



Ecosistema Hadoop

Tendréis que ir respondiendo a las preguntas utilizando comentarios en caso de que os pregunten algo y/o capturas de pantalla.

ANDREI ALEXANDRU MIU

Índice

Instalación y configuración del cluster	3
Almacenamiento y tratamiento de datos	4
HDFS	4
MapReduce Job:	6
Pig Script	8
Sqoop	11
Flume	17
Procesamiento avanzado de datos	17
Apache Spark.....	18

Cluster Hadoop

Tendréis que ir respondiendo a las preguntas utilizando comentarios en caso de que os pregunten algo y/o capturas de pantalla.

IMPORTANTE: Asegúrate de documentar todos los cambios realizados y utiliza capturas de pantalla para justificar cada paso completado. Incluye cualquier código necesario para las configuraciones y ejecuciones.

Instalación y configuración del cluster

1. Describe brevemente los contenedores actuales de vuestro cluster.

NameNode: Administra el sistema de archivos distribuido de Hadoop (HDFS) y mantiene el namespace.

NodeManager: Ejecuta las tareas de las aplicaciones en cada nodo y monitorea el estado de los contenedores.

ResourceManager: Administra los recursos y coordina la ejecución de las aplicaciones en el clúster.

DataNode: Almacena los bloques de datos del HDFS.

2. Haz los cambios necesarios en el Dockerfile y Docker Compose para añadir 4 nuevos nodos Datanode y un nuevo nodo NodeManager. ¿Podría añadir otro nodo ResourceManager o Namenode? ¿Cómo mejorarías el Cluster?

<input type="checkbox"/>	Name	Container ID	Image	Port(s)	CPU (%)	Last started	Actions
<input type="checkbox"/>	hadoop	-	-	-	7.1%	55 seconds ago	<input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/>
<input type="checkbox"/>	namenode	7481b48152dd	hadoop-namenode	8020:8020 Show all ports (2)	1.25%	4 minutes ago	<input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/>
<input type="checkbox"/>	datanode3	c7680e3b4c7c	hadoop-datanode3	9866:9864 Show all ports (2)	0.49%	3 minutes ago	<input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/>
<input type="checkbox"/>	resourcemanager	429353f75033	hadoop-resourcemanager	8088:8088 Show all ports (2)	1.8%	3 minutes ago	<input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/>
<input type="checkbox"/>	datanode2	5d3bdf17e982	hadoop-datanode2	9865:9864 Show all ports (2)	0.78%	3 minutes ago	<input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/>
<input type="checkbox"/>	datanode4	3291c328359e	hadoop-datanode4	9867:9864 Show all ports (2)	0.28%	3 minutes ago	<input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/>
<input type="checkbox"/>	datanode	09411563e0c2	hadoop-datanode	9864:9864 Show all ports (2)	0.37%	3 minutes ago	<input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/>
<input type="checkbox"/>	datanode5	f4dde576feb5	hadoop-datanode5	9868:9864 Show all ports (2)	0.3%	3 minutes ago	<input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/>
<input type="checkbox"/>	nodemanager	730053aad396	hadoop-nodemanager	8043:8043 Show all ports (2)	1.04%	3 minutes ago	<input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/>
<input type="checkbox"/>	nodemanager2	4ec716182b79	hadoop-nodemanager2	8044:8043 Show all ports (2)	0.79%	55 seconds ago	<input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/>

Showing 13 items

Por poder, pueden añadirse tanto ResourceManager como el NameNode.

Para mejorarlo se podría meter un secondary namenode o un history server.

Almacenamiento y tratamiento de datos

HDFS

1. Crea un archivo txt cuyo contenido es tu nombre y la fecha actual y súbelo al cluster. Comprueba que está subido en HDFS a través de la página web ofrecida por el NameNode.

```
PS D:\Workspace_VSCode_IABD\BigDataAplicado\Practicas\Ev2\PracticasHadoop\EcosistemaHadoop> docker cp D:\Workspace_VSCode_IABD\BigDataAplicado\Practicas\Ev2\PracticasHadoop\EcosistemaHadoop\archivo.txt namenode:/home
Successfully copied 2.05kB to namenode:/home
PS D:\Workspace_VSCode_IABD\BigDataAplicado\Practicas\Ev2\PracticasHadoop\EcosistemaHadoop> |
```

Subimos el archivo al namenode

```
root@namenode:/# jps
340 Jps
12 NameNode
```

Como no esta activo el datanode, tendremos que arrancarlo de forma manual

```
root@namenode:/# hdfs namenode -format
```

Este comando lo usaremos solo si es la primera vez, ya que esto borra datos en hdfs

```
root@namenode:/# hdfs --daemon start namenode
WARNING: HADOOP_PREFIX has been replaced by HADOOP_HOME. Using value of HADOOP_PREFIX.
root@namenode:/# hdfs --daemon start datanode
WARNING: HADOOP_PREFIX has been replaced by HADOOP_HOME. Using value of HADOOP_PREFIX.
root@namenode:/# jps
560 DataNode
451 NameNode
12 NameNode
652 Jps
```

Despues iniciamos el namenode y el datanode en -d y con jps vemos que están arrancados.

```
root@namenode:/# hdfs dfs -mkdir /home
```

Creamos un directorio /home

```
root@namenode:/# hdfs dfs -ls /  
WARNING: HADOOP_PREFIX has been replaced by HADOOP_HOME. Using value of HADOOP_PREFIX.  
Found 1 items  
drwxr-xr-x   - root supergroup          0 2025-02-06 20:23 /home  
root@namenode:/#
```

Vemos que se ha creado correctamente

```
root@namenode:/# hdfs dfs -put /home/archivo.txt /home/  
WARNING: HADOOP_PREFIX has been replaced by HADOOP_HOME. Using value of HADOOP_PREFIX.  
root@namenode:/# hdfs dfs -ls /home/  
WARNING: HADOOP_PREFIX has been replaced by HADOOP_HOME. Using value of HADOOP_PREFIX.  
Found 1 items  
-rw-r--r--   2 root supergroup        17 2025-02-06 20:34 /home/archivo.txt  
root@namenode:/#
```

Subimos el archivo.txt a hdfs y comprobamos que se ha subido correctamente.

2. Haz todos los pasos necesarios para descargar tu archivo en el Desktop de tu HOST.

```
root@namenode:/home# hdfs dfs -get /home/archivo.txt C:/  
WARNING: HADOOP_PREFIX has been replaced by HADOOP_HOME. Using value of HADOOP_PREFIX.  
root@namenode:/home#
```

Lo descargamos desde HDFS

```
root@namenode:/# ls  
archivo.txt  boot  etc  
bin          dev   hadoop  
root@namenode:/#
```

Se nos habrá puesto aquí

```
PS D:\Workspace_VSCode_IABD\BigDataAplicado\Practicas\Ev2\PracticasHadoop\EcosistemaHadoop> docker cp namenode:/archivo.txt C:/Users/Andrei/Desktop/archivo.txt  
Successfully copied 2.05kB to C:\Users\Andrei\Desktop\archivo.txt  
PS D:\Workspace_VSCode_IABD\BigDataAplicado\Practicas\Ev2\PracticasHadoop\EcosistemaHadoop>
```

Y desde el namenode al Desktop

3. Muestra los permisos de los archivos subidos en HDFS.

```
root@namenode:/# hdfs dfs -ls /home/  
WARNING: HADOOP_PREFIX has been replaced by HADOOP_HOME. Using value of HADOOP_PREFIX.  
Found 1 items  
-rw-r--r--   2 root supergroup        17 2025-02-06 20:34 /home/archivo.txt  
root@namenode:/#
```

MapReduce Job:

1. Genera un archivo con 5000 líneas donde cada línea será un número aleatorio entre 0 y 200 usando python.

```
import random

def generate_random_file(filename, lines=5000, min_val=0, max_val=200):
    with open(filename, 'w') as file:
        for _ in range(lines):
            file.write(f"{random.randint(min_val, max_val)}\n")

# Llamada a la función para generar el archivo
generate_random_file("random_numbers.txt")
```

Si ejecutamos el .py, generará un archivo random_numbers.txt

2. Sube el archivo a HDFS en código.

```
PS D:\Workspace_VSCode_IABD\BigDataAplicado\Practicas\Ev2\Hadoop> docker cp D:\Workspace_VSCode_IABD\BigDataAplicado\Practicas\Ev2\PracticasHadoop\EcosistemaHadoop\archivo.txt namenode:/home
Successfully copied 2.05kB to namenode:/home
PS D:\Workspace_VSCode_IABD\BigDataAplicado\Practicas\Ev2\Hadoop> |
```

Lo subimos primero al namenode

```
root@namenode:/home# hdfs dfs -put /home/random_numbers.txt /home/
WARNING: HADOOP_PREFIX has been replaced by HADOOP_HOME. Using value of HADOOP_HOME.
root@namenode:/home# |
```

Lo subimos del namenode a HDFS

3. Ejecuta un trabajo MapReduce que calcule la mediana; el resultado debería ser un número.

```
PS D:\Workspace_VSCode_IABD> docker cp D:\Workspace_VSCode_IABD\BigDataAplicado\Practicas\Ev2\PracticasHadoop\EcosistemaHadoop\MapReduce\random_numbers.txt namenode:/home/
Successfully copied 24.1kB to namenode:/home/
PS D:\Workspace_VSCode_IABD> docker cp D:\Workspace_VSCode_IABD\BigDataAplicado\Practicas\Ev2\PracticasHadoop\EcosistemaHadoop\MapReduce\MedianCalculator.java namenode:/home/
Successfully copied 4.61kB to namenode:/home/
PS D:\Workspace_VSCode_IABD> |
```

Subimos los archivos al namenode

```
root@namenode:/# hdfs dfs -put namenode:/home/MedianCalculator.java /home/
WARNING: HADOOP_PREFIX has been replaced by HADOOP_HOME. Using value of HADOOP_HOME.
numbers.txt /home/put: '/home/MedianCalculator.java': File exists
root@namenode:/# hdfs dfs -put namenode:/home/random_numbers.txt /home/
WARNING: HADOOP_PREFIX has been replaced by HADOOP_HOME. Using value of HADOOP_HOME.
put: '/home/random_numbers.txt': File exists
root@namenode:/# |
```

Los subimos al HDFS

```
root@namenode:/# javac -cp "${hadoop classpath}" -d . /home/MedianCalculator.java
WARNING: HADOOP_PREFIX has been replaced by HADOOP_HOME. Using value of HADOOP_HOME.
root@namenode:/# |
```

Compilamos el archivo

```
root@namenode:/# jar cf median.jar MedianCalculator*.class
root@namenode:/#
```

Creamos el archivo jar

```
root@namenode:/# hadoop jar median.jar MedianCalculator /home/random_numbers.txt ./
WARNING: HADOOP_PREFIX has been replaced by HADOOP_HOME. Using value of HADOOP_PREFIX.
2025-02-12 18:24:00,223 INFO client.DefaultNoHARMFailoverProxyProvider: Connecting to Res
2.18.0.5:8032
2025-02-12 18:24:00,597 WARN mapreduce.JobResourceUploader: Hadoop command-line option pa
Tool interface and execute your application with ToolRunner to remedy this.
2025-02-12 18:24:00,617 INFO mapreduce.JobResourceUploader: Disabling Erasure Coding for
t/.staging/job_1739382444610_0009
2025-02-12 18:24:00,904 INFO input.FileInputFormat: Total input files to process : 1
2025-02-12 18:24:01,009 INFO mapreduce.JobSubmitter: number of splits:1
```

Ejecutamos el mapreduce

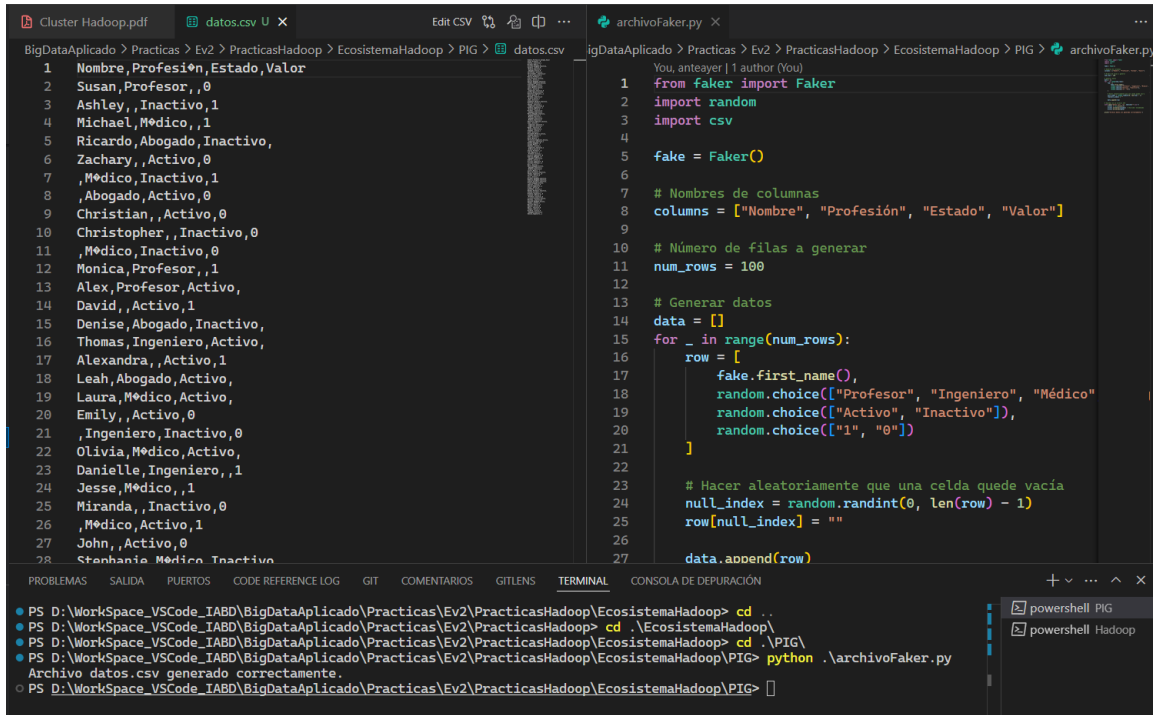
4. Utiliza HDFS para mostrar los datos. Pista: utiliza cat /* de la carpeta generada en HDFS tras el trabajo.

```
root@namenode:/# hdfs dfs -ls ./
WARNING: HADOOP_PREFIX has been replaced by HADOOP_HOME. Using value of HADOOP_PREFIX.
Found 2 items
-rw-r--r--  2 root supergroup          0 2025-02-12 18:24 _SUCCESS
-rw-r--r--  2 root supergroup       15 2025-02-12 18:24 part-r-00000
root@namenode:/# hdfs dfs -cat ./part-r-00000
WARNING: HADOOP_PREFIX has been replaced by HADOOP_HOME. Using value of HADOOP_PREFIX.
Mediana: 100.0
root@namenode:/#
```

Observamos que funciona

Pig Script

1. Genera un archivo utilizando Faker que devuelva un CSV con 4 columnas, de manera aleatoria, añade una celda sin ningún dato. Un ejemplo: [Jorge, Profesor, Activo, 1], [Rafael, Profesor,, 0].

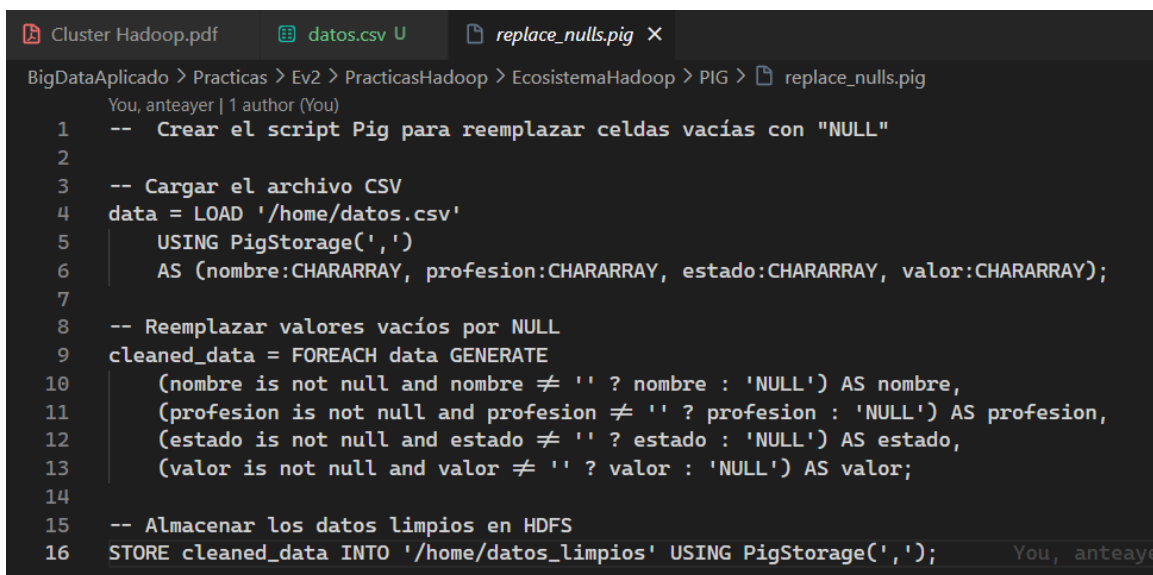


The screenshot shows a VS Code editor with two files open: `datos.csv` and `archivoFaker.py`. The `datos.csv` file contains a CSV with 4 columns: `Nombre`, `Profesión`, `Estado`, and `Valor`. The `archivoFaker.py` file contains Python code using the `Faker` library to generate 100 rows of random data. The terminal at the bottom shows the execution of the script, which successfully generates the `datos.csv` file.

```
BigDataAplicado > Practicas > Ev2 > PracticasHadoop > EcosistemaHadoop > PIG > datos.csv
1 Nombre,Profesión,Estado,Valor
2 Susan,Profesor,,0
3 Ashley,,Inactivo,1
4 Michael,Médico,,1
5 Ricardo,Abogado,Inactivo,
6 Zachary,,Activo,0
7 ,Médico,Inactivo,1
8 ,Abogado,Activo,0
9 Christian,,Activo,0
10 Christopher,,Inactivo,0
11 ,Médico,Inactivo,0
12 Monica,Profesor,,1
13 Alex,Profesor,Activo,
14 David,,Activo,1
15 Denise,Abogado,Inactivo,
16 Thomas,Ingeniero,Activo,
17 Alexandra,,Activo,1
18 Leah,Abogado,Activo,
19 Laura,Médico,Activo,
20 Emily,,Activo,0
21 ,Ingeniero,Inactivo,0
22 Olivia,Médico,Activo,
23 Danielle,Ingeniero,,1
24 Jesse,Médico,,1
25 Miranda,,Inactivo,0
26 ,Médico,Activo,1
27 John,,Activo,0
28 Stanhania,Médico,Inactivo,

BigDataAplicado > Practicas > Ev2 > PracticasHadoop > EcosistemaHadoop > PIG > archivoFaker.py
You, anteayer | 1 author (You)
1 from faker import Faker
2 import random
3 import csv
4
5 fake = Faker()
6
7 # Nombres de columnas
8 columns = ["Nombre", "Profesión", "Estado", "Valor"]
9
10 # Número de filas a generar
11 num_rows = 100
12
13 # Generar datos
14 data = []
15 for _ in range(num_rows):
16     row = [
17         fake.first_name(),
18         random.choice(["Profesor", "Ingeniero", "Médico",
19             random.choice(["Activo", "Inactivo"])),
20         random.choice(["1", "0"])
21     ]
22
23     # Hacer aleatoriamente que una celda quede vacía
24     null_index = random.randint(0, len(row) - 1)
25     row[null_index] = ""
26
27     data.append(row)
```

2. Crea un script Pig que ponga como valor a "NULL" esas celdas. La celda nula no tiene que pasar siempre en la misma columna, una fila puede tener más de 1 celda con "NULL".



The screenshot shows a VS Code editor with a file named `replace_nulls.pig` open. The script is a Pig Latin script that loads the `datos.csv` file, processes it to replace null values with "NULL", and stores the result in `datos_limpios`. The terminal at the bottom shows the execution of the script, which successfully generates the `datos_limpios` file.

```
BigDataAplicado > Practicas > Ev2 > PracticasHadoop > EcosistemaHadoop > PIG > replace_nulls.pig
You, anteayer | 1 author (You)
1 -- Crear el script Pig para reemplazar celdas vacías con "NULL"
2
3 -- Cargar el archivo CSV
4 data = LOAD '/home/datos.csv'
5     USING PigStorage(',')
6     AS (nombre:CHARARRAY, profesion:CHARARRAY, estado:CHARARRAY, valor:CHARARRAY);
7
8 -- Reemplazar valores vacíos por NULL
9 cleaned_data = FOREACH data GENERATE
10     (nombre is not null and nombre != '' ? nombre : 'NULL') AS nombre,
11     (profesion is not null and profesion != '' ? profesion : 'NULL') AS profesion,
12     (estado is not null and estado != '' ? estado : 'NULL') AS estado,
13     (valor is not null and valor != '' ? valor : 'NULL') AS valor;
14
15 -- Almacenar los datos limpios en HDFS
16 STORE cleaned_data INTO '/home/datos_limpios' USING PigStorage(',');
```


3. Ejecuta el script y muestra los datos en HDFS.

```
PS D:\Workspace_VSCode_IABD\BigDataAplicado\Practicas\Ev2\PracticasHadoop\EcosistemaHadoop> docker cp D:\Workspace_VSCode_IABD\BigDataAplicado\Practicas\Ev2\PracticasHadoop\EcosistemaHadoop\PIG\datos.csv namenode:/home
Successfully copied 4.1kB to namenode:/home
PS D:\Workspace_VSCode_IABD\BigDataAplicado\Practicas\Ev2\PracticasHadoop\EcosistemaHadoop> docker cp D:\Workspace_VSCode_IABD\BigDataAplicado\Practicas\Ev2\PracticasHadoop\EcosistemaHadoop\PIG\replace_nulls.pig namenode:/home
Successfully copied 2.56kB to namenode:/home
PS D:\Workspace_VSCode_IABD\BigDataAplicado\Practicas\Ev2\PracticasHadoop\EcosistemaHadoop> docker cp D:\Workspace_VSCode_IABD\BigDataAplicado\Practicas\Ev2\PracticasHadoop\EcosistemaHadoop\PIG\filter_nulls.pig namenode:/home
Successfully copied 2.56kB to namenode:/home
PS D:\Workspace_VSCode_IABD\BigDataAplicado\Practicas\Ev2\PracticasHadoop\EcosistemaHadoop> █
```

movemos los archivos a namenode

```
root@namenode:/# hdfs dfs -put namenode:/home/datos.csv /home/
WARNING: HADOOP_PREFIX has been replaced by HADOOP_HOME. Using value of HADOOP_PREFIX.
root@namenode:/# hdfs dfs -put namenode:/home/filter_nulls.pig /home/
WARNING: HADOOP_PREFIX has been replaced by HADOOP_HOME. Using value of HADOOP_PREFIX.
root@namenode:/# hdfs dfs -put namenode:/home/replace_nulls.pig /home/
WARNING: HADOOP_PREFIX has been replaced by HADOOP_HOME. Using value of HADOOP_PREFIX.
root@namenode:/# █
```

Subimos a hdfs los 3 ficheros

```
root@namenode:/# pig /home/replace_nulls.pig
WARNING: HADOOP_PREFIX has been replaced by HADOOP_HOME. Using value of HADOOP_PREFIX.
2025-02-12 10:50:04,297 INFO pig.ExecTypeProvider:
2025-02-12 10:50:04,298 INFO pig.ExecTypeProvider:
```

Ejecutamos el .pig

```
2025-02-12 10:52:56,106 [main] INFO org.apache.pig.Main - Pig script completed in 2 minutes, 51 seconds and 859 milliseconds (171859 ms)
root@namenode:/# █
```

Captura de que se ha ejecutado correctamente

```
^[[Aroot@namenode:/# hdfs dfs -ls /home/datos_filtrados/
WARNING: HADOOP_PREFIX has been replaced by HADOOP_HOME. Using value of HADOOP_PREFIX.
Found 2 items
-rw-r--r--  2 root supergroup          0 2025-02-12 10:54 /home/datos_filtrados/_SUCCESS
-rw-r--r--  2 root supergroup    2459 2025-02-12 10:54 /home/datos_filtrados/part-m-00000
root@namenode:/# hdfs dfs -cat /home/datos_filtrados/part-m-00000
WARNING: HADOOP_PREFIX has been replaced by HADOOP_HOME. Using value of HADOOP_PREFIX.
Nombre,Profesi n,Estado,Valor
Susan,Profesor,NULL,0
Ashley,NULL,Inactivo,1
Michael,M dico,NULL,1
Ricardo,Abogado,Inactivo,NULL
Zachary,NULL,Activo,0
NULL,M dico,Inactivo,1
NULL,Abogado,Activo,0
Christian,NULL,Activo,0
```

Captura de que los ha reemplazado

4. Crea un archivo Pig que borre las filas con más de 2 celdas con valores "NULL".

```
Cluster Hadoop.pdf  datos.csv U  filter_nulls.pig X
BigDataAplicado > Practicas > Ev2 > PracticasHadoop > EcosistemaHadoop > PIG > filter_nulls.pig
You, anteayer | 1 author (You)
1  -- Crear un script Pig para eliminar filas con más de 2 valores NULL
2
3  -- Cargar los datos procesados
4  data = LOAD '/home/datos_limpios'
5      USING PigStorage(',')
6      AS (nombre:CHARARRAY, profesion:CHARARRAY, estado:CHARARRAY, valor:CHARARRAY);
7
8  -- Contar cuántos valores NULL hay en cada fila
9  filtered_data = FILTER data BY
10     ((nombre == 'NULL' ? 1 : 0) +
11      (profesion == 'NULL' ? 1 : 0) +
12      (estado == 'NULL' ? 1 : 0) +
13      (valor == 'NULL' ? 1 : 0)) ≤ 2;
14
15  -- Guardar el resultado en HDFS
16  STORE filtered_data INTO '/home/datos_filtrados' USING PigStorage(',');
```

5. Ejecuta el script y muestra los datos en HDFS.

```
root@namenode:/# pig /home/filter_nulls.pig
WARNING: HADOOP_PREFIX has been replaced by HADOOP_HOME. Using value of HADOOP_PREFIX.
2025-02-12 10:54:07,690 INFO pig.ExecTypeProvider: Trying ExecType : LOCAL
2025-02-12 10:54:07,691 INFO pig.ExecTypeProvider: Trying ExecType : MAPREDUCE
2025-02-12 10:54:07,691 INFO pig.ExecTypeProvider: Picked MAPREDUCE as the ExecType
2025-02-12 10:54:07,740 [main] INFO org.apache.pig.Main - Apache Pig version 0.16.0
```

Ejecutamos el .pig

```
2025-02-12 10:56:57,750 [main] INFO org.apache.pig.Main - Pig script completed in 2 minutes, 50 seconds and 87 milliseconds (170087 ms)
root@namenode:/#
```

Captura de que se ha ejecutado correctamente

```
root@namenode:/# hdfs dfs -ls /home/
WARNING: HADOOP_PREFIX has been replaced by HADOOP_HOME. Using value of HADOOP_PREFIX.
Found 5 items
-rw-r--r--  2 root supergroup      2118 2025-02-12 10:46 /home/datos.csv
drwxr-xr-x  - root supergroup         0 2025-02-12 10:54 /home/datos_filtrados
drwxr-xr-x  - root supergroup         0 2025-02-12 10:50 /home/datos_limpios
-rw-r--r--  2 root supergroup      592 2025-02-12 10:46 /home/filter_nulls.pig
-rw-r--r--  2 root supergroup      724 2025-02-12 10:46 /home/replace_nulls.pig
^[[Aroot@namenode:/# hdfs dfs -cat /home/datos_limpios/part-r-00000
WARNING: HADOOP_PREFIX has been replaced by HADOOP_HOME. Using value of HADOOP_PREFIX.
cat: '/home/datos_limpios/part-r-00000': No such file or directory
root@namenode:/# hdfs dfs -cat /home/datos_limpios/part-m-00000
WARNING: HADOOP_PREFIX has been replaced by HADOOP_HOME. Using value of HADOOP_PREFIX.
Nombre,Profesion,Estado,Valor
Susan,Profesor,NULL,0
Ashley,NULL,Inactivo,1
Michael,Médico,NULL,1
Ricardo,Abogado,Inactivo,NULL
Zachary,NULL,Activo,0
NULL,Médico,Inactivo,1
```

Vemos que funciona correctamente

Sqoop

1. Añade un contenedor al cluster con una BBDD MySQL. Dentro de la base de datos, crea una tabla llamada empleados con las siguientes columnas: id INT PRIMARY KEY, nombre VARCHAR(50), departamento VARCHAR(50), salario DECIMAL(10,2), fecha_contratacion DATE.

```
apuntes.txt M  docker-compose.yml M x  consola.py  Cluster Hadoop.pdf
BigDataAplicado > Practicas > Ev2 > Hadoop > docker-compose.yml
You, hace 34 segundos | 3 authors (You and others)
1  services:
2      # ----- MYSQL -----
3      mysql:
4          container_name: mysql # Nombre del contenedor
5          hostname: mysql # Nombre del host
6          image: mysql:latest # Imagen de MySQL
7          environment:
8              MYSQL_ROOT_PASSWORD: root # Contraseña del usuario root
9              MYSQL_DATABASE: hadoop # Nombre de la base de datos
10             #MYSQL_USER: user # Nombre del usuario
11             #MYSQL_PASSWORD: password # Contraseña del usuario
12          ports:
13              - "3306:3306" # Puerto de MySQL
14          volumes:
15              - hadoop_mysql:/var/lib/mysql # Volumen de MySQL
16          networks:
17              hadoop_network:
18                  aliases:
19                      - mysql # Alias de MySQL
20
21      # ----- HADOOP -----
```

```
volumes:
  hadoop_namenode:
  # hadoop_namenode2:
  hadoop_datanode:
  # hadoop_datanode2:
  # hadoop_datanode3:
  # hadoop_datanode4:
  # hadoop_datanode5:
  hadoop_mysql: # Volumen de MySQL
```

Modificamos el Docker-compose de manera que en *service* añadimos el mysql y su respectivo volumen en *volumes*

<input type="checkbox"/>	Name	Container ID	Image	Port(s)	CPU (%)	Last st	Actions
<input type="checkbox"/>	hadoop	-	-	-	0%	1 secon	
<input type="checkbox"/>	namenode	12cdbc392339	hadoop-namenode	8020:8020	0%	2 secon	
<input type="checkbox"/>	mysql	ef8045774058	mysql:latest	3306:3306	0%	2 secon	
<input type="checkbox"/>	datanode	ca0cea3c21ee	hadoop-datanode	9864:9864	0%	1 secon	
<input type="checkbox"/>	nodemanager	5c940d46c2cb	hadoop-nodemanager	8043:8043	0%	1 secon	
<input type="checkbox"/>	resourcemanager	eeed4e6554c1	hadoop-resourcemanager	8088:8088	0%	1 secon	

Aquí vemos que se ha creado correctamente.

```
PS D:\Workspace_VSCode_IABD\BigDataAplicado\Practicas\Ev2\Hadoop> docker exec -it mysql bin/bash
bash-5.1# mysql -u root -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 9
Server version: 9.2.0 MySQL Community Server - GPL

Copyright (c) 2000, 2025, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql>
```

Entramos a la bd (usuario root y password root)

```
mysql> use hadoop
Database changed
mysql> CREATE TABLE empleados (
  ->   id INT PRIMARY KEY,
  ->   nombre VARCHAR(50),
  ->   departamento VARCHAR(50),
  ->   salario DECIMAL(10,2),
  ->   fecha_contratacion DATE
  -> );
Query OK, 0 rows affected (0.03 sec)

mysql> show tables;
+-----+
| Tables_in_hadoop |
+-----+
| empleados        |
+-----+
1 row in set (0.01 sec)

mysql>
```

Creamos la tabla. También podemos hacer un .sql para que la cree nada más iniciarse.

2. Inserta al menos 2000 registros en la tabla empleados.

```
Cluster Hadoop.pdf  genera_empleados.py x  apuntes.txt M  empleados_script.sql U
BigDataAplicado > Practicas > Ev2 > Hadoop > mysql > genera_empleados.py > ...
30     values = (i, nombre, departamento, salario, fecha_contratacion)
31
32     # Ejecutar la inserción
33     cursor.execute(query, values)
34
35 # Confirmar las inserciones en la base de datos
36 db.commit()
37
38 # Cerrar la conexión
39 cursor.close()
40 db.close()
41
42 print("2000 registros insertados con éxito en la tabla 'empleados'.")
43
```

PROBLEMAS SALIDA PUERTOS CODE REFERENCE LOG GIT COMENTARIOS GITLENS **TERMINAL** CONSOLA DE DEPURACIÓN

```
File "C:\Users\Andrei\AppData\Local\Packages\PythonSoftwareFoundation.Python.3.13_qbz5n2kfra8p0\LocalCache\local-packag
Python313\site-packages\mysql\connector\opentelemetry\context_propagation.py", line 97, in wrapper
    return method(cnx, *args, **kwargs)
File "C:\Users\Andrei\AppData\Local\Packages\PythonSoftwareFoundation.Python.3.13_qbz5n2kfra8p0\LocalCache\local-packag
Python313\site-packages\mysql\connector\connection.py", line 872, in cmd_query
    result = self._handle_result(self._send_cmd(ServerCmd.QUERY, query))
File "C:\Users\Andrei\AppData\Local\Packages\PythonSoftwareFoundation.Python.3.13_qbz5n2kfra8p0\LocalCache\local-packag
Python313\site-packages\mysql\connector\connection.py", line 648, in _handle_result
    raise get_exception(packet)
mysql.connector.errors.DataError: 1406 (22001): Data too long for column 'departamento' at row 1
PS D:\Workspace_VSCode_IABD\BigDataAplicado\Practicas\Ev2\Hadoop\mysql> python .\genera_empleados.py
● 2000 registros insertados con éxito en la tabla 'empleados'.
○ PS D:\Workspace_VSCode_IABD\BigDataAplicado\Practicas\Ev2\Hadoop\mysql>
```

Ejecutamos el script para añadir los 2000 registros (El error era que el faker me daba 100 caracteres en vez de 50 para departamento).

1994	Anne Hernandez	Public librarian	66365.64	2017-10-08
1995	Brandon Brown	Chemist, analytical	85751.65	2019-08-02
1996	Anthony Schaefer	Clothing/textile technologist	73131.03	2023-03-26
1997	Keith Walton	IT technical support officer	100189.80	2018-02-04
1998	Jamie Hardy DVM	Scientist, biomedical	72277.11	2015-07-08
1999	Jay Velazquez	TEFL teacher	74956.62	2015-08-04
2000	Cynthia Gray	Research scientist (physical sciences)	85309.74	2019-04-20

```
2000 rows in set (0.00 sec)

mysql>
```

Con select * from empleados vemos que tiene 2000 líneas.

3. Utiliza Sqoop para importar los datos de la tabla empleados desde la base de datos MySQL hacia HDFS y almacénalos en un directorio llamado /user/sqoop/empleados en HDFS.

```
root@namenode:/# sqoop import --connect jdbc:mysql://mysql:3306/hadoop \  
> --username root --password root \  
> --table empleados --target-dir /user/sqoop/empleados \  
> --m 1 \  
> --fields-terminated-by ',' --lines-terminated-by '\n'  
Warning: /usr/local/sqoop/./hbase does not exist! HBase imports will fail.  
Please set $HBASE_HOME to the root of your HBase installation.  
Warning: /usr/local/sqoop/./hcatalog does not exist! HCatalog jobs will fail.  
Please set $HCAT_HOME to the root of your HCatalog installation.  
Warning: /usr/local/sqoop/./accumulo does not exist! Accumulo imports will fail.  
Please set $ACCUMULO_HOME to the root of your Accumulo installation.  
Warning: /usr/local/sqoop/./zookeeper does not exist! Accumulo imports will fail.  
Please set $ZOOKEEPER_HOME to the root of your Zookeeper installation.  
WARNING: HADOOP_PREFIX has been replaced by HADOOP_HOME. Using value of HADOOP_PREFIX.  
2025-02-12 13:55:08,801 INFO sqoop.Sqoop: Running Sqoop version: 1.4.7  
2025-02-12 13:55:08,829 WARN tool.BaseSqoopTool: Setting your password on the command-line  
instead.  
2025-02-12 13:55:08,926 INFO manager.MySQLManager: Preparing to use a MySQL streaming resu  
2025-02-12 13:55:08,926 INFO tool.CodeGenTool: Beginning code generation  
Wed Feb 12 13:55:09 UTC 2025 WARN: Establishing SSL connection without server's identity v  
According to MySQL 5.5.45+, 5.6.26+ and 5.7.6+ requirements SSL connection must be establis
```

Ejecutamos el comando para importar

```
2025-02-12 13:55:36,516 INFO mapreduce.ImportJobBase: Transferred 99.8857 KB in 23.3724 seconds (4.2737 KB/sec)  
2025-02-12 13:55:36,520 INFO mapreduce.ImportJobBase: Retrieved 2000 records.  
root@namenode:/#
```

Captura de finalización del comando

4. Comprueba los datos importados en HDFS utilizando los comandos `hdfs dfs -ls` y `hdfs dfs -cat`.

```
root@namenode:/# hdfs dfs -ls /user/sqoop/  
WARNING: HADOOP_PREFIX has been replaced by HADOOP_HOME. Using value of HADOOP_PREFIX.  
Found 1 items  
drwxr-xr-x - root supergroup 0 2025-02-12 13:55 /user/sqoop/empleados  
root@namenode:/# hadoop fs -cat /user/sqoop/empleados/* | head -n 10  
WARNING: HADOOP_PREFIX has been replaced by HADOOP_HOME. Using value of HADOOP_PREFIX.  
Investigación y Desarrollo,2020-05-13,1,Sherry Brooks,114823.97  
IT,2020-12-07,2,Sarah Humphrey,88835.04  
Logística,2022-04-01,3,Catherine Fry,114450.36  
Investigación y Desarrollo,2019-07-19,4,William Morgan,32667.73  
Investigación y Desarrollo,2022-06-09,5,Robert Nunez,90392.15  
IT,2016-01-08,6,Adrienne Smith,40595.54  
Finanzas,2017-11-03,7,Deanna Garrison,51008.60  
Recursos Humanos,2020-04-20,8,Martha Villanueva,92592.03  
Recursos Humanos,2016-07-18,9,Rebekah Jenkins,103418.38  
Ventas,2018-03-12,10,Michelle Morales,52345.88  
cat: Unable to write to output stream.  
root@namenode:/#
```

Comprobación con `-ls` y `-cat`

- Realiza una segunda importación utilizando una consulta SQL que filtre los datos. Por ejemplo: importa solo los empleados del departamento "Ventas".

```
root@namenode:/# sqoop import --connect jdbc:mysql://mysql:3306/hadoop \
> --username root --password root \
> --query "SELECT id, nombre, departamento, salario, fecha_contratacion FROM empleados WHERE departamento = 'Ventas' AND \"
CONDITIONS\" \
> --target-dir /user/sqoop/empleados_ventas2 --m 1 \
> --fields-terminated-by ',' --lines-terminated-by '\n'
Warning: /usr/local/sqoop/./hbase does not exist! HBase imports will fail.
Please set $HBASE_HOME to the root of your HBase installation.
Warning: /usr/local/sqoop/./hcatalog does not exist! HCatalog jobs will fail.
Please set $HCAT_HOME to the root of your HCatalog installation.
Warning: /usr/local/sqoop/./accumulo does not exist! Accumulo imports will fail.
Please set $ACCUMULO_HOME to the root of your Accumulo installation.
Warning: /usr/local/sqoop/./zookeeper does not exist! Accumulo imports will fail.
Please set $ZOOKEEPER_HOME to the root of your Zookeeper installation.
WARNING: HADOOP_PREFIX has been replaced by HADOOP_HOME. Using value of HADOOP_PREFIX.
2025-02-12 14:05:15,158 INFO sqoop.Sqoop: Running Sqoop version: 1.4.7
2025-02-12 14:05:15,187 WARN tool.BaseSqoopTool: Setting your password on the command-line is insecure. Consider using -P i
```

Ejecutamos el comando

```
2025-02-12 14:05:33,158 INFO mapreduce.ImportJobBase: Transferred 7.5244 KB in 15.179 seconds (507.6079 bytes/sec)
2025-02-12 14:05:33,161 INFO mapreduce.ImportJobBase: Retrieved 168 records.
root@namenode:/#
```

Captura de la finalización

```
root@namenode:/# hdfs dfs -ls /user/sqoop/
WARNING: HADOOP_PREFIX has been replaced by HADOOP_HOME. Using value of HADOOP_PREFIX.
Found 2 items
drwxr-xr-x - root supergroup          0 2025-02-12 13:55 /user/sqoop/empleados
drwxr-xr-x - root supergroup          0 2025-02-12 14:05 /user/sqoop/empleados_ventas2
root@namenode:/# hadoop fs -cat /user/sqoop/empleados_ventas2 | head -n 10
WARNING: HADOOP_PREFIX has been replaced by HADOOP_HOME. Using value of HADOOP_PREFIX.
cat: '/user/sqoop/empleados_ventas2': Is a directory
root@namenode:/# hadoop fs -cat /user/sqoop/empleados_ventas2/* | head -n 10
WARNING: HADOOP_PREFIX has been replaced by HADOOP_HOME. Using value of HADOOP_PREFIX.
10,Michelle Morales,Ventas,52345.88,2018-03-12
15,Andre Griffin,Ventas,111782.39,2023-05-19
30,Michael Johnson,Ventas,43764.39,2023-08-12
42,Ricardo Smith,Ventas,87615.23,2015-09-06
66,Ashley Neal,Ventas,59153.09,2015-06-10
71,Michael Lopez,Ventas,73536.97,2017-07-12
74,Jose Moore,Ventas,119620.30,2018-12-17
81,Craig Santos,Ventas,112650.08,2016-10-06
96,Bryan Kelly,Ventas,84052.49,2015-08-05
105,Jason Benjamin,Ventas,86724.73,2021-08-21
root@namenode:/#
```

Comprobación con -ls y -cat

- Automatiza la importación para que se ejecute diariamente mediante un cron job en el contenedor. Configura el cron para que ejecute el comando Sqoop a las 00:00 cada día.

Nos moveremos al namenode y posteriormente haremos un:

- apt-get update
- apt-get install cron
- apt-get install nano -y


```
root@namenode:/# service cron start
Starting periodic command scheduler: cron.
root@namenode:/#
```

Iniciamos el servicio

```
PROBLEMAS  SALIDA  PUERTOS  CODE REFERENCE LOG  GIT  COMENTARIOS  GITLENS  TERMINAL  CONSOLA DE DEPURACIÓN
GNU nano 5.4 /tmp/crontab.ZbziP4/crontab *
# Edit this file to introduce tasks to be run by cron.
#
# Each task to run has to be defined through a single line
# indicating with different fields when the task will be run
# and what command to run for the task
#
# To define the time you can provide concrete values for
# minute (m), hour (h), day of month (dom), month (mon),
# and day of week (dow) or use '*' in these fields (for 'any').
#
# Notice that tasks will be started based on the cron's system
# daemon's notion of time and timezones.
#
# Output of the crontab jobs (including errors) is sent through
# email to the user the crontab file belongs to (unless redirected).
#
# For example, you can run a backup of all your user accounts
# at 5 a.m every week with:
# 0 5 * * 1 tar -zcf /var/backups/home.tgz /home/
#
# For more information see the manual pages of crontab(5) and cron(8)
#
# m h dom mon dow  command
0 0 * * * sqoop import --connect jdbc:mysql://localhost:3306/hadoop --username root --password root --
```

Dentro de crontab -e escribiremos la tarea a hacer:

```
0 0 * * * sqoop import --connect jdbc:mysql://localhost:3306/hadoop --username
root --password root --query "SELECT id, nombre, departamento, salario,
fecha_contratacion FROM empleados WHERE departamento = 'Ventas' AND
\$CONDITIONS" --target-dir /user/sqoop/empleados_ventas2 --m 1 --fields-
terminated-by ',' --lines-terminated-by '\n' > /var/log/sqoop_import.log 2>&1
```

```
root@namenode:/# crontab -l
# Edit this file to introduce tasks to be run by cron.
#
# Each task to run has to be defined through a single line
# indicating with different fields when the task will be run
# and what command to run for the task
#
# To define the time you can provide concrete values for
# minute (m), hour (h), day of month (dom), month (mon),
# and day of week (dow) or use '*' in these fields (for 'any').
#
# Notice that tasks will be started based on the cron's system
# daemon's notion of time and timezones.
#
# Output of the crontab jobs (including errors) is sent through
# email to the user the crontab file belongs to (unless redirected).
#
# For example, you can run a backup of all your user accounts
# at 5 a.m every week with:
# 0 5 * * 1 tar -zcf /var/backups/home.tgz /home/
#
# For more information see the manual pages of crontab(5) and cron(8)
#
# m h dom mon dow  command
0 0 * * * sqoop import --connect jdbc:mysql://localhost:3306/hadoop --username root --password root --query "SELECT id, nombre, departamento, salario, fecha_co
ntratacion FROM empleados WHERE departamento = 'Ventas' AND \$CONDITIONS" --target-dir /user/sqoop/empleados_ventas2 --m 1 --fields-terminated-by ',' --lines-t
erminated-by '\n' > /var/log/sqoop_import.log 2>&1
root@namenode:/#
```

Con crontab -l vemos si se a configurado bien

```
root@namenode:/# service cron start
Starting periodic command scheduler: cron.
root@namenode:/#
```

Iniciamos el servicio

Flume

1. Crea un script bash que genere automáticamente logs de prueba. Este script debería escribir una línea en un archivo `/var/logs/app.log` cada 5 segundos. Un ejemplo de línea podría ser: `INFO - 2025-01-01 12:00:00 - Usuario accedió al sistema.` **Pista:** Usa un comando como `while true; do echo "INFO - $(date '+%Y-%m-%d %H:%M:%S') - Usuario accedió al sistema." >> /var/logs/app.log; sleep 5; done.`
2. Configura los componentes Source, Channel y Sink de Flume para recoger los datos del archivo `/var/logs/app.log` y almacenarlos en un directorio HDFS llamado `/user/flume/logs`.
3. Monitorea el agente Flume para asegurarte de que está recopilando y transfiriendo datos a HDFS. Usa comandos como `hdfs dfs -ls /user/flume/logs` para comprobar los datos almacenados.

Procesamiento avanzado de datos

1. Haz los cambios necesarios en el Dockerfile y Docker Compose para que funcione Apache Hive..
2. Crea una base de datos llamada `bigdata_practica`.
3. Importa el CSV generado en el Task 2 con Pig y crea una tabla externa en Hive llamada `usuarios` con las siguientes columnas: `nombre` STRING, `profesion` STRING, `estado` STRING, `activo` INT.
4. Realiza consultas SQL que permitan:

- a. Obtener el número de filas donde alguna columna tenga el valor "NULL".
- b. Agrupar los datos por la columna profesion y mostrar la frecuencia de cada valor.
- c. Añade particiones a la tabla basándote en la columna estado para mejorar el rendimiento de las consultas.

Apache Spark

1. Haz los cambios necesarios en el Dockerfile y Docker Compose para que se puedan lanzar trabajos Spark.
2. Configura un flujo de datos en Spark Streaming que lea datos en tiempo real desde el directorio /user/flume/logs en HDFS.
3. Modifica el script the bash del apartado de FLUME para que de manera aleatoria añada líneas con valor ERROR- 2025-01-01 12:00:00 o WARN- 2025-01-01 12:00:00.
4. El flujo debe filtrar solo las líneas que contengan el texto "ERROR" y almacenarlas en otro directorio de HDFS llamado /user/spark/errors.
5. Crea un job Spark que analice los datos de logs almacenados en HDFS para calcular:
6. El número total de líneas procesadas.
7. La frecuencia de cada tipo de mensaje (INFO, WARN, ERROR).