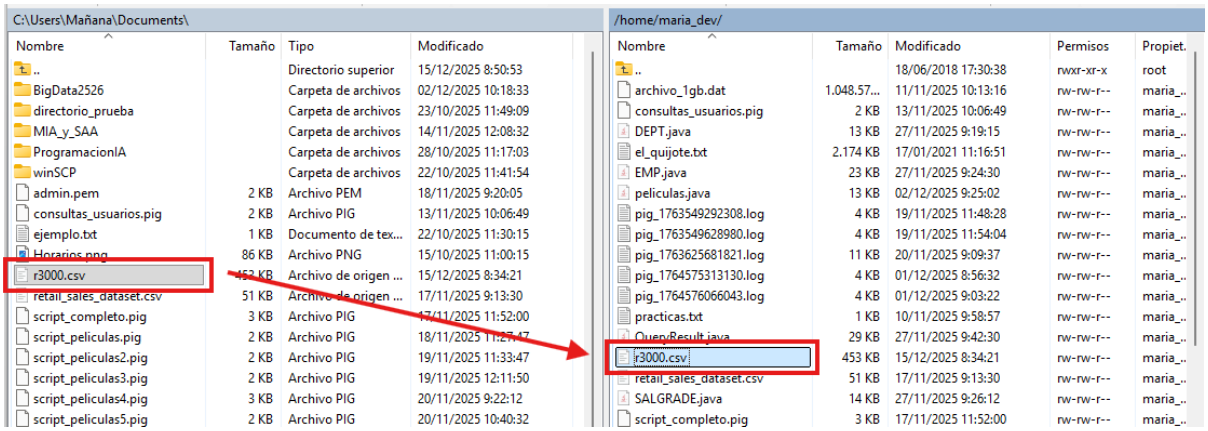
PR_06.4 Dani Gayol Rodríguez

Dataset de Práctica

Primero de todo, tenemos que subir el archivo ".csv" a la máquina virtual, para ello, voy a utilizar "WinSCP"



Ahora voy a pasar el archivo desde el directorio "/home" hacia el HDFS

```
[maria_dev@sandbox-hdp ~]$ hdfs dfs -put /home/maria_dev/r3000.csv /user/maria_dev/
[maria_dev@sandbox-hdp ~]$ hdfs dfs -ls /user/maria_dev/r3*
-rw-r--r--   1 maria_dev hdfs     463816 2025-12-15 07:54 /user/maria_dev/r3000.csv
```

Ejercicio 1: Importa el fichero csv en una tabla HIVE. Has de saltarte la primera fila con el nombre de las columnas

Antes de importar el fichero, hay que crear la base de datos junto con las tablas

```
CREATE DATABASE r3000
```

Ahora creamos la tabla para importarle los datos del archivo csv

```
CREATE TABLE reviews (
    id STRING,
    producto STRING,
    resena STRING
)
ROW FORMAT DELIMITED
FIELDS TERMINATED BY '|'
STORED AS TEXTFILE
TBLPROPERTIES ("skip.header.line.count"="1");
```

Una vez creada la tabla, importamos los datos

```
LOAD DATA INPATH '/user/maria_dev/r3000.csv' INTO TABLE reviews;
```

Finalmente, comprobamos que se cargaron correctamente los datos

reviews.id	reviews.producto	reviews.resena
106	Tablet	"El material se siente barato. La calidad es excelente."
107	Laptop	"El material se siente barato. El diseño es elegante y moderno."
108	Auriculares	"El diseño es elegante y moderno. Las instrucciones no son claras."
109	Monitor 4K	"Las instrucciones no son claras. Superó mis expectativas."
110	Smartwatch	"No cumple con lo prometido. Funciona perfectamente."
111	Ratón Inalámbrico	"Las instrucciones no son claras. Superó mis expectativas."
112	Laptop	"La configuración inicial fue un poco complicada. Muy satisfecho con la compra."

Ejercicio 2: Utilizando SENTENCES, extrae en un array las frases que componen cada reseña en otra tabla (por ejemplo, usando CTAS).

```
1 CREATE TABLE reviews_sentences
2 AS
3 SELECT
4     id,
5     producto,
6     sentences(resena) AS frases
7 FROM reviews;
```

Comprobamos que se hizo correctamente

reviews_sentences.id	reviews_sentences.producto	reviews_sentences.frases
106	Tablet	[["El","material","se","siente","barato"],["La","calidad","es","excelente"]]
107	Laptop	[["El","material","se","siente","barato"],["El","diseño","es","elegante","y","moderno"]]
108	Auriculares	[["El","diseño","es","elegante","y","moderno"],["Las","instrucciones","no","son","claras"]]

Ejercicio 3: Utilizando EXPLODE, aplana la estructura, de modo que cada fila contenga una única frase. Crea una nueva tabla a partir de esta consulta.

```sql
CREATE TABLE reviews_aplanar_sentences
AS
SELECT
    id,
    producto,
    concat_ws(' ', frase) AS frase
FROM reviews_sentences
LATERAL VIEW explode(frases) t AS frase;
```

Comprobamos que se hizo correctamente

reviews_aplanar_sentences.id	reviews_aplanar_sentences.producto	reviews_aplanar_sentences.frase
106	Tablet	El material se siente barato
106	Tablet	La calidad es excelente
107	Laptop	El material se siente barato
107	Laptop	El diseño es elegante y moderno
108	Auriculares	El diseño es elegante y moderno

Ejercicio 4: (INVESTIGA) A partir de la tabla anterior, ¿cómo podrías crear otra tabla que elimine de las frases las siguientes palabras?

```sql
CREATE TABLE reviews_borrar_sentences
AS
SELECT
    id,
    producto,
    trim(
      regexp_replace(
        lower(frase),
        '\\b(el|la|los|las|de|del|al|a|un|una|unos|unas|que|y|o|en|por|para|
        ''
      )
    ) AS frase_limpia
FROM reviews_aplanar_sentences;
```

Comprobamos que se hizo correctamente

reviews_borrar_sentences.id	reviews_borrar_sentences.producto	reviews_borrar_sentences.frase_limpia
106	Tablet	material siente barato
106	Tablet	calidad es excelente
107	Laptop	material siente barato
107	Laptop	diseño es elegante moderno
108	Auriculares	diseño es elegante moderno

Ejercicio 5: Identificación de Frases Clave (Ngramas) Utilizando la tabla del ejercicio anterior, desarrolla una consulta que identifique los trigramas (n=3) más frecuentes en todas las reseñas de clientes.

```sql
SELECT
    ng.ngram       AS ngram,
    ng.estfrequency AS estfrequency
FROM (
    SELECT
        ngrams(split(frase_limpia, ' '), 3, 1000) AS ng
    FROM reviews_borrar_sentences
) t
ORDER BY estfrequency DESC;
```

Comprobamos que se hizo correctamente

trigram	frecuencia
[["envío","fue","lento"],["","",""],["hace","",""], ["","útil",""],["es","","útil"],["producto","es",""], ["útil","","rápido"],["","poco","complicada"], ["configuración","inicial","fue"], ["fue","","poco"],["inicial","fue",""], ["","","compra"],["calidad","es","excelente"], ["","siente","barato"],["material","","siente"], ["","oficina","ergonómica"],	[294.0,223.0,196.0,156.0,156.0,156.0,156.0,155.0,155.0,155.0,155.0,152.0

Ejercicio 6: Análisis de Sentimiento Contextualizado Vistas las palabras que aparecen con más frecuencia en los trigramas, como: envío, rendimiento, diseño, producto, etc.

```sql
SELECT
    ctx.ngram            AS contexto,
    ROUND(ctx.estfrequency, 0) AS frecuencia
FROM (
    SELECT
        context_ngrams(
            split(frase_limpia, ' '),
            array('envío'),
            3,
            1000
        ) AS ctx
    FROM reviews_borrar_sentences
) t
ORDER BY frecuencia DESC
LIMIT 20;
```

Ejercicio 7: Análisis de Sentimiento Contextual por nombre del Producto Modifica la consulta anterior de CONTEXT_NGRAMS para que, además de calcular la frecuencia del contexto de 3 palabras que sigue al término elegido, agrupe estos resultados por la columna producto.

```sql
SELECT
    producto,
    ctx.ngram AS contexto,
    ROUND(SUM(ctx.estfrequency), 0) AS frecuencia_total
FROM (
    SELECT
        producto,
        context_ngrams(
            split(frase_limpia, ' '),
            array('envío'),
            3,
            1000
        ) AS ctx
    FROM reviews_borrar_sentences
) t
GROUP BY producto, ctx.ngram
ORDER BY producto, frecuencia_total DESC;
```