UNIVERSIDAD TORCUATO DI TELLA

Lucas Canda
Lucas Piñera
Santiago Romero

Trabajo Práctico - Ingeniería de Datos

Segunda entrega: arquitectura y flujo de datos

1. Arquitectura

a. ¿Cuál es el origen de los datos?¿Qué cadencia tienen, con qué formato arriban, qué volumen se espera?

Contexto Real

Origen de los datos:

En el contexto de la vida real los datos provienen, principalmente, de la interacción de los usuarios con la aplicación de Uber. Esto incluye:

- **Pasajeros y Conductores:** Información personal y de contacto, calificaciones, etc.
- **Viajes:** Detalles del viaje como origen, destino, fecha y hora, distancia, estado y calificación.
- **Vehículos:** Detalles sobre el vehículo, incluyendo modelo, patente y características.
- **Pagos:** Información sobre las transacciones realizadas, incluyendo el monto, método de pago, y relación con los viajes.

Cadencia, formato y volumen:

- **Cadencia:** Los datos llegan en tiempo real, conforme a la ocurrencia de los eventos (por ejemplo, cuando un viaje se inicia o finaliza, cuando se realiza un pago, etc.).
- **Formato**: Los datos arriban en formatos estructurados, como registros de bases de datos.
- **Volumen esperado:** Debido al alto volumen de operaciones diarias de Uber, se espera una gran cantidad de datos. Esto puede incluir millones de registros diarios, especialmente en grandes áreas metropolitanas.
  - Datos de viajes y pagos: Altos volúmenes de datos debido a la cantidad de transacciones.
  - Datos de vehículos y conductores: Volumen moderado, actualizaciones frecuentes.

Lucas Canda
Lucas Piñera
Santiago Romero

Contexto TP

Origen de los datos:

Para simular esta interacción de los usuarios con la aplicación, lo que hicimos fue generar datos falsos, pero coherentes, con *faker*, de forma que podamos poblar nuestras tablas.

Cadencia, formato y volumen:

- **Cadencia:** Los datos se generan cada un minuto.
- **Formato:** Registros de bases de datos relacionales.
- **Volumen esperado:** La cantidad de datos nuevos generados por minuto la definimos en la función *generate_data* en el archivo *fill_data.py*. Espera modelar el funcionamiento de Uber a una velocidad más baja que nos permita una mejor operabilidad y entendimiento de los datos en el contexto de trabajo práctico.

b. ¿Cómo es el linaje de los datos desde su origen hasta que las distintas aplicaciones los consumen? ¿Cómo son los procesos intermedios y qué patrones siguen?

Contexto Real

Linaje de los datos:

- **Origen:** Los datos se originan de las aplicaciones móviles de los pasajeros y conductores, así como de los sistemas de pago y los sistemas de gestión de flotas de vehículos.
- **Captura:** Los datos son capturados en tiempo real mediante la aplicación de Uber, y son enviados a los servidores.
- **Procesamiento:** En los servidores, los datos pasan por una serie de procesos ETL (Extracción, Transformación y Carga):
    - **Extracción:** Recopilación de datos de diversas fuentes.
    - **Transformación:** Limpieza y normalización de datos para garantizar la consistencia y calidad. También podrían llegar a agregar información adicional que puede ser útil para análisis futuros (por ejemplo, condiciones meteorológicas durante el viaje).
    - **Carga:** Los datos transformados se cargan en la base de datos relacional y son indexados para realizar consultas rápidas y eficiente.:
- **Consumo:** Los datos transformados se consumen mediante APIs, dashboards analíticos y aplicaciones internas para diferentes propósitos operativos y estratégicos.

Lucas Canda
Lucas Piñera
Santiago Romero

Contexto TP

Linaje de los datos:

- **Origen:** Datos generados por *faker* cada minuto.
- **Almacenamiento y Procesamiento:** Utilizamos PostgreSQL como gestor de base de datos. Para orquestar el procesamiento de los datos utilizamos Apache Airflow, donde lo que decidimos hacer es que que los datos se branchen según a qué rango horario del día pertenezcan (mañana, tarde o noche) para poder analizar por separado y generar distintos insights según el horario.
- **Transformación:** Utilizamos herramientas de DBT para realizar transformaciones de los datos según el rango horario.
- **Consumo:** Usamos el resultado de las transformaciones para tomar decisiones y para uso analitico y de reporte

c. ¿Cuáles son los usos que tienen los datos en las distintas etapas de procesamiento?

Contexto Real

Etapas de procesamiento y sus usos:

- **Etapa de Ingestión:** Captura y almacenamiento inicial
  - **Uso:** Registro y almacenamiento seguro de transacciones y eventos en tiempo real.
  - **Aplicaciones:** Gestión operativa, monitorización de servicios en tiempo real.
- **Etapa de Transformación:** Procesamiento intermedio
  - **Uso:** Limpieza, validación, normalización y enriquecimiento de los datos.
  - **Aplicaciones:** Preparación de datos para análisis, generación de reportes operativos.
- **Etapa de Consumo:** Análisis y Predicciones
  - **Uso:** Análisis de comportamiento de usuarios, calidad del servicio, patrones de demanda, etc.
  - **Aplicaciones:** Mejora del servicio, toma de decisiones estratégicas, marketing dirigido.

Lucas Canda
Lucas Piñera
Santiago Romero

Contexto TP

Uso de los datos:

- **Etapa de ingestión:** Al hacer la ingesta a partir del rango horario podemos hacer el almacenamiento inicial y validar.
- **Etapa de transformación**: Enriquecemos los datos a partir de transformaciones DBT (Explicadas en el punto 3)
- **Etapa de Consumo**: Una vez transformados los datos podremos darle un uso Operacional o Anaítico como por ejemplo ver en qué rango horario es cuando hay peores calificaciones, para así poder mejorar la experiencia de usuario

d. ¿Cómo es la gobernanza del dato? ¿Cuáles son los roles y qué tipos de permisos tienen sobre los datos?

Contexto Real

Gobernanza del dato:

La gobernanza de los datos implica la gestión y el control de la calidad, integridad, seguridad y uso de los datos. Incluye la definición de roles y permisos para asegurar que los datos sean utilizados de manera adecuada y segura. En el contexto real, se contemplan distintos permisos para distintos roles.

Roles y permisos:

- **Administradores de Datos:**
  - **Permisos:** Acceso completo a todos los datos y capacidades de gestión de la base de datos.
  - **Responsabilidades:** Mantenimiento de la base de datos, aseguramiento de la calidad e integridad de los datos, implementación de políticas de seguridad y privacidad.
- **Analistas de Datos:**
  - **Permisos:** Acceso a datos necesarios para análisis y generación de reportes, pero con restricciones en la modificación de datos.
  - **Responsabilidades:** Realización de análisis de datos, generación de informes y visualizaciones para soporte en la toma de decisiones.
- **Auditores y Encargados de Cumplimiento:**
  - **Permisos:** Acceso a datos necesarios para auditorías y aseguramiento del cumplimiento de regulaciones.
  - **Responsabilidades:** Realización de auditorías, aseguramiento del cumplimiento de políticas de privacidad y regulaciones.

Políticas de seguridad y privacidad:

- **Seguridad de los Datos:** Implementación de medidas de seguridad para proteger los datos contra accesos no autorizados, pérdida o corrupción. Esto incluye cifrado de datos en tránsito y en reposo, autenticación de usuarios y control de acceso basado en roles.
- **Privacidad de los Datos:** Cumplimiento de las regulaciones de privacidad y protección de datos (como GDPR o CCPA). Esto incluye la anonimización de datos personales cuando sea necesario y la obtención de consentimiento para la recopilación y uso de datos personales.

Contexto TP

En el contexto del Trabajo Práctico, inicializamos 3 roles:

- **Admin**: Tiene todos los permisos sobre todas las tablas
- **Operator**: Tiene permisos para seleccionar, insertar y updatear sobre todas las tablas
- **Auditor**: Solamente tiene permisos para seleccionar elementos de las tablas.

2. Flujo de carga de datos

Generación de Datos con Faker

Para simular datos de forma sintética, pero que sean coherentes y se asemejen a la realidad de la aplicación de Uber utilizamos *faker*. Para eso definimos los siguientes 10 métodos dentro de la clase *Datagenerator*:

- *generate_pasajero*
- *generate_conductor*
- *generate_partnership*
- *generate_modelo_vehiculo*
    - *get_modelos*
    - *get_atributos_modelos*
- *generate_vehiculo*
- *generate_conductor_vehiculo*
- *generate_viaje*
- *generate_pago*

La función de cada uno de estos métodos es generar datos para más adelante poblar las tablas definidas en *create_table.sql*.
Además *get_modelos* y *get_atributos_modelos* son métodos que son llamadas por *generate_modelo_vehiculo* y permiten mantener la consistencia entre el vehículo, la marca del vehículo y sus características.

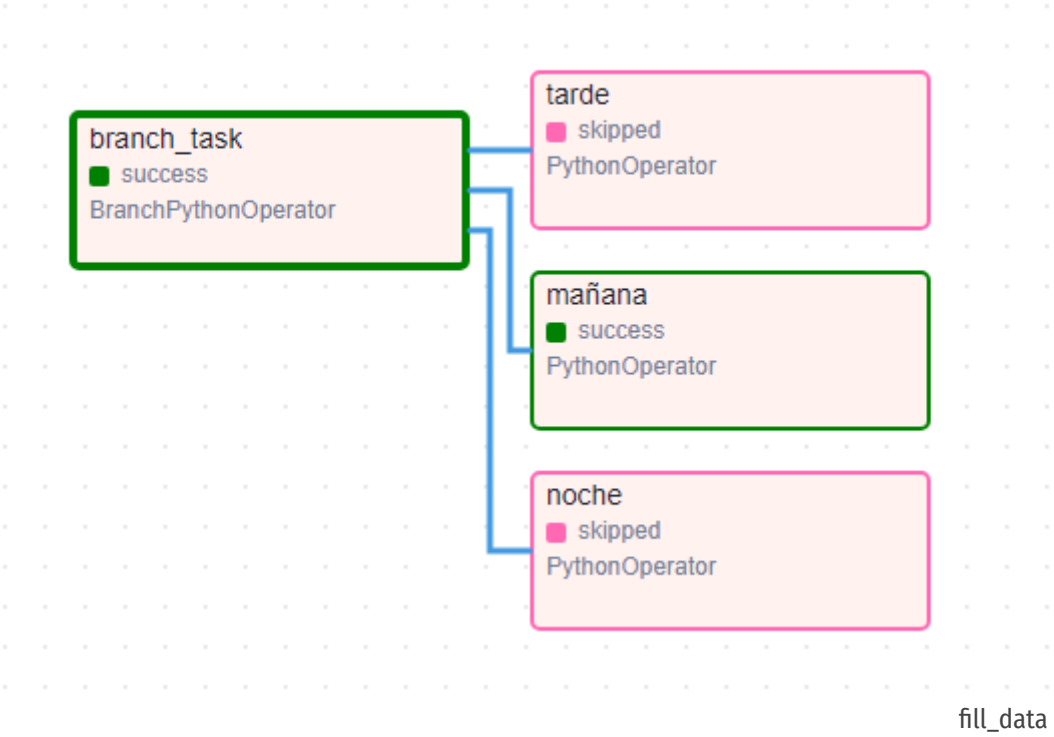
Lucas Canda
Lucas Piñera
Santiago Romero

DAG fill_data: Branching por Hora del Viaje

A continuación, creamos una función llamada generate_data la cual se encarga de inicializar la clase *Datagenerator* y utiliza sus métodos para generar los datos sintéticos e insertarlos en las tablas correspondientes cuando la función sea ejecutada. También definimos la función choose_branch, la cual determina en base a la hora actual el rango horario (mañana, tarde, noche).
Finalmente creamos un DAG, en el que definimos 4 nodos:

- *branch_task*: utiliza la función choose_branch para determinar la siguiente tarea a realizar según la hora actual.
- *mañana*: ejecuta la función generate_data si la hora actual está en el rango de entre las 5hs y las 11hs (determinado por la *branch_task*).
- *tarde*: ejecuta la función generate_data si la hora actual está en el rango de entre las 12hs y las 19hs (determinado por la *branch_task*).
- *noche*: ejecuta la función generate_data si la hora actual está en el rango de entre las 20hs y las 4hs (determinado por la *branch_task*).

En cada PythonOperator le pasamos como parámetro además el rango. De esta forma se van generar registros de viajes con horarios acordes al horario actual.



Decidimos branchear por horarios porque es información valiosa para Uber saber en qué rango horario se están realizando los viajes, para de esta forma poder analizar los datos y tomar decisiones en base a los mismos (como ejemplificaremos más adelante con las transformaciones realizadas).

Lucas Canda
Lucas Piñera
Santiago Romero

3. Enriquecimiento

DAG run_dbt: Transformaciones con DBT

Armamos dos transformaciones con DBT que consideramos que son valiosas para el análisis de datos de Uber.

1. DBT - Viajes totales y monto promedio según modelo y rango horario

La primera transformación en el DAG selecciona y combina datos de varias tablas relacionadas: *viaje*, *vehiculo*, *modelovehiculo* y *pago*. El resultado final agrupa los viajes completados por modelo de vehículo y por el rango horario, calculando el total de viajes y el promedio de ingresos por viaje para cada grupo. La finalidad de esta transformación es poder analizar cuáles son los modelos de autos que realizan más viajes y cuál es el promedio de ingresos según el modelo del vehículo. Este análisis puede ser muy valioso para que la compañía pueda identificar las marcas y modelos de autos más rentables y, en consecuencia, establecer alianzas estratégicas y formar partnerships con estos fabricantes específicos para optimizar su flota y maximizar sus ingresos.

Decidimos que esta transformación debe materializarse como una tabla por razones de rendimiento, ya que involucra múltiples joins y funciones de agregación más complejas.

Ejemplo de tabla generada a partir de la transformación:

```
postgres=# SELECT * FROM my_first_dbt_model;
  modelo   | rango  | total_viajes | avg_monto
-----------+--------+--------------+-----------
 GLE       | noche  |            2 |  21662.00
 Corolla   | noche  |            2 |  14288.00
 TT        | manana |            2 |  13302.00
 Mustang   | tarde  |            2 |   6009.50
 Suburban  | manana |            1 |  19959.00
 Santa Fe  | tarde  |            1 |  19364.00
 Explorer  | manana |            1 |  10841.00
 Corolla   | manana |            1 |  10585.00
 Insight   | noche  |            1 |   7983.00
(9 rows)
```

Lucas Canda
Lucas Piñera
Santiago Romero

2. DBT - Viajes totales, calificación promedio, monto promedio y monto por viaje según rango horario

La segunda transformación del DAG busca comprender mejor la demanda y la oferta de viajes según el rango horario. Para cada rango horario, se calculan el total de viajes, el promedio de calificaciones, el total de ingresos y el promedio de ingresos por viaje.

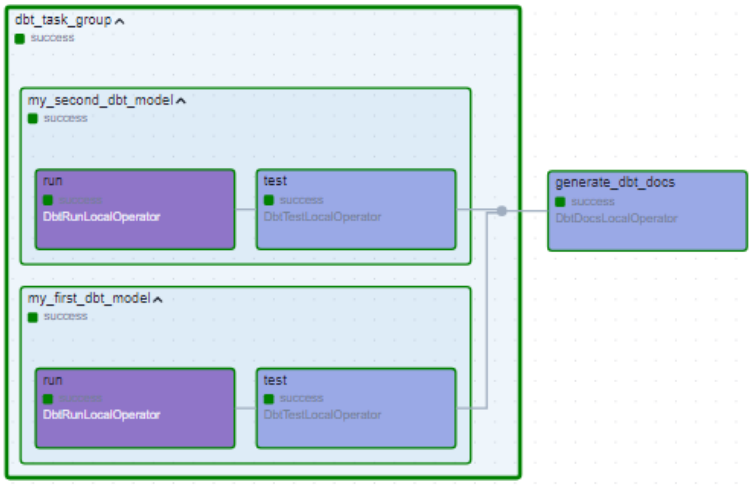
En el contexto del dominio de Uber, esto permite analizar la demanda de viajes en los distintos momentos del día y las calificaciones correspondientes, pudiendo analizar por rango horario la calidad del servicio. De esta forma Uber podría buscar la manera de que el servicio mejore en ese rango horario y así lograr mejorar la experiencia del usuario.

También se pueden detectar patrones de cantidad de viajes y de ganancia según el rango horario, lo que podría servir para identificar cuándo hay menos conductores disponibles y a partir de esto ofrecer incentivos para aumentar la disponibilidad en los rangos donde la oferta es menor.

Como esta transformación no requiere funciones tan complejas decidimos que debe materializarse como una vista para asegurar la frescura de los datos y mantener la transformación ligera.

Ejemplo de vista materializada generada a partir de la transformación:

```
postgres=# SELECT * FROM my_second_dbt_model;
 time_of_day | total_trips | avg_rating | total_revenue | avg_revenue_per_trip
-------------+-------------+------------+---------------+----------------------
 manana      |           5 |        3.6 |      67989.00 |              13597.80
 tarde       |           3 |        2.0 |      31383.00 |              10461.00
 noche       |           5 |        2.6 |      79883.00 |              15976.60
(3 rows)
```



run_dbt