

Datawarehouse y Minería de Datos DMD941 G01T

Catedrático

Ing. Karen Medrano

GRUPO 1

Integrantes :

| | |
|-------------------------------|----------|
| Jorge Marvin Peña Roque | PR243380 |
| Héctor José Márquez | MC233291 |
| Erick Roberto Zavaleta Rivera | ZR171491 |
| Josseline Beatriz Pérez M. | PM171434 |
| Ever Félix De León Medoza | DM191820 |
| Erick Iván Peña Rivas | PR170059 |

San Salvador 14 de Septiembre de 2024



1. El club de deportivo “Cebollitas”, necesita un proceso ETL (Extract, Transform, Load) para manejar un archivo CSV que contiene datos detallados de jugadores de fútbol. Los datos deben ser procesados, almacenados en una base de datos SQL Server y analizados para extraer la siguiente información: correlación entre rating y age de los jugadores, además de generar una tabla para conocer las estadísticas de Height, weight y rating por age.

Programas por utilizar:

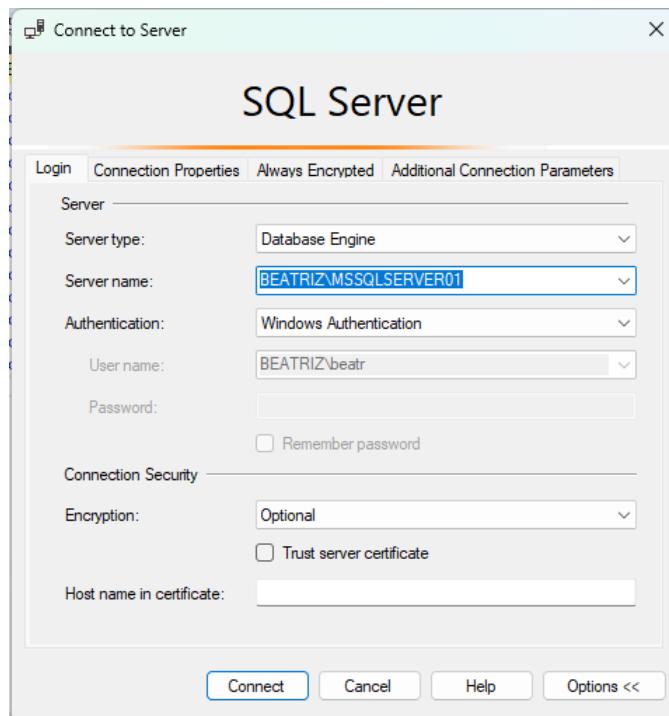
- Visual estudio 2019
- Sql Server Mananger Estudio 2012

Entrar a SQL Server Management Studio

- a. Hacer clic en el botón Inicio
- b. Hacer clic en la opción Todos los programas y hacer clic en Microsoft SQL Server 2012
- c. Hacer clic en SQL Server Management Studio

Para conectarse con el servidor de base de datos elija los siguientes parámetros de autenticación:

- Tipo de servidor: Database Engine
- Nombre del servidor: Colocar el nombre del servidor local, en nuestro caso es: BEATRIZ\MSSQLSERVER01
- Autenticación: SQL Server Authentication



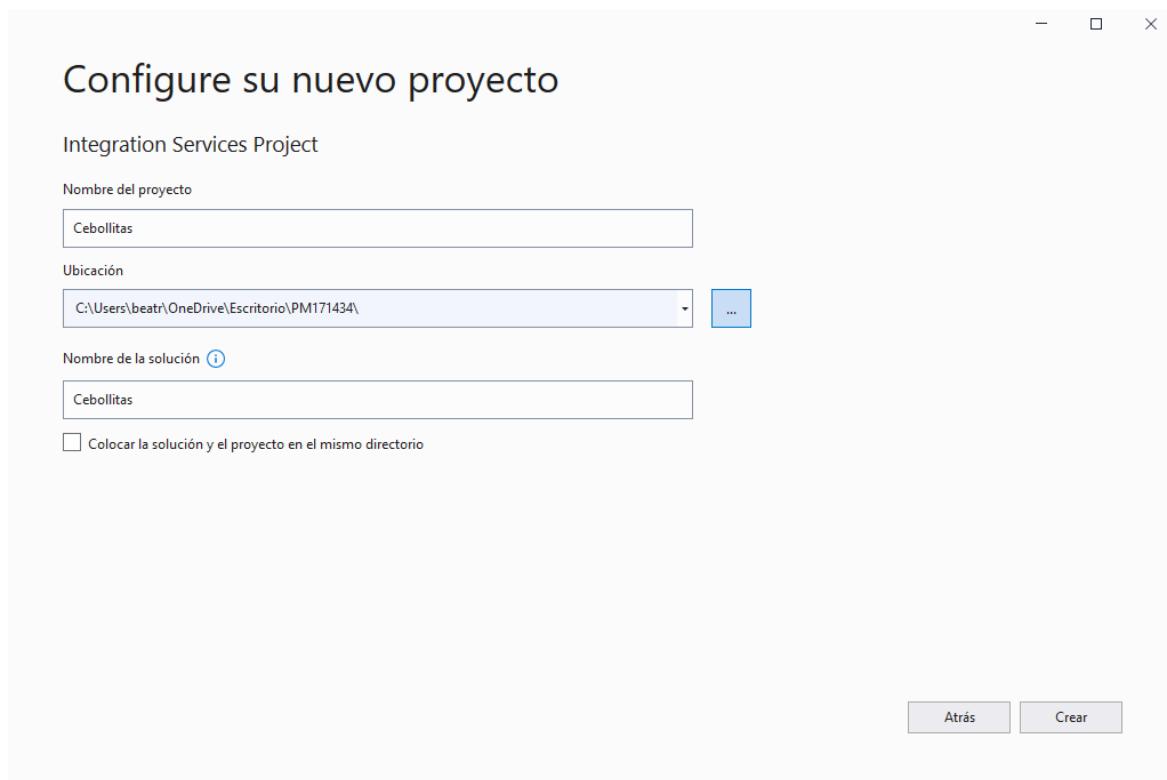
Crear la base de datos CebollitasDB

```
CREATE DATABASE CebollitasDB;
GO
USE CebollitasDB;

CREATE TABLE Players (
    PlayerID INT PRIMARY KEY,
    Name NVARCHAR(100),
    Age INT,
    Height DECIMAL(5, 2),
    Weight DECIMAL(5, 2),
    Rating DECIMAL(5, 2)
);
```

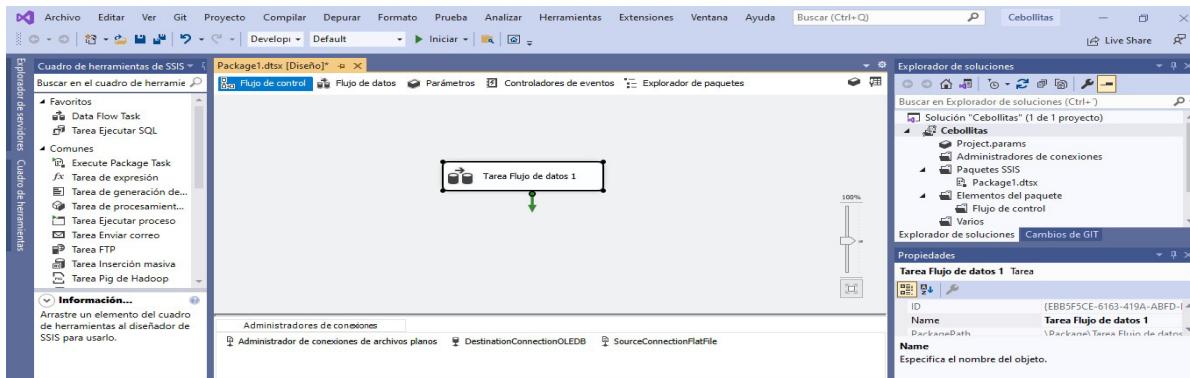
En el visual Studio 2019

En el cuadro de diálogo Nuevo proyecto (New Project), en el panel Plantillas Instaladas (Installed Templates), seleccione el Proyecto de Integration Services o la plantilla del Asistente para proyectos de conexiones de Integration Services y colocamos en nuestra carpeta con todos nuestros recursos.



- Utilizar el paquete que viene por defecto en el `proyecto (Package.dtsx)

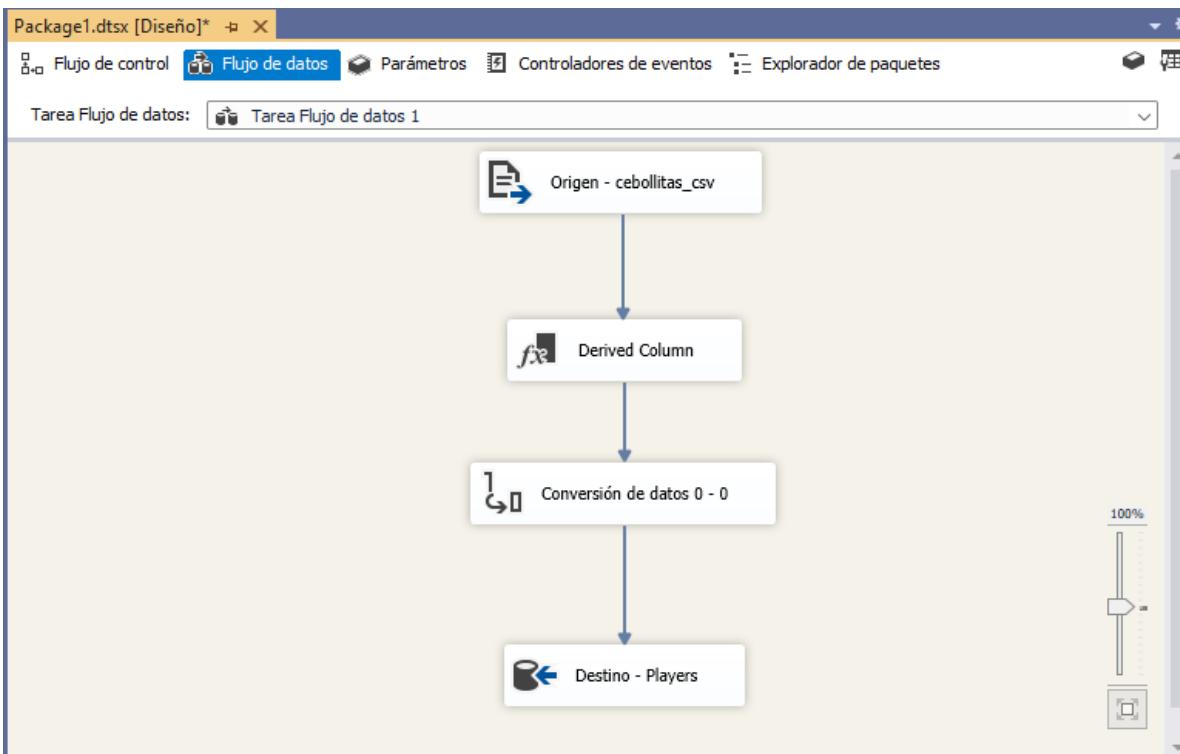
- Agregar en la pestaña Control Flow un control Data Flow Task



. Hacer doble clic en el control y agregar los siguientes controles

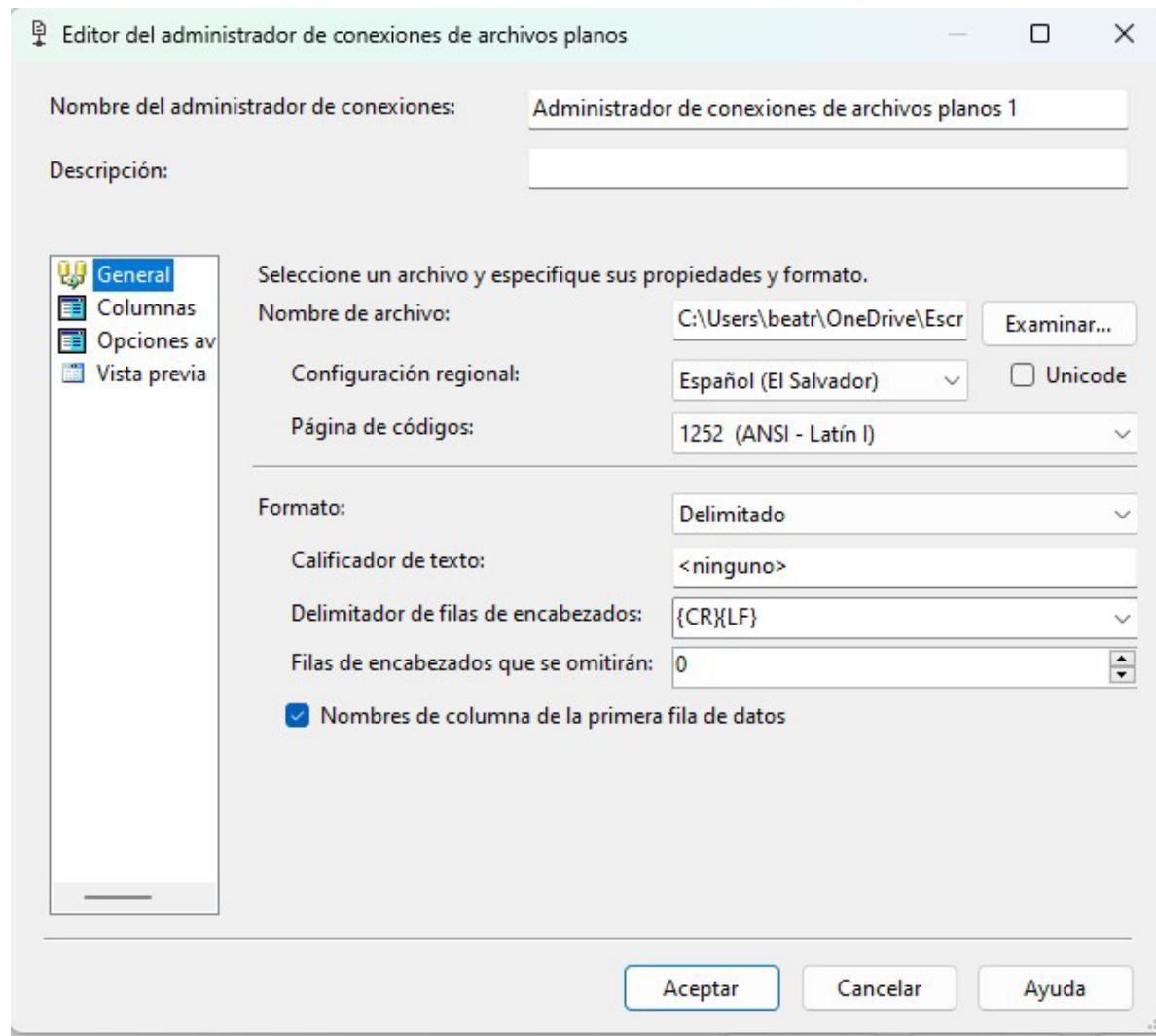
| Nombre del control | Cantidad |
|------------------------------|----------|
| Flat file source | 1 |
| Derived Column | 1 |
| Conversión de datos 0 - 0 | 1 |
| OLE DB Destination | 1 |

El paquete ETL queda de la siguiente manera



Agregamos la siguiente configuracion para el control Flat file Source

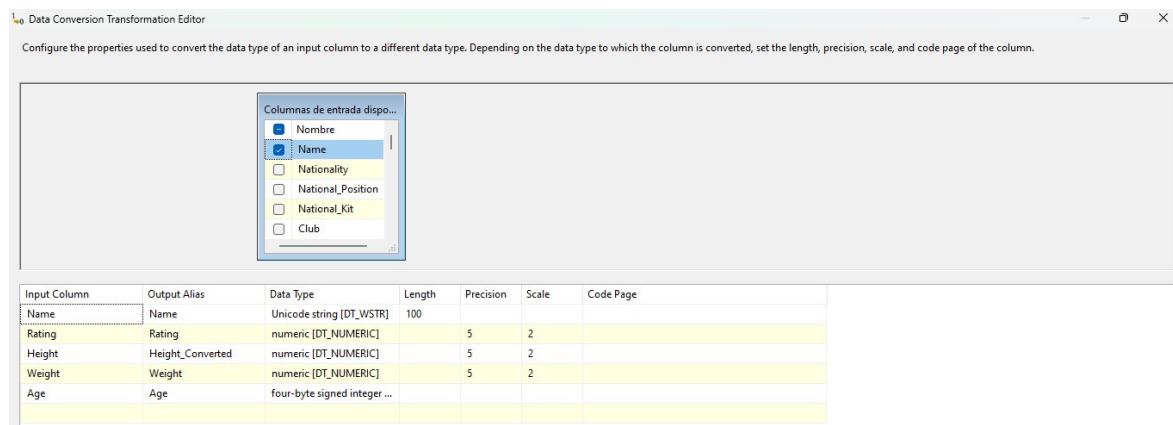
- Cargamos el archivo .csv el cual será mi origen en los datos



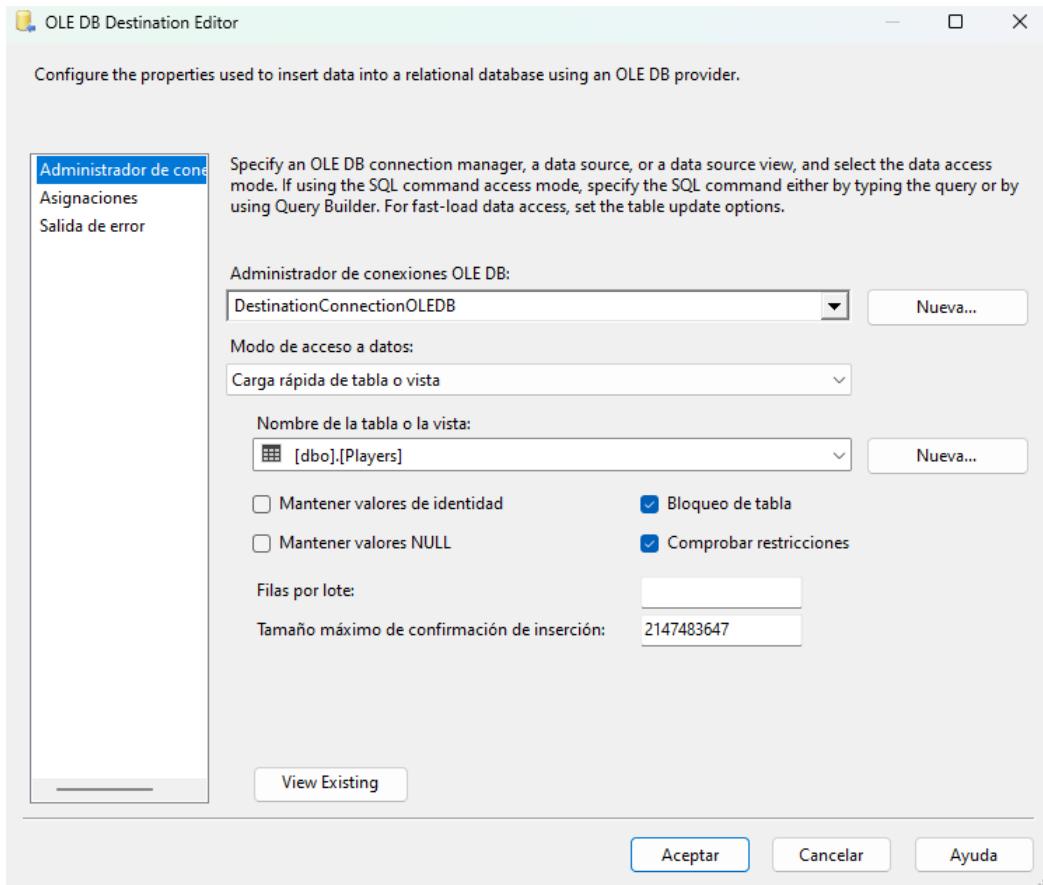
En el Derived Column colocamos las expresiones para que nos tome los valores de Height y Weight debido que vienen sus unidades con los valores que utilizaremos

| Derived Column Name | Derived Column | Expression | Data Type | Length | Precision | Scale | Code Page |
|---------------------|---------------------|--|-----------------|--------|-----------|-------|-----------------------|
| Height | Reemplazar 'Height' | (DT_DECIMAL,5)REPLACE(Height, " cm", "") | string [DT_STR] | 50 | | | 1252 (ANSI - Latin I) |
| Weight | Reemplazar 'Weight' | (DT_DECIMAL,5)REPLACE(Weight, " kg", "") | string [DT_STR] | 50 | | | 1252 (ANSI - Latin I) |

En el Conversión de datos convertimos los datos en su formato que vienen al formato que necesitamos para que guarde en la base de datos SQL



En el OLE DB Destination elegimos la conexión para que nos guarde en nuestra nueva tabla los datos que necesitamos



Ejecutamos el paquete y vemos que no hay ningún error



Abrimos el SQL Server Manager Studio y ejecutamos la siguiente query para visualizar la tabla con los datos que vamos a necesitar para ver las consulta

The screenshot shows the SQL Server Management Studio results grid after executing the query 'select * from Players'. The results show 14 rows of player data:

| | Name | Age | Height | Weight | Rating | PlayerID |
|----|--------------------|-----|--------|--------|--------|----------|
| 1 | Cristiano Ronaldo | 32 | 185.00 | 80.00 | 94.00 | 1 |
| 2 | Lionel Messi | 29 | 170.00 | 72.00 | 93.00 | 2 |
| 3 | Neymar | 25 | 174.00 | 68.00 | 92.00 | 3 |
| 4 | Luis Suárez | 30 | 182.00 | 85.00 | 92.00 | 4 |
| 5 | Manuel Neuer | 31 | 193.00 | 92.00 | 92.00 | 5 |
| 6 | De Gea | 26 | 193.00 | 82.00 | 90.00 | 6 |
| 7 | Robert Lewandowski | 28 | 185.00 | 79.00 | 90.00 | 7 |
| 8 | Gareth Bale | 27 | 183.00 | 74.00 | 90.00 | 8 |
| 9 | Zlatan Ibrahimović | 35 | 195.00 | 95.00 | 90.00 | 9 |
| 10 | Thibaut Courtois | 24 | 199.00 | 91.00 | 89.00 | 10 |
| 11 | Jérôme Boateng | 28 | 192.00 | 90.00 | 89.00 | 11 |
| 12 | Eden Hazard | 26 | 173.00 | 74.00 | 89.00 | 12 |
| 13 | Luka Modrić | 31 | 174.00 | 65.00 | 89.00 | 13 |
| 14 | Mesut Özil | 28 | 180.00 | 76.00 | 89.00 | 14 |

El ejercicio nos pide generar una tabla para conocer las estadísticas de Height, weight y rating por age la cual se ejecutará con el siguiente query

The screenshot shows a SQL query being run in SSMS. The query is:

```
SELECT
    age,
    AVG(height) AS AvgHeight,
    AVG(weight) AS AvgWeight,
    AVG(rating) AS AvgRating
FROM Players
GROUP BY age
ORDER BY age;
```

The results pane displays a table with four columns: age, AvgHeight, AvgWeight, and AvgRating. The data is as follows:

| | age | AvgHeight | AvgWeight | AvgRating |
|----|-----|------------|-----------|-----------|
| 1 | 17 | 179.050890 | 70.918575 | 55.496183 |
| 2 | 18 | 179.893283 | 72.015671 | 56.866791 |
| 3 | 19 | 180.385052 | 72.567673 | 58.791879 |
| 4 | 20 | 180.540067 | 73.235586 | 61.032760 |
| 5 | 21 | 180.390103 | 73.493216 | 63.483320 |
| 6 | 22 | 180.877008 | 74.222903 | 65.015629 |
| 7 | 23 | 180.885577 | 74.791387 | 66.146958 |
| 8 | 24 | 181.230747 | 75.387421 | 67.769252 |
| 9 | 25 | 181.118252 | 75.469288 | 68.456260 |
| 10 | 26 | 181.481052 | 76.222915 | 69.119631 |
| 11 | 27 | 180.971761 | 75.754848 | 69.587276 |
| 12 | 28 | 182.018951 | 76.643849 | 70.035615 |
| 13 | 29 | 181.176544 | 76.247984 | 70.181602 |
| 14 | 30 | 181.656557 | 76.547745 | 70.193237 |
| 15 | 31 | 181.834504 | 77.094977 | 70.685922 |
| 16 | 32 | 181.590853 | 76.929404 | 70.565684 |
| 17 | 33 | 181.409358 | 76.912301 | 69.811653 |

At the bottom, a green bar indicates "Query executed successfully."

Como podemos ver nos promedia el Height, Weight y rating por edades

Luego nos pide que obtengamos la correlación entre rating y age de los jugadores

Y nuestras sentencias de query para realizar dicha correlación es la siguiente

```
SELECT * FROM AgeCorrelationStats WHERE Age = 18;
SELECT * FROM AgeCorrelationStats WHERE Age = 30;
SELECT * FROM AgeCorrelationStats WHERE Age = 47;
```

100 %

Results Messages

| | Age | AvgWeight | AvgHeight | AvgRating | MinWeight | MaxWeight | MinHeight | MaxHeight | MinRating | MaxRating |
|---|-----|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|
| 1 | 18 | 72.02 | 179.90 | 56.89 | 54.00 | 90.00 | 162.00 | 198.00 | 45.00 | 79.00 |

| | Age | AvgWeight | AvgHeight | AvgRating | MinWeight | MaxWeight | MinHeight | MaxHeight | MinRating | MaxRating |
|---|-----|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|
| 1 | 30 | 76.55 | 181.66 | 70.35 | 56.00 | 110.00 | 158.00 | 207.00 | 50.00 | 92.00 |

| | Age | AvgWeight | AvgHeight | AvgRating | MinWeight | MaxWeight | MinHeight | MaxHeight | MinRating | MaxRating |
|---|-----|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|
| 1 | 47 | 77.00 | 185.00 | 45.00 | 77.00 | 77.00 | 185.00 | 185.00 | 45.00 | 45.00 |

Aquí podemos ver una correlación por edad

Que los de 30 años tienen un promedio de rating de 70.35 y es mayor que los de 44 años que tienen un rating de 45

De igual manera podemos ver el promedio que tiene en esas edades tanto de peso como altura

2. El cliente ha proporcionado un archivo CSV con datos de diversos países y múltiples métricas relacionadas con la calidad de vida y las condiciones económicas. El objetivo es extraer, transformar y cargar estos datos en una base de datos SQL Server para analizar cuáles países tienen las mejores condiciones de vida, comparar la seguridad y libertad personal por país, y finalmente calcular el promedio de estas puntuaciones por continente

PASOS

Desde el archivo CSV que se nos proporcionó se extraen los datos a una base de datos SQL server

| Country | AverageScore | SafetySecurity | PersonnelFreedom | Governance | SocialCapital | InvestmentEnvironment | EnterpriseConditions | MarketAccessInfrastructure | EconomicQuality | LivingConditions |
|------------------|--------------|----------------|------------------|------------|---------------|-----------------------|----------------------|----------------------------|-----------------|------------------|
| A Denmark | 84.55 | 92.59 | 94.09 | 89.45 | 82.56 | 82.42 | 79.64 | 78.79 | 76.81 | 95.77 |
| A Sweden | 83.67 | 90.97 | 91.9 | 86.41 | 78.29 | 82.81 | 75.54 | 79.67 | 76.18 | 95.33 |
| A Norway | 83.59 | 93.3 | 94.1 | 89.66 | 79.03 | 82.24 | 75.95 | 75.87 | 77.25 | 94.7 |
| A Finland | 83.47 | 89.56 | 91.96 | 90.41 | 77.27 | 84.12 | 77.25 | 78.74 | 70.28 | 94.46 |
| A Switzerland | 83.42 | 95.68 | 87.5 | 87.67 | 69.14 | 80.81 | 83.84 | 78.65 | 79.71 | 94.65 |
| A Netherlands | 82.32 | 91.19 | 90.08 | 87.34 | 74.03 | 84.11 | 79.09 | 80.82 | 74.34 | 95.86 |
| A Luxembourg | 81.83 | 96.32 | 89.2 | 86.31 | 66.6 | 78.81 | 80.72 | 80.03 | 76.93 | 94.55 |
| A Iceland | 81.02 | 91.64 | 88.74 | 83.3 | 77.75 | 79.2 | 72.86 | 76.07 | 69.92 | 93.82 |
| A Germany | 80.81 | 87.92 | 87.7 | 84.39 | 65.96 | 78.87 | 79.7 | 80.23 | 73.96 | 94.42 |
| A New Zealand | 80.47 | 85.07 | 87.56 | 87.19 | 79.88 | 82.58 | 72.82 | 74.6 | 69.88 | 90.66 |
| A Ireland | 80.31 | 90.97 | 88.59 | 81.72 | 67.73 | 80.43 | 75.29 | 74.07 | 77.81 | 92.65 |
| A United Kingdom | 79.95 | 87.63 | 85.64 | 80.63 | 67.77 | 81.49 | 78.34 | 78.63 | 73.31 | 94.16 |
| A Canada | 79.62 | 87.92 | 86.62 | 82.34 | 73.6 | 80.68 | 76.22 | 77.14 | 65.34 | 93.49 |
| A Austria | 79.38 | 90.94 | 85.99 | 81.19 | 67.94 | 79.61 | 73.26 | 77.61 | 68.41 | 92.51 |
| A Australia | 79.36 | 87.91 | 84.53 | 82.81 | 77.42 | 78.61 | 70.82 | 72.79 | 68.89 | 93.06 |
| A Japan | 78.22 | 92.78 | 79.14 | 79.67 | 43.82 | 83.1 | 80.11 | 79.32 | 66.35 | 92.86 |
| A Singapore | 78.21 | 92.05 | 48.63 | 79.12 | 64.68 | 83.23 | 78.05 | 85.75 | 80.1 | 93.35 |
| A Belgium | 77.84 | 85.76 | 87.7 | 80.31 | 64.55 | 81.12 | 70.26 | 76.63 | 66.39 | 92.78 |
| A United States | 77.44 | 72.43 | 78.65 | 75.18 | 73.91 | 79.48 | 82.85 | 80.4 | 72.34 | 90.74 |
| A Taiwan | 77.36 | 92.96 | 79.23 | 77.68 | 60.42 | 78.6 | 79.66 | 71.15 | 73.86 | 90.22 |
| A Estonia | 77.31 | 86.12 | 87.2 | 79.03 | 61.94 | 73.32 | 70.85 | 71.71 | 73.32 | 91.95 |
| A Hong Kong | 76.9 | 89.16 | 53.28 | 72.31 | 57.03 | 84.99 | 83.63 | 81.07 | 78.19 | 91.36 |
| A France | 76.73 | 82.98 | 79.06 | 77.24 | 60.6 | 79.42 | 73.42 | 76.98 | 65.81 | 92.61 |
| A Spain | 76.03 | 86.87 | 83.65 | 72.48 | 69.27 | 76.13 | 69.93 | 77.68 | 57.91 | 93.81 |
| A Czech Republic | 75.08 | 90.64 | 82.53 | 68.72 | 61.62 | 74.18 | 62.88 | 69.7 | 72.12 | 91.64 |
| A Portugal | 74.64 | 86.03 | 85.78 | 73.19 | 62.92 | 71.81 | 67.94 | 76.33 | 60.63 | 91.85 |

Creamos la base de datos a la que llamaremos COUNTRYY

Extraemos los datos a nuestra base

The screenshot shows the Microsoft SQL Server Management Studio interface. In the Object Explorer, the 'COUNTRY' database is selected. In the center pane, a query window displays the following T-SQL code:

```

SELECT TOP (1000) [Country]
,[AveragScore]
,[SafetySecurity]
,[PersonalFreedom]
,[Governance]
,[SocialCapital]
,[InvestmentEnvironment]
,[EnterpriseConditions]
,[MarketAccessInfrastructure]
,[EconomicQuality]
,[LivingConditions]
,[Health]
,[Education]
,[NaturalEnvironment]
FROM [COUNTRY]..[dbo].[country$DatosExternos_1]

```

The results grid shows 167 rows of data from the 'country\$DatosExternos_1' table. The columns include Country, AveragScore, SafetySecurity, PersonalFreedom, Governance, SocialCapital, InvestmentEnvironment, EnterpriseConditions, MarketAccessInfrastructure, EconomicQuality, LivingConditions, Health, Education, and NaturalEnvironment. The data includes entries for various countries like Denmark, Sweden, Norway, Finland, Switzerland, Netherlands, Luxembourg, Israel, Germany, New Zealand, Iceland, United Kingdom, Canada, India, Australia, Japan, and others.

Empezamos a realizar lo que el ejercicio pide apoyándonos de unos códigos que nos facilitaran a realizar nuestra tarea obteniendo como resultado esto

The screenshot shows the Microsoft SQL Server Management Studio interface. In the Object Explorer, the 'COUNTRY' database is selected. In the center pane, a query window displays the following T-SQL code:

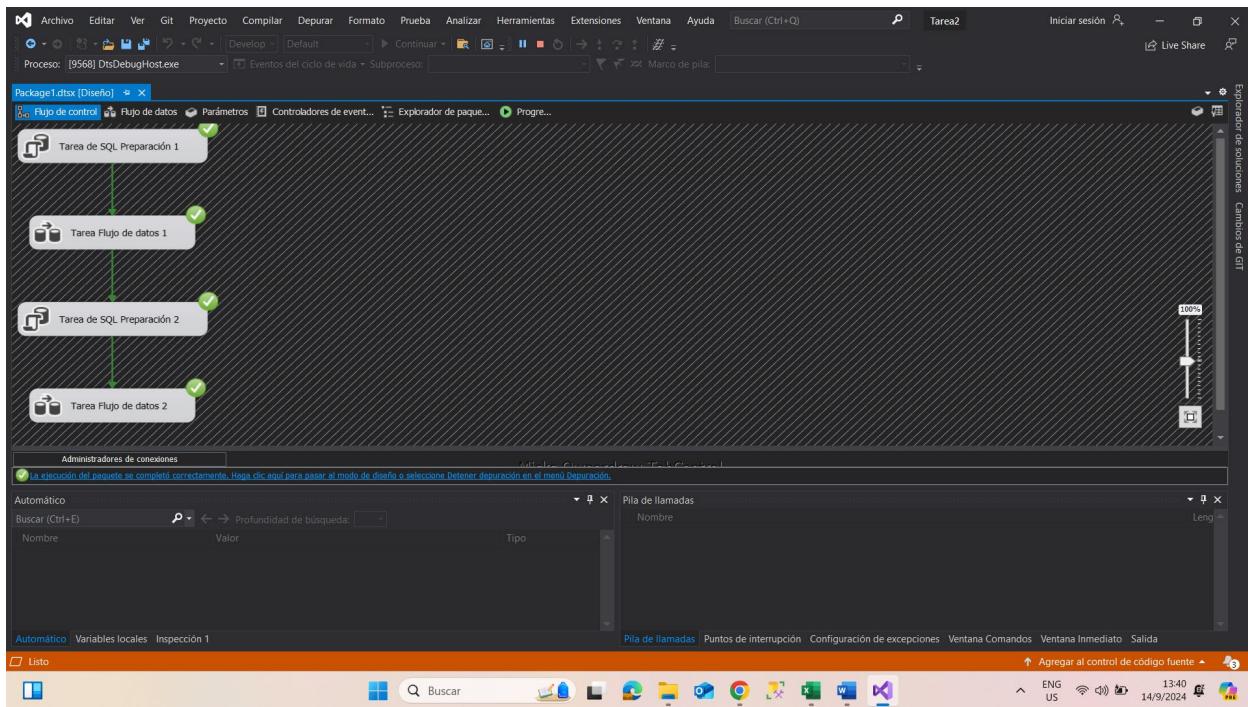
```

USE COUNTRY;
Select Country, SafetySecurity, PersonalFreedom, LivingConditions From country$DatosExternos_1 ORDER BY LivingConditions Desc

```

The results grid shows 167 rows of data from the 'country\$DatosExternos_1' table. The columns include Country, SafetySecurity, PersonalFreedom, and LivingConditions. The data includes entries for various countries like Netherlands, Denmark, Sweden, Norway, Finland, Switzerland, Luxembourg, Israel, Germany, New Zealand, Iceland, United Kingdom, Canada, India, Australia, Japan, and others, ordered by LivingConditions in descending order.

Para el ETL se usan los mismos datos de el archivo CSV y se extraen para que sean ejecutados en Visual Estudio dándonos como resultado una ejecución exitosa



3. El cliente ha proporcionado un archivo CSV con opiniones y valoraciones de los usuarios de la aplicación ChatGPT para Android. El conjunto de datos se actualiza diariamente y captura diversos aspectos de las reseñas, ofreciendo información sobre las experiencias y comentarios de los usuarios a lo largo del tiempo. El objetivo es extraer, transformar y cargar estos datos en una base de datos SQL Server para realizar análisis sobre las valoraciones de los usuarios, identificar tendencias en las opiniones y obtener insights sobre la satisfacción de los usuarios.

Para la ejecución del programa se estará utilizando

Un tipo de archivo plano CSV, llamado chatgpt, el cual se pide se envia a una base de datos para SQL server:

En el cual se procederá a la creación de un programa ETL, en el cual se realizara desde visual studio 2019:

Realizando la creación de este archivo llamado ejercicio3:

En el cual borraremos package1.dtsx que viene por defecto.

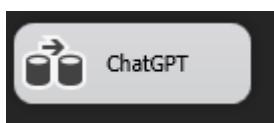
Luego se realizara la conexión de nuestro archivo plano para nuestra base de datos que llamaremos OpiChat:

```
create database OpiChat;  
use OpiChat;
```

Luego realizaremos en esa base de datos una tabla:

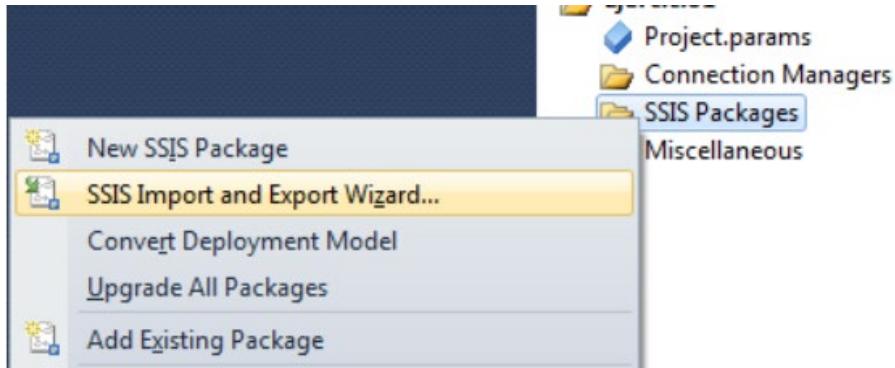
```
|CREATE TABLE UserReviews (  
    ReviewID INT IDENTITY(1,1) PRIMARY KEY,  
    userName NVARCHAR(100),  
    content NVARCHAR(MAX),  
    score INT,  
    reviewDate DATETIME  
)
```

Luego de proceder a crear esto en sql server, pasaremos a colocar una tarea de flujo de datos:

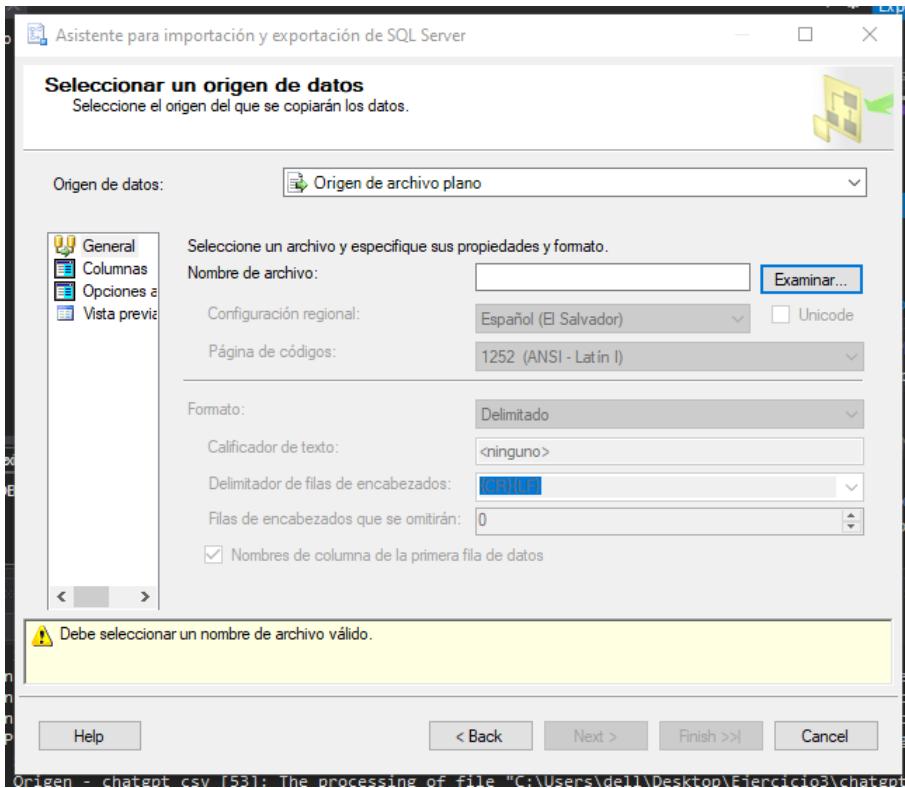


Nota: Para conveniencia procederemos a colocar nombre de nuestro programa.

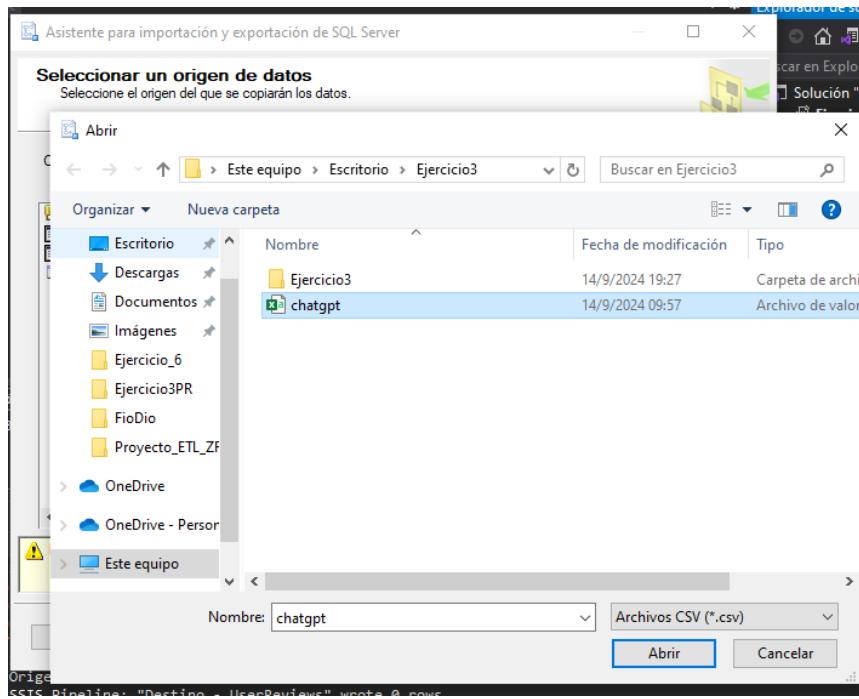
Luego de esto realizaremos el proceso de la conexión en el packages SSIS:



Luego de realizar ese paso buscaremos el tipo de dato plano(csv) que vamos a utilizar para extraer datos:



Se realiza el proceso de selección y luego en la ventana emergente cambiaremos a tipo de archivo .csv:

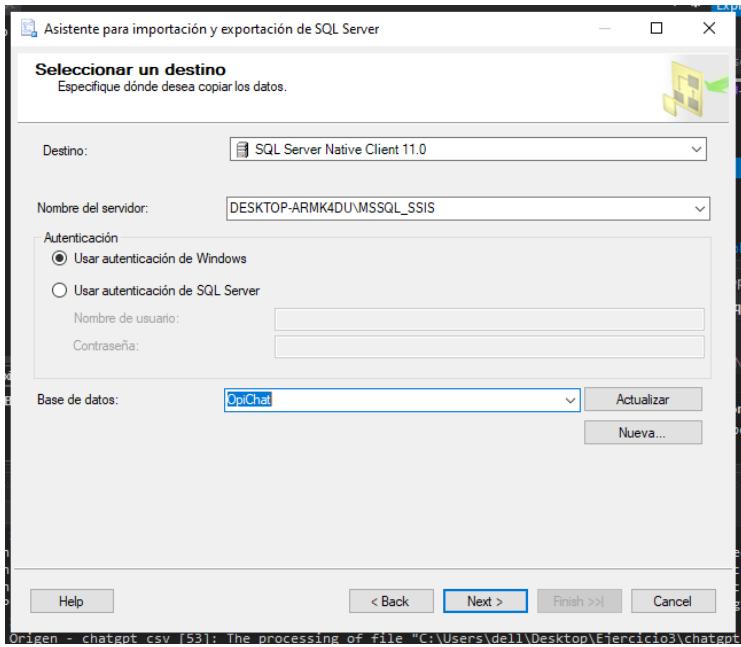


Se realiza los ajustes a los tipos de datos que necesitamos tener estos para la utilización del mismo en nuestro programa ETL.

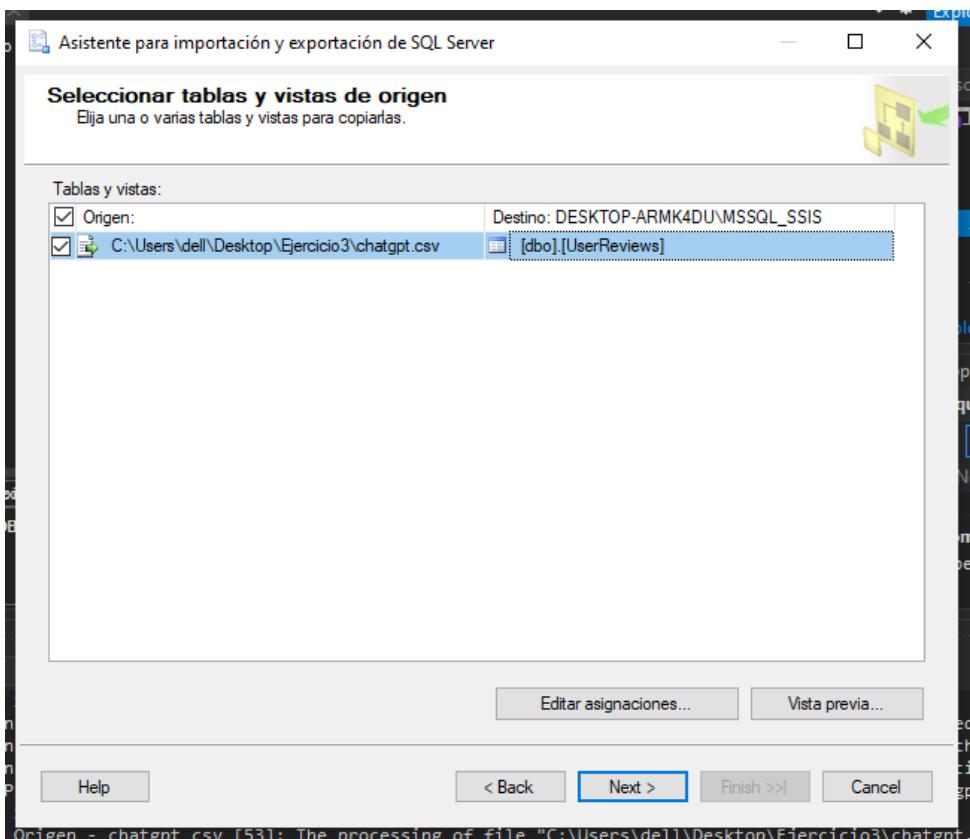
A continuación se procederá a colocar el destino que tendrán estos datos que estamos convitiendo del archivo plano a nuestro sql.

En la ventana se seleccionara el sql native cliente 11, y de ahí el servidor seleccionan el de su servidor

De igual forma se selecciona la base de datos que realizamos la creación anteriormente.



Se coloca la tabla donde se asignaran estos datos que se están convirtiendo del archivo plano al SQL:



Luego de realizar estos pasos daremos doble click sobre la tarea de flujo de datos.

Esto nos enviará a la siguiente ventana de flujo de datos:



Se intenta validar pero la columna score:

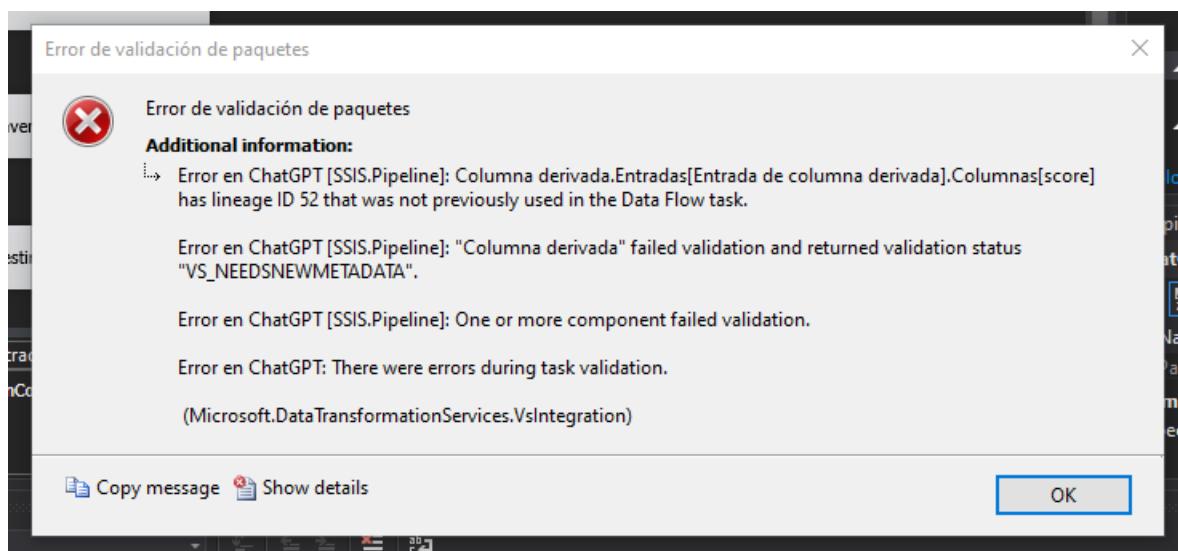
En muchos campos coloca caracteres especiales y emojis, en su estructuración:

Se indica se debe colocar en UTF-8 para que estos caracteres especiales sean leídos y luego enviados a la base de datos, se intentó realizar por medio de expresiones en un derived column:

Adjunto prueba de lo antes mencionado:

excellent Im impressed 🌟 🌟

Esta inserta emojis, lo cual nuestro programa no deja ejecutarse y arrojando el siguiente error:



EJERCICIO 4

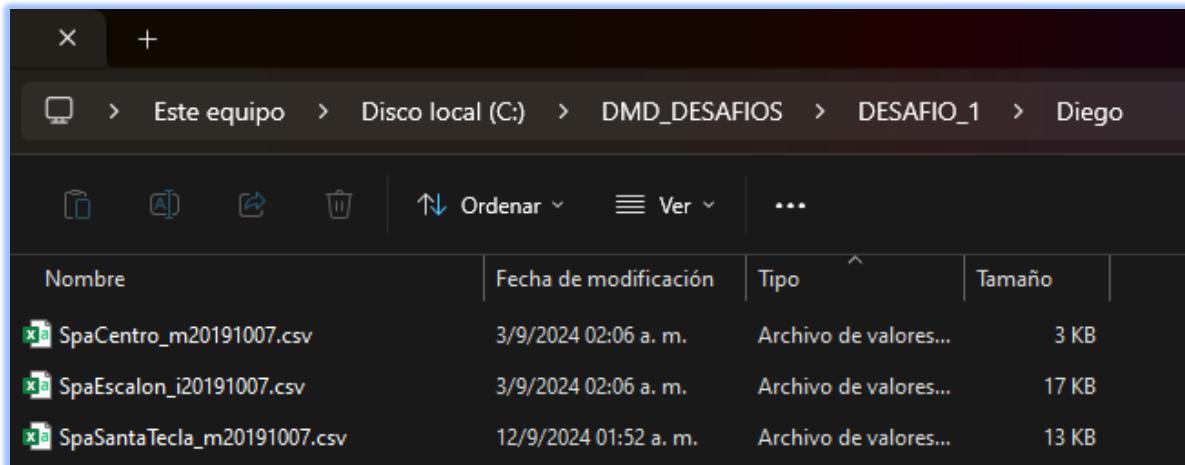
JORGE MARVIN PEÑA

El Spa, "Diego", necesita segmentar sus clientes, para realizar una campaña de fidelización, y le pide a usted que efectué un análisis de sus tres sucursales, que defina **cuantos grupos y que características tienen.**

Desarrollo de ETL

A continuación, se describen los pasos de proceso ETL

1. Revisión de los archivos CSV para verificar que el numero de campos sean el mismo, los campos sean acorde a la información, y que no esten null o vacío.
 - 1.1 Se creo la carpeta con la siguiente ubicación: **C:\DMD_DESAFIOS\DESAFIO_1\Diego** donde inicialmente se almacenaran los archivos .csv



| Nombre | Fecha de modificación | Tipo | Tamaño |
|-----------------------------|-----------------------|-----------------------|--------|
| SpaCentro_m20191007.csv | 3/9/2024 02:06 a. m. | Archivo de valores... | 3 KB |
| SpaEscalon_i20191007.csv | 3/9/2024 02:06 a. m. | Archivo de valores... | 17 KB |
| SpaSantaTecla_m20191007.csv | 12/9/2024 01:52 a. m. | Archivo de valores... | 13 KB |

Se encontró que un archivo tenia una fila que no correspondía a la información por lo cual fue eliminada:



| | |
|-----|--|
| 251 | Alexi Wildman,0,1734.30,3.92,62,0,0,0,1 |
| 252 | id,Sexo,ingresos,PromVisit,Edad,Sauna,Masaje,Hidro,Yoga |
| 253 | Tomkin Stickles,1,2555.23,1.94,21,1,1,0,0 |

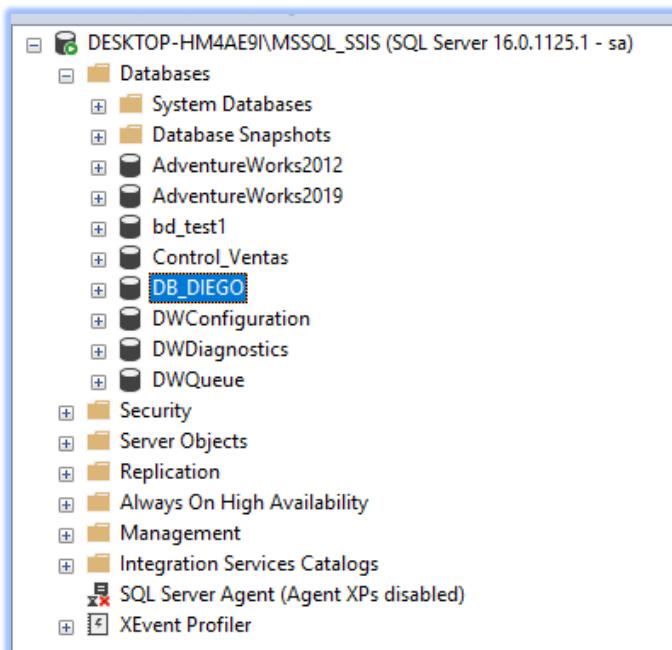
Reporte de Error:

Error: 0xC0202092 en Cargar_Sucursales, Flat File Source [2]: An error occurred while processing file "C:\DMD_DESAFIOS\DESAFIO_1\Diego\SpaSantaTecla_m20191007.csv" on data row 252.

Esto gracias a que al cargar los datos Visual Studio desplego el error.

2. Creación de base de datos y de tabla

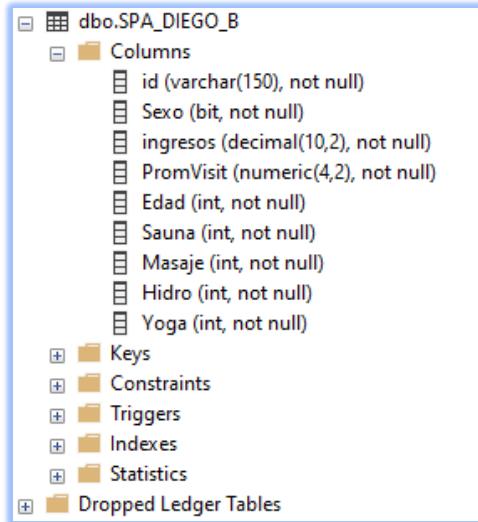
2.1 Creación de Base de Datos desde SQL SERVER 2022



2.2 Creación de Script para la creación de tabla de datos donde se cargarán la data de los tres archivos .csv

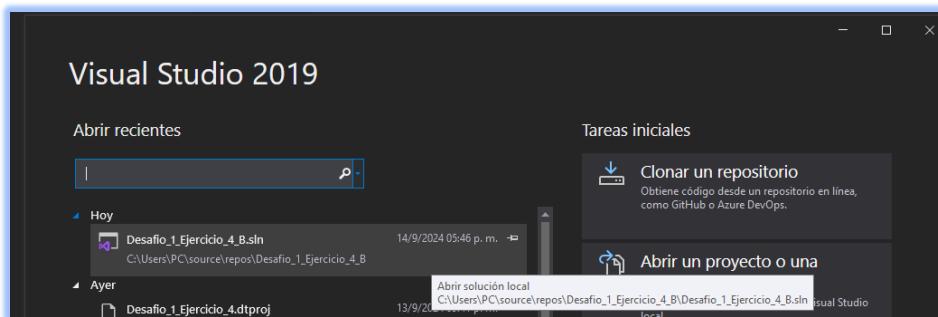
```
IF NOT EXISTS (SELECT * FROM INFORMATION_SCHEMA.TABLES
                WHERE TABLE_SCHEMA = 'DB_DIEGO'
                  AND TABLE_NAME = 'SPA_DIEGO_B')
BEGIN

    CREATE TABLE SPA_DIEGO_B (
        id      VARCHAR(150) NOT NULL,
        Sexo    BIT NOT NULL,
        ingresos DECIMAL(10,2) NOT NULL, -- Ajuste el tipo de datos a DECIMAL para
        representar ingresos con precisión
        PromVisit NUMERIC(4,2) NOT NULL,
        Edad    INT NOT NULL,
        Sauna   INT NOT NULL,
        Masaje  INT NOT NULL,
        Hidro   INT NOT NULL,
        Yoga    INT NOT NULL,
    );
END;
```

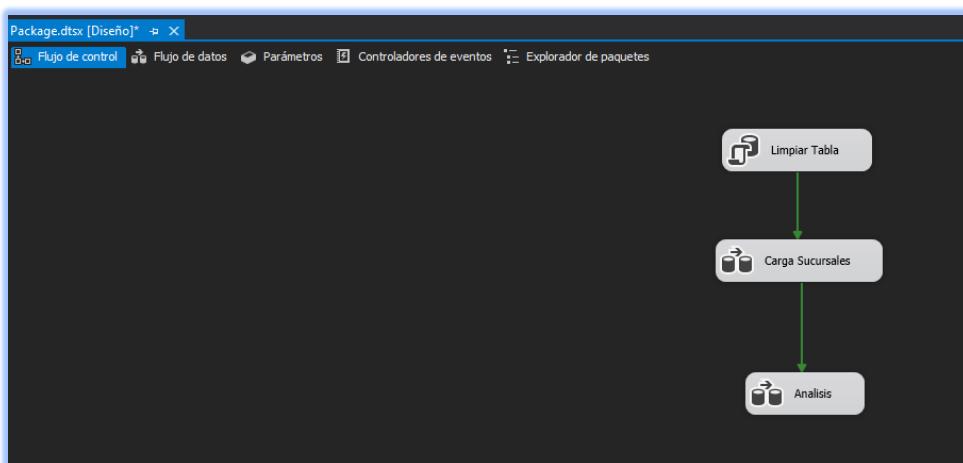


3. Utilización de Herramienta Visual Studio 2019 Community para ETL

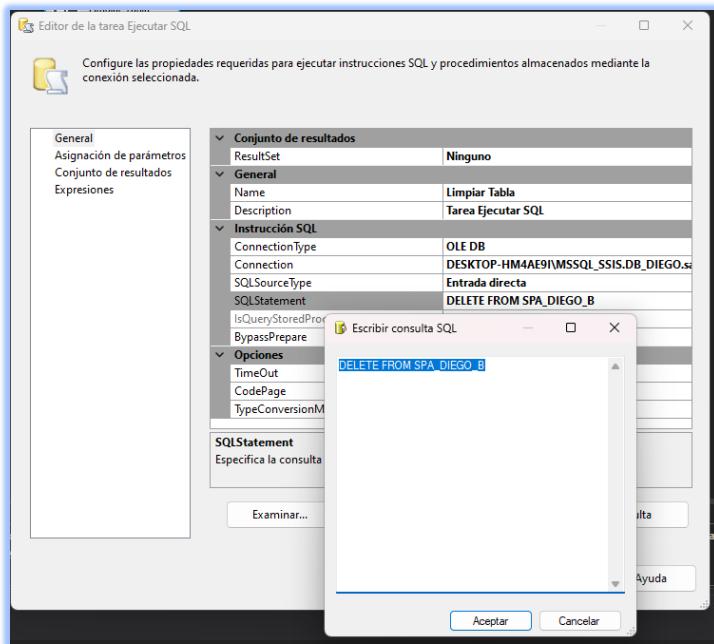
3.1 Se creo proyecto SSIS



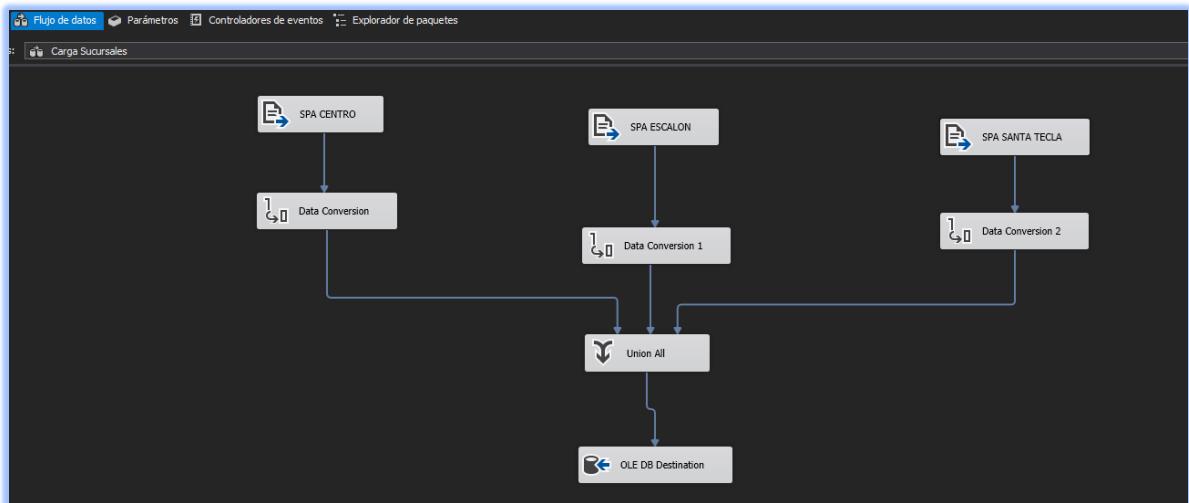
3.2 Se crearon 3 objetos en Flujo de Control:



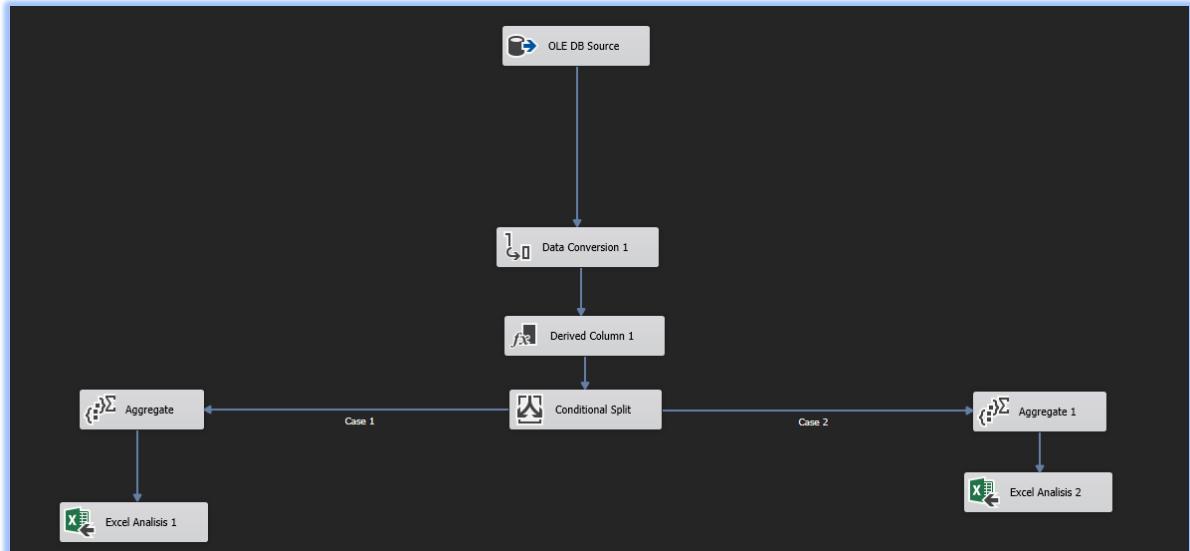
- Script “Tarea Ejecutar SQL” para Limpieza de tabla de Base de datos destino



- Se crea objeto “Data Flow Task” llamado “Carga Sucursales”: que es donde estará la carga de data de los tres archivos .csv, conversión de datos, filtro de limpieza y el almacenamiento de la data a BD de SQL SERVER.



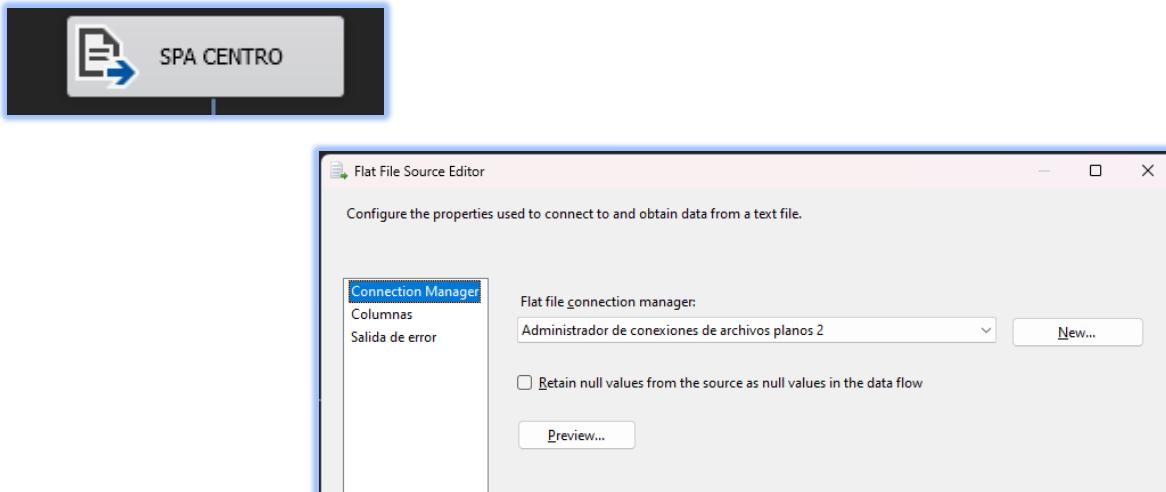
- Creación de Objeto de “Data Flow Task”, llamado “Análisis”: que nos servirá para hacer conversión para reporte, condiciones, y agregaciones para filtrar y automatizar la consulta y pueda exportarse el análisis en Excel



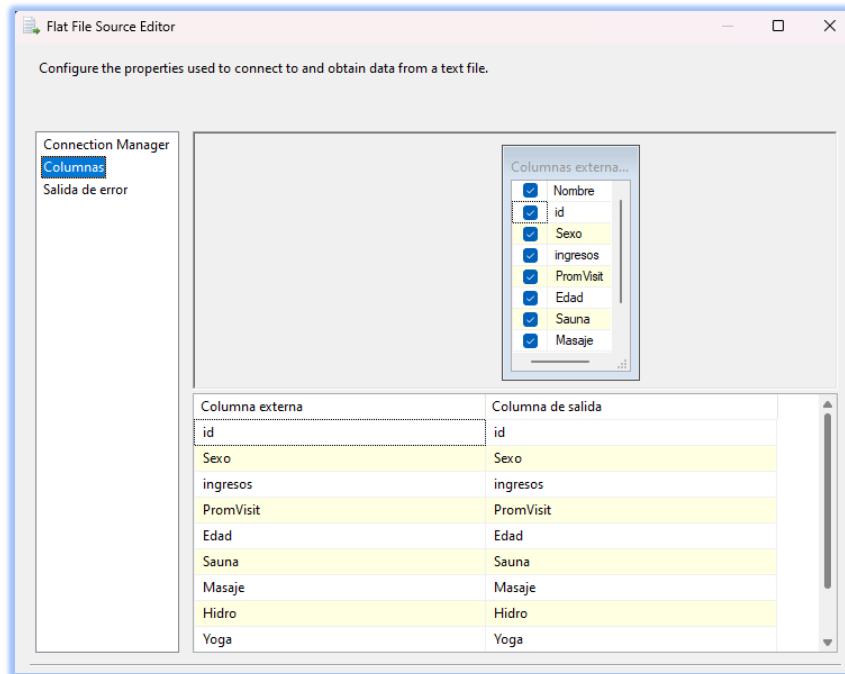
4. Explicación de Procesos de Tarea de Flujos de Datos Carga de datos

4.1 Carga de Sucursales

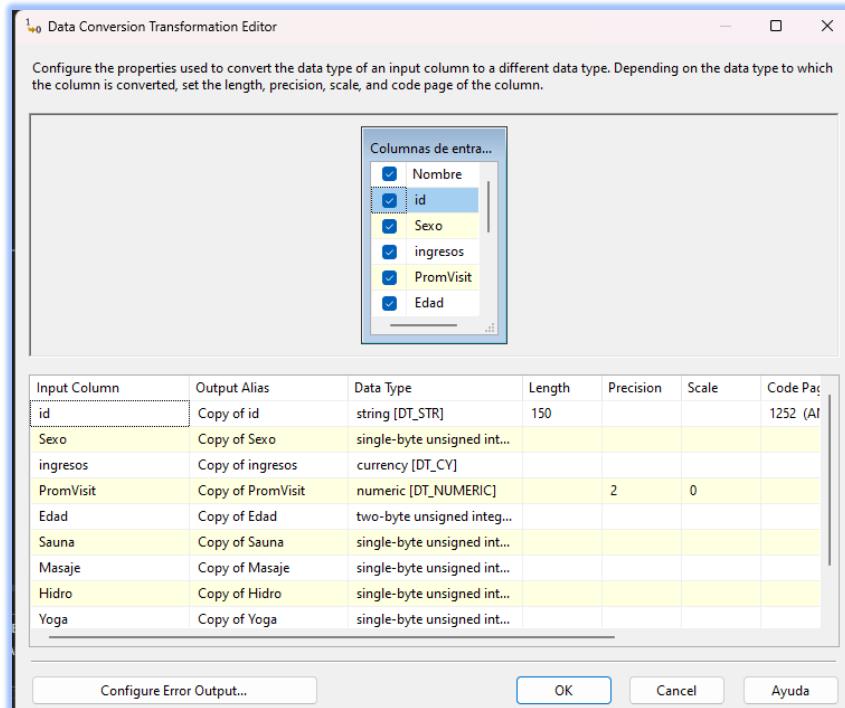
4.1.1 Objeto Flat File Source : para cargar la data de SPA Centro



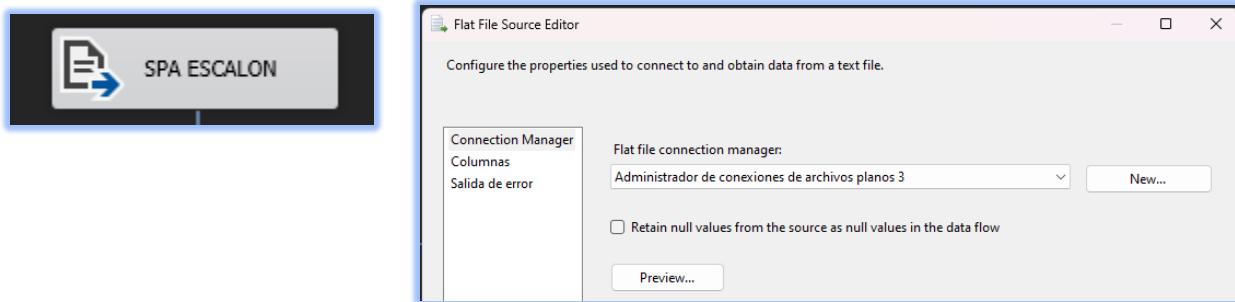
Seleccionamos los campos a cargar



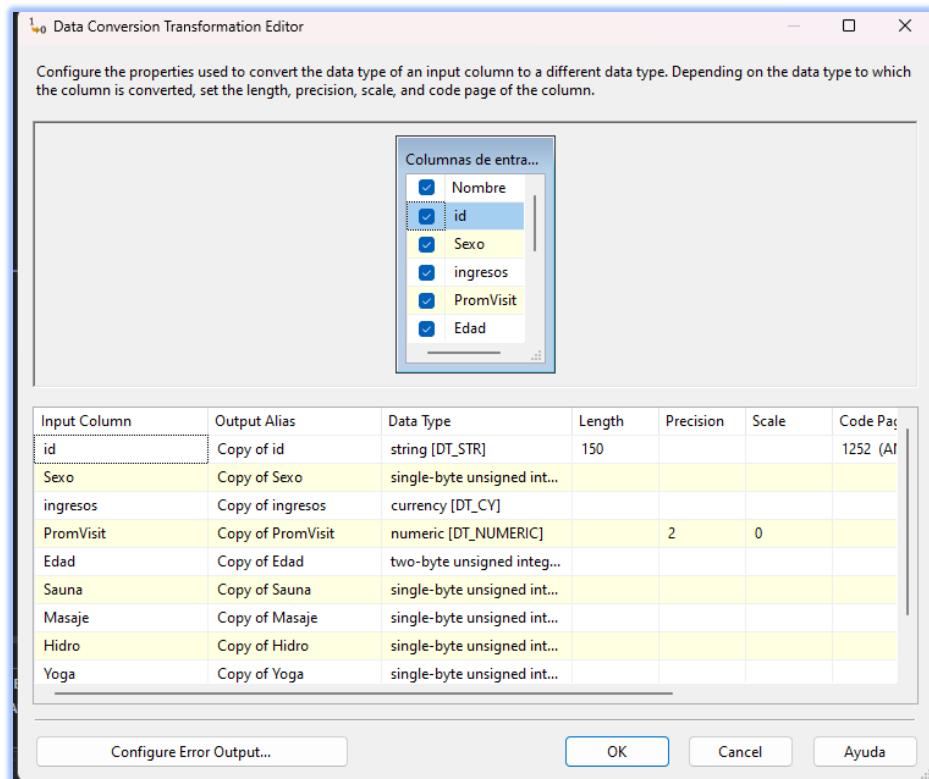
4.1.2 Conversión de data proveniente de Spa Centro



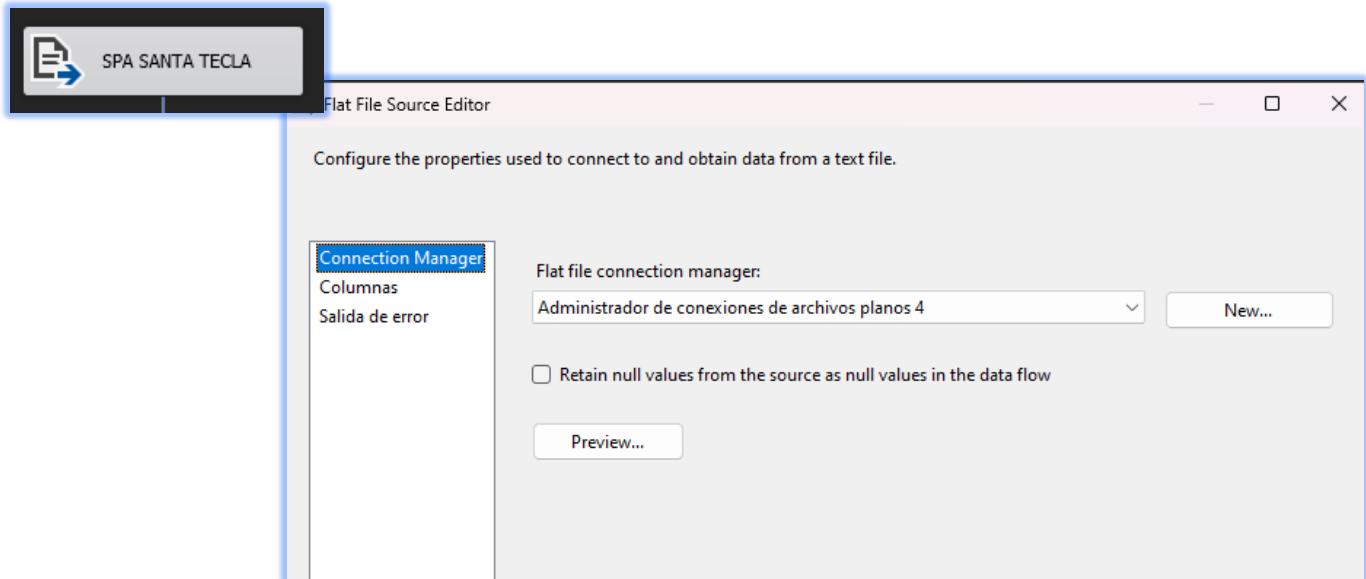
4.1.3 Objeto Flat File Source : para cargar la data de SPA Escalon



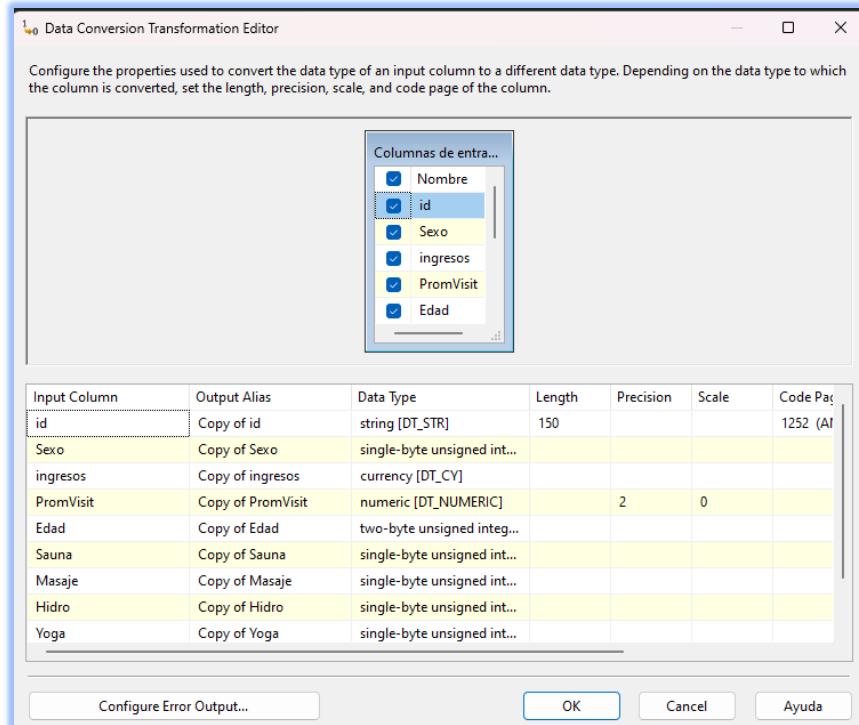
4.1.4 Conversión de data proveniente de Spa Escalon



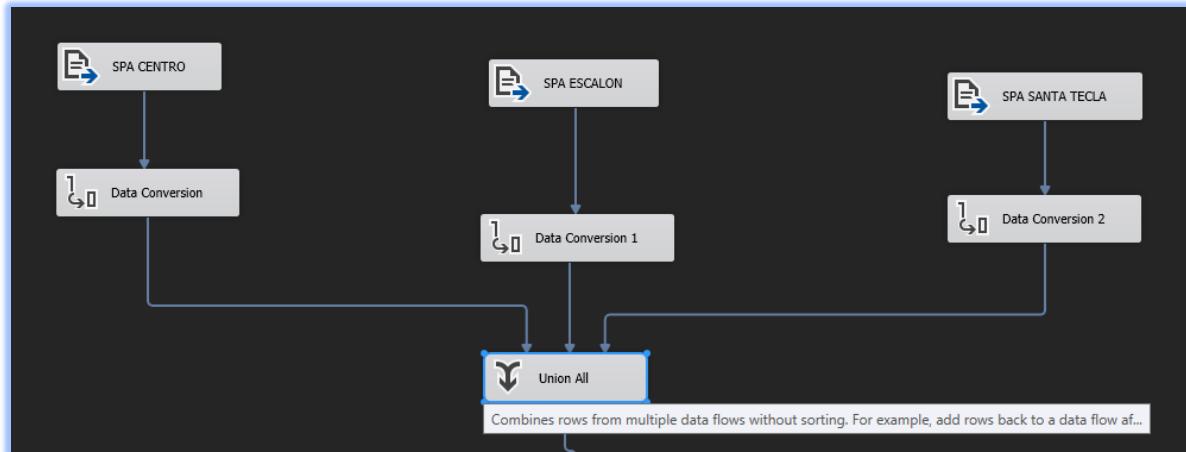
4.1.5 Objeto Flat File Source : para cargar la data de SPA Escalón



4.1.6 Conversión de data proveniente de Spa Santa Tecla



4.2 Unión de orígenes de datos



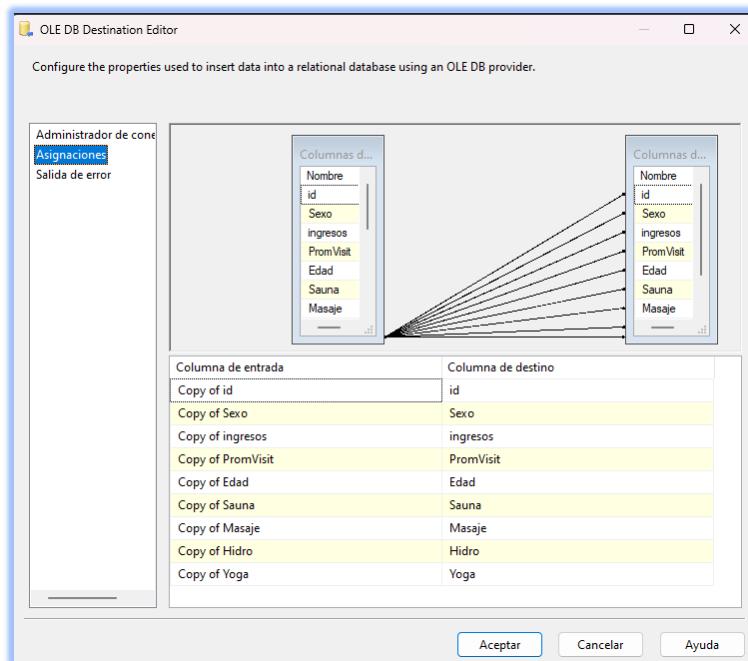
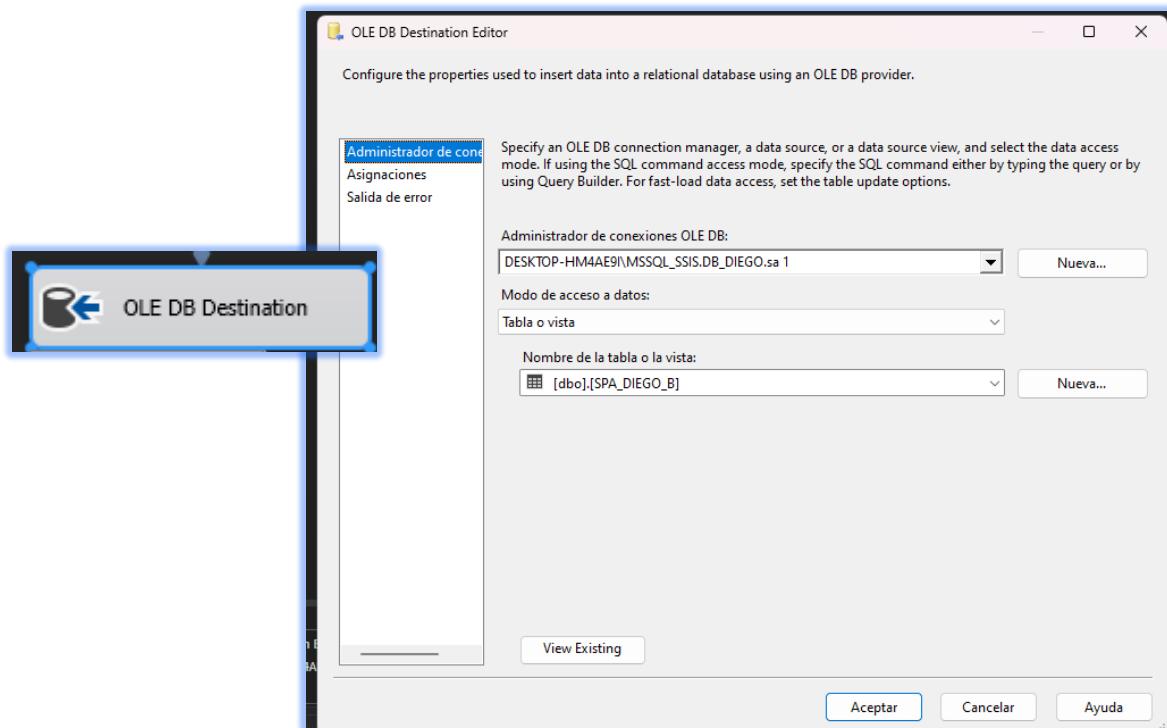
Union All Transformation Editor

Configure the properties used to merge multiple inputs into one output by creating mappings between columns.

| Output Column Name | Union All Input 1 | Union All Input 2 |
|--------------------|-------------------|-------------------|
| id | id | id |
| Sexo | Sexo | Sexo |
| ingresos | ingresos | ingresos |
| PromVisit | PromVisit | PromVisit |
| Edad | Edad | Edad |
| Sauna | Sauna | Sauna |
| Masaje | Masaje | Masaje |
| Hidro | Hidro | Hidro |
| Yoga | Yoga | Yoga |
| Copy of id | Copy of id | Copy of id |
| Copy of Sexo | Copy of Sexo | Copy of Sexo |
| Copy of ingresos | Copy of ingresos | Copy of ingresos |
| Copy of PromVisit | Copy of PromVisit | Copy of PromVisit |
| Copy of Edad | Copy of Edad | Copy of Edad |
| Copy of Sauna | Copy of Sauna | Copy of Sauna |
| Copy of Masaje | Copy of Masaje | Copy of Masaje |
| Copy of Hidro | Copy of Hidro | Copy of Hidro |
| Copy of Yoga | Copy of Yoga | Copy of Yoga |

Aceptar Cancelar Ayuda

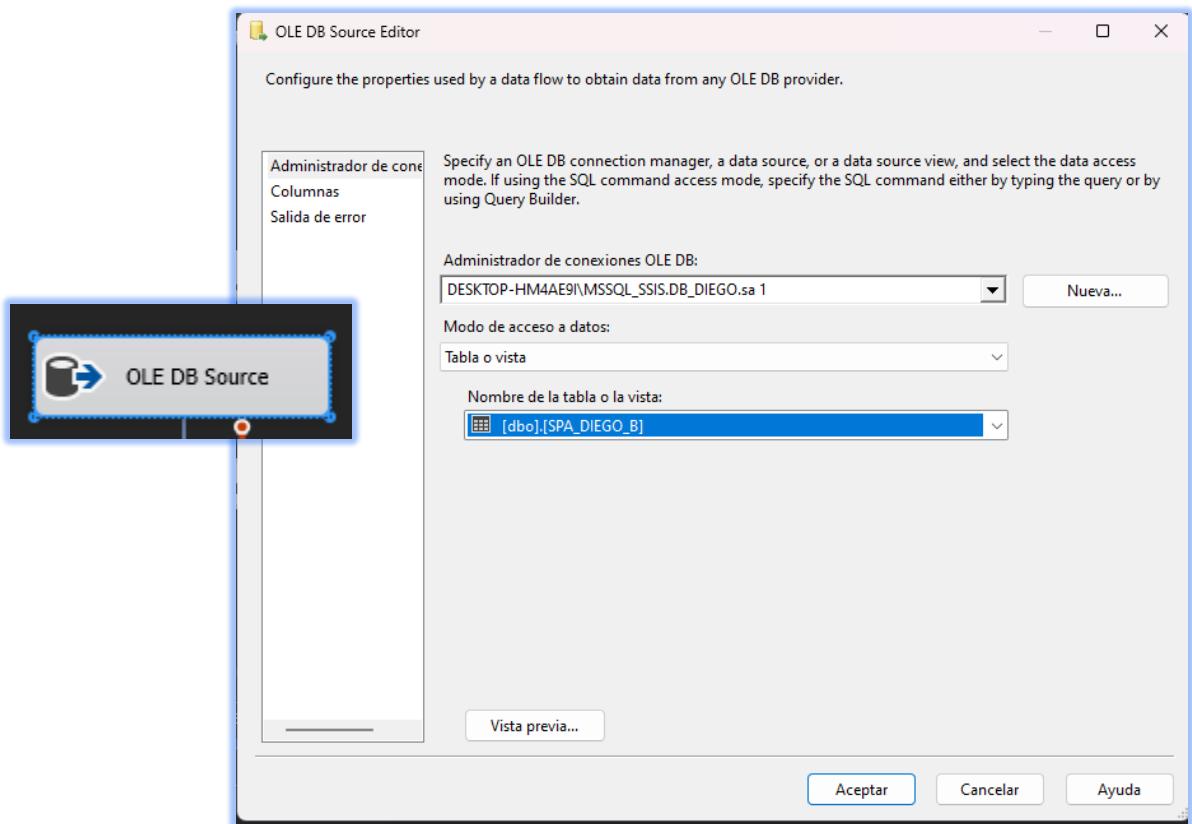
4.3 OLE DB Destination

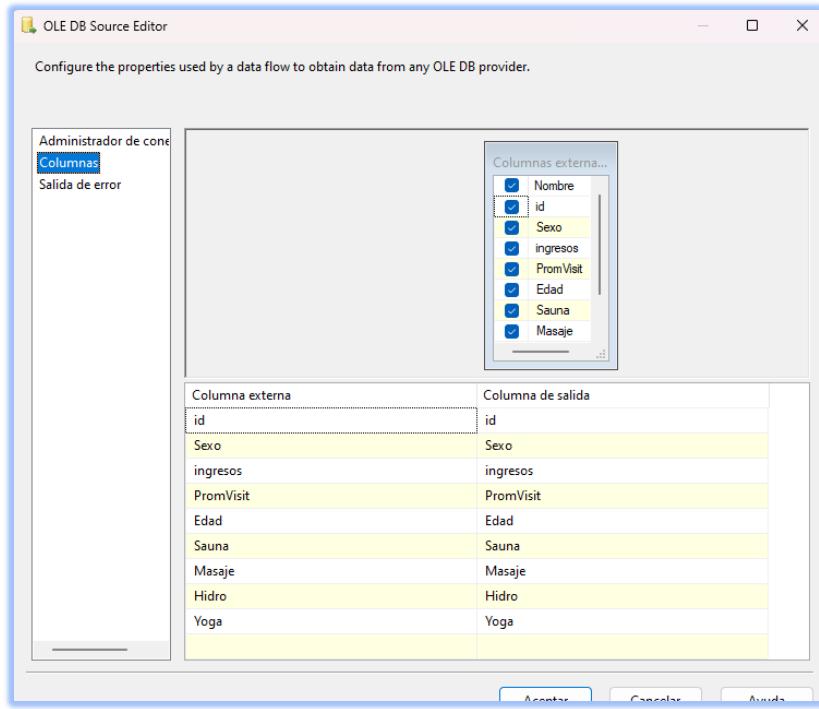


5. Explicación de Procesos de Tarea de Flujo de Análisis de datos

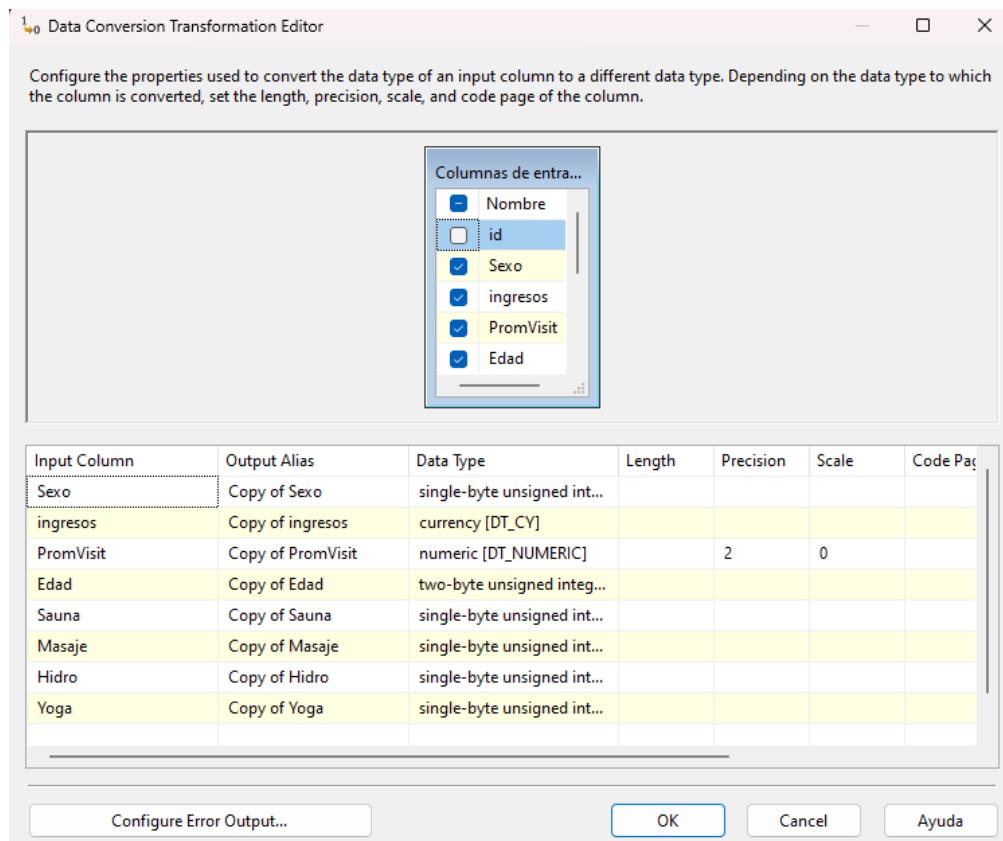
5.1 OLE DB Source

Se encarga de cargar los archivos de base de datos

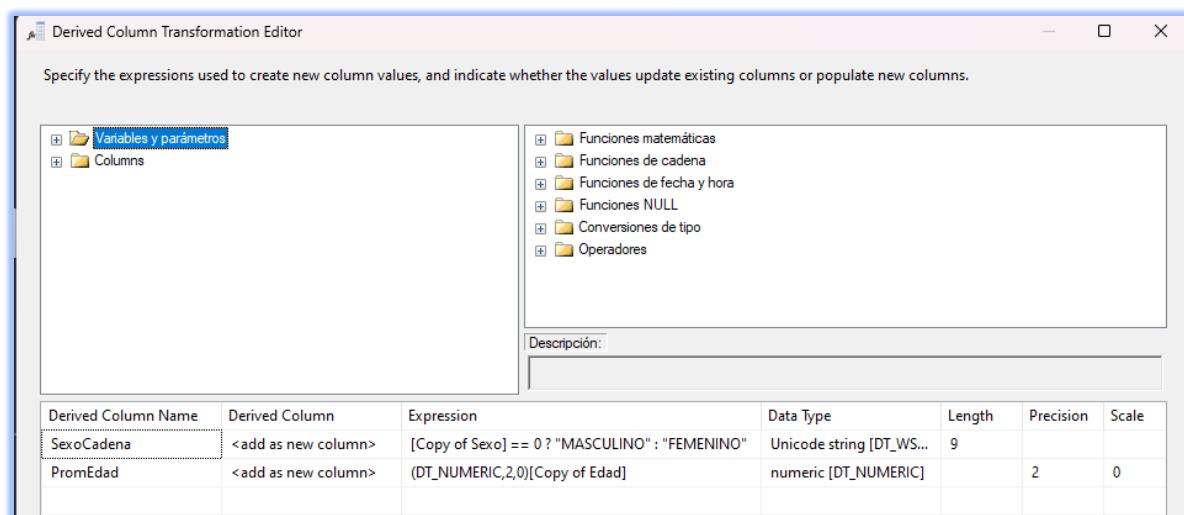




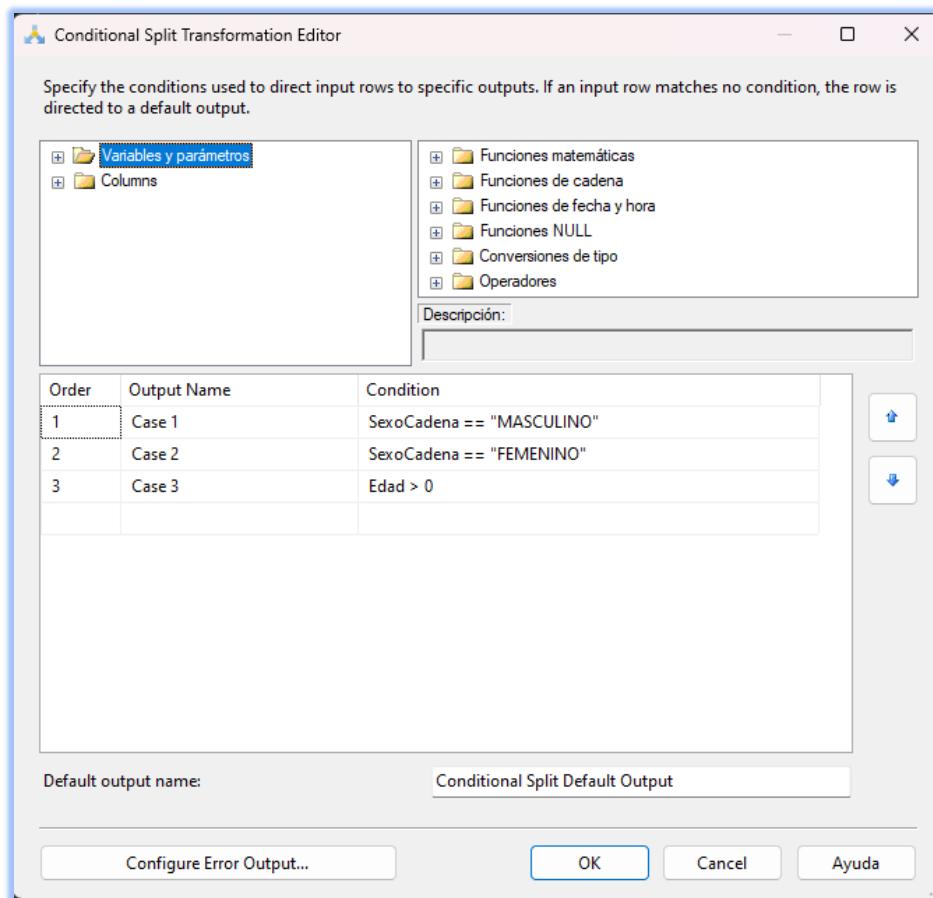
5.2 Conversión de datos



5.3 Columna Derivada



5.4 Condiciones



5.5 Objeto de Agregado para agrupar personas de sexo Masculino

Σ Aggregate Transformation Editor

Aggregations Advanced

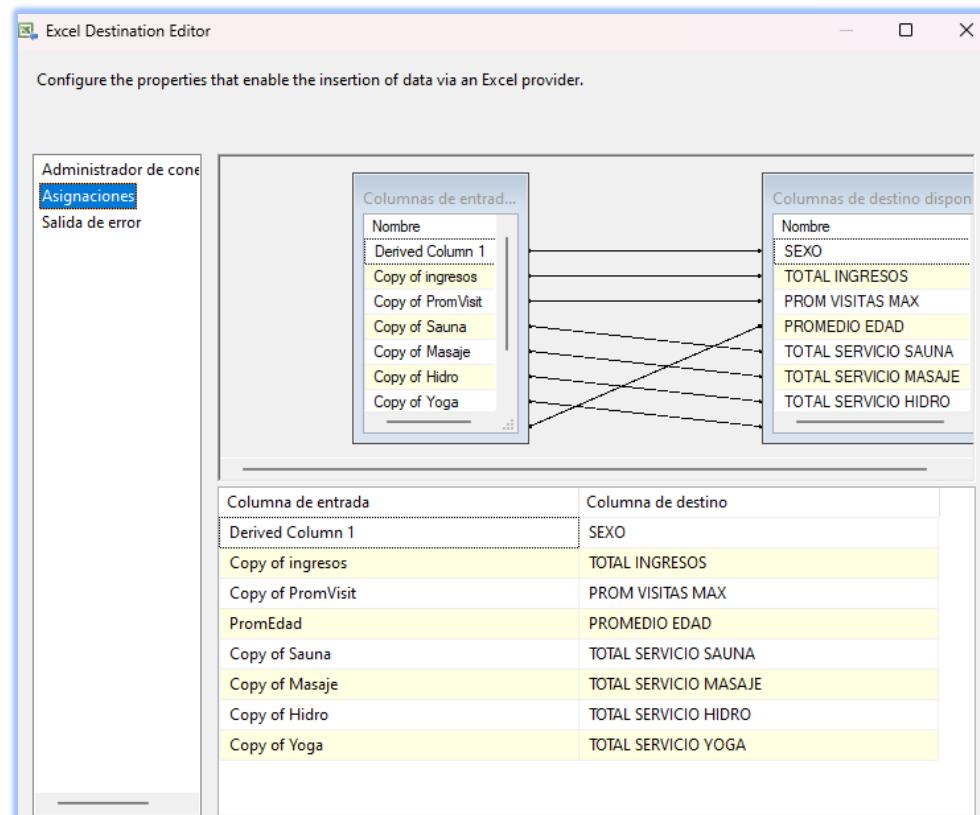
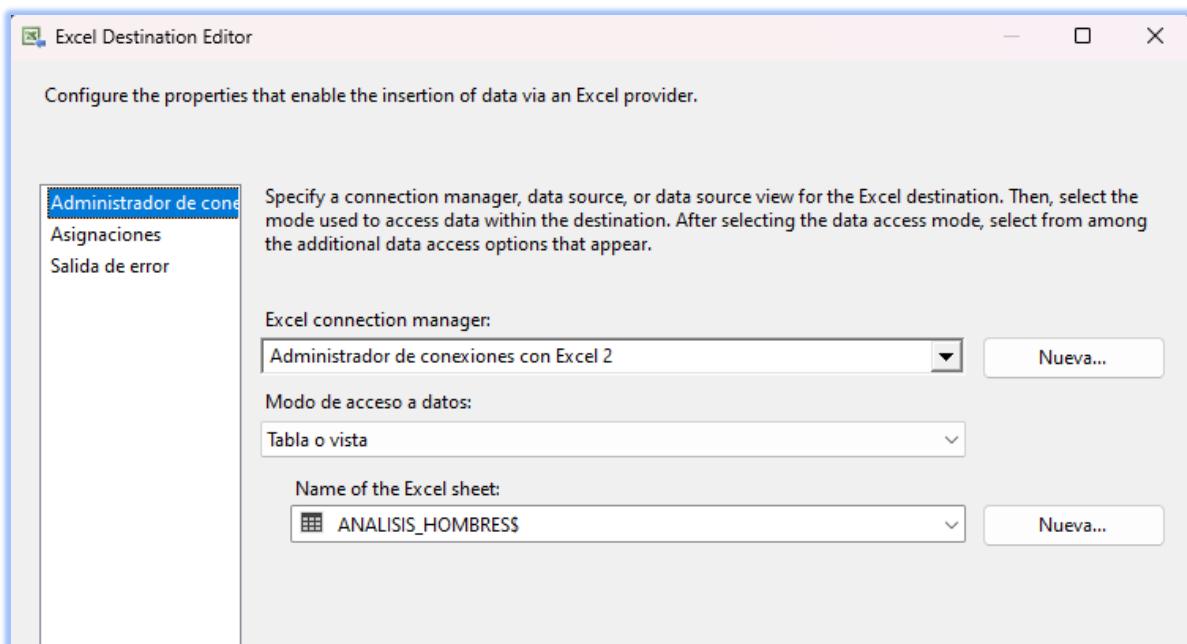
Configure the properties used to perform group by operations and to calculate aggregate values. Optionally, apply comparison options to the operation. To configure multiple group by operations, click Advanced.

Advanced

Columnas de entrada...

| Input Column | Output Alias | Operation | Compa |
|-------------------|-------------------|-----------|-------|
| SexoCadena | Derived Column 1 | Group by | |
| Copy of ingresos | Copy of ingresos | Sum | |
| Copy of PromVisit | Copy of PromVisit | Maximum | |
| Copy of Sauna | Copy of Sauna | Sum | |
| Copy of Masaje | Copy of Masaje | Sum | |
| Copy of Hidro | Copy of Hidro | Sum | |
| Copy of Yoga | Copy of Yoga | Sum | |
| PromEdad | PromEdad | Average | |
| | | | |

5.6 Exportar a Excel por Agrupamiento Masculino



Reporte

The screenshot shows a Microsoft Excel spreadsheet titled "ANALISIS.xls [Modo de compatibilidad] - Excel". The data is organized into columns A through H:

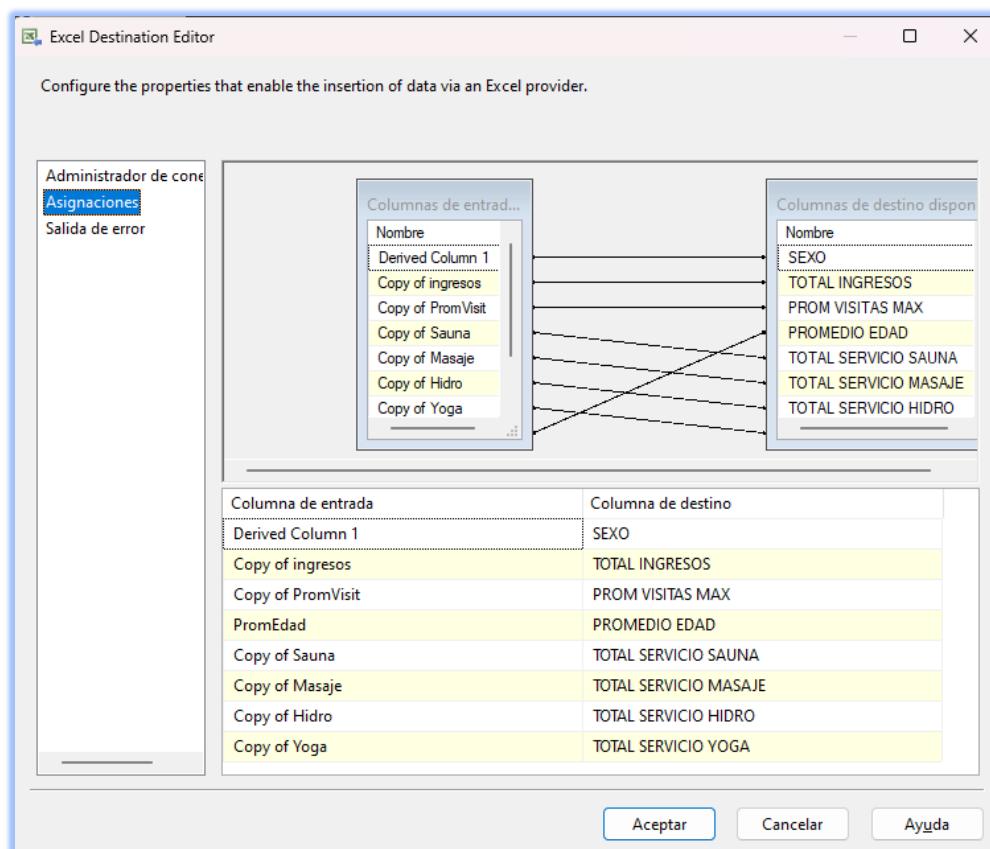
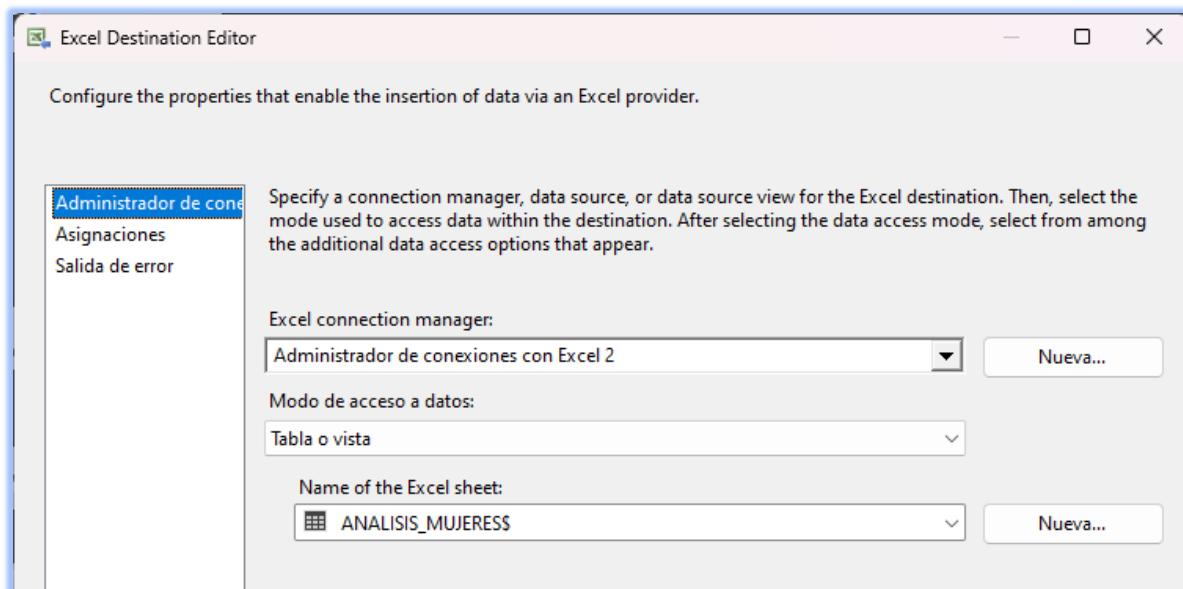
| | A | B | C | D | E | F | G | H |
|---|-----------|----------------|------------------|---------------|----------------------|-----------------------|----------------------|---------------------|
| 1 | SEXO | TOTAL INGRESOS | PROM VISITAS MAX | PROMEDIO EDAD | TOTAL SERVICIO SAUNA | TOTAL SERVICIO MASAJE | TOTAL SERVICIO HIDRO | TOTAL SERVICIO YOGA |
| 2 | MASCULINO | 682061.69 | | 41.946341 | 155 | 223 | 145 | 216 |

5.7 Objeto de Agregado para agrupar personas de sexo Femenino

The screenshot shows the "Aggregate Transformation Editor" dialog box. It has two tabs: "Aggregations" (selected) and "Advanced". The "Aggregations" tab contains the following configuration:

- Input Column:** SexoCadena
- Output Alias:** Derived Column 1
- Operation:** Group by

| Input Column | Output Alias | Operation | Cor |
|-------------------|-------------------|-----------|-----|
| Copy of ingresos | Copy of ingresos | Sum | |
| Copy of PromVisit | Copy of PromVisit | Maximum | |
| Copy of Sauna | Copy of Sauna | Sum | |
| Copy of Masaje | Copy of Masaje | Sum | |
| Copy of Hidro | Copy of Hidro | Sum | |
| Copy of Yoga | Copy of Yoga | Sum | |
| PromEdad | PromEdad | Average | |



Reporte

The screenshot shows a Microsoft Excel spreadsheet titled "ANALISIS.xls [Modo de compatibilidad] - Excel". The data is organized into columns A through H, with headers in row 1 and data in rows 2 and 3. The data includes total income, average age, and counts for various services across different gender categories.

| SEXO | TOTAL INGRESOS | PROM VISITAS MAX | PROMEDIO EDAD | TOTAL SERVICIO SAUNA | TOTAL SERVICIO MASAJE | TOTAL SERVICIO HIDRO | TOTAL SERVICIO YOGA |
|----------|----------------|------------------|---------------|----------------------|-----------------------|----------------------|---------------------|
| FEMENINO | 581237.54 | 7 | 43.108823 | 168 | 185 | 143 | 159 |
| | | | | | | | |

6. Análisis de resultados de datos

The screenshot shows a SQL Server Management Studio (SSMS) window. The top half displays an SQL query that uses a CASE statement with WHEN clauses to map gender values (0, 1, and others) to descriptive strings ('Masculino', 'Femenino', and 'Desconocido'). The query also calculates various aggregate statistics (SUM, MAX, AVG) for each gender group. The bottom half shows the execution results in a grid, which includes the gender description and the corresponding calculated values.

```
SELECT
CASE
    WHEN Sexo = 0 THEN 'Masculino'
    WHEN Sexo = 1 THEN 'Femenino'
    ELSE 'Desconocido' -- Opcional: en caso de que haya otros valores distintos a 0 y 1
END AS SexoDescripcion,
SUM(ingresos) AS TotalIngresos,
MAX(PromVisit) AS PromMaxVisit,
AVG(edad) AS PromedioEdad,
SUM(Sauna) AS TotalSauna,
SUM(Masaje) AS TotalMasaje,
SUM(Hidro) AS TotalHidro,
SUM(Yoga) AS TotalYoga
FROM
    SPA_DIEGO_B
GROUP BY
CASE
    WHEN Sexo = 0 THEN 'Masculino'
    WHEN Sexo = 1 THEN 'Femenino'
    ELSE 'Desconocido' -- Opcional: en caso de que haya otros valores distintos a 0 y 1
END;
```

| | SexoDescripcion | TotalIngresos | PromMaxVisit | PromedioEdad | TotalSauna | TotalMasaje | TotalHidro | TotalYoga |
|---|-----------------|---------------|--------------|--------------|------------|-------------|------------|-----------|
| 1 | Femenino | 581237.54 | 7.00 | 43 | 168 | 185 | 143 | 159 |
| 2 | Masculino | 682061.69 | 7.00 | 41 | 155 | 223 | 145 | 216 |

Al comparar la fidelización de clientes de Sexo Masculino vs Sexo femenino se concluye lo siguiente:

| CONCLUSIONES |
|--|
| Sexo Masculino genero más ingresos, con una diferencia de \$100824.2 más que las Mujeres |
| Promedio Máxima de visitas de ambos sexos es igual 7.0 |
| El promedio de edad con mayor visita los spas esta entre hombres y mujeres mayores a 40 años |
| Las mujeres visitan más los servicios de baños saunas que los hombres |
| Los hombres prefieren los masajes que cualquier otro servicio |
| Los hombres y mujeres toman menos servicio de hidroterapia por lo que deberían de ver mejoras en este servicio |
| Los hombres tienen mayor presencia en los servicios de yoga que las mujeres, por lo que deberían de trabajar para atraer a más mujeres a los servicios de yoga |
| Además, las mujeres toman menos servicios que los hombres en SPA Diego por lo que deben de enfocarse en atraer mujeres y jóvenes ambos sexos |

| | SexoDescripción | TotalIngresos | PromMaxVisit | PromedioEdad | TotalSauna | TotalMasaje | TotalHidro | TotalYoga |
|---|-----------------|---------------|--------------|--------------|------------|-------------|------------|-----------|
| 1 | Femenino | 581237.54 | 7.00 | 43 | 168 | 185 | 143 | 159 |
| 2 | Masculino | 682061.69 | 7.00 | 41 | 155 | 223 | 145 | 216 |

Universidad Don Bosco
Facultad de Ingeniería



Data Warehouse y minería de datos

Tema:

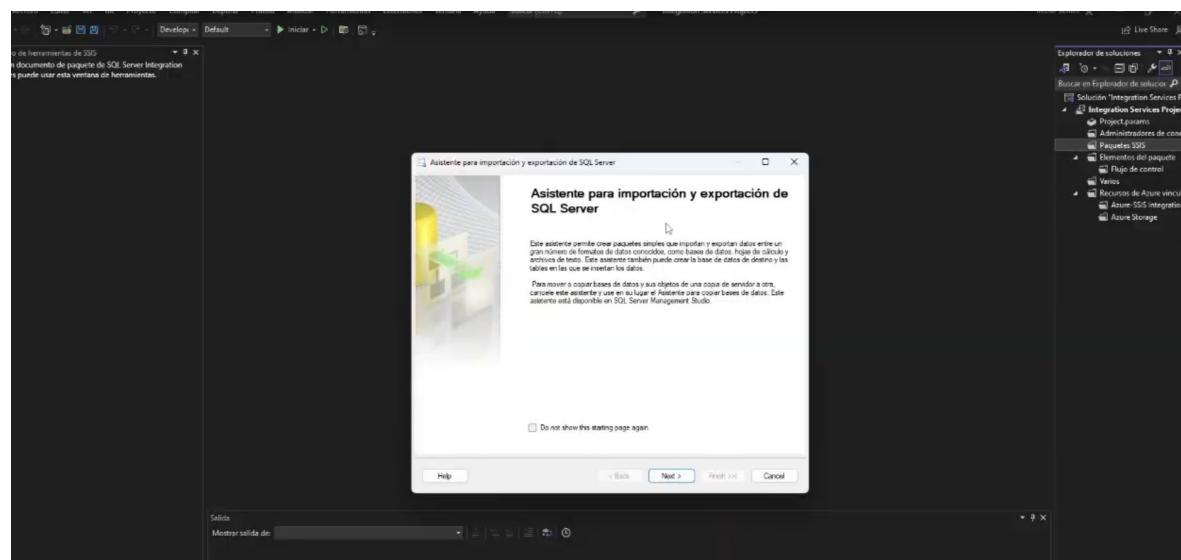
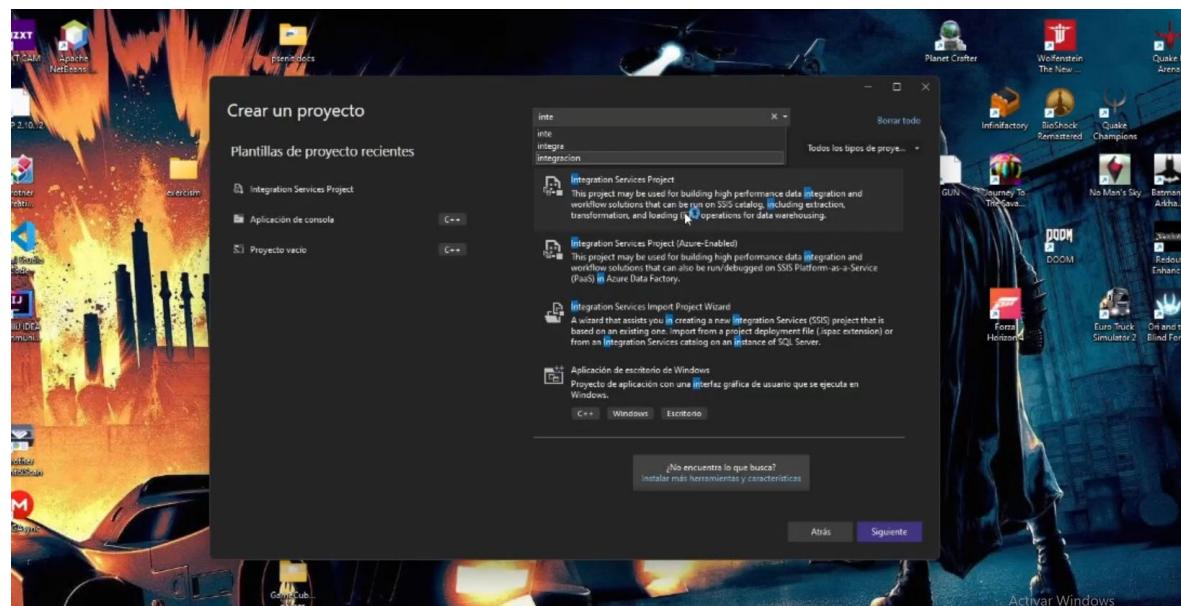
Desafío 1, Ejercicio 5

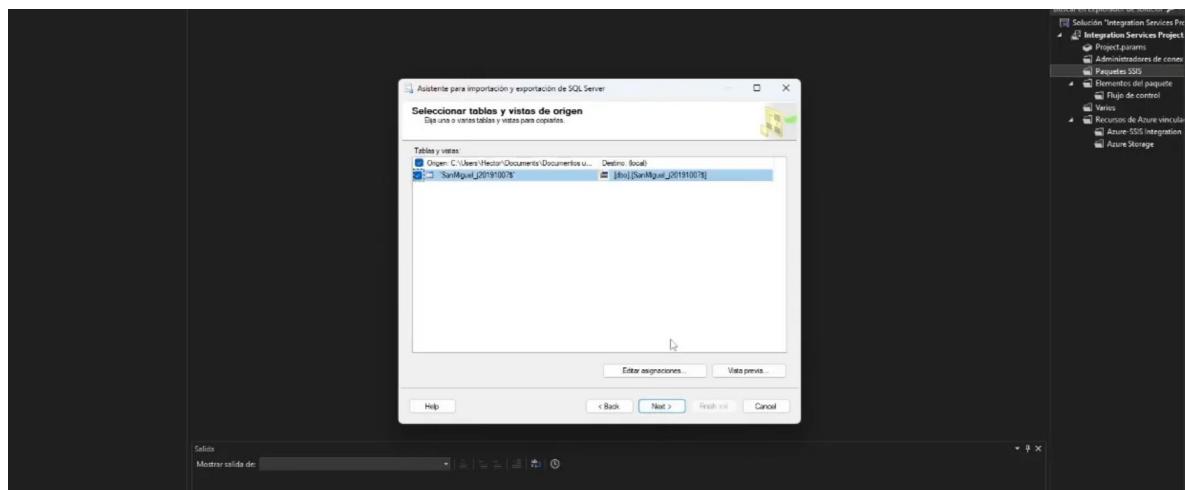
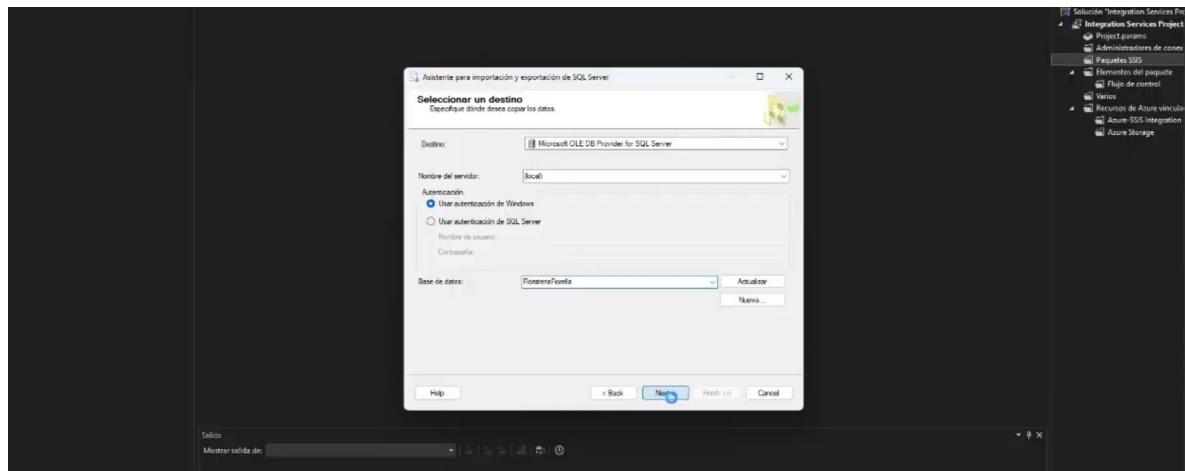
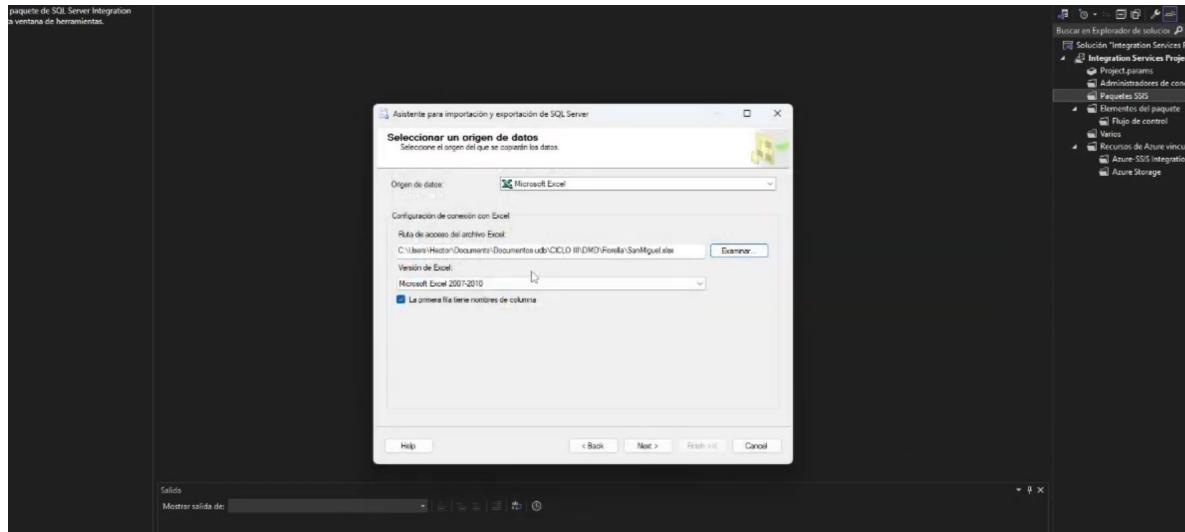
Nombre: Hector José Márquez Chicas

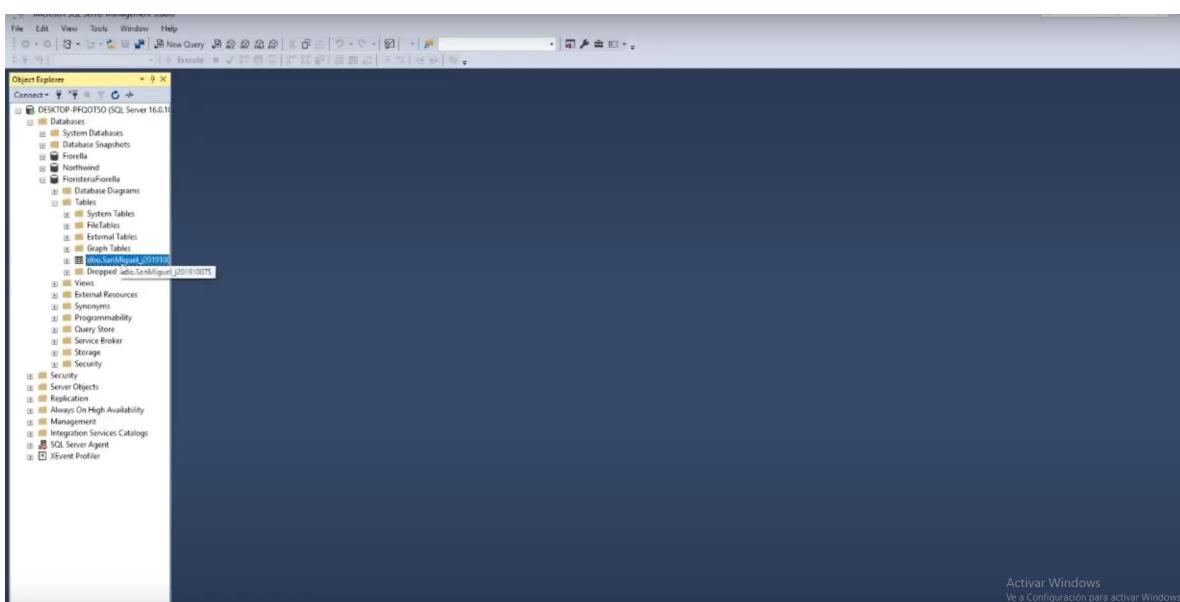
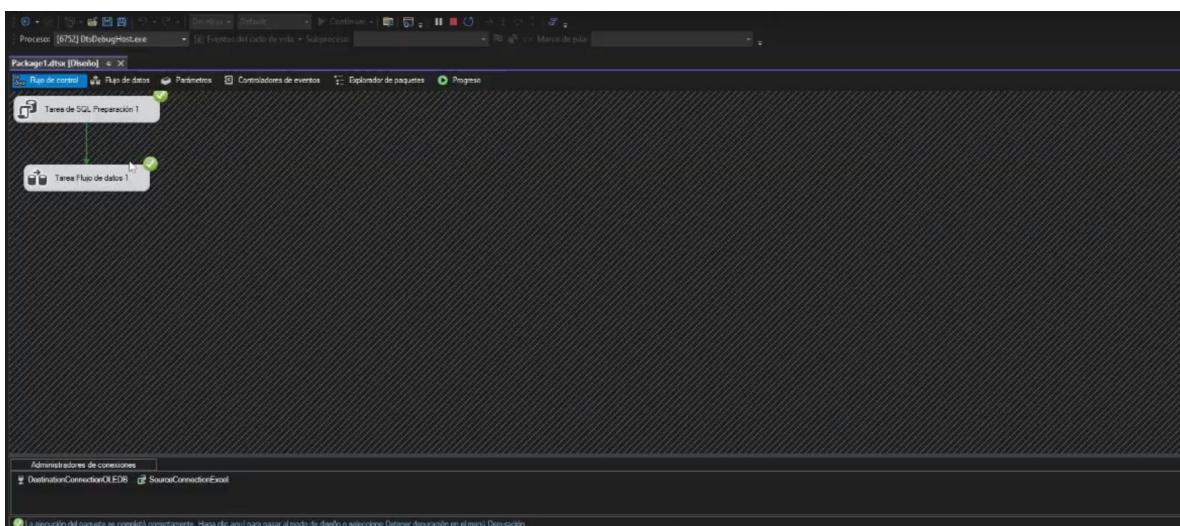
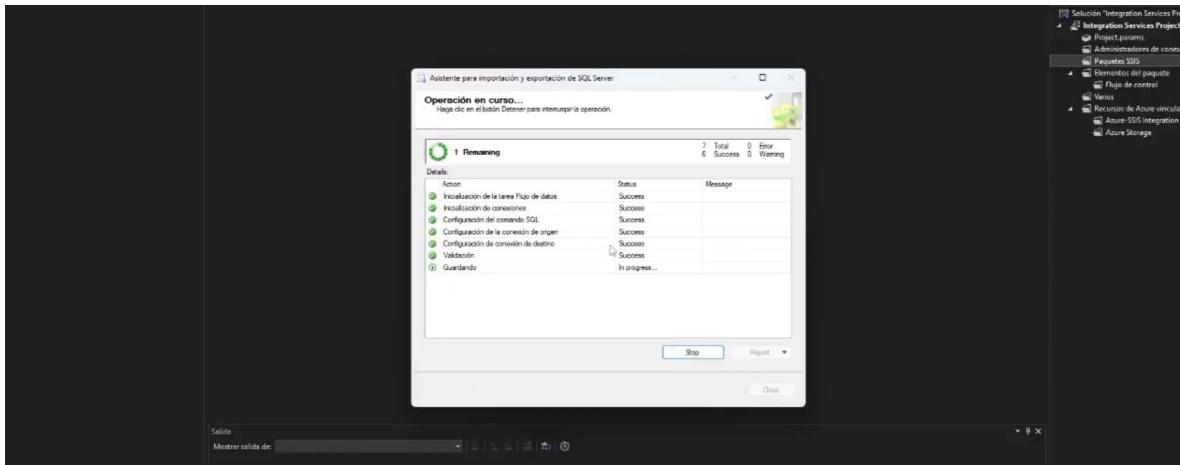
Carnet: MC233291

5. La Floristería "Fiorella" quiere saber cómo se compran sus productos, y tiene la data de tres departamentos del país, por lo cual les pide su opinión sobre qué productos sobresalen, que combinaciones son mejores y quieren este estudio por departamento y también por país

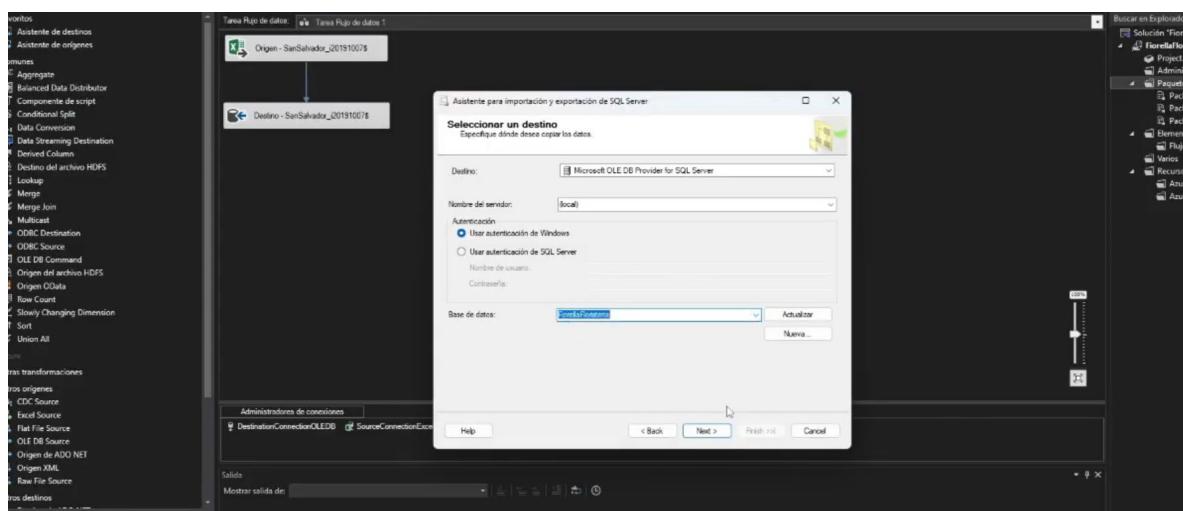
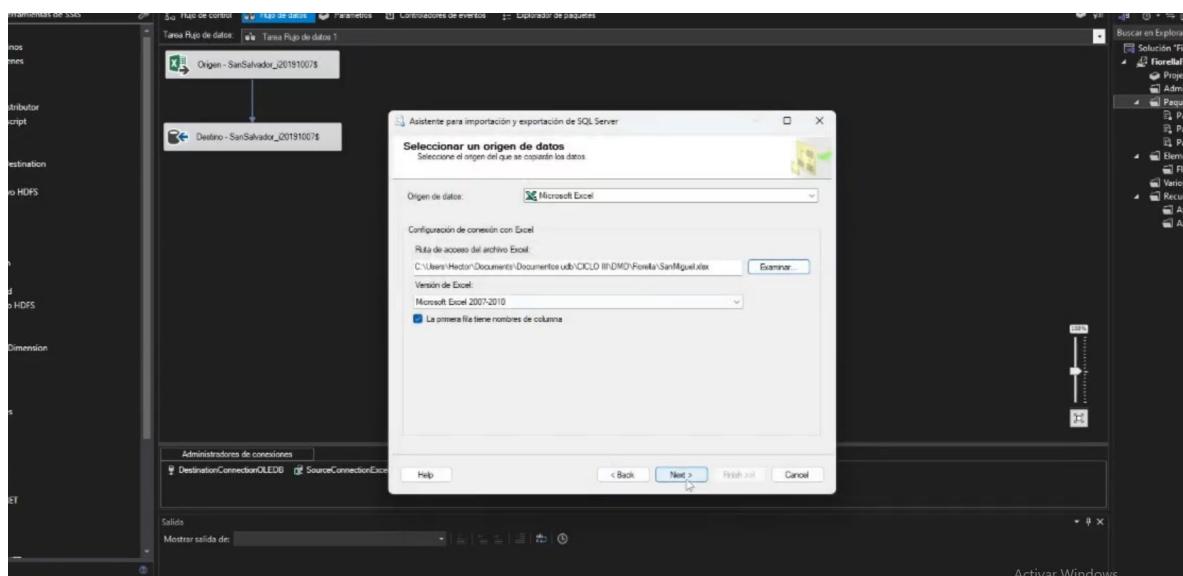
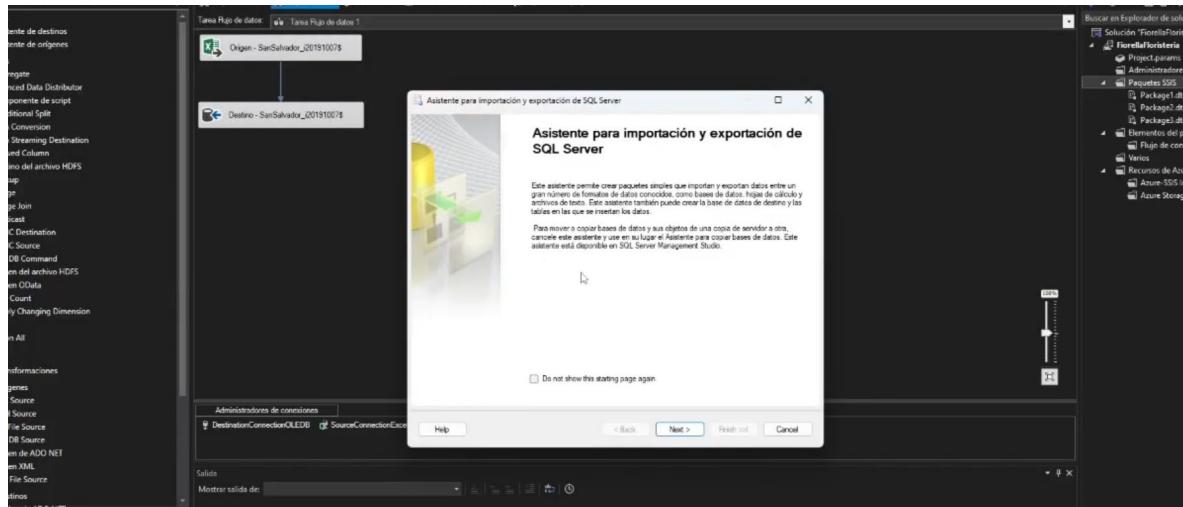
ETL desde archivo Excel a una base de datos SQL

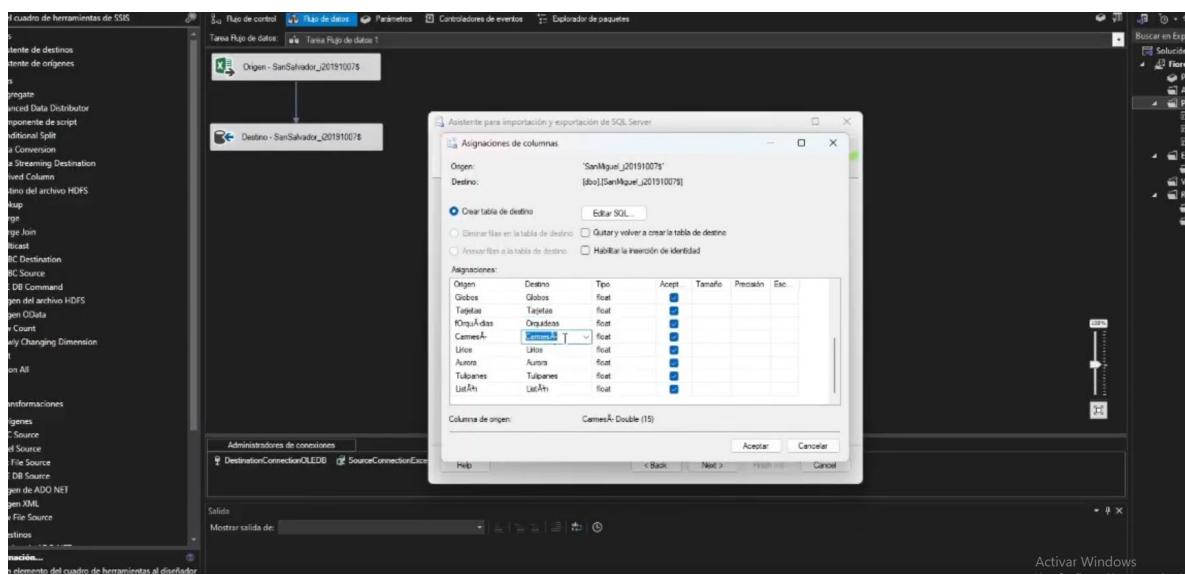
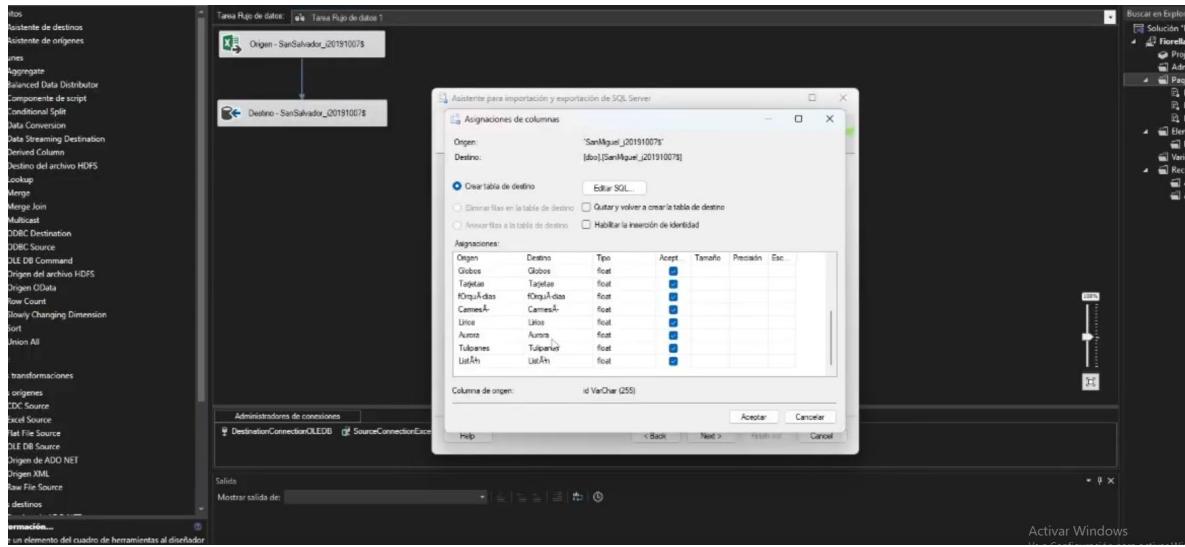


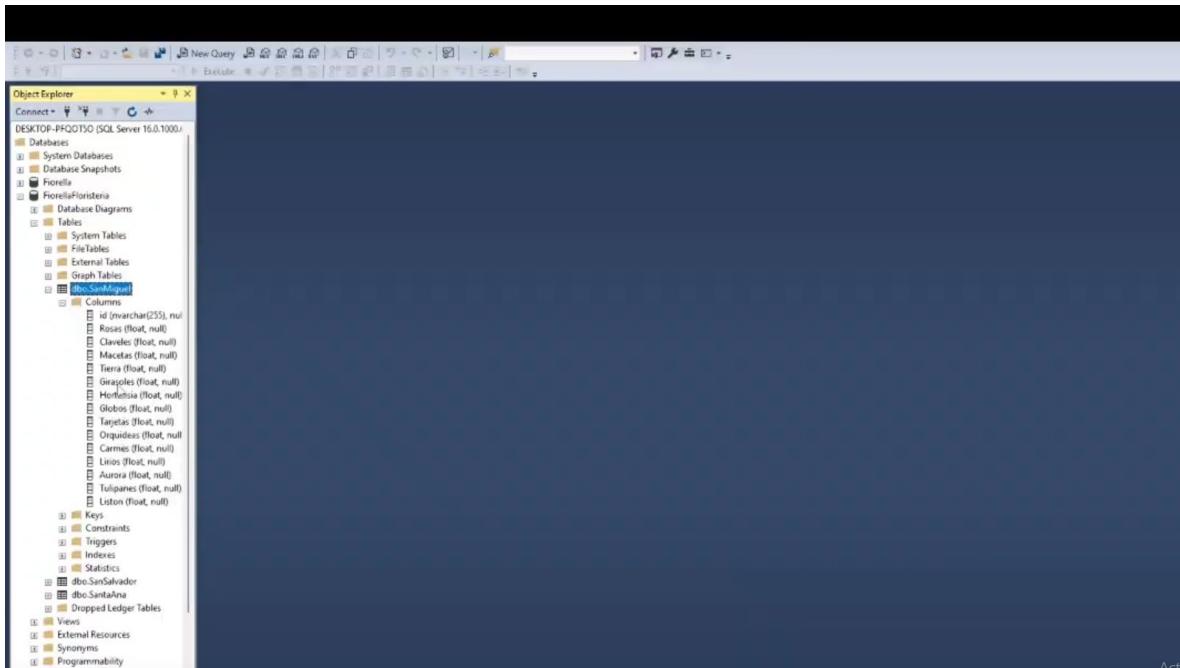




Corrección de los nombres de las tablas en el archivo destino:







Queries para obtener la información requerida según el ejercicio:

```

Object Explorer  SQLQueryLog - D...-00750-Hector (70)  SQLQueryLog - D...-00750-Hector (62)
Connect  Connect  DESKTOP-PFQOTSO (SQL Server 16.0.1000)
  Databases
  System Databases
  Database Snapshots
  Forella
  ForellaFloristeria
    Database Diagrams
    Tables
      System Tables
      FileTables
      External Tables
      Graph Tables
      Columns
        id (nvarchar(255), null)
        Rosas (float, null)
        Claveles (float, null)
        Macetas (float, null)
        Tierra (float, null)
        Flores (float, null)
        Hortensias (float, null)
        Globos (float, null)
        Tarjetas (float, null)
        Orquideas (float, null)
        Carnes (float, null)
        Lirios (float, null)
        Aurora (float, null)
        Tulipanes (float, null)
        Loston (float, null)
      Keys
      Constraints
      Triggers
      Indexes
      Statistics
    dbo.SanSalvador
    dbo.SantaAna
      Dropped Ledger Tables
    Views
    External Resources
    Synonyms
    Programmability

Object Explorer  SQLQueryLog - D...-00750-Hector (70)  SQLQueryLog - D...-00750-Hector (62)
Connect  Connect  DESKTOP-PFQOTSO (SQL Server 16.0.1000)
  Databases
  System Databases
  Database Snapshots
  Forella
  ForellaFloristeria
    Database Diagrams
    Tables
      System Tables
      FileTables
      External Tables
      Graph Tables
      Columns
        id (nvarchar(255), null)
        Rosas (float, null)
        Claveles (float, null)
        Macetas (float, null)
        Tierra (float, null)
        Flores (float, null)
        Hortensias (float, null)
        Globos (float, null)
        Tarjetas (float, null)
        Orquideas (float, null)
        Carnes (float, null)
        Lirios (float, null)
        Aurora (float, null)
        Tulipanes (float, null)
        Loston (float, null)
      Keys
      Constraints
      Triggers
      Indexes
      Statistics
    dbo.SanSalvador
    dbo.SantaAna
      Dropped Ledger Tables
    Views
    External Resources
    Synonyms
    Programmability

SELECT 'San Salvador' AS Departamento, *
FROM SanSalvador
UNION ALL
SELECT 'Santa Ana' AS Departamento, *
FROM SantaAna;
SELECT
  Departamento,
  SUM(Rosas) AS Total_Rosas,
  SUM(Claveles) AS Total_Claveles,
  SUM(Macetas) AS Total_Macetas,
  SUM(Tierra) AS Total_Tierra,
  SUM(Girasoles) AS Total_Girasoles,
  SUM(Flores) AS Total_Flores,
  SUM(Hortensias) AS Total_Hortensias,
  SUM(Globos) AS Total_Globos,
  SUM(Tarjetas) AS Total_Tarjetas,
  SUM(Orquideas) AS Total_Orquideas,
  SUM(Carnes) AS Total_Carnes,
  SUM(Lirios) AS Total_Lirios,
  SUM(Aurora) AS Total_Aurora,
  SUM(Tulipanes) AS Total_Tulipanes,
  SUM(Loston) AS Total_Loston
FROM [dbo].[Sales]
SELECT 'San Miguel' AS Departamento, *
FROM SanMiguel

```

| Departamento | Total_Bosque | Total_Ciervos | Total_Macetas | Total_Tierra | Total_Girasoles | Total_Hortensia | Total_Globos | Total_Tarjetas | Total_Orquideas | Total_Carnes | Total_Lirios | Total_Aurora | Total_Tulipanes | Total_Loston |
|--------------|--------------|---------------|---------------|--------------|-----------------|-----------------|--------------|----------------|-----------------|--------------|--------------|--------------|-----------------|--------------|
| Santa Ana | 176 | 246 | 245 | 236 | 265 | 243 | 154 | 252 | 259 | 236 | 270 | 260 | 247 | 136 |
| San Miguel | 157 | 137 | 141 | 141 | 150 | 157 | 151 | 143 | 158 | 158 | 160 | 160 | 149 | 149 |
| San Salvador | 672 | 350 | 392 | 368 | 371 | 374 | 587 | 384 | 380 | 353 | 365 | 384 | 357 | 690 |

```

Object Explorer  SQLQueryLog - D...-00750-Hector (70)  SQLQueryLog - D...-00750-Hector (62)
Connect  Connect  DESKTOP-PFQOTSO (SQL Server 16.0.1000)
  Databases
  System Databases
  Database Snapshots
  Forella
  ForellaFloristeria
    Database Diagrams
    Tables
      System Tables
      FileTables
      External Tables
      Graph Tables
      Columns
        id (nvarchar(255), null)
        Rosas (float, null)
        Claveles (float, null)
        Macetas (float, null)
        Tierra (float, null)
        Flores (float, null)
        Hortensias (float, null)
        Globos (float, null)
        Tarjetas (float, null)
        Orquideas (float, null)
        Carnes (float, null)
        Lirios (float, null)
        Aurora (float, null)
        Tulipanes (float, null)
        Loston (float, null)
      Keys
      Constraints
      Triggers
      Indexes
      Statistics
    dbo.SanSalvador
    dbo.SantaAna
      Dropped Ledger Tables
    Views
    External Resources
    Synonyms
    Programmability

Object Explorer  SQLQueryLog - D...-00750-Hector (70)  SQLQueryLog - D...-00750-Hector (62)
Connect  Connect  DESKTOP-PFQOTSO (SQL Server 16.0.1000)
  Databases
  System Databases
  Database Snapshots
  Forella
  ForellaFloristeria
    Database Diagrams
    Tables
      System Tables
      FileTables
      External Tables
      Graph Tables
      Columns
        id (nvarchar(255), null)
        Rosas (float, null)
        Claveles (float, null)
        Macetas (float, null)
        Tierra (float, null)
        Flores (float, null)
        Hortensias (float, null)
        Globos (float, null)
        Tarjetas (float, null)
        Orquideas (float, null)
        Carnes (float, null)
        Lirios (float, null)
        Aurora (float, null)
        Tulipanes (float, null)
        Loston (float, null)
      Keys
      Constraints
      Triggers
      Indexes
      Statistics
    dbo.SanSalvador
    dbo.SantaAna
      Dropped Ledger Tables
    Views
    External Resources
    Synonyms
    Programmability

SELECT 'San Salvador' AS Departamento, * FROM SanSalvador
UNION ALL
SELECT 'Santa Ana' AS Departamento, * FROM SantaAna;
SELECT
  Departamento,
  SUM(Rosas) AS Total_Rosas,
  SUM(Claveles) AS Total_Claveles,
  SUM(Macetas) AS Total_Macetas,
  SUM(Tierra) AS Total_Tierra,
  SUM(Girasoles) AS Total_Girasoles,
  SUM(Hortensias) AS Total_Hortensias,
  SUM(Globos) AS Total_Globos,
  SUM(Tarjetas) AS Total_Tarjetas,
  SUM(Orquideas) AS Total_Orquideas,
  SUM(Carnes) AS Total_Carnes,
  SUM(Lirios) AS Total_Lirios,
  SUM(Aurora) AS Total_Aurora,
  SUM(Tulipanes) AS Total_Tulipanes,
  SUM(Loston) AS Total_Loston
FROM [dbo].[Sales]
SELECT 'San Miguel' AS Departamento, * FROM SanMiguel
UNION ALL
SELECT 'Santa Ana' AS Departamento, * FROM SantaAna;
UNION ALL
SELECT 'San Salvador' AS Departamento, * FROM SanSalvador
UNION ALL
SELECT 'Forella' AS Departamento, * FROM Forella;

SELECT
  CASE WHEN Rosas > 0 AND Claveles > 0 THEN 1 ELSE 0 END AS Rosas_Claveles,
  CASE WHEN Rosas > 0 AND Globos > 0 THEN 1 ELSE 0 END AS Rosas_Globos,
  CASE WHEN Rosas > 0 AND Tarjetas > 0 THEN 1 ELSE 0 END AS Rosas_Tarjetas,
  CASE WHEN Rosas > 0 AND Macetas > 0 THEN 1 ELSE 0 END AS Rosas_Macetas,
  CASE WHEN Claveles > 0 AND Girasoles > 0 THEN 1 ELSE 0 END AS Claveles_Girasoles,
  CASE WHEN Claveles > 0 AND Globos > 0 THEN 1 ELSE 0 END AS Claveles_Globos,
  CASE WHEN Girasoles > 0 AND Orquideas > 0 THEN 1 ELSE 0 END AS Girasoles_Orquideas
FROM (
  SELECT * FROM SanMiguel
  UNION ALL
  SELECT * FROM SantaAna
  UNION ALL
  SELECT * FROM SanSalvador
  UNION ALL
  SELECT * FROM Forella
)
```

Activar Windows

External Tables
Graph Tables
dbo.SanMiguel
Columns
14 in varchar(55), n
Roses (float, null)
Claveles (float, null)
Macetas (float, null)
Ternia (float, null)
Globoidea (float, null)
Hortensia (float, null)
Globos (float, null)
Orquideas (float, null)
Carmen (float, null)
Lirio (float, null)
Aurora (float, null)
Tulipanes (float, null)
Lisianthus (float, null)
Keys
Constraints
Triggers
Indexes
Statistics
dbo.SanLopez
dbo.SanSalvador
dbo.SanAndrea
Views
External Resources
System Tables
Programmability
Query Store
Service Broker

```

UNION ALL
SELECT * FROM Santalina
) AS Consolidado;

--SELECT
--    SUM(CASE WHEN Rosas > 0 AND Claveles > 0 THEN 1 ELSE 0 END) AS Rosas_Claveles,
--    SUM(CASE WHEN Rosas > 0 AND Globos > 0 THEN 1 ELSE 0 END) AS Rosas_Globos,
--    SUM(CASE WHEN Rosas > 0 AND Tarjetas > 0 THEN 1 ELSE 0 END) AS Rosas_Tarjetas,
--    SUM(CASE WHEN Rosas > 0 AND Macetas > 0 THEN 1 ELSE 0 END) AS Rosas_Macetas,
--    SUM(CASE WHEN Rosas > 0 AND Ternia > 0 THEN 1 ELSE 0 END) AS Rosas_Ternia,
--    SUM(CASE WHEN Claveles > 0 AND Globos > 0 THEN 1 ELSE 0 END) AS Claveles_Globos,
--    SUM(CASE WHEN Claveles > 0 AND Tarjetas > 0 THEN 1 ELSE 0 END) AS Claveles_Tarjetas,
--    SUM(CASE WHEN Claveles > 0 AND Macetas > 0 THEN 1 ELSE 0 END) AS Claveles_Macetas,
--    SUM(CASE WHEN Globos > 0 AND Tarjetas > 0 THEN 1 ELSE 0 END) AS Globos_Tarjetas,
--    SUM(CASE WHEN Globos > 0 AND Macetas > 0 THEN 1 ELSE 0 END) AS Globos_Macetas,
--    SUM(CASE WHEN Tarjetas > 0 AND Macetas > 0 THEN 1 ELSE 0 END) AS Tarjetas_Macetas
--FROM (
--    SELECT * FROM SanLopez
--) UNION ALL
--SELECT * FROM SanSalvador
--UNION ALL
--SELECT * FROM Santalina
--) AS Consolidado;

```

Results Messages

| | Rosas_Claveles | Rosas_Globos | Rosas_Tarjetas | Rosas_Macetas | Claveles_Globos | Globos_Tarjetas |
|---|----------------|--------------|----------------|---------------|-----------------|-----------------|
| 1 | 443 | 513 | 483 | 497 | 384 | 410 |

Activar Windows
Ve a Configuración para activar Windows.

Ejercicio 6:

La telefonía “FioDio” solicita realizar un ETL que exporte una base de datos de Mysql y SQL Server, al final el destino serán dos archivos de Excel en donde en un archivo estarán los clientes preferenciales y ejecutivos y en el segundo los de gobierno y turista, adicional en los archivos de Excel se deberá crear un campo código de país, que se llenará sustraendo los dos primeros caracteres de código cliente, ver imagen a continuación.

Para la ejecución del programa se estará utilizando

Un tipo de archivo plano CSV, llamado chatgpt, el cual se pide se envia a una base de datos para SQL server:

En el cual se procederá a la creación de un programa ETL, en el cual se realizara desde visual studio 2019:

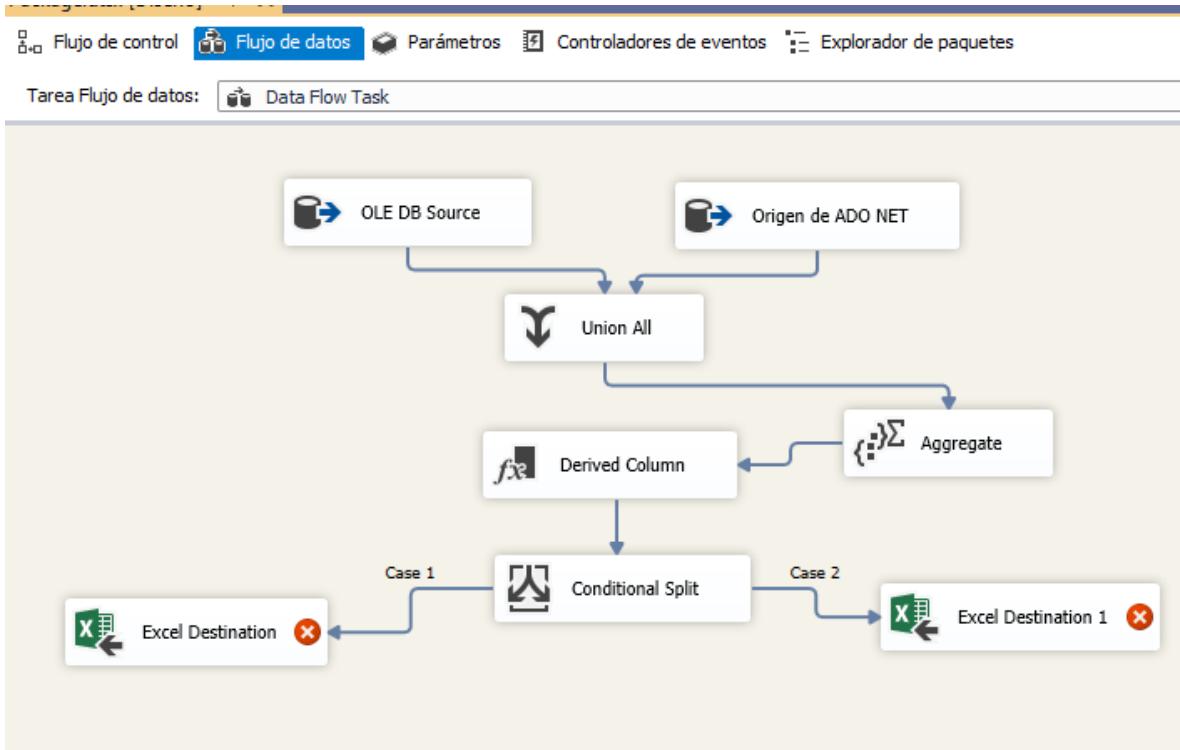
Realizando la creación de este archivo llamado ejercicio6:

En el cual borraremos package1.dtsx que viene por defecto.

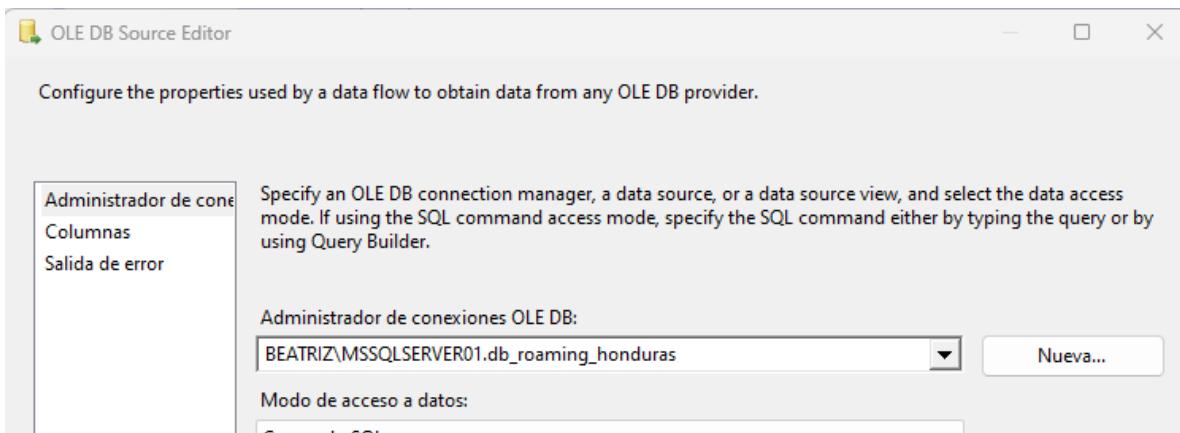
En la tarea flujo de control daremos doble clic en el control de datos y agregaremos los siguientes controles:

| Nombre de control | Cantidad |
|-------------------|----------|
| Ole DB Source | 1 |
| Origen de ADO NET | 1 |
| Union All | 1 |
| Aggregate | 1 |
| Derived Column | 1 |
| Conditional Split | 1 |
| Excel destination | 2 |

Luego de agregar esto se parecerá a este resultado y se realizará al unión de cada uno de ellos::



Luego de este proceso procederemos a realizar las conexiones de las base de datos primeramente con el SQL server, le daremos a nueva para realizar la conexión:

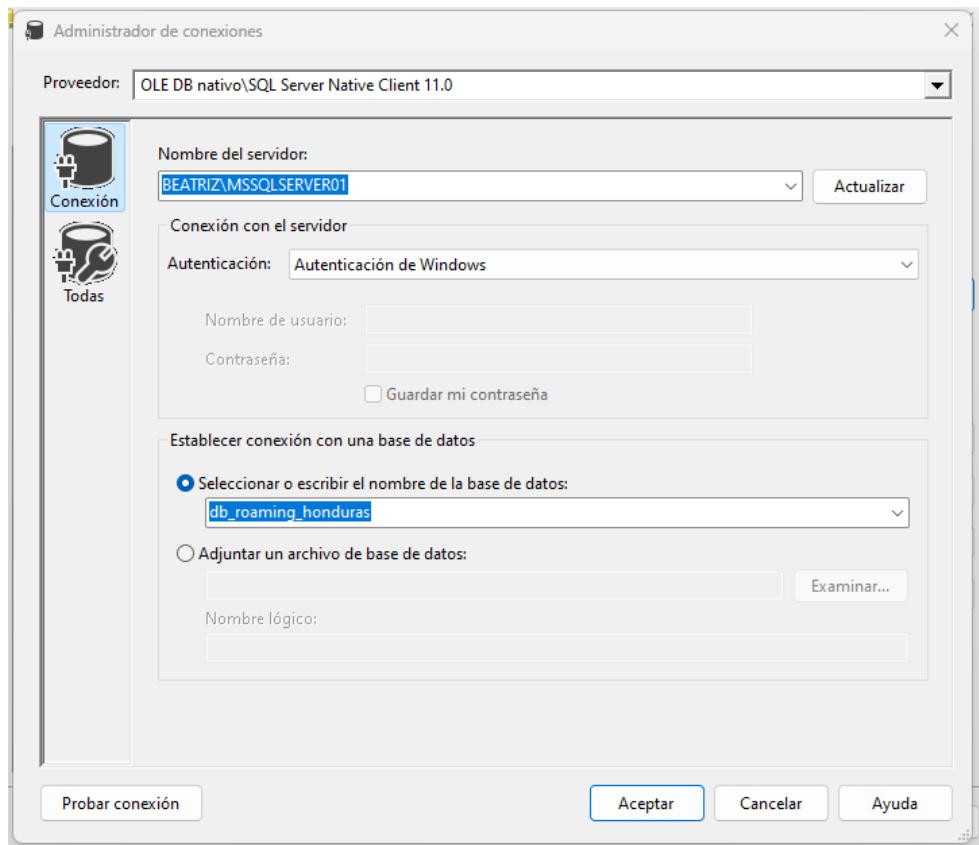


Luego se realizara la conexión a la base de datos que hemos creado con anterioridad:

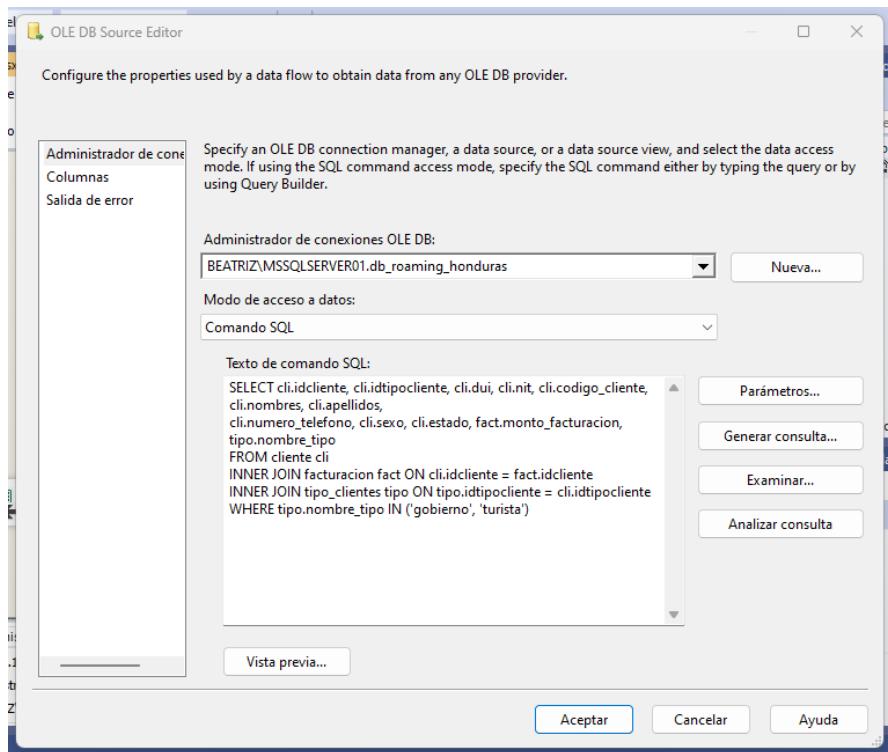
```

create database db_roaming_honduras
GO
USE db_roaming_honduras

```

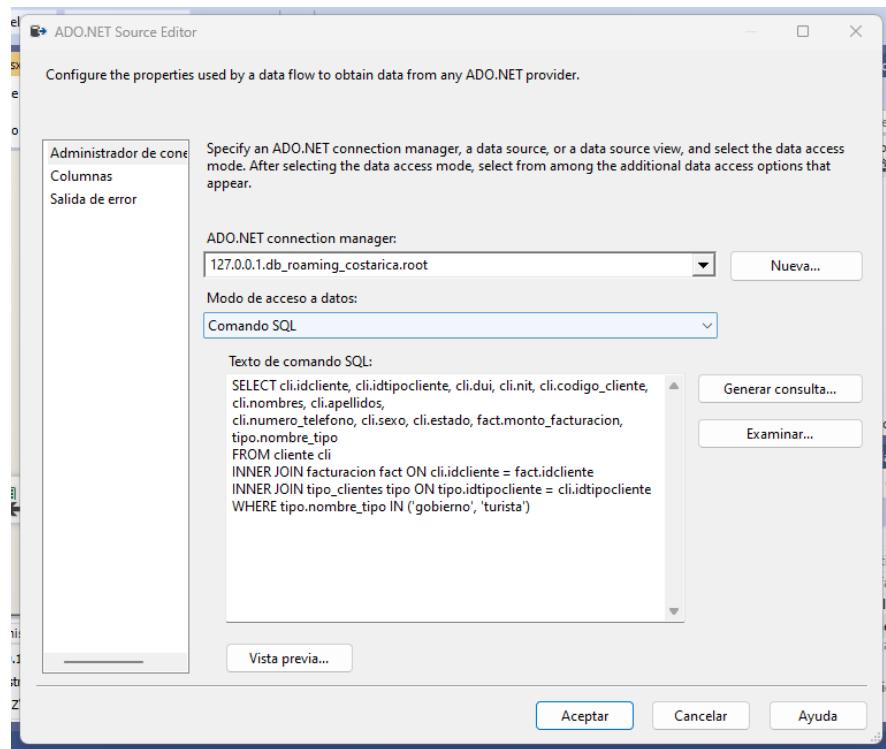


Le damos aceptar a la conexión y Terminar el sql, luego de eso generamos la consulta de todo este proceso de sql

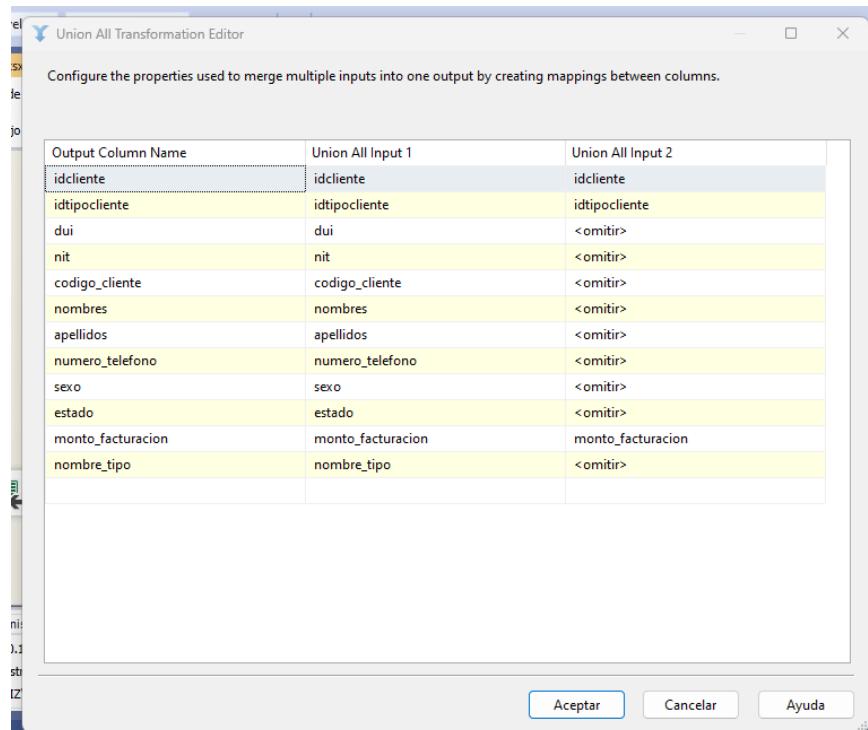


Se realiza el mismo proceso en la conexión de MySQL en Origen de ADO NET, los pasos son iguales a los de SQL server.

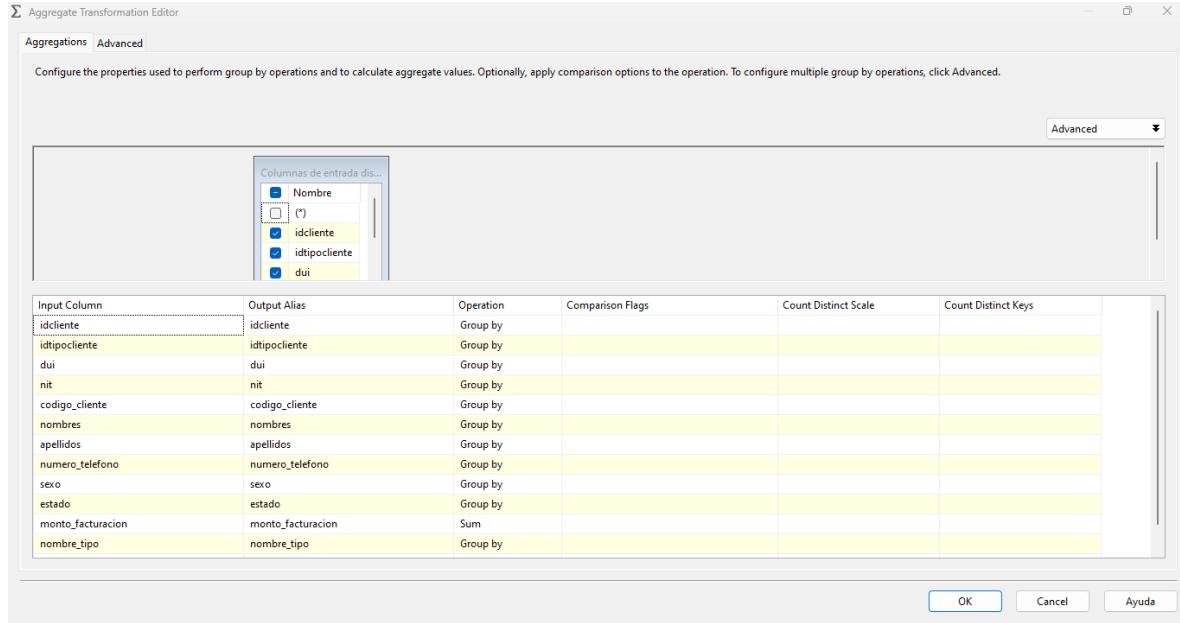
Nos quedara de la siguiente manera:



En el unión All se realizara el proceso de la siguiente manera:

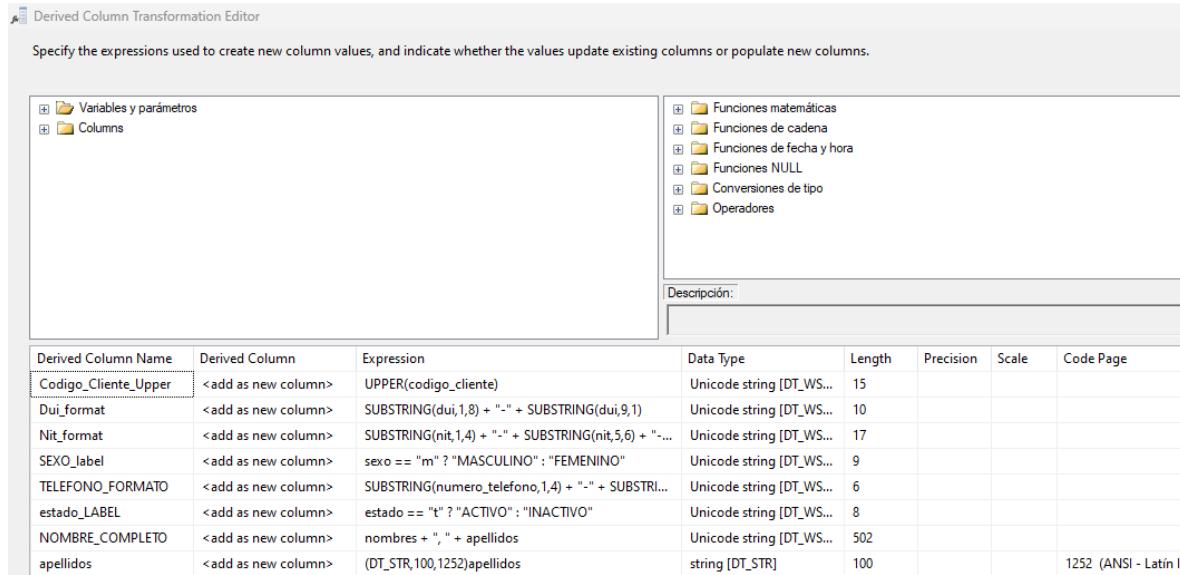


En el Agregate se seleccionara todo y en el Operetation cambiamos para que quede de la siguiente manera:

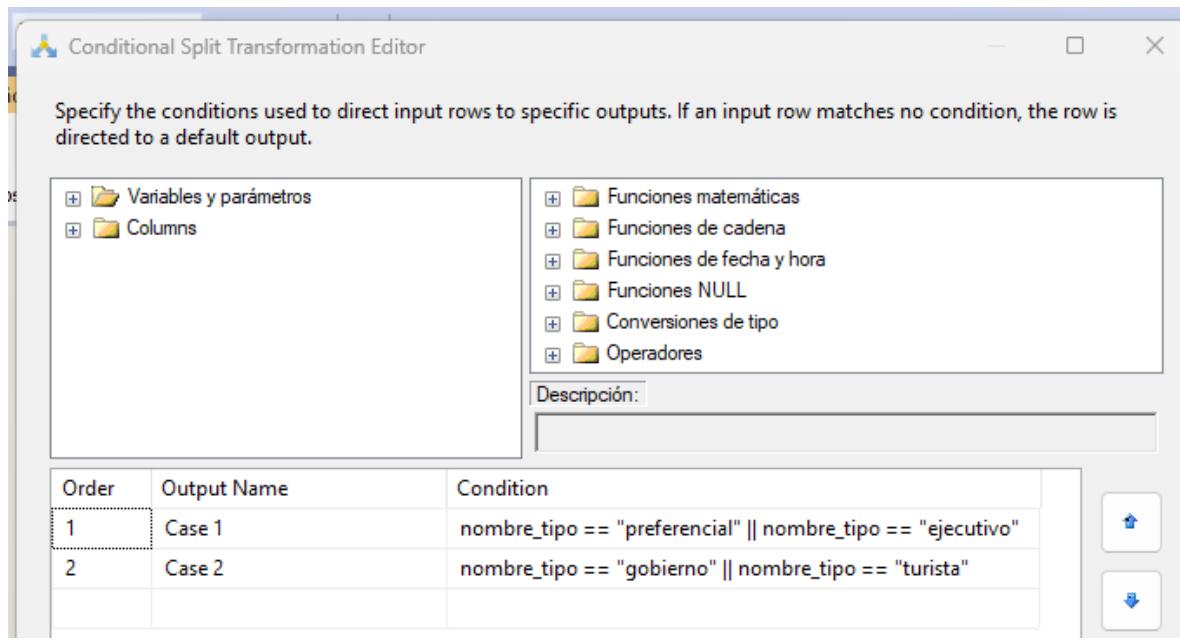


En el Derived Colum se realizara el proceso para que nos quede con las siguientes expresiones para poder cumplir que nuestros datos de ambas bases de datos se junten.

Al realizar la colocaciones de las expresiones nos deberá de quedar de la siguiente manera:

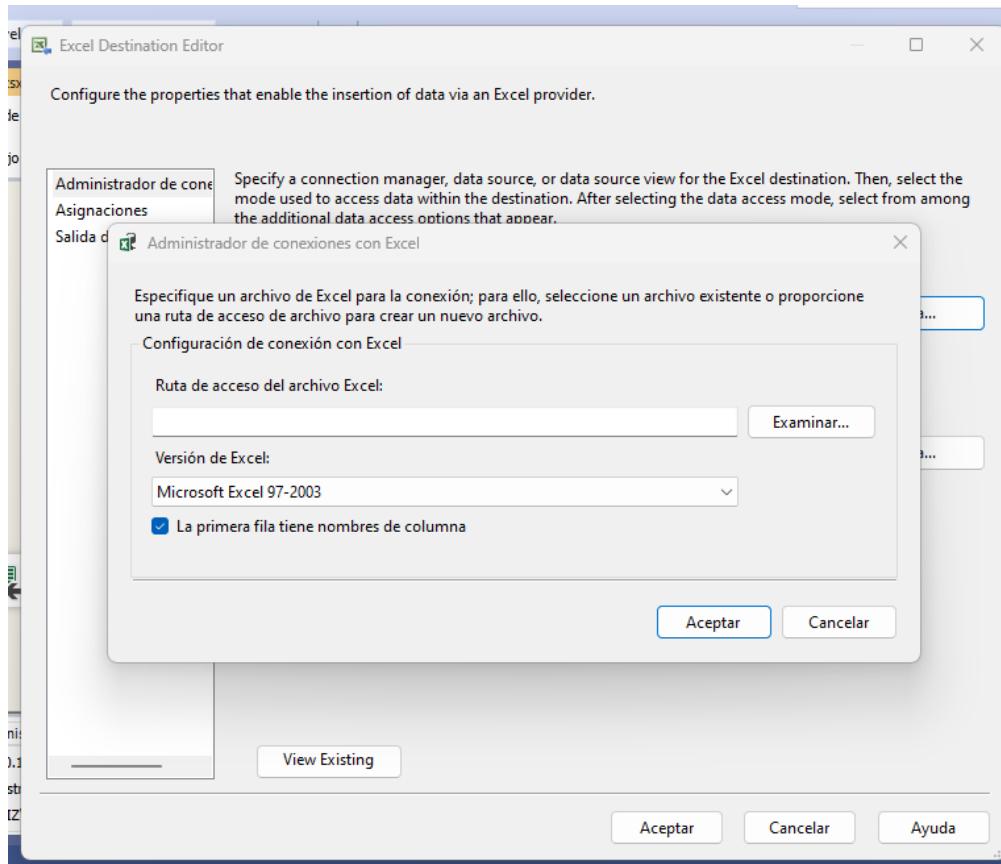


En el conditional Split realizamos el siguiente proceso para las condiciones que se nos pide para la generación de los Excel con los datos solicitados:

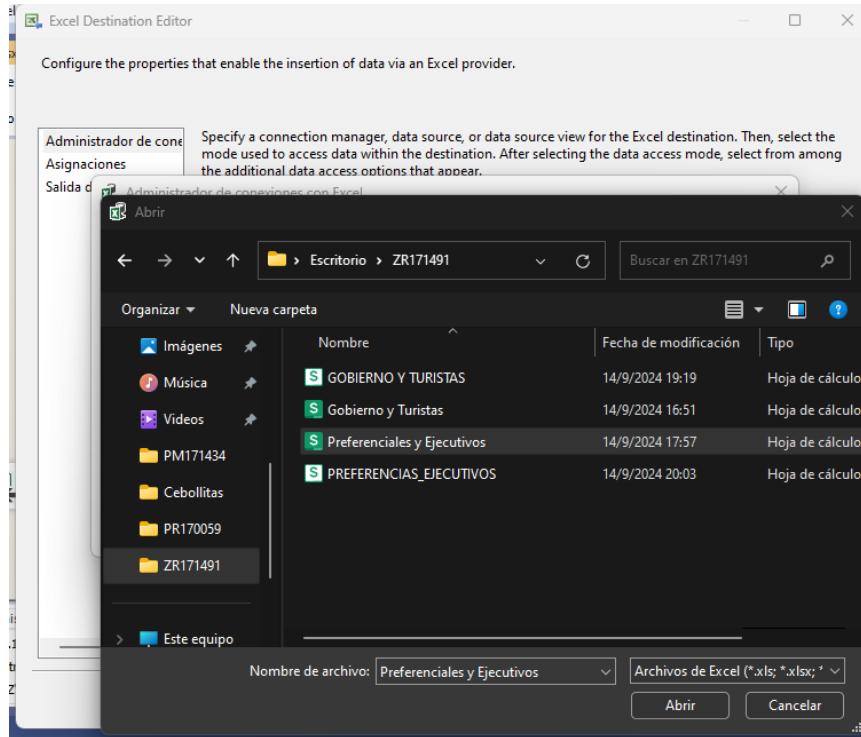


Luego realizamos el proceso en los Excel Destination, el cual se realizara una conexión a los documentos Excel creados se realiza de la siguiente manera:

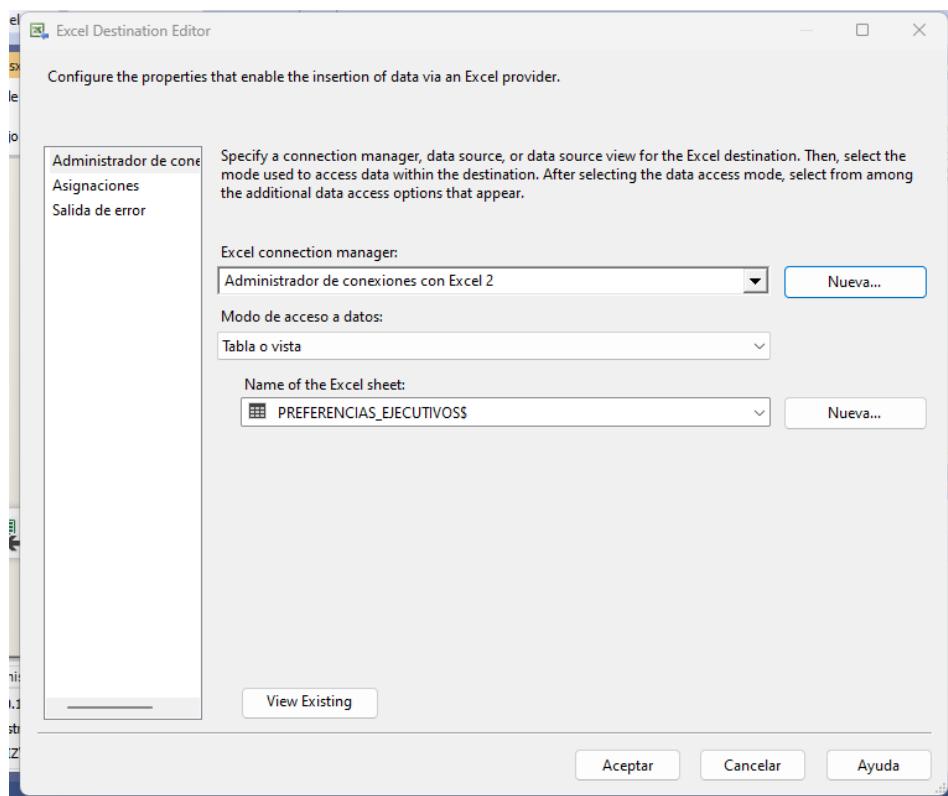
Se examina donde tenemos nuestro documento:



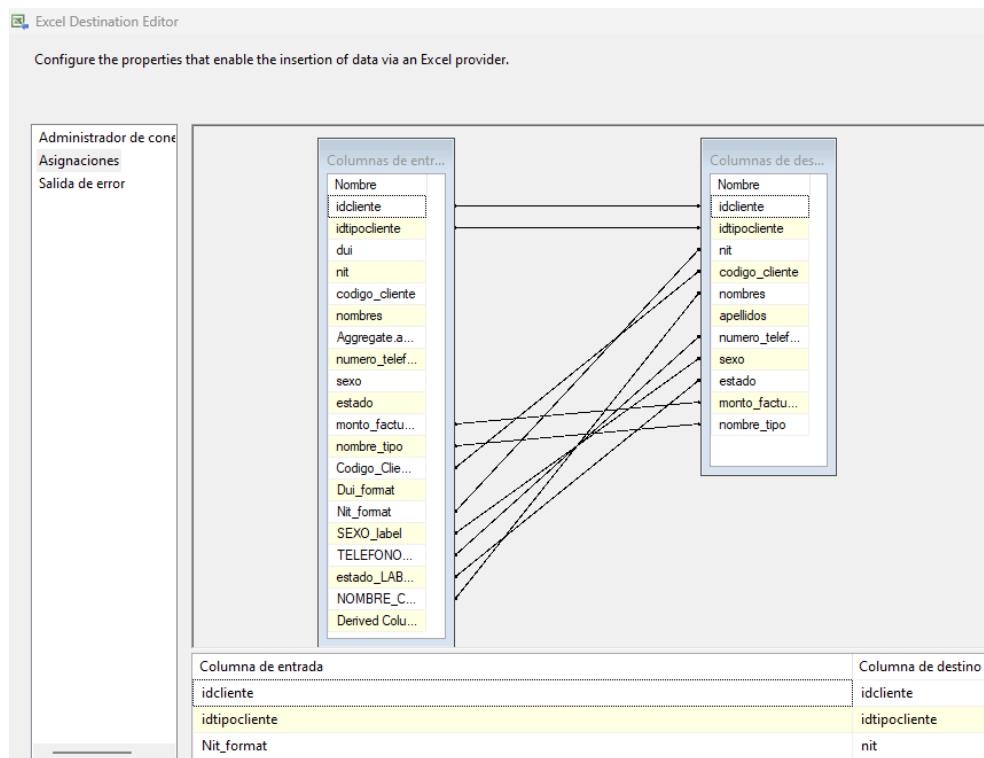
Seleccionamos nuestro documento al cual llegaran los datos que estamos realizando en la filtración:



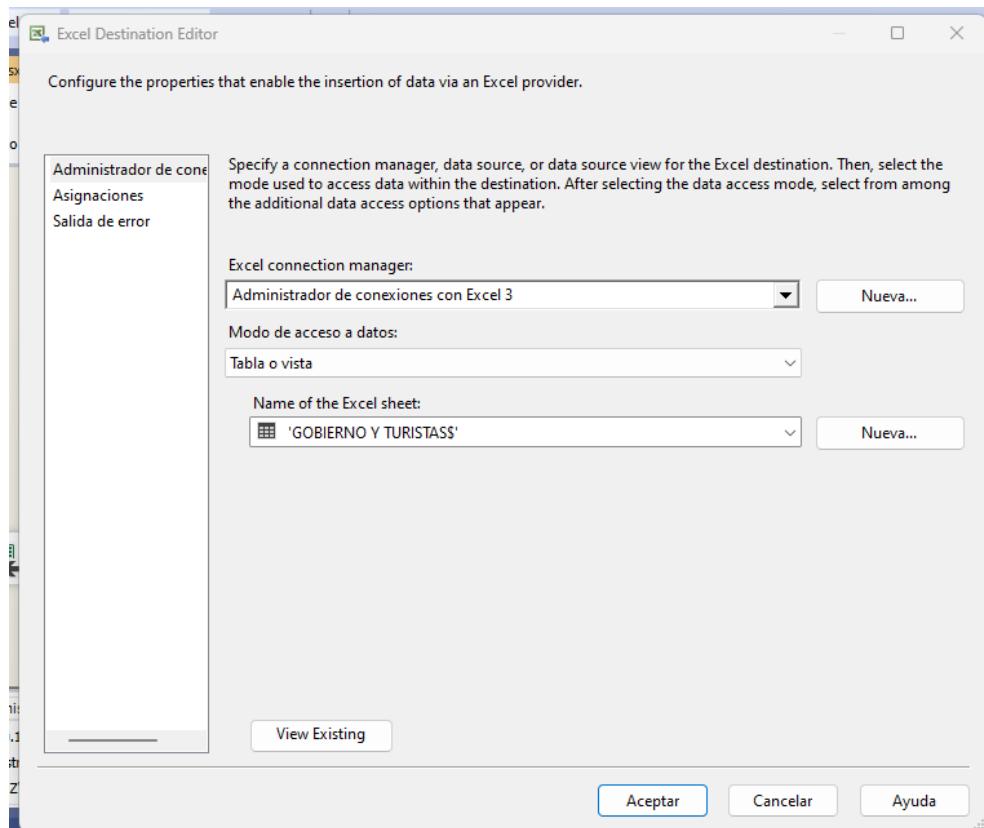
Lo cual quedara de la siguiente manera:



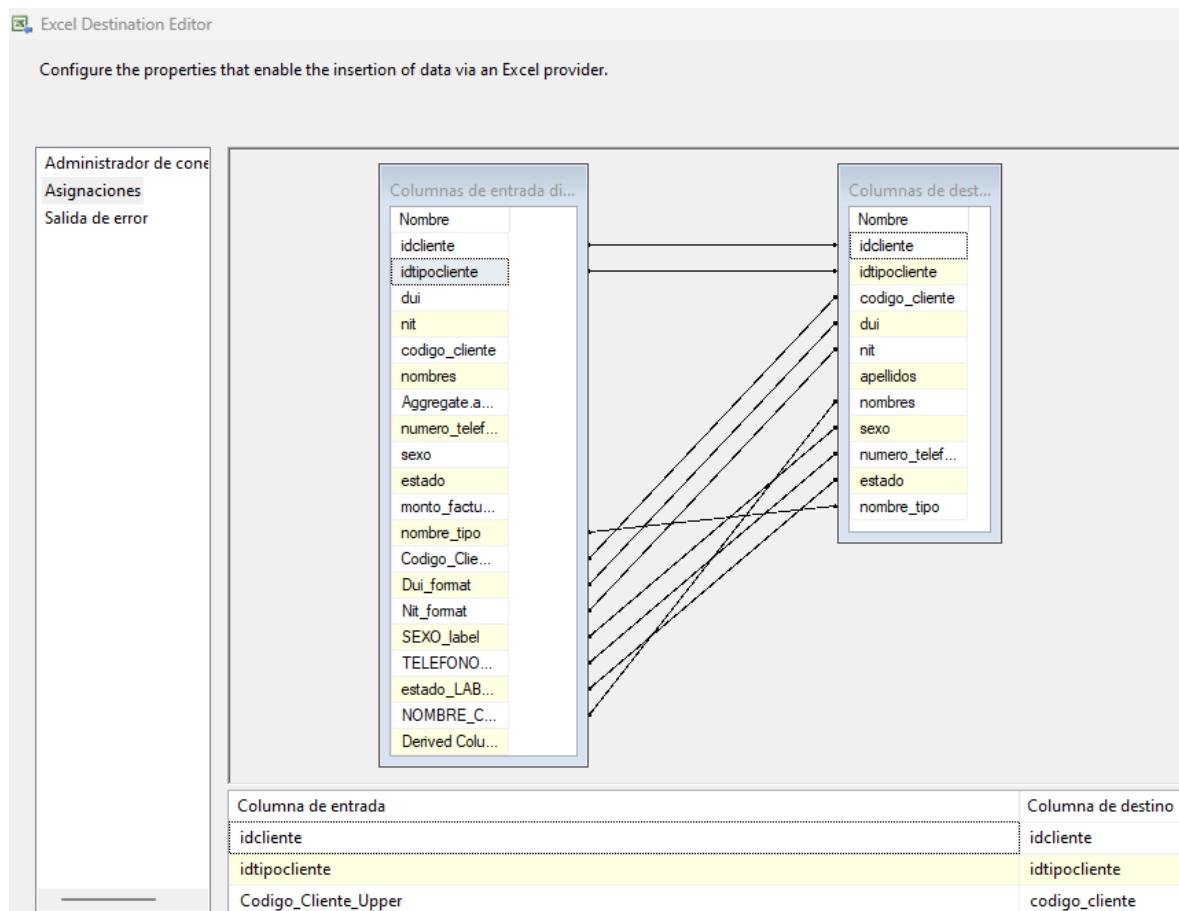
NOTA: Las asignaciones van a quedar de la siguiente manera:



En el otro extremo se realizara los mismos pasos, y nos quedara de la siguiente manera:



NOTA: Las asignaciones deben de quedar de la siguiente manera:



Al realizar la ejecución nos da el siguiente error:

