Comandos para hacer el scafold.

docker-compose build

docker compose up -d

Crear el bucket de trabajo S3 en el localstack

* Desde el archivo: create\_bucket.py
* Por commando: awslocal s3api create-bucket --bucket my-local-bucket

## Preparación del S3

### Archivos

Tenemos un par de archivos que se suben directamente a S3 habitaciones\_data.csv y hoteles.csv

Meterse en el cluster

PS C:\Users\Adrian\Downloads\PROYECTOS\PROYECTO\_BDA\Spark>

docker exec -it spark-spark-master-1 /bin/bash

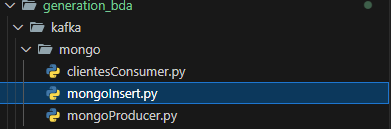
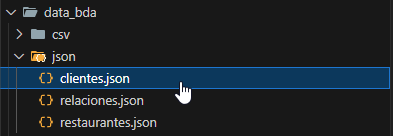
En la carpeta siguiente ejecutamos el archivo

dataroot@b17b7945d0d0:/opt/spark-apps/DataIntegration# python data\_integration.py

### MongoDB

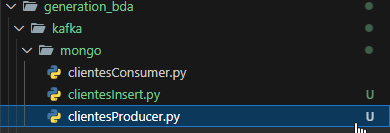
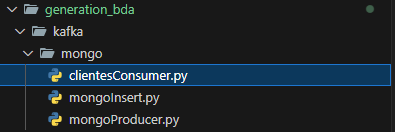
Se basa en leer un archivo .json en insertar los registros en Mongo

* python mongoInsert.py -> inserta los registros en mongo provenientes de clientes.json



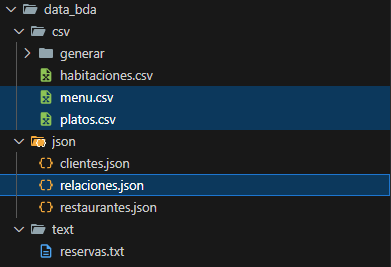
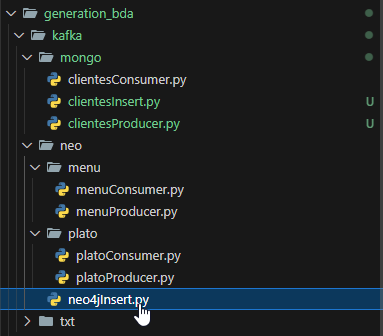
Después por medio de Kafka los registros se alojarán en S3

* clientesConsumer.py -> Se pone a la escucha para cuando se le mande la información del Producer.
* clientesProducer.py -> Se encarga de leer desde mongo\_db y mandar la información al Consumer.

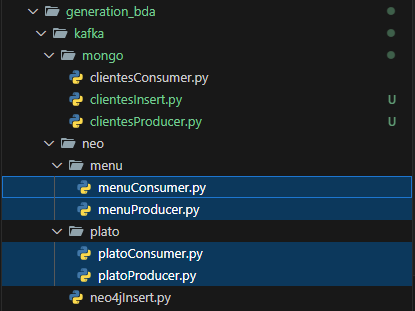
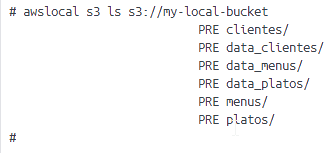
### Neo4j

Se basa en leer un archivo .csv en insertar los registros en Neo4j

Ejecutar:

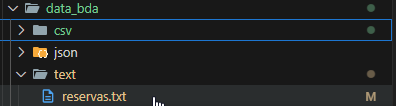
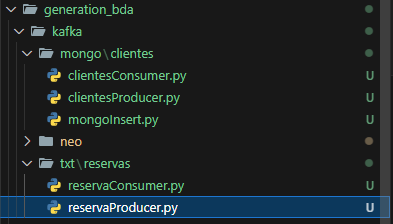
neo4jinsert.py -> inserta los registros en neo4j provenientes de menus.csv, platos.csv y relaciones.json

* El menuProducer.py se encargan de leer los archivos y mandarlos al Consumer.
* El menuConsumer.py se encargan de recibir los mensajes y crear en el bucket los archivos csv o json.
* platoProducer.py -> ídem
* platoConsumer.py -> ídem
* relacionesProducer.py -> ídem
* relacionesConsumer.py \_> ídem

### TXT

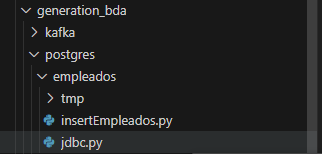
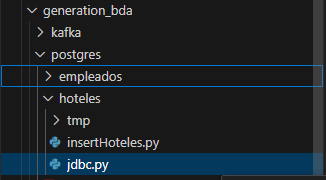
Desde un archivo .txt mandaremos por medio de Kafka sus datos de reservas a S3

* reservasProducer.py => Lee un reservas.txt y produce un mensaje que manda al Consumer
* redservasConsumer.py => Lee el mensaje del Producer y hace en S3 el reservas\_csv/

### POSTGRES

Los registros de archivos de empleados y hoteles se deben alojar previamente en tablas postgres, después, Kafca los mandará a S3.

Empleados

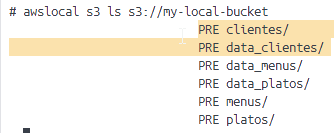
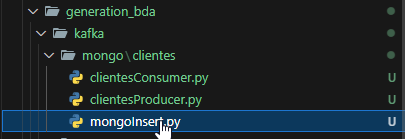
* insertEmpleados.py => Crea tabla empleados, lee el empleados.csv e inserta en postgres.
* Jdbc => consulta la base de datos y crea en bucket de S3 el empleados\_csv

Hoteles

* inserHoteles.py => Crea la tabla hoteles, lee el hoteles.csv e inserta en postgres.
* Jdbc.py => Consulta la base de datos y crea en S3 el hoteles\_json

### MongoDB

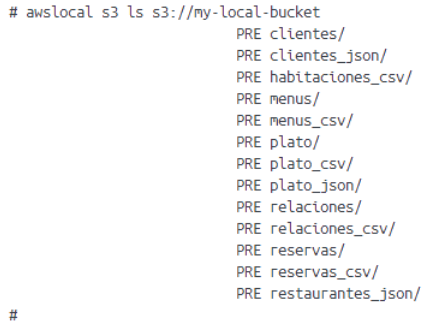
En cuanto a los Clientes, existe el archivo clientes.json que sus registros los insertaremos en MongoDB.



* mongoInsert.py -> Leer los archivos y los inserta en mongo\_db

Kafka:

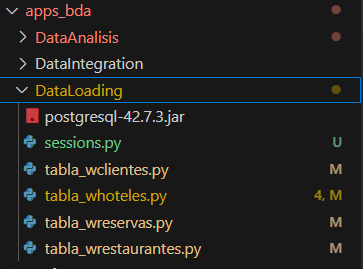
* clientesConsumer.py -> Lee los registros de Mongo mediante consulta y manda el mensaje al Consumer.
* clientesProducer.py -> Lee el mensaje del Producer y forma en el bucket de S3 el archivo de Clientes



***Ya tenemos los archivos necesarios en S3.***

## Preparación del WareHouse

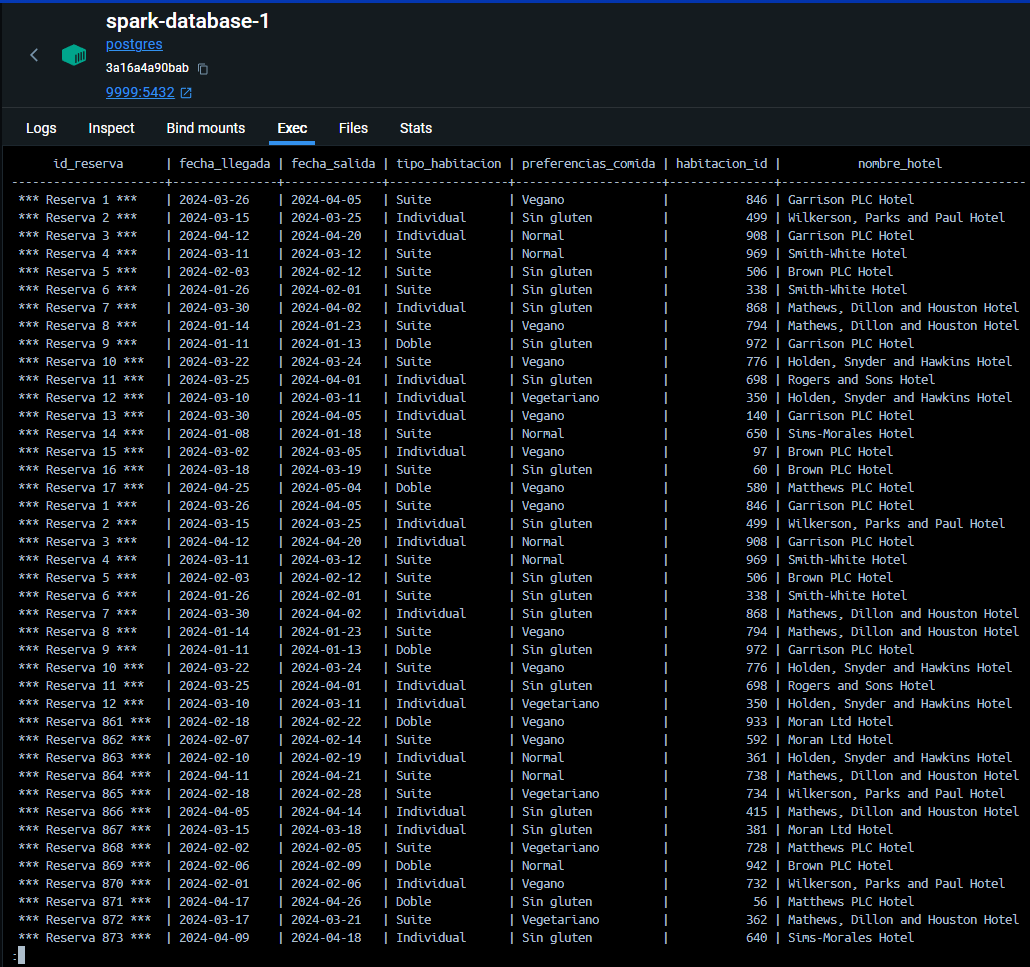
Se crean las 4 tablas con los atributos que nos interesan de los archivos en S3 creados anteriormente para insertar los registros que respondan a determinadas consultas .



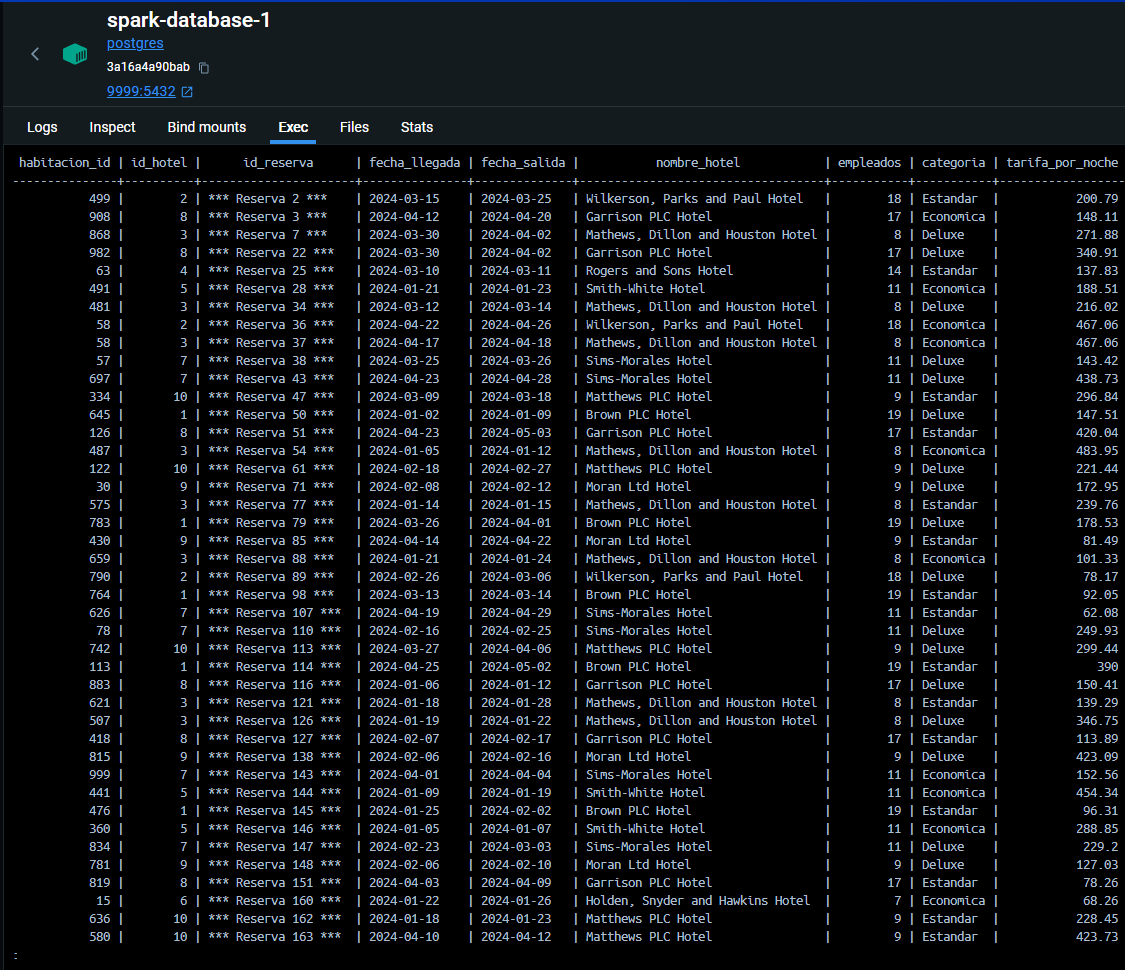
DataLoading

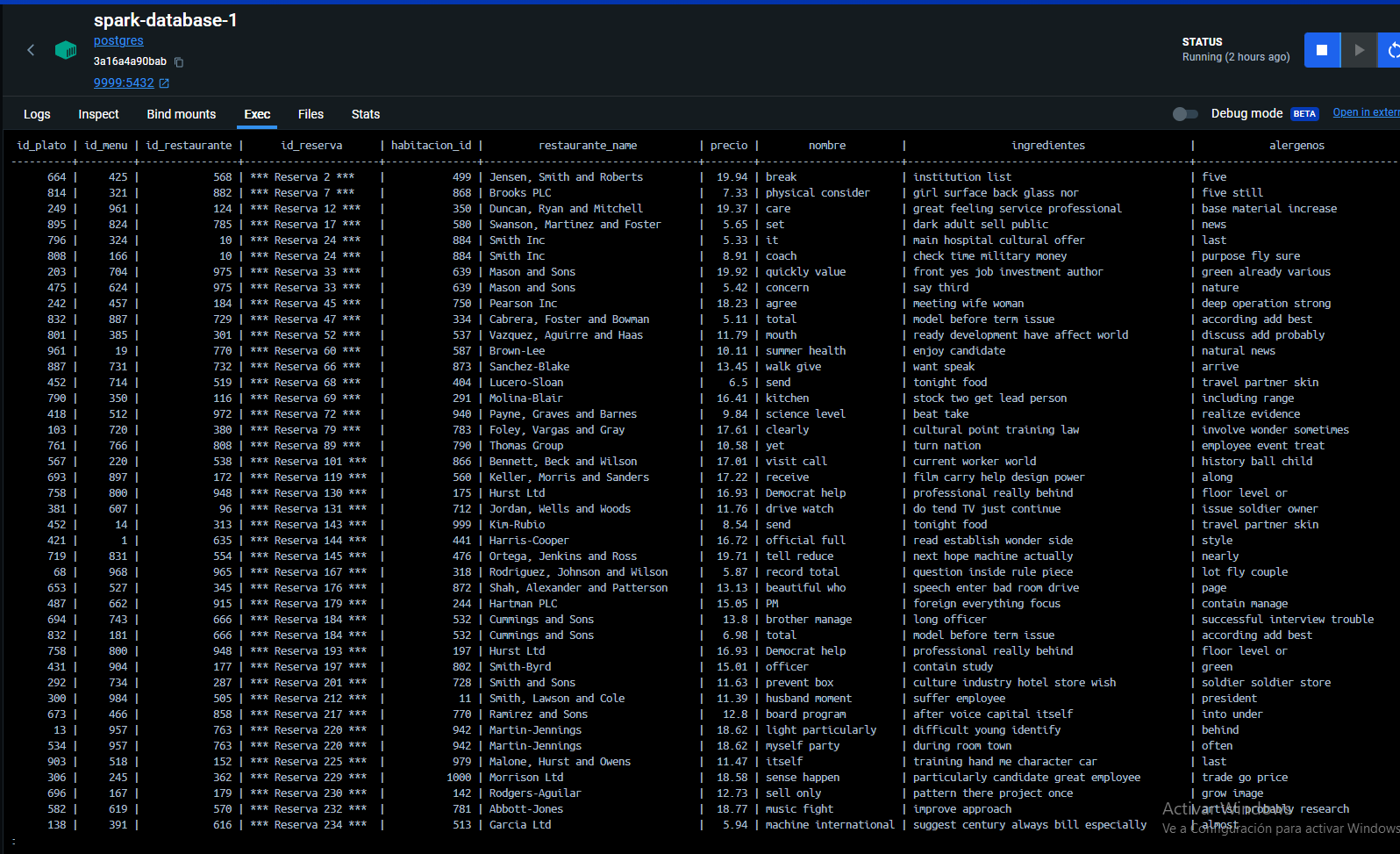
Archivo tabla\_wclientes.py -> Crea la tabla para Análisis de Clientes.

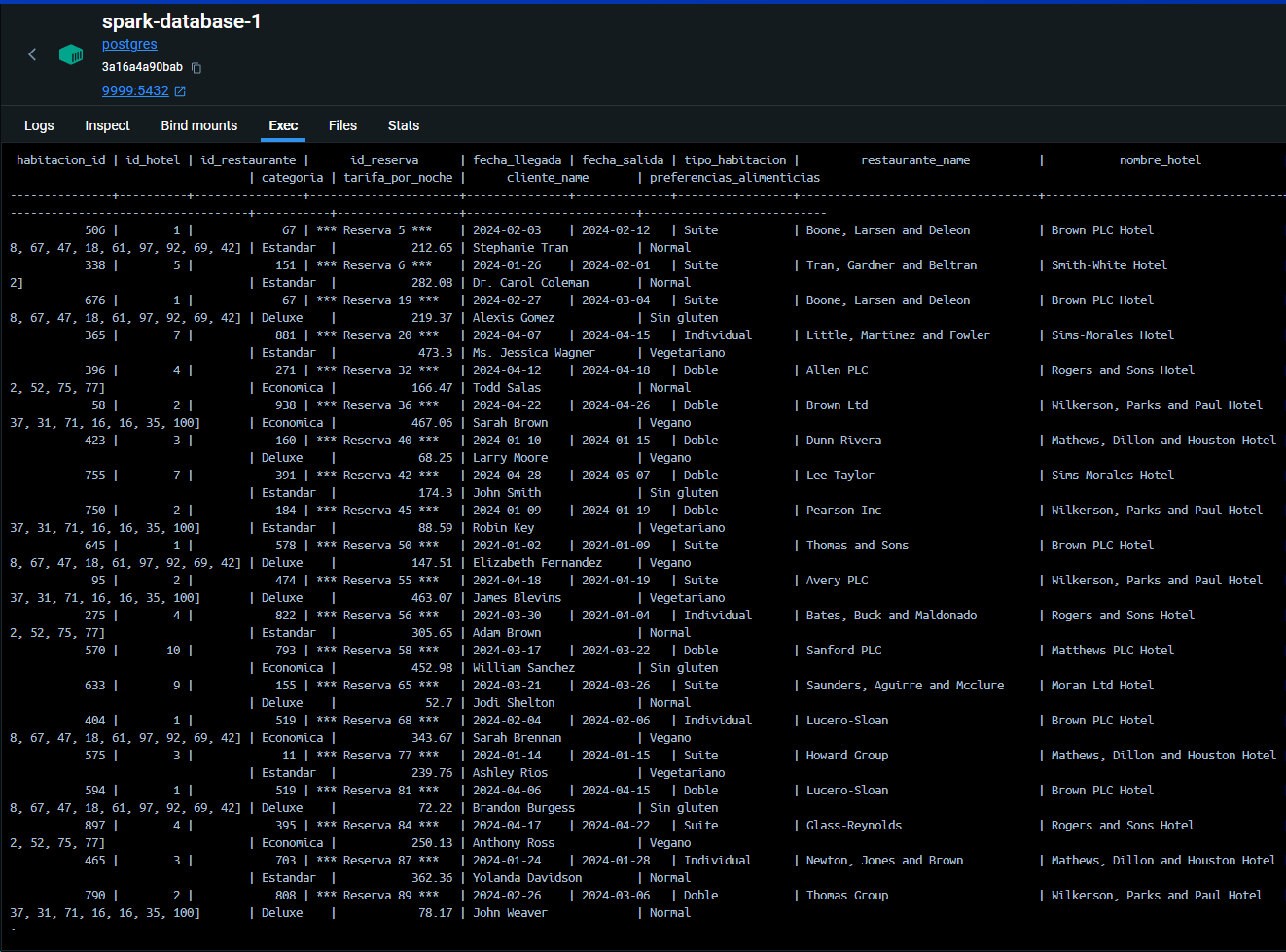
* Crear la tabla.
* Componer Dataframe con los atributos de archivos que nos interesan.
* Insertar los registros.



Archivo tabla\_whoteles.py -> Crea la tabla para Análisis de Hoteles. Ídem Incluido el nº de empleados.

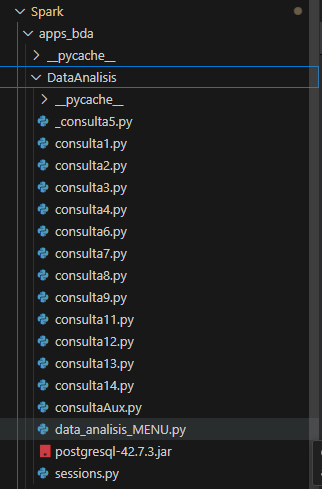
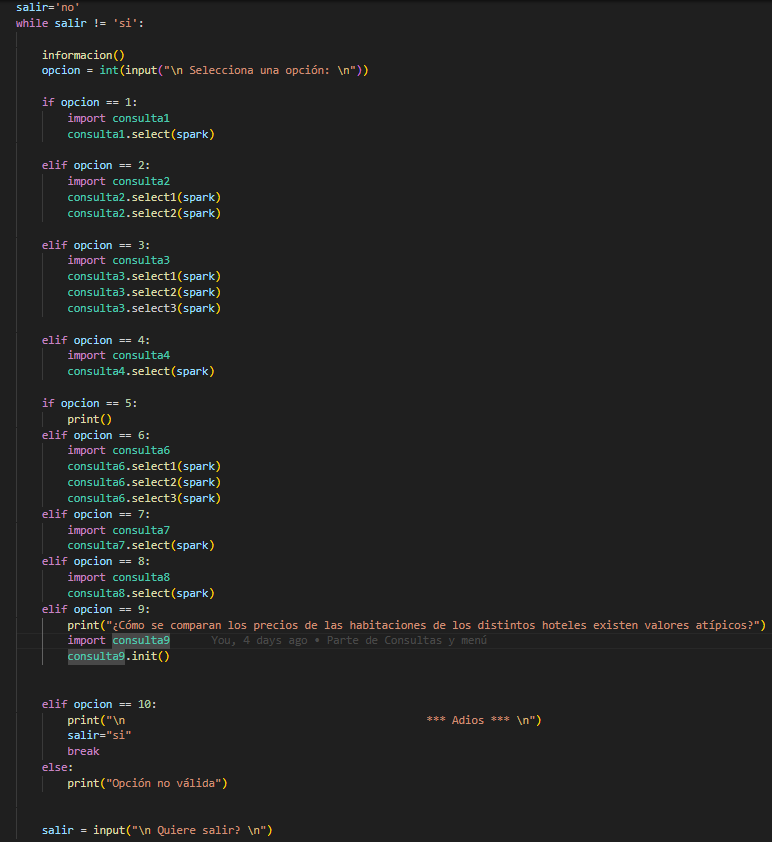


Archivo tabla\_wrestaurantes.py -> Crea la tabla para Análisis de Restaurantes. Ídem

Archivo tabla\_wreservas.py -> Crea la tabla para Análisis de Reservas. Ídem

## Consultas

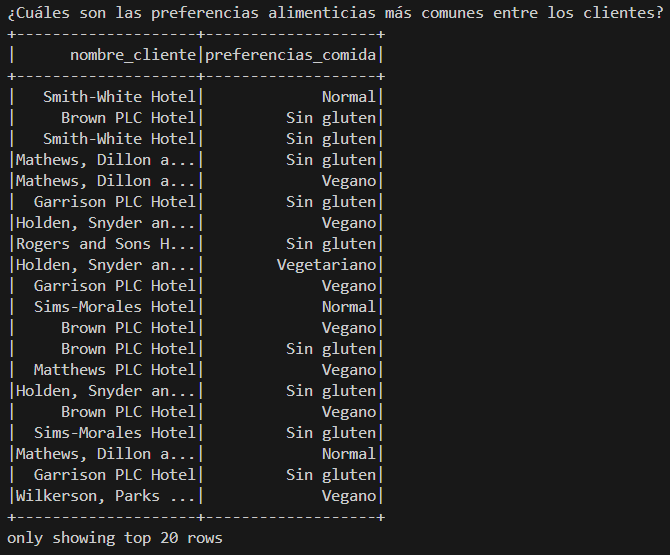
Tenemos un menú que pide al usuario que se decante por una u otra consulta

### Consulta1 OK

5.2.1 Análisis de las preferencias de los clientes

¿Cuáles son las preferencias alimenticias más comunes entre los clientes?

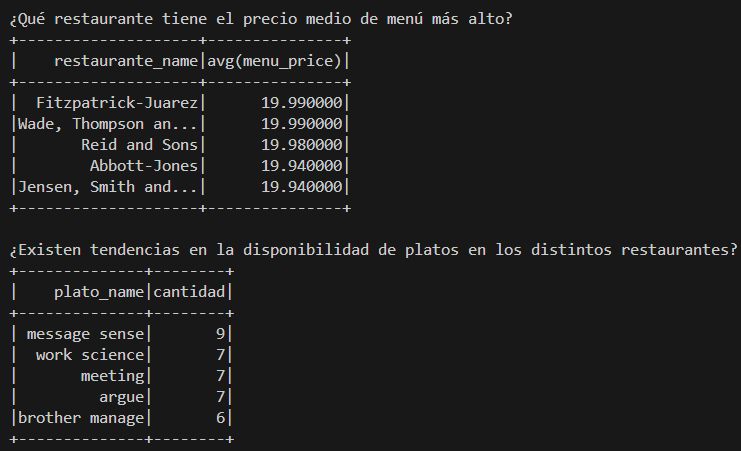


### Consulta2 OK

5.2.2 Análisis del rendimiento del restaurante:

¿Qué restaurante tiene el precio medio de menú más alto?

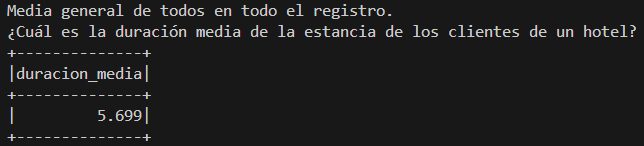
¿Existen tendencias en la disponibilidad de platos en los distintos restaurantes?



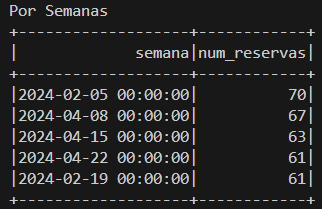
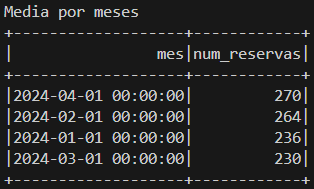
### Consulta3 OK

5.2.3 Patrones de reserva

¿Cuál es la duración media de la estancia de los clientes de un hotel?



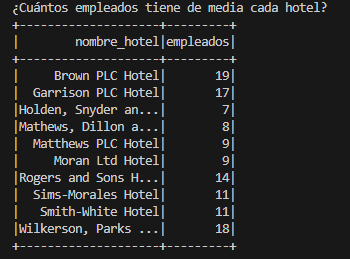
¿Existen periodos de máxima ocupación en función de las fechas de reserva?

 no es media

### Consulta4 OK

5.2.4 Gestión de empleados

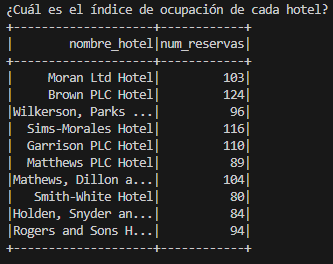
¿Cuántos empleados tiene de media cada hotel?

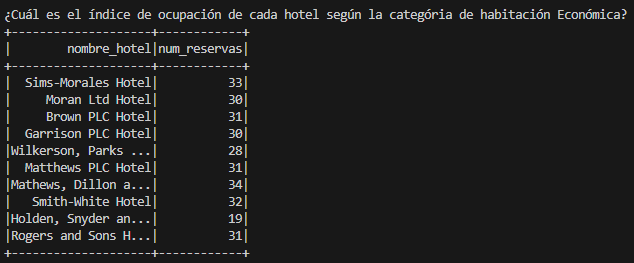


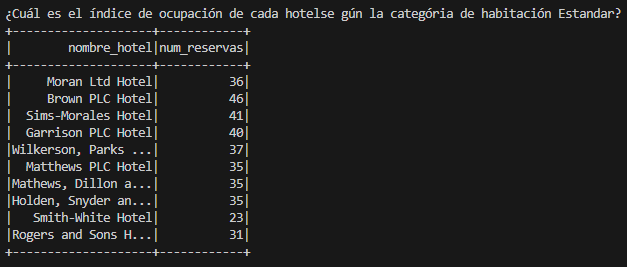
### Consulta5

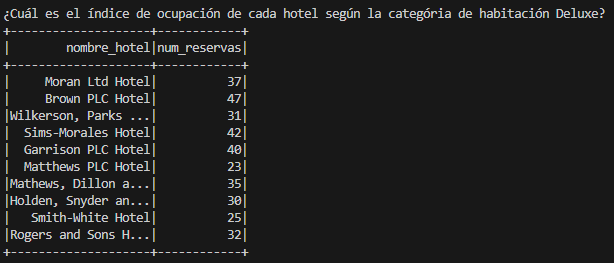
5.2.5 Ocupación e ingresos del hotel

¿Cuál es el índice de ocupación de cada hotel y varía según la categoría de habitación?

totales







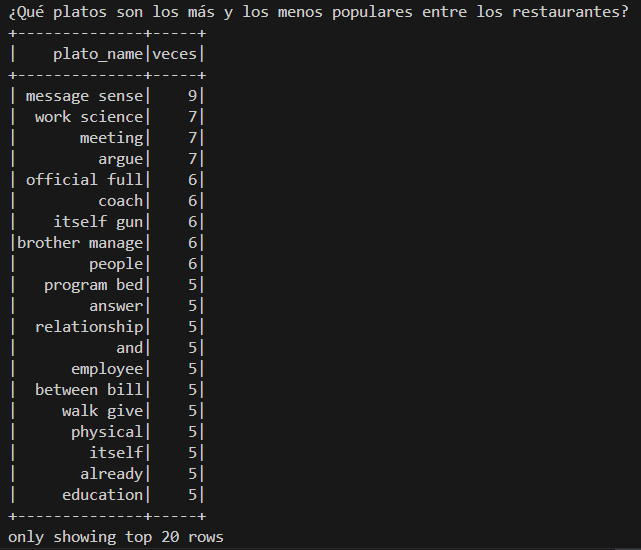
¿Podemos estimar los ingresos generados por cada hotel basándonos en los

precios de las habitaciones y los índices de ocupación?

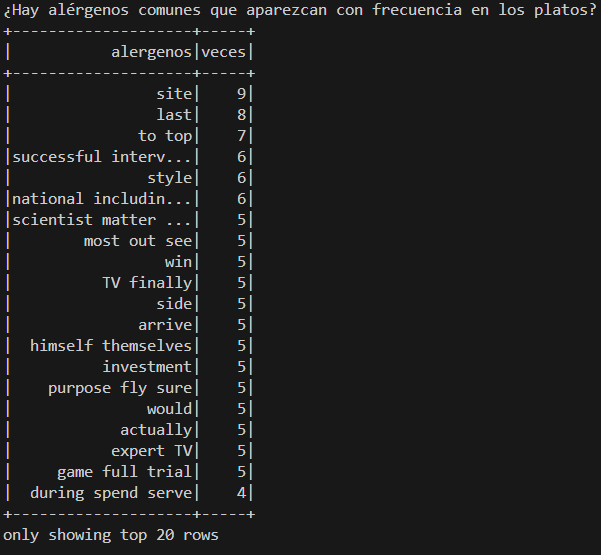
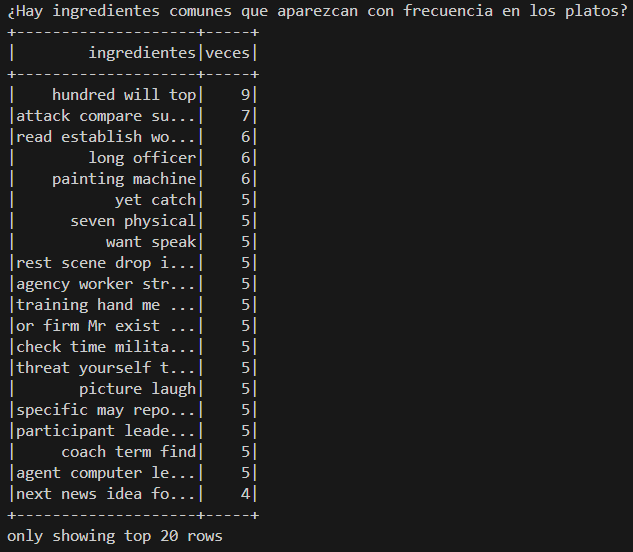
### Consulta6 OK

5.2.6 Análisis de menús

¿Qué platos son los más y los menos populares entre los restaurantes?



¿Hay ingredientes o alérgenos comunes que aparezcan con frecuencia en los platos?

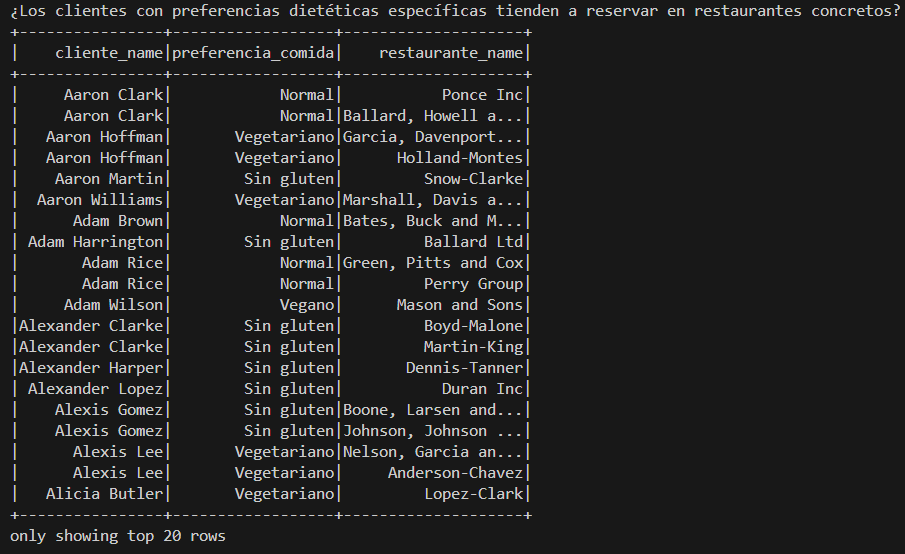
 

### Consulta7

5.2.7 Comportamiento de los clientes

¿Existen pautas en las preferencias de los clientes en función de la época del año?

¿Los clientes con preferencias dietéticas específicas tienden a reservar en restaurantes concretos?



No, un mismo cliente reserva en varios restaurantes

### Consulta8

5.2.8 Garantía de calidad

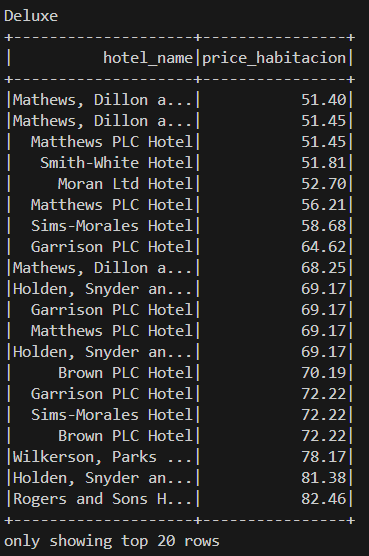
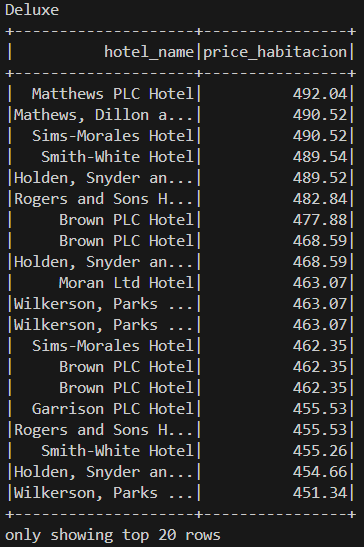
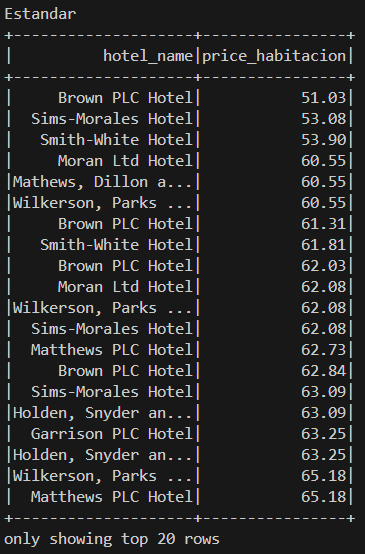
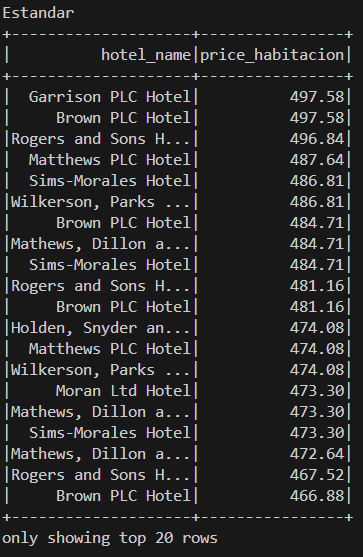
¿Existen discrepancias entre la disponibilidad de platos comunicada y las reservas reales realizadas?

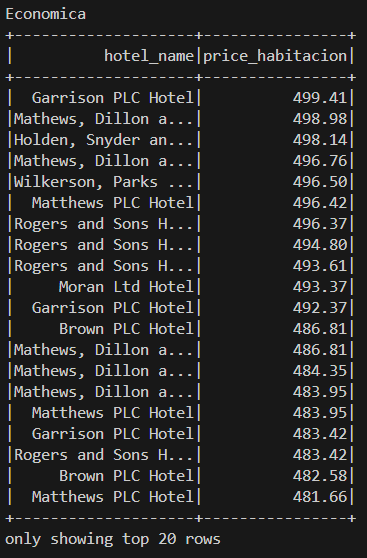
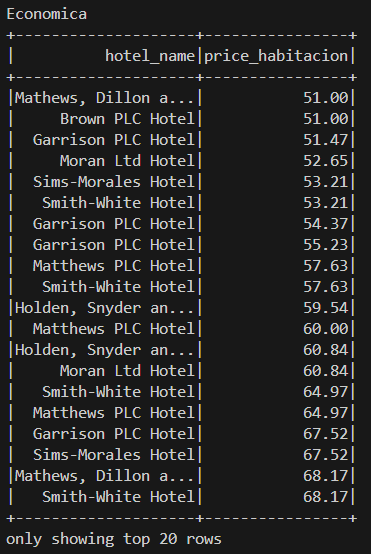
### Consulta 9 OK

5.2.9 Análisis de mercado

¿Cómo se comparan los precios de las habitaciones de los distintos hoteles y

existen valores atípicos?'''

 **** **** ****

**** ****