Examen Primera Ordinaria

1 EXTRACCIÓN de datos

En la extracción de datos se cargará información a s3. Se hará en su totalidad usando Spark, no es válido utilizar librerías como boto3 o derivados.

1.1 Archivos

Será necesario cargar los siguientes archivos de datos **directamente** a S3.

1.1.1 clientes.json

Descripción: Fichero JSON de datos de clientes.
Columnas: id_cliente, nombre, direccion, preferencias_alimenticias.

1.1.2 restaurantes.json:

Descripción: Fichero JSON para datos de restaurantes.
Columnas: id_restaurante, nombre, id_hotel.

1.1.3 habitaciones.csv

Descripción: Fichero CSV para datos de habitaciones.
Columnas: numero_habitacion, categoria, tarifa_por_noche.

1.1.4 menus.csv

Descripción: Archivo CSV para los datos de los menús.
Columnas: id_menu, nombre_plato, precio, disponibilidad, id_restaurante.

1.2 Postgres

Esta información vendrá desde Postgres, para ellos tendréis que leer dos archivos csv y cargarlos en una BBDD de postgres.

- **Puerto BBDD:** 9999
- **Nombre BBDD:** PrimOrd
- **Nombre usuario:** primOrd
- **Password usuario:** bdaPrimOrd

1.2.1 empleados.csv

Descripción: Fichero CSV de datos de empleados.
Columnas: id_empleado, nombre, posicion, fecha_contratacion.
Tabla: Empleados

1.2.2 hoteles.csv

Descripción: Fichero CSV de datos de hoteles.
Columnas: id_hotel, nombre_hotel, direccion_hotel.
Tabla: Hoteles

1.3 Kafka

Esta información vendrá desde kafka, para ellos tendréis que **leer un archivo txt, convertirlo** al formato **json** y **enviarlo** por **kafka**.

1.3.1 reservas.txt

Descripción: Fichero de texto para datos de reservas con separación única:

**** Reserva 1 ****
ID Cliente: 23
Fecha Llegada: 2024-04-20
Fecha Salida: 2024-04-25
Tipo Habitacion: Suite
Preferencias Comida: Vegetariano
ID Restaurante: 17

Cada reserva se indica con "*** Reserva {id} ***".
Para cada reserva, se indica el ID del cliente, la fecha de llegada, la fecha de salida, el tipo de habitación, las preferencias dietéticas y el ID del restaurante.

La información se leerá una vez introducida en kafka a través de Spark, es decir, el **productor** de la información **no utiliza spark,** pero el **consumer** sí **utiliza spark.**

2 TRANSFORMACIÓN de datos

En esta etapa, los datos ya estarán en s3 con LocalStack y tendréis que prepararlos para el apartado de análisis, modificad lo que veais conveniente. **No hace falta hacer ningún tratamiento de errores, vacíos, etc. Es opcional.**

3 LOAD: Data Warehouse

3.1 Data loading

Los datos transformados se cargarán en Postgres para su análisis posterior en 4 tablas distintas que responderán a las preguntas del Data analytics. Solo poner la información de cada tabla que sea interesante para resolver estas preguntas.

3.2 Data analytics

Usando Apache Spark tenéis que obtener los datos a través de postgres y realizar consultas que contengan análisis avanzados sobre los datos almacenados en el almacén de datos.
Pregunta 1: ¿Qué clientes han hecho reservas y cuáles son sus preferencias de habitación y comida?
Pregunta 2: ¿Qué habitaciones hay reservadas para cada reserva, y cuáles son sus respectivas categorías y tarifas nocturnas?
Pregunta 3: ¿Quiénes son los empleados que trabajan en cada restaurante, junto con sus cargos y fechas de contratación?
Pregunta 4: ¿Cuántas reservas se hicieron para cada categoría de habitación, y cuáles son las correspondientes preferencias de comida de los clientes?

Hadoop

Hacer un MapReduce y un archivo Pig Latin que responda a la siguiente pregunta usando un archivo csv dado en los apartados anteriores:
¿Cuántas reservas se hicieron al mes?

4 Estructura del proyecto

Para este proyecto tendréis que seguir la siguiente estructura:

data_Prim_ord/: Esta carpeta contiene todos los datos necesarios para el análisis.

- **json/**: Aquí se almacenan los archivos json.
- **text/:** Aquí se almacena el archivo txt
- **csv/:** Esta carpeta almacena los archivos csv.

data_generation/: Esta carpeta contiene todos los archivos python que generen los archivos anteriores, **tienen que generarlos en la carpeta especificada.**

apps_Prim_ord/: Esta carpeta contiene el resto de archivos py.

- **data_integration**.**py**: Almacena datos en s3
- **data_transformation**.**py**: Organiza/une los datos, es opcional.
- **data_load**.**py**: Guarda los datos en las 4 tablas distintas de Postgres.
- **data_analysis**.**py**: documento donde se realiza el análisis de datos.

Hadoop/: Esta carpeta contiene los archivos asociados a Hadoop, el .Java y .pig

Si existe algún archivo más que sea necesario **comentarlo** con el profesor.

5 Evaluación

Me tendréis que hacer un zip con todos los archivos utilizados para hacer el examen.