

DATE  
05/24/2024

# PRUEBA TÉCNICA

Presentación Resultados

**Candidato:**  
**Diana Catalina Lopera**

**LinkedIn:** <https://www.linkedin.com/in/dianaclopera/>

**PortFolio:** <https://catalinaloperab.webflow.io/about>

**GitHub:** <https://github.com/catalinalopera>

# CONTENIDO

01

## TEST DE CONOCIMIENTOS

Descripción de Piezas claves dentro el ecosistema Azure.

02

## CASO PRÁCTICO MODELADO DE DATOS

Presentación del diagrama modelo Entidad Relación, diagrama RDV y diagrama BDV.

03

## CASO PRÁCTICO ETL CON PYTHON

Explicación de los pasos llevados a cabo para crear el caso práctico, además de la presentación del plan de tareas.

04

## DUDAS Y PREGUNTAS

# 01 TEST DE CONOCIMIENTOS

## 1.Data Lake

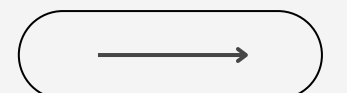
- Organización de los datos: [Azure Data Lake Storage \(ADLS\)](#), porque proporciona un almacenamiento escalable y seguro para grandes volúmenes de datos estructurados y no estructurados en su formato original.
- Gestión de datos sensibles: [Azure Data Lake Storage + Azure Key Vault: ADLS](#) porque permite la encriptación de datos en reposo y en tránsito, mientras que Azure Key Vault gestiona de forma segura las claves de encriptación y secretos, asegurando la protección de datos sensibles.
- Movimientos entre capas: [Azure Data Factory \(ADF\)](#) porque facilita la orquestación y automatización de flujos de datos entre diferentes capas del Data Lake (ingestión, almacenamiento crudo, procesamiento, presentación). Permite integrar datos de múltiples fuentes y moverlos de manera eficiente.

## 2.Data warehouse

- Definición de data warehouse y sus componentes principales: [Azure Synapse Analytics](#) ya que es una solución integrada que combina almacenamiento de datos y capacidades analíticas, permitiendo la ingesta, preparación, manejo y servicio de datos para análisis.
- Diferencias entre data warehouse, data mart y data lake: [Azure Synapse Analytics + Azure Data Lake Storage](#), teniendo en cuenta que Azure Synapse se utiliza para data warehousing y data marts, proporcionando análisis rápidos y consultas complejas, mientras que Azure Data Lake Storage es ideal para almacenar grandes volúmenes de datos en bruto y no estructurados.
- Modelos de datos dimensionales y relacionales: [Azure SQL Database](#) es adecuado para implementar tanto modelos de datos dimensionales como relacionales, proporcionando una base de datos relacional escalable y de alto rendimiento.
- ETL (Extracción, Transformación y Carga) y sus fases: [Azure Data Factory](#) es eficiente para los procesos ETL, permitiendo extraer datos de diversas fuentes, transformarlos mediante flujos de datos y cargarlos en destinos como Azure Synapse Analytics o Azure SQL Database.

### 3. Modelado de datos

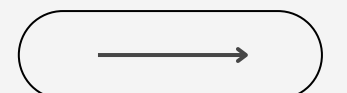
- Entidades, atributos y relaciones en el modelado de datos: [Azure SQL Database](#) es ideal para definir y gestionar entidades, atributos y relaciones en el modelado de datos, manteniendo la integridad referencial y optimizando el rendimiento de las consultas.
- Diagramas de entidad-relación (E-R) y notación Crow's Foot: [Azure Data Studio](#) es una herramienta adecuada para crear diagramas E-R y utilizar notación Crow's Foot, proporcionando capacidades de diseño y visualización de esquemas de bases de datos, existen varias extensiones que se pueden integrar.
- Normalización de datos y sus diferentes niveles: [Azure SQL Database](#) proporciona un entorno robusto para implementar la normalización de datos en diferentes niveles, asegurando la reducción de redundancias y mejora de la integridad de los datos.
- Diseño de dimensiones y hechos en un data warehouse: [Azure Synapse Analytics](#) usado para diseñar y almacenar dimensiones y hechos, optimizando el análisis y las consultas de datos.
- Componentes básicos del modelado Data Vault: [Azure Synapse Analytics](#) soporta el modelado Data Vault, permitiendo la implementación de hubs, links y satélites, y proporcionando un almacenamiento masivo y escalable para datos empresariales.



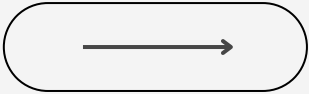
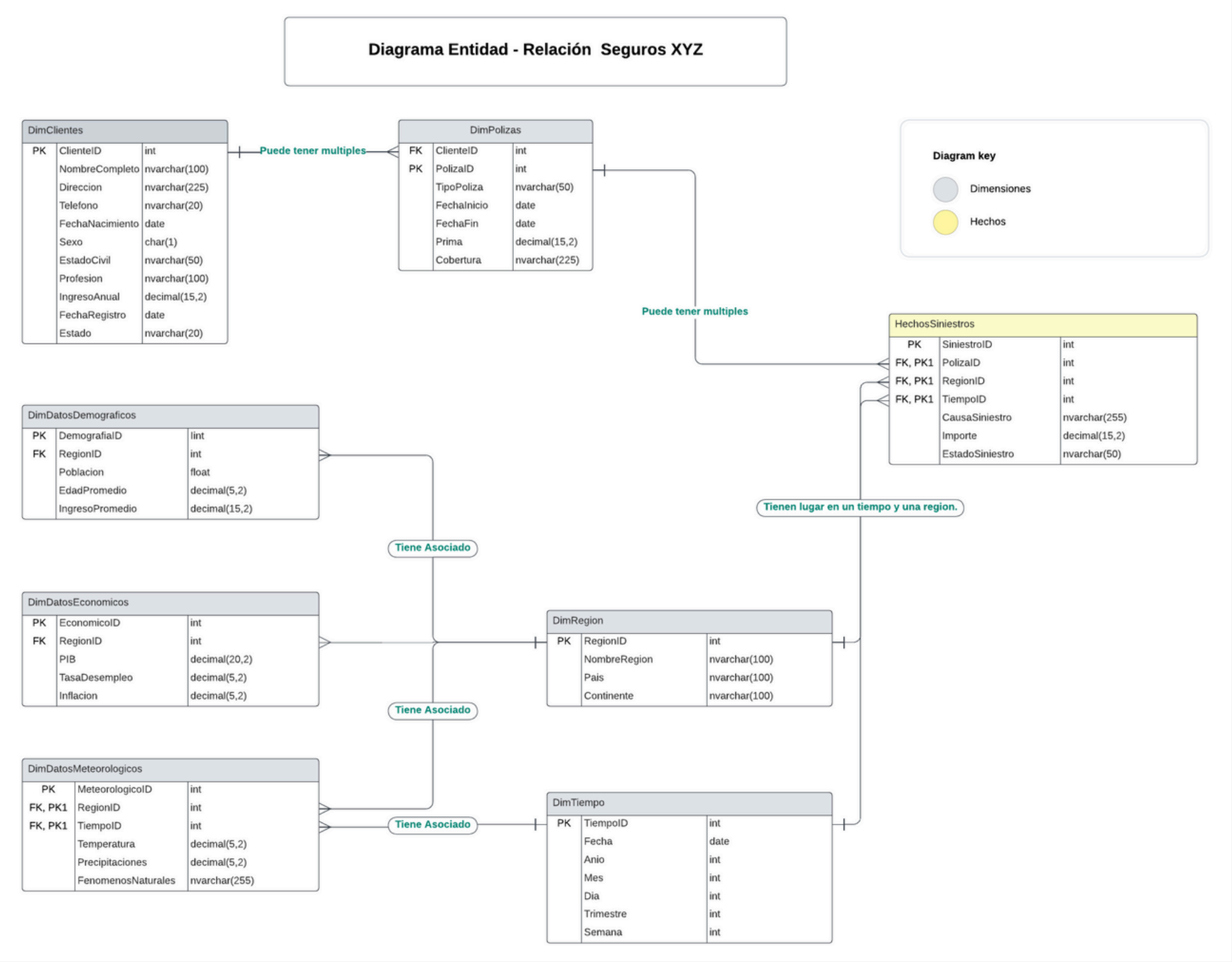
## 4.SQL - Transformaciones (sintaxis)

- Consultas básicas de SQL (SELECT, FROM, WHERE, JOIN, ORDER BY, GROUP BY).
- Agregación de datos (SUM, COUNT, AVG, MIN, MAX).
- Subconsultas y correlaciones.
- Funciones de ventana.
- Manipulación de cadenas y fechas en SQL.
- 

**Azure SQL Database o Azure Synapse Analytics:** Ambas plataformas son adecuadas para ejecutar consultas SQL básicas, ofreciendo un rendimiento robusto y capacidades avanzadas de gestión de datos.



02CASO PRÁCTICO MODELADO DE DATOS



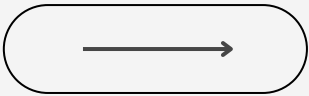
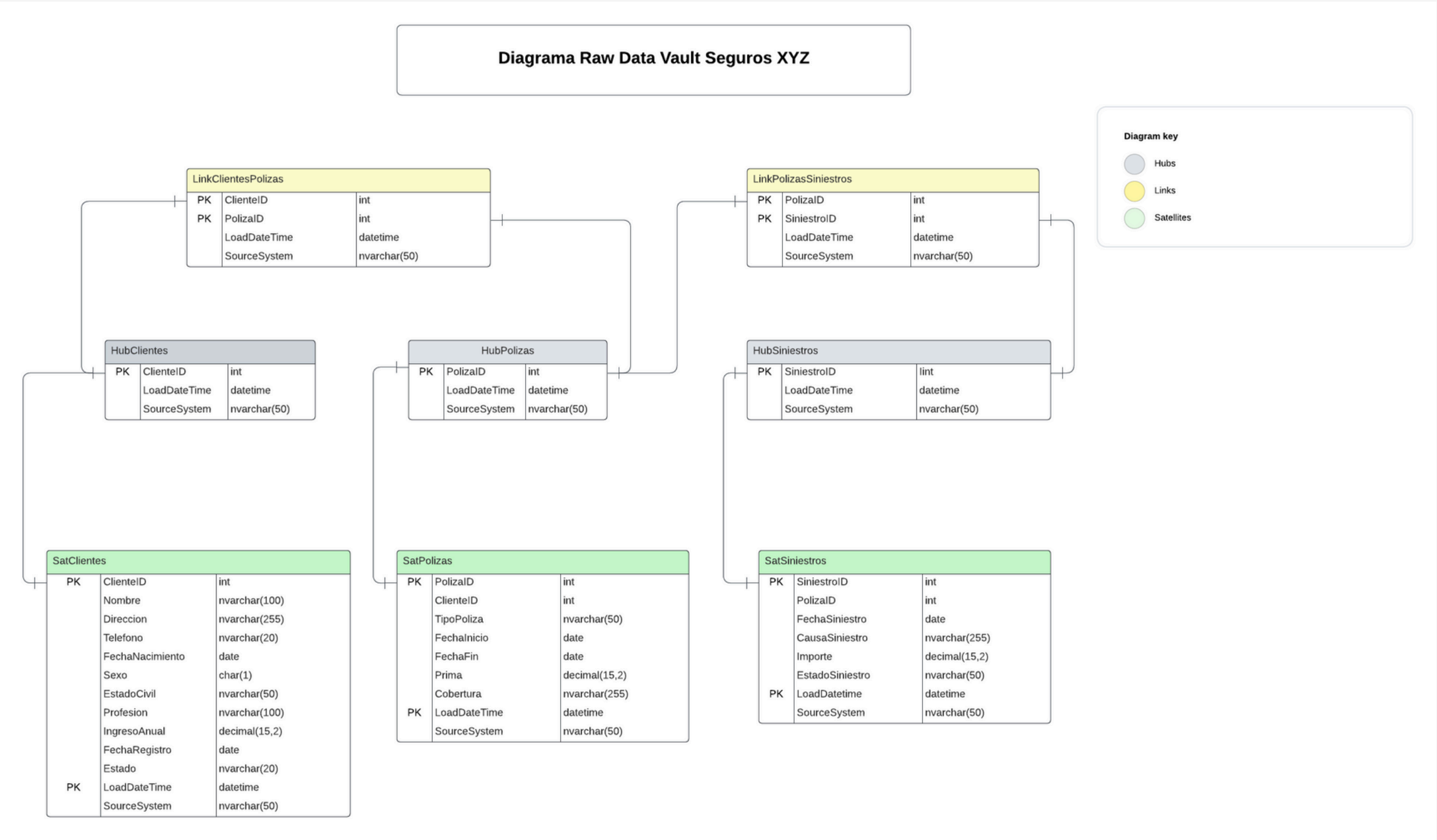
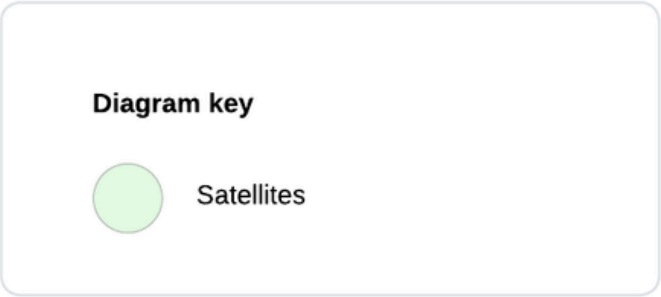


Diagrama Business Data Vault Seguros XYZ

SatClientesBusiness		
PK	ClienteID	int
	Nombre	nvarchar(100)
	Apellido	nvarchar(100)
	FechaNacimiento	date
	Estado	nvarchar(20)
PK	LoadDateTime	datetime
	SourceSystem	nvarchar(50)

SatPolizasBusiness		
PK	PolizaID	int
	ClienteID	int
	TipoPoliza	nvarchar(50)
	FechaInicio	date
	FechaFin	date
PK	LoadDateTime	datetime
	SourceSystem	nvarchar(50)

SatSiniestrosBusiness		
PK	SiniestroID	int
	PolizaID	int
	FechaSiniestro	date
	CausaSiniestro	nvarchar(255)
	Importe	decimal(15,2)
	EstadoSiniestro	nvarchar(50)
	LoadDatetime	datetime
PK	SourceSystem	nvarchar(50)



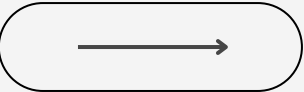
El Datawarehouse diseñado permite a la empresa de seguros XYZ realizar análisis de datos para:

- Identificar clientes con mayor riesgo de siniestralidad.
- Detectar patrones de fraude.
- Desarrollar nuevos productos y servicios de seguros.
- Mejorar la atención al cliente.

Estadísticas relacionadas con el caso práctico de modelado de datos:

Tiempo Invertido: 2 horas

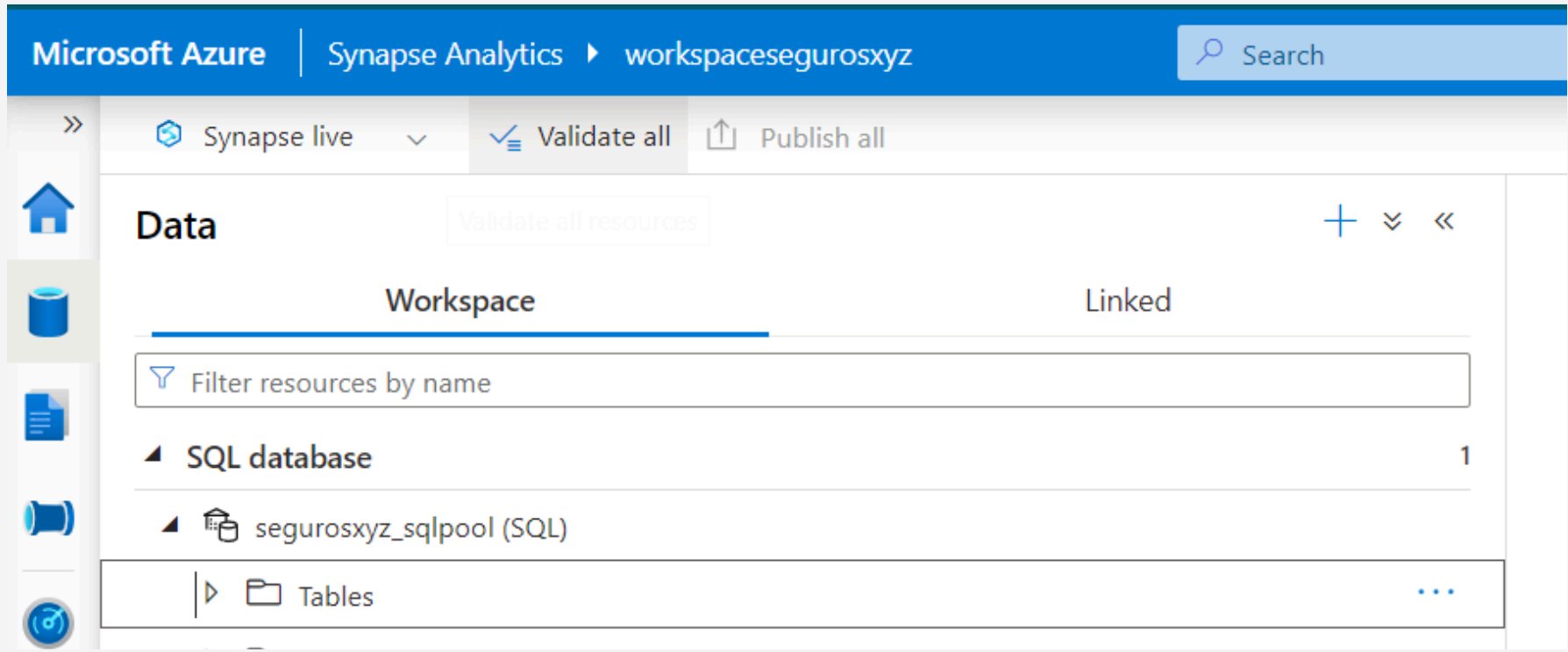
Uso de ChatGPT: Utilicé ChatGPT para aclarar algunas dudas que tenía para modelar el Raw Data Vault y Business Data Vault.



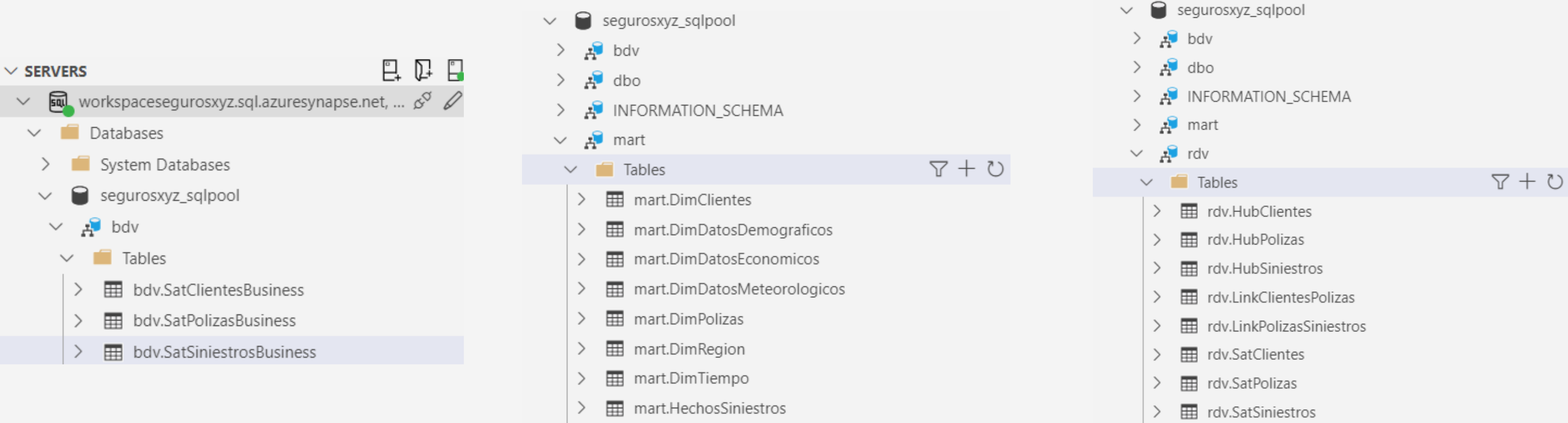


# 03 CASO PRÁCTICO ETL CON PYTHON

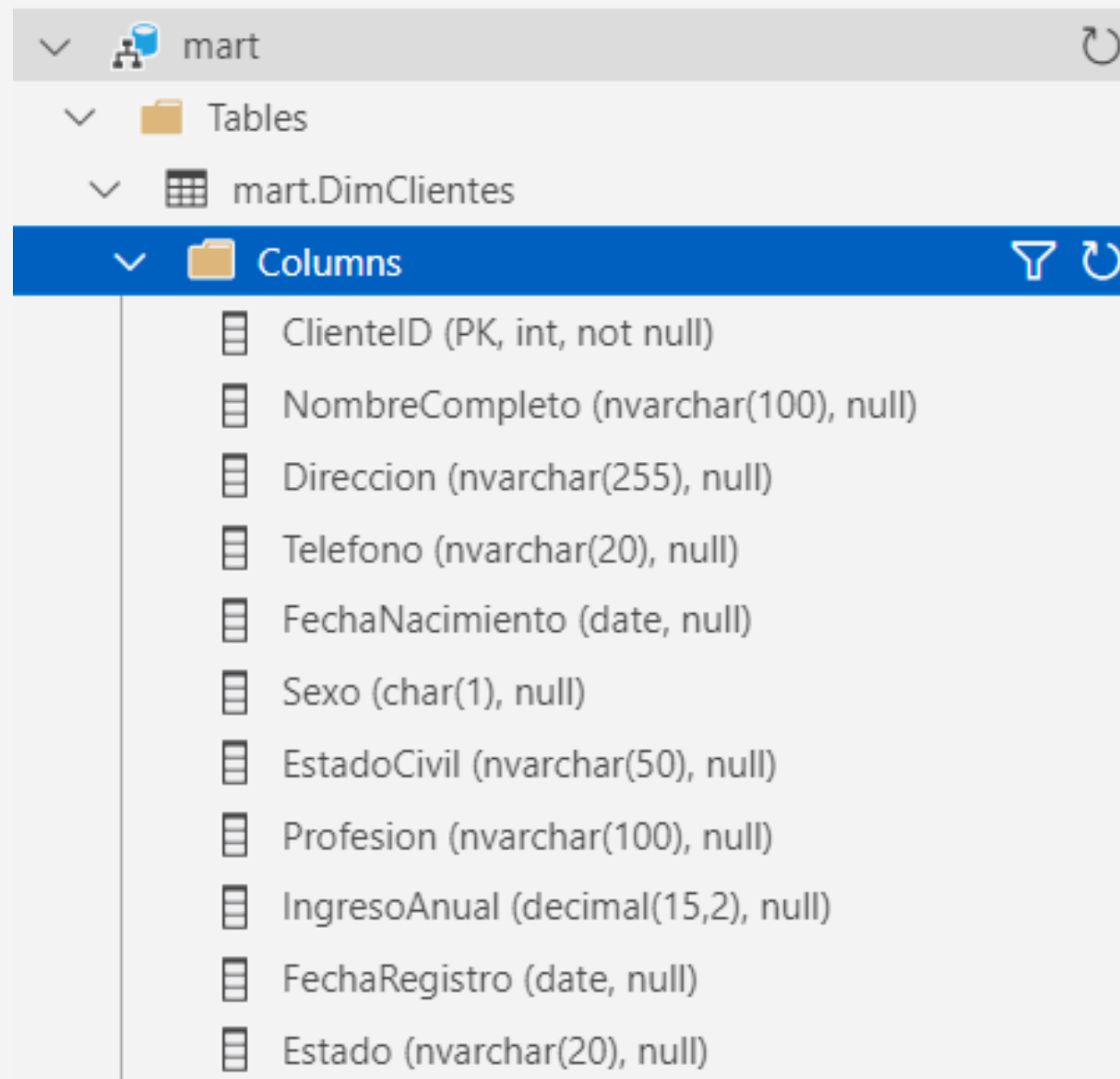
1. Se crea un pool de SQL dedicado en [Azure Synapse Analytics](#), para manipulación de los datos utilicé [Azure Data Studio](#).



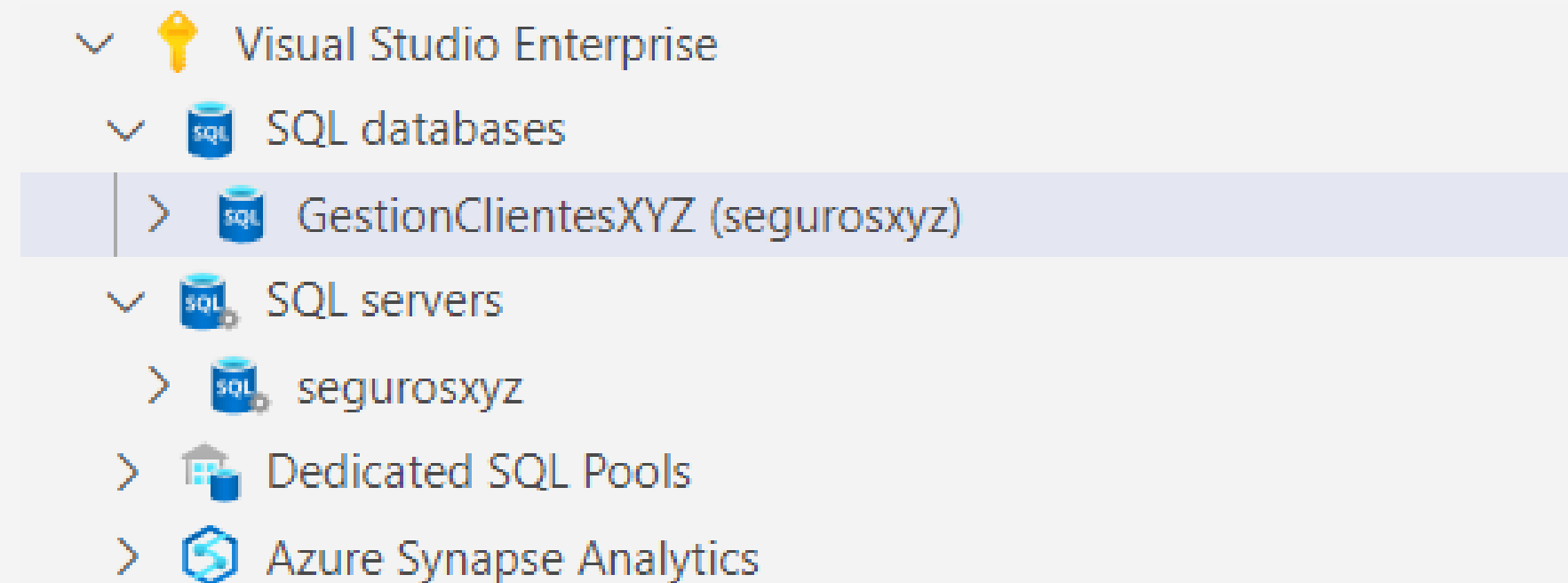
2. Se crean tres esquemas respectivamente mart, bdv y rdv para albergar los objetos creados para cumplir con el caso de negocio.



1. Se crea tabla DimClientes en el datawarehouse.



2. Se crea un servidor [Azure SQL](#) dentro del mismo resourcegroup de Synapse Analytics en Azure, allí se crea una base de datos que alberga el sistema de clientes de la compañía XYZ seguros.



3. Se crea tabla Clientes y se generan 500 registros aleatorios, donde en algunos de ellos se presentan inconsistencias. Por medio de un script de python se resumen las inconsistencias encontradas.

Welcome

SQLQuery\_1 - (54) s...er2024)

SQLQuery\_2 - worksp...inuser)

Run

Cancel

Disconnect

Change

Database: GestionClientesXYZ

Estimated Plan

Enable Actual Plan

Parse

Enable SQLCMD

To Notebook

1

select \* from clientes

Results

Messages

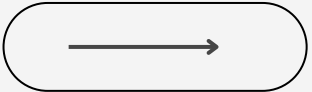
	ClienteID	Nombre	Apellido	Direccion	Telefono	FechaNacimiento	Sexo	EstadoCivil	Profesion	IngresoAnual	FechaRegistro	Estado
30	30	NOMBRE30	APELLIDO30	Direccion30	invalidpho...	2024-04-23	M	Soltero	NULL	30000.00	2024-04-23	Activo
31	31	nombre31	apellido31	Direccion31	0000000031	2024-04-22	F	Casado	Profesion31	31000.00	2024-04-22	Inactivo
32	32	NOMBRE32	APELLIDO32	NULL	0000000032	2024-04-21	M	Divorciado	Profesion32	32000.00	2024-04-21	Activo
33	33	nombre33	apellido33	Direccion33	0000000033	2024-04-20	F	Soltero	Profesion33	33000.00	2024-04-20	Inactivo
34	34	NOMBRE34	APELLIDO34	Direccion34	0000000034	2024-04-19	M	Casado	Profesion34	34000.00	2024-04-19	Activo
35	35	nombre35	apellido35	Direccion35	0000000035	2024-04-18	F	Divorciado	Profesion35	35000.00	2024-04-18	Inactivo
36	36	NOMBRE36	APELLIDO36	Direccion36	0000000036	2024-04-17	M	Soltero	Profesion36	36000.00	2024-04-17	Activo
37	37	nombre37	apellido37	Direccion37	0000000037	2024-04-16	F	Casado	Profesion37	37000.00	2024-04-16	Inactivo
38	38	NULL	APELLIDO38	Direccion38	0000000038	2024-04-15	M	Soltero	Profesion38	38000.00	2024-04-15	Activo
39	39	nombre39	apellido39	Direccion39	0000000039	2024-04-14	F	Soltero	Profesion39	39000.00	2024-04-14	Inactivo
40	40	NOMBRE40	APELLIDO40	NULL	0000000040	2024-04-13	M	NULL	NULL	-40000.00	2024-04-13	Activo
41	41	nombre41	apellido41	Direccion41	0000000041	2024-04-12	F	Divorciado	Profesion41	41000.00	2024-04-12	Inactivo
42	42	NOMBRE42	APELLIDO42	Direccion42	0000000042	2024-04-11	M	Soltero	Profesion42	42000.00	2024-04-11	Activo
43	43	nombre43	apellido43	Direccion43	0000000043	2024-04-10	F	Casado	Profesion43	43000.00	2024-04-10	Inactivo
44	44	NOMBRE44	APELLIDO44	Direccion44	0000000044	2024-04-09	M	Divorciado	Profesion44	44000.00	2024-04-09	Activo
45	45	nombre45	apellido45	Direccion45	0000000045	2024-04-08	F	Soltero	Profesion45	45000.00	2024-04-08	Inactivo
46	46	NOMBRE46	APELLIDO46	Direccion46	0000000046	2024-04-07	M	Casado	Profesion46	46000.00	2024-04-07	Activo
47	47	nombre47	apellido47	Direccion47	0000000047	2024-04-06	F	Divorciado	Profesion47	47000.00	2024-04-06	Inactivo
48	48	NOMBRE48	APELLIDO48	NULL	0000000048	2024-04-05	M	Soltero	Profesion48	48000.00	2024-04-05	Activo
49	49	nombre49	apellido49	Direccion49	0000000049	2024-04-04	F	Casado	Profesion49	49000.00	2024-04-04	Inactivo
50	50	NOMBRE50	APELLIDO50	Direccion50	no-valid-p...	2024-04-03	M	Divorciado	NULL	NULL	2024-04-03	Activo
51	51	nombre51	apellido51	Direccion51	0000000051	2024-04-02	F	Soltero	Profesion51	51000.00	2024-04-02	Inactivo
52	52	NOMBRE52	APELLIDO52	Direccion52	0000000052	2024-04-01	M	Casado	Profesion52	52000.00	2024-04-01	Activo

```
... Inconsistencias encontradas:
Valores nulos:
ClienteID          0
Nombre             45
Apellido           0
Direccion           57
Telefono           0
FechaNacimiento    0
Sexo                0
EstadoCivil        23
Profesion           43
IngresoAnual        7
FechaRegistro      0
Estado             0
dtype: int64

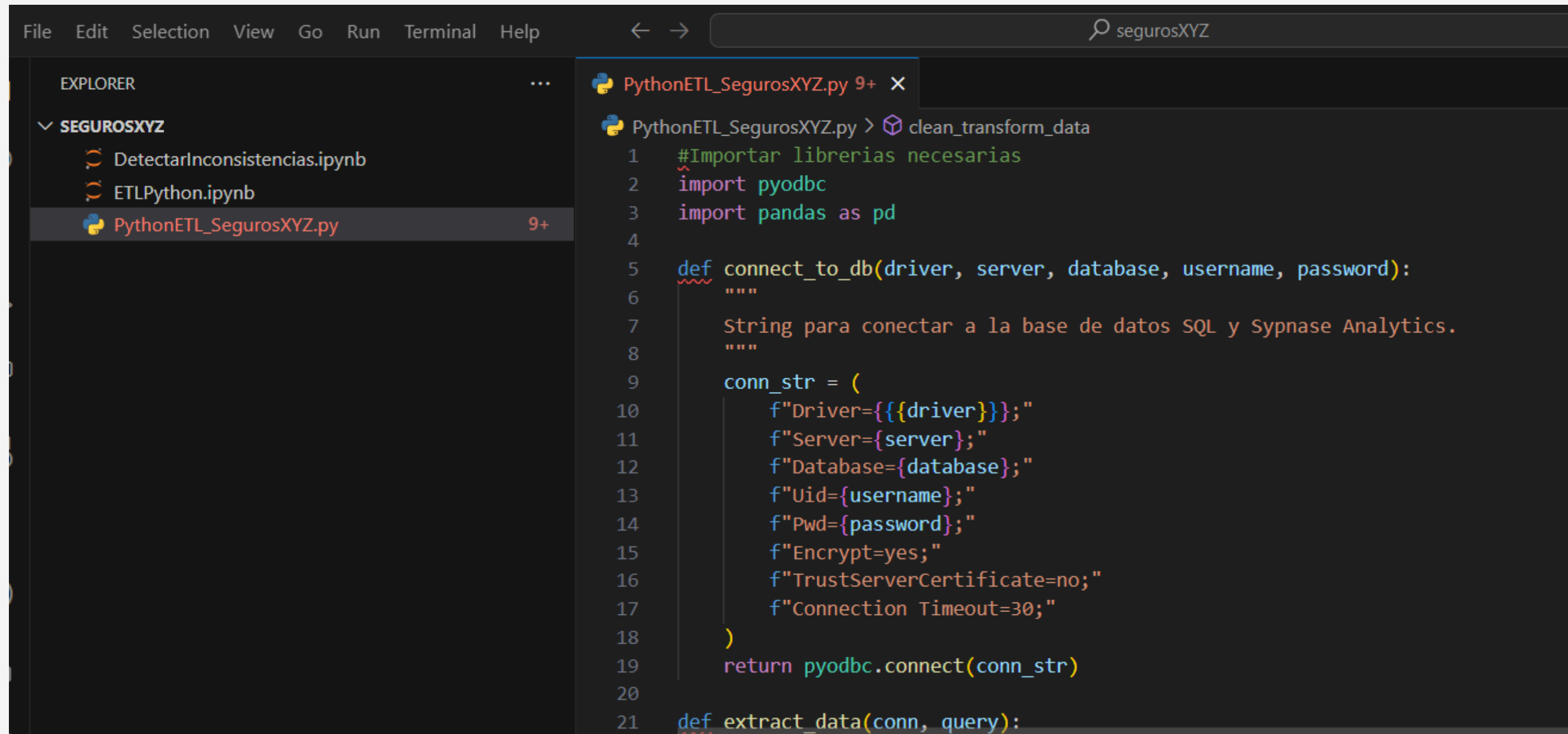
Filas con teléfonos inválidos: 29

Filas con ingresos anuales negativos: 11

Filas duplicadas basadas en ClienteID: 0
```



4. Se crea un ambiente virtual haciendo uso de [Anacanda y visual studio code](#), se instalan las librerías requeridas para la creación del script en python versión 3.8



```
File Edit Selection View Go Run Terminal Help
segurosXYZ

EXPLORER
SEGUROSXYZ
  DetectarInconsistencias.ipynb
  ETLPython.ipynb
  PythonETL_SegurosXYZ.py 9+

PythonETL_SegurosXYZ.py 9+ X
PythonETL_SegurosXYZ.py > clean_transform_data
1  #Importar librerías necesarias
2  import pyodbc
3  import pandas as pd
4
5  def connect_to_db(driver, server, database, username, password):
6      """
7      String para conectar a la base de datos SQL y Synapse Analytics.
8      """
9      conn_str = (
10         f"Driver={{{{driver}}}};"
11         f"Server={server};"
12         f"Database={database};"
13         f"Uid={username};"
14         f"Pwd={password};"
15         f"Encrypt=yes;"
16         f"TrustServerCertificate=no;"
17         f"Connection Timeout=30;"
18     )
19     return pyodbc.connect(conn_str)
20
21 def extract_data(conn, query):
```

5. Se crea un script .py con todos los pasos para realizar el proceso ETL solicitado, dentro del archivo se documenta claramente y de manera precisa cada uno de los pasos.

```
PythonETL_SegurosXYZ.py 9+ X
PythonETL_SegurosXYZ.py > main
1  #Importar librerias necesarias
2  import pyodbc
3  import pandas as pd
4
5  def connect_to_db(driver, server, database, username, password):
6      """
7      String para conectar a la base de datos SQL y Synapse Analytics.
8      """
9      conn_str = (
10         f"Driver={{{{driver}}}};"
11         f"Server={server};"
12         f"Database={database};"
13         f"Uid={username};"
14         f"Pwd={password};"
15         f"Encrypt=yes;"
16         f"TrustServerCertificate=no;"
17         f"Connection Timeout=30;"
18     )
19     return pyodbc.connect(conn_str)
20
21 def extract_data(conn, query):
22     """
23     Extraer los datos de la base de datos SQL - Gestion de Clientes Seguros XYZ
24     """
25     return pd.read_sql(query, conn)
26
27 def validate_phone(phone):
28     """
29     Validar que el número telefónico sea valido
30     """
31     if phone.isdigit() and len(phone) == 10:
32         return phone
33     else:
34         return "DatoIncorrecto"
35
```

```
def clean_transform_data(df):
    """
    Función para Transformar y Limpiar los datos de la fuente de origen para cargarlo al DataWareHouse
    """

    # Borrar Columnas cuyo nombre viene NULL
    df = df[df['Nombre'].notnull()]

    # Verificar los datos que llegan de otras columnas NULL y asignar un nuevo valor como se detalla acontinuación.
    df['Apellido'].fillna('SinDato', inplace=True)
    df['Direccion'].fillna('SinDato', inplace=True)
    df['Telefono'] = df['Telefono'].apply(validate_phone)
    df['EstadoCivil'].fillna('SinDato', inplace=True)
    df['Profesion'].fillna('SinDato', inplace=True)
    df['IngresoAnual'].fillna(0, inplace=True)
    df['IngresoAnual'] = pd.to_numeric(df['IngresoAnual'], errors='coerce')
    df['IngresoAnual'] = df['IngresoAnual'].apply(lambda x: x if x >= 0 else 0) # Reemplazar los valores Negativos con Cero
    df['Estado'].fillna('SinDato', inplace=True)

    # Normalizar los textos en la base de datos
    df['Nombre'] = df['Nombre'].str.strip().str.title()
    df['Apellido'] = df['Apellido'].str.strip().str.title()
    df['Direccion'] = df['Direccion'].str.strip().str.title()
    df['EstadoCivil'] = df['EstadoCivil'].str.strip().str.capitalize()
    df['Profesion'] = df['Profesion'].str.strip().str.title()
    df['Estado'] = df['Estado'].str.strip().str.capitalize()

    # Crear una nueva columna que contenga la combinación del Nombre + Apellido
    df['NombreCompleto'] = df['Nombre'] + ' ' + df['Apellido']

    # En el caso de que se presenten duplicados con la identificación del cliente, se proceden a borrar
    df.drop_duplicates(subset='ClienteID', keep='last', inplace=True)

    return df
```



## 6. Continuación del Script.

```
def load_data(df, conn):
    """
    Función para cargar la data transformada en la tabla del datawarehouse
    """
    cursor = conn.cursor()
    for index, row in df.iterrows():
        cursor.execute("""
        INSERT INTO mart.DimClientes (ClienteID, NombreCompleto, Direccion, Telefono, FechaNacimiento, Sexo, EstadoCivil, Profesion, IngresoAnual, FechaRegistro, Estado)
        VALUES (?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?)""",
        row['ClienteID'], row['NombreCompleto'], row['Direccion'], row['Telefono'], row['FechaNacimiento'], row['Sexo'], row['EstadoCivil'], row['Profesion'], row['IngresoAnual'], row['FechaRegistro'], row['Estado'])
    conn.commit()
```

```
def main():
    # Datos de conexión a la base de datos del sistema de gestión de clientes
    client_db_driver = 'ODBC Driver 18 for SQL Server'
    client_db_server = 'tcp:segurosxyz.database.windows.net,1433'
    client_db_database = 'GestionClientesXYZ'
    client_db_username = 'sqladminuser2024'
    client_db_password = 'Segurosxyz.2024'

    # Datos de conexión al data warehouse en Azure Synapse Analytics
    dw_driver = 'ODBC Driver 18 for SQL Server'
    dw_server = 'tcp:workspacesegurosxyz.sql.azuresynapse.net,1433'
    dw_database = 'segurosxyz_sqlpool'
    dw_username = 'sqladminuser'
    dw_password = 'Segurosxyz.2024'

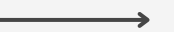
    # Conectar a la base de datos del sistema de gestión de clientes
    conn_clientes = connect_to_db(client_db_driver, client_db_server, client_db_database, client_db_username, client_db_password)
    query_clientes = "SELECT * FROM Clientes"
    df_clientes = extract_data(conn_clientes, query_clientes)

    # Limpiar y transformar los datos
    df_transformed = clean_transform_data(df_clientes)

    # Conectar al data warehouse en Azure Synapse Analytics
    conn_synapse = connect_to_db(dw_driver, dw_server, dw_database, dw_username, dw_password)
    load_data(df_transformed, conn_synapse)

    # Mensaje de confirmación
    print("Los datos se han copiado exitosamente.")

if __name__ == "__main__":
    main()
```



7. Al ejecutarse el Script la tabla DImClientes creada en el datawarehouse en Azure Sypnase Analytics es poblada con un total de 455 registros , y luego de la transaformación de los datos estas son las estadísticas:

WelcomeSQLQuery\_1 - (54) s...er2024SQLQuery\_2 - worksp...inuser

RunCancelDisconnectChange

Database:segurosxyz\_sqlpool

Estimated PlanEnable Actual PlanParseEnable SQLCMDTo Notebook

1select \* from mart.DimClientes

2

3

ResultsMessages

	ClienteID	NombreCompleto	Direccion	Telefono	FechaNacimiento	Sexo	EstadoCivil	Profesion	IngresoAnual	FechaRegistro	Estado
428	476	Nombre476 Apellido476	Direccion476	0000000476	2023-02-02	M	Divorciado	Profesion476	476000.00	2023-02-02	Activo
429	424	Nombre424 Apellido424	Sindato	0000000424	2023-03-26	M	Casado	Profesion424	424000.00	2023-03-26	Activo
430	322	Nombre322 Apellido322	Direccion322	0000000322	2023-07-06	M	Casado	Profesion322	322000.00	2023-07-06	Activo
431	495	Nombre495 Apellido495	Direccion495	0000000495	2023-01-14	F	Soltero	Profesion495	495000.00	2023-01-14	Inactivo
432	358	Nombre358 Apellido358	Direccion358	0000000358	2023-05-31	M	Casado	Profesion358	358000.00	2023-05-31	Activo
433	376	Nombre376 Apellido376	Sindato	0000000376	2023-05-13	M	Casado	Profesion376	376000.00	2023-05-13	Activo
434	463	Nombre463 Apellido463	Direccion463	0000000463	2023-02-15	F	Casado	Profesion463	463000.00	2023-02-15	Inactivo
435	407	Nombre407 Apellido407	Direccion407	0000000407	2023-04-12	F	Divorciado	Profesion407	407000.00	2023-04-12	Inactivo
436	400	Nombre400 Apellido400	Sindato	DatoIncorrecto	2023-04-19	M	Sindato	Sindato	0.00	2023-04-19	Activo
437	488	Nombre488 Apellido488	Sindato	0000000488	2023-01-21	M	Divorciado	Profesion488	488000.00	2023-01-21	Activo
438	489	Nombre489 Apellido489	Direccion489	0000000489	2023-01-20	F	Soltero	Profesion489	489000.00	2023-01-20	Inactivo
439	391	Nombre391 Apellido391	Direccion391	0000000391	2023-04-28	F	Casado	Profesion391	391000.00	2023-04-28	Inactivo
440	368	Nombre368 Apellido368	Sindato	0000000368	2023-05-21	M	Divorciado	Profesion368	368000.00	2023-05-21	Activo
441	388	Nombre388 Apellido388	Direccion388	0000000388	2023-05-01	M	Casado	Profesion388	388000.00	2023-05-01	Activo
442	486	Nombre486 Apellido486	Direccion486	0000000486	2023-01-23	M	Soltero	Profesion486	486000.00	2023-01-23	Activo
443	416	Nombre416 Apellido416	Sindato	0000000416	2023-04-03	M	Divorciado	Profesion416	416000.00	2023-04-03	Activo
444	500	Nombre500 Apellido500	Direccion500	DatoIncorrecto	2023-01-09	M	Sindato	Sindato	0.00	2023-01-09	Activo
445	409	Nombre409 Apellido409	Direccion409	0000000409	2023-04-10	F	Casado	Profesion409	409000.00	2023-04-10	Inactivo
446	429	Nombre429 Apellido429	Direccion429	0000000429	2023-03-21	F	Soltero	Profesion429	429000.00	2023-03-21	Inactivo
447	425	Nombre425 Apellido425	Direccion425	DatoIncorrecto	2023-03-25	F	Divorciado	Profesion425	425000.00	2023-03-25	Inactivo
448	419	Nombre419 Apellido419	Direccion419	0000000419	2023-03-31	F	Divorciado	Profesion419	419000.00	2023-03-31	Inactivo
449	494	Nombre494 Apellido494	Direccion494	0000000494	2023-01-15	M	Divorciado	Profesion494	494000.00	2023-01-15	Activo
450	443	Nombre443 Apellido443	Direccion443	0000000443	2023-03-07	F	Divorciado	Profesion443	443000.00	2023-03-07	Inactivo
451	437	Nombre437 Apellido437	Direccion437	0000000437	2023-03-13	F	Divorciado	Profesion437	437000.00	2023-03-13	Inactivo
452	447	Nombre447 Apellido447	Direccion447	0000000447	2023-03-03	F	Soltero	Profesion447	447000.00	2023-03-03	Inactivo
453	465	Nombre465 Apellido465	Direccion465	0000000465	2023-02-13	F	Soltero	Profesion465	465000.00	2023-02-13	Inactivo
454	480	Nombre480 Apellido480	Sindato	DatoIncorrecto	2023-01-29	M	Sindato	Sindato	0.00	2023-01-29	Activo
455	499	Nombre499 Apellido499	Direccion499	0000000499	2023-01-10	F	Casado	Profesion499	499000.00	2023-01-10	Inactivo

Inconsistencias encontradas:

Valores nulos:

ClienteID0

NombreCompleto0

Direccion0

Telefono0

FechaNacimiento0

Sexo0

EstadoCivil0

Profesion0

IngresoAnual0

FechaRegistro0

Estado0

dtype: int64

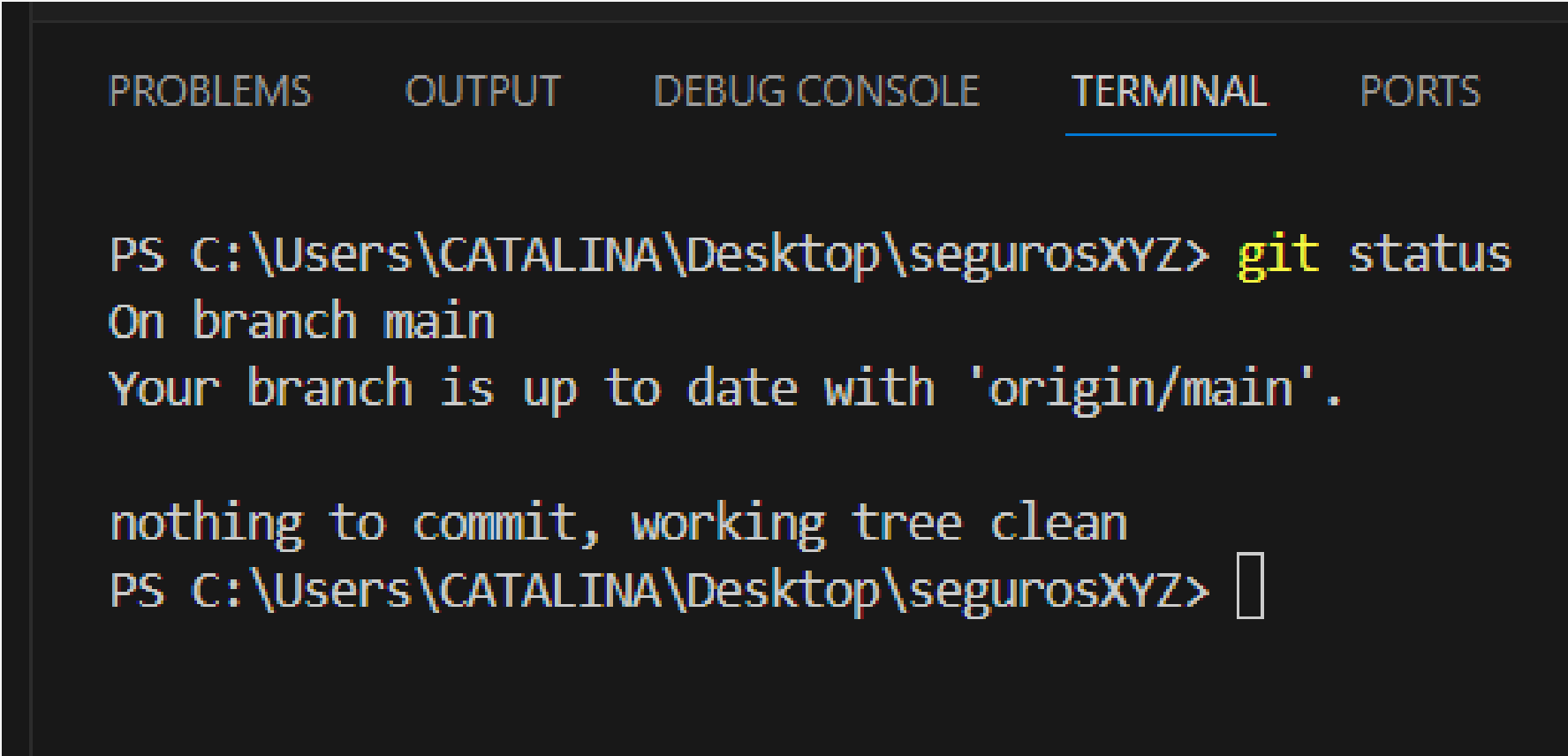
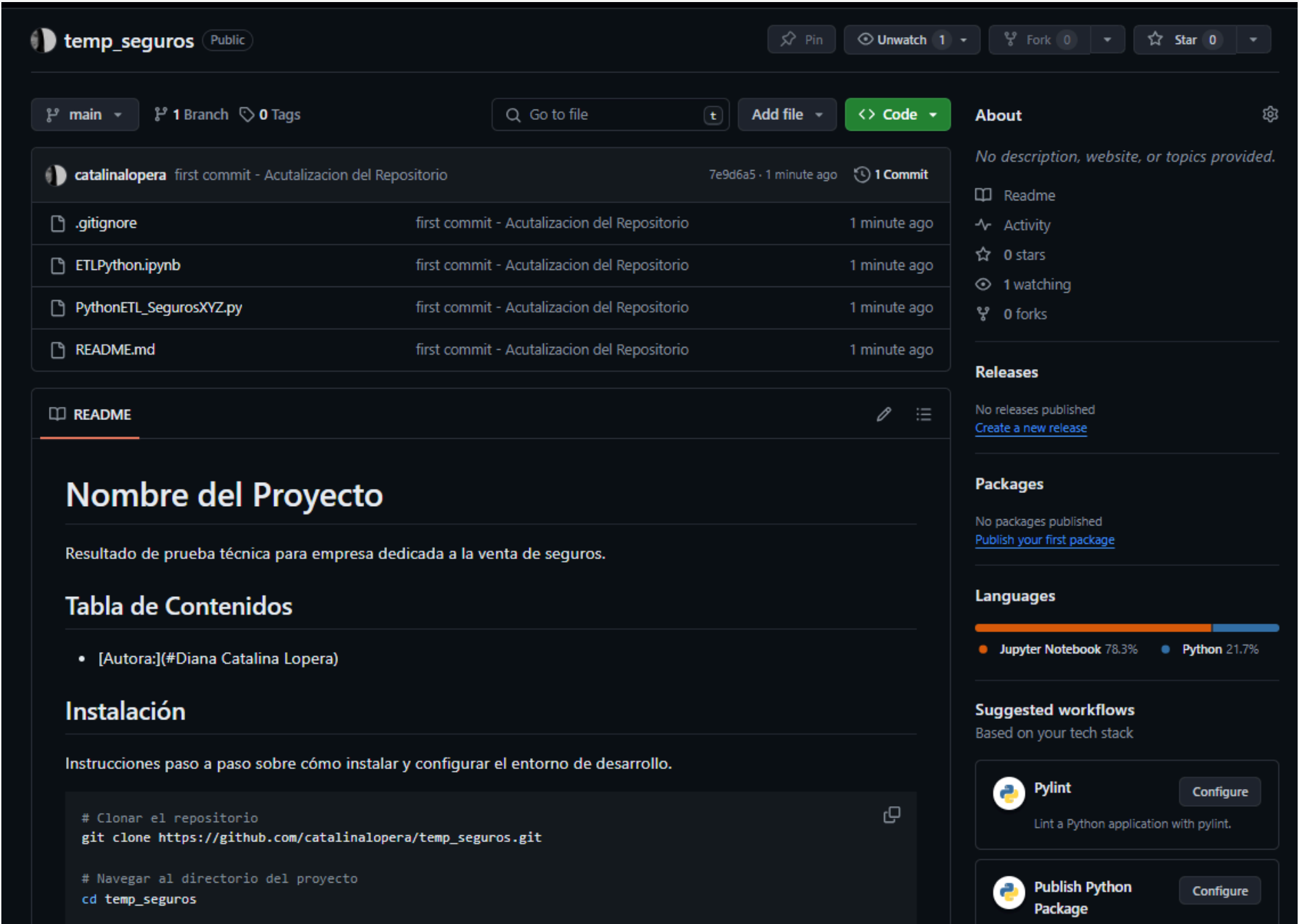
Filas con teléfonos marcados como Dato Incorrecto: 29

Filas con ingresos anuales negativos: 0

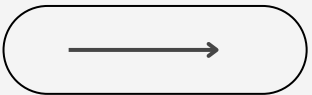
Filas duplicadas basadas en ClienteID: 0



8. Finalmente el script generado es subido a un repositorio de GitHub donde ademas se suben otros archivos utilizados durante la prueba.



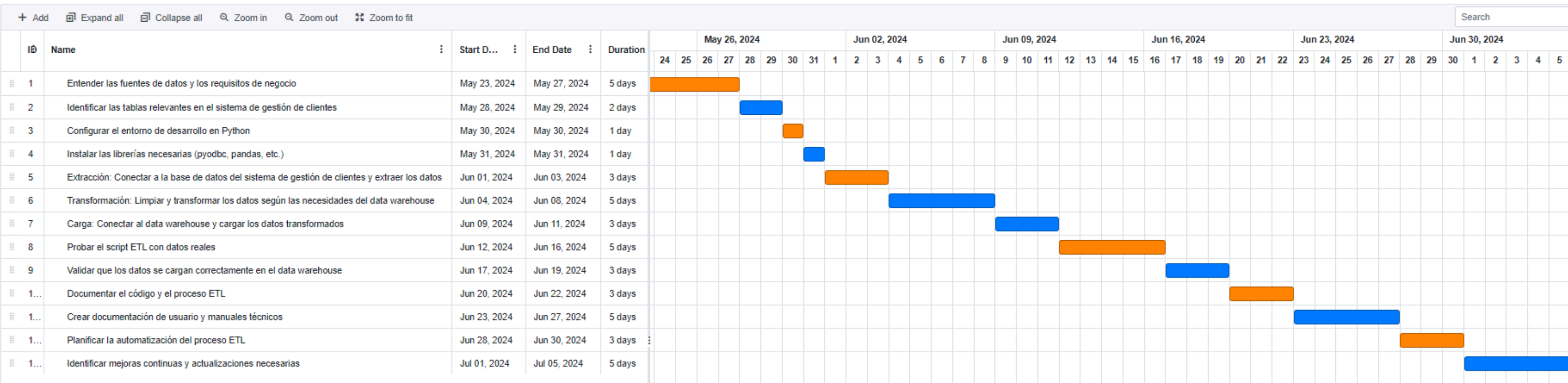
**Repositorio de GitHub:** [https://github.com/catalinalopera/temp\\_seguros](https://github.com/catalinalopera/temp_seguros)





9. Construcción del plan de tareas e hitos asociados al desarrollo y evoluciones futuras.

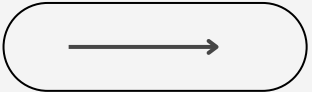
Las tareas pintadas con color naranja hacen parte de la ruta crítica del proyecto de modernización de la infraestructura de datos para mejorar la toma de decisiones y ofrecer un mejor servicio al cliente.



Estadísticas relacionadas al caso práctico ETL con Python:

**Tiempo Invertido:** 3 horas

**Uso de ChatGPT:** Utilicé ChatGPT para resolver varios inconvenientes que se me presentaron a la hora de crear las conexiones desde Data Studio hacia Azure Synapse Analytics, así como también para resolver errores presentados durante la transformación de los datos en el script de python.



04

# DUDAS Y PREGUNTAS