En esta prueba vamos a evaluar:

- Cómo crear canales de integración de datos (carga de extracción) utilizando herramientas de integración de datos o código Python
- Cómo utilizar herramientas en la nube o código Python para transformar datos
- Cómo crear canales de transformación de datos utilizando herramientas como dbt
- Cómo orquestar canales de integración y transformación de datos utilizando una herramienta de orquestación de datos
- Cómo realizar modelado de datos utilizando técnicas como el modelo dimensional de Kimball y One Big Table
- Cómo realizar modelado de datos y visualización de datos por lotes que se actualicen en intervalos que van desde las 12 a 24 horas
- Cómo usar git para confirmar código y colaborar con equipos usando sucursales
- Cómo crear canalizaciones de CI/CD para automatizar las pruebas, la integración y la implementación de código
- Cómo empaquetar y alojar los distintos componentes en servicios Cloud a su elección

Bases de datos sugeridas:

Data source name	Data source type	URL
Public APIs	REST API	https://github.com/public-apis/public-apis
Australian Government Open Source Datasets	REST API, CSV	https://data.gov.au/
Kaggle Open Source Datasets	CSV	https://www.kaggle.com/datasets
Australian Bureau of Statistics	REST API, CSV	https://www.abs.gov.au/

World Bank Open Data	REST API, CSV	https://data.worldbank.org/
Google dataset search	CSV	https://toolbox.google.com/datasetsearch
Sample Postgres databases	Database	https://www.postgresql.org/ftp/projects/pgFoundry/dbsamples/
Confluent Datagen Source Connector	Kafka Producer	https://docs.confluent.io/cloud/current/connectors/cc-datagen-source.html
Custom Faker Kafka Producer	Kafka Prodcuer	https://greencashew.dev/posts/kafka-fake-data-producer-and-consumer-in-python/

El desafío consiste en tomar alguna base de datos a su elección (puede ser de las sugeridas y/o usted usar una que ya conozca) y generar un pipeline de procesamiento de datos que haga al menos las siguientes tareas:

- Extraer los datos desde un lugar X y llevarlos a un lugar Y
- Transformar los datos en batch haciendo cambios como:
    - Transformación de formatos
    - Transformación de valores
    - Transformación de nombres de columnas
    - Transformación de variables para crear otras nuevas
    - Agregaciones
    - Filtros
- Generar unión de distintas tablas y/o fuentes de información
- Usar Técnicas de modelado de datos (por ejemplo, modelado dimensional, una mesa grande, etc.) y crear:
    - Al menos 1 tabla de hechos
    - Al menos 3 tablas de dimensiones.
    - (Bonificación) Tipo 2 Dimensión que cambia lentamente (SCD)
    - (Bonificación) Tablas de hechos avanzadas

- Escribir pruebas de calidad de datos para tareas de transformación (por ejemplo, pruebas dbt, grandes expectativas, soda sql).
- Cree dependencias entre las tareas de integración y transformación de datos. Programe y supervise tareas utilizando una herramienta de orquestación de datos, idealmente
- Cree una aplicación de datos que genere conocimientos o predicciones para los usuarios finales (por ejemplo, un panel preestablecido usando Metabase)
- Implementar solución para servicios en la nube. Proporcione evidencia de captura de pantalla de los servicios configurados/en ejecución cuando sea posible.
- Usando git para colaboración:
    - Git confirma y git push
    - ramificación de git
    - Solicitud de extracción y revisión
- Cree canalizaciones de CI y/o CD para:
    - CI: código de compilación y prueba cuando se activa una solicitud de extracción
    - CD: compila e implementa código en un entorno de destino (por ejemplo, preproducción y producción)
- Estructura y documentación del proyecto.
    - Estructura clara del proyecto utilizando un enfoque mono-repositorio con carpetas como "transformación de datos", "integración de datos", "orquestación de datos" para los distintos componentes.
    - Documentación de código utilizando cadenas de documentación y comentarios de Python o comentarios SQL cuando sea razonable
    - Archivo README en la raíz del repositorio que explica el contexto del proyecto, la arquitectura y las instrucciones de instalación/ejecución..

No esperamos que seas experto en todas estas tareas y es por eso que te recomendamos enfocarte en las que eres mejor y hacer un MVP con eso. Luego, avanza con las que crees que puedes ir terminando según tu grado de conocimiento y/o rapidez para aprenderla. Nos fijaremos mucho en los avances incrementales que vayas haciendo y no solo en el producto final.

El plazo de entrega es hasta el domingo 10 de junio a las 23:59 hrs, sin excepciones.