***Riegos\_Grp1***

Consigna3- ventas

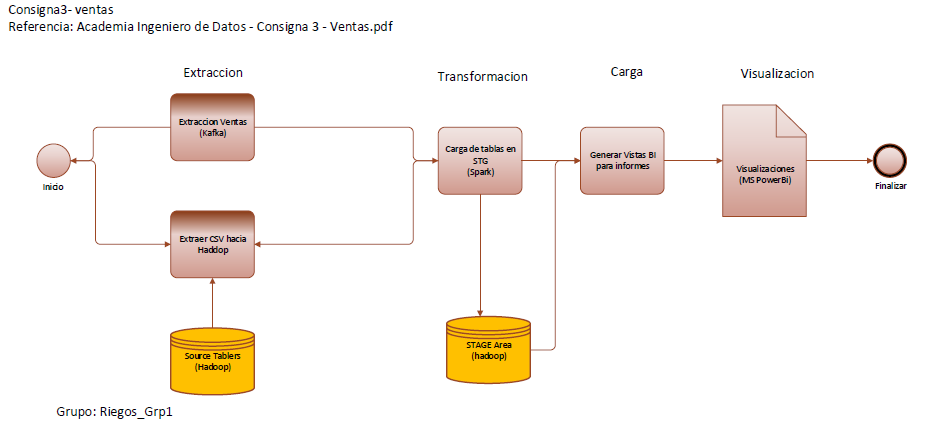
**Integrantes:**

Alan Riquelme Leg: L1000802

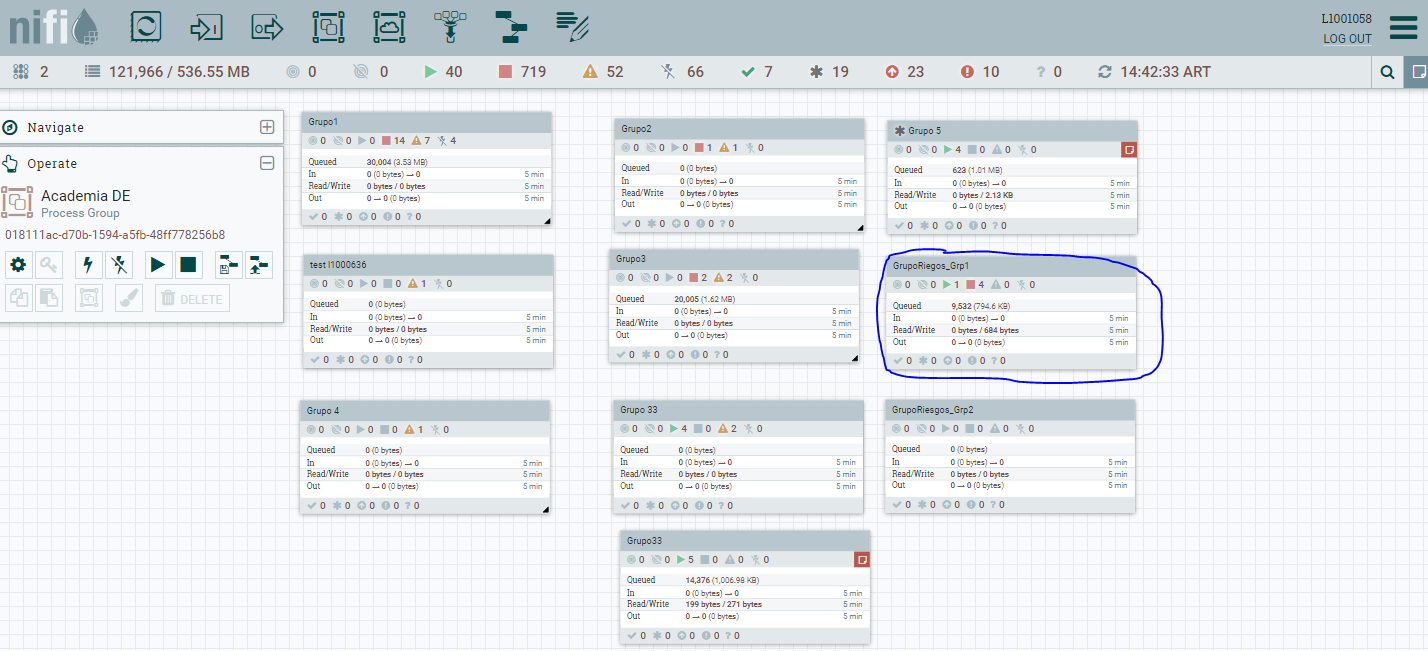
Martin Rotryng Leg: L0513427

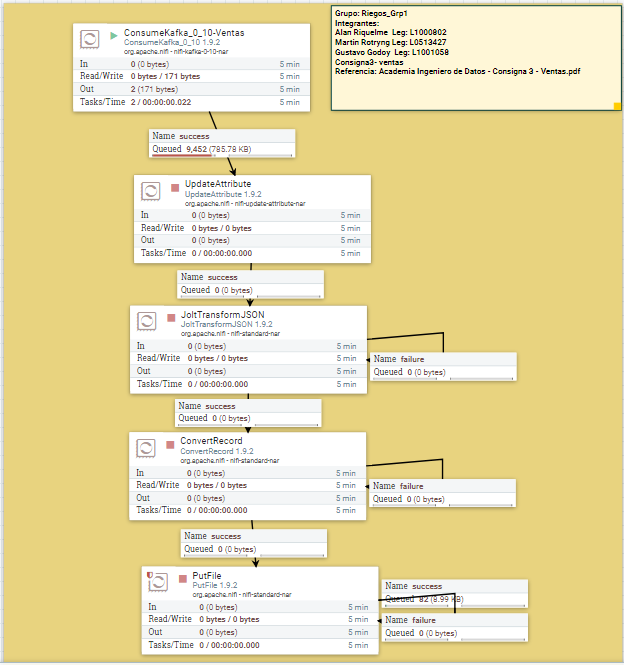
Gustavo Godoy Leg: L1001058

1. Diagrama de proceso
2. Procesos
3. Paso 1 Extracción desde ventas desde KAFKA con NIFI y colocar en CSV Files
4. Paso 2 Carga de CSV Files desde directorio a Hadoop File System (hdfs)
5. Paso 3 Creación y carga de tabla de ventas en Hadoop-Hive
6. Paso 4 Creación de consultas de alcance de negocio
7. Generación de vistas en Spark-Python para visualizaciones
8. Visualización de solución en Microsoft PowerBI
9. **Diagrama de Proceso**

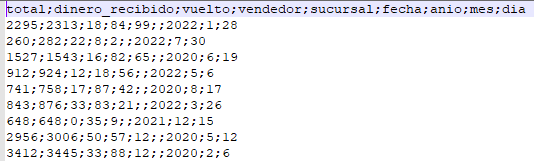


1. **Procesos**
2. Paso1 **Extracción desde ventas desde KAFKA con NIFI y colocar en CSV Files**





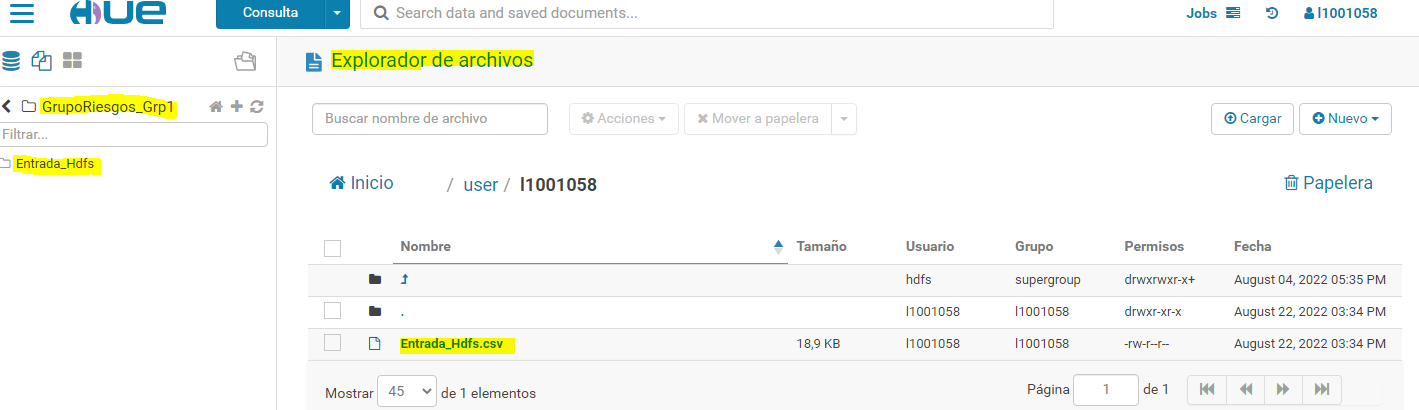
La salida es de este proceso es un archivo plano separado por ; y de formato csv



Esta es la información de las transacciones;

Nota; como no se tenía fechas históricas se agregaron registros de años anteriores para el caso práctico.

1. Paso2 **Carga de CSV Files desde directorio a Hadoop File System (hdfs)**



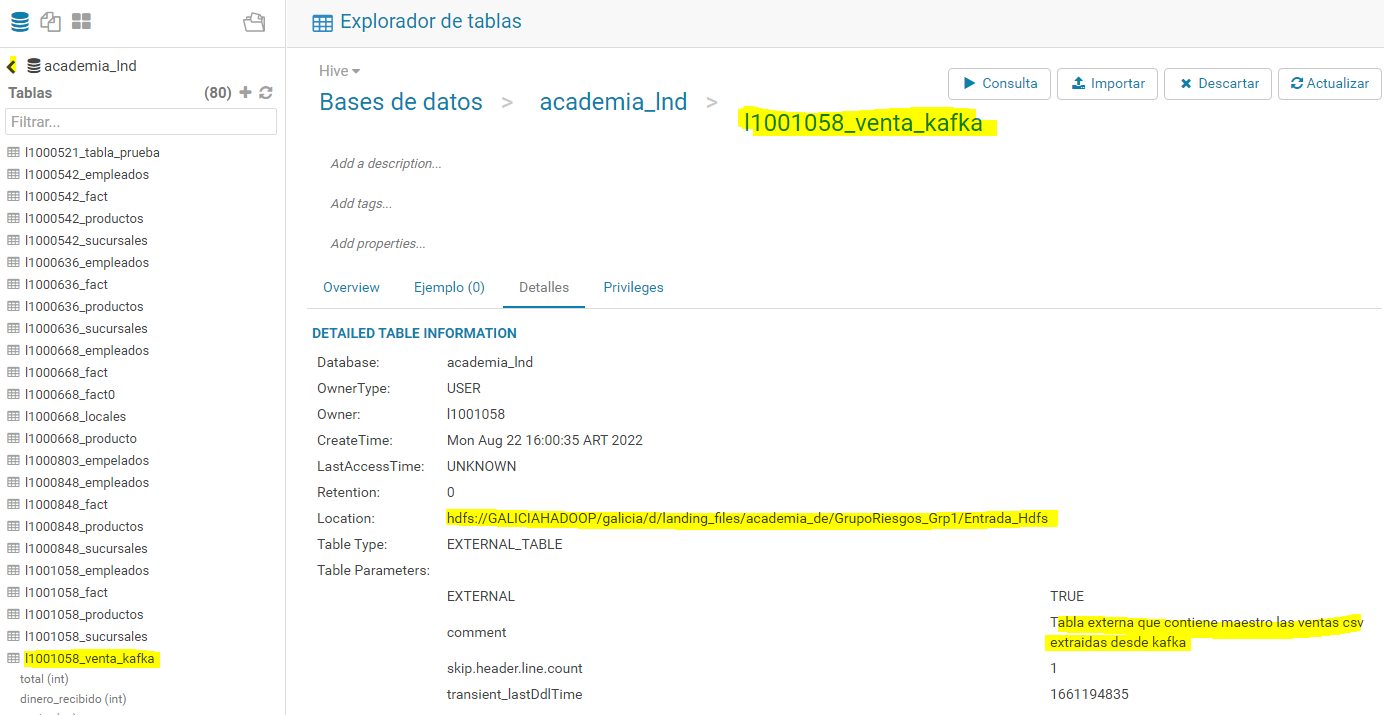
En:

/ galicia/ d/ landing\_files/ academia\_de/ GrupoRiesgos\_Grp1/ Entrada\_Hdfs/ Entrada\_Hdfs.csv

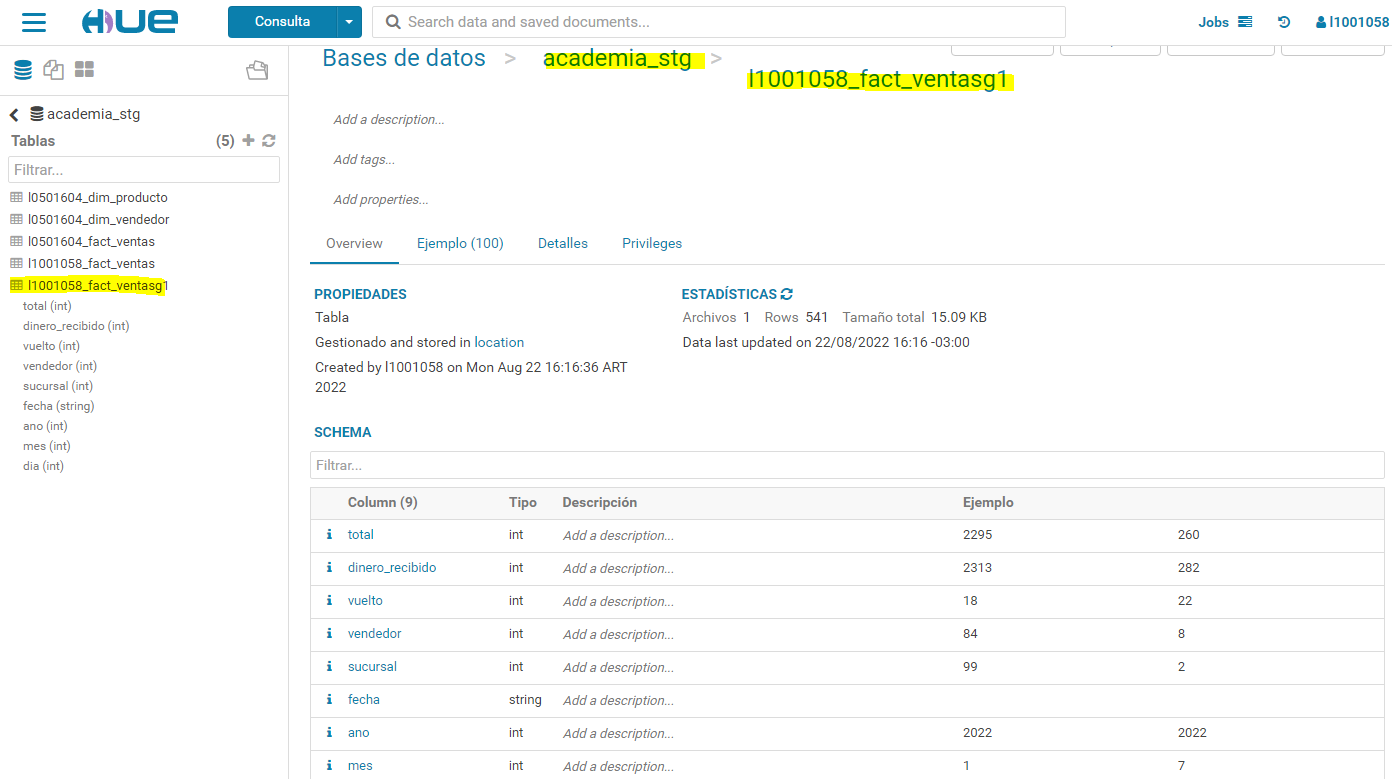


Una vez en el recipiente de Hadoop se procedió a la creación de una tabla de landing en:

Bases de datos >academia\_lnd >l1001058\_venta\_kafka



Luego, se creó la tabla de tipo parquet en hadoop en el schema de staging para ser utilizada en las soluciones del negocio.



1. Paso 3 **Creación y carga de tabla de ventas en Hadoop-Hive**

Nota: para este caso utilizamos tablas de empleados, sucursales de otros usuarios que habían cargado esa data.

A partir de la data cargada más los datos extraídos del CSV; procedimos a armar las consultas de la solución propuesta.

1. Paso4 **Creación de consultas de alcance de negocio**

Basadas en la propuesta del alcance del negocio; se dividió en 2 partes BI del negocio de ventas y reportes transaccionales, con 2 soluciones para cada uno. Con todas las consultas SQL de la solución se crearon vistas en Python con Spark para ser consumidas desde MS PowerBI.

*BI del Negocio Ventas*

1- Ranking de los 5 mejores vendedores

SELECT q.vendedor\_nombre, p.\*

FROM (

SELECT c.vendedor, sum(c.total) as pruebaTotal

FROM academia\_stg.l1001058\_fact\_ventasG1 c

group by vendedor

--ORDER BY pruebaTotal desc

) p

LEFT JOIN academia\_stg.l0501604\_dim\_vendedor q on q.id\_vendedor=p.vendedor

order by p.pruebaTotal desc LIMIT 5 ;

2- Ranking de facturación por sucursal

SELECT q.nombre, p.\*

FROM (

SELECT c.sucursal, sum(c.total) as pruebaTotal

FROM academia\_stg.l1001058\_fact\_ventasG1 c

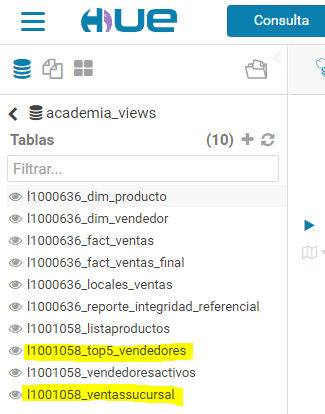
group by c.sucursal

--ORDER BY pruebaTotal desc

) p

LEFT JOIN academia\_lnd.l0339202\_sucursales q on q.id\_sucursal=p.sucursal

order by p.pruebaTotal desc --LIMIT 10 ;



Reportes transaccionales

1- Listado de productos

SELECT

id\_producto,familia,nombre, concat('$ ',precio\_unitario ) as PrecioUnidad

FROM academia\_stg.l0501604\_dim\_producto

order by id\_producto

2- Listado de vendedores activos por sucursal

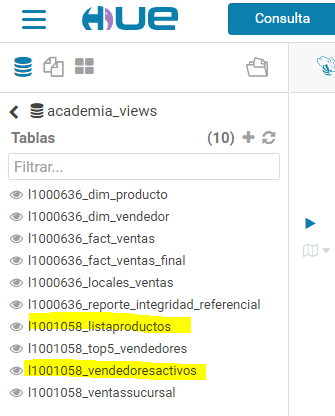
SELECT DISTINCT q.id\_vendedor, q.vendedor\_nombre,suc.nombre

from academia\_stg.l0501604\_dim\_vendedor q

inner JOIN academia\_stg.l1001058\_fact\_ventasG1 Ventas on q.id\_vendedor=Ventas.vendedor

inner JOIN academia\_lnd.l0339202\_sucursales suc on suc.id\_sucursal=Ventas.sucursal

where suc.nombre is not NULL



1. **Generación de vistas en Spark-Python para visualizaciones**
2. **Visualización de solución en Microsoft PowerBI**

Se conecta las bases de datos de Hadoop a través de ODBC Impala para obtener la información de las vistas creadas para la solución.

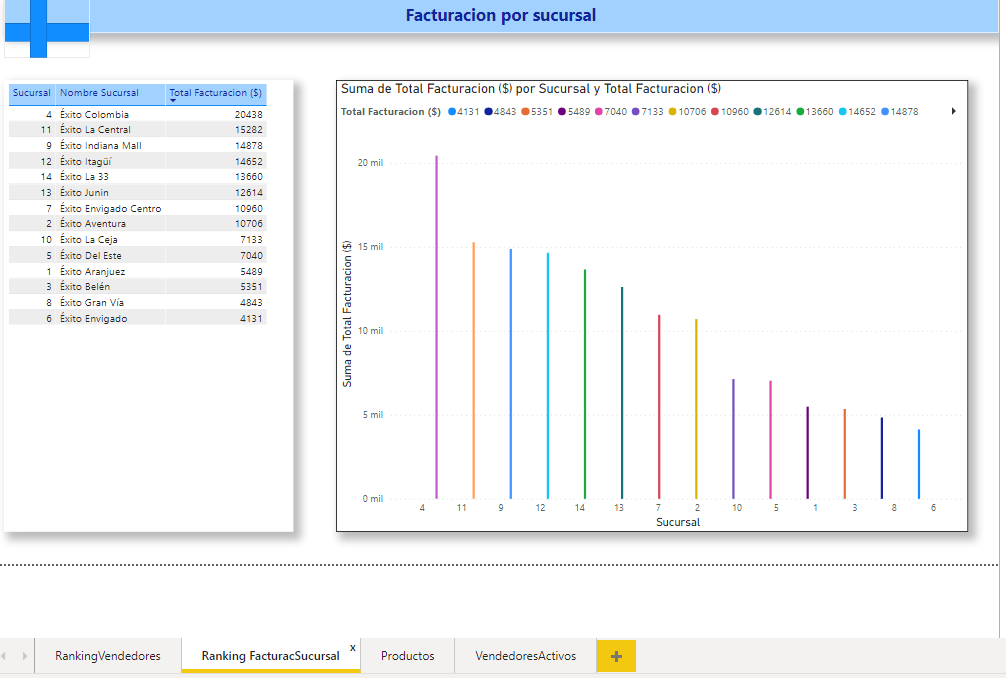
**BI del negocio**

1. Ranking de los 5 mejores vendedores

Graphical user interface

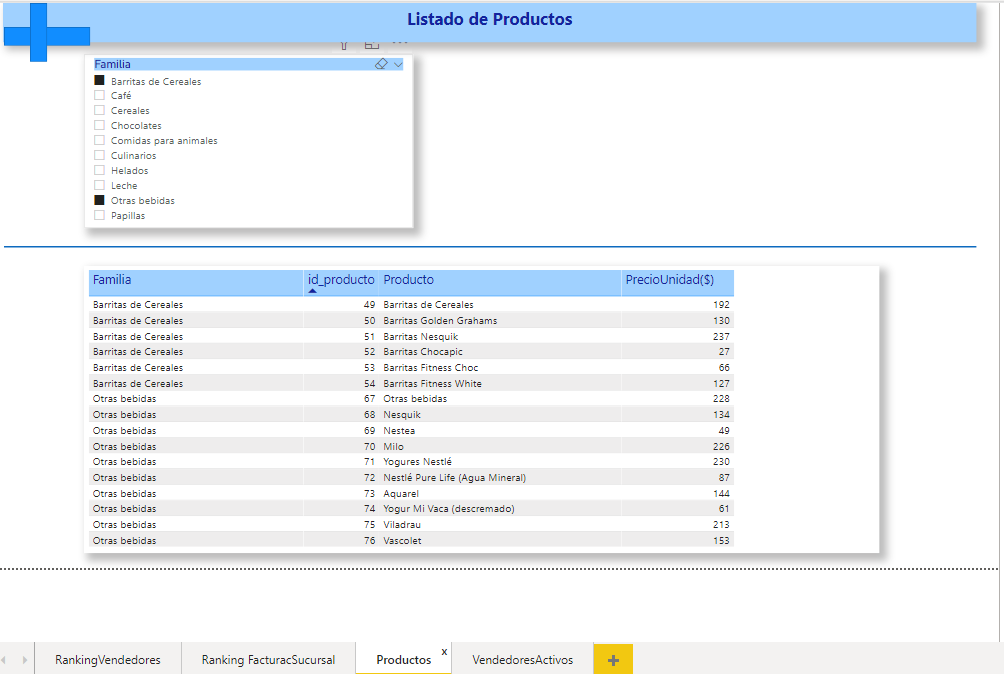
Description automatically generated

1. Ranking de facturación por sucursal



**Reportes transaccionales**

1. Listado de productos



1. Listado de vendedores activos por sucursal

