Taller evaluativo 2
Proceso ETL con los dataset de Airbnb Ciudad de México (20%)

Fecha de entrega: octubre 19, 11:59 PM
Entrega: Tarea Teams

Objetivo: Aplicar los conceptos de Extracción, Transformación y Carga (ETL) sobre los datasets de Airbnb Ciudad de México, almacenados en una base de datos MongoDB local, implementando un proceso automatizado en Python que incluya manejo de logs y documentación del flujo de transformación.

Actividades a desarrollar

1. Conexión y Extracción de Datos

Actividades:

1.1. Conectarse a la base de datos local de MongoDB que contiene las colecciones:
- Listings
- Reviews
- calendar

1.2. Crear una clase Extraccion en Python que:
- Establezca conexión con la base de datos.
- Permita obtener los datos de cada colección y cargarlos en DataFrames de pandas.
- Registre en un log cada conexión y cantidad de registros extraídos.

Entregable: Archivo Python (extraccion.py) con la clase Extraccion debidamente documentada y funcional.

2. Análisis Exploratorio de Datos (EDA)

Objetivo: comprender la estructura, calidad y distribución de los datos antes de transformarlos.

Actividades en Jupyter Notebook:

2.1. **Entendimiento general de los datos:**
- Mostrar las primeras filas de cada colección (head()).
- Identificar cantidad de registros y columnas.
- Verificar tipos de datos (info()).

2.2. **Calidad de datos:**
- Analizar **valores nulos o faltantes** por columna.
- Analizar **duplicados** y decidir si deben eliminarse.
- Detectar posibles **valores atípicos** en campos como price, minimum_nights, availability_365.

2.3. **Transformaciones potenciales:**

- Verificar si alguna colección requiere **desanidar campos** (por ejemplo, amenities, host).
- Evaluar si hay necesidad de **pivotear** o **agrupar** datos (por ejemplo, calendario por mes o semana).
- Estandarizar unidades o formatos de fecha, moneda y texto.

2.4. **Documentar los hallazgos:**

En el Notebook, describir los principales hallazgos: inconsistencias, correlaciones, outliers, etc.

Entregable: Archivo Jupyter (exploracion_airbnb.ipynb) con código, gráficas y análisis interpretativo.

3. Transformación de Datos

Objetivo: preparar los datos para ser cargados en el Data Warehouse o para análisis avanzado.

Actividades:

3.1. Crear la clase Transformacion en Python que implemente al menos las siguientes tareas mínimas:

- Limpieza de valores nulos y duplicados.
- Normalización de precios (quitar símbolos $, ,, convertir a número).
- Conversión de fechas a formato estándar ISO (YYYY-MM-DD).
- Derivación de variables:
  - Mes, año, día, trimestre a partir de date.
  - Categorización de precios (por rangos).
- Expansión o división de campos anidados (por ejemplo, amenities → lista o columnas binarias).
- Generación de un DataFrame limpio y listo para carga.

3.2. Documentar cada transformación aplicada (en comentarios o docstring).

3.3. Integrar el manejo de logs para registrar:

- Transformaciones aplicadas.
- Cantidad de registros antes y después de la limpieza.

Entregable: Archivo Python (transformacion.py) con la clase Transformacion.

4. Carga de Datos

Actividades:

4.1. Crear la clase Carga que:

- Inserte los datos transformados en una nueva base de datos **SQLite.** (SGDB como MS SQL Server, Oracle, PostgreSQL o MySQL Dan Ñapa para proyecto final).
- Inserte los datos transformados en uno o varios archivos XLSX.
- Verifique que los registros se carguen correctamente.

- Registre los eventos en logs.

Entregable: Archivo Python (carga.py) con la clase Carga.

5. Manejo de Logs

Requerimiento obligatorio:

Todos los scripts (extraccion.py, transformacion.py, carga.py) deben incluir una clase Logs que:

- Genere un archivo por ejecución (ej. logs/log_YYYYMMDD_HHMM.txt).
- Registre mensajes con INFO, WARNING o ERROR.
- Incluya fechas y descripciones claras.

6. Informe final

El informe del grupo debe incluir:

1. Portada
2. Introducción
3. Descripción del dataset
4. Resumen del análisis exploratorio.
5. Gráficas y hallazgos principales.
6. Descripción de las transformaciones realizadas.
7. Ejemplo del log generado.
8. Conclusiones sobre la calidad y utilidad de los datos.
9. Referencias.

Formato: PDF.

7. Entrega en repositorio

Cada grupo deberá subir su proyecto completo a un **repositorio GitHub o GitLab público** con la siguiente estructura mínima:

```
etl_airbnb/
│
├── scr/
├── notebooks/
├── logs/
├── README.md
└── requirements.txt
```

El archivo README.md debe incluir:

- Descripción del proyecto y objetivo.
- Instrucciones de instalación (crear entorno virtual, instalar dependencias, ejecutar main).
- Integrantes del grupo y responsabilidades.
- Ejemplo de ejecución del ETL.

8. Rúbrica de evaluación

Criterio	Excelente (5)	Satisfactorio (4)	Básico (3)	Insuficiente (1-2)	Pond.
Exploración y visualización de datos	EDA completo con gráficas interpretadas y hallazgos claros	EDA con gráficas básicas y descripciones parciales	EDA superficial sin interpretación	No realiza análisis ni visualizaciones	25%
Transformaciones aplicadas	Limpieza completa, transformaciones correctas y justificadas	Transformaciones adecuadas con leves fallos	Limpieza parcial o mal documentada	No transforma o genera errores	25%
Implementación de clases Python (ETL + Logs)	Código modular, documentado y funcional	Código funcional pero poco modular	Código con errores o sin documentación	Código incompleto o inejecutable	25%
Informe final	Redacción clara, análisis crítico y evidencias completas	Informe correcto pero poco analítico	Informe superficial	No entrega informe o no corresponde al trabajo	15%
Organización y presentación	Orden lógico, buen formato, trabajo en equipo	Presentación comprensible	Desordenado o mal estructurado	Confuso o incompleto	10%