raw_credits.csv
-rw-r--r-- 1 cloudera cloudera 624984 2025-01-22 08:21 netflix_movies/raw_titles.csv
```

El siguiente paso ha sido crear las tablas e importar los datos de los csv en Hive a través de los siguientes comandos:

```
CREATE DATABASE netflix_movies;

CREATE TABLE titles (
  index INT,
  id STRING,
  title STRING,
  type STRING,
  release_year INT,
  age_certification STRING,
  runtime INT,
  genres STRING,
  production_countries STRING,
  seasons FLOAT,
  imdb_id STRING,
  imdb_score FLOAT,
  imdb_votes FLOAT
)
ROW FORMAT DELIMITED
FIELDS TERMINATED BY '\t'
TBLPROPERTIES (
  "skip.header.line.count" = "1"
);
```



The screenshot shows the Hue web interface. On the left, a sidebar displays the database structure: 'netflix\_movies' with a table 'titles'. The main area shows the Hive SQL code for creating the database and table. Below the code editor, a green success message is visible. At the bottom, the 'Query History' section shows two queries: one executed 'a few seconds ago' (the table creation) and another '3 minutes ago' (the database creation).

```
1 CREATE DATABASE netflix_movies;
2
3 CREATE TABLE titles (
4   index INT,
5   id STRING,
6   title STRING,
7   type STRING,
8   release_year INT,
9   age_certification STRING,
10  runtime INT,
11  genres STRING,
12  production_countries STRING,
13  seasons FLOAT,
14  imdb_id STRING,
15  imdb_score FLOAT,
16  imdb_votes FLOAT
17 )
18 ROW FORMAT DELIMITED
19 FIELDS TERMINATED BY '\t'
20 TBLPROPERTIES (
21   "skip.header.line.count" = "1"
22 );
```

Success.

Query History

Time	Status	Query
a few seconds ago	✓	CREATE TABLE titles ( index INT, id STRING, title STRING, type STRING, release_year INT, age_certification STRING, runtime INT, genres STRING, production_countries STRING, seasons FLOAT, imdb_id STRING, imdb_score FLOAT, imdb_votes FLOAT ) ROW FORMAT DELIMITED FIELDS TERMINATED BY '\t' TBLPROPERTIES ( "skip.header.line.count" = "1" )
3 minutes ago	✓	CREATE DATABASE netflix_movies

```
CREATE TABLE credits (
  index INT,
  person_id INT,
  id STRING,
  name STRING,
  character STRING,
  role STRING
)
ROW FORMAT DELIMITED
FIELDS TERMINATED BY '\t'
TBLPROPERTIES (
  "skip.header.line.count" = "1"
);
```

The screenshot shows the HUE web interface. On the left sidebar, under 'netflix\_movies', there are two tables listed: 'credits' and 'titles'. The main area displays a Hive query that has been executed successfully, as indicated by the 'Success.' message. The query is: `CREATE TABLE credits ( index INT, person_id INT, id STRING, name STRING, character STRING, role STRING ) ROW FORMAT DELIMITED FIELDS TERMINATED BY '\t' TBLPROPERTIES ( \"skip.header.line.count\" = \"1\" );`. Below the query, there is a 'Query History' section showing the query was executed 'a minute ago'.

```
LOAD DATA LOCAL INPATH '/home/cloudera/tarea_4/raw_titles.csv'
INTO TABLE titles;
```

```
LOAD DATA LOCAL INPATH '/home/cloudera/tarea_4/raw_credits.csv'
INTO TABLE credits;
```

The screenshot shows the HUE web interface with two queries executed. The first query, 'LOAD DATA LOCAL INPATH '/home/cloudera/tarea\_4/raw\_credits.csv' INTO TABLE credits;', was executed 'a few seconds ago'. The second query, 'LOAD DATA LOCAL INPATH '/home/cloudera/tarea\_4/raw\_titles.csv' INTO TABLE titles;', was executed '2 minutes ago'. Both queries are marked as successful in the 'Query History' panel at the bottom.

Por último, se crean las tablas para tener los datos en Impala, se ha decidido usar el formato parquet:

```
CREATE TABLE titles_parquet
STORED AS PARQUET
AS SELECT * FROM titles;
```

```
CREATE TABLE credits_parquet
STORED AS PARQUET
AS SELECT * FROM credits;
```



## 2 PREGUNTAS SOBRE LOS DATOS

1. Listado de los 10 programas de TV con más de una temporada que tienen mejor valoración, ordenados por valoración en orden descendente.

```
SELECT *
FROM titles_parquet
WHERE seasons > 1 AND type = 'SHOW' AND imdb_score IS NOT
NULL
ORDER BY imdb_score DESC
LIMIT 10;
```

The screenshot shows the Impala web interface with the query results displayed. The query is the same as the one in the previous block. The results are shown in a table with 10 rows and 8 columns: index, id, title, type, release\_year, age\_certification, and runtime.

	index	id	title	type	release_year	age_certification	runtime
1	656	ts160526	Khawatir	SHOW	2005	TV-14	20
2	243	ts4	Breaking Bad	SHOW	2008	TV-MA	48
3	3827	ts90621	Kota Factory	SHOW	2019	TV-MA	42
4	259	ts3371	Avatar: The Last Airbender	SHOW	2005	TV-Y7	24
5	1099	ts121189	Raja, Rasoi Aur Anya Kahanilyaan	SHOW	2014		29
6	917	ts20682	Attack on Titan	SHOW	2013	TV-MA	24
7	1263	ts52922	Leah Remini: Scientology and the Aftermath	SHOW	2016	TV-14	46
8	717	ts32835	Hunter x Hunter	SHOW	2011	TV-14	23
9	47	ts20681	Seinfeld	SHOW	1989	TV-PG	24
10	367	ts24028	Still Game	SHOW	2002	TV-14	29

2. Listado de los 10 años en los que sus programas de TV (según el año de lanzamiento) han tenido más votos, ordenados por número de votos en orden descendente.

```
SELECT release_year, sum(imdb_votes) AS total_votes
FROM titles_parquet
WHERE type = 'SHOW' AND imdb_votes IS NOT NULL
GROUP BY release_year
ORDER BY total_votes DESC
LIMIT 10;
```

```
SELECT release_year, sum(imdb_votes) AS total_votes
FROM titles_parquet
WHERE type = 'SHOW' AND imdb_votes IS NOT NULL
GROUP BY release_year
ORDER BY total_votes DESC
LIMIT 10;
```

Query History

Saved Queries

Results (10)

	release_year	total_votes
1	2017	3213179
2	2019	3046821
3	2016	2745252
4	2018	2646768
5	2015	2559840
6	2020	2469787
7	2021	2268226
8	2013	2184696
9	2008	1853846
10	2014	1599878



3. Listado de los 10 directores con más películas, ordenados por número de películas en orden descendente.

```
SELECT name, count(name) AS films_count
FROM credits_parquet
WHERE `role` = 'DIRECTOR'
GROUP BY name
ORDER BY films_count DESC
LIMIT 10;
```

```
2 SELECT name, count(name) AS films_count
3 FROM credits_parquet
4 WHERE `role` = 'DIRECTOR'
5 GROUP BY name
6 ORDER BY films_count DESC
7 LIMIT 10;
```

Query History Saved Queries Results (10)

	name	films_count
1	Raúl Campos	21
2	Jan Suter	20
3	Jay Karas	16
4	Marcus Raboy	15
5	Ryan Polito	13
6	Jay Chapman	12
7	Cathy Garcia-Molina	12
8	Youssef Chahine	11
9	Justin G. Dyck	9
10	Troy Miller	8

4. Listado de los 10 actores con mejor valoración media de sus películas, ordenados por valoración media en orden descendente.

```
SELECT credits.name, AVG(credits.imdb_score) AS imdb_avg
FROM credits_parquet AS credits
JOIN titles_parquet AS titles ON credits.id = titles.id
WHERE credits.`role` = 'ACTOR' AND titles.imdb_score IS NOT
NULL
GROUP BY credits.name
ORDER BY imdb_avg DESC
LIMIT 10;
```

```
3 SELECT credits.name, AVG(credits.imdb_score) AS imdb_avg
4 FROM credits_parquet AS credits
5 JOIN titles_parquet AS titles ON credits.id = titles.id
6 WHERE credits.`role` = 'ACTOR' AND titles.imdb_score IS NOT NULL
7 GROUP BY credits.name
8 ORDER BY imdb_avg DESC
9 LIMIT 10;
```

Query History Saved Queries Results (10)

	name	imdb_avg
1	Anna Gunn	9.5
2	Betsy Brandt	9.5
3	Cricket Leigh	9.3000001907348633
4	Zach Tyler	9.3000001907348633
5	Jessie Flower	9.3000001907348633
6	Zhang Feng Yi	9.1999998092651367
7	Lee Hye-ri	9.1999998092651367
8	Lee Ji-ah	9.1999998092651367
9	He Kailang	9.1999998092651367
10	Kim Sung-kyun	9.1999998092651367