

Big Data aplicado al Transporte Público en Santiago



Realizado por:

Nadia Arellano

Ana Karina Muñoz

Contenido

I. Introducción.....	3
II. Desarrollo.....	4
→ Justificación.....	4
→ Herramientas de Big Data.....	5
→ Gobierno de Datos.....	6
→ Arquitectura GCP.....	12
→ Proceso Batch.....	16
Paso 1: Conectar y descargar datos en Cloud Storage.....	16
Paso 2: Construir procesos de limpieza, transformación y carga con DataFlow....	19
Paso 3: Construir reportes en BigQuery y realizar las visualizaciones con Looker Studio.....	31
→ Proceso Pub/Sub.....	38
Paso 1: Conectar con Cloud Storage.....	38
Paso 2: Descargar y/o generar los archivos al dataLake (Cloud Storage).....	42
Paso 3: Construir procesos de limpieza, transformación y carga con DataFlow....	46
Paso 4: Construir reportes en BigQuery y realizar las visualizaciones con Looker Studio.....	52
V. Conclusiones.....	59
VI. Referencias.....	60

I. Introducción

El presente informe se centra en el análisis y la optimización del transporte público en Santiago de Chile, mediante la implementación de tecnologías de Big Data. El sistema de transporte público en Santiago, al igual que en muchas grandes ciudades del mundo, enfrenta desafíos significativos como la congestión, la puntualidad y la eficiencia operativa. La capacidad de recopilar y analizar grandes volúmenes de datos en tiempo real ofrece una oportunidad única para abordar estos problemas de manera integral.

El objetivo principal de este informe es presentar los beneficios y la aplicabilidad de una arquitectura de Big Data en la nube de Google Cloud Platform (GCP), para mejorar la eficiencia y la experiencia del usuario en el transporte público. La adopción de estas tecnologías no sólo permite una mejor planificación y gestión de los recursos, sino que también facilita la identificación de patrones de comportamiento y la anticipación de la demanda.

La implementación de esta plataforma de análisis se enmarca en un esfuerzo más amplio por modernizar el transporte público en Santiago, haciéndolo más inteligente y sostenible. Este informe presenta una visión detallada de cómo las tecnologías de Big Data pueden transformar el transporte público en Santiago, proporcionando datos y análisis que permitan la toma de decisiones informadas y la mejora continua del servicio.

II. Desarrollo

→ *Justificación*

La utilización de Big Data es fundamental para mejorar la eficiencia, planificación y experiencia del usuario en el transporte público. Al analizar algunos puntos claves del caso planteado, podemos inferir algunas razones técnicas y de funcionalidad, por las que sería beneficioso implementar una plataforma de Big Data en el contexto ya mencionado:

Técnicas

1. Escalabilidad y elasticidad: El transporte público implica la gestión de grandes volúmenes de datos, incluidos datos en tiempo real. Una arquitectura de big data en la nube permite escalar horizontalmente los recursos de computación y almacenamiento según sea necesario para manejar picos de carga y grandes volúmenes de datos de manera eficiente.

2. Costos: Implementar y mantener una infraestructura de big data on-premise puede ser costoso, ya que requiere inversiones significativas en hardware, software, mantenimiento y personal especializado. En cambio, una arquitectura de big data en la nube permite pagar sólo por los recursos que se utilizan, lo que puede resultar en costos más bajos y predecibles a lo largo del tiempo.

3. Flexibilidad y agilidad: La nube ofrece una mayor flexibilidad y agilidad en comparación con las implementaciones on-premise. Permite implementar y escalar rápidamente nuevos servicios y capacidades según sea necesario, lo que permite una respuesta más rápida a los cambios en los requisitos del negocio y las necesidades del usuario.

4. Integración de servicios: Las plataformas de big data en la nube suelen ofrecer una amplia gama de servicios y tecnologías avanzadas, como análisis de datos en tiempo real, aprendizaje automático, inteligencia artificial y herramientas de visualización de datos. Esto permite aprovechar las últimas innovaciones y capacidades sin tener que implementar y mantener estas tecnologías por separado.

Funcionales

1. Optimización de recursos y planificación: Con la información histórica de los viajes se pueden identificar patrones de demanda y uso de los diferentes medios de transporte. Esto permite generar información para quienes toman las decisiones y así optimizar la asignación de recursos, como la frecuencia de los servicios y la capacidad de los distintos medios de transporte.

2. Identificación de tendencias y variaciones: El análisis de datos históricos ayudará a identificar tendencias a largo plazo en la demanda de transporte público, así como variaciones temporales y estacionales. Esto permitirá a las autoridades anticipar cambios en la demanda y ajustar la oferta de transporte en consecuencia, evitando así problemas de congestión o subutilización de recursos.

3. Actualización en tiempo real: La automatización de la recopilación de datos diarios y la integración de información en tiempo real proporcionarán a las autoridades y a los usuarios acceso a datos actualizados sobre la oferta y la demanda de transporte público. Esto es crucial para garantizar que la información disponible sea siempre precisa y relevante para la toma de decisiones operativas y de planificación.

4. Mejorar la experiencia del usuario: Al tener acceso a información precisa y actualizada sobre los horarios, rutas y disponibilidad de transporte público, los usuarios podrán planificar sus viajes de manera más eficiente y evitar tiempos de espera innecesarios.

5. Detección de cambios en la oferta de transporte: La plataforma de datos permitirá identificar qué zonas han experimentado cambios en la disponibilidad o variabilidad de los recorridos de transporte público. Esto incluye la detección de nuevas rutas agregadas, recorridos eliminados o modificaciones en los horarios de servicio.

En consecuencia de los puntos mencionados con anterioridad, una arquitectura de big data en la nube ofrece escalabilidad, costos más bajos, flexibilidad, agilidad, disponibilidad, confiabilidad y acceso a tecnologías avanzadas, lo que la hace una opción atractiva para el caso aplicado del transporte público en comparación con las implementaciones on-premise.

→ *Herramientas de Big Data*

Herramientas a Utilizar

Apache Kafka:

Justificación: Apache Kafka es un sistema de mensajería que permite la ingestión de datos en tiempo real. Es adecuado para recibir datos en tiempo real de la API de transportes y distribuirlos para su procesamiento.

Ámbito de Aplicación: Ingesta y transmisión de datos en tiempo real.

Apache Hive:

Justificación: Apache Hive es adecuado para el almacenamiento y análisis de grandes volúmenes de datos mediante consultas SQL. En este proyecto, almacenará y analizará los datos históricos y en tiempo real del transporte público.

Ámbito de Aplicación: Almacenamiento y análisis de grandes volúmenes de datos.

Apache Spark:

Justificación: Apache Spark es ideal para el procesamiento en tiempo real y por lotes de datos. En este proyecto, manejará tanto los datos históricos como los datos diarios de transporte público. Su capacidad de escalabilidad y procesamiento distribuido es clave.

Ámbito de Aplicación: Procesamiento y transformación de datos en tiempo real y por lotes.

Herramientas que no utilizaremos:

Hadoop:

Justificación: Aunque Hadoop es una potente plataforma de almacenamiento y procesamiento de datos, Apache Spark y Apache Hive pueden manejar eficientemente las necesidades del proyecto sin la complejidad adicional de configurar y mantener clústeres de Hadoop.

Por qué no: La complejidad de gestión y la necesidad de un enfoque más moderno y manejable hacen que Hadoop no sea la mejor opción.

→ *Gobierno de Datos*

Ciclo de vida del Dato

El gobierno de datos implica establecer políticas y regulaciones para guiar el ciclo completo de vida de los datos, desde su recopilación y almacenamiento hasta su procesamiento y eliminación. Estas políticas definen quién puede acceder a qué datos y qué normas deben seguirse para garantizar su seguridad, privacidad y calidad. Además, el gobierno de datos implica cumplir con estándares establecidos por las entidades correspondientes para garantizar que el transporte público opere de manera segura, eficiente y confiable.

Etapas del ciclo de vida de los datos:



1. Generación o captura:

¿Quién captura el dato?

Apache Kafka podría utilizarse para recopilar información en tiempo real de los sensores ubicados en los vehículos, sistemas de boletería, o aplicaciones móviles, esto garantiza la disponibilidad de los datos y la entrega en tiempo real a los sistemas y aplicaciones que lo necesiten.

¿Cómo se captura el dato?

Los datos se recopilan desde diversas fuentes, como sensores, bases de datos de la plataforma de datos abiertos del Gobierno de Chile, desde las API ubicadas en www.red.cl.

¿Quién es el dueño del dato?

El Gobierno de Chile

¿Cómo se asegura la calidad en origen?

Con el marco de referencia del gobierno de datos

2. Almacenamiento:

¿Cuánto tiempo se debe mantener?

La planificación de los distintos medios de transportes en Santiago están disponibles por 15 días. La información de cada recorrido se actualiza diariamente.

¿Dónde se almacena el dato?

En Apache Hadoop y Cloud Dataproc de Google Cloud.

¿Existe alguna política de almacenamiento?

Almacenamiento y gestión de datos del Marco de Referencia para la Gobernabilidad de los Datos

3. Modificación:

¿Por cuál canal se puede modificar?

A través de la plataforma <https://datos.gob.cl/>

¿Quién puede modificar el dato?

En la plataforma https://datos.gob.cl/ ,los usuarios registrados como administradores

¿Cómo se genera la modificación?

Iniciando sesión como administrador

4. Uso:

¿Quiénes usan el dato?

Todos los usuarios que tengan acceso a la plataforma <https://datos.gob.cl/> y quienes accedan a las APIs alojadas en https://www.red.cl/

¿Cuáles son los usos del dato?

Los datos son utilizados para obtener información sobre los viajes en el transporte público de Santiago

¿Dónde se registran los usos?

No se entrega información

¿Cómo se asegura la calidad?

Con el marco de Referencia para la Gobernabilidad de los Datos

5. Eliminación o Reutilización:

¿Quién decide la eliminación de un dato?

En https://datos.gob.cl/ usuarios registrados como administradores

¿Se mantiene algún registro?

Sí

¿Por cuál canal se puede eliminar?

Por la plataforma <https://datos.gob.cl/>

DMBOK: Marco de Referencia para la Gobernabilidad de los Datos.

DMBOK (Data Management Body Of Knowledge), Es un marco o conjunto de principios, prácticas y directrices que se utilizan para estandarizar y mejorar la gestión de datos. Busca unificar conceptos, buenas prácticas y ser una referencia sobre la gestión de datos. Propone 11 áreas específicas de conocimiento que cubren todos los aspectos importantes dentro del ciclo de vida de los datos.



Arquitectura de Datos:

La arquitectura de datos en el transporte público implica diseñar sistemas y estructuras eficientes, escalables y flexibles para manejar la recopilación, procesamiento y distribución de datos. Esto incluye seleccionar las tecnologías adecuadas, para almacenar grandes volúmenes de datos y garantizar su accesibilidad y disponibilidad, respondiendo a los requerimientos presentes y futuros de los datos del transporte público. Además, debe soportar todas las etapas del ciclo de vida de los datos, desde la recopilación en tiempo real hasta la distribución a través de diversos canales.

Modelado de Datos y Diseño:

El modelado de datos es el proceso de descubrir, analizar y alcanzar los requerimientos de datos para luego representar y comunicar estos requerimientos

en un modelo de datos, el cual puede ser conceptual, lógico y físico. El modelado de datos es fundamental para manejar el gran volumen de datos en el transporte público tales como horarios, rutas, buses, etc, con el propósito de descubrir y documentar cómo estos datos interactúan y se acoplan en conjunto, permitiendo un mejor entendimiento de estos activos

Almacenamiento de Datos y Operaciones:

Gestiona la disponibilidad de datos a través de los ciclos de vida, esto implica asegurarse de que los datos de los horarios de los buses, del metro, los trayectos, las paradas, los horarios y duración de los trayectos, estén accesibles en todo momento para los usuarios y los operadores del sistema.

Asegurar la integridad de los activos de datos en el transporte público, los cuales deben ser precisos y confiables, que estén actualizados y no hayan sido manipulados de manera no autorizada. También se busca que los datos se registren evitando pérdidas de datos o inconsistencias en el sistema.

Seguridad de Datos:

Incluye la planificación, desarrollo y ejecución de las políticas de seguridad y procedimientos para proporcionar la correcta autenticación, autorización, acceso y auditoría de los datos.

Permite cumplir con las regulaciones y políticas relacionadas a la privacidad, protección y confidencialidad y asegura que las necesidades de privacidad y confidencialidad de todas las partes interesadas estén impuestas y auditadas.

Integración de Datos e Interoperabilidad:

Describe procesos relacionados al movimiento y consolidación de datos dentro y entre almacenes de datos, aplicaciones y organizaciones. Consolida los datos en forma coherentes, ya sean físicas o virtuales. La interoperabilidad de datos es la habilidad de que varios sistemas se comuniquen.

En el caso del transporte público, se requiere una gestión eficiente de los datos para planificar adecuadamente los viajes. Para ello, se recopilan datos históricos y diarios de múltiples fuentes, como el sitio de datos abiertos del Gobierno de Chile y APIs específicas. Estos datos se consolidan en una plataforma centralizada para su análisis. Es importante mantener la información actualizada mediante tareas automatizadas. La interoperabilidad de datos facilita la comunicación entre

diferentes sistemas y fuentes, garantizando una visión completa y actualizada del sistema de transporte público.

Gestión de Documentos y Contenido:

Implica cumplir con las obligaciones legales y expectativas del cliente con respecto a la gestión de registros, asegurar el efectivo y eficiente almacenamiento, recuperación y uso de los datos y asegurar las capacidades de integración entre los datos estructurados y no estructurados.

Datos Maestros y de Referencia:

Gestiona datos compartidos para conocer objetivos organizacionales claves del servicio, tales como horarios, rutas, disponibilidad de vehículos y demanda de los usuarios. También permite reducir riesgos asociados con la redundancia de datos al proporcionar una única fuente confiable y actualizada de datos compartidos que se utiliza en todos los sistemas y aplicaciones relacionadas con el transporte público. Asimismo asegura la calidad de los datos al establecer estándares y procesos para la captura, validación y mantenimiento de los datos. Con lo anterior, se logra reducir los costos de la integración de datos al proporcionar una base de datos compartida y estandarizada.

Data Warehousing e Inteligencia de Negocio:

Un data warehouse permitirá integrar los datos recopilados de diversas fuentes en un repositorio centralizado, proporcionando una vista unificada de la información relacionada con los servicios de transporte.

Los datos almacenados en el data warehouse serían utilizados para apoyar las funciones operativas del sistema de transporte público, como la planificación de rutas, la asignación de recursos y la programación de horarios.

Al tener acceso a datos históricos y en tiempo real, los operadores pueden tomar decisiones más informadas y eficientes para mejorar la calidad del servicio.

En el transporte público, la inteligencia de negocio podría utilizarse para identificar tendencias de viaje, optimizar la distribución de recursos y mejorar la satisfacción del cliente. A la vez permitiría a los analistas tomar decisiones informadas y basadas en los datos, mejorando en todos los aspectos el sistema de transporte.

Gestión de Metadatos

La gestión de metadatos es esencial para organizar y aprovechar eficientemente la información recopilada. El Diccionario de Datos, que incluye tanto metadatos de negocios como técnicos, así como la data de trazabilidad de cada elemento de datos, se genera a partir de estos metadatos. Este proceso implica planificar y documentar meticulosamente los datos, incluyendo su origen, formato y relevancia operativa. Además, se establecen estándares y normativas para garantizar la coherencia y calidad de los datos, lo que facilita su integración en los diversos sistemas y aplicaciones del transporte público.

Se llevan a cabo controles de calidad para asegurar la precisión y confiabilidad de los metadatos, actualizándose según sea necesario. Esto permite un acceso integrado a los metadatos, simplificando la búsqueda, recuperación y comprensión de la información, lo que mejora la eficiencia y la toma de decisiones en el sistema de transporte público.

Gestión de Calidad de Datos

La gestión de calidad de datos en el caso del transporte público se relaciona con la utilización efectiva de la información recopilada para lograr objetivos estratégicos, asegurando que los datos sean diseñados, almacenados, accedidos, compartidos y utilizados de manera segura y adecuada para satisfacer las necesidades del negocio.

→ Arquitectura GCP

1. Estilos de arquitectura

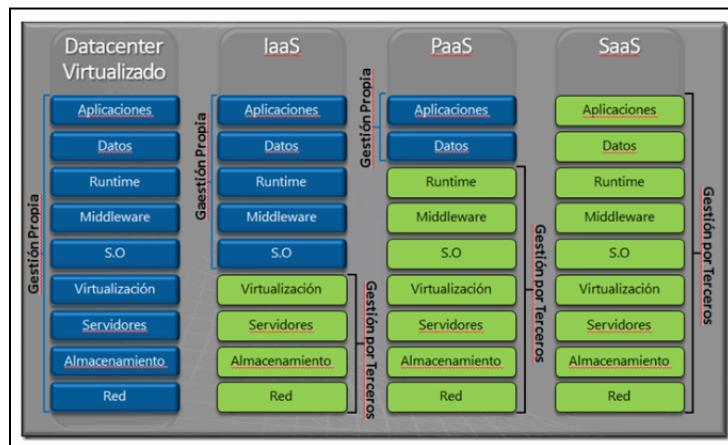
Antes de realizar la implementación de arquitectura al caso, es importante analizar dentro de los estilos de arquitectura cual utilizaremos:

Estilos de Arquitectura
Monolítica
Orientada a servicios (Service Oriented Architecture)
Dirigida por eventos (Event Driven Architecture)
Dirigida por datos (Data Driven Design)
Orientada a Microservicios
Orientado a Dominios (Domain Driven Design)

En el caso del transporte público y la implementación de la plataforma de datos en Google Cloud Platform (GCP), se puede identificar un enfoque de arquitectura dirigida por datos. Este enfoque se caracteriza por centrarse en los datos siendo ellos el centro de atención, lo que serán utilizados como el principal motor para la toma de decisiones.

2. Tipos de Infraestructura de Big Data

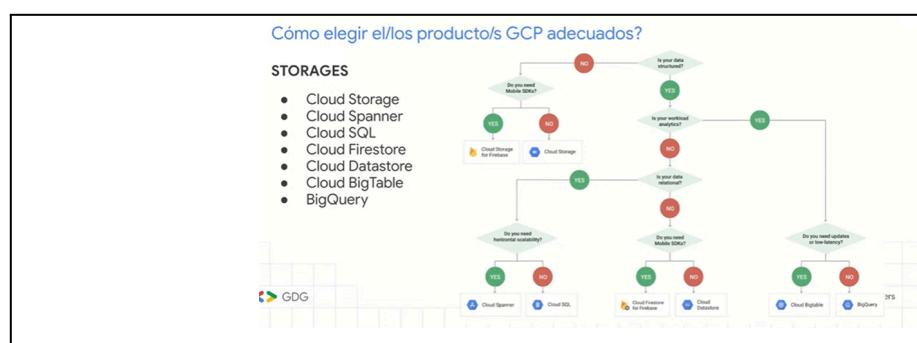
Luego se debe definir qué tipo de infraestructura de Big Data a elegir.



La infraestructura adecuada para este caso sería una combinación de PaaS (Plataforma como Servicio) y SaaS (Software como Servicio), utilizando servicios de nivel de plataforma para el procesamiento y análisis de datos (PaaS), y herramientas de software como servicio para la visualización de datos (SaaS). Esto permitiría aprovechar las ventajas de la escalabilidad, la disponibilidad y la facilidad de uso ofrecidas por los servicios gestionados de GCP.

3. Árboles de decisión

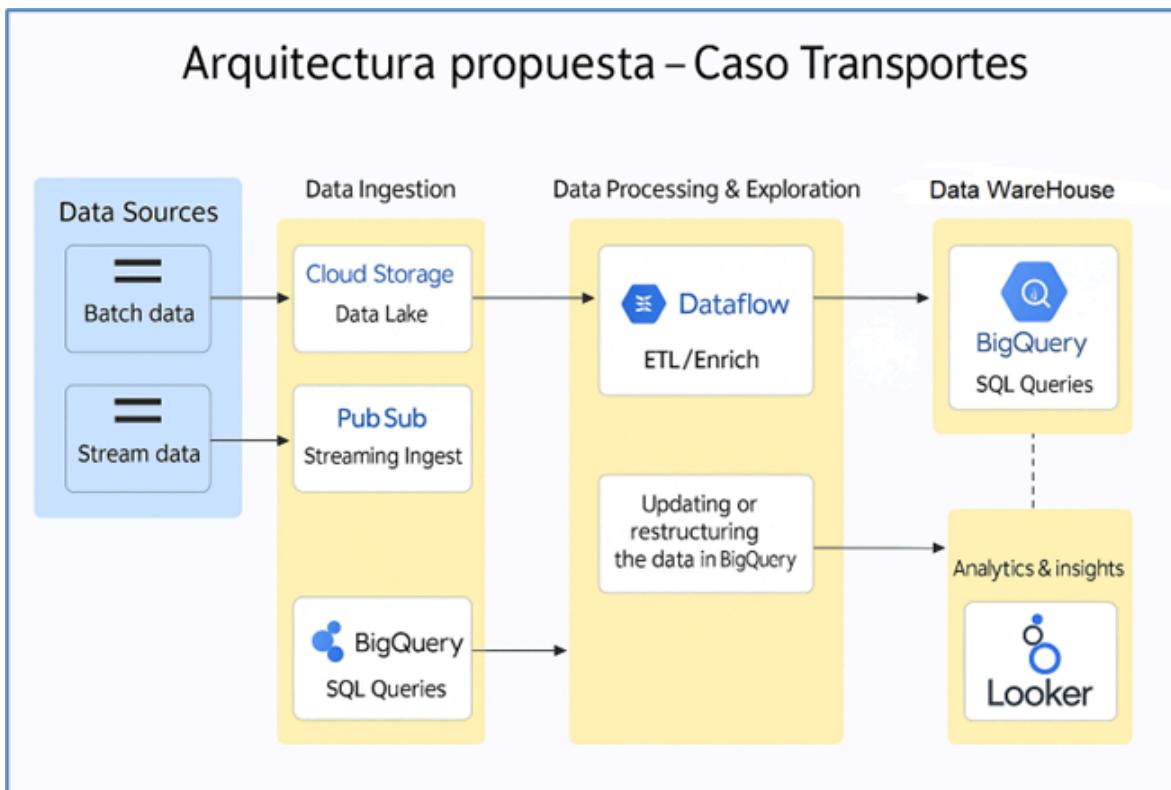
Luego debemos considerar árboles de decisión para la elección de los productos GCP adecuados. Por ejemplo, aplicamos este árbol de decisión para los storages disponibles:



Cloud Storage es la opción ideal para almacenar los datos tipo JSON en GCP. Además, necesitamos un almacenamiento de datos altamente duradero y escalable. Cloud Storage nos proporcionará la durabilidad, disponibilidad y rendimiento necesarios para almacenar y acceder a los datos de manera eficiente y segura.

4. Arquitectura de referencia GCP propuesta

A continuación se muestra para este caso, la arquitectura de referencia GCP propuesta con las explicaciones correspondientes.



Fuentes de datos	Ingesta de datos	Procesamiento y exploración de datos	Almacén de datos	Visualización de datos
<p>Batch Data: Procesamiento por lotes de un conjunto de datos grandes y limitados que no es necesario procesar en tiempo real</p> <p>Por lo general, se ingieren en frecuencias regulares específicas y todos los datos llegan a la vez o no llegan en absoluto</p> <p><u>Datos Históricos:</u> Set de datos que contiene la información mensual de la planificación de los distintos medios de transporte en Santiago (15 de cada mes)</p>	<p>Cloud Storage (Data Lake) Para casos de uso por lotes, Cloud Storage es el lugar recomendado para descargar los datos entrantes. Es un servicio de almacenamiento de objetos duraderos, de alta disponibilidad y rentable. Las empresas pueden mover datos sin procesar por medio de lotes o transmisiones a un data lake sin tener que transformarlos.</p>	<p>DataFlow: Servicio completamente gestionado que soporta procesamiento unificado de datos por lotes y de transmisión rápida, sin servidores y rentable.</p> <p>Utiliza Apache Beam y Spark como plataformas de procesamiento de datos. DataFlow, es más flexible y eficiente para nuevos desarrollos.</p>	<p>Big Query: Almacén de datos procesados completamente administrado que ayuda a administrar y analizar los datos con funciones integradas como el aprendizaje automático e inteligencia empresarial integradas en una plataforma unificada.</p>	<p>Data Studio: Ofrece exploraciones potentes, datos más actualizados y filtros más rápidos:</p> <p>Producto de autoservicio de visualización de datos e inteligencia empresarial</p>
<p>Stream Data: Flujo continuo de datos en tiempo real.</p> <p><u>Datos Diarios:</u> Todos los recorridos disponibles con la información de su trayecto, horarios y paradas.</p>	<p>Pub/Sub: Es un servicio de mensajería totalmente administrado diseñado para la ingestión de datos en tiempo real. Se integra directamente con los servicios de procesamiento de datos.</p>			

En el contexto del caso del transporte público y la implementación de la plataforma de datos en Google Cloud Platform (GCP), podría ser necesario considerar lo siguiente:

Elementos que podrían faltar:

- **Servicios de Machine Learning:** Podrían ser útiles para realizar análisis predictivo o detectar patrones en los datos, pero no son esenciales para los objetivos principales del proyecto.
- **Servicios de Orquestación:** Podrían servir para coordinar y administrar los diversos componentes de nuestro flujo de trabajo.

Elementos que podrían no ser requeridos:

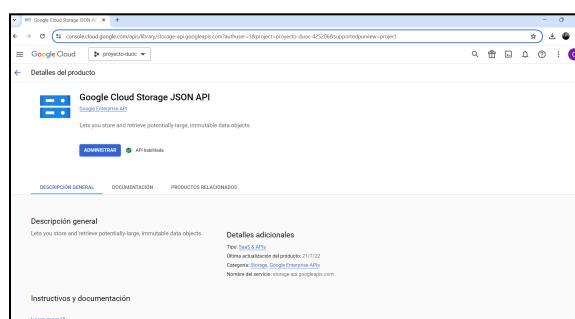
- **Servicios de almacenamiento de datos de alto rendimiento:** Si los volúmenes de datos no son extremadamente grandes y el rendimiento no es una preocupación crítica, servicios como Bigtable o Spanner podrían ser excesivos y no requeridos.

En conclusión, el proyecto involucra actividades típicas de un proyecto de Data Science, como la recopilación y limpieza de datos, análisis estadístico, modelado predictivo y visualización de datos, con el objetivo de generar insights que puedan ser utilizados para la toma de decisiones.

→ *Proceso Batch*

Paso 1: Conectar y descargar datos en Cloud Storage

- Configuramos el entorno de trabajo: Nos aseguramos de tener habilitada la API de Cloud Storage. Se siguen los mismos pasos que con la API de DataFlow en donde se muestra en detalle.



- Creación y configuración del Bucket en Cloud Storage

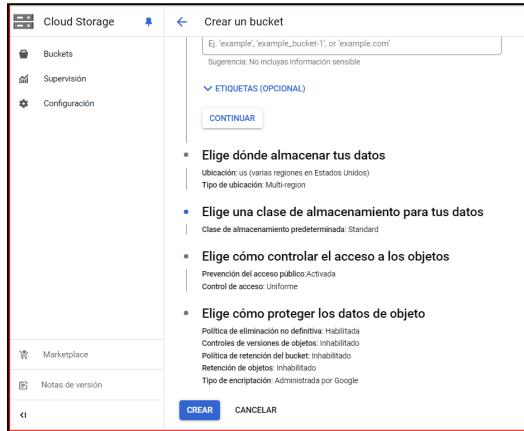
-> **Buckets -> Crear**

Se debe configurar lo siguiente:

- 1.- Asigna nombre

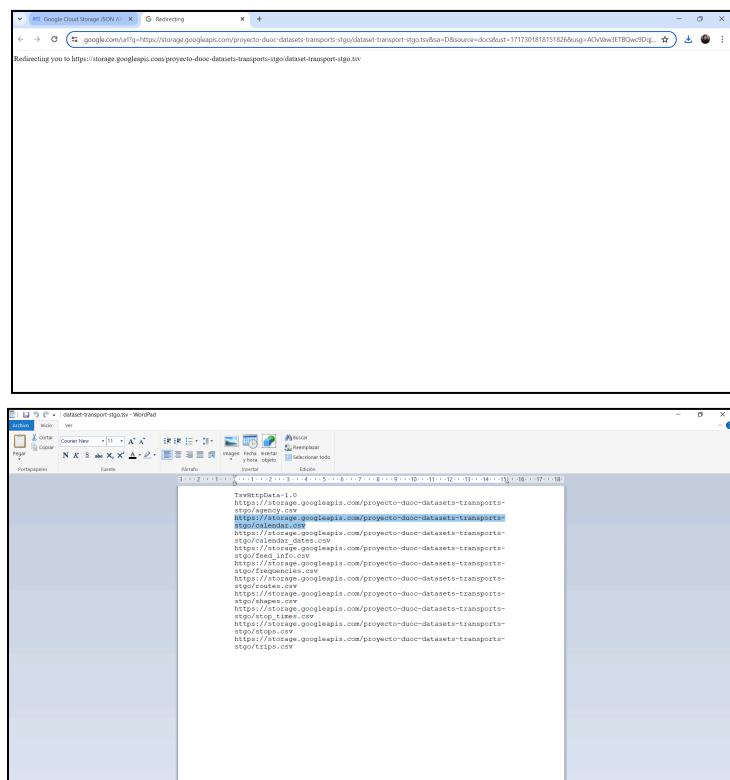
- 2.- Elegir el lugar donde almacenar tus datos
- 3.- Elegir una clase de almacenamiento para los datos

4.- Elegir cómo proteger los datos del objeto.

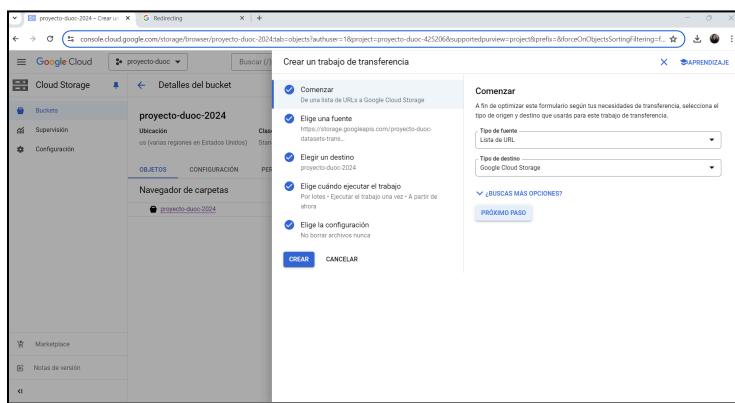


- Realizamos las conexiones con la fuente de origen de datos, desde internet (URL).

Fuente de datos que disponemos del Gobierno, corresponde a un link que contiene un archivo de tipo .tsv en donde se encuentran los links de los 10 csv que contienen la información correspondiente a los datos de transporte:
<https://storage.googleapis.com/proyecto-duoc-datasets-transports-stgo/dataset-transport-stgo.tsv>



- Transferimos los datos (importación al bucket). Ingestamos los 10 archivos CSV.



The screenshot displays two Google Cloud interfaces side-by-side.

Left Panel (Storage):

- Header: proyecto-duoc -> Create new > Redirecting
- Left sidebar: Google Cloud, Cloud Storage, Buckets, Supervisión, Configuración, Marketplace, Notas de versión.
- Bucket details for "proyecto-duoc-2024":
 - Ubicación: Ia (varios registros en Estados Unidos)
 - Opciones: Crear carpeta, Borrar, Borrar todo, Propiedades.
- OBJETOS: Navegador de carpetas, showing a single item "proyecto-duoc-2024".
- BOTONES: CREAR, CANCELAR.

Right Panel (Transfer):

- Header:Crear un trabajo de transferencia > APRENDIZAJE
- Left sidebar: Google Cloud, proyecto-duoc, Buscar (proyecto-duoc).
- Form: "Comenzar" (Start) with "Usa una lista de URLs a Google Cloud Storage".
- Form: "Elige una fuente" (Choose source) with URL: https://storage.googleapis.com/proyecto-duoc-datasets-trans.
- Form: "Elige un destino" (Choose destination) with "proyecto-duoc-2024".
- Form: "Elige cuando ejecutar el trabajo" (Choose when to run the job) with "Hoy a las 10:00 - Ejecutar el trabajo una vez" (Run the job once).
- Form: "Elige la configuración" (Choose configuration) with "No borrar archivos vacíos".
- BOTONES: EXPLORAR, PRÓXIMO PASO.

Bottom Panel (Transfer Job Details):

- Header: Detalles del trabajo > Transfiriendo de atm
- Job ID: 451472452704679821
- Información del trabajo:
 - Nombre completo del trabajo: transfer-job-451472452704679821
 - versión: 1
 - Destino: Google Cloud Storage
 - Estado: En ejecución
 - Método de programación: Batch (no recurrente)
- Operaciones: INICIAR UNA EJECUCIÓN, BORRAR TRABAJO, COPIAR TRABAJO, INHABILITAR TRABAJO.
- Summary:
 - Información de la fuente: Lista de URLs, https://storage.googleapis.com/proyecto-duoc-datasets-trans-etc/destin/transport-etc/tes.
 - Destino: Google Cloud Storage, proyecto-duoc-2024.
- Panel: Panel Preferidos, Filtros: 1 hora, 4 horas, 12 horas, 1 día, 4 días, 7 días (checked), 14 días, 30 días, 6 semanas.
- Summary of bytes copied: 60000, Summary of objects copied: 20.
- Summary of bytes copied by hour: UTC+4, 27 May, 28 May, 29 May, 30 May, 31 May, 1 Jun, 2 Jun, 3 Jun.
- Summary of objects copied by hour: UTC+4, 27 May, 28 May, 29 May, 30 May, 31 May, 1 Jun, 2 Jun, 3 Jun.
- Bandwidth of bytes copied: 2.000mbps, Rate of objects copied: 0.22s.

➤ Datos cargados en el bucket de Cloud Storage

The screenshot shows the Google Cloud Storage console with the 'transfencia de alm' bucket selected. The left sidebar includes 'Buckets', 'Supervisión', and 'Configuración'. The main area displays bucket details like 'Ubicación' (varias regiones en Estados Unidos), 'Standard' storage class, and 'No público' visibility. A 'transfencia de alm' folder is visible under 'OBJETOS'. The 'PROTECCIÓN' tab is active, showing 'Depósitos' (proyecto-duo-2024) and 'storage.googleapis.com' as sources. The 'CICLO DE VIDA' tab lists rules for 'SUBIR ARCHIVOS', 'SUBIR CARPETA', 'CREAR CARPETA', 'EDITAR LA RETENCIÓN', 'DESCARGAR', and 'BORRAR'. The 'OPERACIONES' tab has links for 'INFORMES DE INVENTARIO' and 'ADMINISTRAR CONSERVACIONES'. A large table at the bottom lists objects with columns: Nombre, Tamaño, Tipo, Fecha de creación, Clase de almacenamiento, and Última. Objects listed include agency.csv, calendar.csv, calendar_dates.csv, feed_info.csv, frequencies.csv, routes.csv, shapes.csv, stop_times.csv, stops.csv, and trips.csv.

Paso 2: Construir procesos de limpieza, transformación y carga con DataFlow

El siguiente paso es configurar un pipeline en DataFlow para luego realizar los procesos de limpieza, transformaciones necesarias y carga al modelo de datos final.

- Configurar el entorno de trabajo: Nos aseguramos de tener habilitada la API de DataFlow.

The figure consists of four screenshots from the Google Cloud Platform interface:

- Screenshot 1:** Shows the "Biblioteca de APIs" (API Library) search results for "Api dataflow". It lists the "Dataflow API" and the "Datastream API". The Dataflow API is selected, showing its description: "Manages Google Cloud Dataflow projects on Google Cloud Platform".
- Screenshot 2:** Shows the "Detalles del producto" (Product Details) page for the Dataflow API. It shows the API is enabled ("API habilitada").
- Screenshot 3:** Shows the "RPT API y servicios" (Report API and Services) dashboard. It displays three metrics: "Tráfico" (Traffic), "Errores" (Errors), and "Mediana de latencia" (Median Latency). Both traffic and errors graphs show no data available for the selected period.
- Screenshot 4:** Shows the "Detalles del servicio o la API" (Service or API details) page for the Dataflow API. It shows the service is public and enabled. It includes tabs for "METRICAS" (Metrics), "CUOTAS Y LÍMITES DEL SISTEMA" (System Quotas and Limits), "CREDENCIALES" (Credentials), and "COSTO" (Cost).

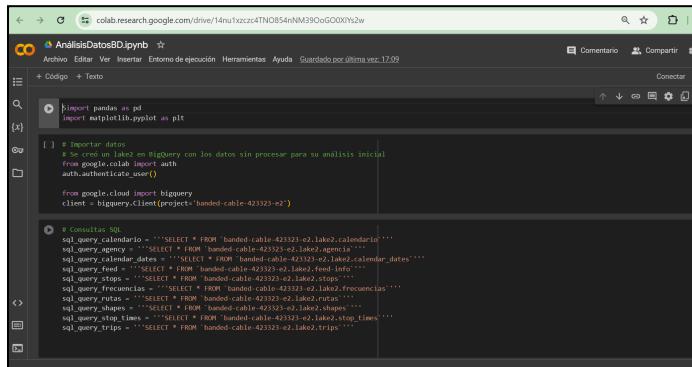
The first screenshot shows a modal dialog titled "¿Quieres inhabilitar Dataflow API?" (Do you want to disable the Dataflow API?). It contains a warning message: "Si Dataflow API creó algún recurso, es posible que se borre poco después de que se inhabilite Dataflow API." Below the message are two buttons: "CANCELAR" and "INHABILITAR". A timer at the top right indicates "30 días" (30 days). The second screenshot shows the main "Dataflow API" page with tabs for "HABILITAR" (Enable) and "PROBAR ESTA API" (Test this API). The third screenshot shows the "Detalles del servicio o la API" (Service or API details) page for the Dataflow API.

This screenshot shows the "Detalles del servicio o la API" (Service or API details) page for the Dataflow API. It includes sections for "Descripción general" (General description), "Detalles adicionales" (Additional details), and "CREDENCIALES" (Credentials). A note says "Es posible que necesites credenciales para usar esta API." (It's possible that you need credentials to use this API.) A "CREA CREDENCIALES" (Create credentials) button is visible.

➤ Creamos el conjunto de datos “lake” en BigQuery

Inicialmente creamos un dataset de prueba, en el cual creamos las tablas desde los archivos CSV. En el dataset de prueba examinamos las tablas, verificando el tipo de datos, realizamos consultas SQL y lo conectamos a un notebook de Google Colab para visualizar mejor la información relevante de las

tablas, tales como el tipo de dato, número de filas, número de columnas y los datos faltantes. Se adjunta pdf de respaldo “Análisis Inicial Dataset”.



```

import pandas as pd
import matplotlib.pyplot as plt

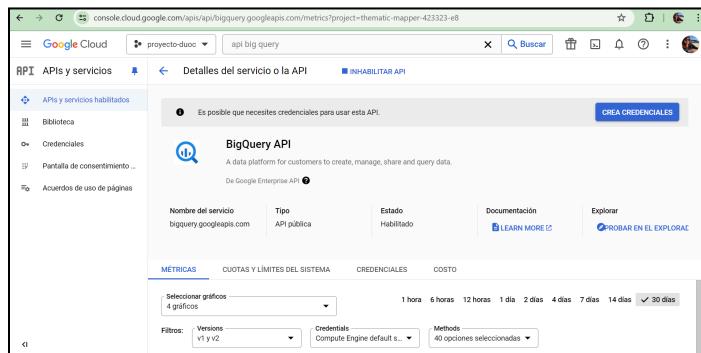
# Importar datos
# Se crea un lake en BigQuery con los datos sin procesar para su análisis inicial
from google.cloud import bigquery
client = bigquery.Client(project='banded-cable-423323-e2')

# Consultas SQL
sql_query_calendario = """SELECT * FROM `banded-cable-423323-e2.lake2.calendario`"""
sql_query_agency = """SELECT * FROM `banded-cable-423323-e2.lake2.agency`"""
sql_query_calendar_dates = """SELECT * FROM `banded-cable-423323-e2.lake2.calendar_dates`"""
sql_query_feed = """SELECT * FROM `banded-cable-423323-e2.lake2.feed_info`"""
sql_query_stops = """SELECT * FROM `banded-cable-423323-e2.lake2.stops`"""
sql_query_rutas = """SELECT * FROM `banded-cable-423323-e2.lake2.rutas`"""
sql_query_shapes = """SELECT * FROM `banded-cable-423323-e2.lake2.shapes`"""
sql_query_stop_times = """SELECT * FROM `banded-cable-423323-e2.lake2.stop_times`"""
sql_query_trips = """SELECT * FROM `banded-cable-423323-e2.lake2.trips`"""

```

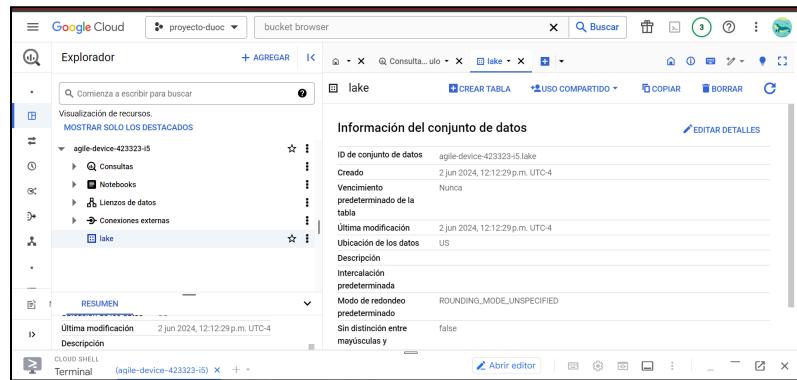
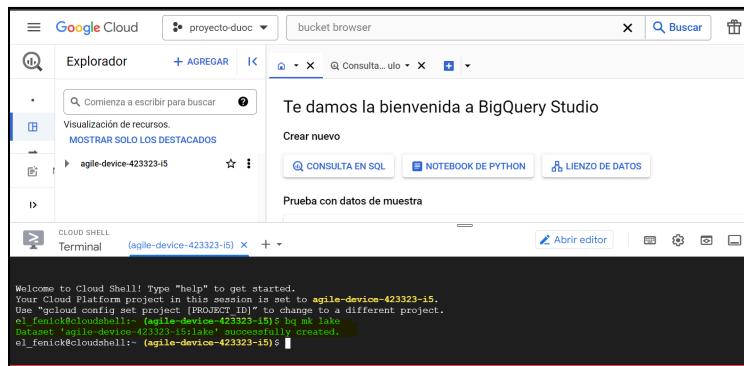
Posteriormente, creamos un conjunto de datos en BigQuery llamado ‘lake’. La importancia de crearlo en esta parte del proceso, es porque debemos tener listo el destino final de los datos que serán transformados, ya que luego de que DataFlow ejecute el código Python con las transformaciones, se moverán los datos preparados al lake ubicado en BigQuery.

- Configurar el entorno de trabajo: Nos aseguramos de tener habilitada la API de BigQuery. Se siguen los mismos pasos que con la API de DataFlow en donde se muestra en detalle.

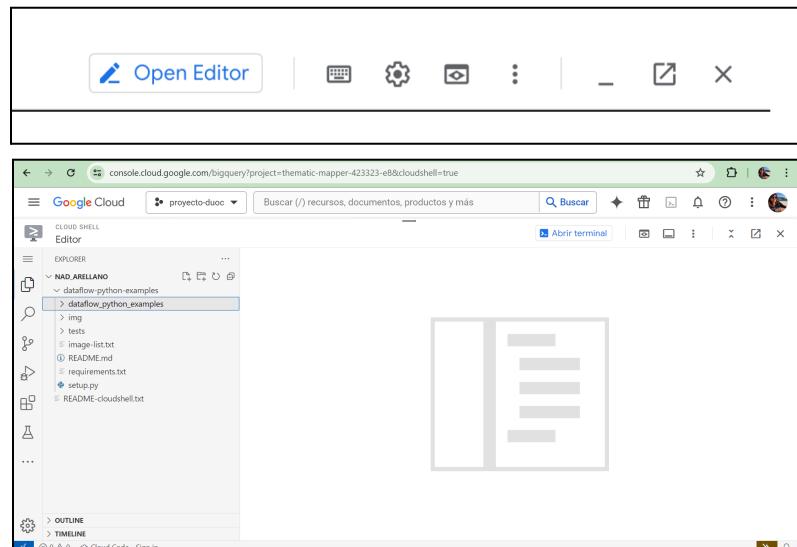


- Creamos el conjunto de datos “lake” en BigQuery.

En *Cloud Shell*, creamos un conjunto de datos en BigQuery llamado “lake” en donde cargaremos todas las tablas. Script: **bq mk lake**



- Desarrollar el script de DataFlow (Python): Abrimos el Editor de Código ubicado en la Cloud Shell y creamos una carpeta llamada `dataflow_python_examples` en donde guardaremos los archivos python.



Creamos 5 archivos Python para definir los pipelines de cada transformación. Se adjuntan en la carpeta de entrega “Códigos Fuente .py”.

Cada uno de los códigos realizan lo siguiente:

1. **data_setear_nulos:** Setea con una cadena de STRING que indica: "Sin Información" todos los datos nulos o vacíos de las columnas: 'stop_code' y 'stop_url' (csv: "stops") y las columnas: 'route_desc' y 'route_url' (csv "routes"). Decidimos realizar este seteo porque la cantidad de datos nulos era mayor a 80% aproximadamente por lo que no teníamos una referencia de la mediana o la media para imputarlos con ese valor por ejemplo. Además, decidimos no eliminar las columnas para evitar conflictos en las futuras ingestas de datos.
2. **data_ingestion_sin_modif:** Se crea un pipeline con los csv en donde no fue necesario realizar transformaciones.
3. **data_eliminar_head:** Elimina el encabezado del csv "agency" porque no corresponde a los encabezados originales, los que se encontraban en la primera fila (imagen de referencia como ejemplo).

Tabla 'Agency' antes de la transformación

Fila	string_field_0	string_field_1	string_field_2	string_field_3
1	agency_id	agency_name	agency_url	agency_timezone
2	RM	Red Metropolitana de Movil...	http://www.red.cl	America/Santiago
3	M	Metro de Santiago	http://www.metro.cl	America/Santiago
4	MT	EFE Trenes de Chile	http://www.efe.cl	America/Santiago
5	BAA	Bus de Acercamiento Aero...	http://www.nuevopudahuel....	America/Santiago

Tabla 'Agency' después de la transformación

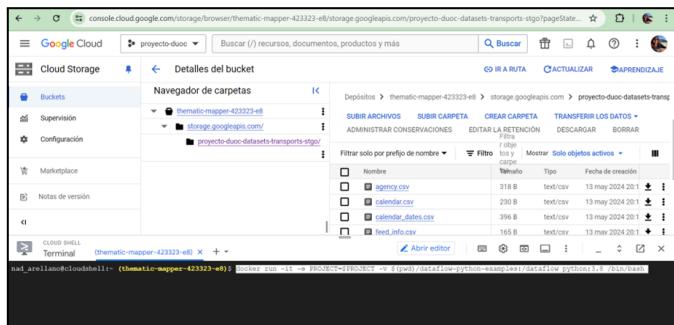
Fila	agency_id	agency_name	agency_url	agency_timezone
1	RM	Red Metropolitana de Movilidad	http://www.red.cl	America/Santiago
2	M	Metro de Santiago	http://www.metro.cl	America/Santiago
3	MT	EFE Trenes de Chile	http://www.efe.cl	America/Santiago
4	BAA	Bus de Acercamiento Aeropuer...	http://www.nuevopudahuel.cl	America/Santiago

4. **data_convertir_time:** Convierte los datos de la columna "end_time" (csv frequencies) desde STRING a TIME.

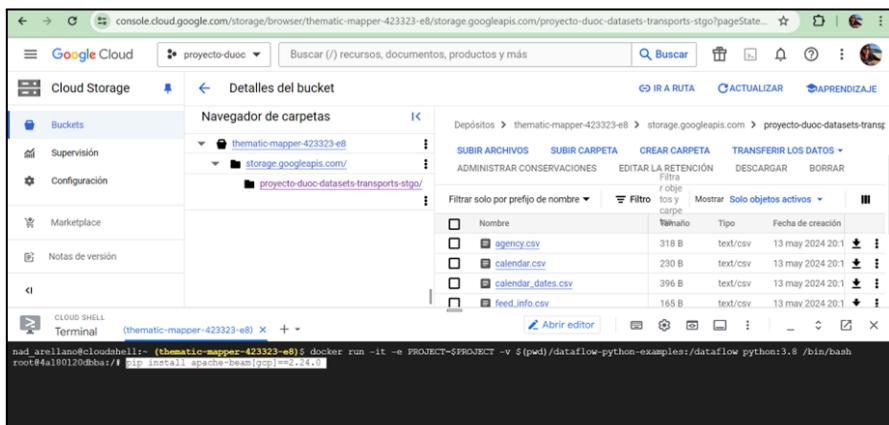
5. **data_convertir_date:** Convierte los datos de la columna “date” (csv calendar_dates), columnas “start_date” y “end_date” (csv calendar), columnas “feed_start_date” y “feed_end_date” (csv feed_info) desde INTEGER a DATE.

➤ Ejecución del pipeline: Ejecutamos los scripts de Python en DataFlow.

En este paso antes de ejecutar los scripts de python iniciamos un contenedor de Docker Python 3.8.



Luego, instalamos apache-beam en este contenedor en ejecución.



Finalmente, en el contenedor en ejecución en Cloud Shell, cambiamos al directorio en el que vinculamos el código fuente.

```

Success! Installed apache-beam-2.24.0 avro-python3-1.10.1 cachetools-3.1.1 certifi-2024.6.1 charset-normalizer-1.1.0 cromecol-1.7.1 datasets-0.1.0 docopt-0.6.2 fa
t-3.2.3 google-cloud-core-1.7.3 google-cloud-datastore-1.15.5 google-cloud-dlp-1.0.2 google-cloud-language-1.1.2 google-cloud-pubsub-1.7.2 google-cloud-spanner-1.19.
3 google-cloud-video-intelligence-1.16.3 google-cloud-vision-1.0.2 google-crc32c-1.5.0 google-renameable-media-1.3.3 googleapis-common-protos-1.63.0 grpc-google-ia
m-v1-0.12.7 grpcio-1.64.0 grpcio-gcp-0.2.2 grpcio-status-1.48.2 hdfs-2.7.3 httpbin2-0.17.4 idna-3.7.2 mock-2.0.0 numpy-1.24.4 oauth2client-3.0.0 phr-6.0.0 protobuf
-9 six-1.16.0 typing-3.1.4 urllib3-2.2.1
WARNING: Running pip as the 'root' user can result in broken permissions and conflicting behaviour with the system package manager. It is recommended to use a vi
rtual environment instead: https://pip.pypa.io/warnings/venv
[...]
A new release of pip is available: 23.0.0 > 24.0
[...]
To update, run: pip install --upgrade pip
root@#al80120dbba:/# cd dataflow/
root@#al80120dbba:/dataflow#

```

Tratamiento de CSVs: Decidimos que como los CSVs deben ser transformados de manera diferente, agruparemos y cargaremos los resultados en distintas tablas en BigQuery.

Comenzamos la ejecución de la canalización.

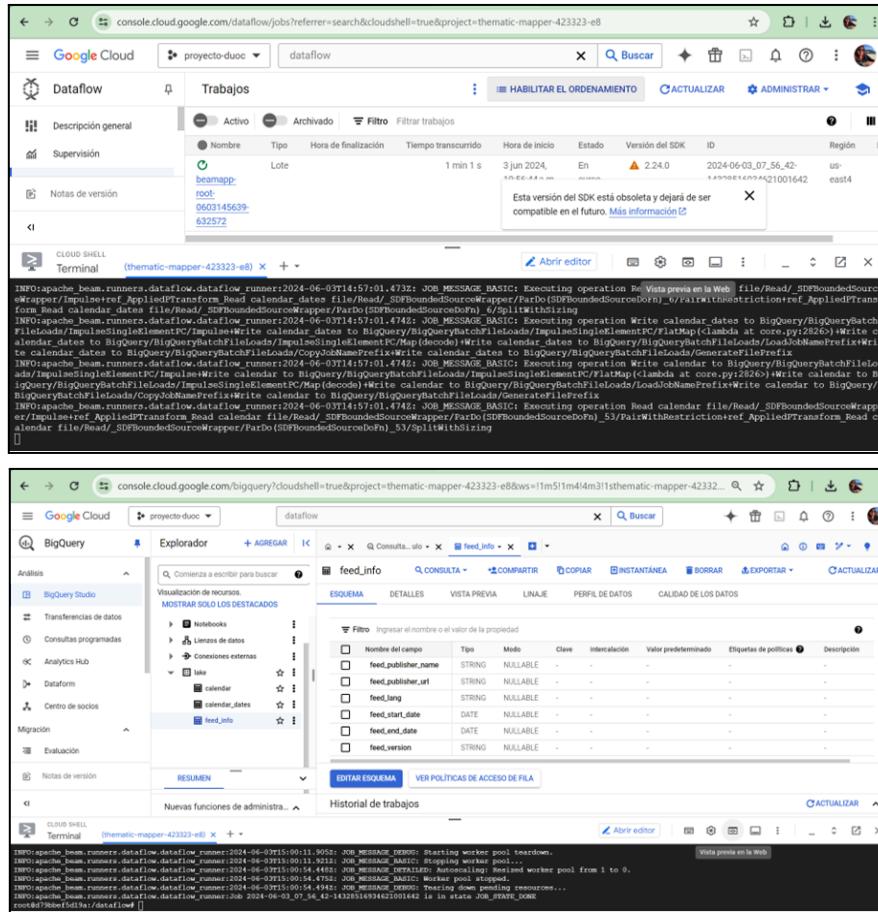
data_convertir_date:

```

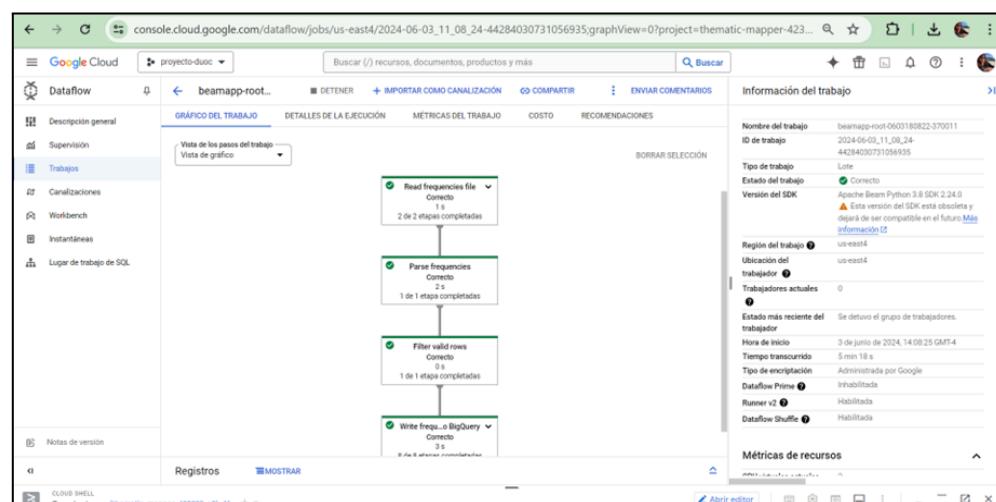
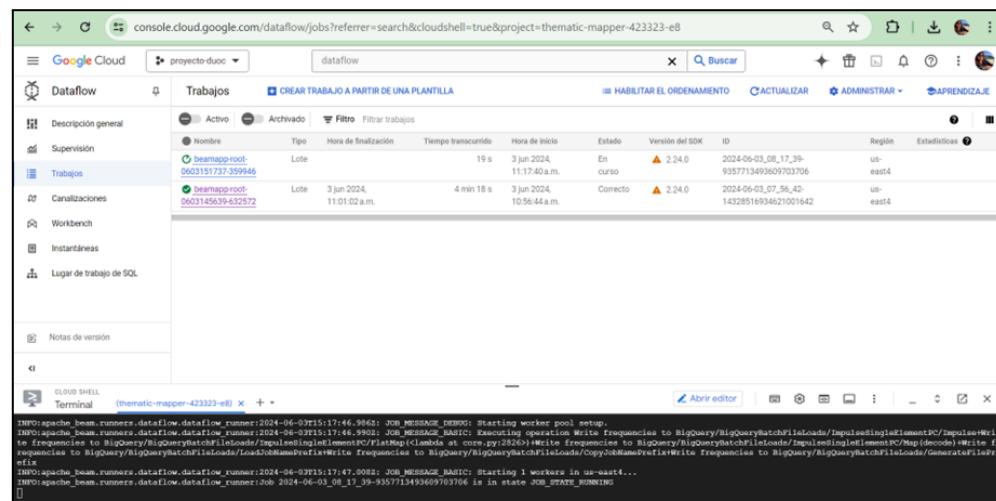
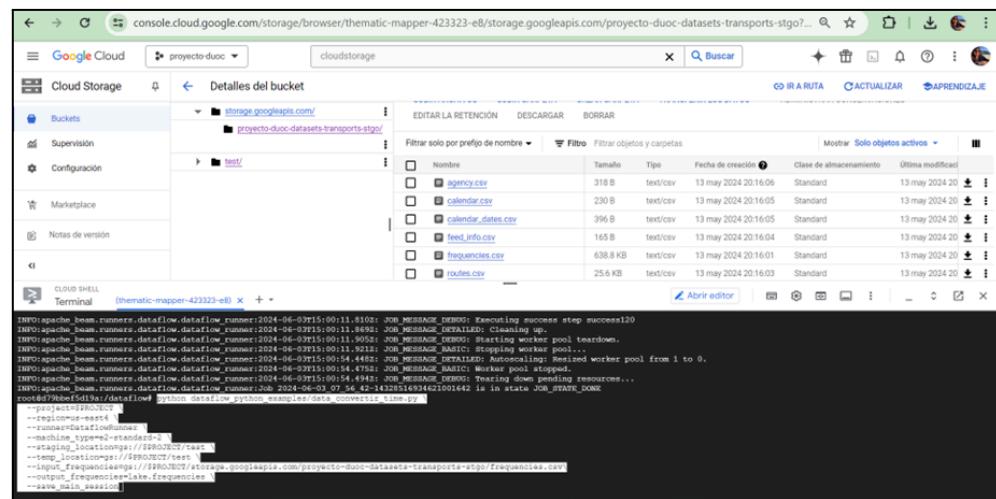
root@#79bebf5d19a:/# cd dataflow/
root@#79bebf5d19a:/dataflow# python dataflow/python/examples/data_convertir_date.py
--project=thematic-mapper
--runner=DataflowRunner
--temp_location=gs://PROJECT/test
--staging_location=gs://PROJECT/test
--region=us-east4
--runner=DataflowRunner
--temp_location=gs://PROJECT/test
--staging_location=gs://PROJECT/test
--input_calendar_dates=gs://PROJECT/storage.googleapis.com/proyecto-duoc-datasets-transports-stgo/calendar_dates.csv
--input_calendar=gs://PROJECT/storage.googleapis.com/proyecto-duoc-datasets-transports-stgo/calendar.csv
--input_feed_info=gs://PROJECT/storage.googleapis.com/proyecto-duoc-datasets-transports-stgo/feed_info.csv
--output_calendar_dates_lake.calendar_dates
--output_calendar_lake.calendar
--output_feed_info_lake.feed_info
--save_main_session

```

Nadia Arellano G. / Ana Karina Muñoz
Ingenieras en Informática



data_converter_time



The screenshot shows the Google Cloud BigQuery interface. On the left, the sidebar includes sections like 'Análisis', 'Transfertas de datos', 'Consultas programadas', 'Analytics Hub', 'Dataform', 'Centro de socios', 'Migración', 'Evaluación', 'Traducción de SQL', 'Administración', 'Supervisión', and 'Administración de capac...'. The main area displays the schema of the 'frequencies' table under the 'lake' dataset. The schema includes columns: trip_id (STRING, NULLABLE), start_time (TIME, NULLABLE), end_time (TIME, NULLABLE), headway_secs (INTEGER, NULLABLE), and exact_times (INTEGER, NULLABLE). Buttons for 'EDITAR ESQUEMA' and 'VER POLÍTICAS DE ACCESO DE FILA' are visible.

data_eliminar_head

The screenshot shows the Google Cloud Dataflow interface. The left sidebar lists 'Descripción general', 'Supervisión', 'Trabajos' (selected), 'Canalizaciones', 'Workbench', 'Instantáneas', 'Lugar de trabajo de SQL', and 'Notas de versión'. The main area shows the execution details of a job named 'beamapp-root-0603160709-515179'. It includes tabs for 'GRÁFICO DEL TRABAJO', 'DETALLES DE LA EJECUCIÓN', 'MÉTRICAS DEL TRABAJO', and 'COSTO'. A table titled 'Vista de los pasos del trabajo' lists various steps: 'Convert to list', 'Extract columns and data', 'Flatten data', 'Format for BigQuery', 'Validate data', 'Filter valid data', and 'Write to BigQuery'. The right side panel provides 'Información del trabajo' with details such as 'Nombre del trabajo', 'ID de trabajo', 'Tipo de trabajo', 'Estado del trabajo', 'Versión del SDK', 'Región del trabajo', 'Ubicación del trabajador', 'Trabajadores actuales', 'Estado más reciente del trabajador', and 'Hora de inicio'.

The screenshot shows the Google Cloud BigQuery interface. The left sidebar is identical to the previous screenshot. The main area displays the data of the 'agency' table under the 'lake' dataset. The table has columns: agency_id (INT64, NULLABLE) and agency_name (STRING, NULLABLE). The data shows four rows: 1 RM (Red Metropolitana de Movilidad), 2 M (Metro de Santiago), 3 MT (EFE Trenes de Chile), and 4 BAA (Bus de Acercamiento Aeropuerto). Buttons for 'CONSULTA', 'COMPARTIR', 'COPIAR', 'INSTANTÁNEA', 'BORRAR', and 'EXPORTAR' are visible.

data_ingestion_sin_modif

```

INFO:apache_beam.runners.beam.runners.dataflow.runner:Job 2024-06-03_09_07_11-16316760694667189360 is in state JOB_STATE_DONE
root@bd79bebef5d19:/dataflow# python dataflow python examples/data-ingestion-sin-modif.py
--project=PROYECTO
--region=us-east4
--temp=DataflowTempDir
--machine_type=e2-standard-2
--staging_location=gs://PROYECTO/test/
--temp_location=gs://PROYECTO/test/
--input_shapes=gs://PROJECT/proyecto-duoc-datasets-transports-stgo/shapes.csv
--input_stop_times=gs://PROJECT/proyecto-duoc-datasets-transports-stgo/stop_times.csv
--input_stops=gs://PROJECT/proyecto-duoc-datasets-transports-stgo/stops.csv
--output_shapes=lake.shapes
--output_stops=lake.stops
--output_stop_times=lake.stop_times
--output_trips=lake.trips
--save_main_session

```



Nombre del campo	Tipo	Modo	Clave	Intercalación	Valor predeterminado	Etiquetas de políticas	Descripción
shape_id	STRING	NULLABLE	-	-	-	-	-
shape_pt_lat	FLOAT	NULLABLE	-	-	-	-	-
shape_pt_lon	FLOAT	NULLABLE	-	-	-	-	-
shape_pt_sequence	INTEGER	NULLABLE	-	-	-	-	-

data_setear_nulos

The screenshot shows the Google Cloud Dataflow job execution graph for the job `beamapp-root-0603204503-637224`. The graph consists of six stages arranged vertically:

- Read routes file**: Status: Correcto, Duration: 0 s, 2 de 2 etapas completadas.
- Parse stops**: Status: Correcto, Duration: 1 s, 2 de 2 etapas completadas.
- Parse nodes**: Status: Correcto, Duration: 0 s, 1 de 1 etapa completadas.
- Write stops..._o_BigQuery**: Status: Correcto, Duration: 2 s, 8 de 8 etapas completadas.
- Write route..._o_BigQuery**: Status: Correcto, Duration: 0 s, 8 de 8 etapas completadas.

The left sidebar shows navigation links for Google Cloud Dataflow, Descripción general, Supervisión, Trabajos, Canalizaciones, Workbench, Instantáneas, and Lugar de trabajo de SQL. The top right includes search, filter, and navigation buttons.

The screenshot shows the Google Cloud BigQuery interface. The left sidebar has sections for 'BigQuery Studio', 'Transferencias de datos', 'Consultas programadas', 'Analytics Hub', 'Dataform', 'Centro de socios', 'Migración', and 'Evaluación'. The main area is titled 'Explorador' and shows a table named 'stops'. The table has columns: stop_id, stop_code, stop_name, stop_lat, stop_lon, and stop_timezone. The data consists of 13 rows, each representing a stop location with its ID, code, name, latitude, longitude, and time zone.

	ESQUEMA	DETALLES	VISTA PREVIA	LINAJE	PERFIL DE DATOS	CALIDAD
Fila	stop_id	stop_code	stop_name	stop_lat	stop_lon	stop_timezone
1	PB241	Sin Información	PB241-Parada / Mall Plaza Nor...	-33.364060	-70.667000	-33.364060
2	PB184	Sin Información	PB184-Los Libertadores / Esq...	-33.366631	-70.667000	-33.366631
3	PB185	Sin Información	PB185-Parada / Pasarela Albany...	-33.366077	-70.667000	-33.366077
4	PB242	Sin Información	PB242-Parada 6 / (M) Los Libe...	-33.366339	-70.667000	-33.366339
5	PB186	Sin Información	PB186-Av. Independencia / Esq...	-33.369142	-70.667000	-33.369142
6	PB187	Sin Información	PB187-Av. Independencia / Esq...	-33.371563	-70.667000	-33.371563
7	PB188	Sin Información	PB188-Parada 1 / (M) Cardenal...	-33.373384	-70.667000	-33.373384
8	PB251	Sin Información	PB251-Av. Independencia / Esq...	-33.376754	-70.667000	-33.376754
9	PB190	Sin Información	PB190-Av. Independencia / Esq...	-33.378221	-70.667000	-33.378221
10	PB191	Sin Información	PB191-Av. Independencia / Esq...	-33.381087	-70.667000	-33.381087
11	PB1707	Sin Información	PB1707-Av. Independencia / Es...	-33.382910	-70.667000	-33.382910
12	PB193	Sin Información	PB193-Parada 1 / (M) Vivaceta...	-33.385698	-70.667000	-33.385698
13	PB194	Sin Información	PB194-Av. Independencia / Esq...	-33.387807	-70.667000	-33.387807

Paso 3: Construir reportes en BigQuery y realizar las visualizaciones con Looker Studio

Para finalizar, nos aseguramos de que los datos procesados se carguen correctamente en BigQuery.

Fila	route_id	agency_id	route_short_name	route_long_name	route_desc
5	L4A	M	L4A	Línea 4A (La Cisterna - Vicuña ...	Sin Información
6	L6	M	L6	Línea 6 (Cerrillos - Los Leones)	Sin Información
7	L5	M	L5	Línea 5 (Plaza de Maipú - Vicen...	Sin Información
8	L5V	M	L5	Línea 5 Ruta Verde (Expreso Plaza de Maipú - Vicente Valdés)	Sin Información
9	L5R	M	L5	Línea 5 Ruta Roja (Expreso Pla...	Sin Información
10	L4	M	L4	Línea 4 (Tobalaba - Plaza de Pu...	Sin Información
11	L4V	M	L4	Línea 4 Ruta Verde (Expreso Tobalaba - Plaza de Puente Alto)	Sin Información
12	L4R	M	L4	Línea 4 Ruta Roja (Expreso Tob...	Sin Información
13	L2	M	L2	Línea 2 (Vespucio Norte - Hosp...	Sin Información

➤ Consultas SQL en BigQuery:

Consulta 1: Listar las agencias

Esta consulta recupera todas las agencias de transporte disponibles, mostrando sus ID, nombres, URL's y zonas horarias.

```

1 SELECT FROM `thematic-mapper-423323-e8.lake.agency` LIMIT 1000
2
3 SELECT agency_id, agency_name, agency_url, agency_timezone
4 FROM `thematic-mapper-423323-e8.lake.agency`;
    
```

agency_id	agency_name	agency_url	agency_timezone
RM	Red Metropolitana de Movilidad	http://www.red.cl	America/Santiago
M	Metro de Santiago	http://www.metro.cl	America/Santiago
MT	EFE Trenes de Chile	http://www.efe.cl	America/Santiago
BAA	Bus de Acercamiento Aeropuerto	http://www.nuevopudahuel.cl	America/Santiago

Consulta 2: Obtener la lista de paradas con accesibilidad para silla de ruedas

Recupera todas las paradas que son accesibles para personas en sillas de ruedas (wheelchair_boarding = 1), mostrando el ID, nombre, latitud, longitud y URL de la parada.

The screenshot shows the Google Cloud BigQuery interface. The sidebar on the left lists various datasets and tables, including 'lake' and its sub-tables like 'agency', 'calendar', 'calendar_dates', 'feed_info', 'frequencies', 'routes', 'shapes', 'stop_times', 'stops', and 'trips'. The main area shows a query titled 'Consulta sin título' (Untitled Query) with the following SQL code:

```

1 SELECT * FROM `thematic-mapper-423323-e8#.lake.stops` LIMIT 1000
2
3 SELECT stop_id, stop_name, stop_lat, stop_lon, stop_url
4 FROM `thematic-mapper-423323-e8#.lake.stops`
5 WHERE wheelchair_boarding = 1;
6

```

The results table displays 11 rows of data, each representing a stop with its ID, name, coordinates, URL, and wheelchair accessibility status. The results are as follows:

stop_id	stop_name	stop_lat	stop_lon	stop_url
PT0201	Estación Central (Ander1)	-33.45115	-70.6788	Sin información
PT0202	Estación Lo Valledor (Ander1)	-33.478	-70.6806	Sin información
PT0203	Estación Pedro Aguirre Cerda (Ander1)	-33.4931	-70.6818	Sin información
PT0204	Estación Lo Espírito (Ander1)	-33.5137	-70.6857	Sin información
PT0205	Estación Lo Blanco (Ander1)	-33.5278	-70.69029	Sin información
PT0206	Estación Freire (Ander1)	-33.58714	-70.6996	Sin información
PT0207	Estación San Bernardo (Ander1)	-33.59429	-70.69795	Sin información
PT0208	Estación Maestranza (Ander1)	-33.60648	-70.69702	Sin información
PT0209	Estación Cinco Pinos (Ander1)	-33.6232	-70.70068	Sin información
PT0210	Estación Nos (Ander1)	-33.63285	-70.70489	Sin información
PT0211	Estación Central (Ander2)	-33.45117	-70.6789	Sin información

Consulta 3: Contar el número de viajes por cada ruta

Cuenta el número de viajes (`trip_id`) por cada ruta, mostrando el ID de la ruta, su nombre corto y largo. Esto puede ser útil para analizar la frecuencia de los viajes en cada ruta.

The screenshot shows the Google Cloud BigQuery interface. The sidebar on the left lists various datasets and tables, including 'thematic-mapper-423323-e8#' and its sub-tables like 'routes', 'route_short_name', 'route_long_name', and 'number_of_trips'. The main area shows a query titled 'Consulta sin título' (Untitled Query) with the following SQL code:

```

1 SELECT * FROM `thematic-mapper-423323-e8#.lake.routes` LIMIT 1000
2
3 SELECT *
4   FROM `thematic-mapper-423323-e8#.lake.routes`
5   JOIN `thematic-mapper-423323-e8#.lake.trips` trips
6     ON routes.route_id = trips.route_id
7   GROUP BY routes.route_id, routes.route_short_name, routes.route_long_name;
8

```

The results table displays 8 rows of data, each representing a route with its ID, short name, long name, and the number of trips. The results are as follows:

route_id	route_short_name	route_long_name	number_of_trips
MTN	MTN	Tren Nos - Estación Central	42
MTR	MTR	Tren Rancagua - Estación Centr...	120
L1	L1	Línea 1 (San Pablo - Los Domin...	18
L3	L3	Línea 3 (Plaza Colicura - Frena...	20
L4A	L4A	Línea 4A (La Cisterna - Vicuña ...	18
L6	L6	Línea 6 (Cerrillos - Los Leones)	18
L5	L5	Línea 5 (Plaza de Maipú - Vicen...	10
L5V	L5	Línea 5 Ruta Verde (Expreso Plaza de Maipú - Vicente	8

➤ Visualización en Looker Studio:

1. La primera opción para visualizar los datos es comenzando desde un trabajo desde cero y se debe conectar con BigQuery el proyecto de Looker Studio.

The screenshot shows the 'Add data to the report' dialog in Looker Studio. It lists several data source options under 'Connect to data': 'De Google' (Google Sheets, Google Analytics, Google Ads), 'Hoja de cálculo de Google' (Google Sheets), 'BigQuery' (Google BigQuery), 'AppSheet' (Google AppSheet), 'Subida de archivos' (Upload files), 'Amazon Redshift', and 'Apigee'. The 'BigQuery' option is highlighted.

Luego se añaden los Datos del proyecto al Informe

The screenshot shows the 'Add data to the report' dialog with 'BigQuery' selected. The 'Credenciales de datos' (Data credentials) dropdown shows 'NADIA AREACELLY ARELLANO GONZALEZ'. Below it, the 'PROJECTS RECENTS' section lists 'MIS PROYECTOS' (introduzca el ID del proyecto manualmente) and 'PROYECTOS COMPARTIDOS' (proyecto-duoc). The 'DATOS' section shows a list of tables: agency, calendar, calendar_dates, feed_info, frequencies, routes, stops, stop_times, stops, and trips. At the bottom right are 'Cancelar' and 'Añadir' buttons.

Y se crean las tablas que queremos mostrar.

Tabla Listado de Agencias.

The screenshot shows a Looker Studio visualization titled 'Listado de Agencias'. The table has columns: agency_id, agency_name, agency_url, and agency_timezone. The data includes rows for: 1. ITC Interactiva de Chile, 2. ITC Televisión de Chile, 3. Metro de Santiago, and 4. RAA. The visualization interface on the right shows the 'CONFIGURACIÓN' (Configuration) panel with 'Nombre de datos' (Data name) set to 'BigQuery' and 'Añadir DATOS' (Add data) button. The 'ESTILO' (Style) panel shows 'Propiedad de la tabla' (Table property) and 'Barra de filtros' (Filter bar).

Tabla Paradas con accesibilidad para silla de ruedas

stop_id	stop_name	stop_lat	stop_lon	wheelchair_boarding
1	Metro de Santiago	-33.426151	-70.700707	False
2	BAA	-33.426151	-70.700707	False
3	Deportes	-33.426151	-70.642441	False
4	Cerro	-33.423151	-70.699506	False
5	Pintana	-33.423151	-70.699506	False
6	Dpto. Pinto Aguirre	-33.426151	-70.644706	False
7	Alto	-33.426151	-70.644706	False

- La segunda opción es realizar consultas SQL en las tablas creadas en BigQuery, con las que se pueden generar reportes en Looker Studio sin necesidad de crear un proyecto desde cero.

Paso 1: Se crea la consulta SQL

```

SELECT
  routes.route_id,
  routes.route_short_name,
  routes.route_long_name,
  COUNT(trips.trip_id) AS number_of_trips
FROM `thematic-mapper-423323-e8d` lake.routes
JOIN `thematic-mapper-423323-e8d` lake.trips
ON routes.route_id = trips.route_id
GROUP BY routes.route_id, routes.route_short_name, routes.route_long_name;
  
```

route_id	route_short_name	route_long_name	number_of_trips
211	211	La Florida - Nos	58
MTN	MTN	Tren Nos - Estación Central	42
I10	I10	Villa Los Heros - Usach	58
S18	S18	J.J. Pérez - Bilbao	74
B30N	B30N	Renca - Centro	40
I14N	I14N	Mall Plaza Oeste - Centro	16

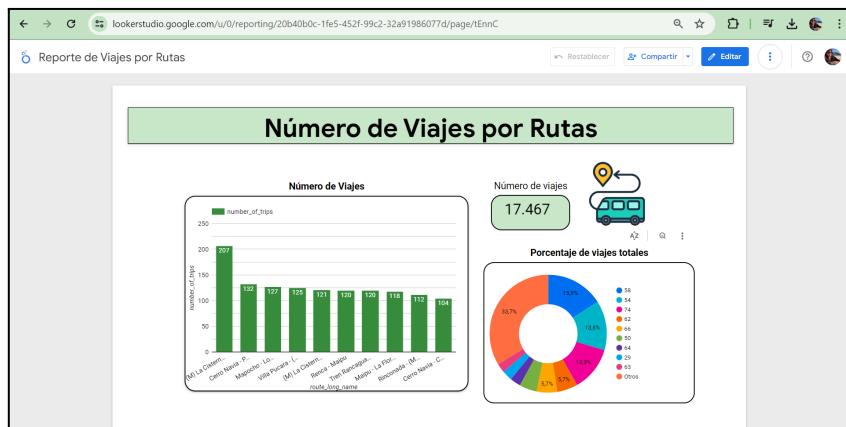
Paso 2: Se exporta el resultado a Looker Studio:

route_id	number_of_trips
1	120
2	74
3	74
4	74
5	74
6	74
7	40
8	74
9	74
10	74
11	74

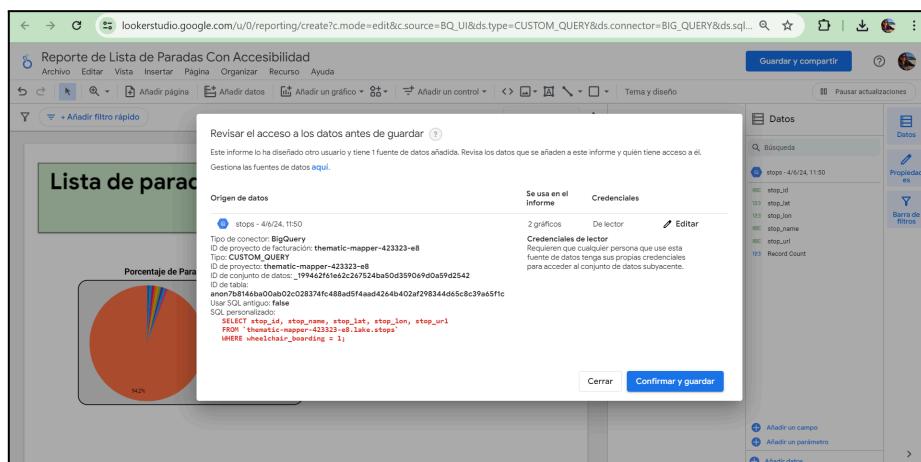
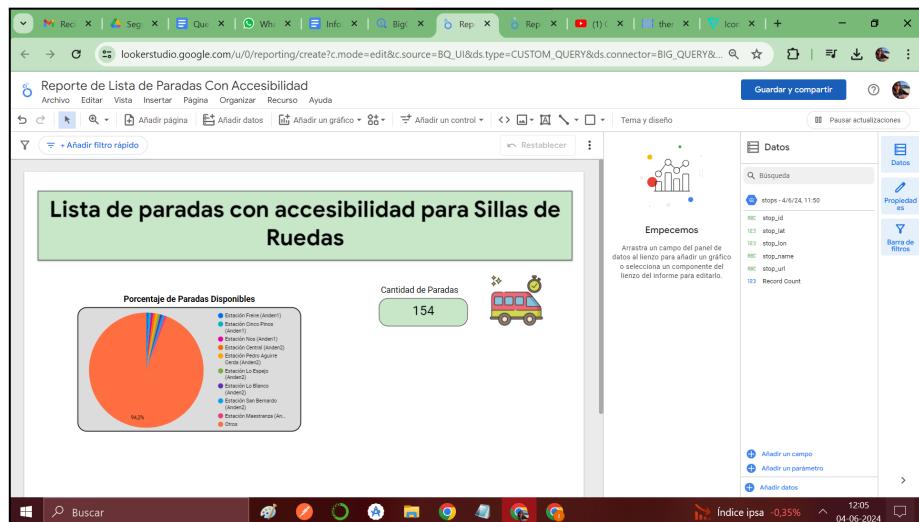
Por ejemplo, creamos y visualizamos el reporte de Número de Viajes por ruta.

```

SQL personalizado:
SELECT
    routes.route_id,
    routes.route_short_name,
    routes.route_long_name,
    routes.number_of_trips
    FROM `thematic-mapper-423323-e8` AS routes
    JOIN `thematic-mapper-423323-e8` AS trips
    ON routes.route_id = trips.route_id
    ORDER BY routes.route_short_name, routes.route_long_name;
  
```

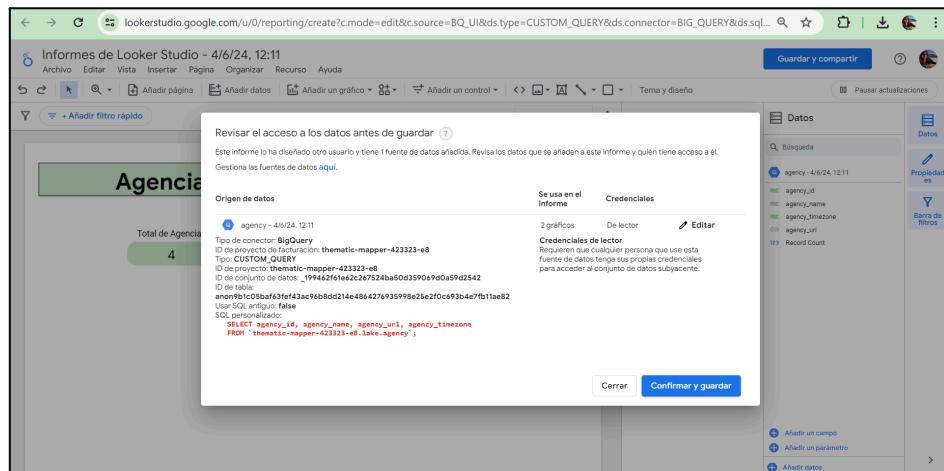
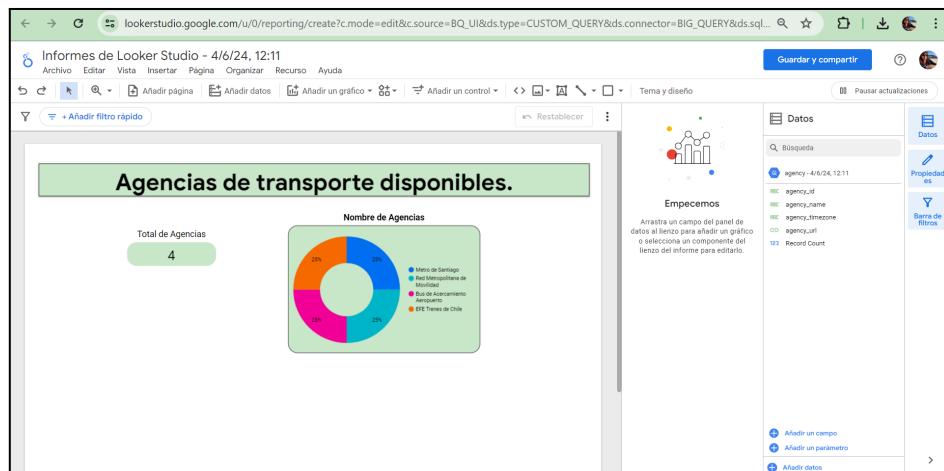


También, creamos y visualizamos el reporte de Lista de paradas con accesibilidad para sillas de ruedas.





Finalmente, creamos y visualizamos el reporte de todas las agencias de transporte disponibles.

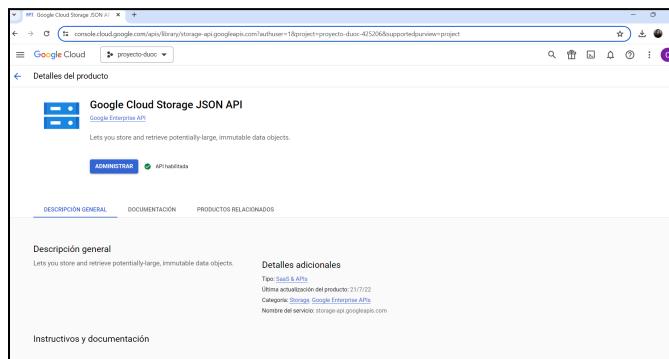




→ *Proceso Pub/Sub*

Paso 1: Conectar con Cloud Storage

- Configuramos el entorno de trabajo: Nos aseguramos de tener habilitada la API de Cloud Storage. Se siguen los mismos pasos que con la API de DataFlow en donde se muestra en detalle.

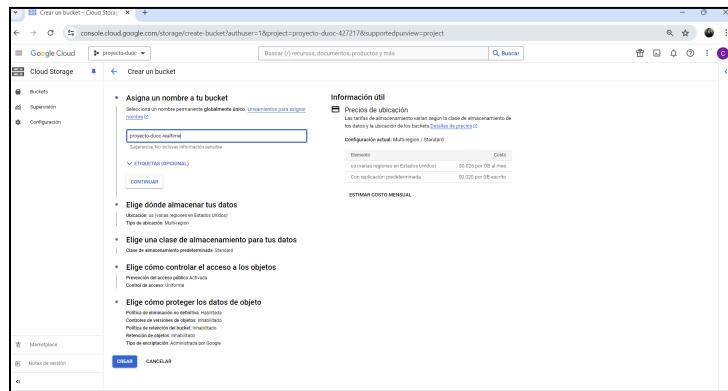


- Creación y configuración del Bucket en Cloud Storage

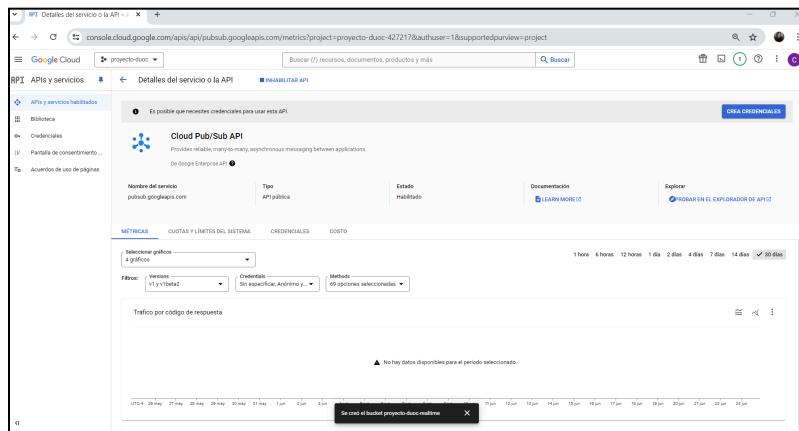
-> Buckets -> Crear

Se debe configurar lo siguiente:

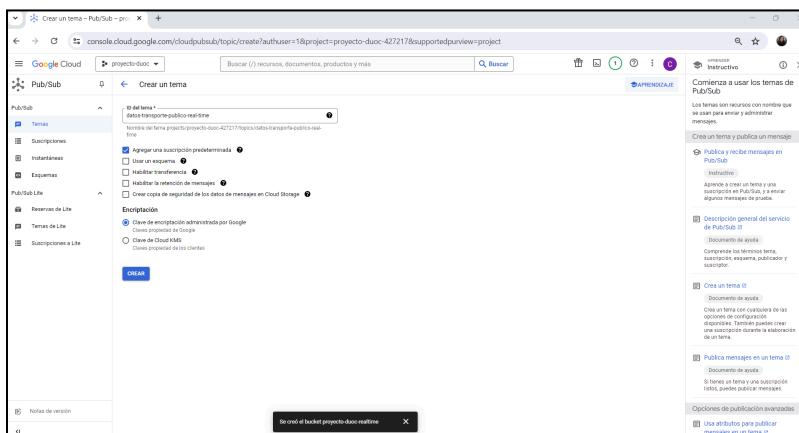
- 1.- Asigna nombre
- 2.- Elegir el lugar donde almacenar tus datos
- 3.- Elegir una clase de almacenamiento para los datos
- 4.- Elegir cómo proteger los datos del objeto.



➤ Verificamos si la API de Servicio PUB/SUB está habilitada



➤ En la consola de Pub/Sub creamos un nuevo tema.



The screenshot shows the Google Cloud Pub/Sub Topics page. On the left, there's a sidebar with 'Pub/Sub' selected under 'Temas'. The main area has a table titled 'LISTA' with columns 'ID del tema', 'Clave de encriptación', and 'Nombre del tema'. A single row is visible: 'datos-transporte-publico-real-time' with 'Google-managed' in the encryption key column and 'projects/thematic-mapper-423323-e8/topics/datos-transporte-publico-real-time' in the name column. To the right, there's a sidebar titled 'Selecciona un tema' with tabs for 'PERMISOS', 'ETIQUETAS', and 'POLÍTICA DE A.'. The 'PERMISOS' tab is active, showing a message: 'Selecione por lo menos un recurso.'

- A continuación, creamos una suscripción al tema y otorgamos los permisos necesarios para que Pub/Sub lea y escriba el Bucket.

This screenshot shows the 'Crear suscripción' (Create subscription) page. In the 'ID de la suscripción' field, 'suscripcion-datos-transporte-publico-real-time' is entered. In the 'Selección un tema de Cloud Pub/Sub' dropdown, 'projects/thematic-mapper-423323-e8/topics/datos-transporte-publico-real-time' is selected. Under 'Tipo de envío', the 'Escribir en Cloud Storage' option is selected. Below this, a sub-modal window titled 'Escribir en el bucket de Cloud Storage' is open, showing two sections: 'Asigna el rol de Lector' and 'Asignar rol de creador'. Both sections have checkboxes checked. The 'Bucket' dropdown contains 'thematic-mapper-423323-e8-real-time'. At the bottom, a success message says 'Se otorgó el rol de creador correctamente.'

This screenshot shows the same 'Crear suscripción' page as the previous one, but the 'ESTABLECER PERMISO' (Set permission) button at the bottom is now highlighted. The sub-modal window from the previous screenshot is still visible, showing the 'Escribir en el bucket de Cloud Storage' settings. The message 'Se otorgó el rol de creador correctamente.' is also present here.

Nadia Arellano G. / Ana Karina Muñoz

Ingenieras en Informática

Crear suscripción

Escritor en Cloud Storage
Una variante de la operación de envío. Selecciona esta opción si deseas que Pub/Sub anexe los mensajes directamente a un bucket de Cloud Storage existente. [Más información](#)

Bucket *
 EXPLORAR
El bucket de Cloud Storage al que la suscripción escribe los archivos de salida.

El agente de servicio de Pub/Sub tiene los permisos necesarios para escribir en el destino.

Formato de archivo *
 El formato de archivo para los archivos de salida escritos en el bucket de Cloud Storage.

Prefijo del nombre de archivo

Cuando se especifica, todos los archivos de salida contienen el prefijo del nombre del archivo.

Sufijo del nombre de archivo

Agrega un sufijo a los nombres de archivo que contiene la suscripción.

Procesamiento de archivos por lotes

Duración máx. del lote de almacenamiento (minutos)

Configurar la cantidad máxima de bytes de almacenamiento
La cantidad máxima de bytes permitidos por archivo. Debe ser un número entre 1 KB y 10 GB. Es posible que Pub/Sub cree un nuevo archivo de salida antes de que alcance la cantidad máxima de bytes.

Tamaño del lote
 Unidades

Tiempo de retención de mensajes ●
De 10 minutos a 7 días de duración
 Días Horas Minutos

Retener mensajes confirmados
Al habilitar esta opción, los mensajes confirmados se conservan durante el período de retención de mensajes especificado anteriormente. Esto hace que las tarifas de

Período de vencimiento ●
 Vence después de todos estos días de inactividad (hasta 365)
Una suscripción está inactiva si no hay actividad del suscriptor como conexiones abiertas, extracciones activas o inserciones exitosas.
 Días

Nunca vence
La suscripción no tiene vencimiento, independientemente de la actividad.

Plazo de confirmación ●
 Segundos
El plazo de confirmación es de 10 a 600 segundos. Cuando usas bibliotecas cliente de alto nivel, se aplica la configuración de administración de asignación de tiempo en lugar de este plazo de confirmación. [Más información](#)

Filtro de suscripción
Si se proporciona una sintaxis de filtro, los suscriptores solo recibirán mensajes que coincidan con el filtro.
[Más información](#)

Mensajes no entregados
 Ordenar mensajes con una clave de ordenamiento
Cuando está habilitado, los mensajes etiquetados con la misma clave de ordenamiento se reciben en el orden en que se publican. Esta opción no se puede cambiar más adelante. La habilitación del pedido del mensaje puede aumentar la latencia de publicación-suscripción y disminuir la disponibilidad de la publicación.

Mensajes no entregados
 Habilitar mensajes no entregados
Las suscripciones podrán configurar una cantidad máxima de intentos de entrega. Cuando no se puede entregar un mensaje, se vuelve a publicar en el tema de mensajes no entregados especificado.

Política de reintentos
La política de reintentos se activará en los eventos excedidos del plazo de confirmación o el NACK de un mensaje determinado. [Más información](#)

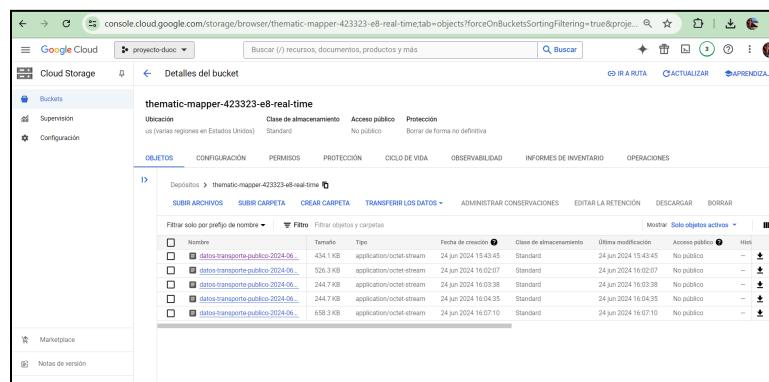
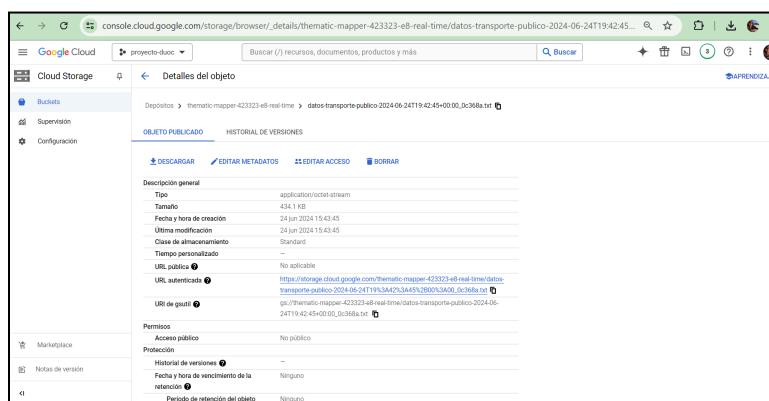
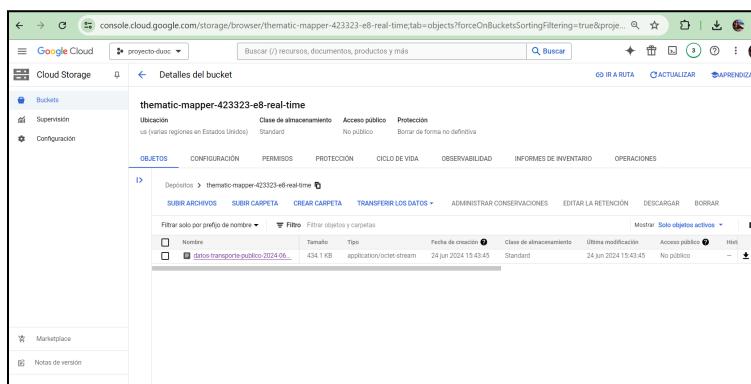
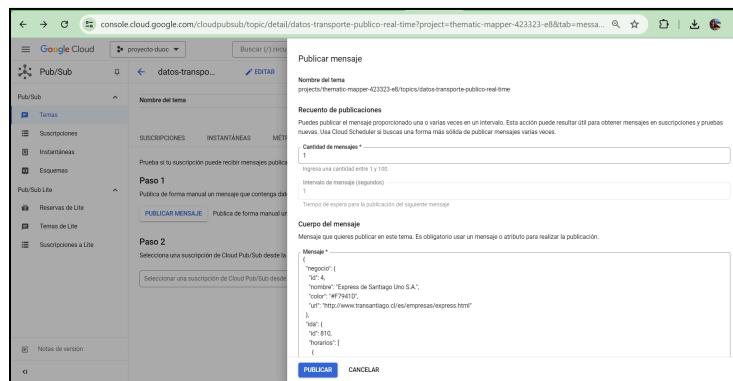
Reintentar inmediatamente
 Reintentar después de un retraso de retirada exponencial

Paso 2: Descargar y/o generar los archivos al dataLake (Cloud Storage)

- Enviamos los archivos al bucket de Cloud Storage por medio de mensajes Pub/Sub publicándolos de forma manual y también a través de código python.

Forma Manual: El mensaje se obtiene desde la url ["https://www.red.cl/restservice_v2/rest/conocerecorrido?codsint=101"](https://www.red.cl/restservice_v2/rest/conocerecorrido?codsint=101), donde "101" es un código de servicio devuelto por la siguiente URL: https://www.red.cl/restservice_v2/rest/getservicios/all el mensaje enviado corresponde a un extracto del formato JSON.

Nadia Arellano G. / Ana Karina Muñoz
Ingenieras en Informática



Con código python: Efectúa la publicación de los mensajes desde la fuente.

Primero creamos un entorno virtual para crear todas las dependencias de python, ejecutando los siguientes comando en cloud shell:

`python -m venv env // crea entorno virtual`

`source env/bin/activate // activa entorno virtual`

The screenshot shows the Google Cloud Storage interface. On the left, a sidebar lists 'Google Cloud' and 'Cloud Storage'. The main area displays a bucket named 'thematic-mapper-423323-e8-real-time'. The bucket details show it's located in 'us (varias regiones en Estados Unidos)', has 'Standard' storage class, is 'No público' (public), and has 'Borrar de forma no definitiva' (Delete) protection. Below this, there are tabs for 'OBJETOS' (Objects), 'CONFIGURACIÓN' (Configuration), 'PERMISOS' (Permissions), 'PROTECCIÓN' (Protection), 'CICLO DE VIDA' (Lifecycle), 'OBSERVABILIDAD' (Observability), 'INFORMES DE INVENTARIO' (Inventory Reports), and 'OPERACIONES' (Operations). A 'Notas de versión' (Version Notes) section is also present. At the bottom, there's a terminal window titled 'thematic-mapper-423323-e8\$' showing initial setup commands.

A continuación, creamos un archivo con el código Python que se asegura de manejar adecuadamente los reintentos en la obtención de datos de 5 servicios de transporte: 205, 210, 226, 101, 508, publicar esos datos en un servicio de mensajería (Pub/Sub), y almacenarlos de manera persistente en Cloud Storage, eliminando los archivos temporales locales una vez completada la tarea.

El archivo python “mensaje-test.py” se guarda en la carpeta del entorno virtual llamada “env”.



```
#!/usr/bin/env python3
# [START main]
# [START imports]
from google.cloud import pubsub_v1
from google.cloud import storage
import json
import time
import logging
# [END imports]

# [START configuration]
# Configuration de logging
logging.basicConfig(level=logging.INFO, format='%(asctime)s - %(levelname)s - %(message)s')
# [END configuration]

# [START publisher]
topic_id = 'the-thematic-mapper-42332-e8-real-time'
topic_path = f'projects/the-thematic-mapper-42332/topics/{topic_id}'
bucket_name = 'the-thematic-mapper-42332-e8-real-time'
# [END publisher]

def fetch_data_with_retries(url, retries=5, delay=10, timeout=60):
    """Fetch data from a URL with exponential backoff and a timeout.

    Args:
        url (str): The URL to fetch data from.
        retries (int): The number of retries to attempt.
        delay (int): The delay between consecutive retries in seconds.
        timeout (int): The total timeout for the request in seconds.

    Returns:
        bytes: The fetched data as bytes.
    """
    for attempt in range(retries):
        response = requests.get(url, timeout=timeout)
        if response.status_code == 200:
            return response.json()
        else:
            raise requests.exceptions.RequestException(f"Request failed with status code {response.status_code} after {attempt + 1} tries")
    raise requests.exceptions.ConnectionError("All attempts failed after {retries} tries")

def publish_message(publisher, topic_path, message_bytes):
    try:
        future = publisher.publish(topic_path, message=message_bytes)
        future.result(timeout=120) # Aumentar el tiempo de espera a 120 segundos
        print(f'Mensaje publicado con éxito a la topic: {topic_path}')
    except Exception as e:
        logging.error(f'Failed to publish message: {e}')

def main(topic_name, file_name, limit):
    # [START subscriber]
    subscriber = pubsub_v1.SubscriberClient()
    subscription_path = subscriber.subscription_path(bucket_name, topic_name)
    # [END subscriber]

    # [START receive_messages]
    def callback(message):
        print(f'Received message: {message.data.decode("utf-8")}')
        message.ack()

    stream = subscriber.subscribe(subscription_path, callback=callback)
    # [END receive_messages]

    # [START process_data]
    while True:
        data = fetch_data_with_retries('https://www.googleapis.com/storage/v1/b/the-thematic-mapper-42332-e8-real-time/o?projection=noACL&recursive=true&maxResults=100', retries=5, delay=10, timeout=60)
        for item in data['items']:
            if item['name'].endswith('.json'):
                blob = storage.Blob(item['name'], storage_client)
                blob.download_to_file('/tmp/' + item['name'])
                with open('/tmp/' + item['name']) as f:
                    data = json.load(f)
                    if len(data) < limit:
                        publish_message(publisher, topic_path, message_bytes=json.dumps(data).encode('utf-8'))
                    else:
                        print(f'Skiping file {item["name"]}, size: {len(data)}')
        time.sleep(60)
    # [END process_data]
    stream.close()
    # [START acknowledge_message]
    subscriber.acknowledge(subscription_path, message_ids=[message.message_id])
    # [END acknowledge_message]
    # [START shutdown]
    subscriber.close()
    # [END shutdown]
# [END main]
```

Antes de ejecutar el código, se instala apache beam dentro del entorno con el siguiente comando:

```
pip install 'apache-beam[gcp]'
```

Comienza tu prueba gratuita con un crédito de \$300. No te preocupes, no se te cobrará si no utilizas los créditos. [Más información](#)

DESCATAR [COMENZAR GRATIS](#)

Google Cloud [proyecto-duck](#) Buscar (/) recursos, documentos, productos y más [Buscar](#)

Cloud Storage [Detalles del bucket](#) [IR A RUTA](#) [ACTUALIZAR](#) [APRENDIZAJE](#)

Buckets [Supervisión](#) [Marketplace](#) [Notas de versión](#)

thematic-mapper-423323-e8-real-time

Ubicación	Clase de almacenamiento	Acceso público	Protección
us (varias regiones en Estados Unidos)	Standard	No público	Borrar de forma no definitiva

[OBJETOS](#) [CONFIGURACIÓN](#) [PERMISOS](#) [PROTECCIÓN](#) [CICLO DE VIDA](#) [OBSERVABILIDAD](#) [INFORMES DE INVENTARIO](#) [OPERACIONES](#)

Navegador de carpetas [Depósitos > thematic-mapper-423323-e8-real-time](#)

[thematic-mapper-423323-e8-real-time](#) [SUBIR ARCHIVOS](#) [SUBIR CARPETA](#) [CREAR CARPETA](#) [TRANSFIRIR LOS DATOS](#) [Abrir editor](#) [ADMINISTRAR CONSERVACIONES](#)

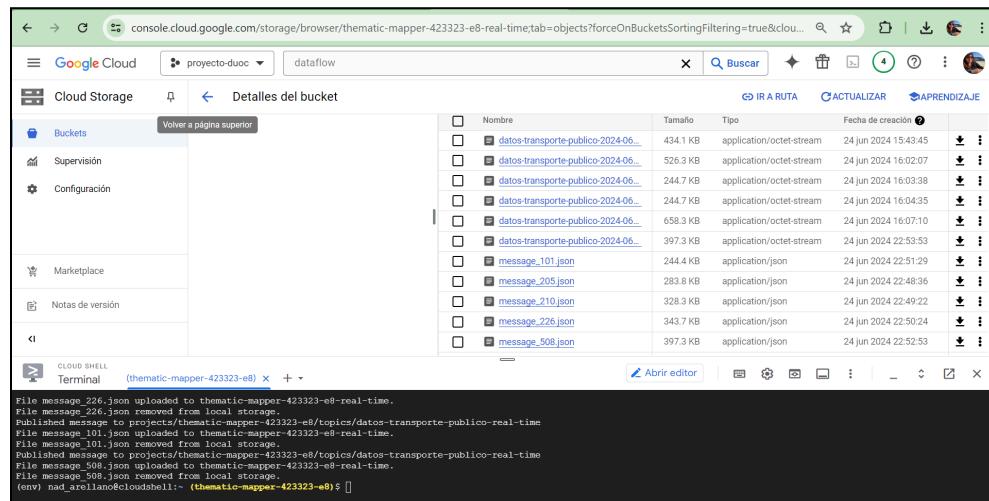
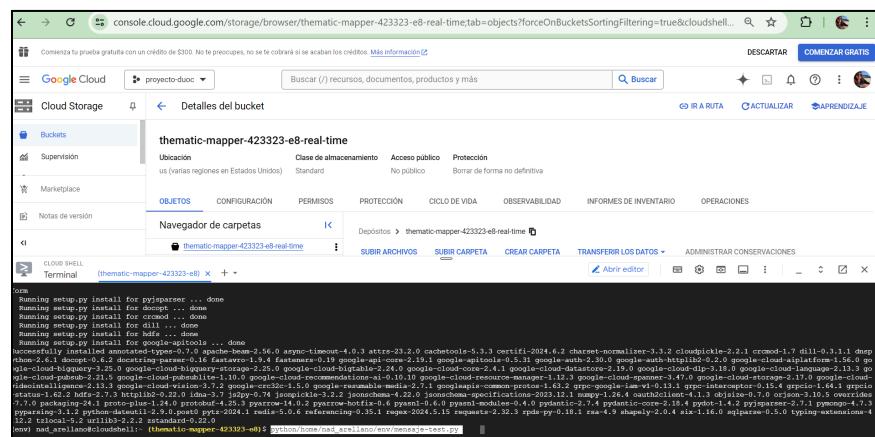
CLOUD SHELL Terminal (thematic-mapper-423323-e8) +

```
Welcome to Cloud Shell! Type "help" to get started.  
Your Cloud Platform project in this session is set to thematic-mapper-423323-e8.  
Use "gcloud config set project [PROJECT_ID]" to change to a different project.  
Collecting apache-beam dependencies...  
Collecting apache-beam[gcp]  
  Using cached apache-beam-2.56.0-py310-cp310-manylinux2_17_x86_64.manylinux2014_x86_64.whl (14.5 kB)  
Collecting pytz==2018.3  
  Using cached pytz-2018.3-py2.py3-none-any.whl (505 kB)  
Collecting cronet<0.1.7  
  Using cached cronet-0.1.7.tar.gz (89 kB)  
  Preparing metadata (setup.py)... done  
Collecting dill<0.3.2,>=0.3.1.1  
  Using cached dill-0.3.1.1-py3-none-any.whl (151 kB)
```

The screenshot shows a Google Cloud Storage browser interface. The URL is console.cloud.google.com/storage/browser/thematic-mapper-423323-e8-real-time;tab=objects?forceOnBucketsSortingFiltering=true&cloudshell.... The main header includes the Google Cloud logo, a search bar, and navigation icons. Below the header, there's a banner about a free trial and a 'DETENER' button. The left sidebar has sections for 'Buckets', 'Supervisión', 'Marketplace', and 'Notas de versión'. The 'Cloud Storage' section is selected. The main content area shows a bucket named 'thematic-mapper-423323-e8-real-time'. It displays details like 'Ubicación: us (varias regiones en Estados Unidos)', 'Clase de almacenamiento: Standard', 'Acceso público: No público', and 'Protección: Borrar de forma no definitiva'. Below this, there are tabs for 'OBJETOS', 'CONFIGURACIÓN', 'PERMISOS', 'PROTECCIÓN', 'CICLO DE VIDA', 'OBSERVABILIDAD', 'INFORMES DE INVENTARIO', and 'OPERACIONES'. A 'Navegador de carpetas' sidebar shows a single item: 'Depósitos > thematic-mapper-423323-e8-real-time'. At the bottom, there are buttons for 'SUBIR ARCHIVOS', 'SUBIR CARPETA', 'CREAR CARPETA', 'TRANSFERIR LOS DATOS', and 'ADMINISTRAR CONSERVACIONES'. A terminal window at the bottom left shows the command 'Running setup.py install for bffs ... done'.

Posteriormente se ejecutará el código python dentro del entorno virtual con el siguiente comando:

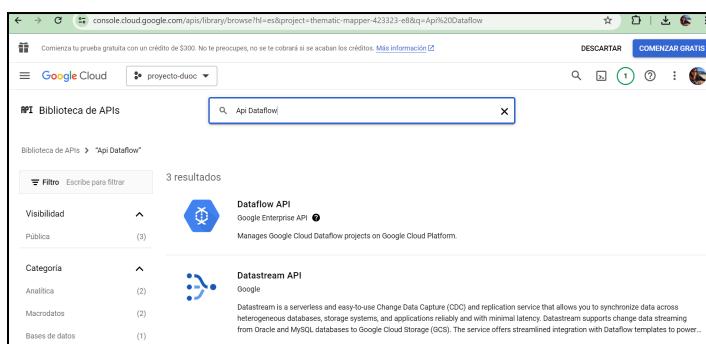
```
python /home/nad_arellano/env/mensaje-test.py
```



Paso 3: Construir procesos de limpieza, transformación y carga con DataFlow

El siguiente paso es configurar un pipeline en DataFlow para luego realizar los procesos de limpieza, transformaciones necesarias y carga al modelo de datos final.

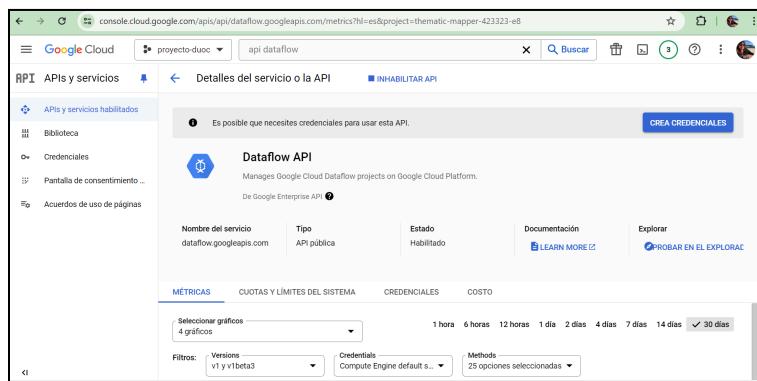
- Configurar el entorno de trabajo: Nos aseguramos de tener habilitada la API de DataFlow.



Nadia Arellano G. / Ana Karina Muñoz
Ingenieras en Informática

The image consists of four vertically stacked screenshots of the Google Cloud Platform API library interface, specifically for the Dataflow API.

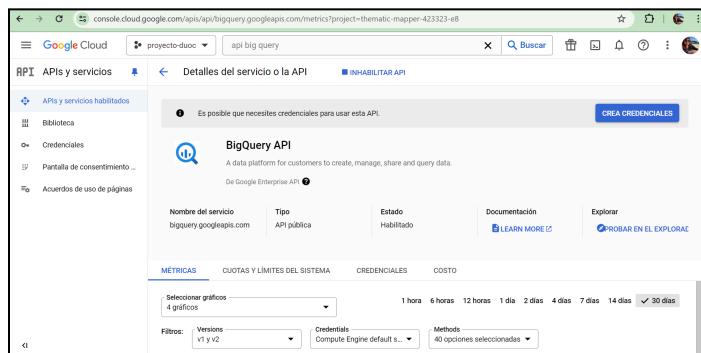
- Screenshot 1:** Shows the "Detalles del producto" (Product details) page for the Dataflow API. It includes a summary, a "PROBAR ESTA API" (Try this API) button, and tabs for "DESCRIPCIÓN GENERAL" (Description), "DOCUMENTACIÓN" (Documentation), and "PRODUCTOS RELACIONADOS" (Related products).
- Screenshot 2:** Shows the "APIs y servicios" (APIs and services) dashboard for the project. It displays traffic and error metrics for the Dataflow API, both of which show no data available for the selected date range.
- Screenshot 3:** Shows the "Detalles del servicio o la API" (Service or API details) page for the Dataflow API. It provides service information like name, type, and status, along with metrics, quotas, and documentation links.
- Screenshot 4:** Shows the "Detalles del producto" (Product details) page again, this time with the "HABILITAR" (Enable) button highlighted, indicating the process of enabling the API.



➤ Creamos el conjunto de datos “lake_real_time” en BigQuery

La importancia de crearlo en esta parte del proceso, es porque debemos tener listo el destino final de los datos que serán transformados, ya que luego de que DataFlow ejecute el código Python con las transformaciones, se moverán los datos preparados al lake ubicado en BigQuery.

➤ Configurar el entorno de trabajo: Nos aseguramos de tener habilitada la API de BigQuery. Se siguen los mismos pasos que con la API de DataFlow en donde se muestra en detalle.



➤ Creamos el conjunto de datos “lake_real_time” en BigQuery. En *Cloud Shell*, creamos un conjunto de datos en BigQuery llamado “lake_real_time” en donde cargaremos todas las tablas.

Script: **bq mk lake_real_time**

```

2024-06-24 23:16:29,002 - INFO - File message python_210.txt removed from local storage.
2024-06-24 23:17:10,080 - INFO - Published message to projects/thematic-mapper-423323-e8/topics/datos-transporte-publico-real-time
2024-06-24 23:17:10,381 - INFO - File message python_211.txt uploaded to thematic-mapper-423323-e8-real-time.
2024-06-24 23:17:10,392 - INFO - File message python_212.txt uploaded to thematic-mapper-423323-e8-real-time.
2024-06-24 23:19:10,512 - ERROR - Received error: HTTPConnectionPool(host='www.red.cl', port=443): Max retries exceeded with url: /restservice_v2/rest/conocerelcorrido?codsint=101 (Caused by ConnectTimeoutError(curl#13, connection.HTTPSConnection object at 0x7ef779798d20>; connect timeout=120)
2024-06-24 23:19:10,513 - INFO - Retrying in 10 seconds... (attempt 1/5)
2024-06-24 23:20:52,093 - INFO - Published message to projects/thematic-mapper-423323-e8/topics/datos-transporte-publico-real-time
2024-06-24 23:21:05,393 - INFO - File message python_213.txt uploaded to thematic-mapper-423323-e8-real-time.
2024-06-24 23:21:05,391 - INFO - File message python_214.txt uploaded to thematic-mapper-423323-e8-real-time.
2024-06-24 23:21:35,953 - INFO - Published message to projects/thematic-mapper-423323-e8/topics/datos-transporte-publico-real-time
2024-06-24 23:21:56,257 - INFO - File message python_226.txt uploaded to thematic-mapper-423323-e8-real-time.
2024-06-24 23:21:56,257 - INFO - File message python_226.txt removed from local storage.
2024-06-24 23:22:42,367 - INFO - Published message to projects/thematic-mapper-423323-e8/topics/datos-transporte-publico-real-time
2024-06-24 23:22:42,362 - INFO - File message python_205.txt uploaded to thematic-mapper-423323-e8-real-time.
2024-06-24 23:22:42,363 - INFO - File message python_205.txt removed from local storage.
(env) nad_arellano@cloudshell:~ (thematic-mapper-423323-e8)$ bq mk lake_real_time

```

ID de conjunto de datos	thematic-mapper-423323-e8.lake_real_time
Creado	24 jun 2024, 7:41:39 p.m. UTC-4
Vencimiento predeterminado de la tabla	Nunca
Última modificación	24 jun 2024, 7:41:39 p.m. UTC-4
Ubicación de los datos	US
Descripción	
Intervalación predeterminada	

```

2024-06-24 23:20:52,391 - INFO - File message python_201.txt removed from local storage.
2024-06-24 23:21:55,956 - INFO - Published message to projects/thematic-mapper-423323-e8/topics/datos-transporte-publico-real-time
2024-06-24 23:21:56,257 - INFO - File message python_226.txt uploaded to thematic-mapper-423323-e8-real-time.
2024-06-24 23:22:42,362 - INFO - File message python_205.txt removed from local storage.
2024-06-24 23:22:42,363 - INFO - File message python_205.txt uploaded to thematic-mapper-423323-e8-real-time.
2024-06-24 23:22:42,363 - INFO - File message python_205.txt removed from local storage.
(env) nad_arellano@cloudshell:~ (thematic-mapper-423323-e8)$ bq mk lake_real_time
dataset 'thematic-mapper-423323-e8:lake_real_time' successfully created.
(env) nad_arellano@cloudshell:~ (thematic-mapper-423323-e8)$ 

```

➤ Creamos un archivo JSON (schema_tabla.json) que define el esquema de las tablas en BigQuery para definir sus columnas y el tipos de datos.

```

1 [ {"name": "id", "type": "INTEGER", "mode": "NULLABLE"}, 
2  {"name": "tipo0ls", "type": "STRING", "mode": "NULLABLE"}, 
3  {"name": "inicio", "type": "STRING", "mode": "NULLABLE"}, 
4  {"name": "fin", "type": "STRING", "mode": "NULLABLE"}, 
5  {"name": "paradero_id", "type": "INTEGER", "mode": "NULLABLE"}, 
6  {"name": "paradero_cod", "type": "STRING", "mode": "NULLABLE"}, 
7  {"name": "paradero_num", "type": "INTEGER", "mode": "NULLABLE"}, 
8  {"name": "paradero_comuna", "type": "STRING", "mode": "NULLABLE"}, 
9  {"name": "paradero_latitud", "type": "FLOAT", "mode": "NULLABLE"}, 
10 {"name": "paradero_longitud", "type": "FLOAT", "mode": "NULLABLE"}, 
11 {"name": "servicio_id", "type": "INTEGER", "mode": "NULLABLE"}, 
12 {"name": "servicio_cod", "type": "STRING", "mode": "NULLABLE"}, 
13 {"name": "servicio_nombre", "type": "STRING", "mode": "NULLABLE"}, 
14 {"name": "servicio_color", "type": "STRING", "mode": "NULLABLE"}, 
15 {"name": "empresa_nombre", "type": "STRING", "mode": "NULLABLE"}, 
16 {"name": "empresa_color", "type": "STRING", "mode": "NULLABLE"}, 
17 {"name": "recorrido_destino", "type": "STRING", "mode": "NULLABLE"}, 
18 {"name": "itinerario", "type": "BOOLEAN", "mode": "NULLABLE"}, 
19 {"name": "codigo_servicio", "type": "STRING", "mode": "NULLABLE"}, 
20 {"name": "timestamp", "type": "TIMESTAMP", "mode": "NULLABLE"} ]

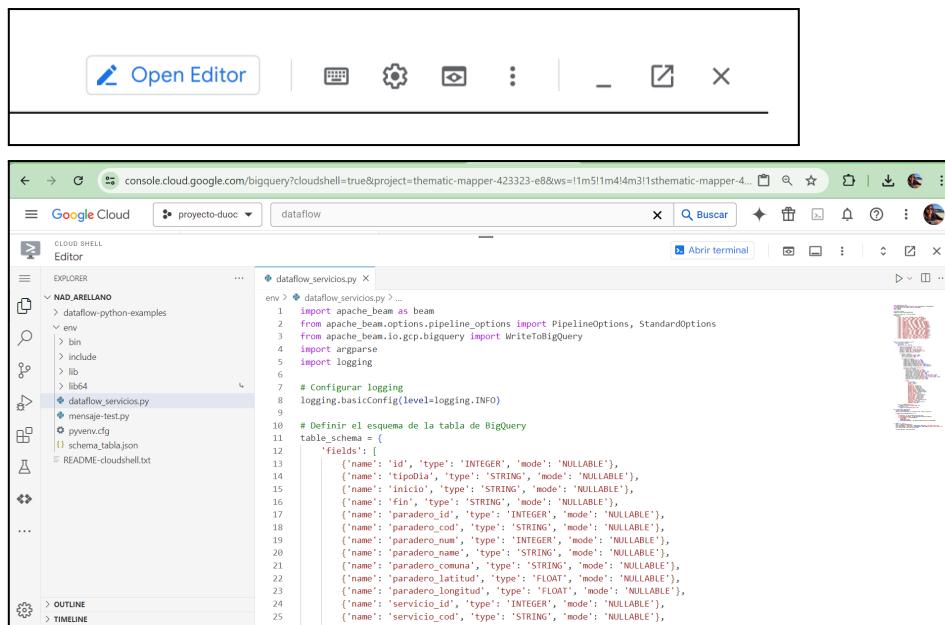
```

- Creamos 5 tablas en BigQuery (se reemplaza el número del servicio por cada script) usando la línea de comandos `bq`, especificando el archivo anterior de esquema para definir las columnas de las tablas y sus tipos de datos.
Scripts:

```
bq mk --table --description "Tabla con datos de transporte servicio 205"  
thematic-mapper-423323-e8:lake_real_time.tabla_205  
/home/nad_arellano/env/schema_tabla.json
```

The screenshot shows the Google Cloud BigQuery interface. On the left sidebar, there are several projects listed under 'Análisis': 'BigQuery Studio', 'Transferencias de datos', 'Consultas programadas', 'Analytics Hub', 'Dataform', and 'Notas de versión'. The main area displays the schema for the 'lake_real_time' table, which has 11 columns: id, tipodia, inicio, fin, paradero_id, paradero_cod, paradero_num, paradero_name, and paradero_comunas. The 'Tipos de datos' column indicates that all columns are of type STRING except for 'id' which is INTEGER. The 'Modo' column shows that all columns are nullable. The 'Clave' column is empty. The 'Intercalación' column shows '-' for all columns. The 'Valor predeterminado' column is empty. The 'Etiquetas d' column is empty.

- Desarrollar el script de DataFlow (Python): En la carpeta “env” ubicada en el Editor de Cloud Shell creamos una archivo llamado “dataflow_servicios” que enviará los datos a las tablas creadas anteriormente en Bigquery.



- Ejecución del pipeline: Ejecutamos los 5 scripts de Python de cada servicio en DataFlow.

```
python /home/nad_arellano/env/dataflow_servicios.py \
--runner DataflowRunner \
--project thematic-mapper-423323-e8 \
--region us-central1 \
--staging_location gs://thematic-mapper-423323-e8-real-time/temp \
--temp_location gs://thematic-mapper-423323-e8-real-time/temp \
--job_name tabla_101-prueba1 \
--input_path gs://thematic-mapper-423323-e8-real-time/message_101.json \
--output_table thematic-mapper-423323-e8:lake_real_time.tabla_101
```

```
a 205 '/home/nad_arellano/env/schema tabla.json
Table 'thematic-mapper-423323-e8:lake_real_time.tabla_205' successfully created.
(env) nad_arellano@cloudshell:~ (thematic-mapper-423323-e8)$ python /home/nad_arellano/env/dataflow_servicios.py \
--runner DataflowRunner \
--project thematic-mapper-423323-e8 \
--region us-central1 \
--staging_location gs://thematic-mapper-423323-e8-real-time/temp \
--temp_location gs://thematic-mapper-423323-e8-real-time/temp \
--job_name tabla_101-prueba1 \
--input_path gs://thematic-mapper-423323-e8-real-time/message_101.json \
--output_table thematic-mapper-423323-e8:lake_real_time.tabla_101'
```

Por ejemplo, visualizamos la carga de datos del servicio 101

```
INFO:apache_beam.runners.dataflow.dataflow_runner:2024-06-25T00:00:00+00:00 : JOB_MESSAGE_BASIC: All workers have finished the startup processes and began to receive work requests.
INFO:apache_beam.runners.dataflow.dataflow_runner:2024-06-25T04:06:12,778Z : JOB_MESSAGE_BASIC: Stopping worker pool...
INFO:apache_beam.runners.dataflow.dataflow_runner:2024-06-25T04:06:12,820Z : JOB_MESSAGE_BASIC: Worker pool stopped.
INFO:apache_beam.runners.dataflow.dataflow_runner:2024-06-25T04:06:59,346Z : JOB_MESSAGE_BASIC: Worker pool stopped.
INFO:apache_beam.runners.dataflow.dataflow_runner:Job 2024-06-24_21_00_38-6507466117226790762 is in state JOB_STATE_DONE
(env) nad_arellano@cloudshell:~ (thematic-mapper-423323-e8)$
```

Fila	id	tipoDia	inicio	fin	paradero_id
1	810	Sábado	05:30	23:50	101
2	810	Lunes a Viernes	05:30	23:50	101
3	810	Lunes a Viernes	05:30	23:50	136
4	810	Lunes a Viernes	05:30	23:50	101
5	810	Domingo y Festivos	05:30	23:50	18
6	810	Domingo y Festivos	05:30	23:50	105
7	810	Sábado	05:30	23:50	18
8	810	Domingo y Festivos	05:30	23:50	18
9	810	Sábado	05:30	23:50	105
10	810	Lunes a Viernes	05:30	23:50	18
11	810	Lunes a Viernes	05:30	23:50	106

Paso 4: Construir reportes en BigQuery y realizar las visualizaciones con Looker Studio

Para finalizar, nos aseguramos de que los datos procesados se carguen correctamente en las tablas de BigQuery.

Fila	id	tipoDia	inicio	fin	paradero_id
1	53	Sábado	05:30	00:40	2
2	53	Sábado	05:30	00:40	2
3	53	Sábado	05:30	00:40	3
4	53	Sábado	05:30	00:40	6
5	53	Sábado	05:30	00:40	6
6	53	Sábado	05:30	00:40	94
7	53	Sábado	05:30	00:40	64
8	53	Sábado	05:30	00:40	2
9	53	Sábado	05:30	00:40	64
10	53	Domingo y Festivos	05:30	00:40	94
11	53	Lunes a Viernes	05:30	00:40	6
12	53	Sábado	05:30	00:40	94
13	53	Lunes a Viernes	05:30	00:40	6

Fila	id	tipoDia	inicio	fin	paradero_id
1	61	Sábado	00:00	23:59	1
2	61	Lunes a Viernes	00:00	23:59	64
3	61	Sábado	00:00	23:59	92
4	61	Domingo y Festivos	00:00	23:59	41
5	61	Sábado	00:00	23:59	64
6	61	Lunes a Viernes	00:00	23:59	101
7	61	Sábado	00:00	23:59	134
8	61	Sábado	00:00	23:59	101
9	61	Domingo y Festivos	00:00	23:59	64
10	61	Lunes a Viernes	00:00	23:59	132
11	61	Domingo y Festivos	00:00	23:59	92
12	61	Sábado	00:00	23:59	57
13	61	Sábado	00:00	23:59	64
14	61	Domingo y Festivos	00:00	23:59	57
15	61	Sábado	00:00	23:59	57

	ESQUEMA	DETALLES	VISTA PREVIA	LINAJE	PERFIL DE DATOS	CALIDAD DE LOS DATOS
Fila	// id	// tipoDia	// inicio	// fin	// paradero_id	
1	77	Lunes a Viernes	05:30	00:30	6:	
2	77	Lunes a Viernes	05:30	00:30	95:	
3	77	Lunes a Viernes	05:30	00:30	53:	
4	77	Domingo y Festivos	05:30	00:30	6:	
5	77	Domingo y Festivos	05:30	00:30	52:	
6	77	Lunes a Viernes	05:30	00:30	6:	
7	77	Sábado	05:30	00:30	95:	
8	77	Lunes a Viernes	05:30	00:30	6:	
9	77	Lunes a Viernes	05:30	00:30	95:	
10	77	Sábado	05:30	00:30	95:	
11	77	Lunes a Viernes	05:30	00:30	5:	
12	77	Sábado	05:30	00:30	6:	
13	77	Domingo y Festivos	05:30	00:30	95:	
14	77	Domingo y Festivos	05:30	00:30	54:	
15	77	Lunes a Viernes	nn:nn	nn:nn	65:	

	ESQUEMA	DETALLES	VISTA PREVIA	LINAJE	PERFIL DE DATOS	CALIDAD DE LOS DATOS
Fila	// id	// tipoDia	// inicio	// fin	// paradero_id	
1	173	Sábado	00:00	23:59	118:	
2	173	Sábado	00:00	23:59	27:	
3	173	Lunes a Viernes	00:00	23:59	46:	
4	173	Lunes a Viernes	00:00	23:59	51:	
5	173	Domingo y Festivos	00:00	23:59	50:	
6	173	Sábado	00:00	23:59	3:	
7	173	Lunes a Viernes	00:00	23:59	28:	
8	173	Domingo y Festivos	00:00	23:59	27:	
9	173	Domingo y Festivos	00:00	23:59	55:	
10	173	Sábado	00:00	23:59	47:	
11	173	Domingo y Festivos	00:00	23:59	27:	
12	173	Domingo y Festivos	00:00	23:59	28:	
13	173	Domingo y Festivos	00:00	23:59	41:	
14	173	Sábado	00:00	23:59	130:	
15	173	Lunes a Viernes	nn:nn	23:59	129:	

➤ Consultas SQL en BigQuery:

Consulta 1: Número de servicios por comuna

Esta consulta agrupa los datos por comuna y cuenta el número total de servicios para cada comuna en todas las tablas.

```

SELECT
    paradero_comuna,
    COUNT(servicio_id) AS total_servicios
FROM
    thematic-mapper-423323-e8.lake_real_time.tabla_205
UNION ALL
SELECT
    paradero_comuna,
    COUNT(servicio_id) AS total_servicios
FROM
    thematic-mapper-423323-e8.lake_real_time.tabla_101
UNION ALL
SELECT

```

```

paradero_comuna,
COUNT(servicio_id) AS total_servicios
FROM
thematic-mapper-423323-e8.lake_real_time.tabla_226
UNION ALL
SELECT
paradero_comuna,
COUNT(servicio_id) AS total_servicios
FROM
thematic-mapper-423323-e8.lake_real_time.tabla_210
UNION ALL
SELECT
paradero_comuna,
COUNT(servicio_id) AS total_servicios
FROM
thematic-mapper-423323-e8.lake_real_time.tabla_508
GROUP BY
paradero_comuna
ORDER BY
total_servicios DESC;

```

paradero_comuna	total_servicios
CERRILLOS	510
SANTIAGO	507
PUENTE ALTO	495
SANTIAGO	468
ESTACIÓN CENTRAL	420

 The results are sorted by 'total_servicios' in descending order."/>

Consulta 2: Paraderos con mayor número de servicios

Esta consulta encuentra los paraderos con el mayor número de servicios en todas las tablas, mostrando el paradero_name y paradero_comuna.

```

SELECT
paradero_name,
paradero_comuna,
COUNT(servicio_id) AS numero_servicios
FROM
thematic-mapper-423323-e8.lake_real_time.tabla_205
UNION ALL
SELECT
paradero_name,
paradero_comuna,
COUNT(servicio_id) AS numero_servicios
FROM
thematic-mapper-423323-e8.lake_real_time.tabla_210
UNION ALL
SELECT
paradero_name,
paradero_comuna,
COUNT(servicio_id) AS numero_servicios
FROM
`thematic-mapper-423323-e8.lake_real_time.tabla_226`

```

```

UNION ALL
SELECT
    paradero_name,
    paradero_comuna,
    COUNT(servicio_id) AS numero_servicios
FROM
    thematic-mapper-423323-e8.lake_real_time.tabla_101
UNION ALL
SELECT
    paradero_name,
    paradero_comuna,
    COUNT(servicio_id) AS numero_servicios
FROM
    thematic-mapper-423323-e8.lake_real_time.tabla_508
GROUP BY
    paradero_name,
    paradero_comuna
ORDER BY
    numero_servicios DESC
LIMIT 50;

```

The screenshot shows the Google Cloud BigQuery interface. On the left, there's a sidebar with navigation links like 'BigQuery Studio', 'Transferencias de datos', 'Consultas programadas', 'Analytics Hub', 'Dataform', 'Centro de socios', 'Organización', 'Migración', 'Evaluación', and 'Notas de versión'. The main area has a title bar 'Consulta sin título' with tabs for 'Ejecutar', 'Programación', 'Más', 'Guardar', 'Descargar', 'Compartir', and a note 'Se completó la consulta'. Below this is a code editor with the query provided above. To the right is a results table titled 'Resultados de la consulta' with columns 'INFORMACIÓN DEL TRABAJO', 'RESULTADOS', 'GRÁFICO', 'JSON', 'DETALLES DE LA EJECUCIÓN', and 'GRÁFICO DE EJECUCIÓN'. The 'RESULTADOS' tab is selected, showing a table with five rows of data:

	paradero_name	paradero_comuna	numero_servicios
1	Avenida 5 de Abril esq. Vista H...	ESTACIÓN CENTRAL	114
2	Avenida 5 de Abril esq. Curicaví	ESTACIÓN CENTRAL	84
3	Avenida 5 de Abril esq. El Altar...	ESTACIÓN CENTRAL	66
4	Alameda esq. Nataniel Cox	SANTIAGO	60
5	Av. Cerróncha v Tron esq. Av. Fvra...	PINTO ALTO	57

Consulta 3: Servicios por empresa.

Consulta que agrupa los servicios por empresa y destino

```

SELECT
    empresa_nombre
    COUNT(*) AS total_servicios
FROM
    thematic-mapper-423323-e8.lake_real_time.tabla_101
UNION ALL
SELECT
    empresa_nombre
    COUNT(*) AS total_servicios
FROM
    thematic-mapper-423323-e8.lake_real_time.tabla_210
UNION ALL
SELECT
    empresa_nombre
    COUNT(*) AS total_servicios
FROM
    thematic-mapper-423323-e8.lake_real_time.tabla_226
UNION ALL
SELECT
    empresa_nombre

```

```
COUNT(*) AS total_servicios
FROM
    thematic-mapper-423323-e8.lake_real_time.tabla_205
UNION ALL
SELECT
    empresa_nombre
    COUNT(*) AS total_servicios
FROM
    thematic-mapper-423323-e8.lake_real_time.tabla_508
GROUP BY
    empresa_nombre
ORDER BY
    total_servicios DESC;
```

The screenshot shows the Google Cloud BigQuery Studio interface. On the left, there's a sidebar with various options like 'BigQuery Studio', 'Transferencias de datos', 'Consultas programadas', etc. The main area has a query editor with the following SQL code:

```
1 SELECT
2     empresa_nombre,
3     servicio_destino,
4     total_servicios
5 FROM (
6     SELECT
7         empresa_nombre,
```

Below the query editor is a table titled 'Resultados de la consulta' (Results of the query) showing the following data:

Fila	empresa_nombre	servicio_destino	total_servicios
1	Express de Santiago Uno S.A.	Cerrillos	654
2	Buses Metropolitana S.A.	Av. Las Torres	342
3	SU-BUS Chile S.A.	Puente Alto	324
4	SU-BUS Chile S.A.	Nonato Coo	288
5	SU-BUS Chile S.A.	Puente Alto	255
6	STU US5	Av. Departamental	234
7	Buses Vule S.A.	Mall Plaza Oeste	198

At the bottom of the results table, it says 'Resultados por página: 50 1 - 50 de 234' and has a 'ACTUALIZAR' button.

➤ Visualización en Looker Studio:

Con las consultas SQL en las tablas creadas en BigQuery, se pueden generar reportes en Looker Studio sin necesidad de crear un proyecto desde cero.

Se exporta el resultado a Looker Studio:

The screenshot shows the Google Cloud BigQuery Studio interface. On the left, there's a sidebar with various analysis tools like BigQuery Studio, Transferencias de datos, Consultas programadas, Analytics Hub, Dataform, Centro de socios, Organización, Migración, Evaluación, and Notas de versión. The main area displays a query titled "Consulta sin título" with the following SQL code:

```

1 SELECT
2     empresa_nombre,
3     servicio_destino,
4     total_servicios
5 FROM (
6     SELECT
7         empresa_nombre,

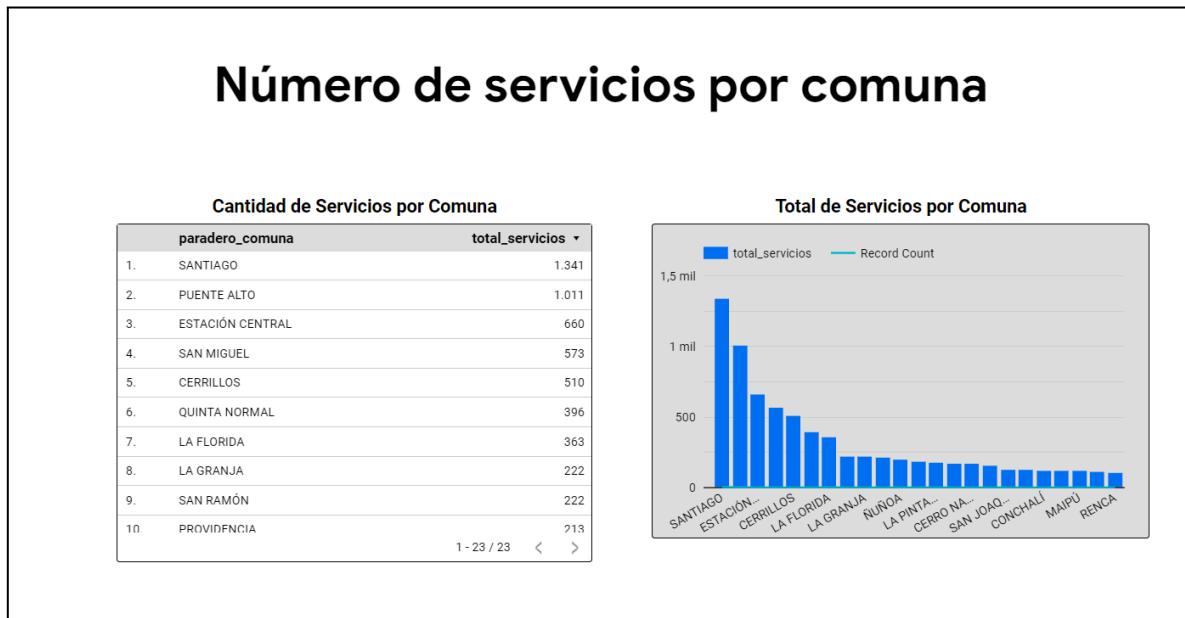
```

Below the code, the results table is shown with columns: INFORMACIÓN DEL TRABAJO, RESULTADOS, GRÁFICO, and JSON. The results show data for 7 rows:

Fila	empresa_nombre	servicio_destino	total_servicios
1	Express de Santiago Uno S.A.	Cerrillos	
2	Buses Metropolitana S.A.	Av. Las Torres	
3	SU-BUS Chile S.A.	Puente Alto	
4	SU-BUS Chile S.A.	Nonato Coo	
5	SU-BUS Chile S.A.	Puente Alto	
6	STU US5	Av. Departamental	
7	Buses Vule S.A.	Mall Plaza Oeste	

On the right, there are several exploration options: Explorar con Hojas de cálculo, Explorar con Looker Studio, Explorar con un notebook de Python, Explorar con lienzo de datos, and Vista previa. Below the table, it says "Resultados por página: 50 1 - 50 de 234".

Por ejemplo, creamos y visualizamos el reporte de Número de servicios por comuna



También, creamos y visualizamos el reporte de Paraderos con mayor número de servicios

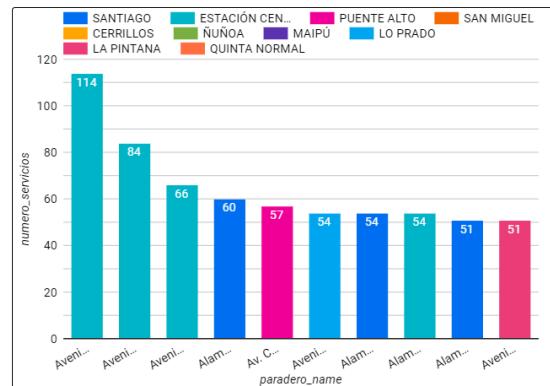
Paraderos con mayor número de servicios

Paraderos con Número de Servicios

paradero_name	numero_servicios
1. Avenida 5 de Abril esq. Vista Hermosa	114
2. Avenida 5 de Abril esq. Curacaví	84
3. Avenida 5 de Abril esq. El Altarcillo	66
4. Alameda esq. Nataniel Cox	60
5. Av. Concha y Toro esq. Av. Eyzaguirre	57
6. Avenida Las Rejas esq. Alameda	54
7. Alameda esq. Avenida Portugal	54
8. Alameda esq. Obispo M. Umaña	54
9. Alameda esq. Carmen	51
10. Avenida Gabriela esq. Avenida Juanita	51
11. Alameda esq. Arturo Prat	51

1 - 50 / 50 < >

Cantidad de servicios por Paradero



Finalmente, creamos y visualizamos el reporte de Servicios por empresa

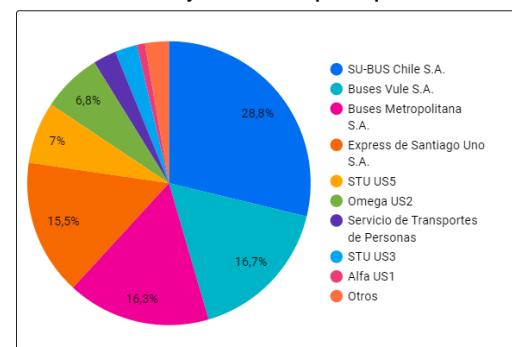
Servicios por empresa

Cantidad de Servicios por Empresa

empresa_nombre	total_servicios
1. SU-BUS Chile S.A.	2.112
2. Buses Vule S.A.	1.227
3. Buses Metropolitana S.A.	1.200
4. Express de Santiago Uno S.A.	1.137
5. STU US5	516
6. Omega US2	501
7. Servicio de Transportes de Personas...	195
8. STU US3	186
9. Redbus Urbano S.A.	66
10. Alfa US1	66

1 - 13 / 13 < >

Porcentaje de Servicios por Empresa



V. Conclusiones

En conclusión, GCP nos entrega la capacidad de integrar y procesar grandes volúmenes de datos ya sean históricos o en tiempo real, utilizando sus herramientas avanzadas como Cloud Storage y sus buckets de almacenamiento. Además, se logró configurar un pipeline de procesamiento de datos con DataFlow, lo que permitió realizar transformaciones como la limpieza y el formateo de datos.

El almacenamiento de los datos procesados en BigQuery facilitó la realización de consultas complejas y la generación de reportes.

Finalmente, las visualizaciones, presentadas en Looker Studio, ofrecen una perspectiva clara y accesible sobre la disponibilidad y variabilidad del transporte público en Santiago facilitando la planificación y optimización de los servicios.

VI. Referencias

- Documentación de Google Cloud Platform (GCP): Se utilizó la documentación oficial de GCP proporcionada por Google para comprender las características, funcionalidades y mejores prácticas de los servicios de GCP utilizados en el proyecto, como Cloud Storage, Dataflow, BigQuery y Composer.
- Datos de transporte del Gobierno de Chile: Datos históricos y en tiempo real del transporte público de Santiago desde la plataforma de datos abiertos del Gobierno de Chile. Estos datos fueron utilizados como fuente principal de información para el análisis de la arquitectura de referencia propuesta.
- Ejemplos de casos relevantes de la página web de Google Cloud: Se revisaron casos de estudio y ejemplos de aplicaciones similares en la página web oficial de Google Cloud Platform para obtener ideas y mejores prácticas para el diseño y la implementación del proyecto de plataforma de datos de transporte público en GCP.