

Segunda convocatoria
ordinaria

Diseño y Explotación de Almacenes de Datos

Uso de una herramienta para ETL

Juan Manuel Cárdenas Recio

1. Introducción

Para la realización de esta práctica se proponía el uso de una herramienta ETL diferente de Integration Services de SQL Server que la Universidad de Málaga tuviera licencia o que fuera Open Source, por tanto, tras haber realizado un estudio de las múltiples opciones que había posibles he acabado eligiendo Azure Data Factory, herramienta que gracias a la universidad de Málaga dispongo de una prueba gratuita, que además de esto me gustó por su interfaz sencilla pero muy completa para realizar labores de Extracción, Transformación y Carga.

2. Uso de la herramienta ETL

Para comenzar el tutorial guiado que se pide en la realización en este apartado de la práctica voy a comenzar desde lo más básico para iniciar un proyecto de Factoría de Datos en Azure.

Obviamente lo primero que debemos hacer es iniciar sesión en el portal Azure con nuestra cuenta y en mi caso activar la suscripción gratuita que es la que me va a permitir usar todos los elementos que necesito de la plataforma, y antes de comenzar a trabajar debemos crear un grupo de recursos, que es donde vamos a ir añadiendo todos los componentes de nuestro proyecto, y tras esto creamos un servidor SQL para poder alojar la BD y el AD.

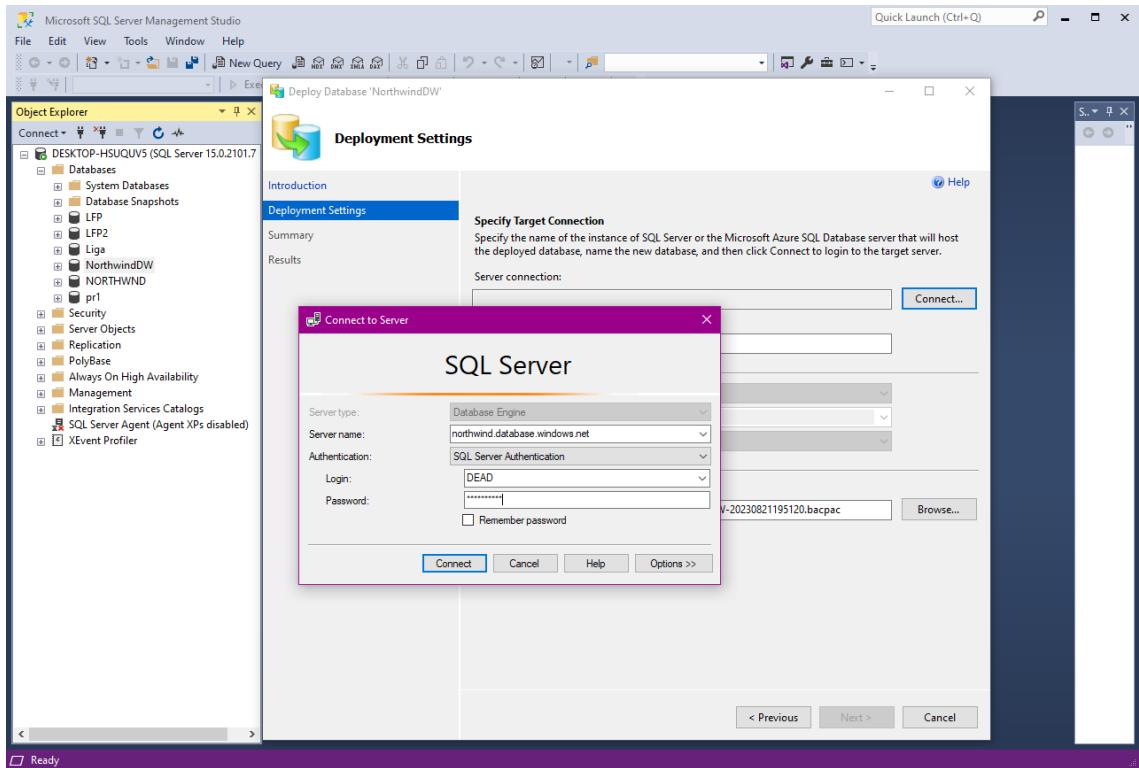
Nombre	Tipo	Ubicación
ETLndrvnd	Factoría de datos (V2)	West Europe
northwind	SQL Server	West Europe
NorthwindDW (northwind/NorthwindDW)	Base de datos SQL	West Europe
NORTHWND (northwind/NORTHWND)	Base de datos SQL	West Europe

Aquí si ve lo que tengo hasta ahora alojado en el grupo que es el servidor, con la base de datos, el almacén y el Data Factory.

The screenshot shows the Microsoft Azure portal interface. At the top, there's a search bar and a user profile. Below it, the main navigation bar includes 'Inicio > miGrupo > northwind'. The main content area displays the 'Información general' (General Information) for the 'northwind' server. It shows the group resource (miGrupo), state (Disponible), location (West Europe), subscription (Azure subscription 1), server ID, and tags. A 'Notificaciones' (Notifications) section indicates a free trial for Microsoft Defender for SQL is active, expiring in 26 days. On the left sidebar, there are links for Azure Active Directory, SQL Database, Groups, Quota of DTU, Properties, and Locks.

Aquí muestro el servidor que he creado con sus características correspondientes, como configuración adicional para poder realizar el siguiente paso debemos permitir en el apartado de redes, dentro de seguridad, el acceso a redes públicas para poder conectarnos desde SQL Server Management Studio, ahora sí podemos las bases de datos las cuales las tengo restauradas en SSMS.

The screenshot shows the Microsoft SQL Server Management Studio (SSMS) interface. The 'Object Explorer' pane on the left shows a connection to 'DESKTOP-HSUQUV5 (SQL Server 15.0.2101.7)' with several databases listed. In the center, a context menu is open over a database named 'Nex'. The menu options include 'Detach...', 'Take Offline', 'Stretch', 'Encrypt Columns...', 'Data Discovery and Classification', 'Vulnerability Assessment', 'Shrink', 'Back Up...', 'Restore', 'Mirror...', 'Launch Database Mirroring Monitor...', 'Ship Transaction Logs...', 'Generate Scripts...', 'Generate In-Memory OLTP Migration Checklists', 'Extract Data-tier Application...', 'Deploy Database to Microsoft Azure SQL Database...', 'Export Data-tier Application...', 'Register as Data-tier Application...', 'Upgrade Data-tier Application...', 'Delete Data-tier Application...', 'Import Flat File...', 'Import Data...', 'Export Data...', 'Copy Database...', 'Manage Database Encryption...', and 'Database Upgrade'. The 'Deploy Database to Microsoft Azure SQL Database...' option is highlighted.



Debemos conectarnos al servidor de SQL que hemos creado en Azure con el Usuario y Contraseña que hayamos elegido, también si es la primera vez que accedemos nos pedirá por motivos de seguridad que iniciemos sesión en Azure y registremos la IP de nuestro dispositivo local para poder entrar.

Una vez realizado esto el proceso será automático y se creará un SQL Database dentro del servidor que hayamos seleccionado y será una copia idéntica a la BD que tenemos en SSMS.

En el momento de la exportación de la base de datos vacía de Northwind, tuve un par de errores que no los tuve con la otra, que tras un tiempo investigando al respecto de porqué podría ser esto, averigüé que se debía a unos problemas de referencias no resueltas y ambiguas a unas vistas que estaban creadas en la base de datos, debido a esto corté por lo sano y las eliminé para que si son necesarias en un futuro añadirlas manualmente de nuevo.

Y una vez resuelto este error, me surgió otro, ya que había un usuario creado en la BD llamado esc, que tenía un AuthenticationType que no era compatible con Azure, y este al no ser un usuario necesario, al menos a priori, también lo eliminé, no sin antes tener que quitarle el permiso que tenía sobre un esquema para que me lo dejara hacer.

Ahora vamos a crear un Data Factory, para comenzar con el ETL.

Crearlo es muy sencillo al igual que se han creado los elementos anteriores, aquí le muestro las características que contiene, y para hacer el ETL accedemos a Iniciar Studio.

Para trabajar con las bases de datos tenemos que ir a Manage en la barra de la izquierda y debemos crear dos servicios vinculados con las BD que hemos creado, aquí podría aparecer un error debido a la IP de nuestro ordenador, la cual no estaría añadida al servidor, por tanto, vamos al servidor, en el mismo apartado de redes y añadir manualmente nuestra IPv4.

Y hacemos el proceso para la BD y el AD, el Integration Runtime viene por defecto en Azure para unir todos sus servicios y ya después es añadir los mismos datos que al hacer la exportación desde SSMS.

Ahora tras esto vamos a crear los Datasets para poder manejar los datos que se encuentran en la BD, para ello vamos a Author en la barra izquierda.

Vamos a comenzar con la tabla de Categorías, creando los dataset y su correspondiente flujo de datos.

Para crear el conjunto de datos lo primero que tenemos que hacer es seleccionar el tipo que obviamente es SQL Database y los creamos de la siguiente forma.

Microsoft Azure | Data Factory > ETLnrdwdn Fábrica de búsqueda y documentación juanmanuelcardenas@uma.es UNIVERSIDAD DE MÁLAGA

Microsoft anunció recientemente la versión preliminar pública de Microsoft Fabric, una forma totalmente nueva y emocionante de trabajar con tus datos.

Resursos de fábrica

Nombre: CategoríasORG

Servicio vinculado: NORTHWND

Conectar mediante Integration Runtime: AutoResolveIntegrationRuntime (Red virtual administrada)

Nombre de tabla: dbo.Categories

Importar esquema: Desde una conexión o un almacén

Aceptar Atrás Cancelar

Hay que habilitar la creación interactiva si no nos permitirá seguir, y una vez terminado podemos hacer una vista previa de los datos para comprobar que todo está bien.

Vista previa de los datos

Servicio: NORTHWND

vinculado:

Objeto: dbo.Categories

	CategoryID	CategoryName	Description	Picture
1	1	Beverages	Soft drinks, coffees, teas, beers, and ales	FRwvAAIAAAANAA4AFAAhAP///9CaXrtYXAgSW1hZ2UAUGFpbnQ
2	2	Condiments	Sweet and savory sauces, relishes, spreads, and seasonings	FRwvAAIAAAANAA4AFAAhAP///9CaXrtYXAgSW1hZ2UAUGFpbnQ
3	3	Confections	Desserts, candies, and sweet breads	FRwvAAIAAAANAA4AFAAhAP///9CaXrtYXAgSW1hZ2UAUGFpbnQ
4	4	Dairy Products	Cheeses	FRwvAAIAAAANAA4AFAAhAP///9CaXrtYXAgSW1hZ2UAUGFpbnQ

Propiedades

General Relacionado

Nombre: CategoríasORG

Descripción:

Opciones

Nuevo

The screenshot shows the Microsoft Azure Data Factory interface. On the left, there's a navigation pane with 'Recursos de fábrica' (Resources) expanded, showing 'CategoriasDST' under Datasets. The main area is titled 'Vista previa de los datos' (Data Preview) for 'CategoriasDST'. It shows a table with columns: CategoryKey, CategoryName, and Description. The table has one row with values: CategoryKey 1, CategoryName 'Beverages', and Description 'Soft drinks and beer'. To the right, there's a 'Propiedades' (Properties) panel with tabs for General, Relational, and other settings like 'Nombre' (Name) and 'Descripción' (Description). A status bar at the bottom says 'CategoríasDST'.

Ahora crearemos el flujo de datos para la exportación de estos.

This screenshot shows the configuration of a data flow named 'dataflow1'. The top part shows the data flow visualization with two stages: 'OrigenCAT' (Importar datos de CategoriasORG) and 'DestinoCAT' (Exportar datos a CategoriasDST). The 'DestinoCAT' stage shows 'Columnas Total: 4'. The bottom part shows the 'Configuración de origen' (Source Configuration) tab. Under 'Nombre de la secuencia de salida' (Output sequence name), it says 'OrigenCAT' and 'Importar datos de CategoriasORG'. Under 'Tipo de origen' (Source type), it says 'Conjunto de datos' (Dataset). Under 'Conjunto de datos', it says 'CategoriasORG'. There are also 'Opciones' (Options) checkboxes for schema drift and schema validation. The 'Receptor' (Sink) tab at the bottom shows 'Nombre de la secuencia de salida' set to 'DestinoCAT' and 'Exportar datos a CategoriasDST'. Under 'Tipo de receptor', it says 'Conjunto de datos' (Dataset) and 'CategoriasDST'. There are also 'Opciones' checkboxes for schema drift and schema validation.

Agregamos un origen de esta forma seleccionando el conjunto de datos y le agregamos un receptor, en la práctica se hizo cambiando el tipo de datos de la descripción a String, pero en data factory esto nos lo realiza de manera automática, ya que si escogemos el conversor de datos nos aparecen tipos normales ya sea string, integer, float, etc. Pero no tipo nvarchar por ejemplo, y no especificar su longitud.

The screenshot displays the Microsoft Azure Data Factory Data Flow designer interface. It shows two separate configurations for a data flow named 'dataflow1'.

Top Configuration:

- Source:** 'OrigenCAT' dataset (Importar datos de CategoríasCAT).
- Transformation:** A 'cast1' step that changes the type of the 'CategoryID' column from 'string' to 'integer'. This transformation is part of a sequence named 'cast1'.
- Destination:** 'DestinoCAT' dataset (Exportar datos a CategoríasCAT).
- Properties:** The data flow is named 'dataflow1'.
- Conversion Configuration:** Shows the conversion of 'CategoryID' from 'string' to 'integer'.

Bottom Configuration:

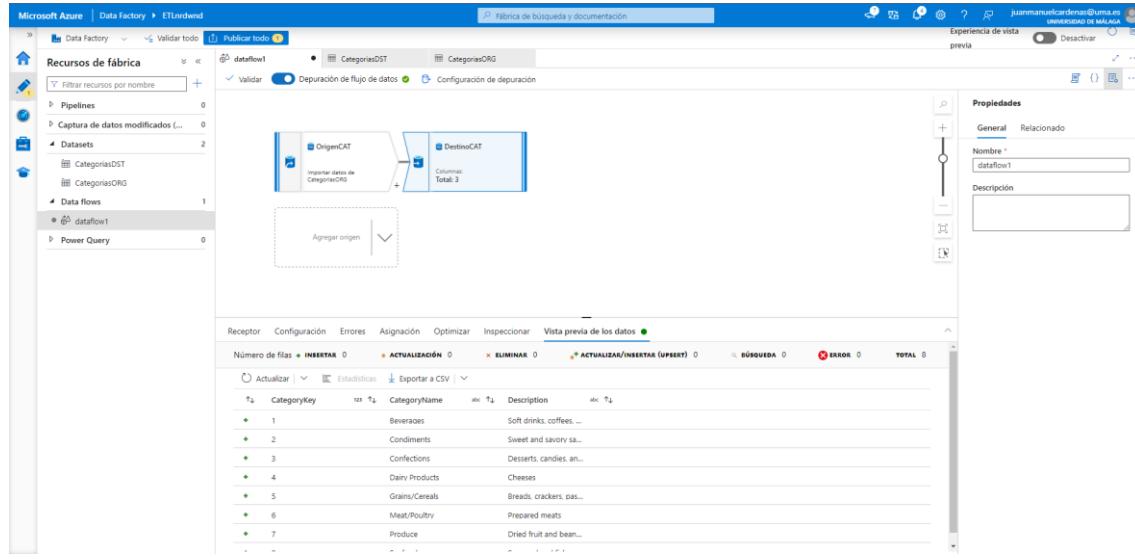
- Source:** 'OrigenCAT' dataset (Importar datos de CategoríasCAT).
- Transformation:** An 'Agregar origen' (Add Source) step, which is currently selected.
- Destination:** 'DestinoCAT' dataset (Exportar datos a CategoríasCAT).
- Properties:** The data flow is named 'dataflow1'.
- Schema:** Shows the schema mapping between the source and destination columns.

The screenshot shows the Microsoft Azure Data Factory Studio interface. On the left, the 'Recursos de fábrica' (Factory Resources) sidebar lists 'Pipelines' (0), 'Captura de datos modificados (...)' (0), 'Dataset' (2), and 'Data flows' (1). The selected item is 'CategoríasDST'. The main workspace displays a dataset named 'CategoríasDST' connected to an 'Azure SQL Database' source. The 'Esquema' (Schema) tab is selected, showing the structure of the 'CategoríasDST' table. The table has three columns: 'CategoryKey' (int), 'CategoryName' (nvarchar), and 'Description' (nvarchar). Below the schema, there are buttons for 'Importar esquema' (Import schema) and 'Borrar' (Delete).

Así que en destino seleccionamos la asignación de los atributos que queremos hacer, activamos la depuración del flujo de datos sobre hora y ya podríamos tener una vista previa de nuestro flujo.

The screenshot shows the Microsoft Azure Data Factory Studio interface. The 'Recursos de fábrica' (Factory Resources) sidebar lists 'Pipelines' (0), 'Captura de datos modificados (...)' (0), 'Dataset' (2), and 'Data flows' (1). The selected item is 'dataflow1'. The main workspace shows a data flow named 'dataflow1' with two stages: 'OrigenCAT' (Importar datos de CategoríasORG) and 'DestinoCAT' (CategoríasDST). The 'Asignación' (Mapping) tab is selected. In the 'Opciones' (Options) section, two checkboxes are checked: 'Omitir columnas de entrada duplicadas' (Skip input duplicate columns) and 'Omitir columnas de salida duplicadas' (Skip output duplicate columns). Below this, there are buttons for 'Restablecer' (Reset), 'Agregar asignación' (Add mapping), and 'Formato de salida' (Output format). A note indicates '3 asignaciones: se han asignado todas las salidas.' (3 mappings: all outputs have been assigned). The 'Propiedades' (Properties) panel on the right shows the name 'dataflow1' and a description field.

Y observamos que se ha realizado correctamente.



Ahora antes de crear el pipeline donde se van a ejecutar los dataflows vamos a crear un procedimiento almacenado que se encargue en cada iteración de borrar lo que tenemos almacenado en NorthwindDW para evitarnos errores, que iré actualizando conforme vaya avanzando en el proyecto.

El procedimiento se puede crear desde SSMS conectándose al servidor que tenemos creado en Azure de la misma forma que hicimos a la hora de la exportación.

The screenshot shows the Microsoft SQL Server Management Studio (SSMS) interface. The 'Object Explorer' pane shows the database structure for 'northwind.database.windows.net.NorthwindDW'. The 'Tables' node under 'NorthwindDW' is expanded, showing 'Categories' and 'Products'. The 'Script' tab for the 'Categories' table is selected, displaying the T-SQL code for creating a stored procedure named 'Borrar'. The code includes setting ANSI_NULLS and QUOTED_IDENTIFIER, defining the stored procedure to delete from the 'Category' table, and providing comments for the author, create date, and description. The 'SQLQuery6.sql' file is open in the main window, showing the stored procedure definition.

```

SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
<-- ===== Object: StoredProcedure [dbo].[Borrar] Script Date: 23/08/2023 13:51:13 =====
-- Author: <Author>, <Name>
-- Create Date: <Create Date>, >
-- Description: <Description>, >
ALTER PROCEDURE [dbo].[Borrar]
AS
BEGIN
    delete from dbo.Category
END

```

Ahora creamos el pipeline y lo agregamos buscándolo en la barra de la izquierda y lo configuraremos de la siguiente forma.

The screenshot shows the Azure Data Factory interface for creating an ETL pipeline. In the left sidebar, under 'Recursos de fábrica', 'Pipelines' is selected. The main workspace displays a single activity named 'Stored procedure1'. The right-hand panel shows the 'Propiedades' (Properties) section for this activity, specifically the 'Configuración' (Configuration) tab. Key settings include:

- Servicio vinculado:** NorthwindDW (selected from a dropdown).
- Entorno de ejecución de integración:** AutoResolveIntegrationRuntime (selected from a dropdown).
- Nombre del procedimiento almacenado:** [dbo].[Borrado] (selected from a dropdown).

Ahora ya agregamos el flujo de datos y lo unimos mediante la flecha azul que nos indica que se ejecutará tras acabar el proceso y se configura también así.

The screenshot shows the same pipeline configuration screen. A new 'Flujo de datos' (Data Flow) activity has been added to the pipeline. It is connected to the 'Stored procedure1' activity via a blue arrow, indicating a sequential execution flow. The 'Configuración' (Configuration) tab for the 'Flujo de datos' activity shows the following settings:

- Flujo de datos:** CategoríaFLOW (selected from a dropdown).
- Ejecutar en (Azure IR):** AutoResolveIntegrationRuntime (selected from a dropdown).
- Tamaño de proceso:** Small (selected from a dropdown).

Ahora procedemos depurar para comprobar que no hay errores.

The screenshot shows the 'Historial de ejecución' (Run History) page for the pipeline. It displays the execution details for two activities:

Nombre de actividad	Estado de actividad	Tipo de actividad	Inicio de la ejecución	Duración	Registro	Entorno de ejecución
Categorías	Correcto	Flujo de datos	8/23/2023, 2:04:11 PM	55s		AutoResolveIn
Borrado	Correcto	Procedimiento almacenado	8/23/2023, 2:03:13 PM	58s		AutoResolveIn

Ahora continuamos con productos, que como antes creamos los datasets de las tablas y creamos el flujo de datos asignando los atributos correspondientes.

Microsoft Azure | Data Factory > ETLnwind

Fábrica de búsqueda y documentación juanmanuelcardenas@uma.es UNIVERSIDAD DE MÁLAGA

Experiencia de vista previa Desactivar

Recursos de fábrica

- Pipelines
- Captura de datos modificados...
- Datasets

 - CategoríasDST
 - CategoríasORG
 - ProductosDST
 - ProductosORG

- Data flows

 - CategoríaFLOW
 - dataflow1

- Power Query

CategoríaFLOW

Validar Depuración de flujo de datos Configuración de depuración

OrigenPRD DestinoPRD

importar datos de ProductsORG

Columnas Total: 6

Agregar origen

Propiedades

General Relacionado

Nombre: dataflow1

Descripción

dataflow1

Receptor Configuración Errores Asignación Optimizar Inspeccionar Vista previa de los datos

Opciones

- Omitir columnas de entrada duplicadas
- Omitir columnas de salida duplicadas

Asignación automática Restablecer Agregar asignación Eliminar

Formato de salida

6 asignaciones: se han asignado todas las salidas.

Columnas de entrada Columnas de salida

Columna de Entrada	Columna de Salida
ProductID	ProductKey
ProductName	ProductName
QuantityPerUnit	QuantityPerUnit
UnitPrice	UnitPrice
Discontinued	Discontinued
CategoryID	CategoryKey

Microsoft Azure | Data Factory > ETLnwind

Fábrica de búsqueda y documentación juanmanuelcardenas@uma.es UNIVERSIDAD DE MÁLAGA

Experiencia de vista previa Desactivar

Recursos de fábrica

- Pipelines
- Captura de datos modificados...
- Datasets

 - CategoríasDST
 - CategoríasORG
 - ProductosDST
 - ProductosORG

- Data flows

 - CategoríaFLOW
 - ProductosFLOW

- Power Query

ETL

Actividades

Validar Depurar Agregar desencadenador Depuración de flujo de datos

Borrar → Carga Categorías → Carga Productos

Procedimiento almacenado → Flujos de datos

Propiedades

General Relacionado

Nombre: ETL

Descripción

Anotaciones

ETL

Parámetros Variables Configuración Salida

Id. de ejecución de canalización: f7d8a339-f7e0-4d35-8ec-03559f1d2931 Estado de la canalización Correcto

All status Exportar a CSV

Mostrando elementos del 1 al 3 de un total de 3

Nombre de actividad	Estado de actividad	Tipo de actividad	Inicio de la ejecución	Duración	Registro	Entorno de ejecución
Carga Productos	Correcto	Flujo de datos	8/23/2023, 2:34:14 PM	17s		AutoResolveV
Carga Categorías	Correcto	Flujo de datos	8/23/2023, 2:30:19 PM	3m 53s		AutoResolveV
Borrado	Correcto	Procedimiento almacenado	8/23/2023, 2:30:14 PM	5s		AutoResolveV

Object Explorer

Connect to database northwind.database.windows.net (SQL Server)

Databases

- NorthwindDW
- Northwind

Tables

- Customers
- CustomerDemographics
- Employees
- EmployeeTerritories
- Products
- Regions
- Shippers
- Suppliers
- Categories
- ProductCategories
- ProductDescription
- ProductDetails
- ProductPhoto
- ProductSpecification
- ProductSubcategory
- Region
- Territory
- View
- External Resources
- Programmability
- Query Store
- Extended Events
- Extended Properties
- Security
- NORTHWND
- Security
- Integration Services Catalogs

SQL Query Log - nwindDW (READ (88)) = N

```
***** Script for SelectTopNRows command from SSMS *****
SELECT TOP 1000 [ProductKey]
      ,[ProductName]
      ,[QuantityPerUnit]
      ,[UnitPrice]
      ,[Discontinued]
      ,[CategoryKey]
   FROM [dbo].[Product]
```

Results Messages

ProductKey	ProductName	QuantityPerUnit	UnitPrice	Discontinued	CategoryKey
1	Che	10 boxes of 20 bags	18.00	0	1
2	Ch	24 oz (600 g) cans	18.00	0	1
3	Arroz	12.50 oz bottles	10.00	0	2
4	Arroz	40- 45 oz bottles	22.00	0	2
5	Chef Anton's Cajun Seasoning	40- 64 oz jars	21.35	1	2
6	Chef Anton's Gumbo Mix	36 boxes	21.35	1	2
7	Grandma's Boysenberry Spread	12- 16 oz jars	25.00	0	2
8	Uncle Bob's Organic Dried Pears	12- 16 oz jars	30.00	0	7
9	Northwind Cranberry Sauce	12- 12 oz jars	40.00	0	2
9	Mahi Kobe Niku	18- 500 g pkgs.	97.00	1	6
10	Ikura	12- 200 ml jars	31.00	0	8
11	Queso Cabrío	1 kg pkgs	21.00	0	4
12	Queso Manchego La Pastorita	10- 300 g pkgs	20.00	0	4
13	Korku	2 kg pkgs	6.00	0	8
14	Tofu	40- 100 g pkgs.	23.25	0	7
15	Genen Shouyu	24- 250 ml bottles	15.50	0	2
16	Pavlova	32- 500 g boxes	17.45	0	3
17	Alice Mutton	20- 1 kg lbs	31.00	1	6
18	Cottage Cheese	1kg pkgs	63.50	0	8
19	Teatime Chocolate Biscuits	10 boxes of 12 pieces	9.20	0	3

Query executed successfully.

Y observamos que se realiza correctamente ahora continuamos con empleados y transportista, y modificamos el procedimiento almacenado para incluir esta última.

La realización de estos es igual que las anteriores.

The screenshot displays the Microsoft Azure Data Factory ETL designer interface. It shows two separate data flow configurations and an activity timeline.

Data Flow Configuration 1:

- Source:** OrigenEMP (Importar datos de EmpleadosCRS)
- Destination:** DestinoEMP (Categorías, Total: 13)
- Properties:** Relacionado, Nombre: EmpleadoFLOW
- Assignment Tab:** Shows 13 assignments where all outputs are assigned.
- Columns:** Mappings between EmployeeID, FirstName, LastName, Title, TitleOfCourtesy, BirthDate, HireDate, Address, City, Region, PostalCode, Country, and SupervisorKey.

Data Flow Configuration 2:

- Source:** EmpleadoORG
- Destination:** EmpleadoORG
- Properties:** Relacionado, Nombre: EmpleadoORG
- Assignment Tab:** Shows 13 assignments where all outputs are assigned.
- Columns:** Mappings between EmployeeID, FirstName, LastName, Title, TitleOfCourtesy, BirthDate, HireDate, Address, City, Region, PostalCode, Country, and SupervisorKey.

Activity Timeline:

- Activities:** Includes 'Borrado' (Delete), 'Carga Categorías' (Load Categories), 'Carga Productos' (Load Products), and 'Carga Empleados' (Load Employees).
- Flow:** The 'Borrado' activity connects to 'Carga Categorías'. 'Carga Categorías' connects to both 'Carga Productos' and 'Carga Empleados'.
- Properties:** Relacionado, Nombre: ETL
- Timeline View:** Shows the execution status of the data flows and activities.

The screenshot displays two separate Azure Data Factory pipelines:

- Pipeline 1 (Top):** This pipeline is titled "ETL" and consists of a single activity named "Importar datos de TransportistaORG". It maps data from a source dataset "OrigenTRS" to a destination dataset "DestinoTRS". The configuration pane shows options for skipping duplicate rows in both input and output.
- Pipeline 2 (Bottom):** This pipeline is also titled "ETL" and represents a complex data flow. It includes several activities:
 - "Carga Categorías"
 - "Carga Productos"
 - "Borrado" (Delete)
 - "Carga Empleados"
 - "Carga Transportista"
 These activities are interconnected by data flows, forming a larger data processing logic.

El siguiente paso es la dimensión tiempo, que ahora tenemos algo más de complejidad puesto que los datos no los extraemos de otra tabla si no con un excel.

Para trabajar tenemos dos opciones subirlos a la nube de Azure o que se pueda conectar con el directorio en el que tenemos el archivo en nuestro ordenador, yo para evitar fallos he optado por la primera opción, por tanto he creado una cuenta de almacenamiento en el grupo de recursos.

Datos básicos

Detalles del proyecto

Seleccione la suscripción en la que se creará la nueva cuenta de almacenamiento. Elija un grupo de recursos nuevo o uno ya existente para organizar y administrar la cuenta de almacenamiento junto con otros recursos.

Suscripción * Azure subscription 1

Grupo de recursos * miGrupo

Crear nuevo

Detalles de la instancia

Nombre de la cuenta de almacenamiento archivosenorth

Región (Europe) West Europe

Revisar < Anterior Siguiente: Opciones avanzadas >

Enviar comentarios

Creamos un contenedor dentro de la cuenta de almacenamiento y añadimos los archivos que serán necesarios ahora y en un futuro en el proyecto, y ya los podríamos usar como conjunto de datos en nuestro Data Factory.

Información general

Método de autenticación: Clave de acceso (Cambiar a la cuenta de usuario de Azure AD)

Ubicación: dimensiones

Nombre Modificado Nivel de acceso Estado del archivo

- cities.txt 23/8/2023, 19:12:21 Frecuente (infértil)
- Territories.xml 23/8/2023, 19:12:21 Frecuente (infértil)
- Territories.xsd 23/8/2023, 19:12:21 Frecuente (infértil)
- Time.xls 23/8/2023, 19:12:21 Frecuente (infértil)

Cargar blob

Arastrar y colocar archivos aquí o Buscar archivos

Sobre escribir los archivos si ya existen

Cargar Enviar comentarios

Cargas actuales Descartar: Completado Todo

- Territories.xml 211.99 KB / 211.99 KB
- Territories.xsd 2.23 KB / 2.23 KB
- Time.xls 172.5 KB / 172.5 KB
- cities.txt 3.68 KB / 3.68 KB

Una vez terminado la subida de archivos podemos pasar a la creación del conjunto de datos, que en esta ocasión lo debemos crear del tipo Almacenamiento de Blobs y no como SQL Database, como las anteriores, tras esto nos pedirán especificar el tipo el cual al ser un archivo .xsl debemos poner Excel.

Yo al no haber creado con anterioridad el servicio vinculado automáticamente me pedirá que lo especifique, y de la misma forma que hicimos con la conexión a las dos bases de datos lo hacemos aquí.

Terminada la conexión con la cuenta de almacenamiento nos pedirá que indiquemos el directorio donde lo tenemos almacenado, que en este caso es en el contenedor que se mencionó antes. Le damos a Aceptar y ya lo tendríamos creado, como se ve en la segunda foto

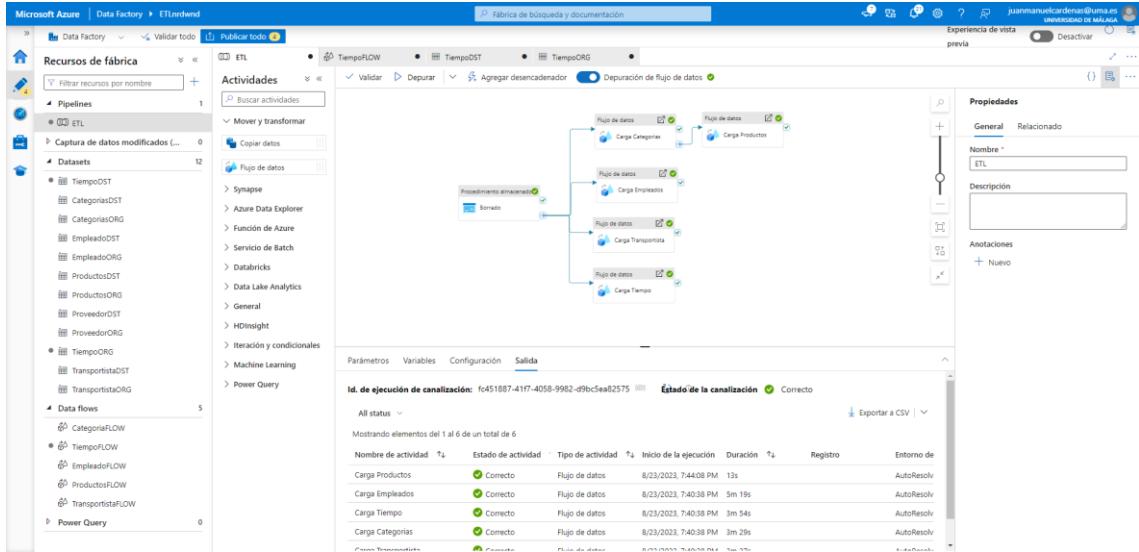
The screenshot shows the Microsoft Azure Data Factory interface. On the left, the 'Recursos de fábrica' (Fabric resources) sidebar lists various datasets, pipelines, and data flows. In the center, a pipeline named 'ETL' is selected. On the right, the 'TransportistaDST' dataset properties are being edited. The 'Nombre' (Name) field is set to 'Tiempo'. Under 'Servicio vinculado' (Linked service), 'ArchivoETL' is selected. Under 'Conectar mediante Integración Runtime', 'AutodeskIntegrationRuntime (Red virtual administrada)' is chosen. The 'Ruta de acceso del archivo' (File path) is set to 'dimensiones / Directorio / Time.xls'. The 'Modo de hoja de cálculo' (Calculation sheet mode) is set to 'Nombre'. The 'Nombre de hoja' (Sheet name) is set to 'Ninguna'. The 'Primer fila como encabezado' (First row as header) checkbox is checked. The 'Importar esquema' (Import schema) section has 'Revisar una conexión o un almacén' (Review connection or store) selected. At the bottom, the 'Aceptar' (Accept) button is highlighted.

This screenshot shows the Microsoft Azure Data Factory interface again. The 'Recursos de fábrica' sidebar is visible on the left. In the center, the 'TiempoORG' dataset properties are being edited. The 'Nombre' (Name) is set to 'TiempoORG'. The 'Vista previa de los datos' (Data preview) window is open, showing a preview of the 'Time.xls' file. The preview table has columns: DateAltKey, DayNbWeek, DayNameWeek, DayNbMonth, DayNbYear, WeekNbYear, MonthNumber. The data starts with: 1 1990-07-01, 2 Monday, 1, 182, 27, 7. The 'Propiedades' (Properties) pane on the right shows the 'General' tab selected, with 'Nombre' set to 'TiempoORG'. The 'Aceptar' (Accept) button is visible at the bottom.

Debido a que esto es un archivo Excel los datos extraídos para las filas son todos de tipo String, por tanto, vamos a necesitar un conversor para modificarlos al que necesitamos, para saber a que tipo convertirlos yo he usado un “truco” podiendo un segundo origen como se puede ver en las imágenes con el dataset de destino, para guiarme mejor y no cometer errores.

Terminado esta especificación borramos el segundo origen que no nos hace falta y creamos un receptor tras el conversor, hacemos como anteriormente la asignación de atributos, sin poner nada en TimeKey ya que está definido con la propiedad Identity (1,1) y no es necesario.

Y como podemos ver está correctamente funcionando.



Ahora pasamos a la dimensión Geografía, que la primera que crearemos será Continente, ya que no depende de ninguna otra tabla.

Para comenzar tenemos que crear un dataset del archivo que también subimos antes de Territories.xml, de igual forma que hicimos anteriormente con la dimensión tiempo, que se encuentran alojados los datos de esta dimensión, además también tenemos al archivo XSD para especificar su esquema.

Azure no nos da directamente una herramienta para enlazar ambos archivos, por lo tanto, debemos incluir la ruta relativa del archivo en la cabecera del elemento raíz, que en mi caso al estar ambos dentro de la misma carpeta en el mismo contenedor solo tengo que agregar el nombre se la siguiente manera:

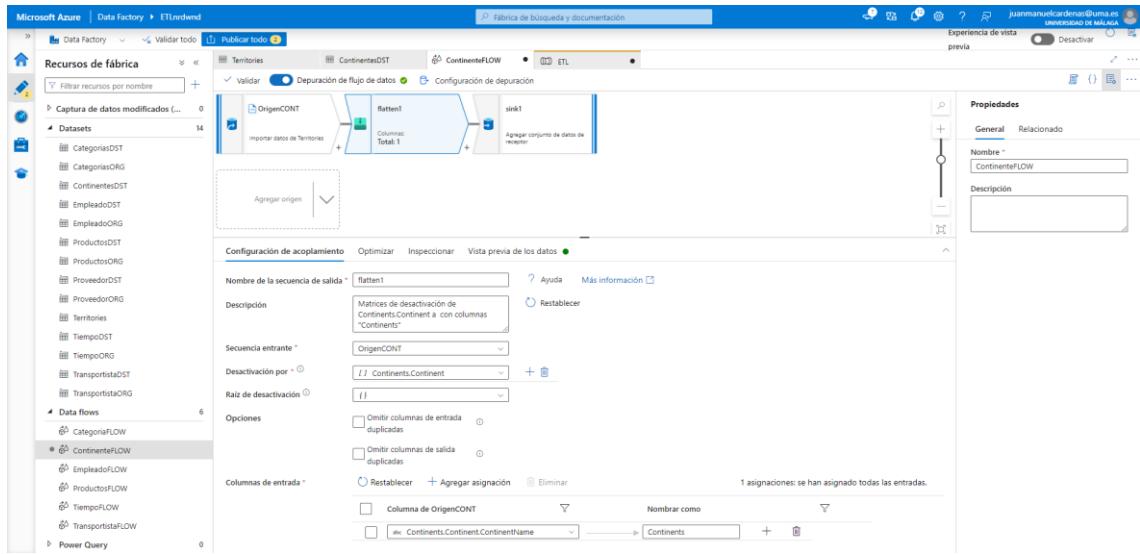
```
<Continents xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:noNamespaceSchemaLocation="Territories.xsd">
```

Una vez creado el conjunto de datos de territorio, creamos otro como hemos venido haciendo, con la tabla de continente en la base de datos de destino.

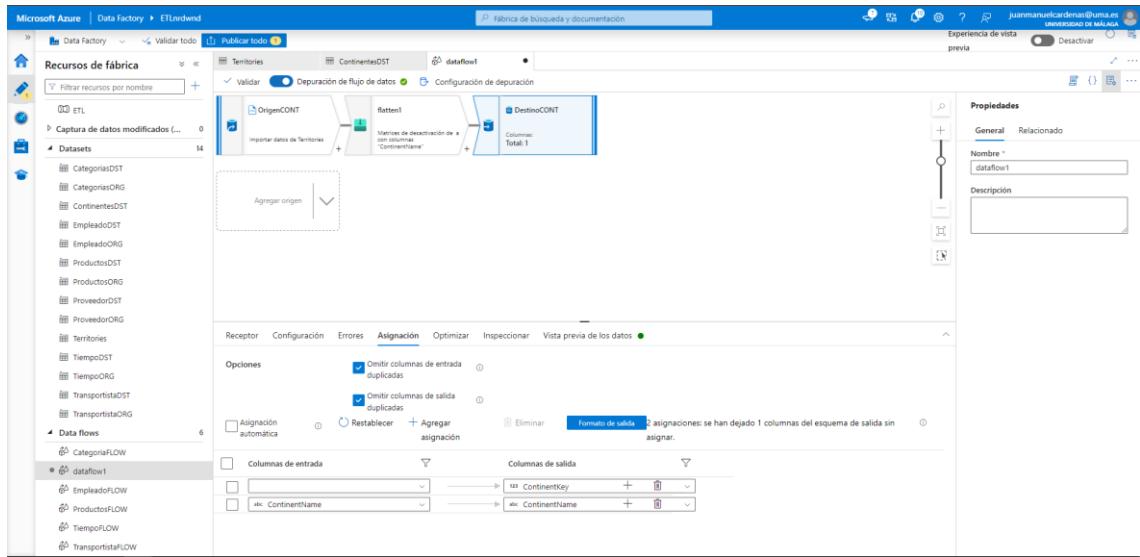
También terminado este paso, en el flujo de datos con el cual vamos a hacer la exportación, en el origen como siempre seleccionamos ahora el conjunto de datos de Territories, con la novedad que al ser un XML y debemos de indicar que queremos que se valide, ya que por defecto está desactivada, a través de la información que pusimos en el elemento raíz, si en vez XSD fuera DTD también tenemos la opción.

Y ahora ya podemos importar el esquema como se muestra en la segunda imagen de abajo, yendo a Proyección -> Importar proyección y se realiza automáticamente.

Ahora que ya tenemos nuestro origen de datos correctamente configurado vamos a necesitar la herramienta de acoplar, como su nombre indica acoplar los datos, y que no aparezcan errores de columnas NULL porque no se hayan detectado bien, le indicamos que queremos que solo se acoplen los elementos de “Continent”, y hacemos la asignación, en este caso como solo necesitamos el nombre de los continentes es la única asignación que he hecho.

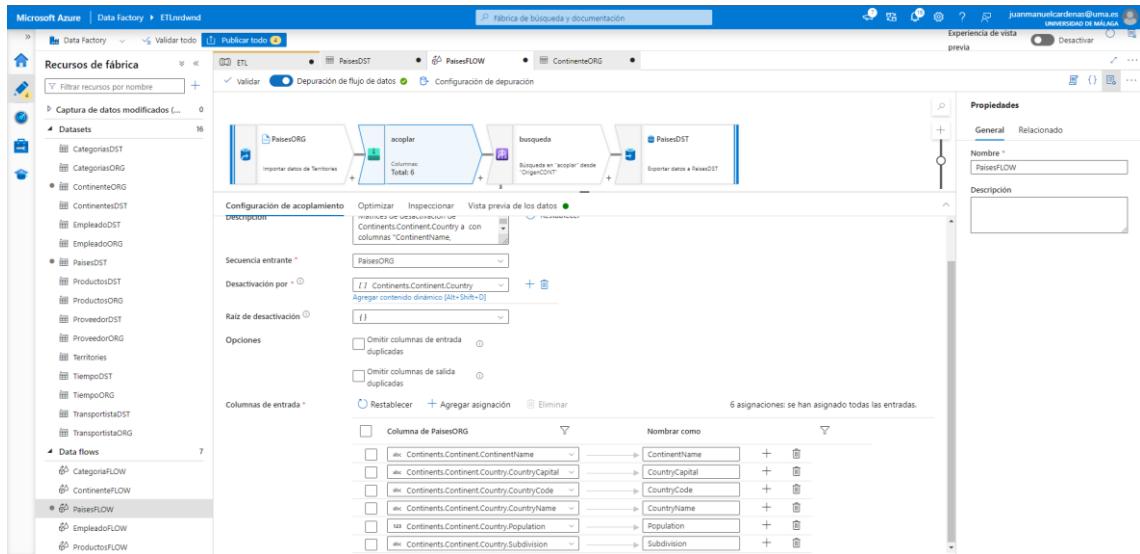


Y como en tiempo como el ID se asigna automáticamente no necesitamos asignarle nada.

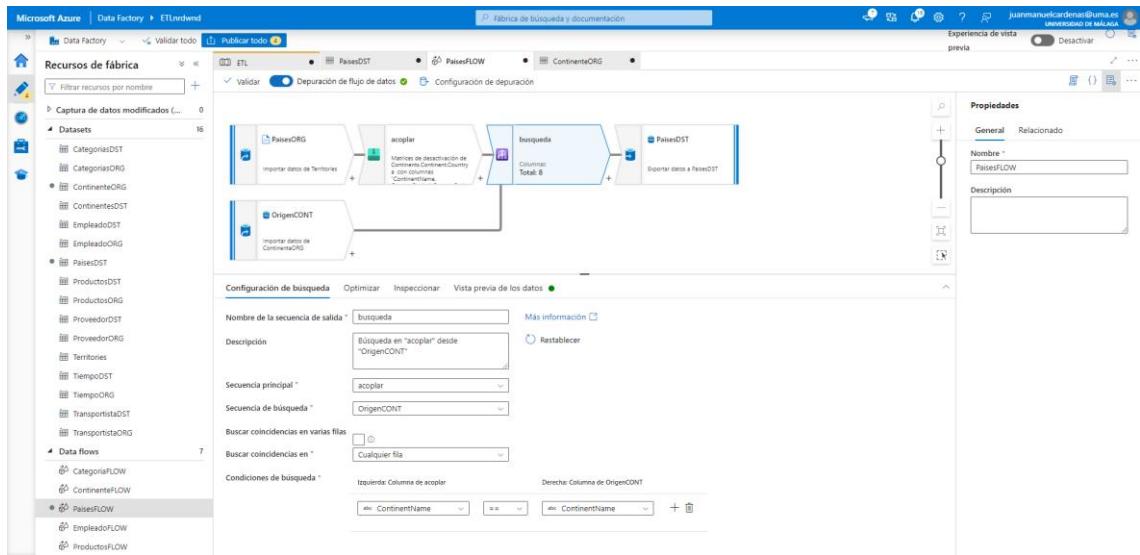


Ahora continuamos con los países y comenzamos el flujo como anteriormente añadiendo el origen de datos de Territories, y haciendo los mismos pasos para el esquema.

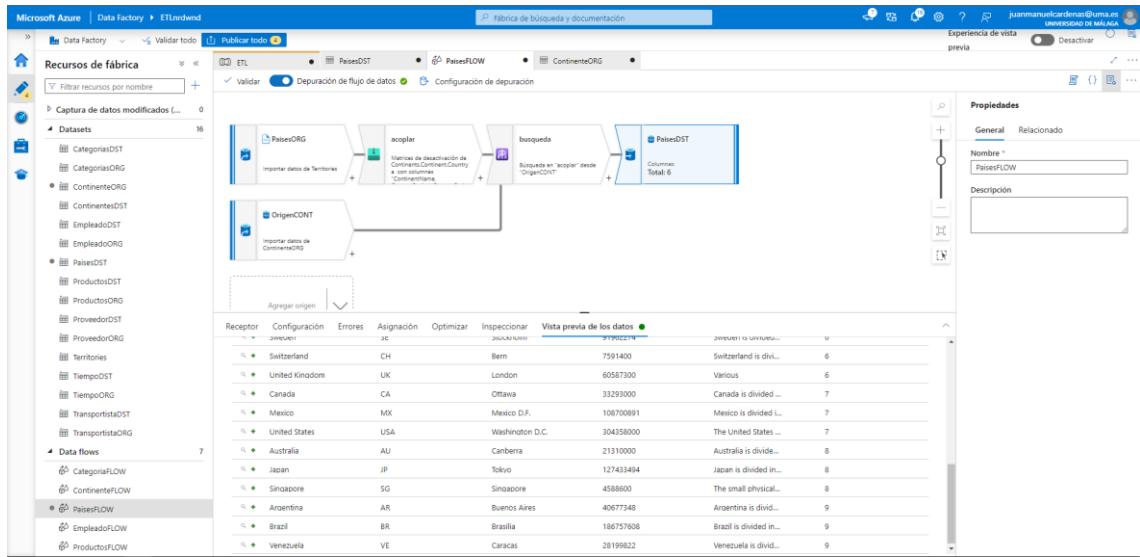
Lo primero que he realizado ha sido como antes acoplar los datos de la salida del XML para poder utilizarlos adecuadamente, pero esta vez haciendo que se desactive en tabla que queremos llenar que es Country, asignamos los atributos.



Posteriormente se añade el elemento búsqueda que nos permitirá encontrar el ID asignado en la base de datos a el nombre del continente al que pertenece el país, por esta razón añadí como segunda fuente la tabla Continente que se rellenaba en el paso anterior, y como se observa el resultado son 8 tablas porque une las dos tablas de Continente a las 6 que hemos extraído del XML, así nos es más facil comprobar que la comparación ha sido correcta.

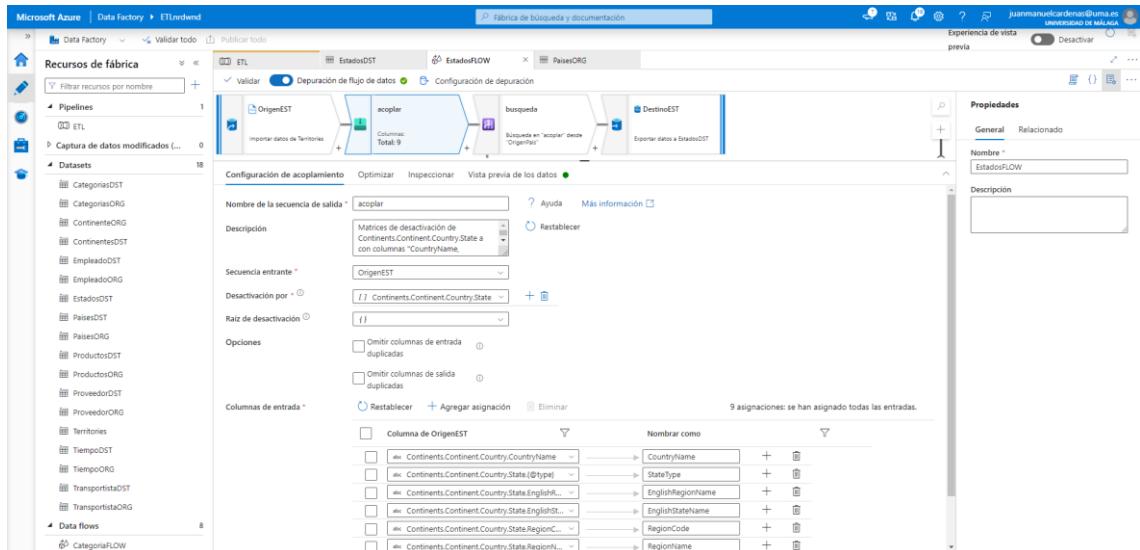


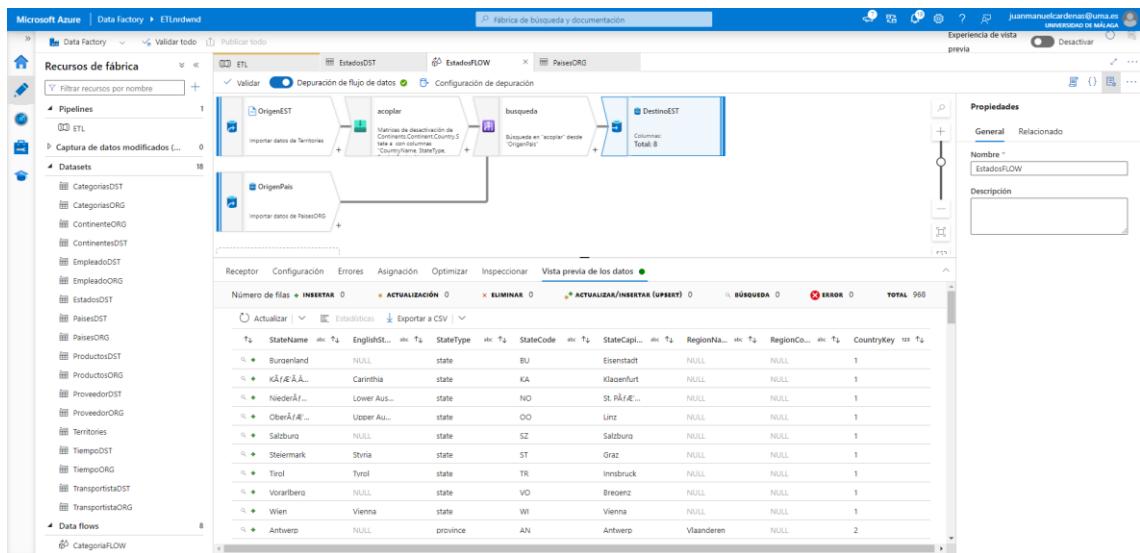
Una vez obtenido el valor que queremos en esta operación, ya podemos completar la tabla de Paises con el elemento ID que nos faltaba.



Ahora continuamos con los estados que se realiza de manera muy similar a los países puesto que también tenemos que buscar en la BD el ID que tiene asociado.

Para ello con la herramienta para acoplar extraemos el nombre del país que tiene asociado cada estado, y en la herramienta de búsqueda, como hemos hecho antes he conectado previamente un nuevo origen a la tabla de Paises, y hacemos la comparación de que ambos CountryName sean iguales, y de la unión de ese resultado extraemos el ID para asignarlo a en el destino.

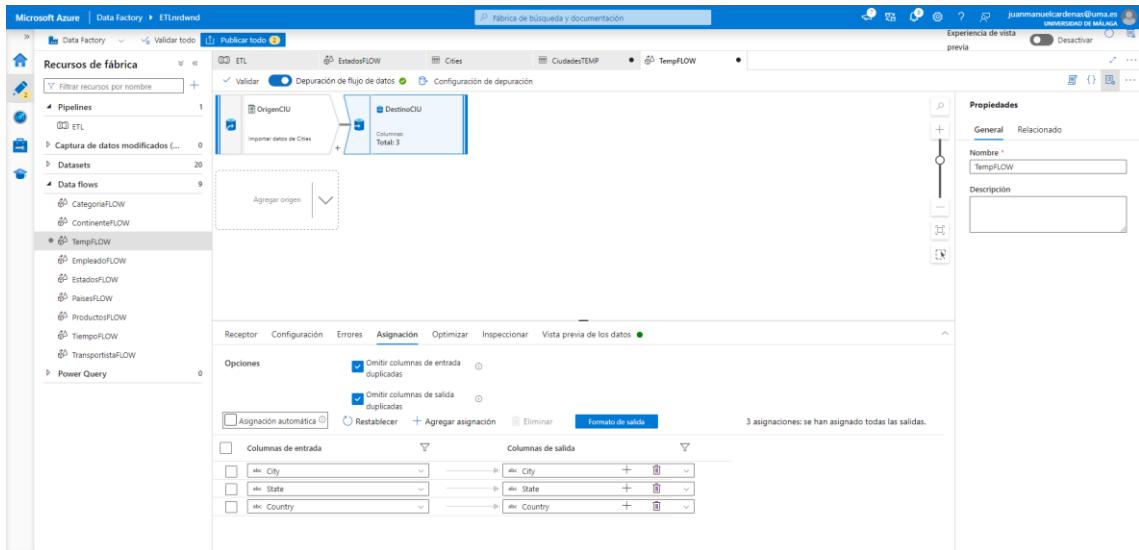




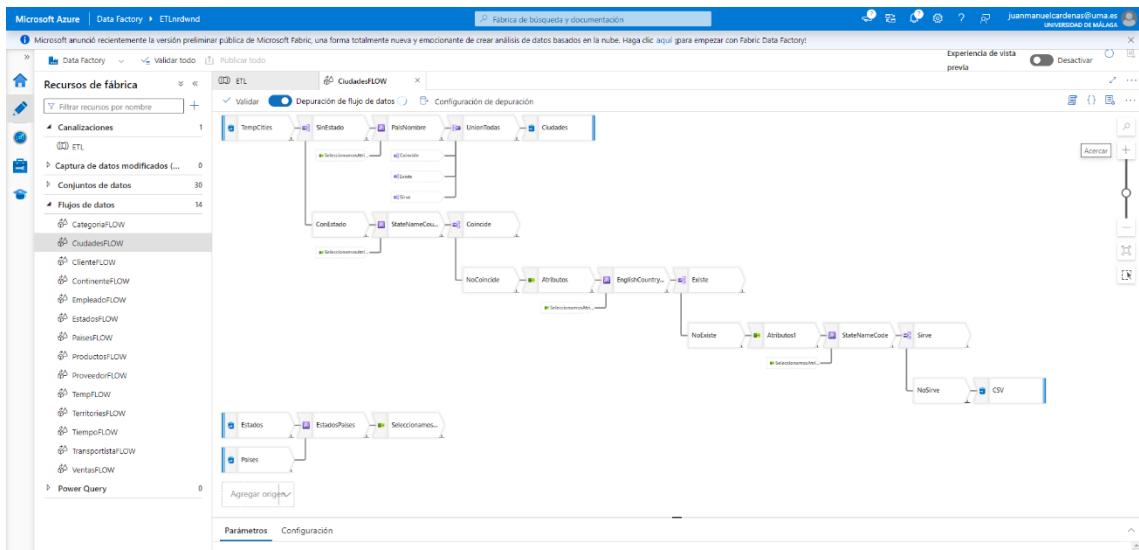
El siguiente paso será completar la tabla de Ciudades, que aquí ya cambia un poco más la cosa, para comenzar debemos usar una tabla que hay en la BD que no está relacionada con ninguna otra que es TempCities, en la cual vamos a cargar el contenido de las ciudades que hay en el archivo cities.txt que previamente ya había cargado en el contenedor con los otros.

Creamos el conjunto de datos con este fichero, como hicimos con el XLS y XML, la única variación es que como se vió en la primera captura, al crear el dataset para el XLS, no tenemos un formato específico para TXT por lo que tenemos que agregarlo como si fuera un CSV, pero esto no es ningún inconveniente, ya que como se puede ver en la captura de abajo lo único que tenemos que indicar es cómo está especificado el formato de la tabla, que en este caso es separaciones con tabulador, para el cambio de columna y cambio de filas con el retorno de carro, y ya se identificaría el esquema.

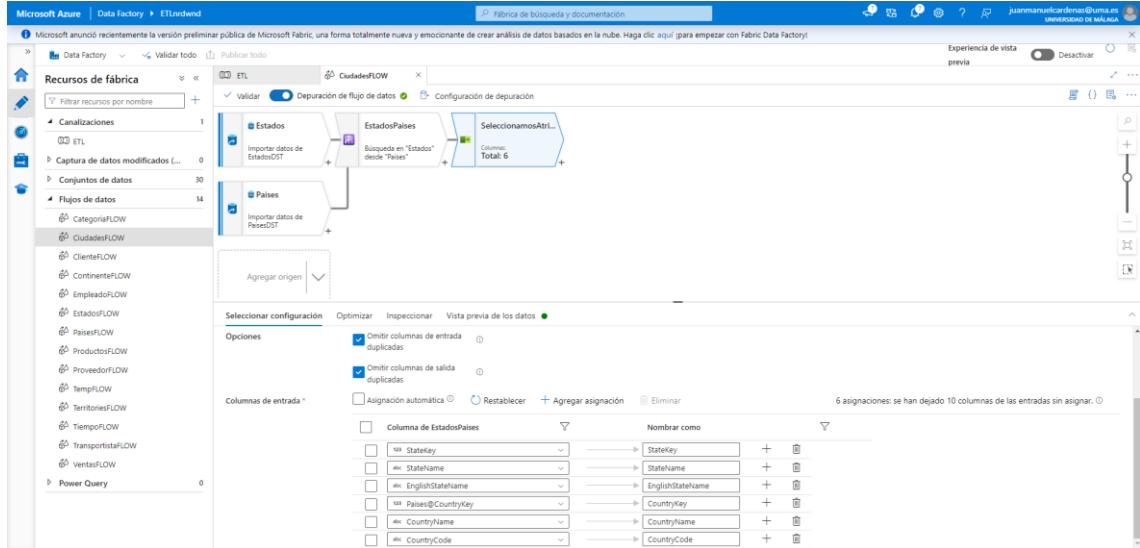
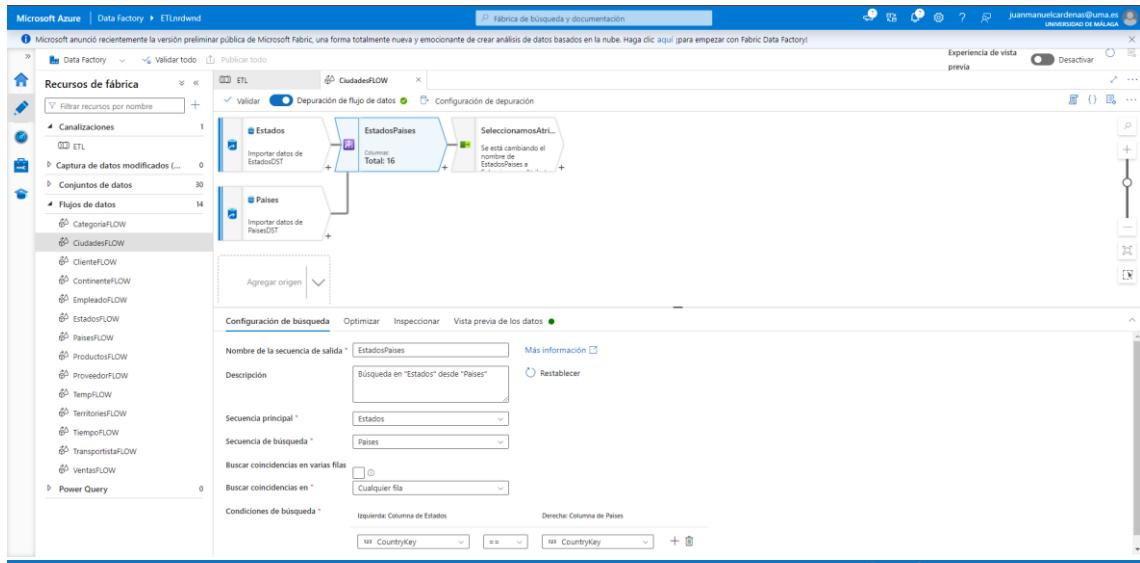
Como en las anteriores ocasiones no necesitamos modificar el tipo de los atributos, por tanto, es tan sencillo como la asignación directa dentro del flujo de datos y ya tendríamos la importación a la tabla TempCities.



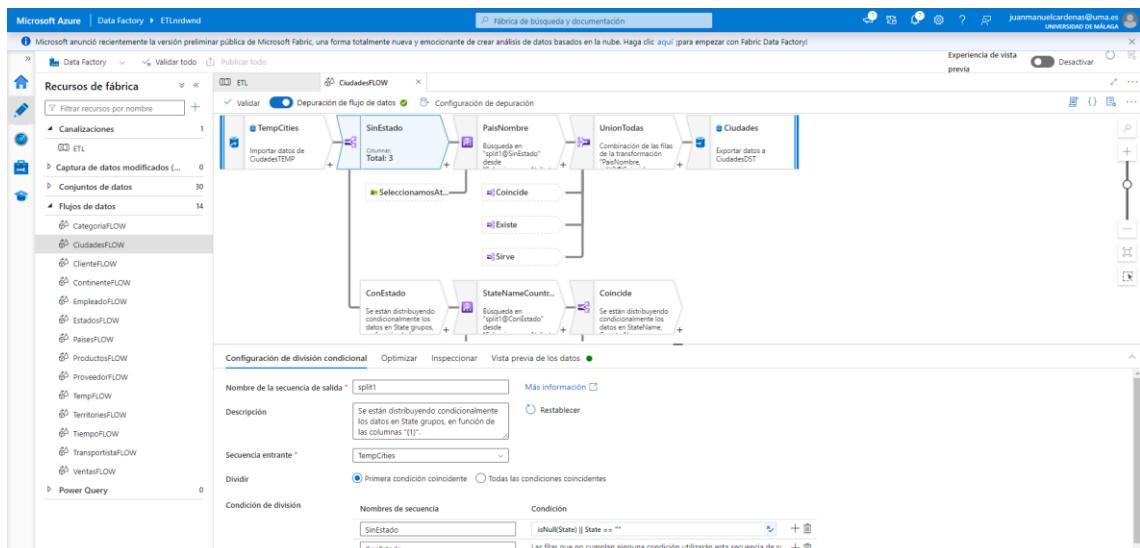
Ahora ya podemos completar la tabla City, como muestro este es el esquema que yo he usado para completar la tabla.



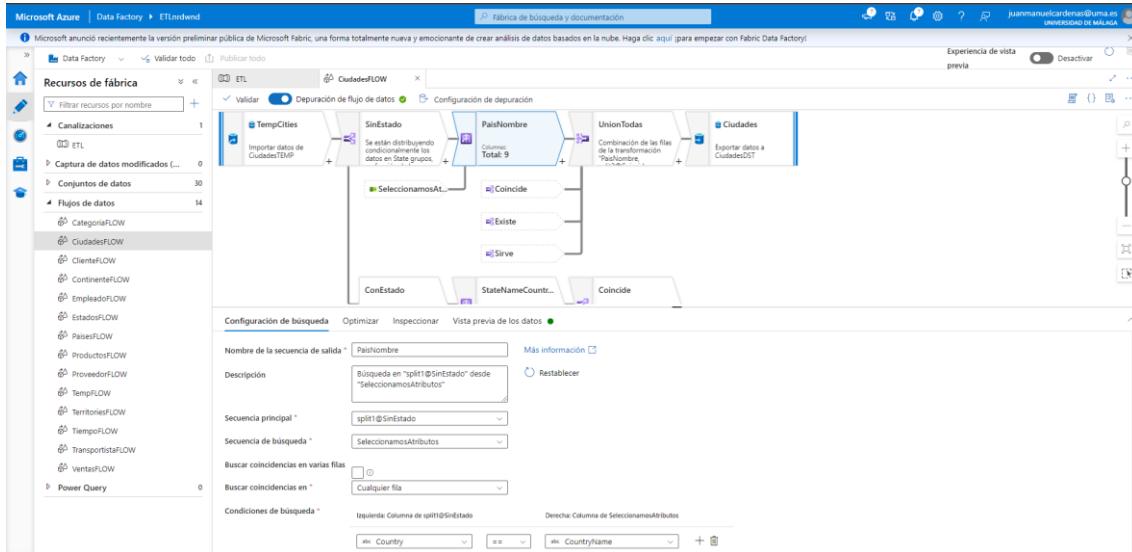
Lo primero que he añadido han sido los orígenes de los datos, que en este caso necesitamos la tabla TempCities que se añadió antes, junto con Estados y Paises, para poder obtener los StateKey y CountryKey necesarios para City, por tanto, lo que he hecho ha sido usar la búsqueda para obtener el estado junto al país que pertenece, ya que este operador la tabla que usamos como principal la añade completa, mientras la secundaria, que es la que usamos como búsqueda, sólo le añade sus atributos a aquellos que cumplen la condición, y debido a que el resultado da la unión de las columnas de ambas, las que no esos elementos aparecen como NULL, he puesto un seleccionar después para solo tener las que necesitamos en este caso StateKey, StateName, EnglishStateName, CountryKey, CountryName y CountryCode.



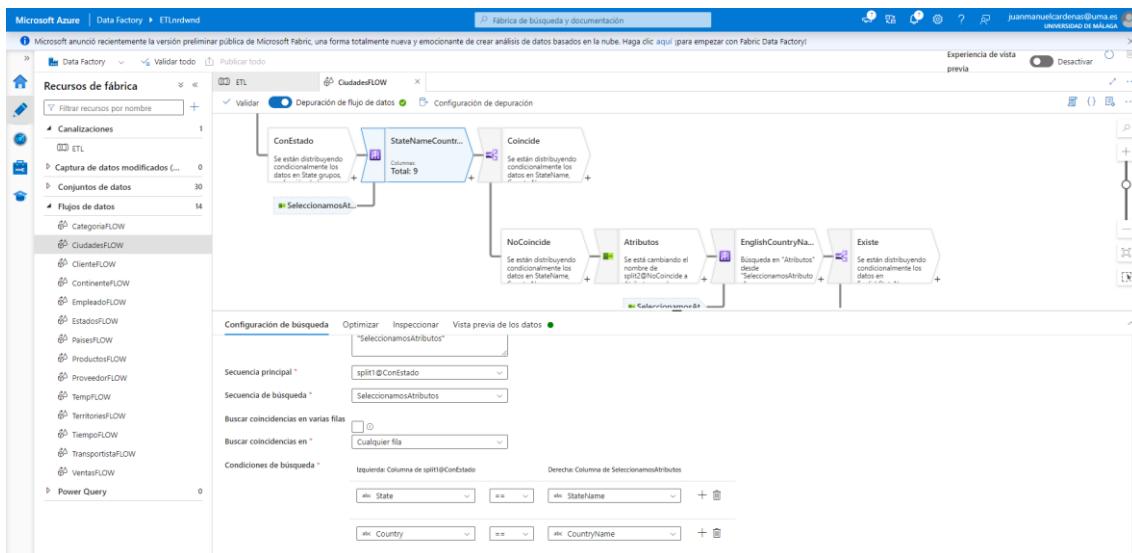
Ahora continuamos con TempCities, que lo que debemos hacer es dividirlo en aquellos que tienen estado de los que no mediante una división condicional.



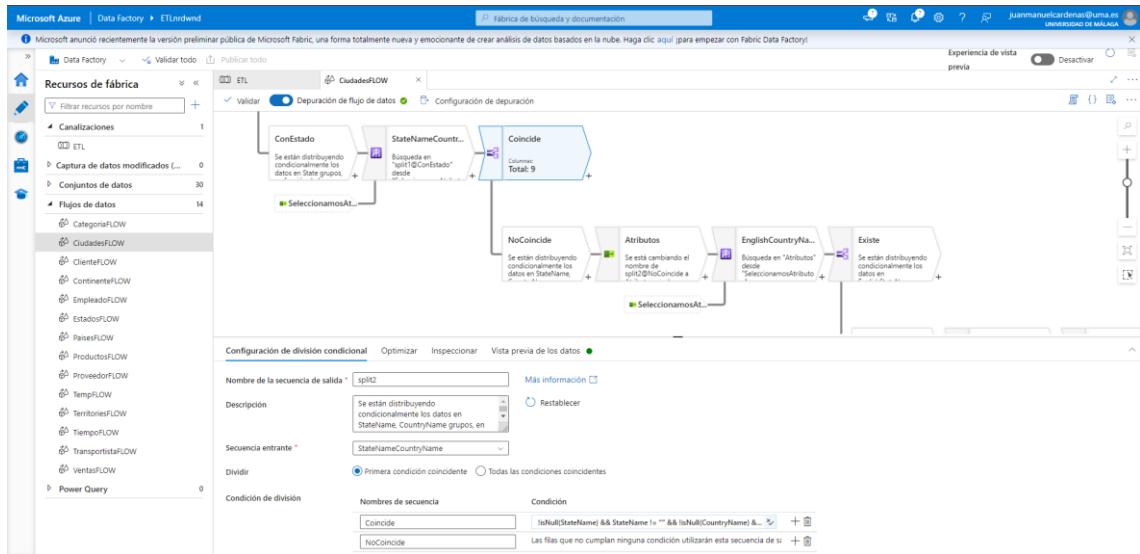
Tras ello, para encontrar a que país pertenece esa ciudad, buscamos en el resultado del seleccionar anterior que CountryName coincide con el nombre del país que tiene la tabla TempCities, este proceso se repetirá de forma similar a lo largo de este proceso.



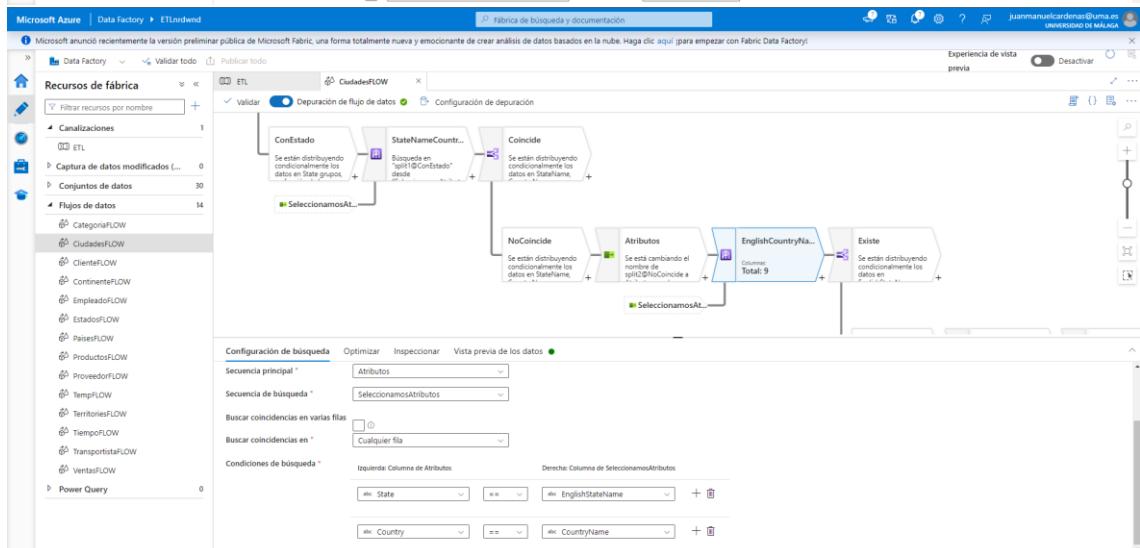
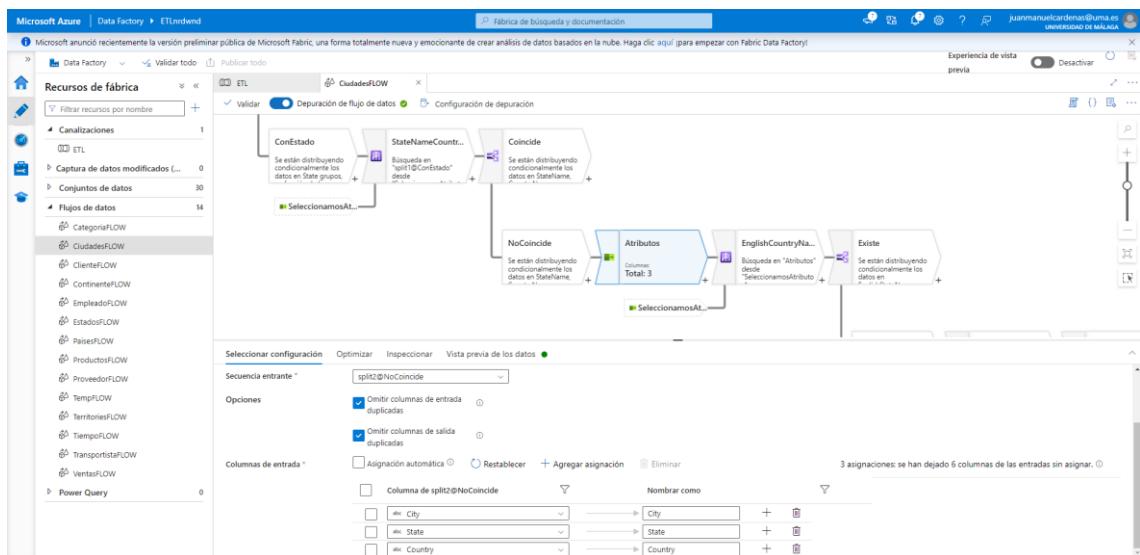
Ahora para las que poseen estado lo que debemos realizar en un primer momento es buscar cuales paises coinciden su StateName y CountryName, con el estado y país que tiene el la tabla dicha ciudad.



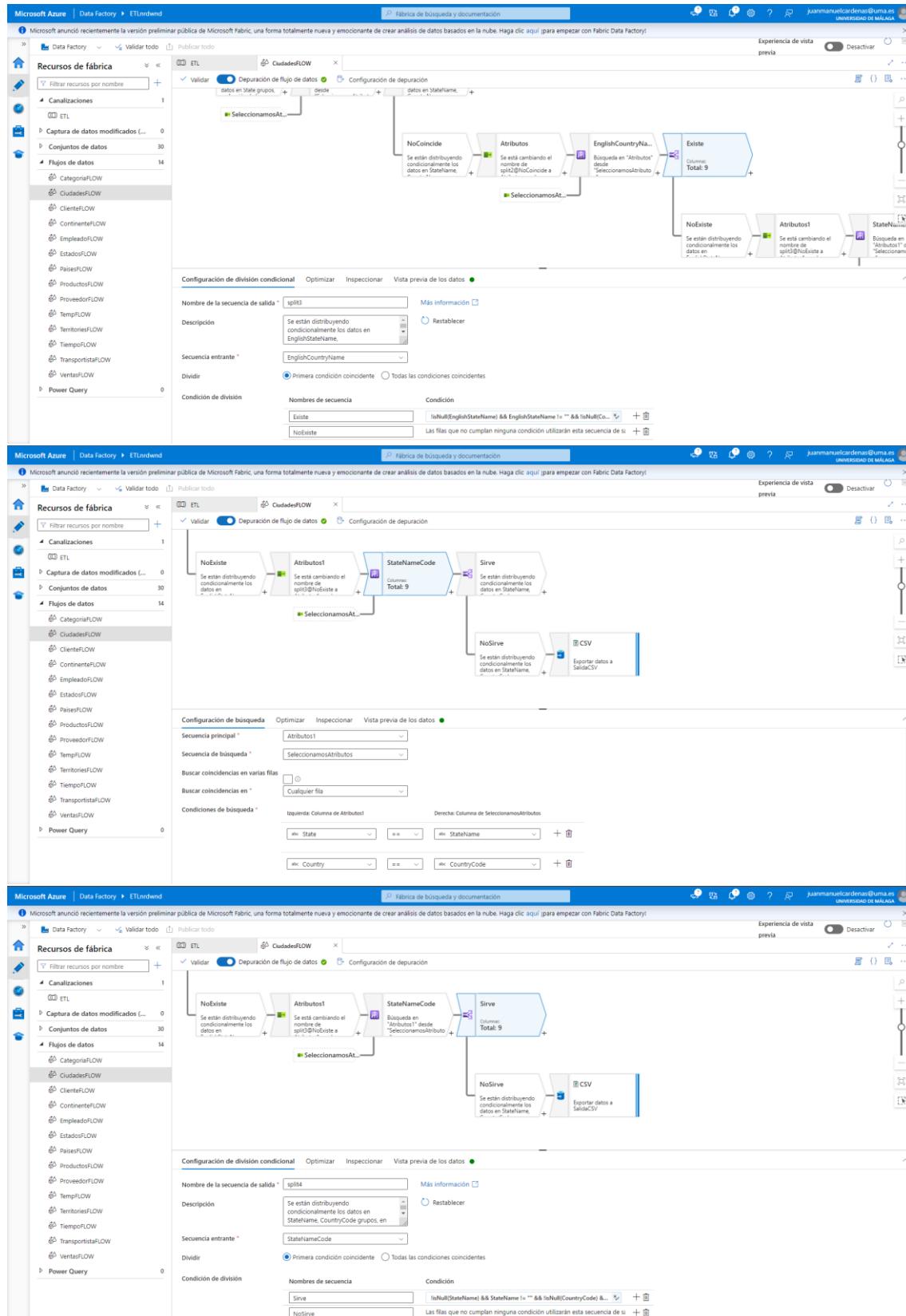
Como el operador búsqueda no nos da una opción directa de separar los resultados, si no como se mencionó antes la diferencia de que cumpla la condición o no, es que el atributo que estamos buscando aparezca como NULL o con un dato si existe, tenemos que añadir manualmente un divisor condicional para separarlos.



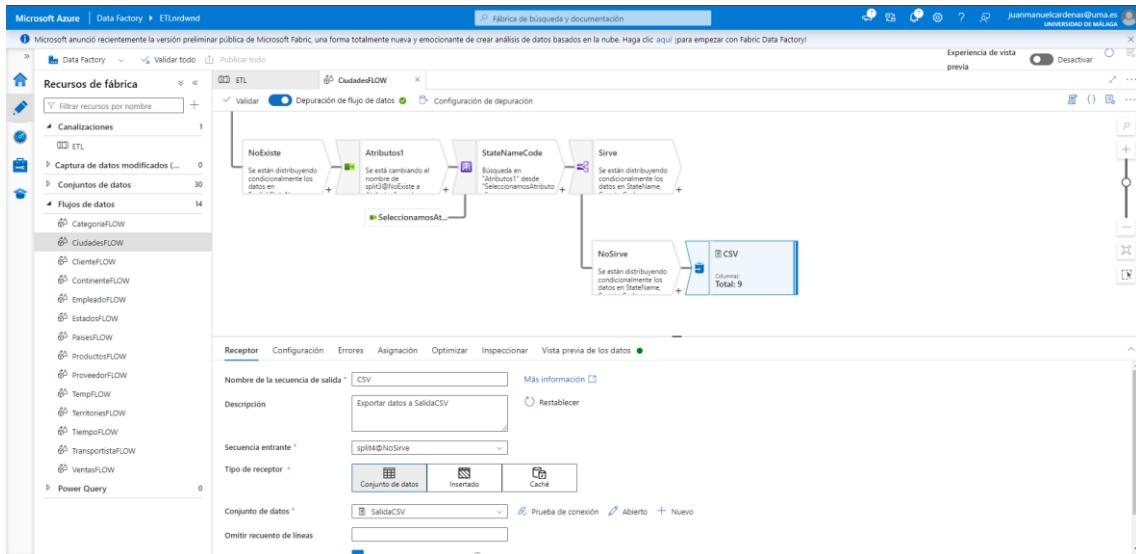
Los que no hayan cumplido esto realizamos una nueva búsqueda con el EnglishStateName y CountryName, y he usado el operador seleccionar de nuevo para eliminar columnas innecesarias que en un futuro nos pueden ocasionar errores.



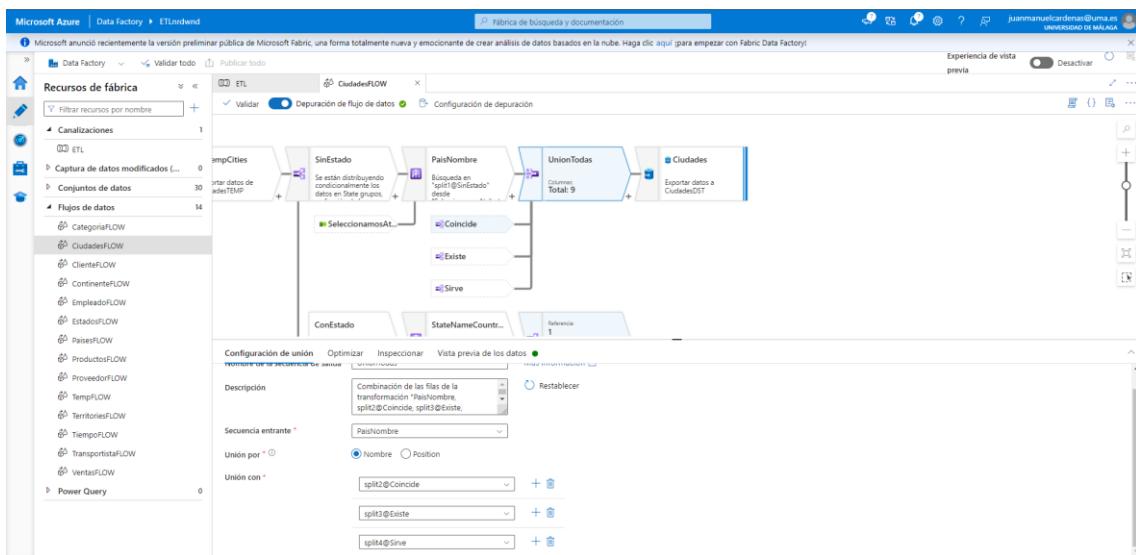
Ahora hacemos de nuevo el proceso de las que no hayan coincidido pero con el StateName y CountryCode.

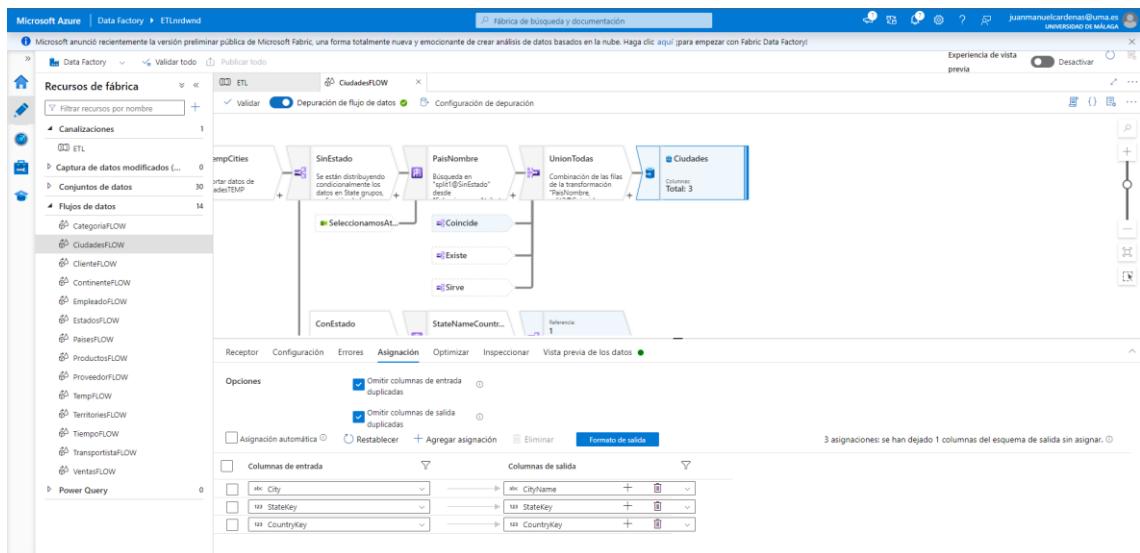


Para tener un control de si el proceso se ha realizado correctamente ponemos un destino al contenedor que creamos al principio que nos creará un archivo CSV con los datos que no han coincidido con ninguno de los criterios anteriores.

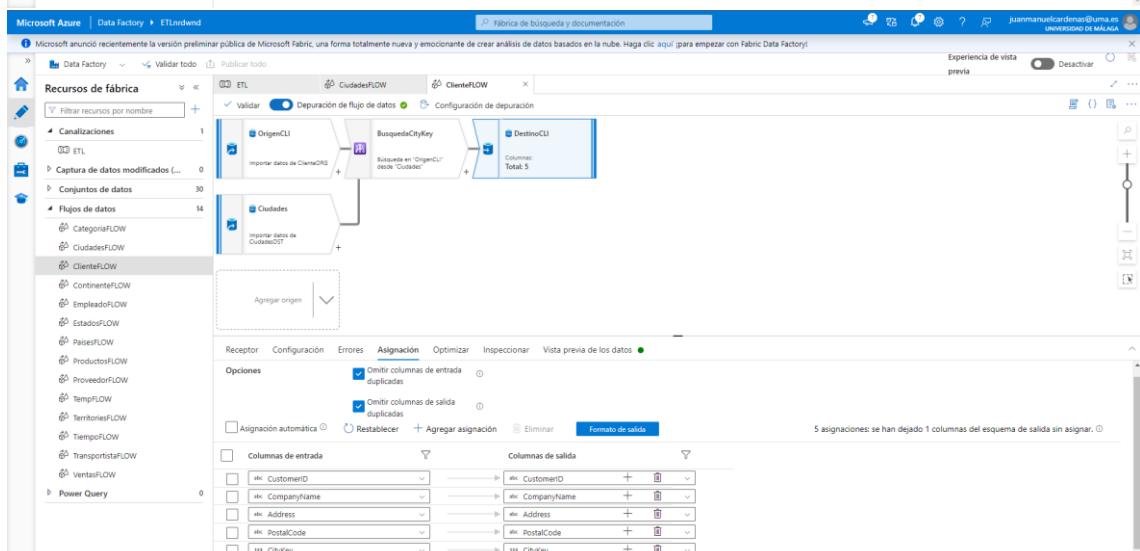
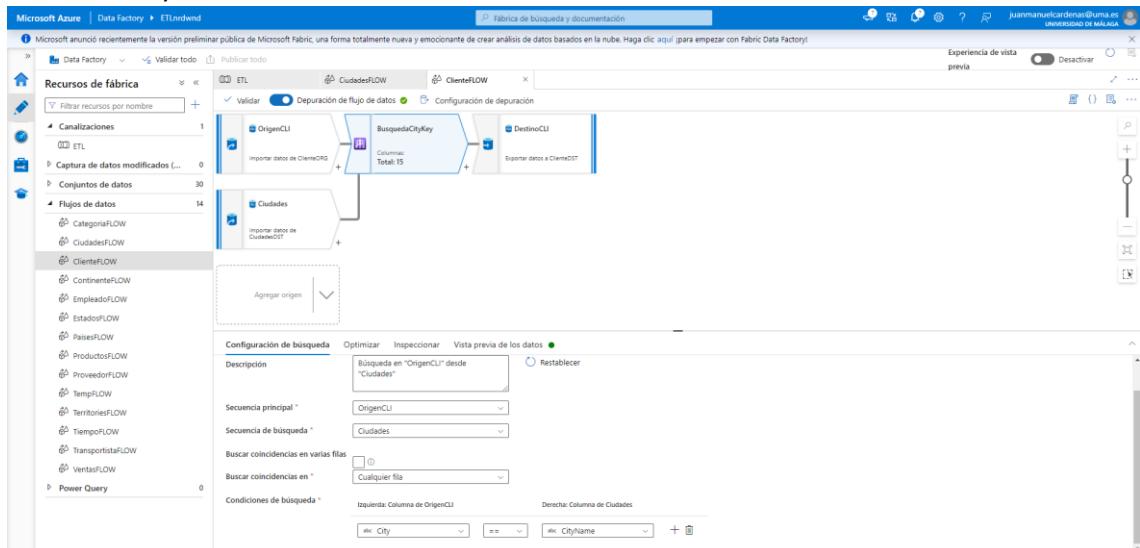


Las que si han coincidido, las vamos a usar el operador union para añadirlas definitivamente a la tabla City.

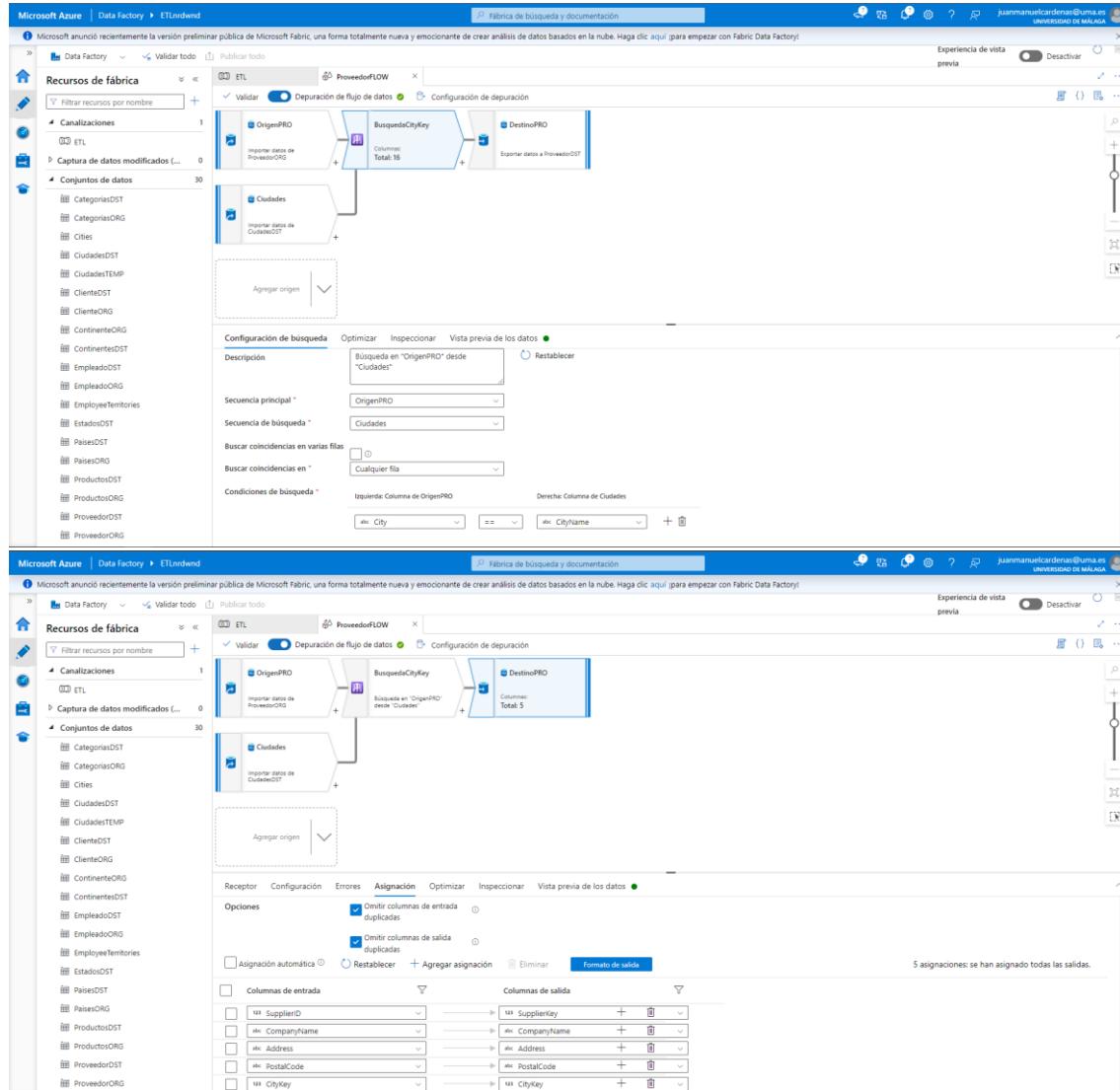




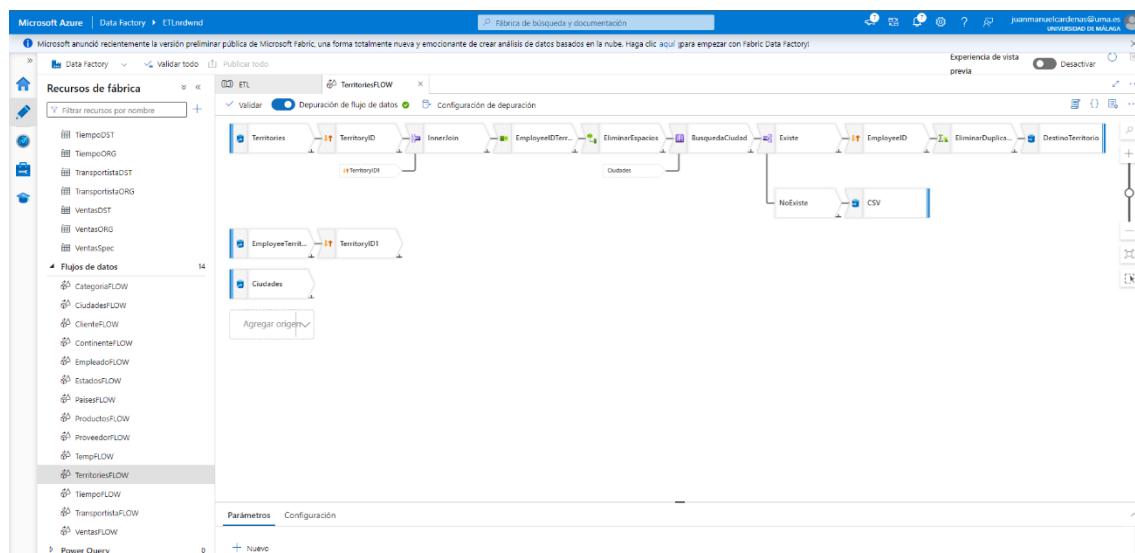
Ya tenemos la tabla completada, ahora continuaremos con Cliente, que para la cual lo único que necesitamos es un origen a la tabla Customers de la BD y buscar en la tabla que hemos creado de City el nombre de la ciudad que coincide con el dato que tiene Customer, para así extraer su key.



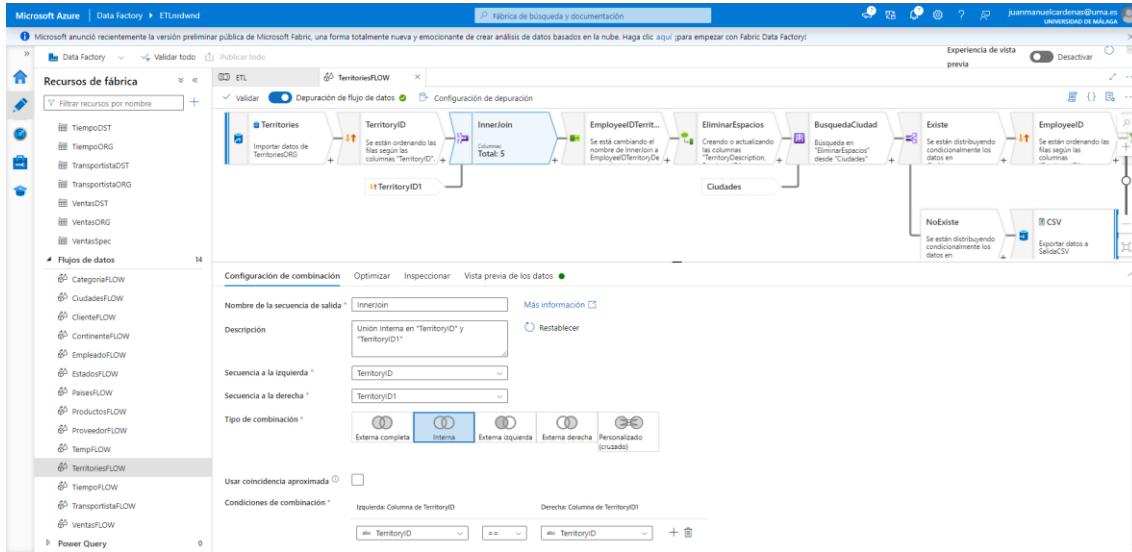
Ahora, pasamos a proveedor que se hace igual que Customer.



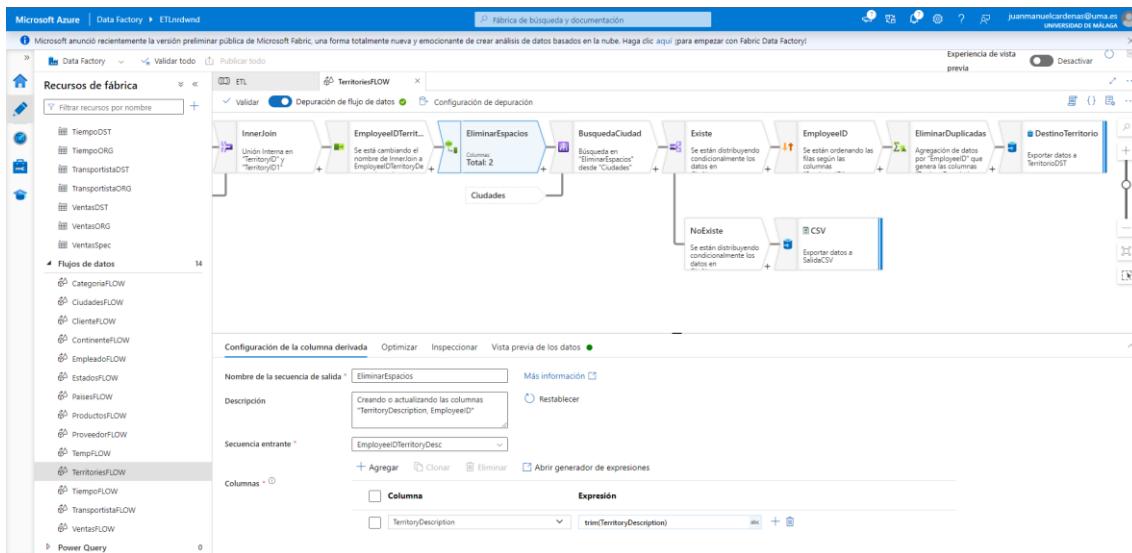
Ya solo nos faltan dos tablas la de hechos Sales y Territories, por tanto, vamos realizar esta última, para comenzar vamos a necesitar como origen de datos las tablas Territories y EmployeeTerritory de la BD, que ambas las vamos a ordenar por TerritoryID.



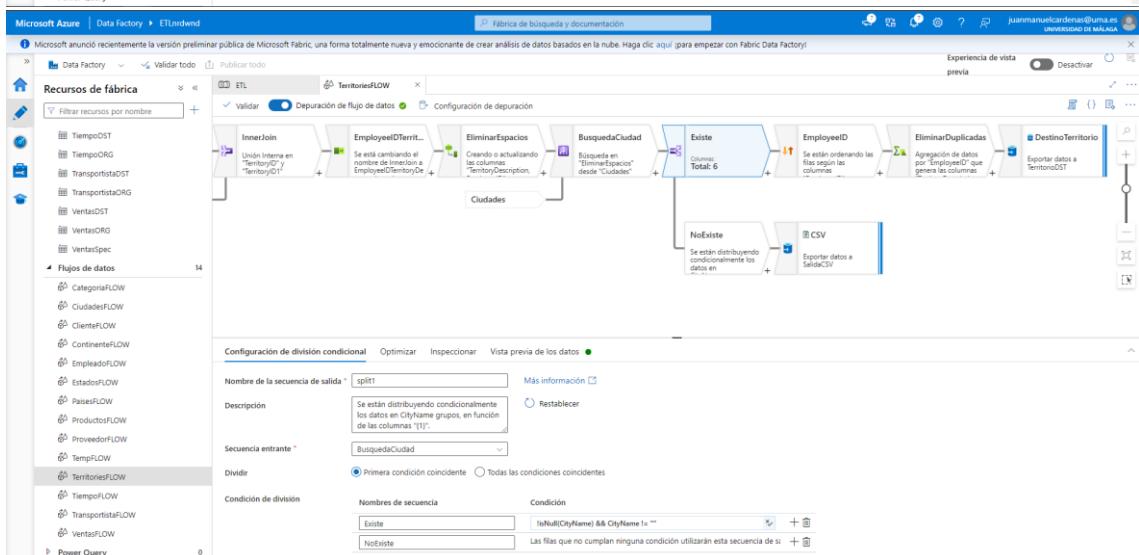
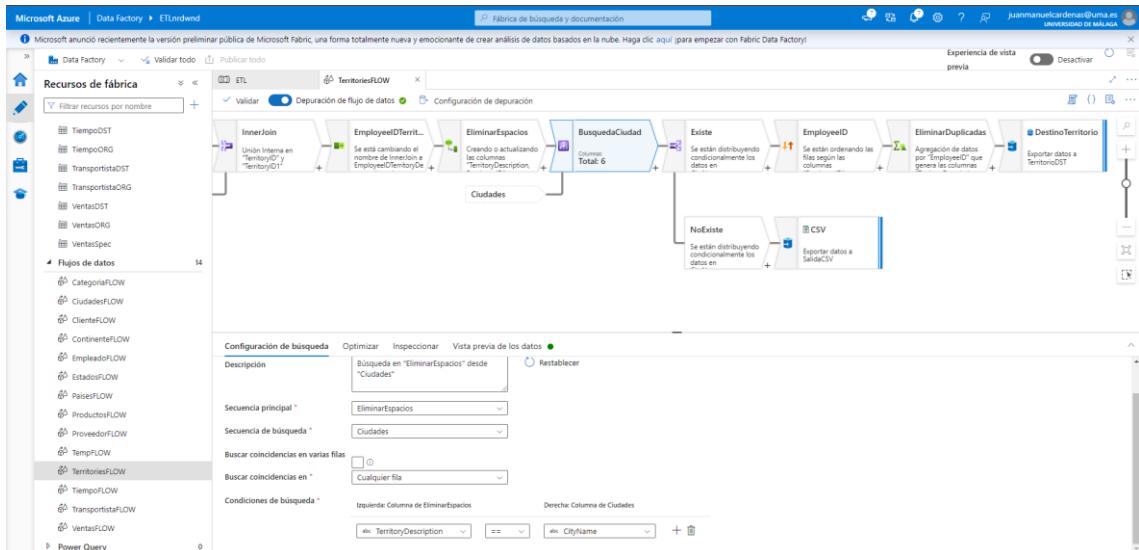
Después de la ordenación lo que haremos es un Inner Join de ambas tablas por TerritoryID y después seleccionamos (operador select) los dos únicos tributos que nos interesan que es EmployeeID y Territory Description que es igual que CityName.



Ahora por errores que nos puede ocasionar el formato en el que está Territory Description vamos a añadir un operador de columna derivada en la cual vamos a poner la función TRIM() para eliminar los espacios que contiene.

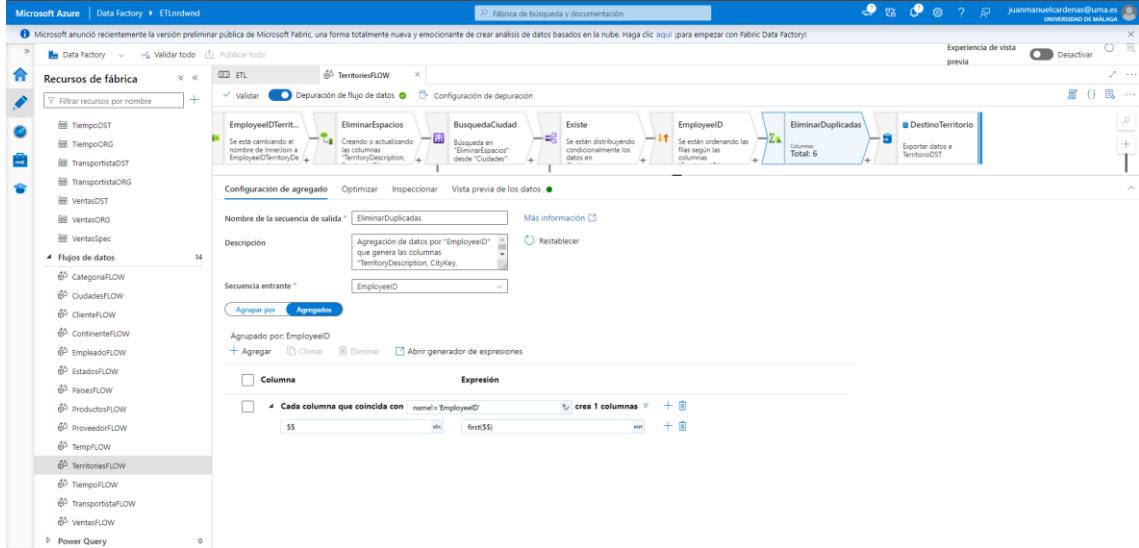
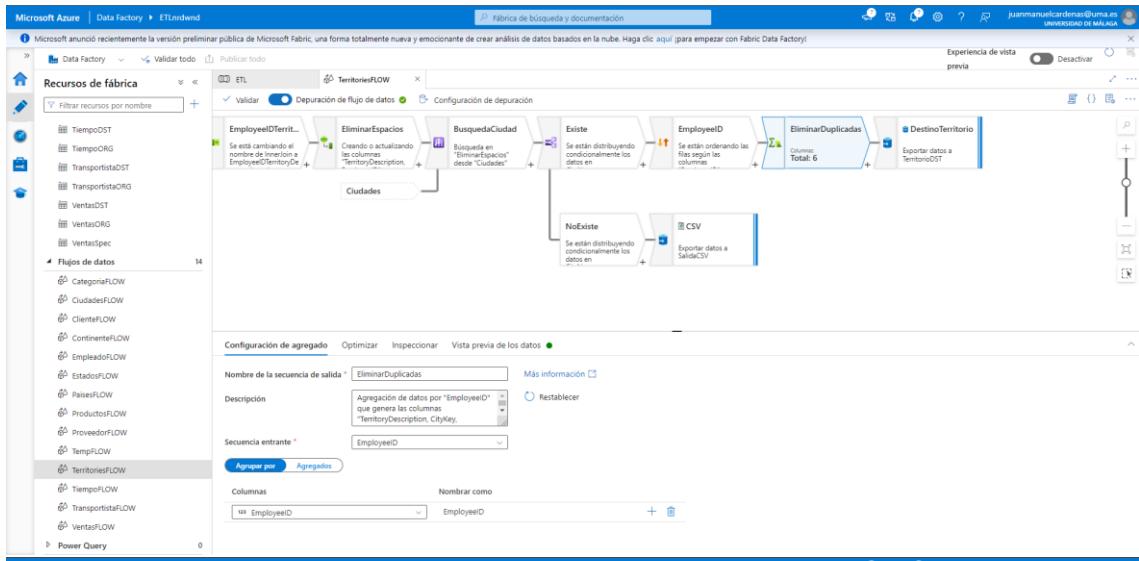


Ahora ya si podemos realizar la búsqueda del con la tabla del AD de ciudades, que debemos añadirla como origen de datos, la condición es que CityName y Territory Description sean iguales, para así poder tener la CityKey, las que no coincidan haremos como antes y la enviaremos a un fichero CSV en nuestro contenedor.

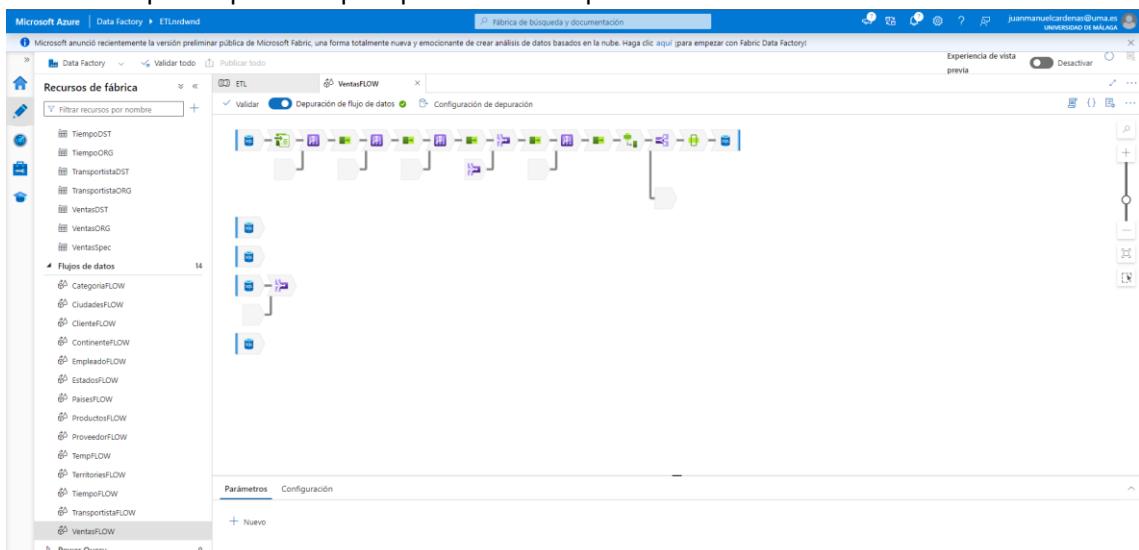


Ahora si vamos a la vista previa de los datos, podemos observar que hay empleados que su ID aparece en varias ocasiones esto nos ocasionaría errores puesto que una persona no puede estar registrada en más de una ubicación, por tanto, para solucionar esto vamos a eliminar las columnas repeditas, ordenando antes por EmployeeID.

Para llevar a cabo esto, en Data Factory debemos añadir el operador agregado, y agrupar por EmployeeID, y en agregado agregamos un patrón de columna en el cual pondremos que el name debe ser distinto de Employee ID y el operador first(\$\$) indica que se quede con el primer resultado que encuentre,y así eliminamos la duplicidad.

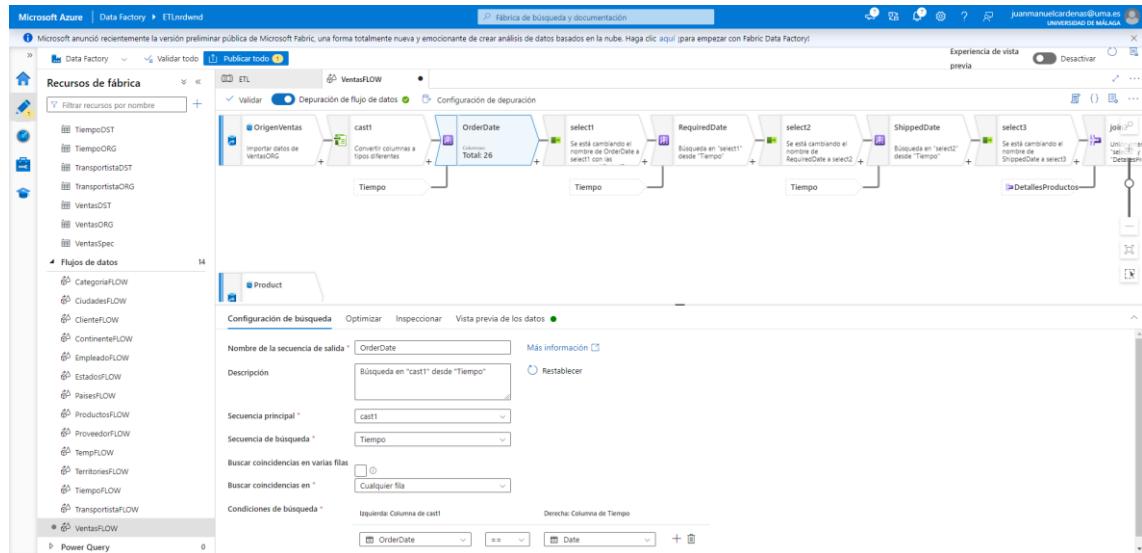
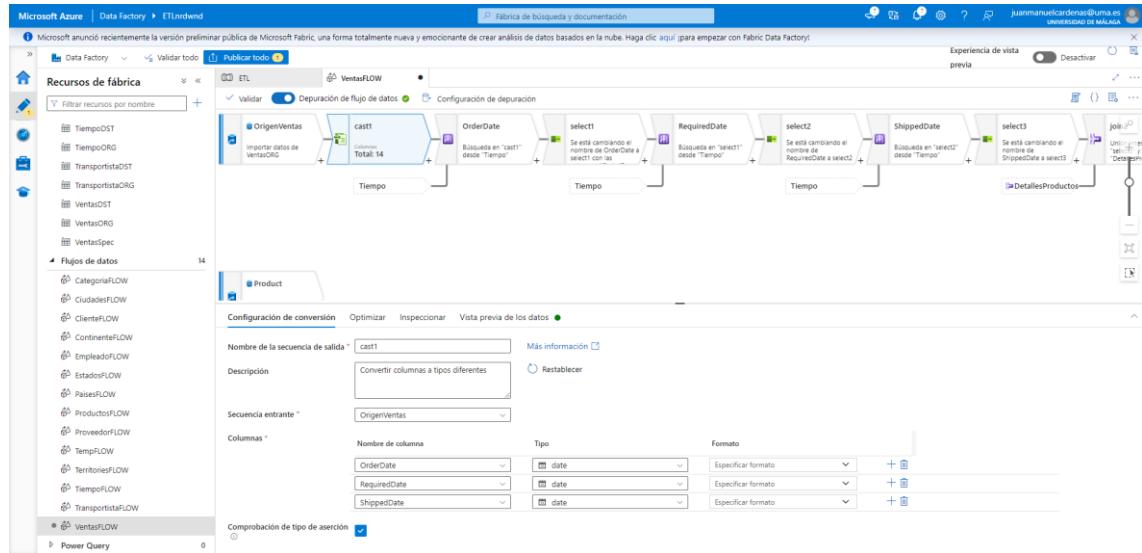


Y ya tendríamos Territory completada ahora solo nos queda el hecho Sales, que debido a los numerosos pasos que tiene para poder ver el esquema se ve así.

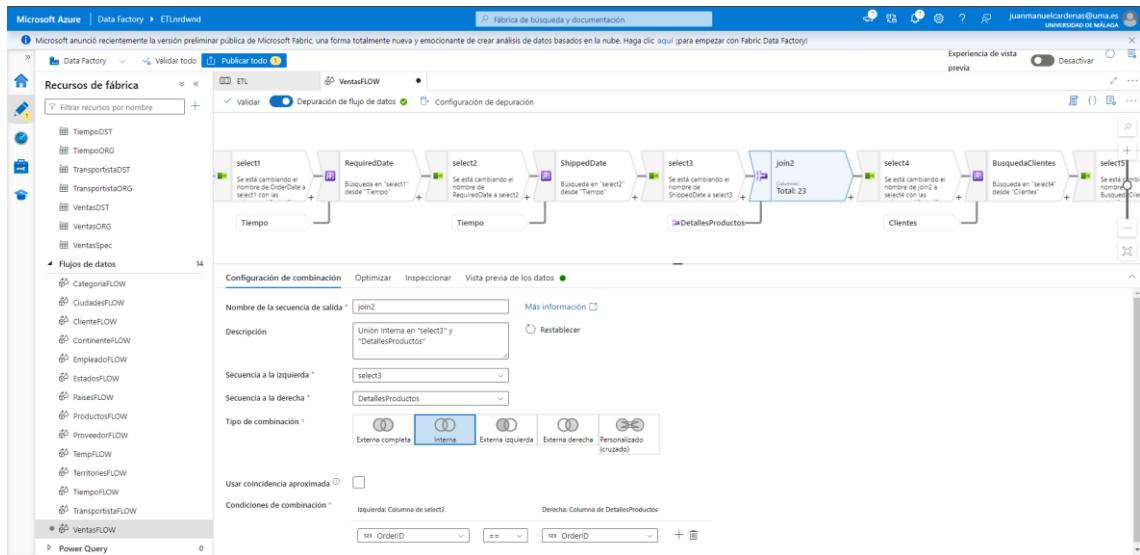
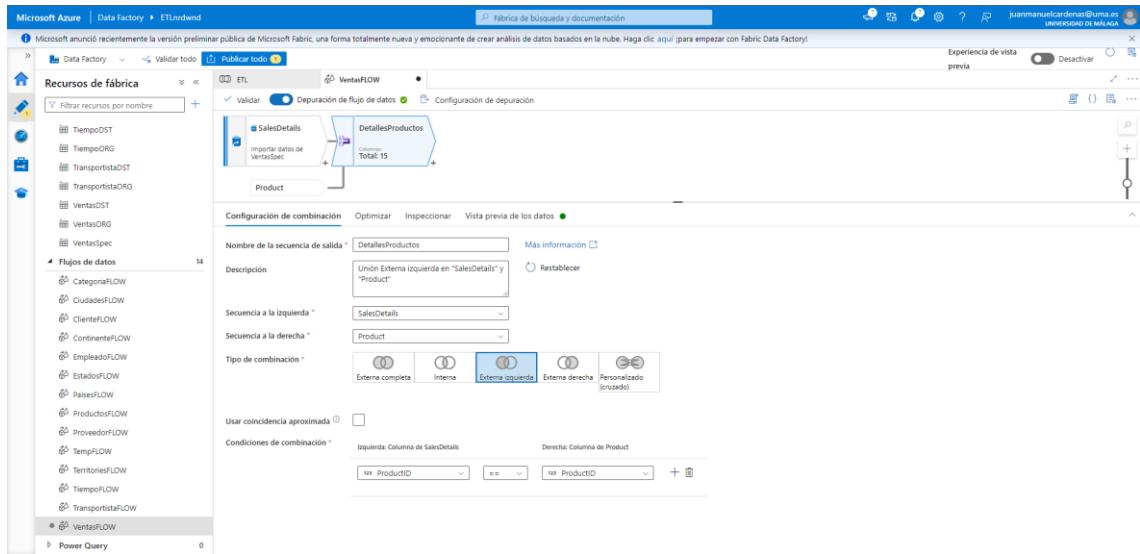


Principalmente vamos a trabajar con las tablas de Orders y Orders Details de la BD que contiene la mayoría de la información que necesitamos, yo he comenzado cambiando el tipo

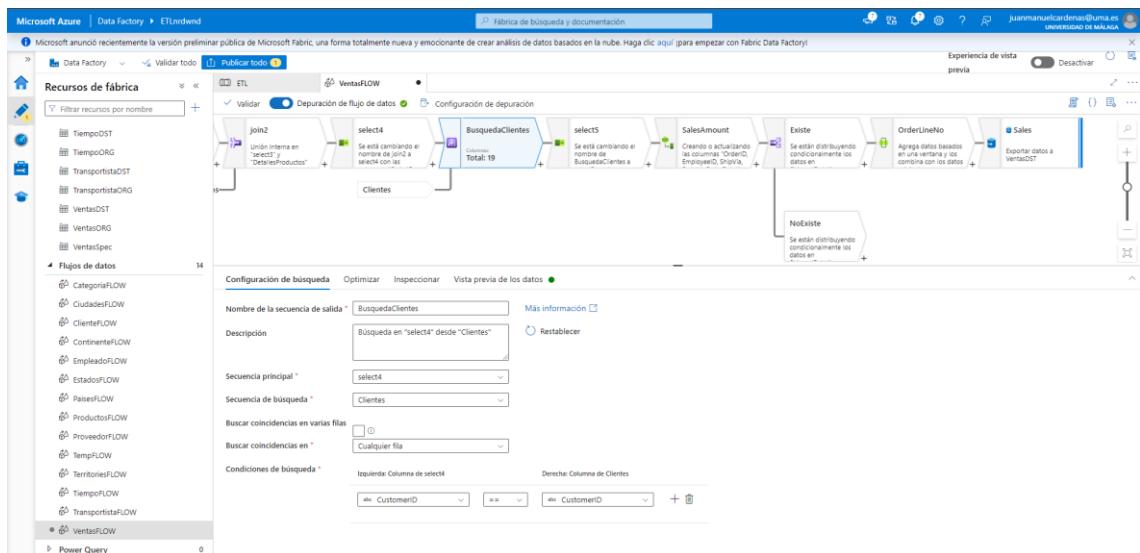
de fecha a los atributos OrderDate, RequiredDate y ShippedDate, para compararlos con los datos de la tabla Tiempo, que tenemos que añadirla como source, que tenemos en el AD y así extraer su key.



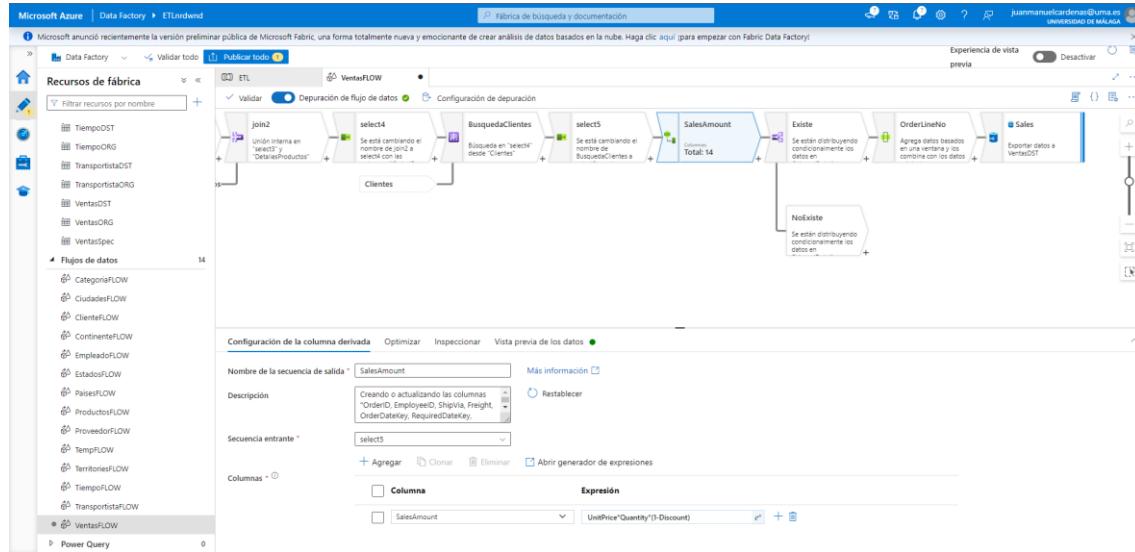
Este proceso se realiza una vez por cada fecha que tenemos, y los seleccionar posteriores a la búsqueda los he añadido, como hice anteriormente eliminar columnas innecesarias, ya que se agregan todas las que tiene la tabla tiempo y sólo necesitamos el DateKey, ahora para obtener los datos como ProductKey, SupplierKey, UnitPrice, etc. Hacemos un left outer join de las tablas Producto y Order Details de la BD, usando el atributo ProductID, para usarlo posteriormente en un inner join del último select al terminar la búsqueda de las fechas con el, usando OrderID.



De nuevo hacemos select para eliminar columnas que no vamos a necesitar agregar al hecho Sales, y después hacemos una búsqueda en nuestra tabla de clientes del AD, usando ClientID, para obtener el ClientKey, y después filtramos de nuevo con el select.

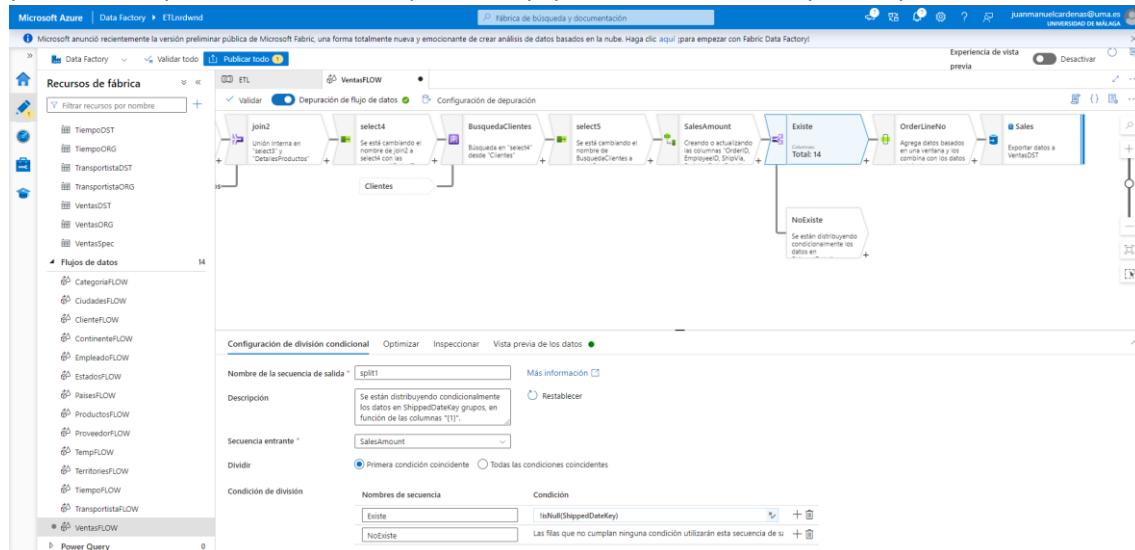


Ahora el siguiente dato que necesitamos es SalesAmount (Cantidad vendida), que para calcularlo se añade una columna derivada y usamos los datos de las columnas UnitPrice, Quantity y Discount para calcularlo.

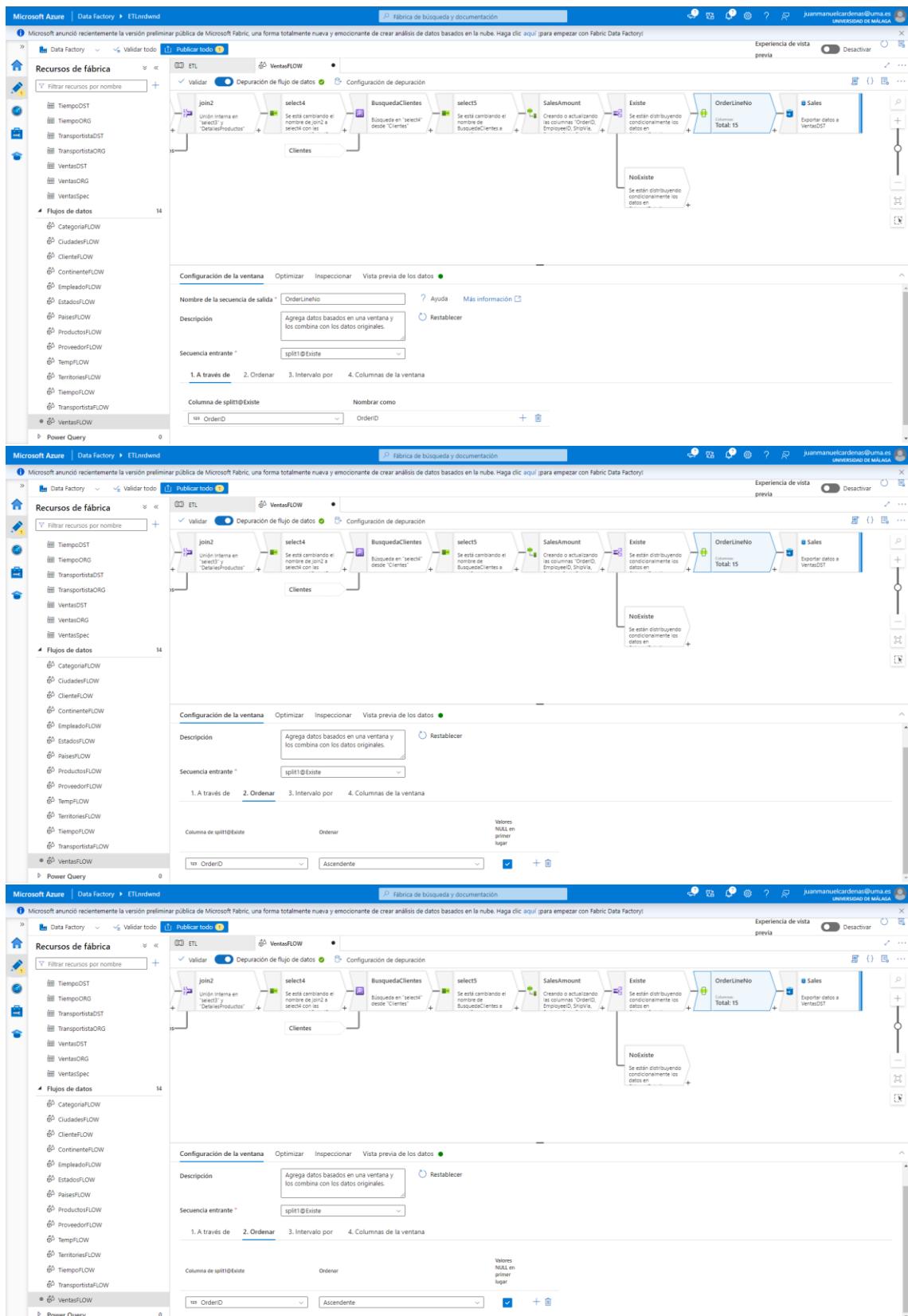


Ahora solo nos falta corregir un error que nos puede ocasionar la columna ShippedDate y agregar el atributo OrderLineNo, que es el identificador que tiene cada artículo dentro de un pedido.

La columna ShippedDate puede aparecer como NULL, y en el AD no tenemos modelado que pueda serlo, por tanto, tenemos que filtrar y quedarnos solo con aquellas que tienen.



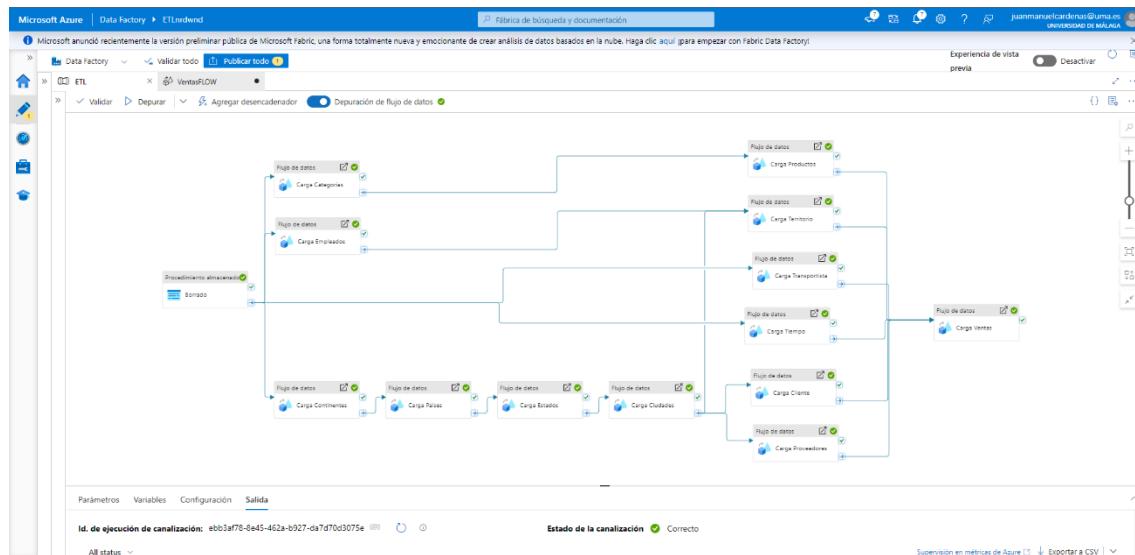
Ahora para crear el OrderLineNo, yo he usado el operador ventana que es el que nos permite usar la función rownum(), por tanto, he agrupado y ordenado por OrderID, y OrderLineNo es el resultado que de la función, esto hace de contador que va aumentando hasta que el OrderID cambie, por eso hemos agrupado por el, y cuando lo hace el contador se resetea a 1 de nuevo.



Y ya tendríamos todas las columnas necesarias.

The screenshot shows the Microsoft Azure Data Factory interface for the 'VentasFLOW' pipeline. The 'Asignación' (Assignment) tab is active, displaying the mapping of columns from the source to the destination. The 'Formato de salida' (Output Format) section indicates that all outputs have been assigned. The 'Flujos de datos' (Data Flows) section lists 14 flows, including 'CategoríaFLOW', 'CiudadesFLOW', 'ClienteFLOW', 'ContinenteFLOW', 'EmpleadosFLOW', 'EstadosFLOW', 'PaísesFLOW', 'ProductosFLOW', 'ProveedoresFLOW', 'TempFlow', 'TerritoriosFLOW', 'TiempoFLOW', 'TransportistaFLOW', and 'VentasFLOW'. The 'Power Query' section shows 0 flows.

Ahora vamos a ejecutar el Pipeline que se encuentras todos los flujos de datos para ver que todo está bien.



Y el hecho en el almacén de datos se vería así.

```

SELECT TOP (1000) /*CustomerKey*/
       [CustomerKey]
      ,[CustomerName]
      ,[Address]
      ,[City]
      ,[PostalCode]
      ,[Country]
      ,[Phone]
      ,[Fax]
      ,[EmployeeKey]
      ,[OrderDateKey]
      ,[DueDateKey]
      ,[ShippedDateKey]
      ,[ProductKey]
      ,[SupplierKey]
      ,[OrderLineNo]
      ,[UnitPrice]
      ,[Quantity]
      ,[Discount]
      ,[SalesAmount]
      ,[Freight]
  FROM [dbo].[Sales]
  
```

CustomerKey	EmployeeKey	OrderDateKey	DueDateKey	ShippedDateKey	ProductKey	SupplierKey	OrderNo	OrderLineNo	UnitPrice	Quantity	Discount	SalesAmount	Freight
1	766	2	21454	21482	21473	1	67	16	10502	3	14.00	30	0
2	752	8	21508	21626	21608	2	22	9	10551	1	21.00	20	0.25
3	752	8	21598	21625	21609	2	18	9	10551	2	9.95	20	0.25
4	773	8	21804	21832	21809	2	27	14	10579	1	12.50	24	0
5	772	8	21804	21832	21809	2	27	11	10579	2	43.90	30	0
6	772	8	21804	21832	21809	2	12	5	10579	3	38.00	20	0
7	772	8	21804	21832	21809	2	7	3	10579	4	30.00	18	0
8	809	4	21199	21227	21207	1	6	23	10507	1	14.40	10	0.15
9	809	4	21199	21237	21207	1	40	19	10507	2	14.70	50	0
10	809	4	21199	21227	21207	1	59	26	10507	3	44.00	70	0.15
11	739	8	21321	21363	21353	2	65	2	10366	1	16.80	5	0
12	739	8	21321	21363	21353	2	77	12	10366	2	10.40	5	0
13	818	8	21597	21625	21599	1	28	12	10541	1	36.40	32	0
14	818	8	21597	21625	21599	1	11	5	10541	2	16.80	6	0.64
15	737	3	21402	21510	21496	2	76	23	10530	1	10.00	50	0
16	737	3	21402	21510	21496	2	43	20	10530	2	46.00	25	0
17	737	3	21402	21510	21496	2	61	29	10530	3	28.50	20	0
18	737	3	21402	21510	21496	2	17	7	10530	4	35.00	40	0
19	803	7	21402	21511	21496	3	30	13	10532	1	25.89	10	0

Y el procedimiento almacenado del principio se estructuraría de la siguiente forma.

```

SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
/*
-- Author: Customer , Name
-- Create Date: <create Date , >
-- Description: <Description , >
*/
ALTER PROCEDURE [dbo].[Borrar]
AS
BEGIN
    Delete From dbo.Sales
    Delete From dbo.Territories
    Delete From dbo.Employee
    Delete From dbo.Customer
    Delete From dbo.City
    Delete From dbo.Region
    Delete From dbo.Country
    Delete From dbo.Product
    Delete From dbo.Category
    Delete From dbo.Shipping
    Delete From dbo.Customer
    Delete From dbo.Employee
    Delete From dbo.Shipper
    Delete From dbo.Time
    Delete From dbo.Continent
END

```

3. Conclusiones

El aprendizaje de esta nueva herramienta para realizar el ETL me ha sorprendido, la escogí por todas la buenas opiniones que tenía de los diferentes usuarios, pero ha cumplido las expectativas, ya que es sencilla de aprender desde cero además que tiene muchas herramientas que hacen el modelado muy ameno, obviamente, hay otras cosas que tienes que darles más vueltas para implementarla que en Visual Studio, pero si fueran iguales no estaríamos hablando de entornos diferentes, por esto mismo yo a gusto personal me ha gustado más trabajar aquí, habiendo realizado proyectos similares en los dos entornos.

También para comentar existen 4 tablas en el AD que no han sido necesarias, que no suponen ningún problema para la implementación porque no están conectadas con ninguna otra, y como no suponía ningún problema las he dejado sin eliminar, estas son CustomerDemo, CustomerManagement, NewCustomer y Salestest, también comentar que tuve que hacer una pequeña modificación en el flujo de datos de Estados y fue añadir una división condicional ya que reconocía un estado que no tenía ningún dato.