



Big Data Aplicado

Profesor/a: José Manuel González Rodríguez

ACTIVIDAD EVALUABLE 2.3.- INGESTA DE DATOS
14/05/2025

ÍNDICE

1.- Introducción	3
1.1.- Objetivos de la actividad	3
1.2.- Descripción del entorno de trabajo	3
1.3.- Herramientas utilizadas	4
2.- Preparación del entorno	5
2.1.- Máquina virtual: configuración y acceso	5
2.2.- Verificación de servicios: HDFS, YARN y Job History Server	6
2.3.- Comunicación entre sistema de ficheros local y HDFS	9
3.- Ejercicio 1 – Instalación de Apache Pig	12
3.1.- Descarga y extracción de Pig	12
3.2.- Configuración del entorno (.bashrc)	13
4.- Ejercicio 2 – Trabajo con Pig y procesamiento de datos	15
4.1.- Carga del archivo bike_trips.csv en HDFS	15
4.2.- Script Pig 1: duración media de viajes (>15 min) por tipo de usuario	16
4.2.1.- Código del script	16
4.2.2.- Ejecución y salida (PMGA-A)	17
4.3.- Script Pig 2: número de viajes por estación de salida	18
4.3.1.- Código del script	18
4.3.2.- Ejecución y salida (PMGA-B)	19
5.- Conclusiones	20
5.1.- Valoración del uso de Pig para ingesta y análisis	20
5.2.- Dificultades encontradas y cómo se resolvieron	20
5.3.- Posibles mejoras o ampliaciones del trabajo	21
6.- Mapa Mental del Trabajo	22
7.- Anexo: Fichero de Entregables	23
7.1.- Directorio: principal	23
7.2.- Directorio: dataset	23
7.3.- Directorio: Enunciado	23
7.4.- Directorio: script	23

1.- INTRODUCCIÓN

La presente actividad evaluable 2.3 del módulo “**Big Data Aplicado**” tiene como finalidad familiarizar al alumno con herramientas de ingesta de datos utilizadas en entornos Big Data. En particular, se centra en el uso y configuración de **Apache Pig** como lenguaje de alto nivel para el procesamiento de grandes volúmenes de datos, además de reforzar conceptos esenciales sobre **HDFS** (Hadoop Distributed File System) y su interacción con el sistema de ficheros local.

La ingesta y transformación de datos en un entorno distribuido es una fase crítica en cualquier arquitectura de Big Data, ya que determina la eficiencia y calidad del procesamiento posterior. Por ello, el objetivo de este trabajo es realizar ejercicios prácticos que permitan instalar, configurar y utilizar Pig para procesar un conjunto de datos reales, mediante scripts en PigLatin y ejecución en modo MapReduce.

1.1.- OBJETIVOS DE LA ACTIVIDAD

- Instalar correctamente la herramienta Apache Pig en una máquina virtual con Hadoop.
- Configurar las variables de entorno necesarias y activar los servicios relacionados (Job History Server).
- Familiarizarse con los comandos de interacción entre el sistema de ficheros local y HDFS.
- Desarrollar scripts en PigLatin para transformar e interpretar datos masivos.
- Ejecutar tareas en modo MapReduce y almacenar resultados en directorios definidos dentro de HDFS.
- Documentar el proceso con capturas de pantalla y explicaciones claras de cada paso realizado.

1.2.- DESCRIPCIÓN DEL ENTORNO DE TRABAJO

El entorno de trabajo está basado en una **máquina virtual preconfigurada con Hadoop**, que puede ser importada a través de VirtualBox a partir de un archivo .ova. Esta máquina dispone de HDFS operativo y es posible añadirle YARN manualmente si es necesario.

Entre las características clave del entorno destacan:

- Usuario predeterminado: hadoop
- Contraseñas: Admin1. tanto para root como para hadoop

- Redirección de puertos:
 - 22222 (host) → 22 (SSH en la VM)
 - 9880 (host) → 9870 (navegador HDFS Web UI)
- Directorio base de trabajo: /home/hadoop/
- El sistema operativo es una distribución de Linux optimizada para el ecosistema Hadoop.

1.3.- HERRAMIENTAS UTILIZADAS

Las herramientas y tecnologías utilizadas para el desarrollo de esta actividad son las siguientes:

Herramienta / Tecnología	Descripción
Hadoop HDFS	Sistema de ficheros distribuido para almacenamiento escalable y tolerante a fallos.
YARN (opcional)	Framework para la gestión de recursos y ejecución de tareas distribuidas.
Apache Pig 0.17.0	Plataforma de alto nivel para crear programas MapReduce usando PigLatin.
PigLatin	Lenguaje de scripts de Apache Pig para la manipulación de datos.
VirtualBox	Software de virtualización utilizado para importar y ejecutar la máquina virtual.
Linux Bash	Shell de comandos para la administración del entorno y ejecución de tareas.

2.- PREPARACIÓN DEL ENTORNO

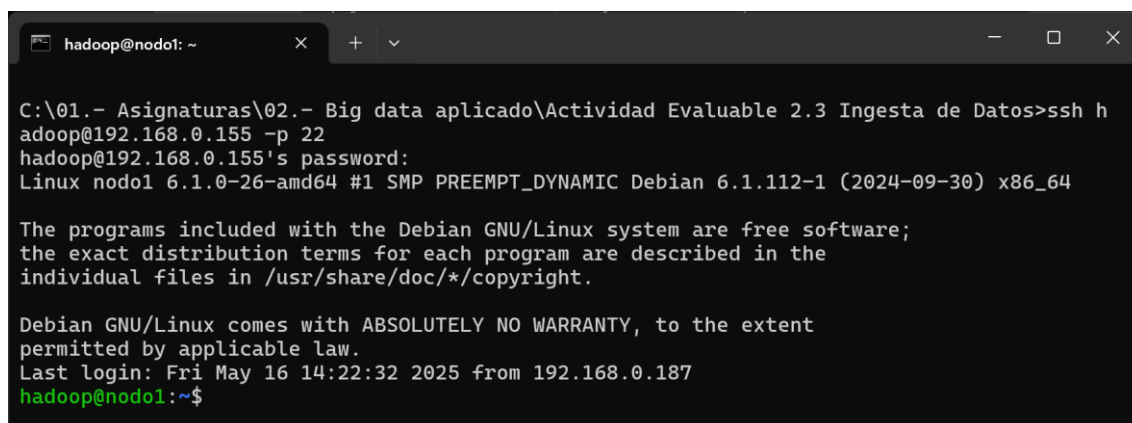
Antes de comenzar con la ejecución de scripts en Pig, es imprescindible disponer de un entorno de trabajo completamente operativo. Este apartado describe los pasos para configurar la máquina virtual, verificar el estado de los servicios esenciales del ecosistema Hadoop, y establecer correctamente la comunicación entre el sistema de archivos local y HDFS.

2.1.- MÁQUINA VIRTUAL: CONFIGURACIÓN Y ACCESO

Para esta actividad, se ha utilizado una **máquina virtual proporcionada por el profesor**, la cual viene preconfigurada con el sistema operativo Linux y el framework Hadoop. Esta máquina se importa en **VirtualBox** mediante un archivo con extensión .ova.

Una vez importada la máquina, se ha verificado y/o configurado lo siguiente:

- **Usuario:** hadoop
- **Contraseña:** Admin1.
- **Redirección de puertos:**
 - **22222** del host → **22** de la VM (permite acceso SSH)
 - **9880** del host → **9870** de la VM (interfaz web de HDFS)



```
hadoop@nodo1: ~  
C:\01.- Asignaturas\02.- Big data aplicado\Actividad Evaluable 2.3 Ingesta de Datos>ssh h  
adoop@192.168.0.155 -p 22  
hadoop@192.168.0.155's password:  
Linux nodo1 6.1.0-26-amd64 #1 SMP PREEMPT_DYNAMIC Debian 6.1.112-1 (2024-09-30) x86_64  
  
The programs included with the Debian GNU/Linux system are free software;  
the exact distribution terms for each program are described in the  
individual files in /usr/share/doc/*/copyright.  
  
Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent  
permitted by applicable law.  
Last login: Fri May 16 14:22:32 2025 from 192.168.0.187  
hadoop@nodo1:~$
```

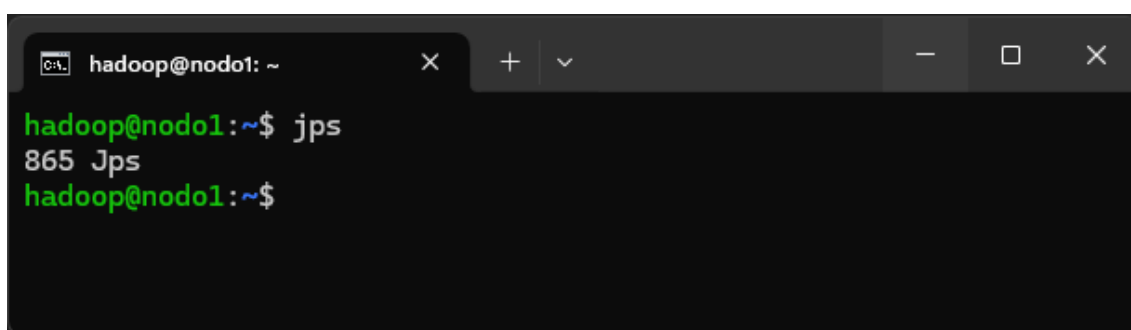
La pantalla muestra cómo un usuario, desde una terminal de Windows, ha utilizado el comando **ssh** [hadoop@192.168.0.155](ssh://hadoop@192.168.0.155) -p 22 para conectarse exitosamente al servidor Linux 192.168.0.155 (llamado nodo1) como el usuario hadoop.

2.2.- VERIFICACIÓN DE SERVICIOS: HDFS, YARN Y JOB HISTORY SERVER

Para poder ejecutar scripts de Pig en modo MapReduce, es necesario que estén activos los siguientes servicios del ecosistema Hadoop:

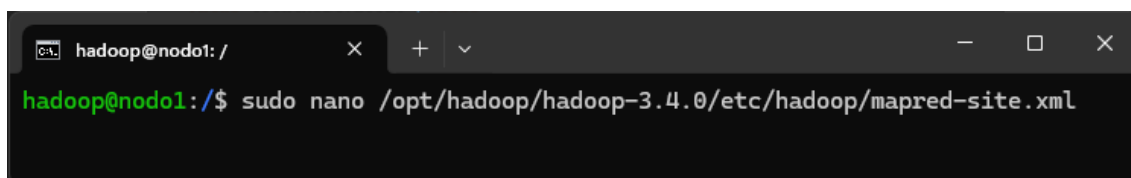
- HDFS (NameNode y DataNode)
- YARN (ResourceManager y NodeManager)
- Job History Server (para la monitorización de jobs MapReduce)

Se ha utilizado el comando `jps` para comprobar qué servicios están en ejecución:



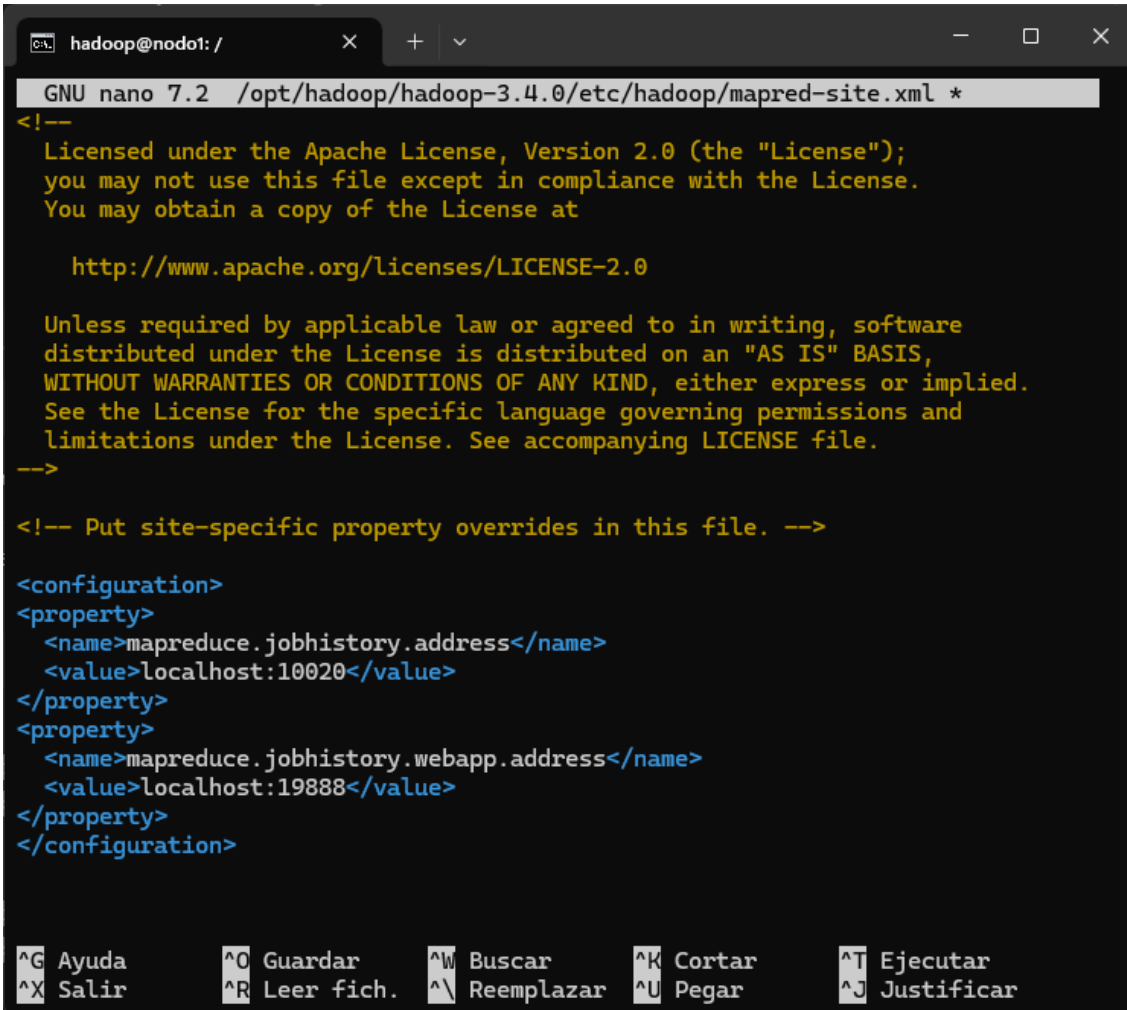
```
hadoop@nodo1: ~  
hadoop@nodo1:~$ jps  
865 Jps  
hadoop@nodo1:~$
```

En esta pantalla, se ejecuta el comando `jps` para verificar si los servicios están en ejecución, observando que ninguno de ellos se encuentra activo.



```
hadoop@nodo1: /  
hadoop@nodo1:/$ sudo nano /opt/hadoop/hadoop-3.4.0/etc/hadoop/mapred-site.xml
```

En esta pantalla, se ejecuta el comando `sudo nano /opt/hadoop/hadoop-3.4.0/etc/hadoop/mapred-site.xml` para editar el archivo de configuración.



```
hadoop@nodo1: /
GNU nano 7.2 /opt/hadoop/hadoop-3.4.0/etc/hadoop/mapred-site.xml *
<!--
Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

    http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License. See accompanying LICENSE file.
-->

<!-- Put site-specific property overrides in this file. -->

<configuration>
<property>
  <name>mapreduce.jobhistory.address</name>
  <value>localhost:10020</value>
</property>
<property>
  <name>mapreduce.jobhistory.webapp.address</name>
  <value>localhost:19888</value>
</property>
</configuration>

^G Ayuda      ^O Guardar    ^W Buscar     ^K Cortar     ^T Ejecutar
^X Salir      ^R Leer fich. ^\ Reemplazar ^U Pegar      ^J Justificar
```

En esta pantalla, se utiliza el editor nano para escribir lo siguiente:

```
<property>
  <name>mapreduce.jobhistory.address</name>
  <value>localhost:10020</value>
</property>
<property>
  <name>mapreduce.jobhistory.webapp.address</name>
  <value>localhost:19888</value>
</property>
```

A continuación, se presiona CTRL+O para guardar los cambios y luego CTRL+X para salir del editor.

```
hadoop@nodo1: ~  
hadoop@nodo1:~$ start-dfs.sh  
Starting namenodes on [localhost]  
Starting datanodes  
Starting secondary namenodes [nodo1]  
hadoop@nodo1:~$
```

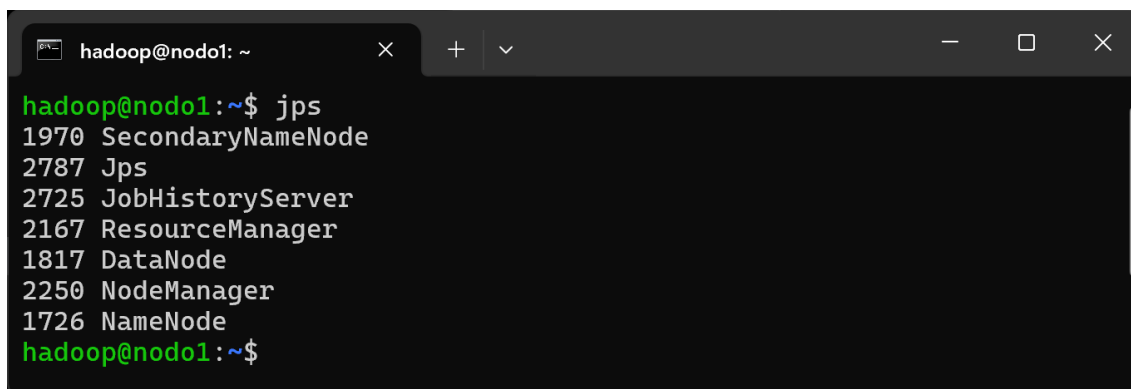
En esta pantalla, el comando `start-dfs.sh` se ha ejecutado para iniciar los servicios del sistema de archivos distribuido de Hadoop (HDFS), como los namenodes, datanodes y secondary namenodes.

```
hadoop@nodo1: ~  
hadoop@nodo1:~$ start-yarn.sh  
Starting resourcemanager  
Starting nodemanagers  
hadoop@nodo1:~$
```

En esta pantalla, el comando `start-yarn.sh` se ha ejecutado para iniciar los servicios de YARN (Yet Another Resource Negotiator) de Hadoop, como el resourcemanager y los nodemanagers.

```
hadoop@nodo1: ~  
hadoop@nodo1:~$ mapred --daemon start historyserver  
hadoop@nodo1:~$
```

En esta pantalla, se ejecuta el comando `mapred --daemon start historyserver` y, tras pulsar la tecla Intro.

A terminal window titled 'hadoop@nodo1: ~' showing the output of the 'jps' command. The output lists several Hadoop processes and their IDs: 1970 SecondaryNameNode, 2787 Jps, 2725 JobHistoryServer, 2167 ResourceManager, 1817 DataNode, 2250 NodeManager, and 1726 NameNode. The prompt 'hadoop@nodo1:~\$' is visible at the bottom.

```
hadoop@nodo1:~$ jps
1970 SecondaryNameNode
2787 Jps
2725 JobHistoryServer
2167 ResourceManager
1817 DataNode
2250 NodeManager
1726 NameNode
hadoop@nodo1:~$
```

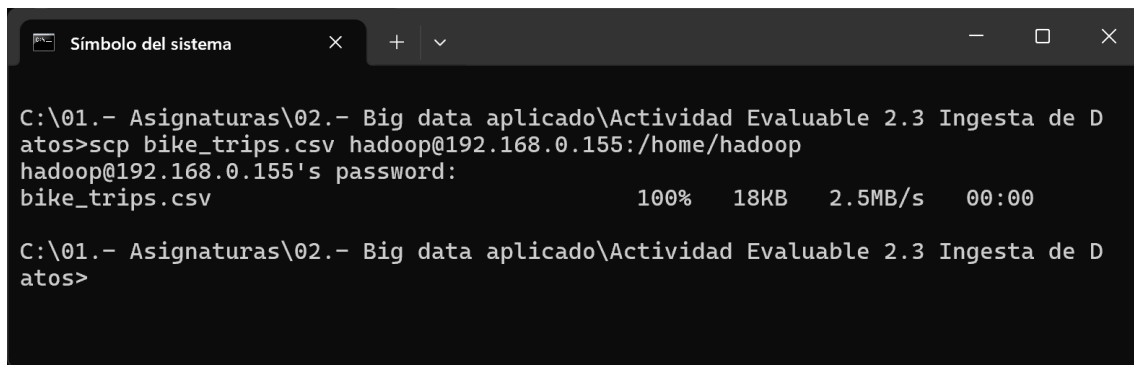
En esta pantalla, el comando `jps` se ha ejecutado para listar los procesos de Java en ejecución, mostrando que varios demonios de Hadoop como NameNode, DataNode, ResourceManager, NodeManager, SecondaryNameNode y JobHistoryServer están activos.

2.3.- COMUNICACIÓN ENTRE SISTEMA DE FICHEROS LOCAL Y HDFS

Uno de los objetivos principales de la actividad es trabajar con datos almacenados en HDFS. Para ello, es esencial conocer los comandos que permiten intercambiar archivos entre el sistema de ficheros local (Linux) y el distribuido (HDFS). Estos comandos se ejecutan con `hdfs dfs`.

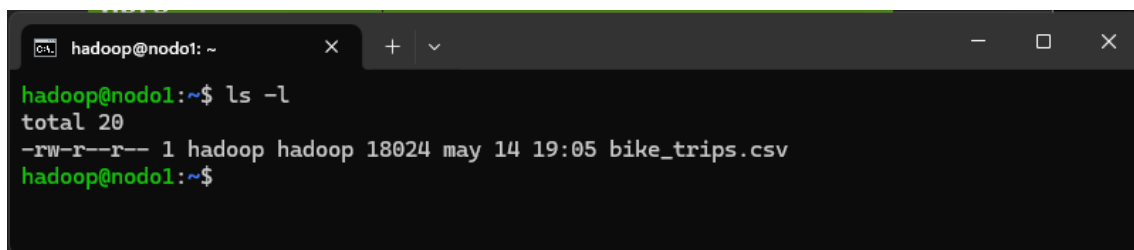
Comandos clave:

Acción	Comando
Subir archivo local a HDFS	<code>hdfs dfs -put archivo.csv /ruta/hdfs/</code>
Descargar archivo desde HDFS	<code>hdfs dfs -get /ruta/hdfs/archivo.csv ./</code>
Ver contenido de un directorio HDFS	<code>hdfs dfs -ls /ruta/</code>
Eliminar archivo/directorio de HDFS	<code>hdfs dfs -rm -r /ruta/</code>



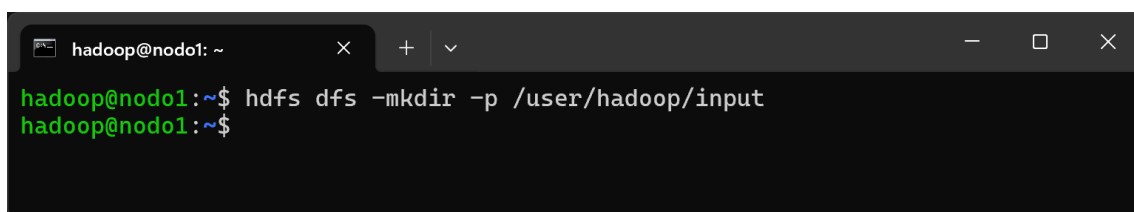
```
Símbolo del sistema x + v
C:\01.- Asignaturas\02.- Big data aplicado\Actividad Evaluable 2.3 Ingesta de D
atos>scp bike_trips.csv hadoop@192.168.0.155:/home/hadoop
hadoop@192.168.0.155's password:
bike_trips.csv 100% 18KB 2.5MB/s 00:00
C:\01.- Asignaturas\02.- Big data aplicado\Actividad Evaluable 2.3 Ingesta de D
atos>
```

En esta pantalla, se ejecuta el comando `scp bike_trips.csv hadoop@192.168.0.155:/home/hadoop` y, tras pulsar la tecla Intro, se transfiere el archivo desde el ordenador local a la máquina virtual Hadoop.



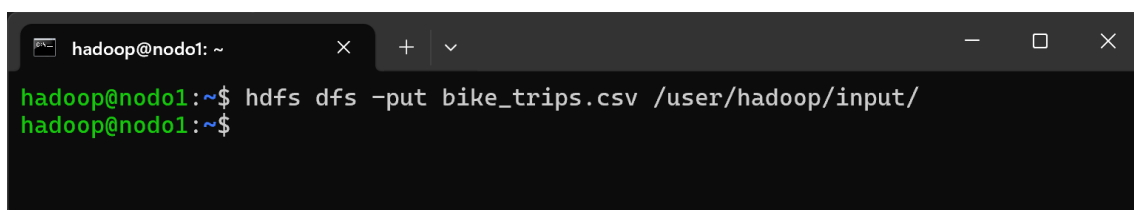
```
hadoop@nodo1: ~ x + v
hadoop@nodo1:~$ ls -l
total 20
-rw-r--r-- 1 hadoop hadoop 18024 may 14 19:05 bike_trips.csv
hadoop@nodo1:~$
```

En esta pantalla, se ejecuta el comando `ls -l` en el directorio del usuario hadoop para verificar que el archivo `bike_trips.csv` se haya transferido correctamente, pulsando la tecla Intro para ejecutar el comando.



```
hadoop@nodo1: ~ x + v
hadoop@nodo1:~$ hdfs dfs -mkdir -p /user/hadoop/input
hadoop@nodo1:~$
```

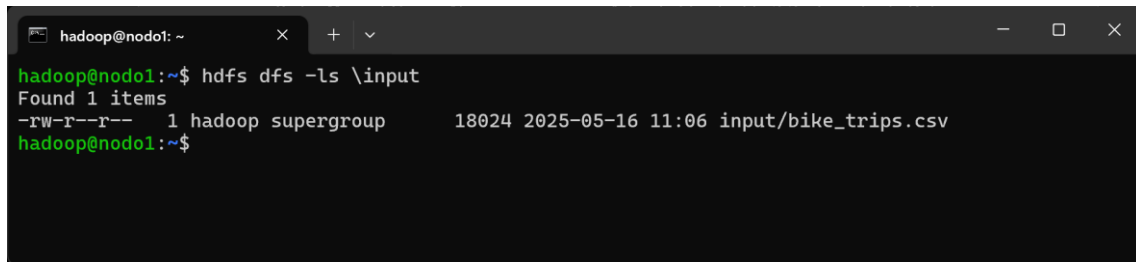
En esta pantalla, el comando `hdfs dfs -mkdir -p /user/hadoop/input` se emplea para crear el directorio `/user/hadoop/input` en el sistema de archivos HDFS, generando automáticamente los directorios padre necesarios si aún no existen.



```
hadoop@nodo1: ~ x + v
hadoop@nodo1:~$ hdfs dfs -put bike_trips.csv /user/hadoop/input/
hadoop@nodo1:~$
```

En esta pantalla, el comando `hdfs dfs -put bike_trips.csv /user/hadoop/input/` se utiliza para copiar el archivo local

bike_trips.csv al directorio /user/hadoop/input/ dentro del sistema de archivos HDFS.

A terminal window titled 'hadoop@nodo1: ~' with standard window controls. The terminal shows the command 'hdfs dfs -ls \input' being executed. The output is 'Found 1 items' followed by a line of file details: '-rw-r--r-- 1 hadoop supergroup 18024 2025-05-16 11:06 input/bike_trips.csv'. The prompt 'hadoop@nodo1:~\$' is visible at the bottom.

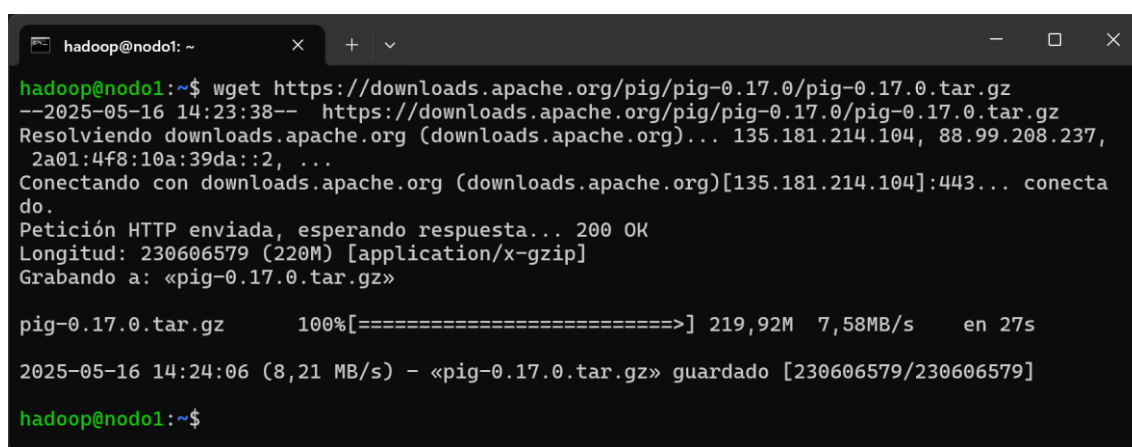
```
hadoop@nodo1:~$ hdfs dfs -ls \input
Found 1 items
-rw-r--r-- 1 hadoop supergroup      18024 2025-05-16 11:06 input/bike_trips.csv
hadoop@nodo1:~$
```

En esta pantalla, verificamos si el archivo se copió correctamente ejecutando el comando `hdfs dfs -ls \input` y presionando la tecla Intro.

3.- EJERCICIO 1 – INSTALACIÓN DE APACHE PIG

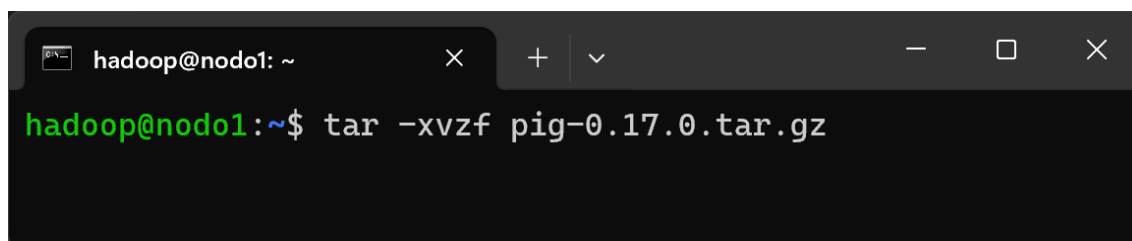
Este apartado describe los pasos necesarios para instalar y configurar correctamente **Apache Pig**, una herramienta de procesamiento de datos que permite ejecutar scripts en el sistema Hadoop mediante el lenguaje **Pig Latin**, facilitando tareas de transformación e ingesta.

3.1.- DESCARGA Y EXTRACCIÓN DE PIG



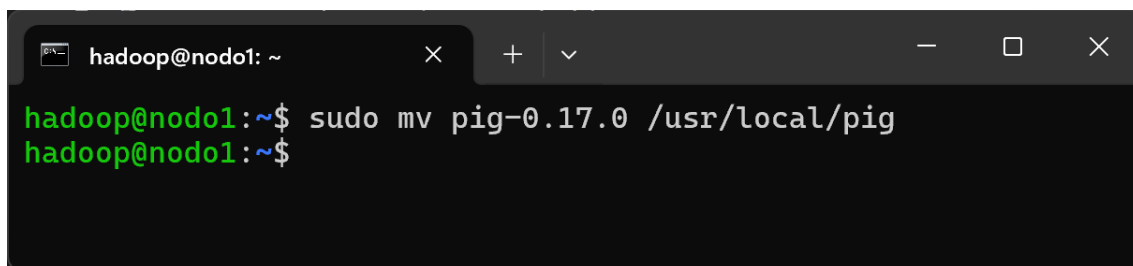
```
hadoop@nodo1: ~  
hadoop@nodo1:~$ wget https://downloads.apache.org/pig/pig-0.17.0/pig-0.17.0.tar.gz  
--2025-05-16 14:23:38-- https://downloads.apache.org/pig/pig-0.17.0/pig-0.17.0.tar.gz  
Resolviendo downloads.apache.org (downloads.apache.org)... 135.181.214.104, 88.99.208.237,  
2a01:4f8:10a:39da::2, ...  
Conectando con downloads.apache.org (downloads.apache.org)[135.181.214.104]:443... conecta  
do.  
Petición HTTP enviada, esperando respuesta... 200 OK  
Longitud: 230606579 (220M) [application/x-gzip]  
Grabando a: «pig-0.17.0.tar.gz»  
  
pig-0.17.0.tar.gz      100%[=====] 219,92M  7,58MB/s   en 27s  
2025-05-16 14:24:06 (8,21 MB/s) - «pig-0.17.0.tar.gz» guardado [230606579/230606579]  
hadoop@nodo1:~$
```

En esta pantalla, el comando **wget https://downloads.apache.org/pig/pig-0.17.0/pig-0.17.0.tar.gz** se está utilizando para descargar el archivo pig-0.17.0.tar.gz desde la URL especificada, mostrando el progreso y la finalización exitosa de la descarga.



```
hadoop@nodo1: ~  
hadoop@nodo1:~$ tar -xvzf pig-0.17.0.tar.gz
```

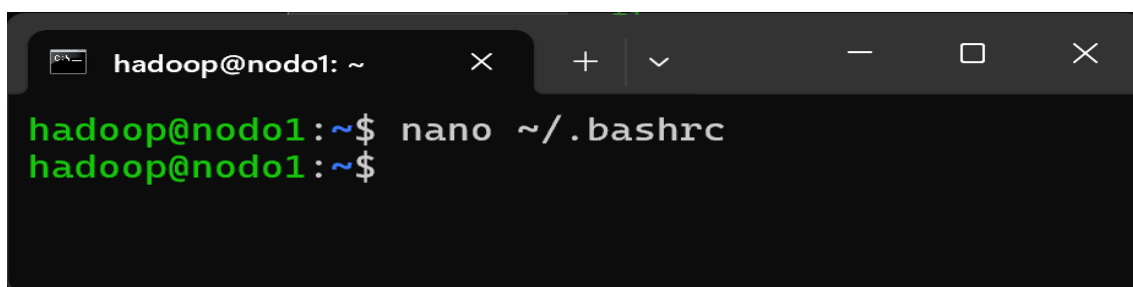
En esta pantalla, el comando **tar -xvzf pig-0.17.0.tar.gz** se está utilizando para extraer el contenido del archivo comprimido pig-0.17.0.tar.gz.



```
hadoop@nodo1: ~  
hadoop@nodo1:~$ sudo mv pig-0.17.0 /usr/local/pig  
hadoop@nodo1:~$
```

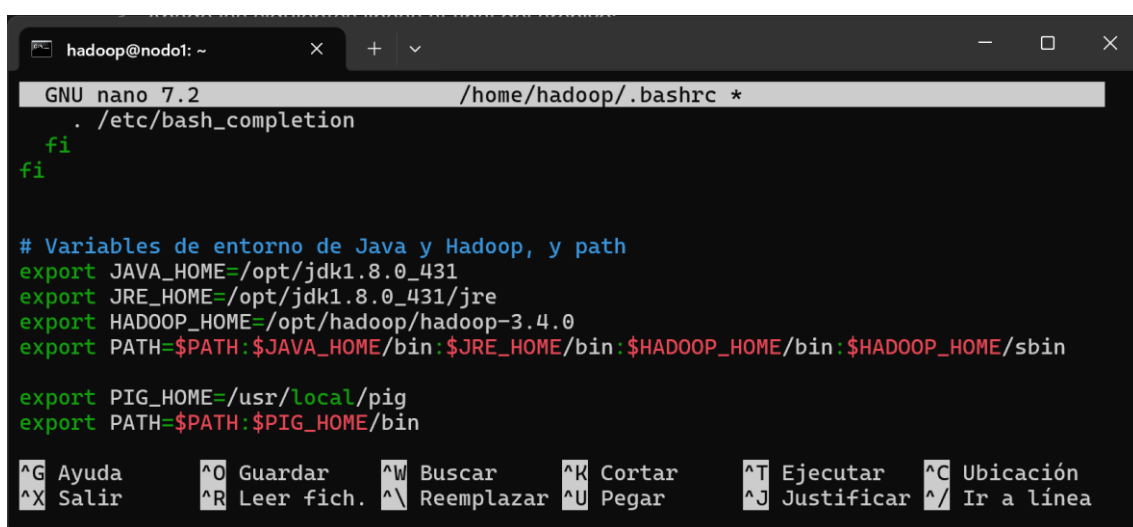
En esta pantalla, el comando `sudo mv pig-0.17.0 /usr/local/pig` se utiliza para mover el directorio `pig-0.17.0` (probablemente el directorio extraído en el paso anterior) a la ubicación `/usr/local/pig` con privilegios de superusuario.

3.2.- CONFIGURACIÓN DEL ENTORNO (.BASHRC)



```
hadoop@nodo1: ~  
hadoop@nodo1:~$ nano ~/.bashrc  
hadoop@nodo1:~$
```

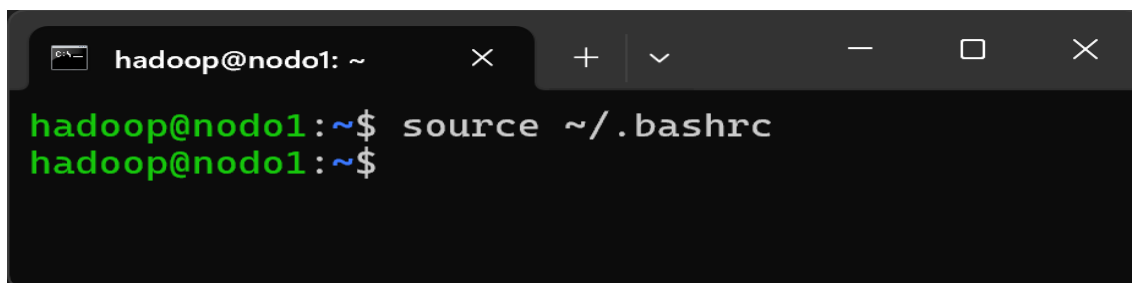
En esta pantalla, el comando `nano ~/.bashrc` se utiliza para abrir y editar el archivo de configuración de la terminal Bash `.bashrc` ubicado en el directorio principal del usuario, utilizando el editor de texto `nano`.



```
GNU nano 7.2 /home/hadoop/.bashrc *  
. /etc/bash_completion  
fi  
fi  
  
# Variables de entorno de Java y Hadoop, y path  
export JAVA_HOME=/opt/jdk1.8.0_431  
export JRE_HOME=/opt/jdk1.8.0_431/jre  
export HADOOP_HOME=/opt/hadoop/hadoop-3.4.0  
export PATH=$PATH:$JAVA_HOME/bin:$JRE_HOME/bin:$HADOOP_HOME/bin:$HADOOP_HOME/sbin  
  
export PIG_HOME=/usr/local/pig  
export PATH=$PATH:$PIG_HOME/bin  
  
^G Ayuda      ^O Guardar    ^W Buscar     ^K Cortar     ^T Ejecutar   ^C Ubicación  
^X Salir      ^R Leer fich. ^\ Reemplazar  ^U Pegar      ^J Justificar ^_ Ir a línea
```

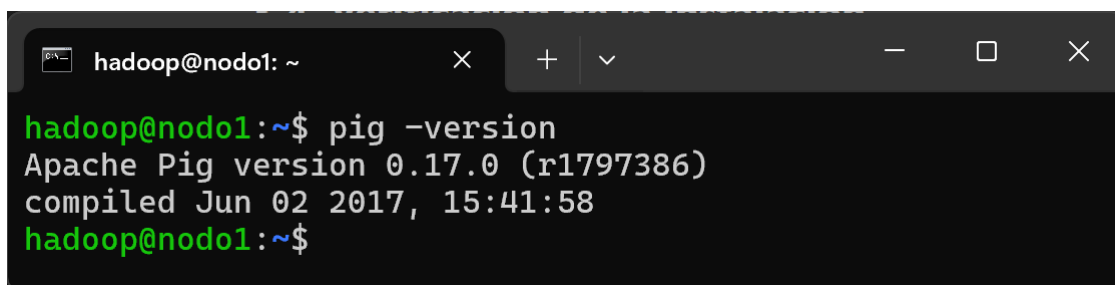
Para que el sistema reconozca los comandos de Pig desde cualquier ubicación, es necesario agregar al final del archivo `.bashrc`

del usuario `hadoop` las líneas `export PIG_HOME=/usr/local/pig` y `export PATH=$PATH:$PIG_HOME/bin`.



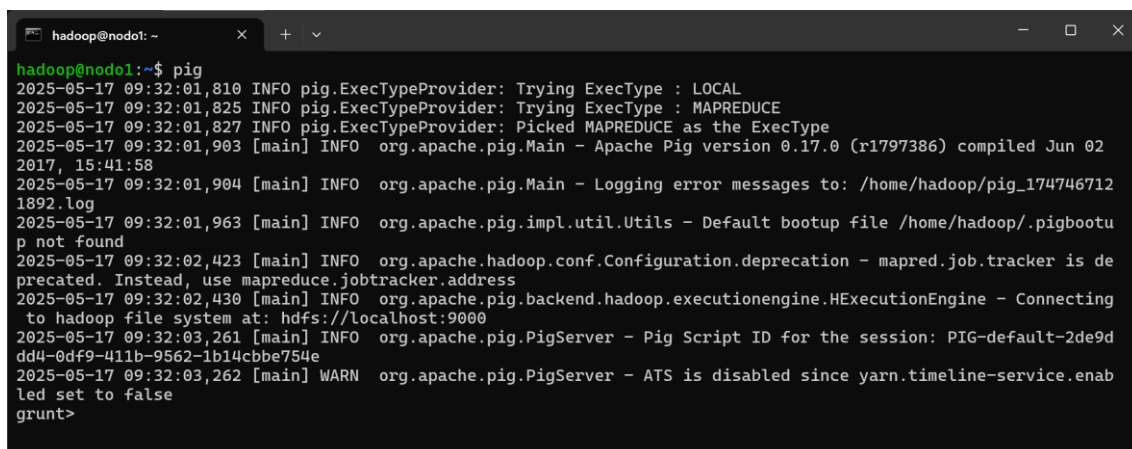
```
hadoop@nodo1: ~  
hadoop@nodo1:~$ source ~/.bashrc  
hadoop@nodo1:~$
```

En esta pantalla, el comando `source ~/.bashrc` se utiliza para recargar el archivo de configuración de la terminal Bash `.bashrc` en la sesión actual, aplicando inmediatamente cualquier cambio que se haya realizado en él.



```
hadoop@nodo1:~$ pig -version  
Apache Pig version 0.17.0 (r1797386)  
compiled Jun 02 2017, 15:41:58  
hadoop@nodo1:~$
```

En esta pantalla, el comando `pig -version` se ha ejecutado para mostrar la versión instalada de Apache Pig, que es la 0.17.0.



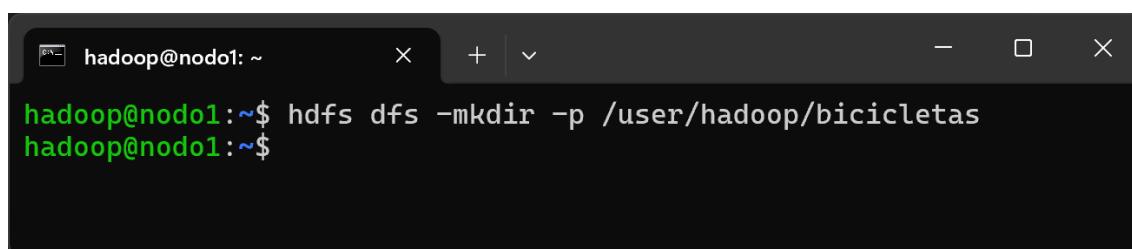
```
hadoop@nodo1:~$ pig  
2025-05-17 09:32:01,810 INFO pig.ExecTypeProvider: Trying ExecType : LOCAL  
2025-05-17 09:32:01,825 INFO pig.ExecTypeProvider: Trying ExecType : MAPREDUCE  
2025-05-17 09:32:01,827 INFO pig.ExecTypeProvider: Picked MAPREDUCE as the ExecType  
2025-05-17 09:32:01,903 [main] INFO org.apache.pig.Main - Apache Pig version 0.17.0 (r1797386) compiled Jun 02 2017, 15:41:58  
2025-05-17 09:32:01,904 [main] INFO org.apache.pig.Main - Logging error messages to: /home/hadoop/pig_1747467121892.log  
2025-05-17 09:32:01,963 [main] INFO org.apache.pig.impl.util.Utils - Default bootup file /home/hadoop/.pigbootup not found  
2025-05-17 09:32:02,423 [main] INFO org.apache.hadoop.conf.Configuration.deprecation - mapred.job.tracker is deprecated. Instead, use mapreduce.jobtracker.address  
2025-05-17 09:32:02,430 [main] INFO org.apache.pig.backend.hadoop.executionengine.HExecutionEngine - Connecting to hadoop file system at: hdfs://localhost:9000  
2025-05-17 09:32:03,261 [main] INFO org.apache.pig.PigServer - Pig Script ID for the session: PIG-default-2de9d4d4-0df9-411b-9562-1b14cbb754e  
2025-05-17 09:32:03,262 [main] WARN org.apache.pig.PigServer - ATS is disabled since yarn.timeline-service.enabled set to false  
grunt>
```

En esta pantalla, el comando `pig` se ha ejecutado para iniciar el shell interactivo Grunt de Apache Pig, el cual se ha configurado para usar el modo de ejecución MAPREDUCE y está listo para recibir comandos.

4.- EJERCICIO 2 – TRABAJO CON PIG Y PROCESAMIENTO DE DATOS

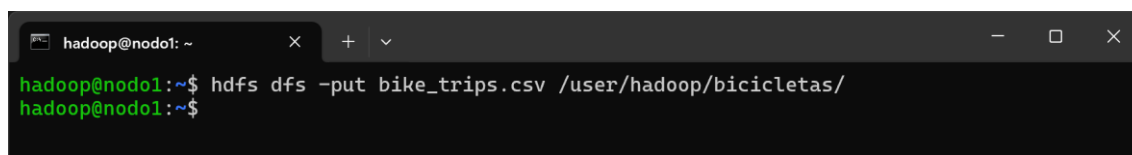
Este ejercicio tiene como objetivo aplicar Apache Pig sobre un dataset real, realizando consultas analíticas a través de scripts en Pig Latin. El conjunto de datos empleado es bike_trips.csv, que contiene información sobre trayectos en bicicleta realizados en una ciudad inteligente.

4.1.- CARGA DEL ARCHIVO BIKE_TRIPS.CSV EN HDFS



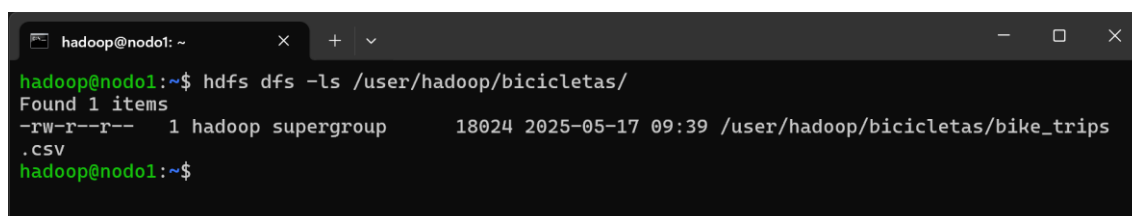
```
hadoop@nodo1: ~  
hadoop@nodo1:~$ hdfs dfs -mkdir -p /user/hadoop/bicicletas  
hadoop@nodo1:~$
```

En esta pantalla, el comando `hdfs dfs -mkdir -p /user/hadoop/bicicletas` se utiliza para crear el directorio `/user/hadoop/bicicletas` dentro del sistema de archivos HDFS, creando también el directorio padre si no existen.



```
hadoop@nodo1: ~  
hadoop@nodo1:~$ hdfs dfs -put bike_trips.csv /user/hadoop/bicicletas/  
hadoop@nodo1:~$
```

En esta pantalla, el comando `hdfs dfs -put bike_trips.csv /user/hadoop/bicicletas/` se utiliza para copiar el archivo local `bike_trips.csv` al directorio `/user/hadoop/bicicletas/` dentro del sistema de archivos HDFS.



```
hadoop@nodo1: ~  
hadoop@nodo1:~$ hdfs dfs -ls /user/hadoop/bicicletas/  
Found 1 items  
-rw-r--r-- 1 hadoop supergroup 18024 2025-05-17 09:39 /user/hadoop/bicicletas/bike_trips.csv  
hadoop@nodo1:~$
```

En esta pantalla, el comando `hdfs dfs -ls /user/hadoop/bicicletas/` se utiliza para listar el contenido del directorio `/user/hadoop/bicicletas/` en HDFS, mostrando que contiene el archivo `bike_trips.csv`.

4.2.- SCRIPT PIG 1: DURACIÓN MEDIA DE VIAJES (>15 MIN) POR TIPO DE USUARIO

El primer análisis busca calcular la **duración media de viajes superiores a 15 minutos**, agrupando los resultados por el **tipo de usuario** (por ejemplo, "Subscriber" o "Customer").

4.2.1.- Código del script

```
-- Cargar el archivo CSV ignorando la cabecera
raw = LOAD '/user/hadoop/bicicletas/bike_trips.csv'
      USING PigStorage(',')
      AS (trip_id:int, user_type:chararray, start_station:chararray, end_station:chararray, duration_minutes:int);

-- Eliminar cabecera (opcional, si no se eliminó antes de cargar)
data = FILTER raw BY trip_id != 0;

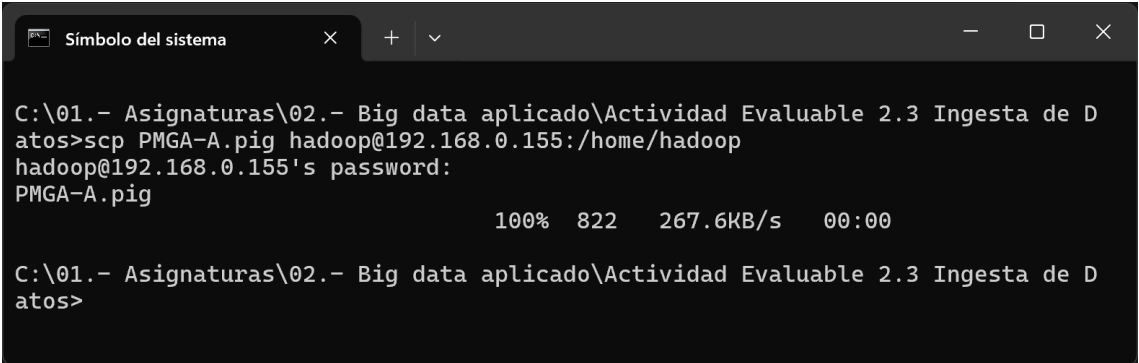
-- Filtrar viajes con duración mayor a 15 minutos
long_trips = FILTER data BY duration_minutes > 15;

-- Agrupar por tipo de usuario
grouped = GROUP long_trips BY user_type;

-- Calcular duración media
avg_duration = FOREACH grouped GENERATE
               group AS user_type,
               AVG(long_trips.duration_minutes) AS avg_duration;

-- Guardar resultados
STORE avg_duration INTO '/user/hadoop/resultados/PMGA-A' USING PigStorage(',');
```

Esta pantalla muestra un script de Pig Latin diseñado para cargar datos de viajes en bicicleta desde HDFS, filtrar aquellos con una duración superior a 15 minutos (900 segundos), agruparlos por tipo de usuario, calcular la duración media de estos viajes largos para cada grupo y, finalmente, almacenar los resultados en HDFS.



```
Símbolo del sistema  X  +  v

C:\01.- Asignaturas\02.- Big data aplicado\Actividad Evaluable 2.3 Ingesta de D
atos>scp PMGA-A.pig hadoop@192.168.0.155:/home/hadoop
hadoop@192.168.0.155's password:
PMGA-A.pig
                                     100% 822  267.6KB/s   00:00

C:\01.- Asignaturas\02.- Big data aplicado\Actividad Evaluable 2.3 Ingesta de D
atos>
```

En esta pantalla, el comando **scp PMGA-A.pig hadoop@192.168.0.155:/home/hadoop** se ha utilizado para copiar de forma segura el archivo PMGA-A.pig desde la máquina local (Windows)

al directorio /home/hadoop del servidor remoto con la dirección IP 192.168.0.155, utilizando el usuario hadoop.

4.2.2.- Ejecución y salida (PMGA-A)

```
hadoop@nodo1: ~  
hadoop@nodo1:~$ pig PMGA-A.pig  
2025-05-17 10:53:50,345 INFO pig.ExecTypeProvider: Trying ExecType : LOCAL  
2025-05-17 10:53:50,359 INFO pig.ExecTypeProvider: Trying ExecType : MAPREDUCE  
2025-05-17 10:53:50,360 INFO pig.ExecTypeProvider: Picked MAPREDUCE as the ExecType  
2025-05-17 10:53:50,423 [main] INFO org.apache.pig.Main - Apache Pig version 0.17.0 (r1797386) compiled Jun 02 2017, 15:41:58  
2025-05-17 10:53:50,424 [main] INFO org.apache.pig.Main - Logging error messages to: /home/hadoop/pig_1747472030419.log  
2025-05-17 10:53:50,766 [main] INFO org.apache.pig.impl.util.Utils - Default bootup file /home/hadoop/.pigbootup not found  
2025-05-17 10:53:50,843 [main] INFO org.apache.hadoop.conf.Configuration.deprecation - mapred.job.tracker is deprecated. Instead, use mapreduce.jobtracker.address  
2025-05-17 10:53:50,844 [main] INFO org.apache.pig.backend.hadoop.executionengine.HExecutionEngine - Connecting to hadoop file system at: hdfs://localhost:9000  
2025-05-17 10:53:51,380 [main] INFO org.apache.pig.PigServer - Pig Script ID for the session: PIG-PMGA-A.pig-6dd1645e-68d8-4cf3-a8d2-08daa8ad1d20  
2025-05-17 10:53:51,381 [main] WARN org.apache.pig.PigServer - ATS is disabled since yarn.timeline-service.enabled set to false
```

En esta pantalla, se ejecuta el comando `pig PMGA-A.pig` para procesar dicho script con Apache Pig, el cual inicia e intenta utilizar el modo de ejecución MAPREDUCE.

```
hadoop@nodo1: ~  
hadoop@nodo1:~$ hdfs dfs -cat /user/hadoop/resultados/avg_duration_user/part-r-000000  
Customer,30.5752688172043  
Subscriber,30.310344827586206  
hadoop@nodo1:~$
```

En esta pantalla, el comando `hdfs dfs -cat /user/hadoop/resultados/avg_duration_user/part-r-000000` se utiliza para mostrar el contenido del archivo de resultados part-r-000000 en HDFS, el cual contiene la duración promedio de los viajes para los tipos de usuario "Customer" y "Subscriber".

4.3.- SCRIPT PIG 2: NÚMERO DE VIAJES POR ESTACIÓN DE SALIDA

4.3.1.- Código del script

```
-- Cargar los datos desde el archivo de entrada
datos = LOAD '/user/hadoop/bicicletas/bike_trips.csv'
        USING PigStorage(',')
        AS (id:chararray, start_station:chararray, end_station:chararray, user_type:chararray, ...); -- completa los campos reales

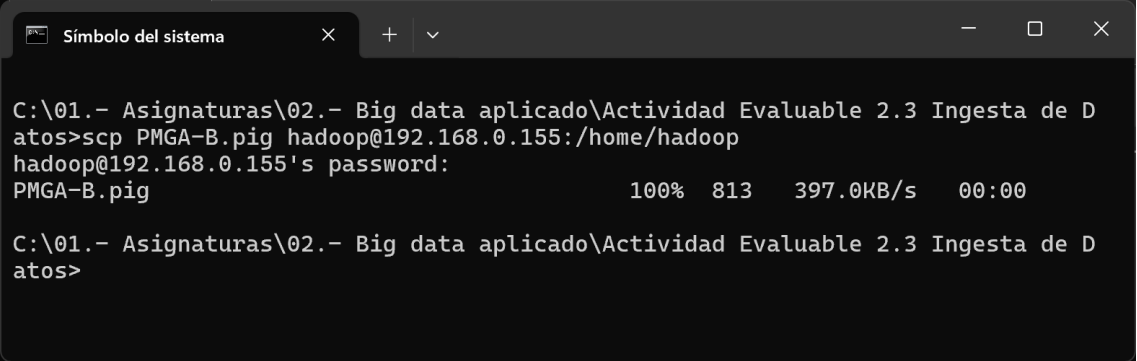
-- Filtrar registros con estación de origen no nula (opcional)
datos_filtrados = FILTER datos BY start_station IS NOT NULL;

-- Agrupar por estación de origen
agrupados = GROUP datos_filtrados BY start_station;

-- Contar cuántos viajes parten desde cada estación
cuenta_viajes = FOREACH agrupados GENERATE
                group AS estacion_origen,
                COUNT(datos_filtrados) AS total_viajes;

-- Almacenar el resultado en un directorio de salida
STORE cuenta_viajes INTO 'PMGA-B'
        USING PigStorage(',');
```

Esta pantalla muestra un script de Pig Latin que carga datos de viajes en bicicleta desde HDFS, opcionalmente filtra aquellos con estación de origen no nula, los agrupa por estación de origen, cuenta cuántos viajes parten desde cada estación y finalmente almacena estos conteos en un nuevo directorio en HDFS llamado 'PMGA-B'.



```
Símbolo del sistema
C:\01.- Asignaturas\02.- Big data aplicado\Actividad Evaluable 2.3 Ingesta de D
atos>scp PMGA-B.pig hadoop@192.168.0.155:/home/hadoop
hadoop@192.168.0.155's password:
PMGA-B.pig                                100% 813   397.0KB/s   00:00

C:\01.- Asignaturas\02.- Big data aplicado\Actividad Evaluable 2.3 Ingesta de D
atos>
```

En esta pantalla, el comando `scp PMGA-B.pig hadoop@192.168.0.155:/home/hadoop` se ha utilizado para copiar de forma segura el archivo de script Pig PMGA-B.pig desde la máquina local (Windows) al directorio /home/hadoop del servidor remoto con la dirección IP 192.168.0.155, utilizando el usuario hadoop.

4.3.2.- Ejecución y salida (PMGA-B)

```
hadoop@nodo1: ~  
hadoop@nodo1:~$ pig PMGA-B.pig  
2025-05-17 10:42:56,127 INFO pig.ExecTypeProvider: Trying ExecType : LOCAL  
2025-05-17 10:42:56,131 INFO pig.ExecTypeProvider: Trying ExecType : MAPREDUCE  
2025-05-17 10:42:56,132 INFO pig.ExecTypeProvider: Picked MAPREDUCE as the ExecType  
2025-05-17 10:42:56,197 [main] INFO org.apache.pig.Main - Apache Pig version 0.17.0 (r1797386) compiled Jun 02 2017, 15:41:58  
2025-05-17 10:42:56,198 [main] INFO org.apache.pig.Main - Logging error messages to: /home/hadoop/pig_1747471376190.log  
2025-05-17 10:42:56,539 [main] INFO org.apache.pig.impl.util.Utils - Default bootup file /home/hadoop/.pigbootup not found  
2025-05-17 10:42:56,617 [main] INFO org.apache.hadoop.conf.Configuration.deprecation - mapred.job.tracker is deprecated. Instead, use mapreduce.jobtracker.address  
2025-05-17 10:42:56,618 [main] INFO org.apache.pig.backend.hadoop.executionengine.HExecutionEngine - Connecting to hadoop file system at: hdfs://localhost:9000
```

En esta pantalla, el comando `pig PMGA-B.pig` se ha ejecutado para procesar el script de Pig `PMGA-B.pig`, el cual inicia Apache Pig e intenta utilizar el modo de ejecución MAPREDUCE, conectándose al sistema de archivos HDFS en `hdfs://localhost:9000`.

```
hadoop@nodo1: ~  
hadoop@nodo1:~$ hdfs dfs -cat /user/hadoop/resultados/PMGA-B/part-r-00000  
Sol,85  
Atocha,94  
Retiro,109  
Gran Vía,106  
Plaza España,106  
hadoop@nodo1:~$
```

En esta pantalla, el comando `hdfs dfs -cat /user/hadoop/resultados/PMGA-B/part-r-00000` se utiliza para mostrar el contenido del archivo de resultados `part-r-00000` en HDFS, el cual lista las estaciones de origen y el número de viajes que parten desde cada una.

5.- CONCLUSIONES

5.1.- VALORACIÓN DEL USO DE PIG PARA INGESTA Y ANÁLISIS

El uso de **Apache Pig** ha demostrado ser una herramienta eficaz para realizar tareas de **procesamiento de datos a gran escala** en el ecosistema Hadoop. Su lenguaje de alto nivel, **Pig Latin**, permite definir transformaciones de datos de forma declarativa, sin necesidad de programar explícitamente en Java o en MapReduce, lo cual **acelera el desarrollo y la legibilidad del código**.

En este trabajo, se ha podido realizar de manera sencilla la **ingesta de un archivo CSV** desde HDFS, la **eliminación de cabeceras**, la **agrupación y conteo de datos** según distintos criterios (estación de salida y año del viaje), así como el almacenamiento de resultados en directorios separados en HDFS. Esto demuestra que Pig es especialmente útil para tareas de **ETL (extracción, transformación y carga)** dentro de un pipeline de Big Data.

Sin embargo, aunque Pig resulta útil para procesos **batch**, también presenta limitaciones en comparación con herramientas más modernas como **Apache Spark**, especialmente en cuanto a rendimiento y flexibilidad para análisis avanzados o en tiempo real.

5.2.- DIFICULTADES ENCONTRADAS Y CÓMO SE RESOLVIERON

Durante el desarrollo del trabajo, se encontraron varias dificultades técnicas que fueron resueltas con distintas estrategias:

- **Error de sintaxis en Pig Latin:** al escribir el segundo script, se generó un error debido a un símbolo mal colocado (...). Se resolvió revisando cuidadosamente la sintaxis del script y eliminando caracteres incorrectos o innecesarios.
- **Archivos no ubicados correctamente en HDFS:** al intentar ejecutar los scripts, Pig no encontraba los archivos de entrada. Se solucionó asegurando que los archivos estuvieran **correctamente subidos a HDFS** mediante el comando **hdfs dfs -put**, y verificando sus rutas con **hdfs dfs -ls**.
- **Problemas con los nombres de las rutas de salida:** en algunos casos, Pig arrojaba errores al sobrescribir directorios

existentes. Se solucionó **eliminando previamente el directorio de salida** con `hdfs dfs -rm -r` antes de ejecutar de nuevo el script.

Estas dificultades reforzaron la comprensión de cómo interactúan las distintas herramientas dentro del ecosistema Hadoop (Pig, HDFS, YARN) y la importancia de cuidar **la sintaxis, rutas y permisos**.

5.3.- POSIBLES MEJORAS O AMPLIACIONES DEL TRABAJO

Este trabajo podría ampliarse o mejorarse en varias direcciones:

- **Normalización de datos:** realizar una limpieza más profunda de los datos de entrada, eliminando posibles inconsistencias, valores nulos o duplicados, antes de procesarlos con Pig.
- **Enriquecimiento del análisis:** incluir más variables en el análisis, como el tipo de usuario (`user_type`) o la duración del viaje (`duration_minutes`), lo que permitiría generar estadísticas más ricas (e.g., duración media por estación).
- **Automatización de la ejecución:** crear scripts Bash o workflows en Oozie que ejecuten automáticamente los scripts Pig como parte de un pipeline de procesamiento diario.
- **Comparación con otras herramientas:** replicar el mismo análisis usando Apache Hive o Spark para evaluar diferencias en complejidad, tiempo de ejecución y rendimiento.

Estas posibles mejoras ayudarían a escalar el análisis y hacerlo más útil en un entorno de datos en producción.

6.- MAPA MENTAL DEL TRABAJO



El mapa mental visualiza el proceso completo, desde la configuración del entorno Hadoop y la instalación de Apache Pig, hasta la ingesta y el procesamiento de datos de viajes en bicicleta mediante scripts PigLatin en HDFS para generar análisis sobre la duración de los viajes y el conteo por estación.

7.- ANEXO: FICHERO DE ENTREGABLES

Este anexo detalla los directorios y ficheros clave generados, utilizados y configurados como parte de la actividad de ingesta y procesamiento de datos con Hadoop y Pig.

7.1.- DIRECTORIO: PRINCIPAL

El Archivo [Actividad Evaluable 2.3.- Ingesta de datos.pdf](#) es el documento principal de la actividad evaluable, presumiblemente el informe o la solución escrita del ejercicio.

El Archivo [Mapa Mental Actividad Evaluable 2.3 Ingesta de Datos con Apache Pig.pdf](#) es el documento que contiene la representación visual y esquemática del flujo de trabajo y los componentes de la actividad.

7.2.- DIRECTORIO: DATASET

El Archivo [bike_trips.csv](#) contiene CSV original con los datos de viajes en bicicleta, utilizado como fuente para la ingesta en HDFS.

7.3.- DIRECTORIO: ENUNCIADO

El archivo [Actividad Evaluable 2.3.pdf](#) documento PDF con el enunciado y las especificaciones de la Actividad Evaluable 2.3.

7.4.- DIRECTORIO: SCRIPT

El archivo [PMGA-A.pig](#) contiene el Script Pig Latín para el análisis de datos de viajes, enfocado en calcular la duración media de viaje por tipo de usuario.

El archivo [PMGA-B.pig](#) contiene el Script Pig Latín para el análisis de datos de viajes, enfocada en contar el número total de viajes que parten de cada estación de origen.

ÍNDICE ALFABÉTICO

A

acceso..... 2, 5
 actividad 2, 3, 4, 5, 9, 23
 activos 6, 9
 actual..... 14, 17
 address 7
 almacenamiento 4, 20
 almacenar 3, 16
 alto 3, 4, 20
 análisis..... 2, 16, 20, 21, 22, 23
 anexo..... 23
 apache 12
 apartado..... 5, 12
 aquellos..... 16, 18
 archivo 2, 3, 5, 6, 9, 10, 11, 12, 13, 14, 15, 16,
 17, 18, 19, 20, 23
 archivos 5, 8, 9, 10, 11, 15, 17, 19, 20
 automáticamente..... 10, 21

B

bashrc..... 2, 13, 14
 bicicleta 15, 16, 18, 22, 23
 bicicletas 15

C

cada 3, 16, 18, 19, 23
 calcular 16, 23
 carga 18, 20
 cat..... 17, 19
 clave..... 3, 9, 23
 código 20
 comando .5, 6, 8, 9, 10, 11, 12, 13, 14, 15, 16,
 17, 18, 19, 20
 comandos..... 3, 4, 9, 13, 14
 cómo 2, 5, 20, 21
 comparación 20
 comunicación 5
 configuración 2, 3, 5, 6, 13, 14, 22
 configurado..... 5, 14
 configurar 3, 5, 12
 conjunto 3, 15
 contenido 9, 12, 15, 17, 19
 conteo 20, 22
 contiene 15, 17, 23
 copiar 10, 15, 16, 18
 correctamente..... 3, 5, 10, 11, 12, 20
 crear 4, 10, 15, 21
 csv 2, 9, 10, 15, 23
 cualquier 3, 13, 14

D

dataset..... 15, 23
 datos . 2, 3, 4, 9, 12, 15, 16, 18, 20, 21, 22, 23
 dentro..... 3, 11, 15, 20, 21

desarrollo 4, 20
 descarga 12
 descargar 12
 describe 5, 12
 dfs..... 8, 9, 10, 11, 15, 17, 19, 20, 21
 dificultades 20, 21
 dirección..... 17, 18
 directorio..... 9, 10, 11, 13, 15, 17, 18, 21
 directorios 3, 10, 20, 23
 distintas 20, 21
 distribuido 3, 4, 8, 9
 documento 23
 duración 2, 16, 17, 21, 22, 23

E

ecosistema 4, 5, 6, 20, 21
 editar 6, 13
 editor 7, 13
 ejecución..... 3, 4, 5, 6, 9, 14, 17, 19, 21
 ejecuta 6, 8, 10, 17
 ejecutado 8, 9, 14, 19
 ejecutar..... 4, 6, 10, 12, 20, 21
 ejercicio 15, 23
 eliminando..... 20, 21
 ello 3, 9
 entorno 2, 3, 4, 5, 13, 21, 22
 entrada 20, 21
 enunciado..... 23
 error 20
 escribir..... 7, 20
 esenciales..... 3, 5
 especialmente 20
 estación 2, 18, 20, 21, 22, 23
 evaluable..... 3, 23
 existen 10, 15
 export..... 13
 extracción..... 2, 12, 20

F

ficheros 2, 3, 4, 9, 23
 finalmente 16, 18
 forma 16, 18, 20
 framework 5

G

generar 21, 22

H

hadoop..... 3, 4, 5, 6, 10, 13, 15, 16, 17, 18, 19
 hdfs 9, 10, 11, 15, 17, 19, 20, 21
 herramienta 3, 12, 20
 herramientas 3, 4, 20, 21
 home 4, 10, 16, 18
 host 4, 5

I

importada 3, 5
 ingesta 2, 3, 12, 20, 22, 23
 inicia 17, 19
 iniciar 8, 14
 input 10, 11
 instalación 22
 instalar 3, 12
 intenta 17, 19
 interacción 3

J

jobhistory 7
 jps 6, 9

L

lenguaje 3, 12, 20
 listar 9, 15, 17

LI

llamado 5, 18

L

local 2, 3, 5, 9, 10, 13, 15, 16, 18
 localhost 7, 19
 luego 7, 17

M

mapa 22
 mapred 6, 7, 8
 mapreduce 7
 máquina 3, 4, 5, 10, 16, 18
 media 2, 16, 21, 23
 mediante 3, 5, 12, 20, 22
 mejoras 2, 21
 mental 22
 minutos 16
 mkdir 10, 15
 modo 3, 6, 14, 17, 19
 mostrando 9, 12, 15, 17
 mostrar 14, 17, 19
 muestra 5, 16, 18

N

name 7
 namenodes 8
 nano 6, 7, 13
 necesario 3, 6, 13
 necesarios 10, 12
 nivel 3, 4, 20
 nuevo 18, 21
 número 2, 18, 19, 23

O

objetivo 3, 15
 objetivos 9
 operativo 3, 4, 5
 origen 18, 19, 23
 ova 3, 5

P

padre 10, 15
 pantalla 3, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19
 part 17, 19
 parte 21, 23
 parten 18, 19, 23
 paso 3, 13
 pasos 5, 12
 pdf 23
 permite 5, 12, 20
 pig 12, 13, 14, 16, 17, 18, 19, 23
 pipeline 20, 21
 posibles 21
 preconfigurada 3, 5
 principal 13, 23
 procesamiento 2, 3, 12, 15, 20, 21, 22, 23
 procesar 3, 17, 19
 proceso 3, 22
 procesos 9, 20
 property 7
 puertos 4, 5
 pulsar 8, 10
 put 9, 10, 15, 20

R

real 15, 20
 realizado 3, 14
 realizar 3, 20, 21
 remoto 17, 18
 rendimiento 20, 21
 resourcemanager 8
 resultados 3, 16, 17, 19, 20
 ruta 9
 rutas 20, 21

S

salida 2, 17, 18, 19, 20
 scp 10, 16, 18
 script 2, 16, 17, 18, 19, 20, 21, 23
 scripts 3, 4, 5, 6, 12, 15, 20, 21, 22
 segura 16, 18
 ser 3, 20
 servicios 2, 3, 5, 6, 8
 servidor 5, 17, 18
 shell 14
 siguiente 5, 7
 siguientes 4, 6
 sintaxis 20, 21
 sistema 2, 3, 4, 5, 8, 9, 10, 11, 12, 13, 15, 19
 solucionó 20, 21
 sql 16, 17

ssh5
 start8
 sudo 6, 13

T

tar12
 tareas 3, 4, 12, 20
 tecla8, 10, 11
 terminal5, 13, 14
 tiempo 20, 21
 tipo 2, 16, 21, 23
 trabajo 2, 3, 4, 5, 20, 21, 23
 transformación3, 12, 20
 tras..... 8, 10
 través 3, 15

U

ubicación 13
 usando 4, 21
 user 10, 15, 17, 19, 21
 uso 2, 3, 20
 usr..... 13
 usuario..... 2, 5, 10, 13, 16, 17, 18, 21, 23

útil..... 20, 21
 utiliza.....7, 10, 13, 14, 15, 17, 19
 utilizadas..... 2, 3, 4
 utilizado4, 5, 6, 16, 18, 23
 utilizando 12, 13, 17, 18
 utilizar3, 17, 19

V

value.....7
 variables 3, 21
 varias..... 20, 21
 verificar..... 5, 6, 10
 viaje 20, 21, 23
 viajes.....2, 16, 17, 18, 19, 22, 23
 virtual 2, 3, 4, 5, 10

W

web5

Y

yarn.....8