

PIPELINE INTEGRAL DE INGENIERÍA DE DATOS

Juan Archidona Ahijado



UNIVERSIDAD
COMPLUTENSE
MADRID

ntic
master

ÍNDICE

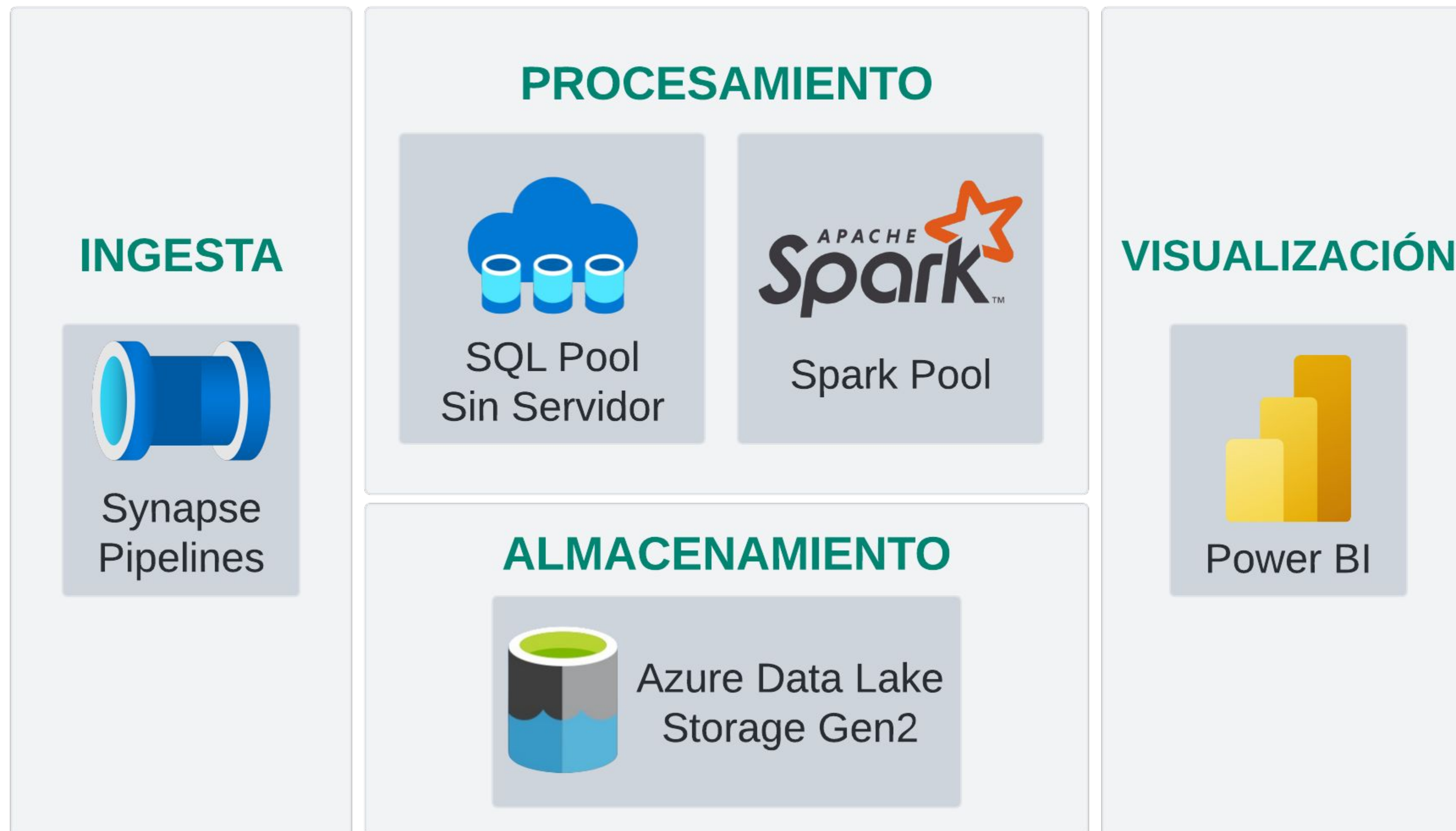
1. Introducción	3
2. Datos Disponibles	5
3. Descripción de Archivos	6
4. Virtualización de Datos	7
5. Ingesta de Datos	12
6. Transformación de Datos	14
7. Synapse Pipelines	16
8. Spark Pool	20
9. Integración de Power BI	22

1. INTRODUCCIÓN

El proyecto analiza los datos de taxis verdes en Nueva York, que operan en todas las áreas como Brooklyn, Queens y el Bronx, a diferencia de los taxis amarillos autorizados a operar únicamente en Manhattan. El objetivo es usar herramientas avanzadas de ingeniería de datos para optimizar los viajes en taxis verdes, cumplir con los requerimientos de negocio y generar cuadros de mando eficientes. Esto ayudará a mejorar la infraestructura de movilidad en la ciudad y proporcionar una arquitectura de datos robusta aplicable a futuros proyectos.

En cuanto al marco teórico, elegimos Azure Synapse como la plataforma central para el desarrollo del proyecto, ya que ofrece una solución unificada que permite gestionar todas las etapas del ciclo de vida de los datos, desde la ingesta hasta la visualización, superando las limitaciones de arquitecturas tradicionales fragmentadas. Esta plataforma combina las capacidades de Data Warehouses y Data Lakes, utilizando SQL Serverless Pool para manejar datos estructurados y Azure Data Lake Gen2 para almacenar grandes volúmenes de datos no estructurados. Con su capacidad de integración fluida y un enfoque end-to-end, azure Synapse es ideal para este proyecto de motor de ingesta, procesamiento y visualización.

El cuadro presentado a continuación, organiza las herramientas de Azure Synapse que vamos a utilizar, distribuidas en categorías clave para el flujo de datos y análisis:



2. DATOS DISPONIBLES

Los datos que estamos utilizando en este proyecto provienen de la web [Taxi & Limousine Commission \(TLC\)](#) de la Ciudad de Nueva York, específicamente de los taxis verdes, que cubren áreas periféricas de la ciudad. Estos datos incluyen información sobre recogidas, distancias y pagos, almacenados en archivos mensuales.

Además, se han creado archivos adicionales para enriquecer los datos tales como zonas, tipos de tarifas y métodos de pago, lo que facilita su procesamiento y segmentación. Esto optimiza el uso de herramientas como SQL Pool y Spark Pool, mejorando la integración y flexibilidad para futuras etapas del proyecto.

Se ha optado por trabajar con diferentes tipos de archivos en origen para demostrar la versatilidad de Synapse en la gestión y transformación de formatos a lo largo de las diferentes fases del proyecto. El siguiente esquema muestra los archivos disponibles:



3. DESCRIPCIÓN DE ARCHIVOS

Los archivos descritos a continuación se pueden encontrar en la carpeta `nyc-taxi-data/raw`.

- Zona de Taxi: proporciona una lista de zonas de taxis identificando cada área por su nombre y municipio. Incluye campos como id de Ubicación y Distrito, lo que facilita el análisis geográfico de los viajes y la movilidad entre zonas.
- Calendario: generado mediante un script de Python adjunto, incluye fechas, días de la semana, semanas, meses y años. Facilita el análisis temporal de los datos, ayudando a identificar patrones estacionales o tendencias diarias.
- Trip Data: contiene información detallada de los viajes, como fechas y horas de recogida y entrega, distancia recorrida, número de pasajeros y montos pagados. Este archivo es el centro del análisis y permite estudiar las características y generar modelos predictivos.
- Tipo de Viaje: identifica si el viaje fue recogido directamente en la calle o mediante despacho, lo que permite analizar la demanda según el tipo de servicio.
- Código de Tarifa: refleja el tipo de tarifa aplicada en cada viaje, como estándar o negociadas, y facilita el análisis del comportamiento de los usuarios según las propias tarifas.
- Tipo de Pago: define si el pago se realizó en efectivo, con tarjeta de crédito o si hubo alguna disputa. Es clave para entender las preferencias de pago de los usuarios.
- Proveedor: identifica las empresas proveedoras de los servicios de taxi, permitiendo analizar la participación de distintas compañías y su rendimiento.

4. VIRTUALIZACIÓN DE DATOS

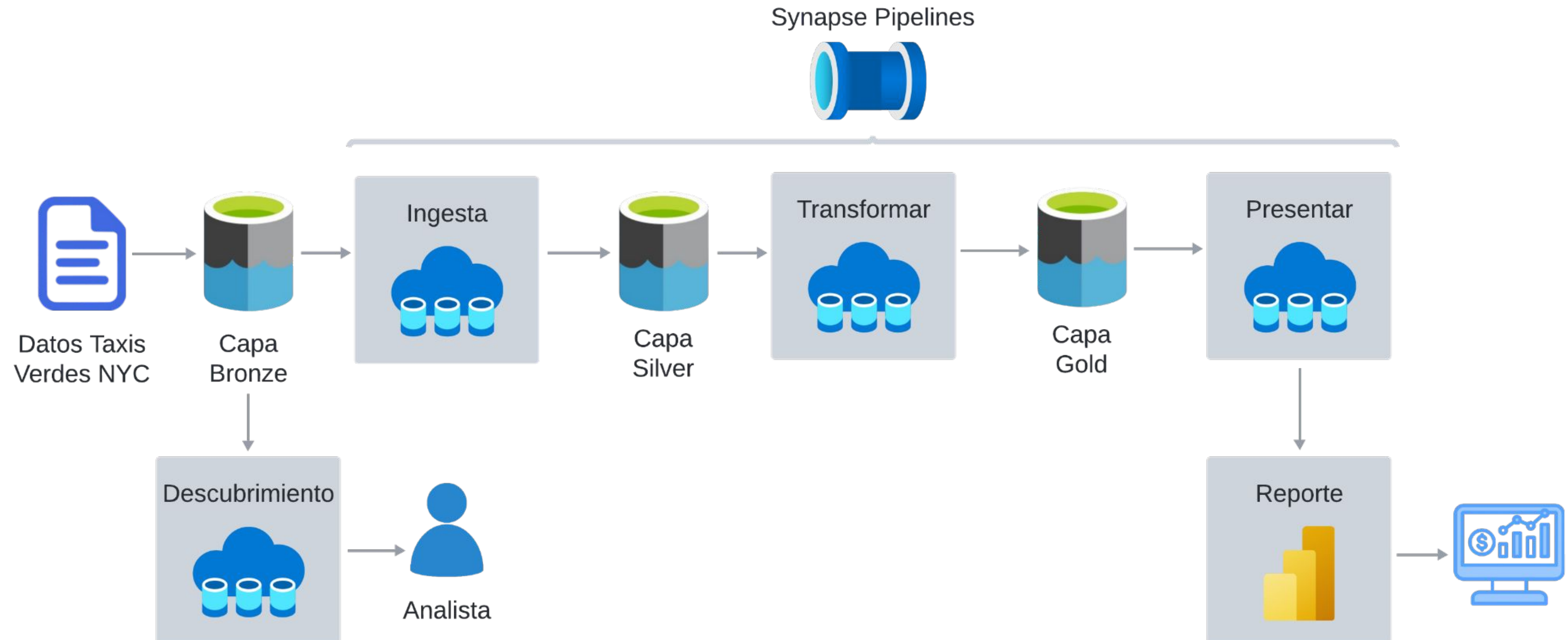
4.1 Requisitos

Vamos a definir los criterios que los datos deben cumplir, para garantizar una ejecución eficiente y que consigamos obtener resultados provechosos:

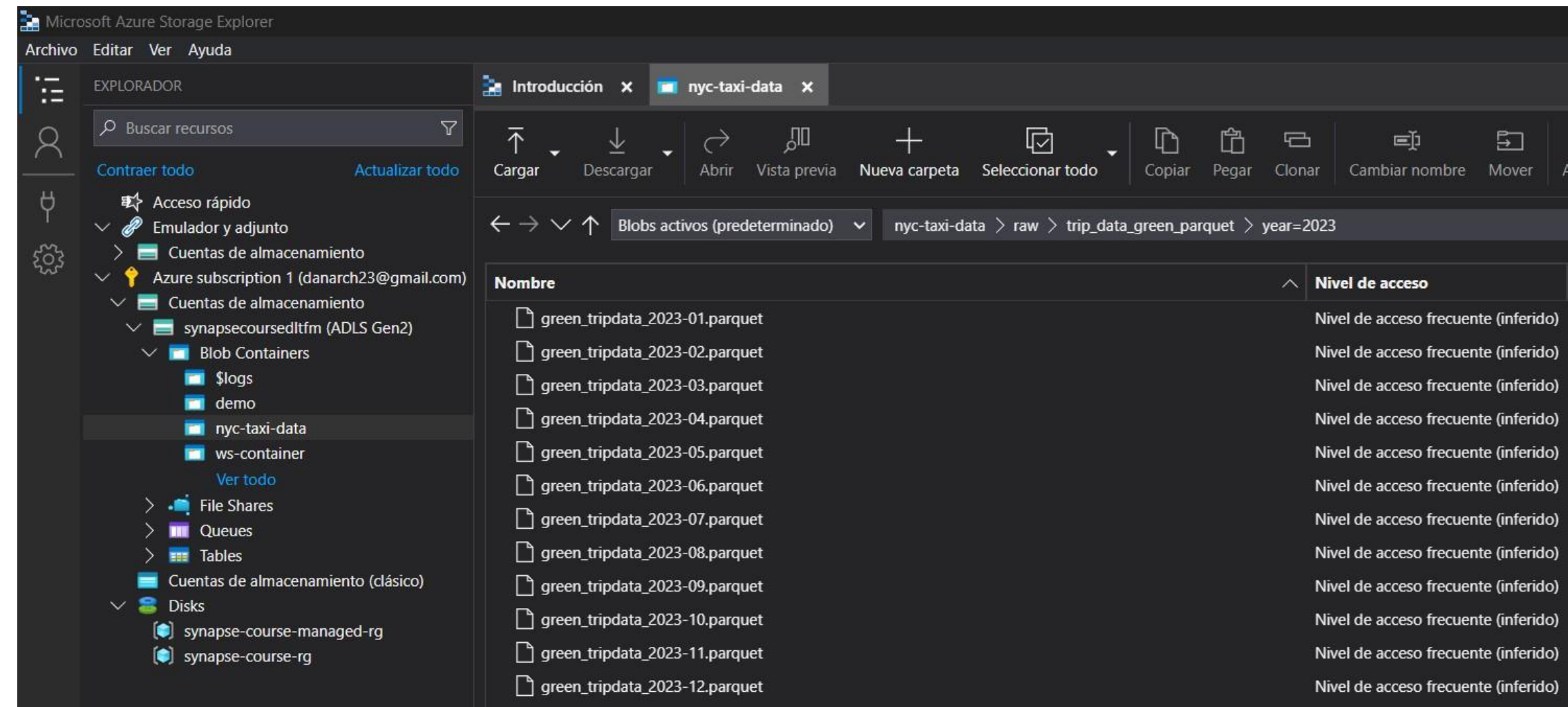
- Como vimos en el apartado anterior, ya hemos definido un esquema que facilita la comprensión de los datos, garantiza su calidad y valor empresarial antes de su ingesta.
- Utilizar un modelo de pago por consulta a través de una infraestructura sin servidor, para minimizar costos en esta fase inicial.
- Ingerir los siete archivos en formatos variados y almacenarlos en un formato columnar como parquet, con esquemas correctos y tipos de datos precisos.
- Unir datos clave, como los de viajes y zonas de taxis, para crear tablas listas para informes BI.
- Elaborar informes en Power BI sobre la demanda de taxis, según el día de la semana y la ubicación.

4.2 Propuesta de Arquitectura

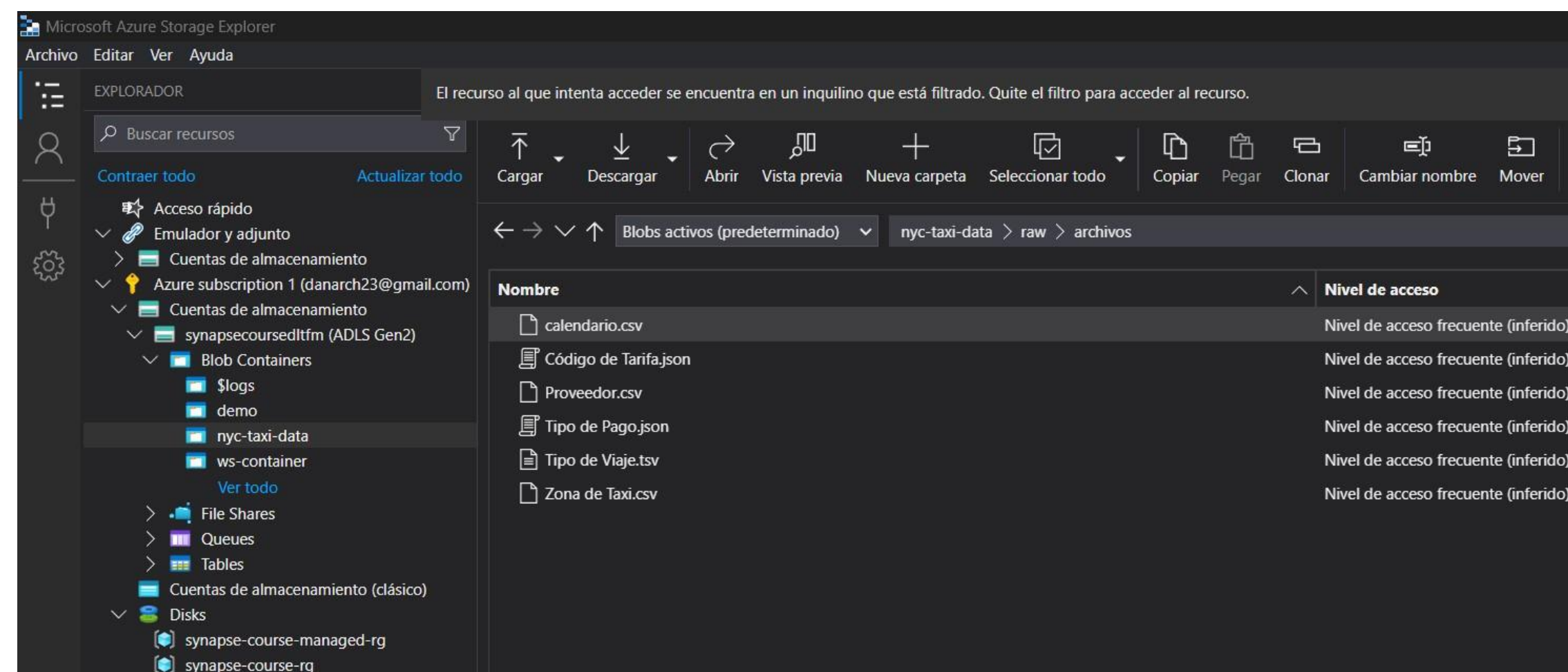
Para su diseño se han tenido en cuenta la naturaleza de los datos disponibles y los requisitos planteados anteriormente. En los apartados posteriores, describiremos el desarrollo de cada una de las fases de nuestro proyecto. El siguiente diagrama muestra el flujo de los datos a través de dichas fases:



4.3 Gestión de Archivos



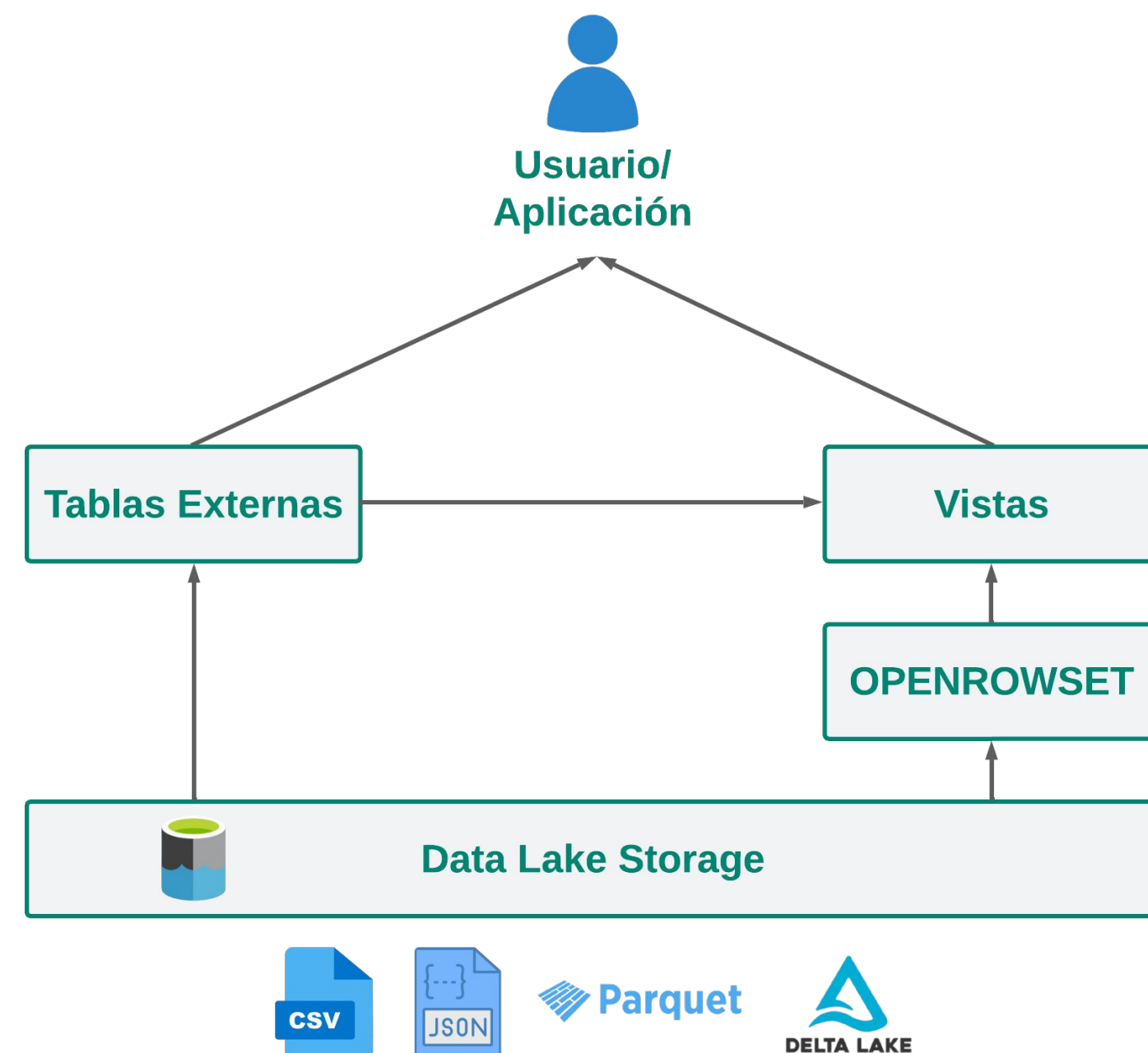
Utilizando Azure Storage Explorer, **en la ruta nyc-taxi-data/raw se han copiado los archivos “green_tripdata_202x- mm” desde diciembre del 2020 hasta junio del 2024**, y han sido replicados con diferentes formatos (csv, parquet e incluso alguno en delta a modo de prueba). El uso de ADLS Gen2 permite un almacenamiento optimizado y consultas rápidas con Synapse.



Tal y como se puede apreciar en la imagen izquierda, también se ha procedido a copiar los diferentes archivos descritos anteriormente. Se ha mantenido el formato con el que los hemos creado, con el objetivo de demostrar la versatilidad de Synapse a la hora de manejar diferentes tipos de archivo.

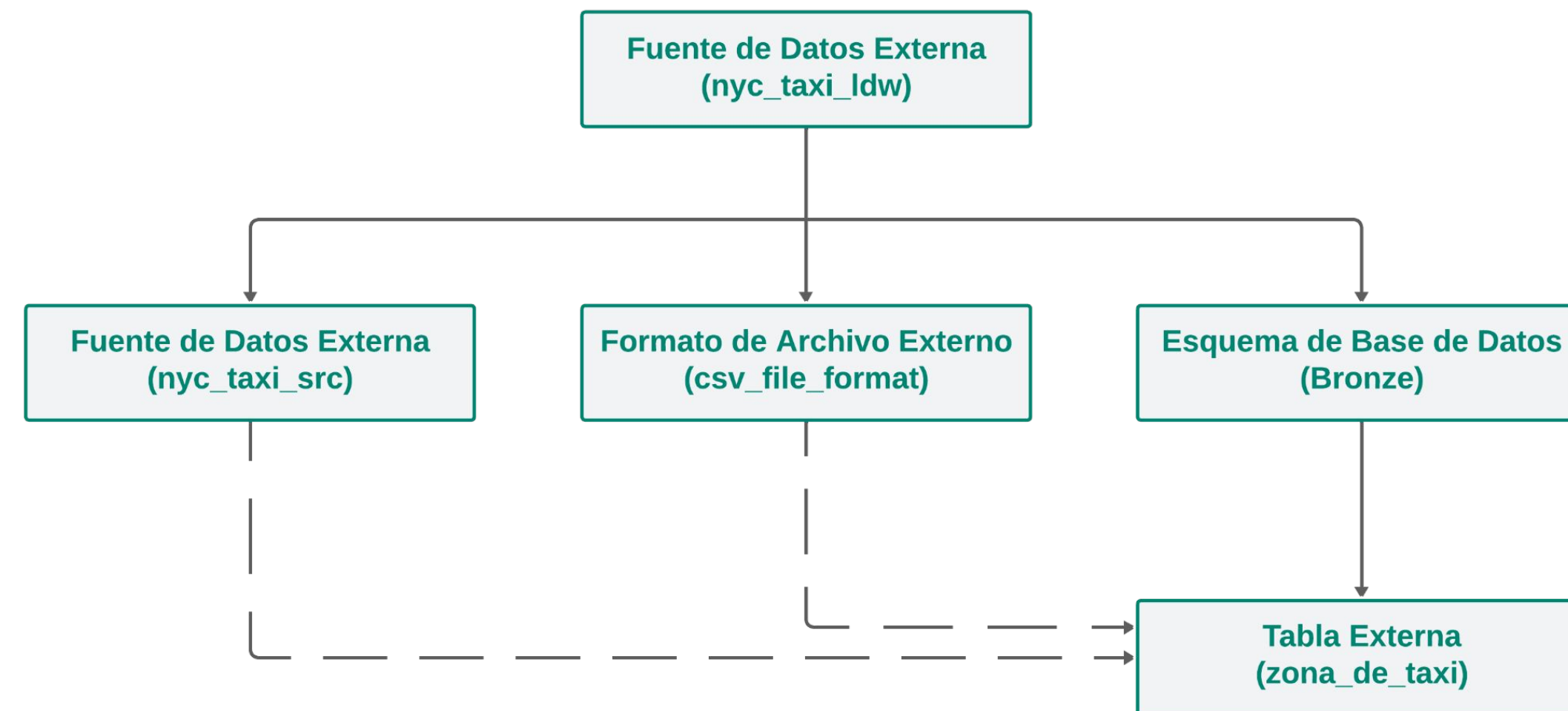
4.4 Descubrimiento de Datos

Hemos elegido Azure Serverless SQL Pool para procesar los datos porque elimina la necesidad de gestionar infraestructura, ofrece un modelo de pago por consulta, permite analizar datos no estructurados directamente desde almacenamiento, escala automáticamente y es compatible con T-SQL. Además, se integra fácilmente con Synapse Analytics para un análisis de datos más eficiente y rentable.



Se han creado objetos típicos de bases de datos como tablas externas o vistas, ya que abstraen la complejidad de la conexión a los datos, estandarizan el acceso a la información y permiten a los usuarios consultar los datos de manera más eficiente y sin tener que preocuparse por los detalles de almacenamiento o estructura de los archivos. **Todos los scripts desarrollados para descubrir los datos se pueden encontrar en la ruta `nyc_taxi/discovery`.**

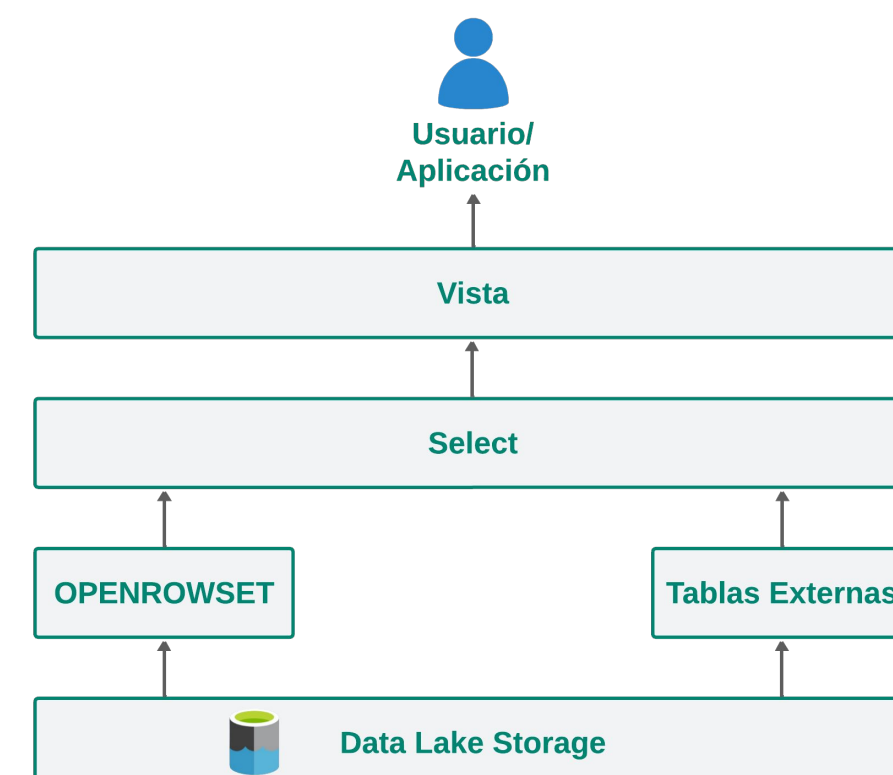
4.5 Creación de Tablas Externas



Permiten consultas rápidas en el Data Lake sin mover los datos, facilitando el análisis con distintos formatos y su integración con herramientas de BI. **Los scripts de creación de tablas externas se encuentran en la carpeta del repositorio nyc_taxi/ldw.** Se han generado tablas externas para todos los archivos descritos en apartados anteriores, exceptuando tipo_de_pago y codigo_de_tarifa que tienen formato json y que serán tratados posteriormente con la creación de vistas.

El diagrama izquierdo muestra cómo hemos creado una tabla externa para el archivo zona_de_taxi, utilizando una fuente de datos y formato externo, conectados al esquema de la base de datos y optimizando el acceso directo a los datos.

4.6 Creación de Vistas

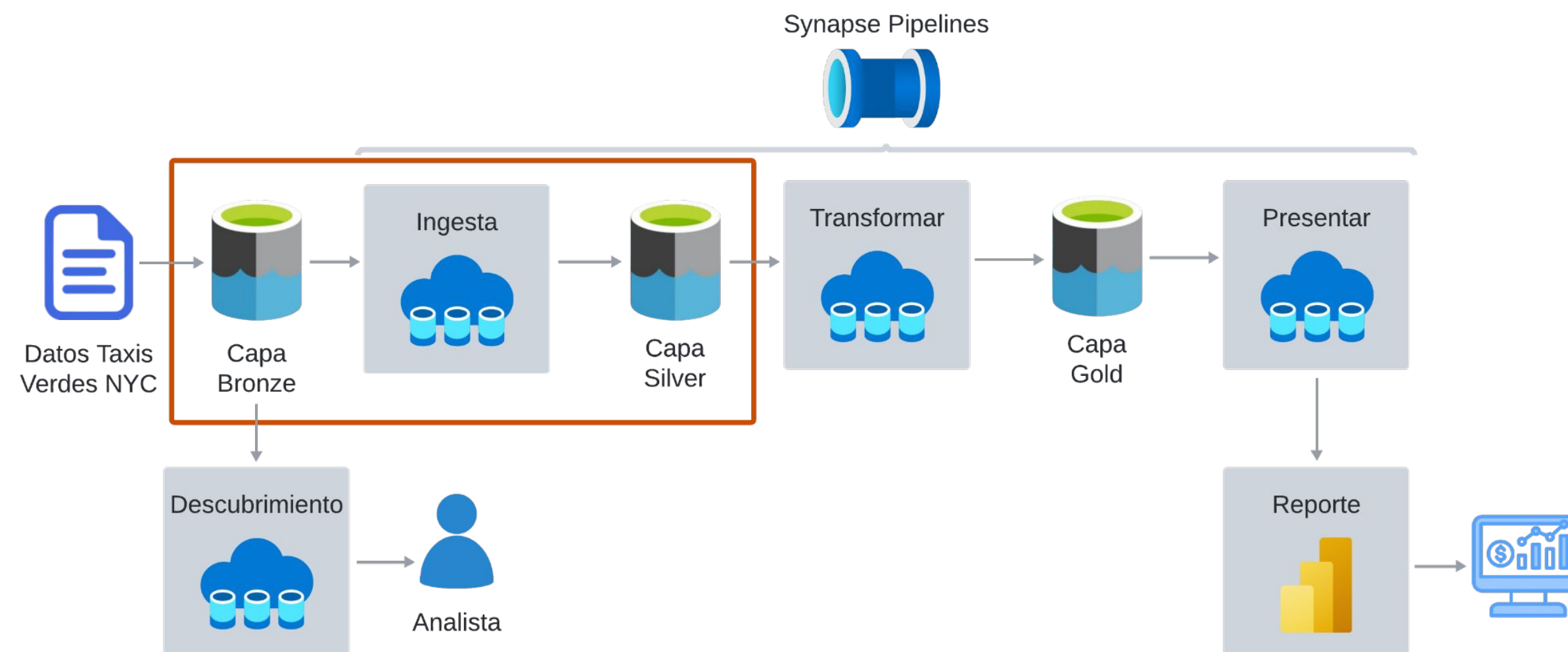


Las vistas nos permiten acceder a los datos de manera eficiente y estructurada, y sin moverlos del Data Lake. Se han creado vistas para los archivos json tipo_de_pago y codigo_de_tarifa, **y los scripts con los que hemos generado dichas vistas se encuentran en la carpeta del repositorio nyc_taxi/ldw.**

El diagrama izquierdo muestra cómo las vistas se apoyan en la función openrowset, optimizando así el uso del almacenamiento y los recursos computacionales.

5. INGESTA DE DATOS

5.1 Descripción



La ingesta de datos es la primera fase del proceso en el flujo de trabajo, donde los datos brutos de taxis verdes de NYC son cargados en la capa Bronze manteniendo su forma original. Como veremos más adelante, los datos serán transformados y mejorados en capas posteriores (silver y gold).

Para que los datos sean accesibles de manera eficiente y estén preparados para las siguientes fases (transformación, análisis y reporte), desarrollaremos una ingesta fundamentada en los siguientes pasos:

- Utilizamos cetags (create external table as select) para crear tablas externas y acceder a los datos en el Data Lake.
- Convertiremos los archivos csv a formato parquet dentro de la capa silver, para mejorar el rendimiento en las consultas.
- Los archivos json serán igualmente transformados a parquet en la capa silver, para estandarizar el almacenamiento.
- Abordaremos los desafíos de procesar datos particionados.
- Introduciremos procedimientos almacenados para automatizar el procesamiento de datos.
- Procesaremos los datos particionados para optimizar el análisis.
- Finalmente, crearemos vistas para facilitar el acceso y análisis de los datos transformados.

5.2 Conversión de Archivos

Se han convertido los archivos CSV y JSON de la capa bronze a formato parquet en la capa silver mediante CETAS, lo que optimiza las consultas y reduce el consumo de almacenamiento gracias a la compresión eficiente, además de permitir lecturas más rápidas debido a su estructura columnar. **Los scripts de conversión que hemos desarrollado se encuentran en la carpeta del repositorio nyc_taxi/ldw.**

5.3 Optimización con Stored Procedures

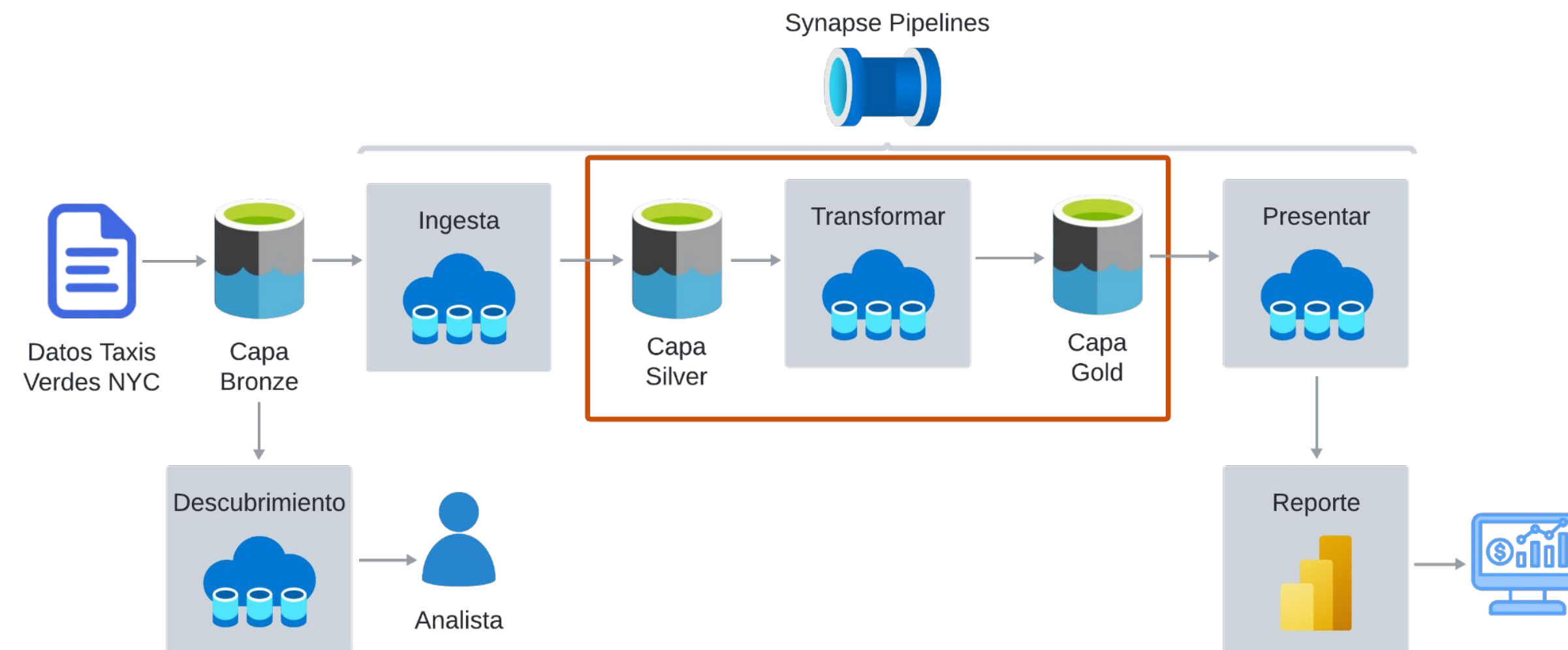
SQL Serverless Pool no permite la generación dinámica de archivos parquet desagregados por año y mes, ya que Azure Synapse Serverless SQL Pool tiene una limitación que uniría todos los datos en un único archivo Parquet por defecto. Por lo que hemos desarrollado stored procedures para generar los archivos Parquet correspondientes a cada mes y año en la capa Silver, partiendo de las vistas CSV en la capa Bronze. **Todos los scripts de generación de stored procedures se encuentran en nyc_taxi/ldw/usp.**

5.4 Creación de vistas

Generar la vista a partir de los archivos Parquet de la capa silver con columnas de año y mes permite filtrar datos fácilmente por estos campos, simplificando las consultas sin necesidad de manipular rutas o nombres de archivos. Además, las consultas que requieren analizar datos por periodos específicos, como meses o años, se vuelven más rápidas y sencillas. Los nuevos datos que se añadan a la carpeta silver se integran automáticamente en la vista, facilitando la gestión de grandes volúmenes de datos a lo largo del tiempo. **Los scripts de creación de vistas se encuentran en la carpeta del repositorio nyc_taxi/ldw.**

6. TRANSFORMACIÓN DE DATOS

6.1 Descripción



La transformación de datos desde la capa silver a la capa gold es crucial en nuestro proyecto, ya que permite depurar y estructurar los datos de manera que puedan cumplir con los requisitos de negocio y ser consumidos eficientemente para análisis y generación de informes. La capa silver contiene datos parcialmente procesados, mientras que la gold almacena datos finales optimizados para consultas y reportes.

A continuación describimos los objetivos que buscamos alcanzar esta fase de nuestra arquitectura:

- Identificar los requerimientos del proyecto para definir cómo los datos deben transformarse y qué resultados se esperan de la capa gold.
- Los datos se preparan para análisis de campañas, lo que nos ayudará a evaluar el impacto de las acciones y decisiones comerciales.
- Los datos se transforman para obtener insights sobre la demanda de taxis, lo que es clave para nuestro análisis y la toma de decisiones basadas en patrones de uso.
- Finalmente, se crean vistas para facilitar el acceso a los datos transformados en la capa gold, optimizando su consulta desde herramientas de BI como Power BI.

6.2 Requerimientos de Negocio

Se busca fomentar el pago con tarjeta en los taxis verdes de Nueva York. Para cumplir con estos objetivos, se definen tres áreas clave de análisis: el número de viajes pagados en efectivo o con tarjeta, el comportamiento de pago según los días de la semana y fines de semana, y el análisis de pagos por boroughs. Además, se establecen requerimientos no funcionales en la capa gold para mejorar el rendimiento, como la pre-agregación de datos y la lectura eficiente de grandes volúmenes por meses o años, minimizando la creación de tablas agregadas.

6.3 Transformación con Stored Procedures

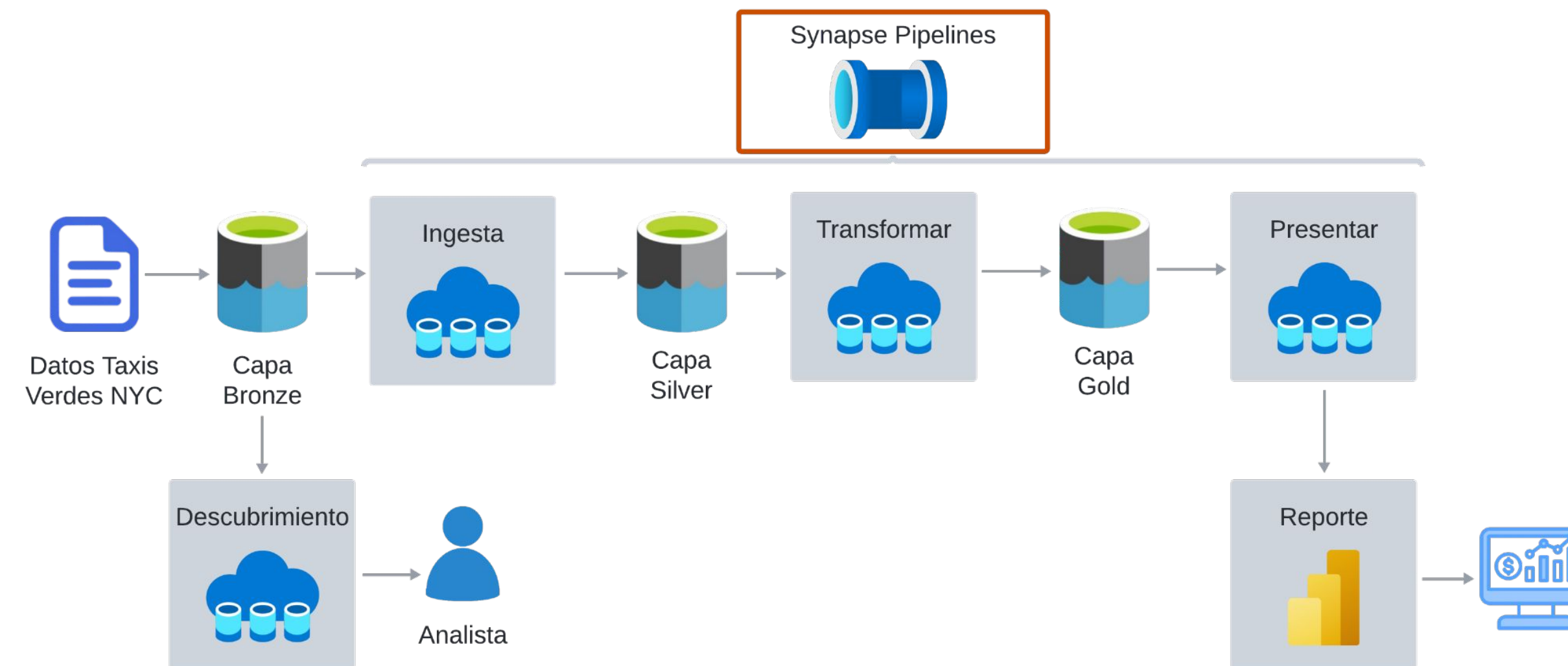
La transformación de datos se realiza mediante stored procedures que automatizan la creación de archivos parquet en la capa gold. Estos stored procedures agregan y transforman datos por año y mes, calculando indicadores como el número de viajes pagados en efectivo y tarjeta, y si ocurrieron en fines de semana. El proceso asegura que los datos estén estructurados y optimizados para el análisis, integrando diferentes tablas de la capa silver, y eliminando tablas temporales para optimizar el almacenamiento. **Todos los scripts de generación de stored procedures se encuentran en `nyc_taxi/ldw/usp`.**

6.4 Creación de vistas

Generar la vista a partir de los archivos Parquet de la capa silver con columnas de año y mes permite filtrar datos fácilmente por estos campos, simplificando las consultas sin necesidad de manipular rutas o nombres de archivos. Además, las consultas que requieren analizar datos por periodos específicos, como meses o años, se vuelven más rápidas y sencillas. Los nuevos datos que se añadan a la carpeta silver se integran automáticamente en la vista, facilitando la gestión de grandes volúmenes de datos a lo largo del tiempo. **Todos los scripts de creación de vistas se encuentran en `nyc_taxi/ldw`.**

7. SYNAPSE PIPELINES

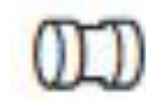
7.1 Descripción

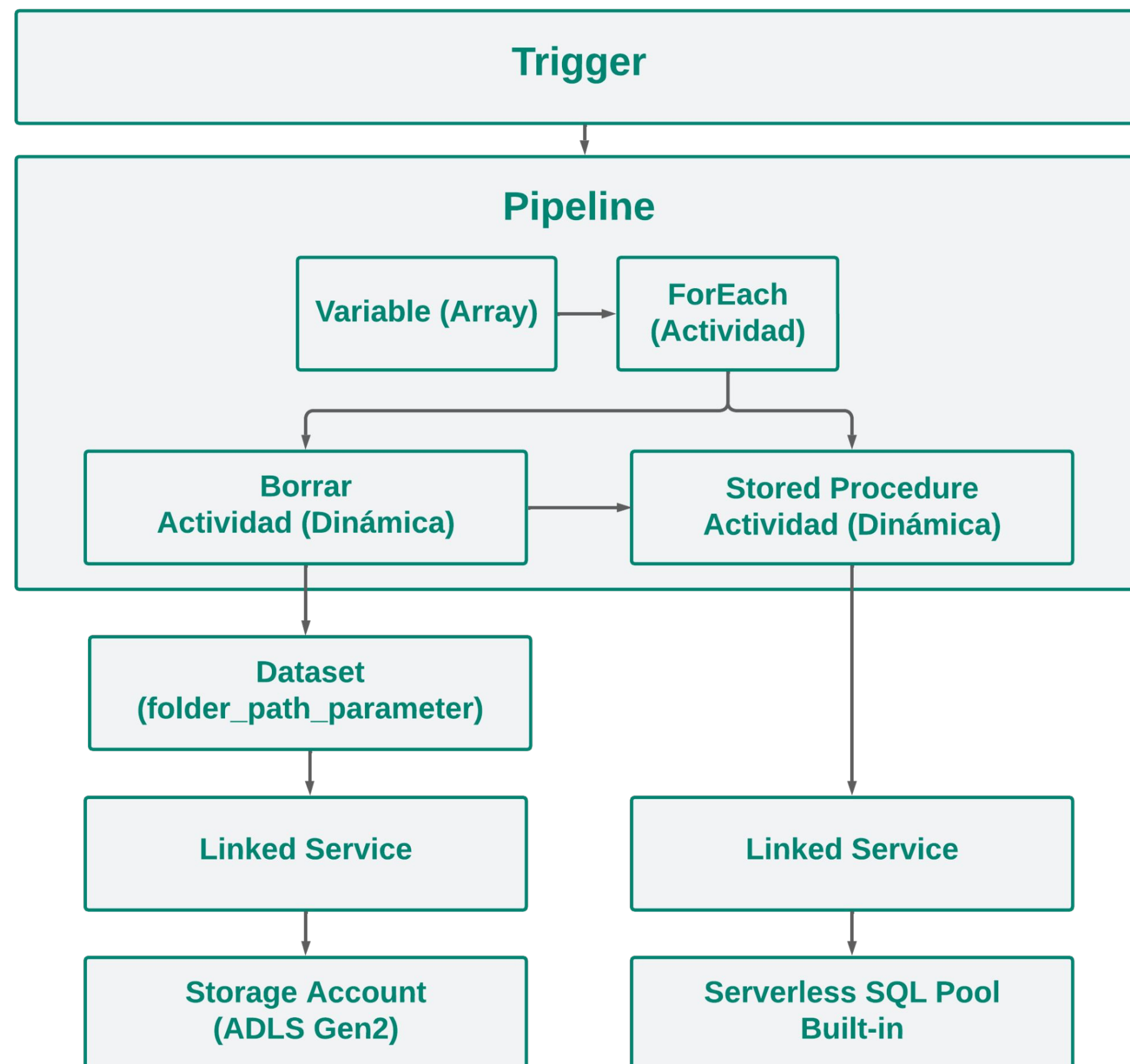


Nuestro objetivo es automatizar el procesamiento de datos de forma continua y eficiente entre las distintas capas bronze, silver y gold. Para ello, empleamos Synapse Pipelines, una herramienta clave en nuestro proyecto que permite la orquestación de tareas como la ingesta de datos, la conversión de archivos, la creación de tablas, vistas y la ejecución de stored procedures de manera dinámica. Las pipelines que describiremos a continuación han sido diseñadas para:

- Procesar múltiples archivos de diferentes formatos sin duplicar actividades, estableciendo las conexiones necesarias a servicios de almacenamiento y procesamiento.
- Optimizar el manejo de los datos y garantizar flexibilidad en la transformación de los mismos.
- Automatizar y monitorear la puesta en marcha, parada y frecuencia de ejecución de las pipelines, gracias al uso de triggers.

7.2 Pipeline Dinámica Para Todos los Archivos Salvo Trip Data (Bronze a Silver)

 pl_create_silver_tables



Necesitamos evitar la duplicación innecesaria de actividades y pipelines individuales para cada uno de los seis archivos parquet (excluyendo trip data por su lógica particular de partición), en su conversión de bronze a silver. El uso de una pipeline dinámica nos permite automatizar el procesamiento de cada tipo de archivo sin tener que replicar actividades manualmente. Lo que optimiza el tiempo y recursos en la gestión de datos, manteniendo la eficiencia y simplicidad en la arquitectura del proyecto. El flujo de esta pipeline dinámica se basa en:

- **Arrays y Variables:** para almacenar tanto las rutas de las carpetas de los archivos como los nombres de los Stored Procedures correspondientes para cada archivo.
- **Actividad ForEach:** esta actividad recorre el array y ejecuta las operaciones necesarias para cada archivo en la iteración, invocando los parámetros almacenados.
- **Actividades Dinámicas:** eliminar datos antiguos, limpiando las ubicaciones de almacenamiento y ejecutar un Stored Procedure que procesa cada archivo en su iteración, utilizando el nombre y ruta extraídos del array.
- **Dataset y Linked Services:** se manejan dinámicamente los datos de los archivos parquet y se vinculan a los servicios de almacenamiento y cómputo en Azure.

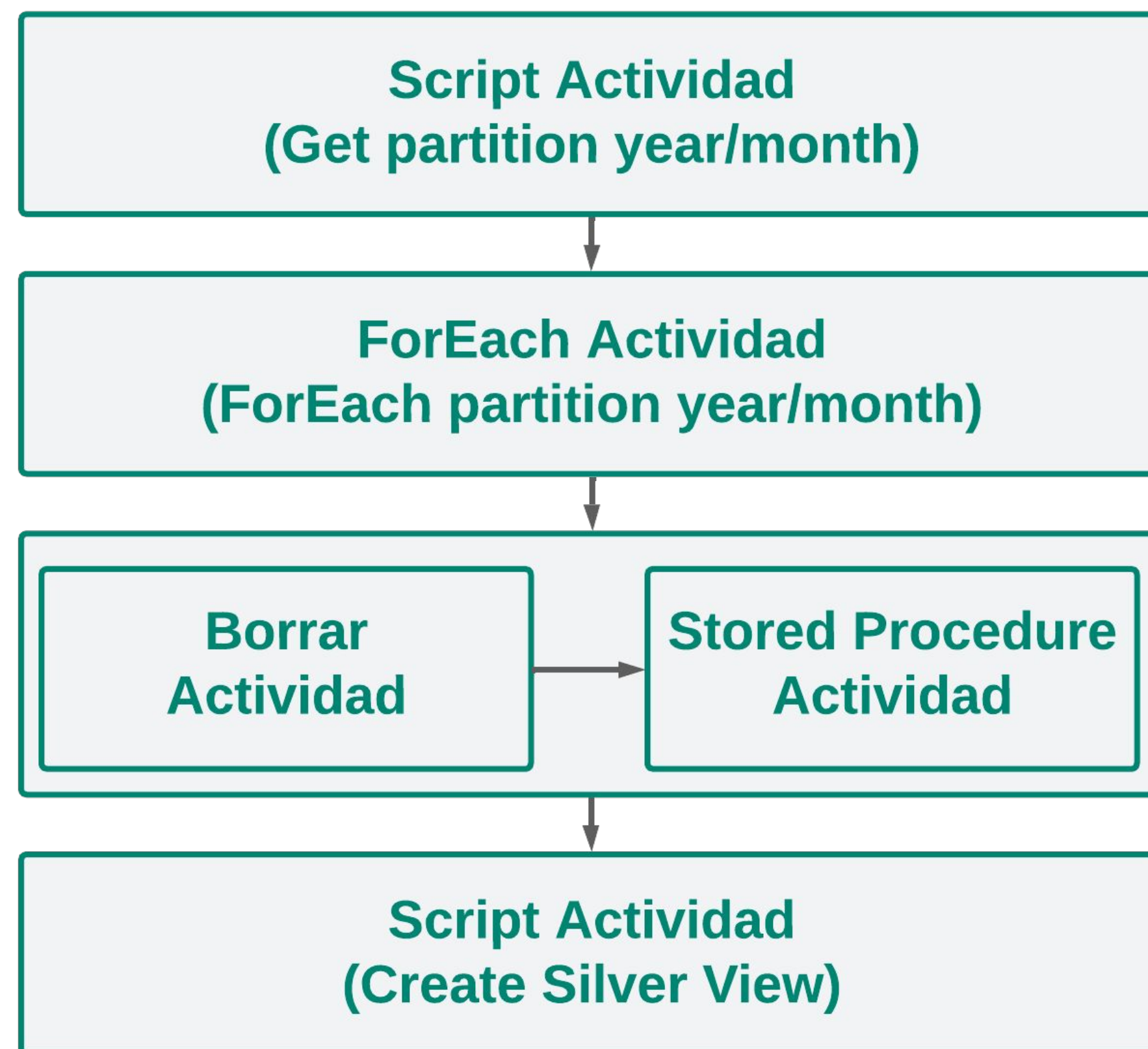
Estos factores aseguran un procesamiento centralizado y automatizado para todos los tipos de archivos que no son trip data.

7.3 Pipelines Para Trip Data (Bronze a Silver y Silver a Gold)

📖 pl_create_silver_trip_data_green

📖 pl_create_gold_trip_data_green

Pipeline

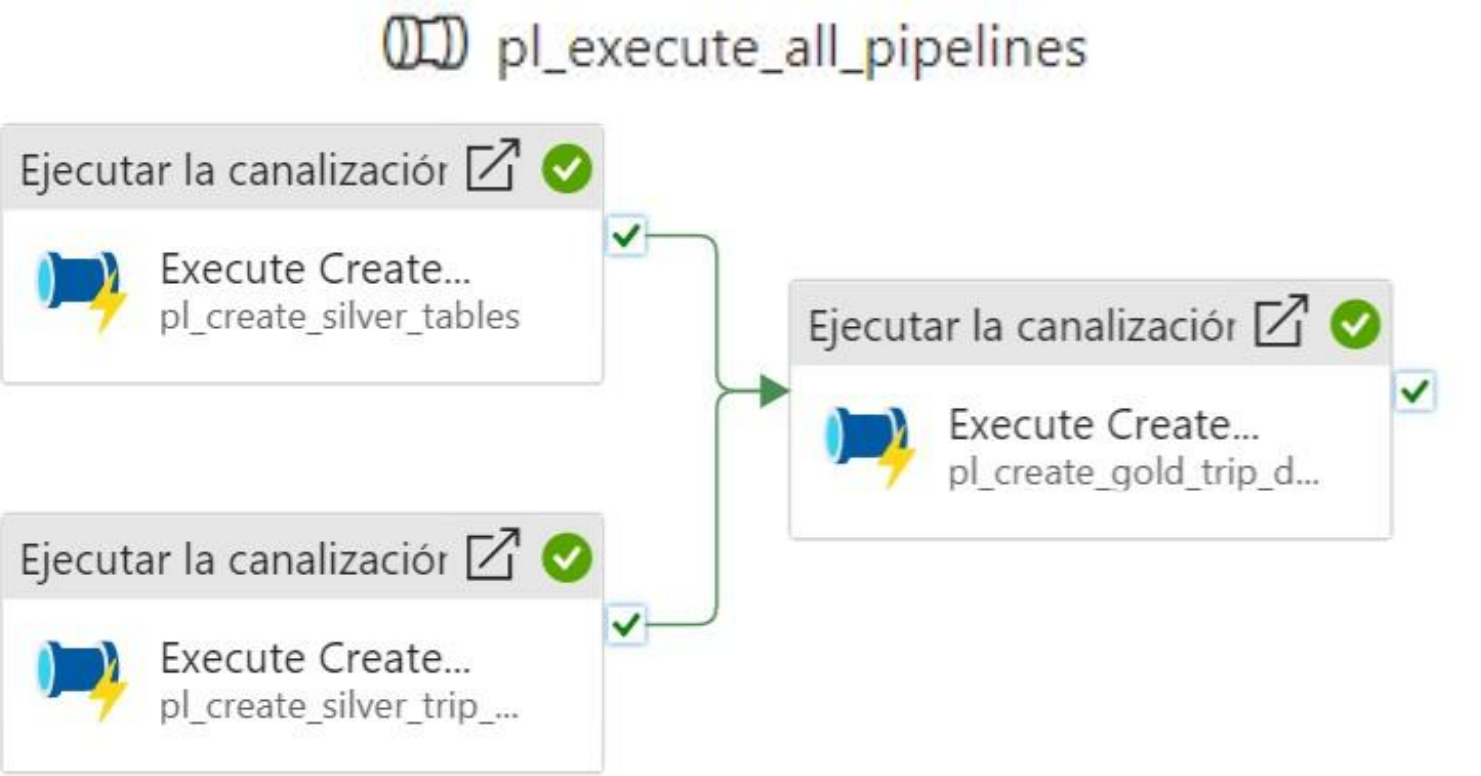


Hemos creado un nuevo tipo de pipeline para procesar los archivos de trip data, debido a que estos están particionados por año y mes. A diferencia de los otros seis tipos de archivo, que no siguen esta estructura. Por lo que se requiere un procesamiento más detallado para cada partición temporal (año/mes) de trip data, garantizando que los datos sean tratados correctamente en la conversión de bronze a silver y de silver a gold. El flujo de este tipo de pipeline sigue el siguiente esquema:

- Actividad Script: se inicia la pipeline con un script que identifica todas las particiones de los archivos trip data, es decir, los diferentes años y meses que se encuentran en las carpetas.
- Actividad ForEach: luego, se itera sobre cada partición (año y mes) mediante la actividad ForEach, asegurando que cada conjunto de datos particionado sea procesado individualmente.
- Actividad Borrar y Actividad Stored Procedure: durante cada iteración, primero se ejecuta una actividad que borra la partición correspondiente, seguida de la ejecución de un stored procedure que crea las tablas o procesa los datos de esa partición de manera adecuada.
- Actividad Script: al final del ciclo, se ejecuta un script que crea o actualiza una vista que facilita el acceso a los datos procesados de la capa Silver, permitiendo así que los analistas trabajen con los datos consolidados.

7.4 Pipeline Maestra y Trigger

pl_execute_all_pipelines



Parámetros Variables Configuración **Salida**

Id. de ejecución de canalización: b0a3117a-ef51-47b6-ba7d-6dbfb02d183b [🔗] [🔄] [📄]

All status ▾

Mostrando elementos del 1 al 3 de un total de 3

Nombre de actividad ↑↓	Estado de actividad ↑↓
Execute Create Gold Trip Data Green	✓ Correcto
Execute Create Silver Table	✓ Correcto
Execute Create Silver Trip Data Green	✓ Correcto

Nombre ↑↓	Tipo ↑↓
tr_nyc_taxi_data_load	Schedule

Se ha creado una pipeline maestra para orquestar todas las pipelines descritas anteriormente en actividades, provocando un flujo de trabajo de procesamiento de datos eficiente y automatizado. Esta metodología proporciona varias ventajas clave:

- **Automatización Completa:** al ejecutar todas las pipelines a través de una única pipeline maestra, podemos asegurarnos de que todas las tareas de procesamiento de datos se ejecuten sin intervención manual. Esto incluye la conversión de archivos desde la capa bronze a silver y de silver a gold.
- **Control Centralizado:** en lugar de ejecutar cada pipeline de manera aislada, la pipeline maestra coordina todo el proceso. Esto simplifica la gestión y asegura que todas las actividades necesarias se completen en el orden correcto.
- **Uso Eficiente de Recursos:** al tener un control global sobre todas las pipelines, podemos optimizar el uso de recursos, asegurando que las tareas no se solapen innecesariamente y que las dependencias entre pipelines se respeten.
- **Trigger Programado:** el ScheduledTrigger que se ha implementado añade una capa adicional de automatización, permitiendo definir cuándo iniciar, detener y con qué frecuencia ejecutar la pipeline maestra. Esto nos permite programar la ejecución en función de las necesidades del proyecto, como procesamiento diario, semanal o mensual, asegurando que los datos estén siempre actualizados sin intervención manual.

8. SPARK POOL

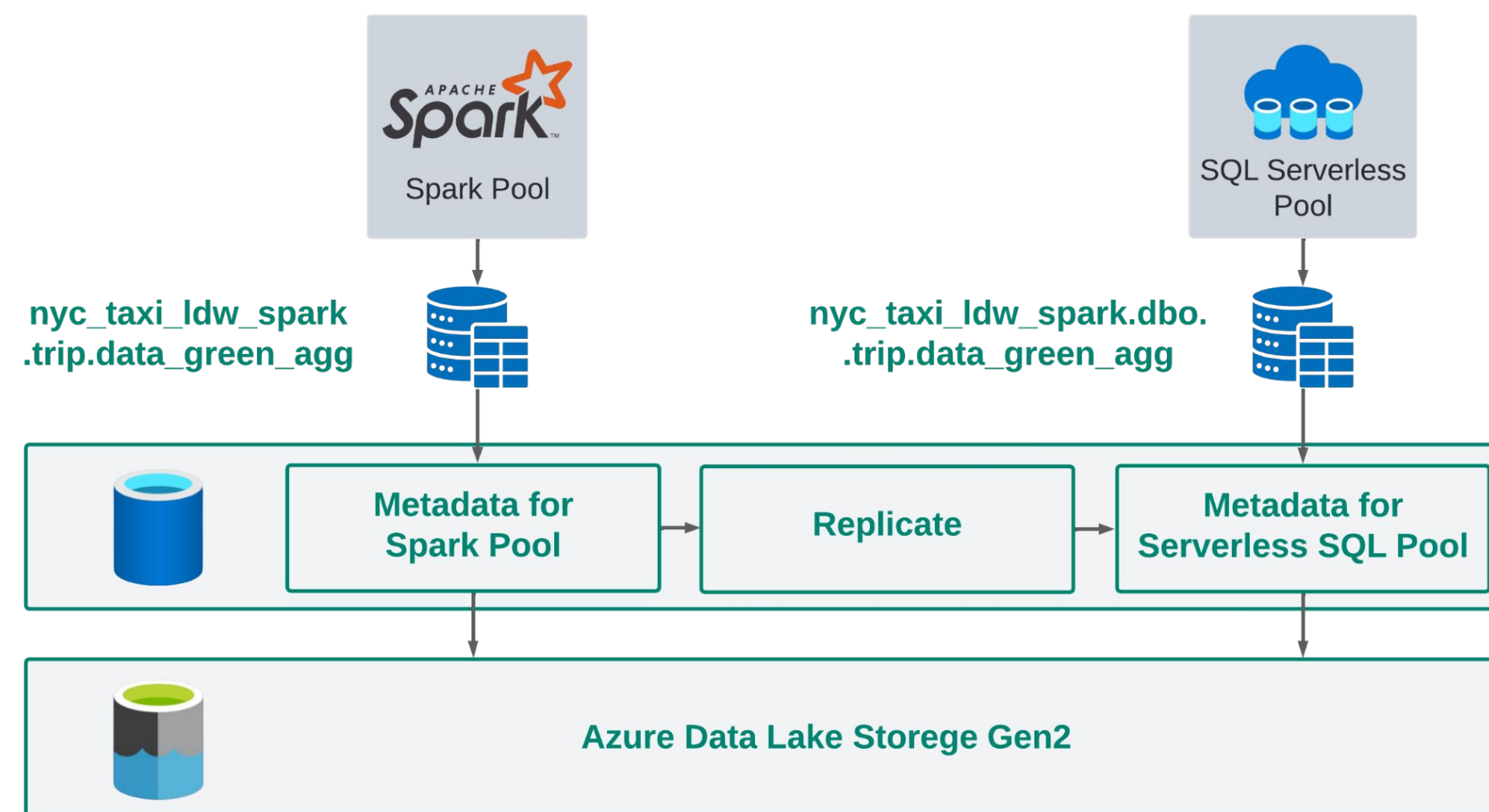
8.1 Integración Entre Spark Pool y SQL Serverless Pool

La integración de Spark Pool con SQL Serverless Pool en Azure Synapse nos permite procesar grandes volúmenes de datos de manera eficiente con Spark y luego hacer accesibles esos datos a través de consultas SQL en Serverless Pool. Spark se utiliza para realizar las agregaciones y transformaciones complejas, mientras que SQL Serverless Pool facilita consultas rápidas sin necesidad de manejar el procesamiento intensivo.

Agregación de Datos de Silver y Creación de Spark Table en Spark Pool



Replicación de Spark Table en SQL Serverless Pool

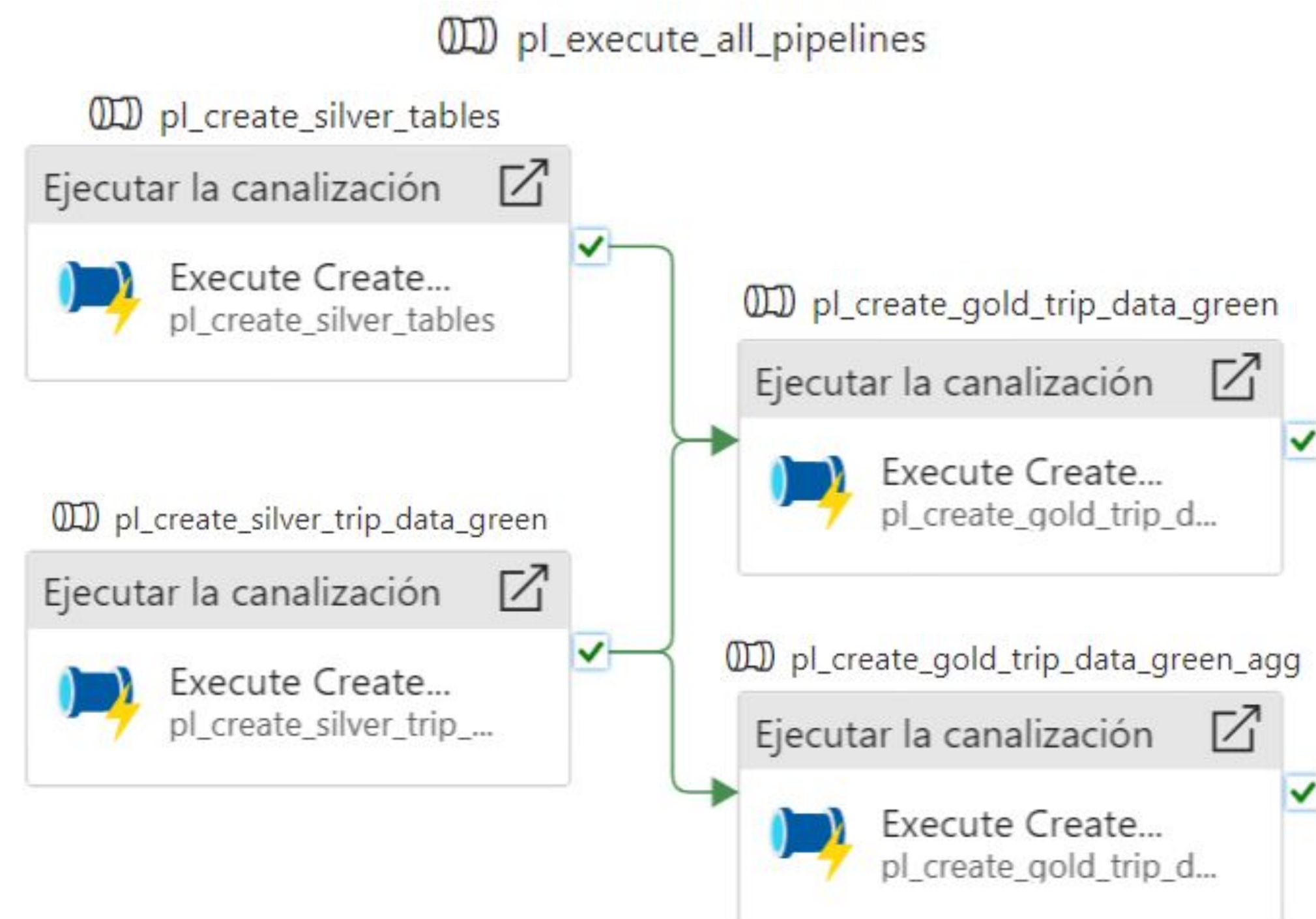


Vamos a complementar la pipeline maestra, procesando los datos de la capa Silver (específicamente de trip_data_green) y agregándolos utilizando un Spark Pool, que es ideal para manejar grandes volúmenes de datos distribuidos. Posteriormente, se creará una tabla agregada en el Spark Pool con estas transformaciones y se almacenará en la capa Gold con el nombre trip_data_green_agg.

Después de crear la tabla en Spark, se replicará la tabla en SQL Serverless Pool, permitiendo acceder y consultar los datos con SQL estándar de una manera más accesible. Esta replicación implica almacenar los metadatos de la tabla tanto en Spark como en SQL Serverless Pool, con la ventaja de que las consultas pueden hacerse directamente sin tener que manejar el procesamiento intensivo en Spark.

8.2 Desarrollo en Spark e Integración en Pipeline Alternativa

Se ha desarrollado el cuaderno **1_spark_create_gold_trip_data_green_agg** para procesar y agregar los datos de viaje desde la capa Silver, realizando cálculos como el total de viajes y el monto total por zonas de recogida y destino, agrupados por año y mes. Luego, guarda estos datos agregados en la capa Gold en formato Parquet (trip_data_green_agg en Gold), particionados por año y mes. Además, replica los metadatos de la tabla creada en Spark Pool para que sean accesibles desde SQL Serverless Pool, lo que permite que herramientas como Power BI o Azure ML consuman los datos sin necesidad de usar Spark directamente.

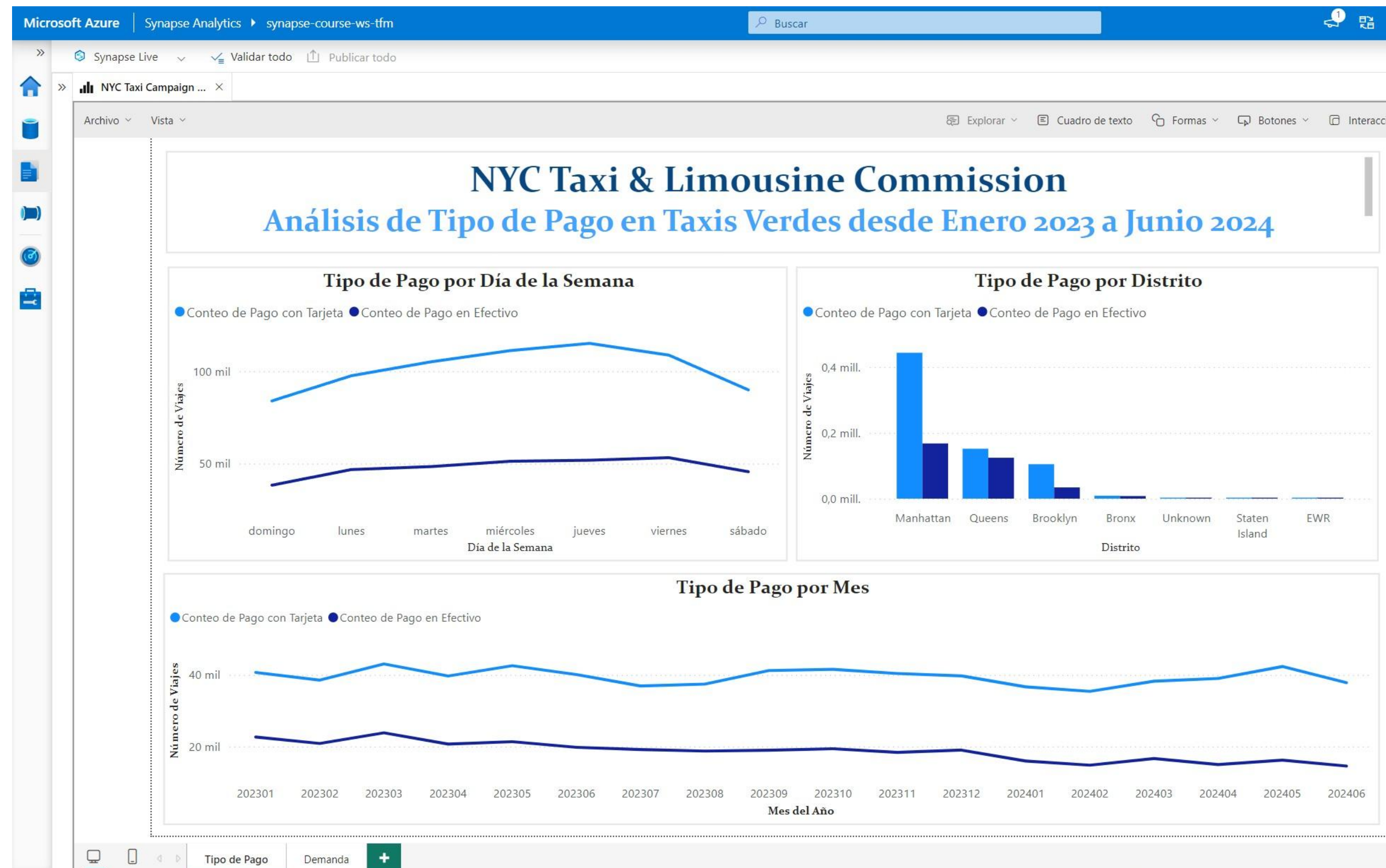


La nueva distribución de la pipeline maestra integra la pipeline **pl_create_gold_trip_data_green_agg**, que realiza agregaciones sobre los datos de trip data en Silver y los carga en Gold usando Apache Spark. Esta pipeline permite análisis eficientes, como el número total de viajes o el monto total de tarifas, almacenando los datos agregados en formato Parquet.

Además, incluye una salida alternativa mediante la integración entre Spark Pool y Serverless SQL Pool, lo que facilita el acceso directo a los datos agregados. Esto optimiza el procesamiento en Spark mientras Serverless SQL actúa como interfaz ligera para consultas y análisis.

9. INTEGRACIÓN DE POWER BI

Por último, hemos procedido a integrar Power BI en Synapse y creado el cuadro de mando **NYC Taxi Campaign Analysis** a partir de la vista **gold.vw_trip_data_green**. Pudiendo de esta manera visualizar datos de enero de 2023 a junio de 2024 y que dan respuesta a los siguientes requerimientos de negocio.



- Organizar una campaña para aumentar el pago con tarjeta: la pestaña Tipo de Pago muestra los acumulados de pago con tarjeta o en efectivo en base al día de la semana, distrito o mes del año, facilitando el éxito de la campaña.
- Identificar la demanda de taxis: la pestaña Demanda refleja los acumulados de viajes por día de la semana, distrito y mes, optimizando la asignación de recursos y mejorando la eficiencia operativa del servicio de taxis.

El cuadro de mando **NYC Taxi Campaign Analysis** se muestra y describe con mayor detalle en el vídeo adjunto al proyecto.