**Proyecto Final: Esperanza de Vida al Nacer**





Somos una consultora especializada en análisis de datos y toma de decisiones basada en datos. Nuestro equipo está conformado por profesionales altamente capacitados y con amplia experiencia en proyectos de consultoría de datos, lo que nos permite abordar de manera integral proyectos de diversa complejidad y magnitud.

Nuestra misión es ayudar a nuestros clientes a obtener información valiosa a partir de sus datos para mejorar su toma de decisiones y optimizar sus procesos. Nos destacamos por brindar soluciones personalizadas, adaptadas a las necesidades y objetivos específicos de cada cliente. Trabajamos de manera colaborativa, estableciendo una estrecha relación con nuestros clientes para comprender a fondo su problemática y ofrecerles las mejores soluciones.

👨‍💻 **Semana 2: Data Engineering**

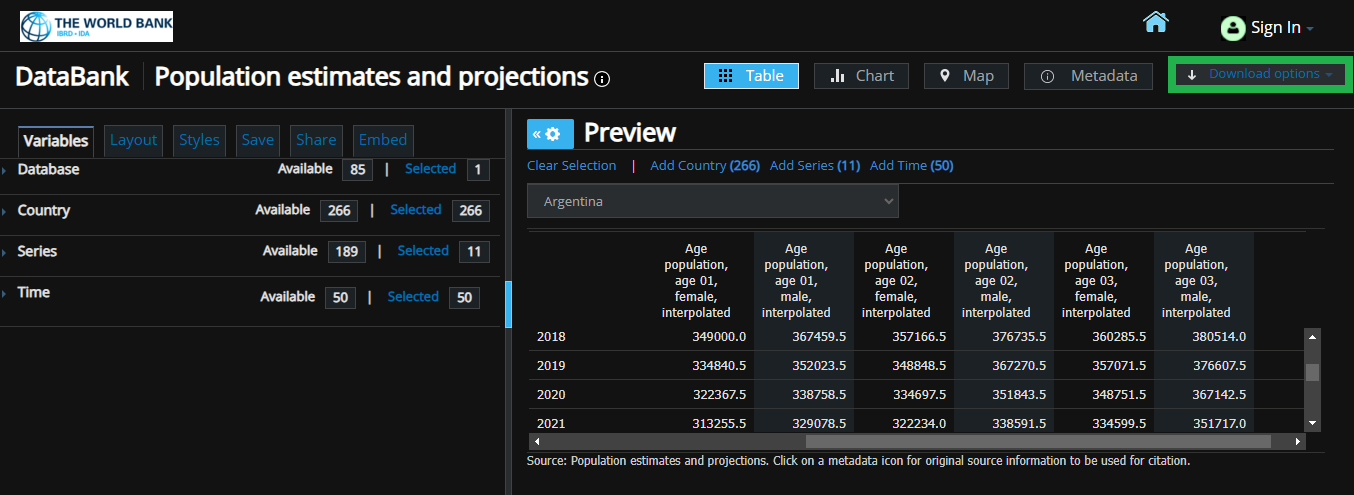
**Variedad de fuentes de datos:**

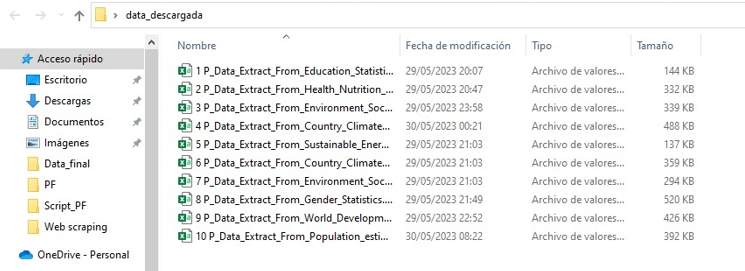
Al utilizar tanto la descarga de archivos CSV del Banco Mundial como la API de la OMS, se ha aprovechado la variedad de fuentes de datos disponibles. Esto permite obtener información diversa y enriquecedora de diferentes organismos y fuentes confiables, lo que a su vez mejora la calidad y la relevancia de los datos extraídos.

Te presentamos el proceso de obtención y extracción de datos, donde utilizamos tanto fuentes estáticas como dinámicas.

* **Data estática (.CSV)**

Desde la página del Banco Mundial, realizamos una selección de los indicadores relevantes para nuestro análisis. Aplicamos filtros según los países, años y series de datos que nos interesan, para obtener una información más específica y precisa. Una vez configurados los parámetros de búsqueda, procedemos a descargar los datos en formato CSV.



Para completar el proceso de descarga de datos, se siguió el mismo enfoque que se muestra en la imagen para los otros .CSV, obteniendo los 10 datasets:

* **Data dinámica (API)**

Paso a detallar descarga descarga de la API OMS:

Import request: para realizar solicitudes HTTP

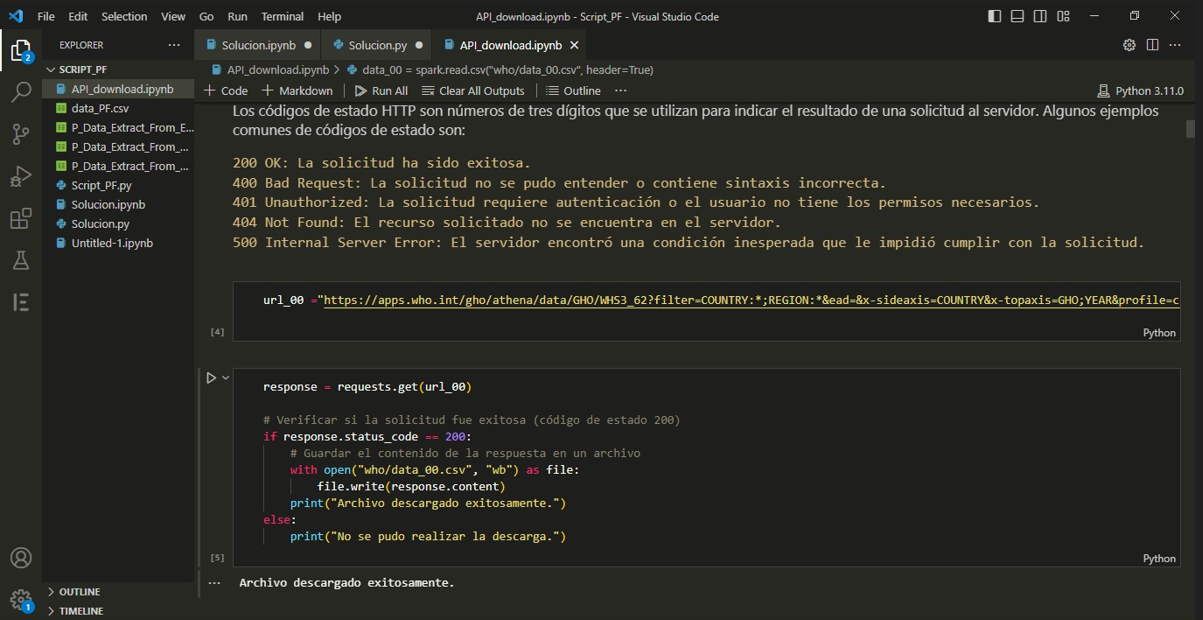
Import io: tipo E/S de texto, binario o sin formato

import csv: manipulación de archivos csv.

Importación de la librería pyspark.

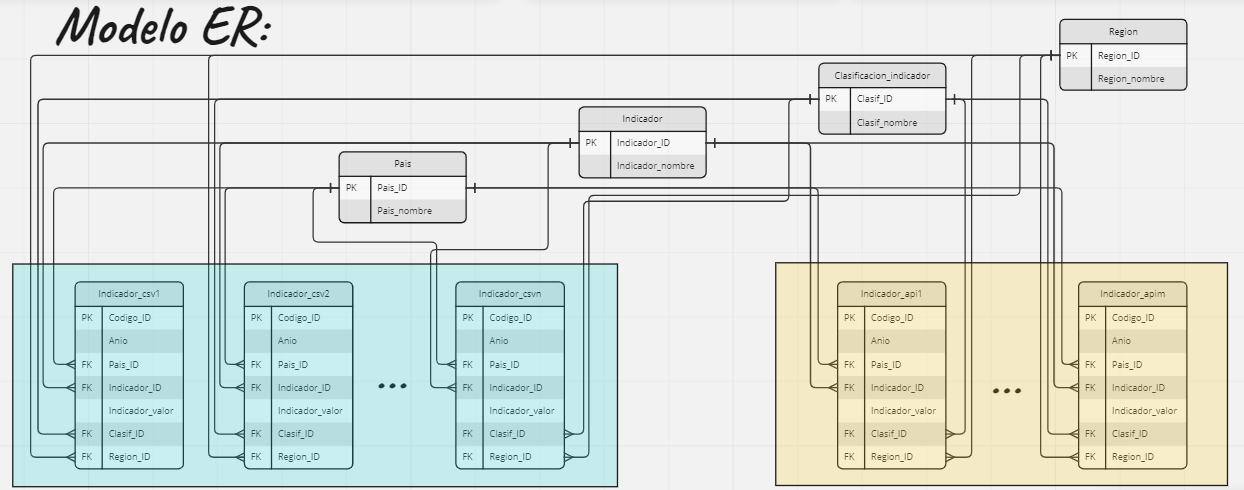
Luego, se pide respuesta a la url donde se encuentra el archivo con el indicador pedido. Se realiza la descarga. Se lee el archivo y se elimina la primera fila donde tiene anexos que no servirán al estudio.

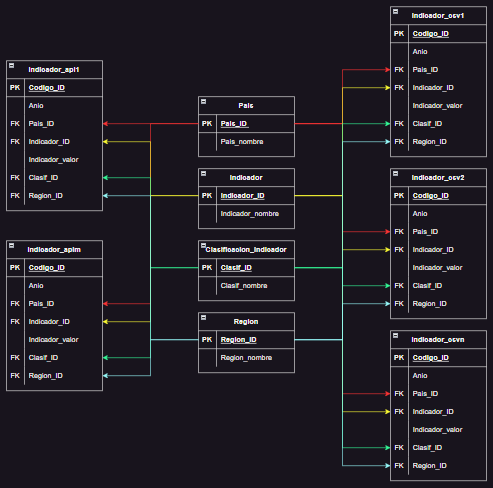
En algunos casos se procede a transponer los datos del dataframe, teniendo como columnas los años y convirtiéndolas en dos columnas siendo una el año y a otra columna el valor del indicador.



**Diseño adecuado del Modelo ER:**

Para las estructuras de almacenamiento de datos, adoptaremos un modelo entidad-relación (ER). Diseñaremos y especificaremos las tablas, relaciones y tipos de datos necesarios para representar de manera adecuada los datos relacionados con la esperanza de vida al nacer.





**Pipelines para alimentar el Data Lake:**

Desarrollaremos pipelines de extracción, transformación y carga (ETL) para alimentar el Data Lake. Estos pipelines automatizan el proceso de integración y limpieza de los datos, asegurando la calidad de los mismos.

El primer paso será establecer la conexión con nuestro Data Lake en Azure Storage Blob utilizando las bibliotecas de Azure Storage Blobs Client. Esta conexión nos permitirá acceder y cargar los datos almacenados en el Data Lake.

A continuación, importamos los datos en formato “CSV”.

Luego aplicaremos las transformaciones necesarias para adaptarlos a nuestros requisitos. Esto incluirá procesos como limpieza de datos, filtrado, agregación y normalización. Utilizamos la biblioteca pandas para llevar a cabo estas transformaciones de manera eficiente.

Posterior a las transformaciones, procederemos a cargar los datos en nuestro Data Lake utilizando las bibliotecas de Azure Storage Blobs Client. Aseguraremos una estructura adecuada de carpetas y archivos que se ajuste a nuestras necesidades y al esquema de datos definido para nuestro Data Lake.

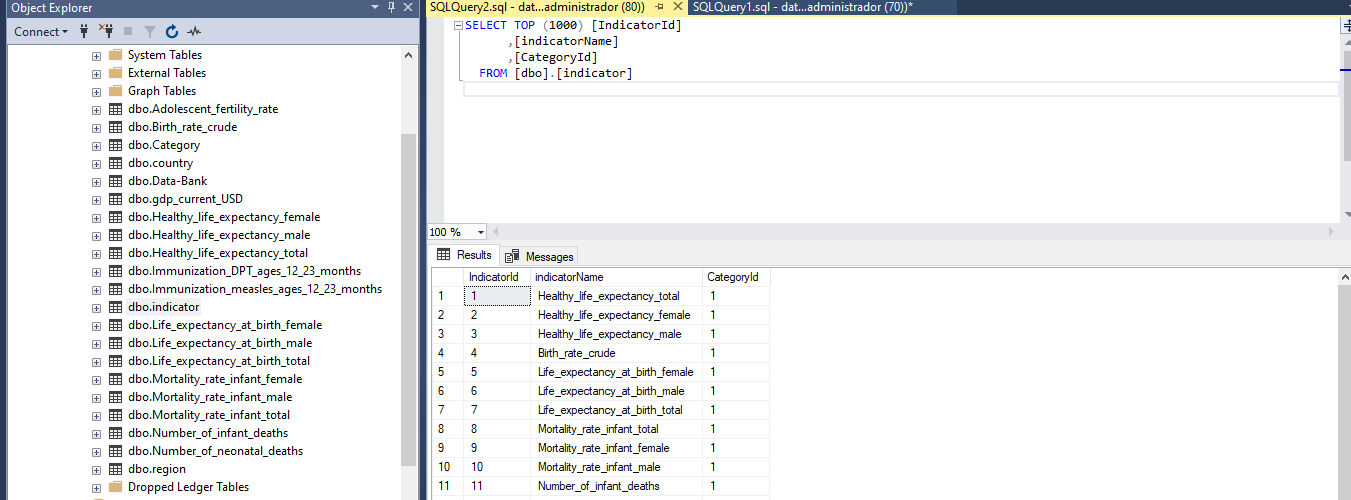
**Data Warehouse:**

Implementaremos una estructura de Data Warehouse para almacenar los datos preparados y listos para su análisis. Para automatizar el proceso de carga y creación de una tablas base de datos en Azure SQL DataBase, utilizaremos Azure Data Factory.

Azure Data Factory es una herramienta de orquestación de datos en la nube, para crear un flujo de trabajo que nos permita extraer los datos de Azure Storage Blob y cargarlos en una base de datos en Azure SQL DataBase.

Azure Data Factory nos brinda la capacidad de definir tareas de extracción, transformación y carga de datos (ETL) dentro de nuestro flujo de trabajo.

Una vez configurado el flujo de trabajo en Azure Data Factory, los datos se enviarán desde Azure Storage Blob a Azure SQL DataBase.



Posteriormente a enviar los datos a Azure SQL Database, llega el momento de mejorar la estructura de la base de datos normalizándola. Para lograr esto, creamos cuatro tablas dimensionales: "Country", "Region", "Indicadores" y "Category". Además, creamos 15 tablas de datos , una por cada indicador, con los siguientes campos: "id\_country", "id\_region", "id\_indicador", "id\_category" y el valor del indicador.

Es importante destacar que hemos implementado una clave surrogada utilizando el código del país y los dos últimos dígitos del año correspondiente. Esta clave nos proporciona un identificador único para cada registro. Las tablas dimensionales y las tablas de datos se relacionan mediante claves primarias y foráneas.

Gracias a este enfoque, hemos logrado una base de datos completamente normalizada y relacionada, lista para realizar análisis posteriormente mediante una conexión con Power BI.

**Automatización:**

La automatización de los procesos de ETL es una estrategia valiosa cuando se trata de manejar grandes volúmenes de datos que requieren actualizaciones frecuentes. Sin embargo, en el caso específico donde los datos se cargan con una frecuencia muy baja, no es necesario implementar una automatización completa por las siguientes razones:

* **Frecuencia de carga:** Si los datos se cargan solo una vez al año o con una frecuencia muy baja, el esfuerzo y el tiempo necesario para desarrollar y mantener un sistema automatizado de ETL pueden superar los beneficios obtenidos. La automatización se vuelve más relevante cuando hay actualizaciones periódicas y regulares que justifican la inversión en tiempo y recursos para configurar y mantener los procesos automatizados.
* **Complejidad de los procesos:** Los procesos de carga simples no involucran transformaciones complejas o una gran cantidad de datos, realizarlos localmente puede ser más eficiente y práctico.
* **Costo y recursos:** La implementación de una automatización completa de los procesos de ETL implica invertir en herramientas, infraestructura y recursos humanos. Si los datos se cargan con baja frecuencia, destinar recursos significativos para desarrollar y mantener la automatización puede no ser justificado desde una perspectiva de costos.

En cambio, en escenarios donde los datos se cargan con poca frecuencia, es más práctico realizar el proceso de ETL de forma manual cuando sea necesario. Esto implica dedicar el tiempo y los recursos adecuados para garantizar la calidad y la integridad de los datos durante la carga.

Es importante tener en cuenta que, aunque no se requiera una automatización completa, seguir buenas prácticas en la ejecución manual de los procesos de ETL sigue siendo esencial para garantizar la consistencia de los datos y mantener la calidad de los mismos en el Data Warehouse.

**Validación de datos:**

En el caso del proyecto, donde se utilizan múltiples fuentes de datos y se aplican transformaciones durante el proceso de extracción, transformación y carga (ETL), la validación de datos juega un papel crucial para asegurar la confiabilidad de la información analizada.

A continuación, se presentan algunos argumentos para respaldar la importancia de la validación de datos en este proyecto:

* **Consistencia de los datos:** Validar la coherencia de los valores entre diferentes fuentes y asegurar que estén dentro de los rangos esperados.
* **Integridad de los datos:** Verificar que las transformaciones aplicadas durante el proceso de ETL no introduzcan errores y que los datos resultantes sean consistentes.
* **Completitud de los datos:** Asegurarse de que no falten valores importantes y que los conjuntos de datos estén completos.
* **Detección de errores y anomalías:** Identificar valores atípicos, inconsistentes o perdidos, y aplicar reglas de negocio para garantizar la coherencia de los datos.
* **Confiabilidad del análisis:** Mejorar la calidad de los resultados y reducir el riesgo de tomar decisiones incorrectas al asegurar la precisión, completitud y consistencia de los datos utilizados.
* **Diccionario de datos:**

Elaboramos un diccionario de datos que describa el significado y la definición de cada columna en las tablas del Data Warehouse. Esto facilitará la comprensión y el análisis de los datos para todo el equipo.

| **Columna** | **Definición** |
| --- | --- |
| - `Time` | : El año o período al que corresponden los datos. |
| - `Country Name` | : El nombre del país. |
| - `Country Code` | : El código del país. |
| - `Birth rate, crude (per 1,000 people)` | : Tasa de natalidad bruta, es decir, el número de nacimientos por cada 1,000 personas. |
| - `Death rate, crude (per 1,000 people)` | : Tasa de mortalidad bruta, es decir, el número de muertes por cada 1,000 personas. |
| - `Fertility rate, total (births per woman)` | : Tasa de fertilidad total, es decir, el número promedio de hijos que una mujer tendría a lo largo de su vida. |
| - `Life expectancy at birth, female (years)` | : Esperanza de vida al nacer para mujeres en años. |
| - `Life expectancy at birth, male (years)` | : Esperanza de vida al nacer para hombres en años. |
| - `Life expectancy at birth, total (years)` | : Esperanza de vida al nacer total en años. |
| - `Mortality rate, adult, female (per 1,000 female adults)` | : Tasa de mortalidad de adultos femeninos, es decir, el número de muertes de mujeres adultas por cada 1,000 mujeres adultas. |
| - `Mortality rate, adult, male (per 1,000 male adults)` | : Tasa de mortalidad de adultos masculinos, es decir, el número de muertes de hombres adultos por cada 1,000 hombres adultos. |
| - `Mortality rate, infant (per 1,000 live births)` | : Tasa de mortalidad infantil, es decir, el número de muertes de niños menores de un año por cada 1,000 nacidos vivos. |
| - `Mortality rate, infant, female (per 1,000 live births)` | : Tasa de mortalidad infantil femenina, es decir, el número de muertes de niñas menores de un año por cada 1,000 nacidos vivos. |
| - `Mortality rate, infant, male (per 1,000 live births)` | : Tasa de mortalidad infantil masculina, es decir, el número de muertes de niños menores de un año por cada 1,000 nacidos vivos. |
| - `Mortality rate, neonatal (per 1,000 live births)` | : Tasa de mortalidad neonatal, es decir, el número de muertes de niños menores de 28 días por cada 1,000 nacidos vivos. |
| - `Population growth (annual %)` | : Tasa de crecimiento de la población anual en porcentaje. |
| - `Number of infant deaths` | : Número de muertes de niños menores de un año. |
| - `Number of neonatal deaths` | : Número de muertes de niños menores de 28 días. |
| - `Net migration` | : Migración neta, es la diferencia entre el número de inmigrantes y el número de emigrantes. |
| - `Population, female` | : Población femenina. |
| - `Population, female (% of total population)` | : Porcentaje de la población total que es femenina. |
| - `Population, male` | : Población masculina. |
| - `Population, male (% of total population)` | : Porcentaje de la población total que es masculina. |
| - `Population, total` | : Población total. |
| - `Rural population` | : Población rural. |
| - `Rural population (% of total population)` | : Porcentaje de la población total que vive en áreas rurales. |
| - `Rural population growth (annual %)` | : Tasa de crecimiento de la población rural anual en porcentaje. |
| - `Urban population` | : Población urbana. |
| - `Urban population (% of total population)` | : Porcentaje de la población total que vive en áreas urbanas. |
| - `Urban population growth (annual %)` | : Tasa de crecimiento de la población urbana anual en porcentaje. |
| - `Sex ratio at birth (male births per female births)` | : Relación de sexos al nacer, es decir, la cantidad de nacimientos masculinos por cada nacimiento femenino. |
| - `Adolescent fertility rate (births per 1,000 women ages 15-19)` | : Tasa de fertilidad adolescente, es decir, el número de nacimientos por cada 1,000 mujeres de edades entre 15 y 19 años. |
| - `GDP (current US$)` | : Producto Interno Bruto (PIB) actual en dólares estadounidenses. |
| - `GDP growth (annual %)` | : Tasa de crecimiento del Producto Interno Bruto (PIB) anual en porcentaje. |
| - `Immunization, DPT (% of children ages 12-23 months)` | : Porcentaje de niños de 12 a 23 meses vacunados contra difteria, tos ferina y tétanos. |
| - `Immunization, measles (% of children ages 12-23 months)` | : Porcentaje de niños de 12 a 23 meses vacunados contra el sarampión. |

* **Workflow de tecnologías:**

Una descripción ampliada del flujo de trabajo detallando las tecnologías utilizadas en cada etapa:

1. **Extracción de datos:**

* **Utilización de descargas de archivos CSV del Banco Mundial:** Se accede a la página del Banco Mundial y se seleccionan los indicadores relevantes. Se aplican filtros según países, años y series de datos de interés. Luego, se descargan los datos en formato CSV. Para esta etapa, se pueden utilizar tecnologías como Python (con bibliotecas como requests) para realizar solicitudes HTTP y descargar los archivos CSV.
* **Utilización de la API de la OMS:** Se utiliza la API de la OMS para obtener datos dinámicos. Se envían solicitudes HTTP a la API para obtener los indicadores deseados. La respuesta se recibe en formato CSV y se procesa para extraer los datos relevantes. Para esta etapa, se pueden utilizar tecnologías como Python (con bibliotecas como requests) para realizar solicitudes HTTP y procesar los datos CSV recibidos.

1. **Transformación de datos:**

* **Limpieza y procesamiento de datos estáticos:** Una vez que se tienen los archivos CSV descargados, se procede a leerlos y realizar las transformaciones necesarias. Esto puede incluir la eliminación de filas o columnas no relevantes, la corrección de errores, la normalización de formatos, etc. Para esta etapa, se pueden utilizar tecnologías como Python (con bibliotecas como pandas) para manipular y transformar los datos en formato CSV.
* **Procesamiento de datos dinámicos:** Los datos obtenidos de la API de la OMS, en formato CSV, también requieren transformación. Se extraen los datos relevantes del CSV y se realiza cualquier procesamiento adicional necesario, como la transposición de los datos para tener años como columnas y valores de indicadores en filas. Para esta etapa, se pueden utilizar tecnologías como Python (con bibliotecas como pandas) para manipular y transformar los datos.

1. **Almacenamiento de datos:**

* **Data Lake**: Se utiliza Azure Storage Blob como Data Lake para almacenar los datos extraídos y transformados. Se puede acceder a Azure Storage Blob mediante bibliotecas como Azure Storage Blobs Client en Python para cargar los datos en formato CSV.
* **Data Warehouse:** Se implementa un Data Warehouse utilizando Azure SQL DataBase para almacenar los datos preparados y listos para su análisis. Se utiliza Azure Data Factory para automatizar el proceso de carga y creación de tablas en Azure SQL DataBase. Azure Data Factory permite definir tareas de extracción, transformación y carga de datos (ETL) y transferir los datos desde Azure Storage Blob a Azure SQL DataBase.

1. **Análisis de datos:**

* **Exploración de datos:** Una vez que los datos se encuentran en el Data Warehouse, se pueden realizar análisis exploratorios utilizando herramientas como Azure Synapse Analytics, que proporciona capacidades de análisis y consulta avanzadas.

