

2024

# S30 - EVIDENCIA DE APRENDIZAJE 3. PROCESO DE TRANSFORMACIÓN DE DATOS Y CARGA EN EL DATA MART FINAL

OSCAR DAVID MADRIGAL FONNEGRA  
BASES DE DATOS II -IUD

VICTOR HUGO MERCADO - PROFESOR |

## **INTRODUCCIÓN**

En el marco del proceso de transformación de datos para Jardinería S.A., se ha llevado a cabo un proyecto integral de Extracción, Transformación y Carga (ETL). Este proceso es fundamental para convertir datos dispersos en fuentes diversas en información coherente y accesible, facilitando la toma de decisiones fundamentadas. En este documento, se detallarán las etapas del proceso de ETL, así como las correcciones y ajustes realizados para garantizar la calidad de los datos resultantes. Este trabajo destaca la importancia de la limpieza, la transformación y la integración de datos en el contexto empresarial actual, donde la información precisa y oportuna es clave para el éxito organizacional.

## **OBJETIVOS**

### **OBJETIVO GENERAL:**

- Hacer las transformaciones necesarias para que los datos queden listos para su posterior carga en el DataMark

### **ANÁLISIS DEL PROBLEMA**

El objetivo general de hacer las transformaciones necesarias para preparar los datos para su carga en el DataMark implica una serie de desafíos y consideraciones importantes. Este proceso de transformación es crucial para garantizar la integridad, coherencia y calidad de los datos que se utilizarán en el DataMark, que a su vez servirá como base para la toma de decisiones empresariales fundamentadas.

En resumen, el objetivo de preparar los datos para su carga en el DataMark requiere un enfoque meticuloso y cuidadoso para garantizar la calidad y coherencia de los datos transformados. Este proceso es fundamental para aprovechar al máximo el potencial de los datos como recurso empresarial estratégico.

# 1. Empezamos con la limpieza de los datos y la primera transformación de la dimensión tiempo

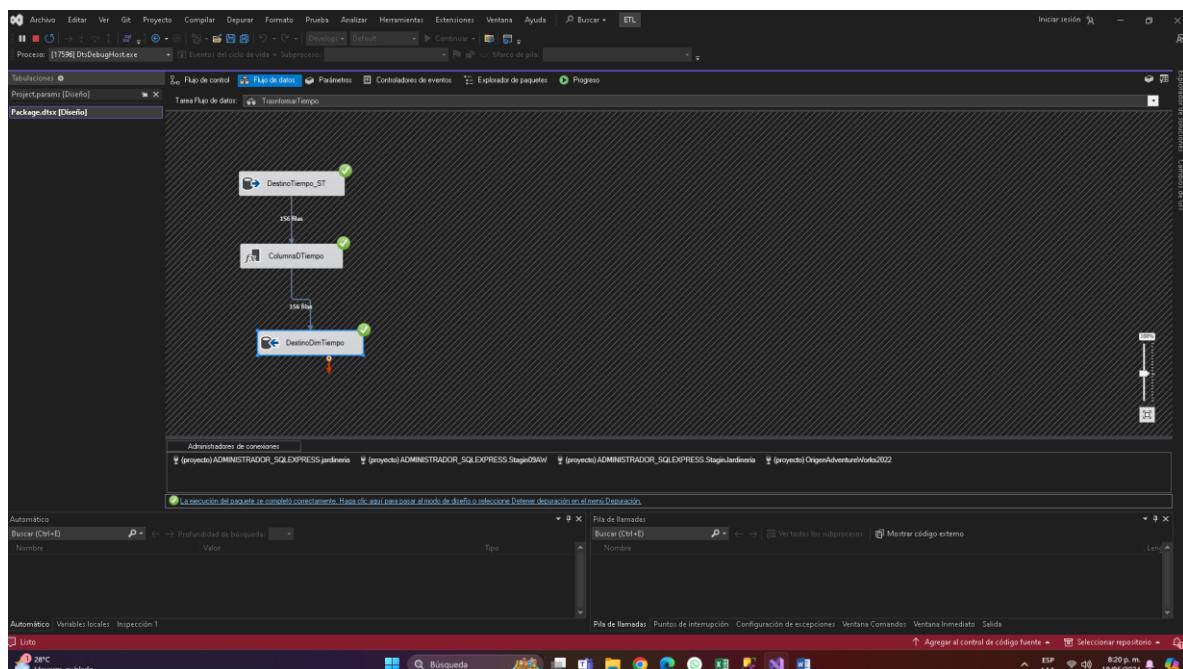
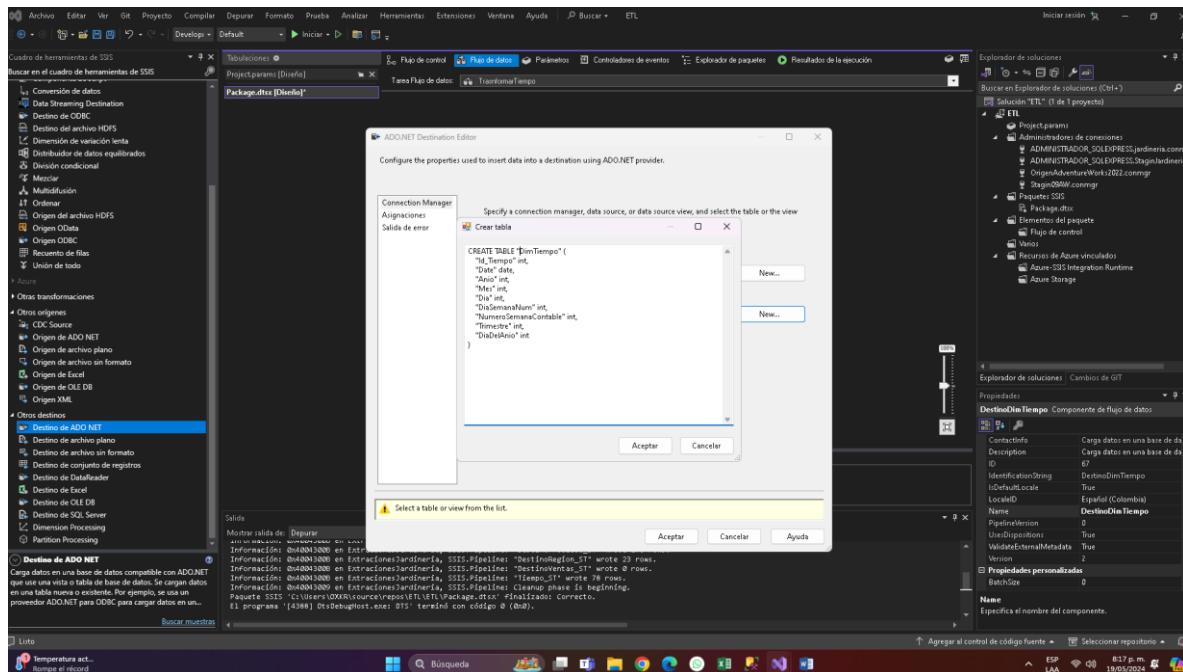
The screenshot shows two instances of the Microsoft SQL Server Data Tools (SSDT) interface, likely running on a Windows 10 desktop. Both windows are titled "Package.dtsx [Diseño]".

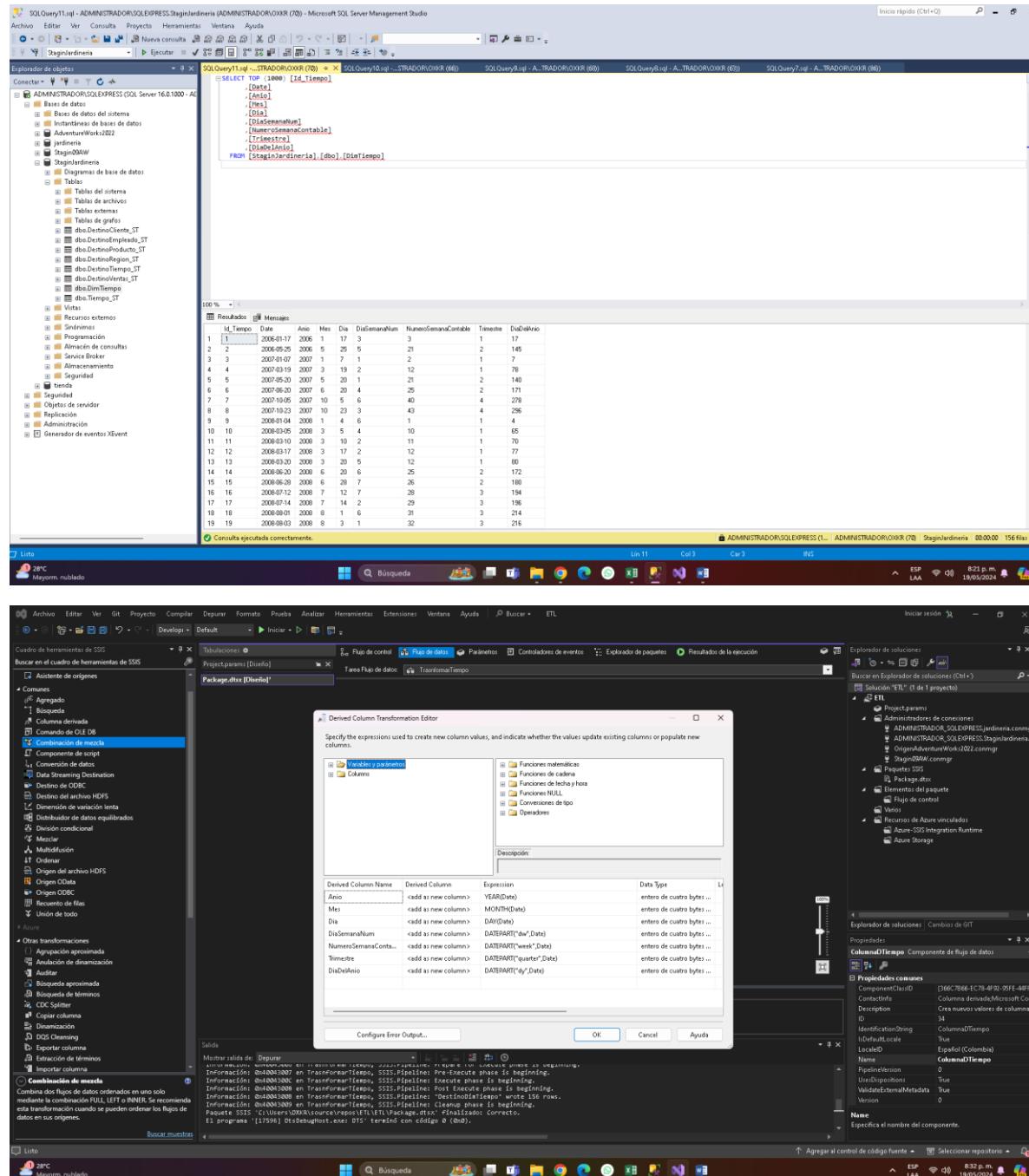
**Top Window (Screenshot 1):**

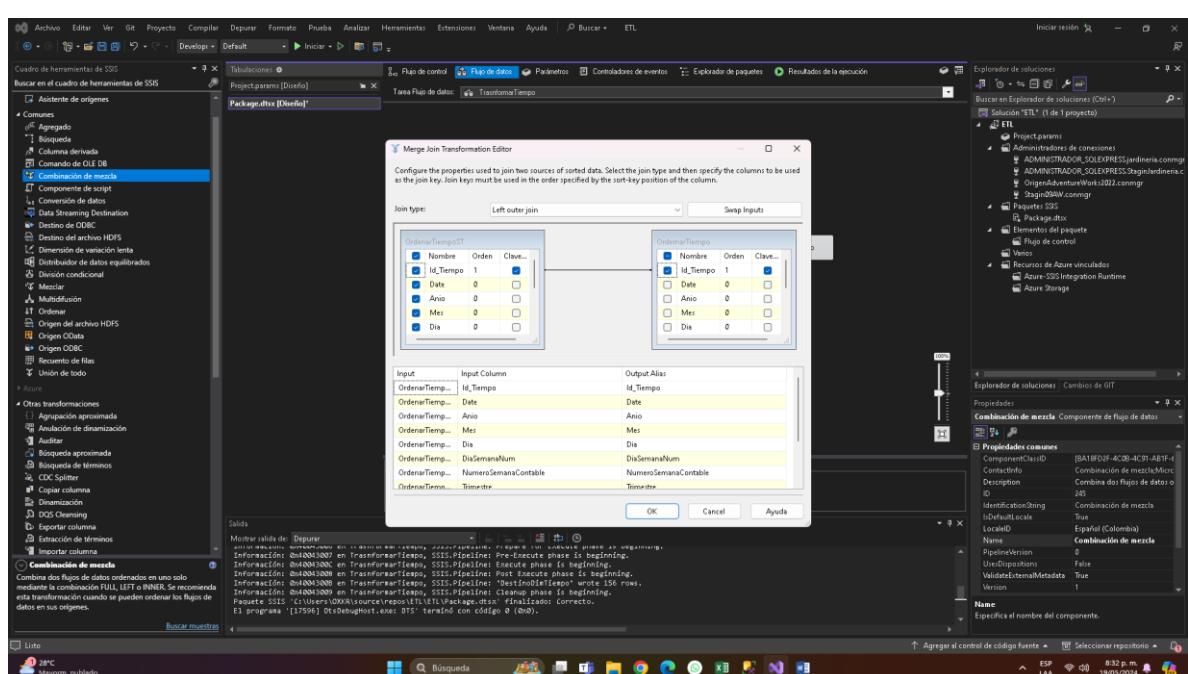
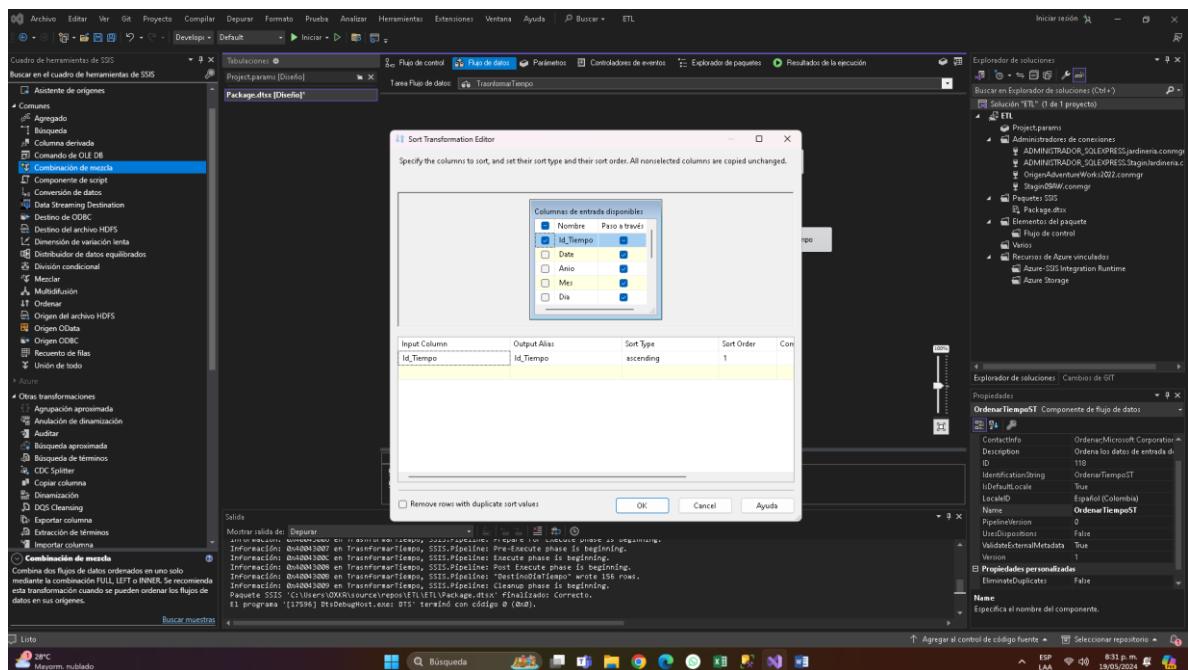
- Toolbars:** Archivo, Editar, Ver, Git, Proyectos, Compilar, Depurar, Formato, Prueba, Analizar, Herramientas, Extensiones, Ventana, Ayuda.
- Left Sidebar (Cuadro de herramientas de SSIS):**
  - Favoritos: Tarea Ejecutar SQL, Tarea Flujos de datos.
  - Comunes: Tarea de expresión, Tarea de generación de perfiles de datos, Tarea de procesamiento de Analysis Services, Tarea Ejecutar paquete, Tarea Ejecutar proceso, Tarea Enviar correo, Tarea FTP, Tarea Inserción masiva, Tarea Pig de Hadoop, Tarea Script, Tarea Submision Web, Tarea Sistema de archivos, Tarea Sistema de archivos de Hadoop, Tarea Subárbol de Hadoop, Tarea XML.
  - Azure: Contenedores, Contenedor de bucles For, Contenedor de bucles Foreach, Contenedor de secuencias.
  - Otras tareas: Elementos del paquete.
- Middle Area:** Diagrama de flujo de datos. Un control "ImprimirTablas" (PrintTables) tiene una flecha que apunta a un control "ExtraccionesJardineria" (ExtractJardineria). El control "ExtraccionesJardineria" tiene una flecha que apunta a un control "TransformarTiempo" (TransformTime).
- Right Sidebar (Explorador de soluciones):**
  - Proyecto: "ETL" (1 de 1 proyecto).
    - Administradores de conexiones: ADMINISTRADOR\_SOLEXPRESS\_jardineria, ADMINISTRADOR\_SOLEXPRESS\_Stagjardineria, OriginAdventureWorks2012.conmgr.
    - Paquetes: Package.dtsx.
    - Elementos del paquete: Flujo de control, Vínculos.
    - Recursos de Azure vinculados: Azure-SSIS Integration Runtime, Azure Storage.
- Bottom Area (Salida):** Muestra el resultado de la ejecución del paquete. Información detallada incluye:
  - (proyecto) ADMINISTRADOR\_SOLEXPRESS\_stagjardineria, SSIS.Pipeline: "destInglolandia\_ST" wrote 23 rows.
  - (proyecto) ADMINISTRADOR\_SOLEXPRESS\_Stagjardineria, SSIS.Pipeline: "DestInglolandia\_ST" wrote 23 rows.
  - (proyecto) ADMINISTRADOR\_SOLEXPRESS\_Stagjardineria, SSIS.Pipeline: "DestInglolandia\_ST" wrote 8 rows.
  - (proyecto) ADMINISTRADOR\_SOLEXPRESS\_Stagjardineria, SSIS.Pipeline: "DestInglolandia\_ST" wrote 8 rows.
  - (proyecto) ADMINISTRADOR\_SOLEXPRESS\_Stagjardineria, SSIS.Pipeline: cleanup phase is beginning.
  - Paquete SSIS: C:\Users\DXW\source\repos\ETL\ETL\Package.dtsx #Finalizado: Correcto.

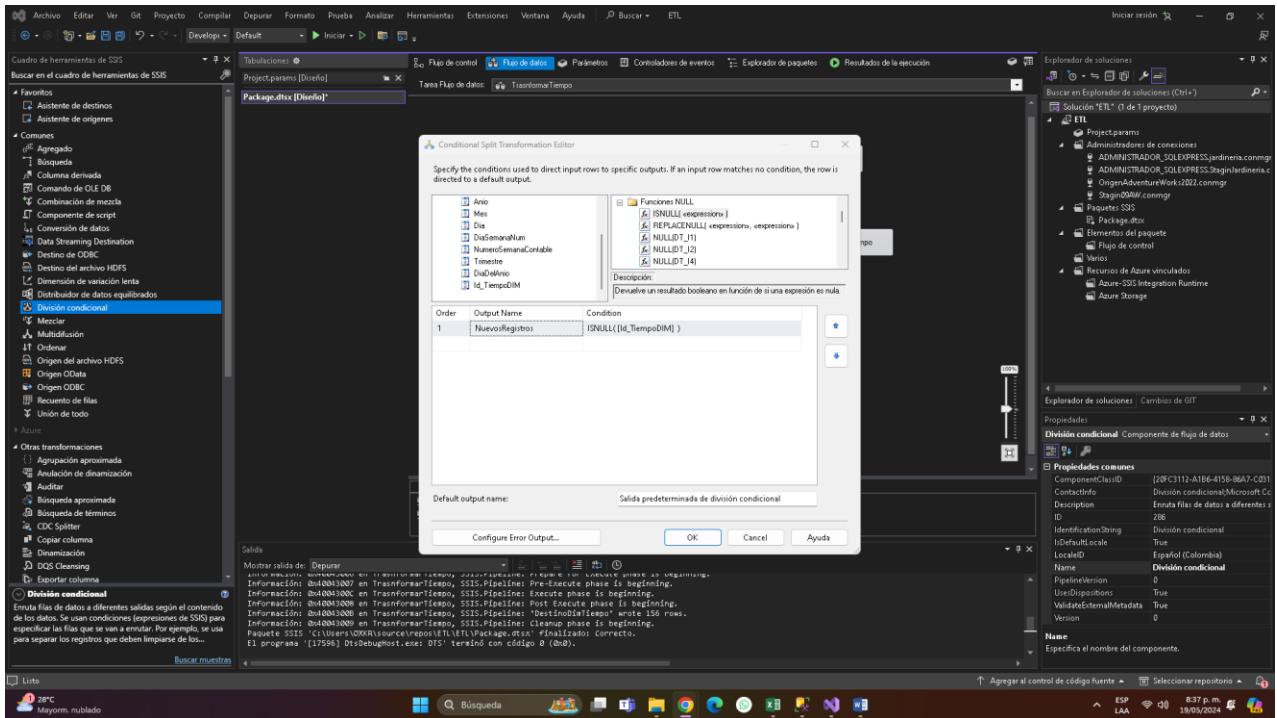
**Bottom Window (Screenshot 2):**

- Toolbars:** Archivo, Editar, Ver, Git, Proyectos, Compilar, Depurar, Formato, Prueba, Analizar, Herramientas, Extensiones, Ventana, Ayuda.
- Left Sidebar (Cuadro de herramientas de SSIS):**
  - Columnas derivadas: Columna derivada.
  - Comunes: Agregado, Búsqueda.
  - Otras transformaciones: Recuento de filas, Unión de todo.
  - Azure: Otros orígenes, Origen ADO.NET, Origen de archivo plano, Origen de archivo sin formato, Origen de Excel, Origen de XML, Otros destinos.
- Middle Area:** Ventana "Derived Column Transformation Editor". Se muestra la configuración para la columna "Dia" (Day):
  - Expresión: <add as new column> [DAY([Date])]
  - Tipo de dato: entero de cuatro bytes ...
- Right Sidebar (Explorador de soluciones):**
  - Proyecto: "ETL" (1 de 1 proyecto).
    - Administradores de conexiones: ADMINISTRADOR\_SOLEXPRESS\_jardineria, ADMINISTRADOR\_SOLEXPRESS\_Stagjardineria, OriginAdventureWorks2012.conmgr.
    - Paquetes: Package.dtsx.
    - Elementos del paquete: Flujo de control, Vínculos.
    - Recursos de Azure vinculados: Azure-SSIS Integration Runtime, Azure Storage.
- Bottom Area (Salida):** Muestra el resultado de la ejecución del paquete. Información detallada incluye:
  - (proyecto) ADMINISTRADOR\_SOLEXPRESS\_stagjardineria, SSIS.Pipeline: "destProductos\_ST" wrote 276 rows.
  - (proyecto) ADMINISTRADOR\_SOLEXPRESS\_Stagjardineria, SSIS.Pipeline: "DestInglolandia\_ST" wrote 23 rows.
  - (proyecto) ADMINISTRADOR\_SOLEXPRESS\_Stagjardineria, SSIS.Pipeline: "DestInglolandia\_ST" wrote 8 rows.
  - (proyecto) ADMINISTRADOR\_SOLEXPRESS\_Stagjardineria, SSIS.Pipeline: "Tiempo\_ST" wrote 78 rows.
  - (proyecto) ADMINISTRADOR\_SOLEXPRESS\_Stagjardineria, SSIS.Pipeline: cleanup phase is beginning.
  - Paquete SSIS: C:\Users\DXW\source\repos\ETL\ETL\Package.dtsx #Finalizado: Correcto.

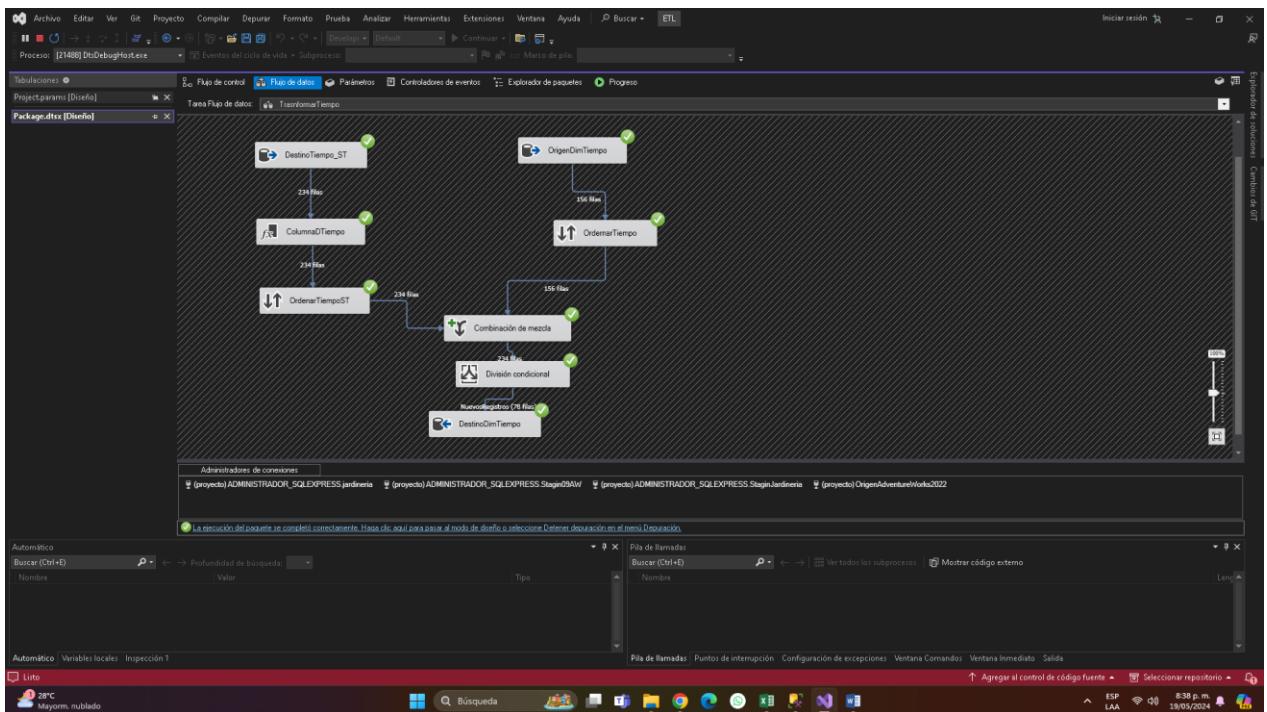




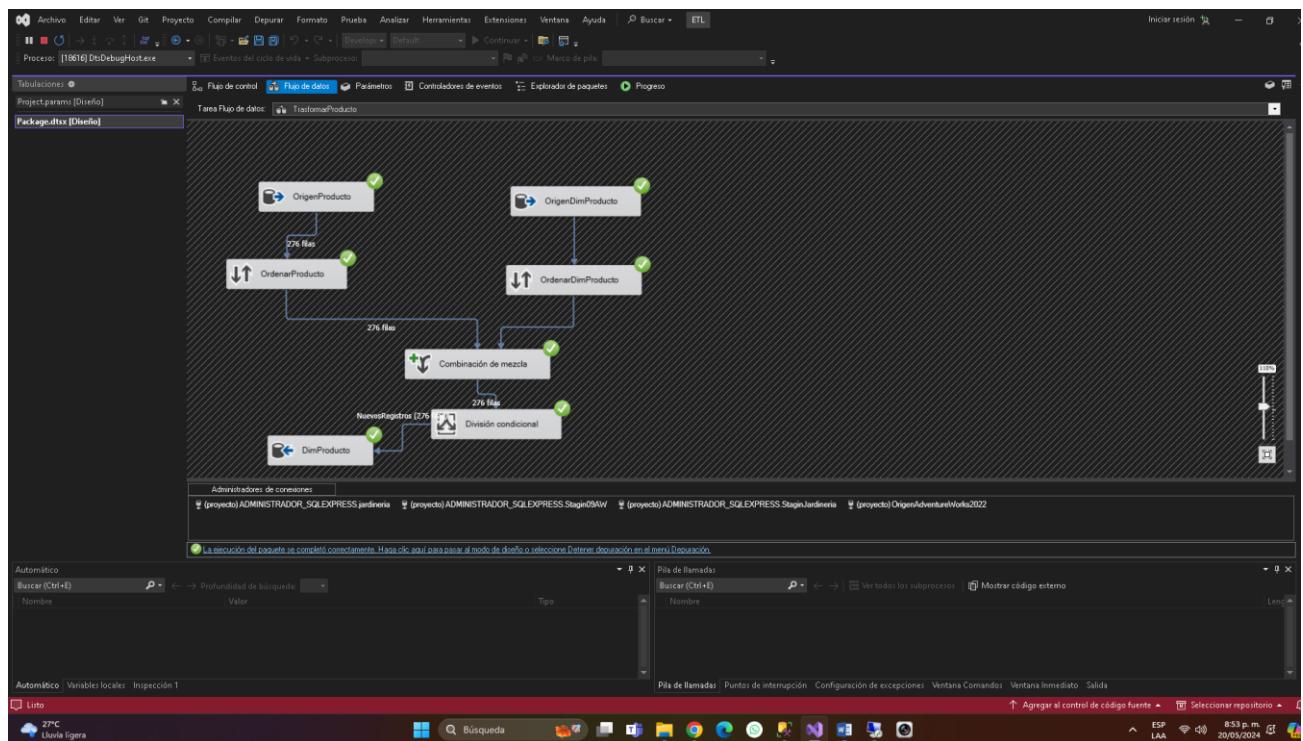




Vamos a llenar la tabla sin necesidad de incrementar solo con los registros nuevos, sin tener que duplicar información de los que ya están



2. Luego seguimos con la Transformación de tabla producto en esta no la vamos hacer como en la anterior transformación donde se utilizó la columna derivada del Visual si no que lo haremos por código sql, lo cual nos hará mas rápido el procesao



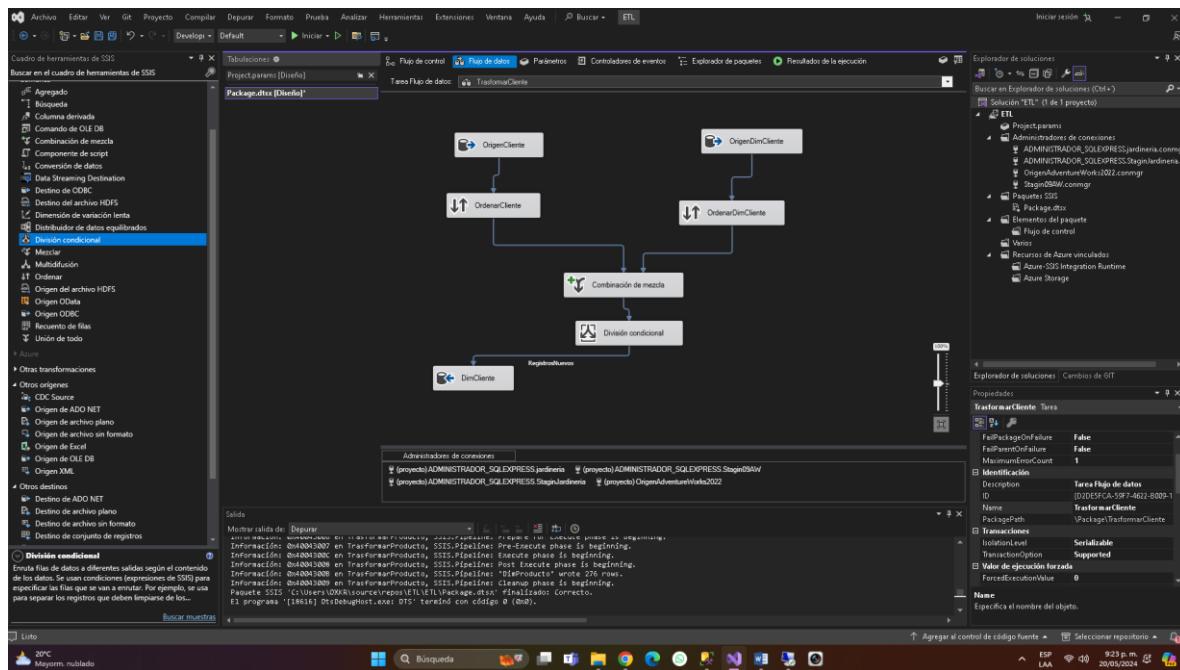
```
--Codigo trasformacion de producto
SELECT [ID_producto],
       [codigoBarras],
       UPPER([nombre]) AS [nombre],
       [Categoria],
       [dimensiones],
       UPPER([proveedor]) AS [proveedor],
       UPPER([descripcion]) AS [descripcion],
       [cantidad_en_stock],
       [precio_venta],
       [precio_proveedor],
       ([precio_venta] - [precio_proveedor]) AS [utilidad]
FROM [dbo].[DestinoProducto_ST];
```

### 3. Trasformación DimClientes:

```

SELECT
    UPPER([ID_cliente]) AS [ID_cliente],
    UPPER([nombre_cliente]) AS [nombre_cliente],
    UPPER(CONCAT([nombre_contacto], ' ', [apellido_contacto])) AS
    [nombre_apellido_contacto],
    UPPER([telefono]) AS [telefono],
    UPPER([Direccion]) AS [direccion],
    UPPER([limite_credito]) AS [limite_credito]
FROM
    [StaginJardineria].[dbo].[DestinoCliente_ST];

```



ID_cliente	nombre_cliente	nombre_apellido_contacto	telefono	direccion	limite_credito
1	GOLDFISH GARDEN	DANIEL G GOLDFISH	55956240	FALSE STREET 52 2 A	3000.00
2	MAGENTA JARDINES DE RIEGOS	JOSÉ TACON	63092340	CALLE 12 345	2000.00
3	LASAKS S.A.	ANTONIO LASAKS	349140246	CELANES 15	1500.00
4	CAMINAS JARDINES S.L.	PEDRO CAMINAS	349140246	CARREÑESES 45	1400.00
5	JARDENA S.A.	JUAN RODRIGUEZ	3491403217	CANJUEVA YORK 74	32100.00
6	JARDIN DE FLORES	JAVIER VILLAR	65489643	C/ ÓÑA 34	4000.00
7	FLORES MARM	MARIA RODRIGUEZ	66559544	CALEGNES24	1500.00
8	FLOWERS, S.A.	BEATRIZ FERNANDEZ	63092359	CALLE SALLERILLO4	3500.00
9	NATURJARDIN	VICTORIO	63242929	CASA SALÓN 15	500.00
10	GOLF S.A.	LUIS MARTINEZ	914940762	DESTANCIJO	2000.00
11	AMERIC GOLF MANAGEMENT SL	MARIO SUAREZ	964493072	CALETARIO	2000.00
12	GARDENING ASSOCIATES	ANNE WRIGHT	559741030	WAVILLE AVENUE	6000.00
13	ALDHA	CHRISTIAN RODRIGEZ	916498952	CARROMA 3	5000.00
14	ELTRUST	FRANCISCO GOMEZ	630923232	AVALO DEL RIBERO	3000.00
15	SOTORGRANDE	MARIA SANTILLANA	630923232	PARQUE DEL PARQUE	6000.00
16	VIVERO HUMADES	FEDERICO GOMEZ	65493760	CAMIGUEL ECHEGARAY 54	7400.00
17	FUENLA CITY	TONY MUÑOZ MENA	675942139	CACILO 52	4500.00
18	JARDINES Y MANZANAS CACTUS SL	MATIAS SAN SÁNCHEZ	916877449	POLOGONO INDUSTRIAL MASPALOMAS, N°52	76000.00
19	JARDINERIAS NATIAS SL	MATIAS SAN MARTIN	916544147	CARACASICO ARCE, NM	105000.00

Aquí revisamos y miramos como queda la base de datos trasformada

#### 4. Vamos con la tabla empleado:

En la cual vamos a modificar también la tabla de las extracciones para traer la descripción de la oficina y no el id

```

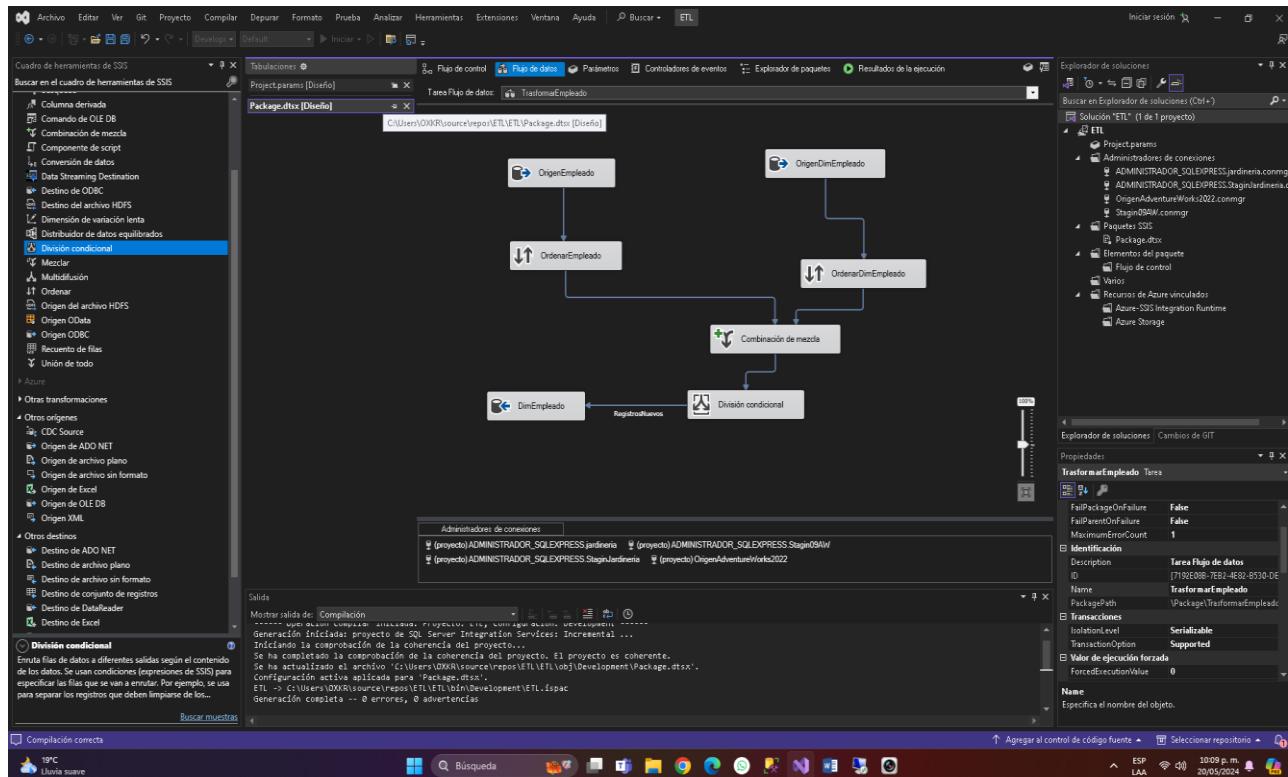
SELECT
    e.id_empleado,
    e.nombre,
    e.apellido1,
    e.apellido2,
    o.descripcion AS descripcion_oficina,
    e.ID_jefe,
    e.puesto
FROM
    empleado e
JOIN
    oficina o ON e.ID_oficina = o.id_oficina;
  
```

al final del trabajo dejaremos el scrip con las modificaciones de las extracciones hechas y el scrip de las trasformaciones

```

SELECT
[id_empleado],
UPPER([nombre]) AS [nombre],
UPPER(CONCAT([apellido1], ' ', [apellido2])) AS [apellidos],
UPPER([descripcion_oficina]) AS [descripcion_oficina],
[ID_jefe],
UPPER([puesto]) AS [puesto]
FROM
[dbo].[DestinoEmpleado_ST];

```



SQLQuery71.sql - ADMINISTRADOR\SQLEXPRESS.StaginJardineria (ADMINISTRADOR\OXR (76) - Microsoft SQL Server Management Studio

```

SELECT TOP (1000) [id_empleado]
      ,[nombre]
      ,[apellido]
      ,[direccion_oficina]
      ,[ID_jefe]
      ,[puesto]
  FROM [StaginJardineria].[dbo].[DimEmpleado]

```

	id_empleado	nombre	apellido	descripcion_oficina	ID_jefe	puesto
1	1	MARCOS	MAGANA FRIEZ	TAL-ES	NULL	DIRECTOR GENERAL
2	2	RUBEN	LÓPEZ MARTINEZ	TAL-ES	1	SUPERDIRECTOR MARKETING
3	3	ALBERTO	SOPHIA CAPRASCO	TAL-ES	2	SUPERDIRECTOR VENTAS
4	4	MARIA	SOLIS JEREZ	TAL-ES	2	SECRETARIA
5	5	FELIPE	ROSAS MARQUEZ	TAL-ES	3	REPRESENTANTE VENTAS
6	6	JUAN CARLOS	ORTIZ SERRANO	TAL-ES	3	REPRESENTANTE VENTAS
7	7	CARLOS	SORIA JIMENEZ	MAD-ES	3	DIRECTOR OFICINA
8	8	MARIANO	LÓPEZ MURICA	MAD-ES	7	REPRESENTANTE VENTAS
9	9	LUIS	CAMPOMOR MARTIN	MAD-ES	7	REPRESENTANTE VENTAS
10	10	HILARIO	RODRIGUEZ HUERTAS	MAD-ES	7	REPRESENTANTE VENTAS
11	11	EMANUEL	MAGSÁN TELLO	BCN-ES	3	DIRECTOR OFICINA
12	12	JOSÉ MANUEL	MARTINEZ DE LA ROSA	BCN-ES	11	REPRESENTANTE VENTAS
13	13	DAVID	PALMA ACETIUNO	BCN-ES	11	REPRESENTANTE VENTAS
14	14	OSCAR	PALMA ACETIUNO	BCN-ES	11	REPRESENTANTE VENTAS
15	15	FRANCOIS	FIGNON	PAR-FR	3	DIRECTOR OFICINA
16	16	LIONEL	NARVAEZ	PAR-FR	15	REPRESENTANTE VENTAS
17	17	LAURENT	SERRA	PAR-FR	15	REPRESENTANTE VENTAS
18	18	MICHAEL	BOLTON	SFC-USA	3	DIRECTOR OFICINA
19	19	WALTER SANTIAGO	SANCHEZ LOPEZ	SFC-USA	18	REPRESENTANTE VENTAS

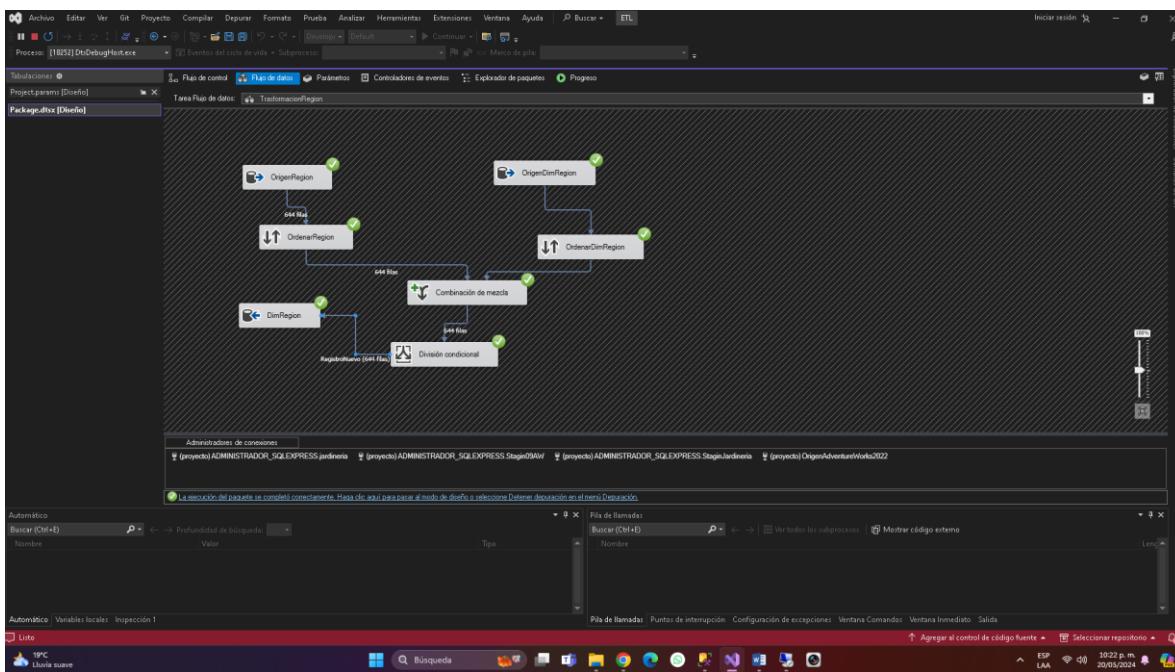
Consulta ejecutada correctamente.

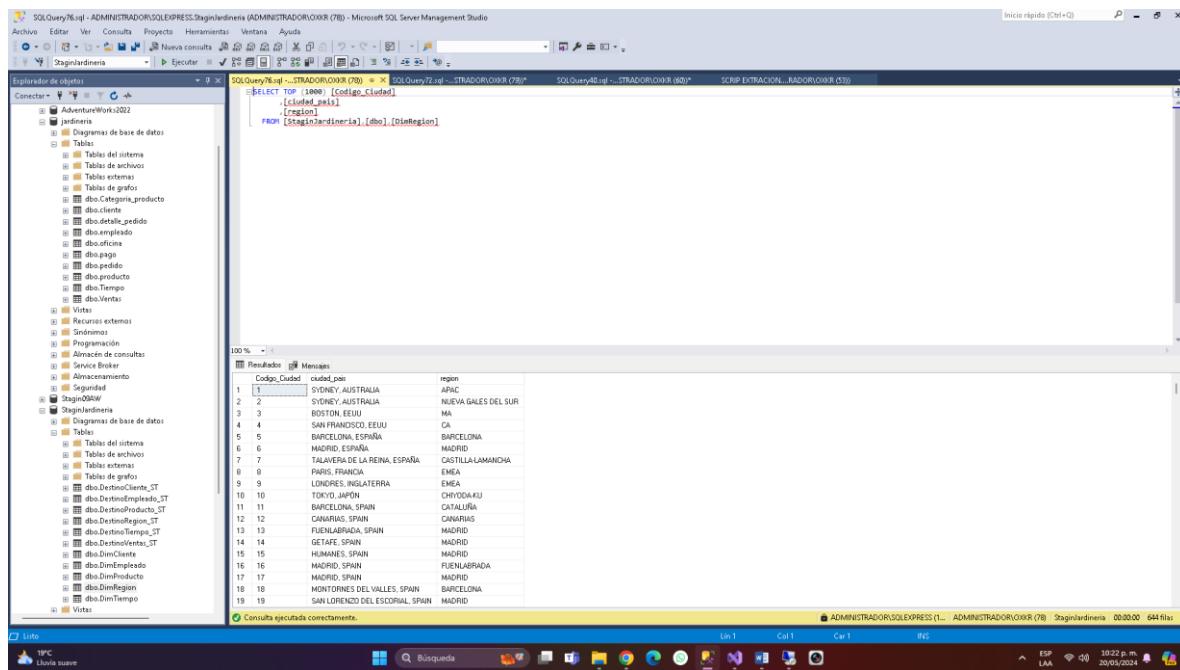
## 5. Ahora Vamos con la Trasformación de Región en la cual solo vamos a concatenar la ciudad con el país

```

SELECT
    [ciudad] AS [Codigo_Ciudad],
    UPPER(CONCAT([ciudad_0], ', ', [pais])) AS [ciudad_pais],
    UPPER([region]) AS [region]
FROM
    [StaginJardineria].[dbo].[DestinoRegion_ST];

```





6. En la tabla de ventas de Hechos, siguiendo las instrucciones del profesor y como era una tabla en la cual estaba vacía, cambié la extracción y utilicé varios join donde, pude cruzar y extraer los datos de los pedidos, los cuales según mi análisis tome como si fueran ventas, y convine con detallas pedido, con clientes, con empleado, con productos y como resultado, obtuvimos una extracción con datos nutridas, ya que esta sería nuestra tabla de hechos, y en la base de datos inicial nos la dieron vacía.

```
SELECT
    dp.[ID_pedido] AS [ID_Venta],
    p.[fecha_entrega] AS [Fecha_Venta],
    c.[ID_cliente] AS [ID_Cliente],
    dp.[ID_producto] AS [ID_Producto],
    dp.[cantidad] AS [Cantidad_Vendida],
    dp.[precio_unidad] AS [Precio_Unitario],
    e.[ID_empleado] AS [ID_Empleado]
FROM
    [jardineria].[dbo].[detalle_pedido] dp
JOIN
    [jardineria].[dbo].[pedido] p ON dp.[ID_pedido] = p.[ID_pedido]
JOIN
    [jardineria].[dbo].[producto] pr ON dp.[ID_producto] = pr.[ID_producto]
JOIN
    [jardineria].[dbo].[cliente] c ON p.[ID_cliente] = c.[ID_cliente]
JOIN
    [jardineria].[dbo].[empleado] e ON p.[ID_cliente] = e.[ID_empleado];
```

Ahora luego vamos a ver como quedó en nuestra base de datos, esto vendría siendo como una corrección de la segunda entrega, pero es un

paso muy necesario, ya que entonces en las trasformaciones no tendría sentido porque estuviéramos trasformando campos vacíos.

Secuencia	ID_Venta	ID_Pedido	Fecha_Venta	ID_Cliente	ID_Producto	Cantidad_Vendida	Precio_Unitario	ID_Empleado	
257	257	NULL	2009-01-11	14	67	10	70.00	14	
258	258	NULL	NULL	1	67	10	70.00	1	
259	259	NULL	NULL	1	67	10	70.00	1	
260	260	NULL	61	2009-01-30	1	67	10	70.00	1
261	261	NULL	62	NULL	1	67	10	70.00	1
262	262	NULL	63	NULL	3	67	10	70.00	3
263	263	NULL	64	NULL	3	67	10	70.00	3
264	264	NULL	65	2009-01-20	3	67	10	70.00	3
265	265	NULL	66	15	67	10	70.00	15	
266	266	NULL	67	2009-01-13	15	67	10	70.00	15
267	267	NULL	68	2009-01-23	15	67	10	70.00	15
268	268	NULL	111	2009-01-06	30	67	10	70.00	30
269	269	NULL	112	2009-01-07	3	67	10	70.00	30
270	270	NULL	113	2009-05-20	30	67	10	70.00	30
271	271	NULL	114	2009-04-10	30	67	10	70.00	30
272	272	NULL	108	2009-01-08	16	67	10	70.00	16
273	273	NULL	109	2009-04-15	16	67	10	70.00	16
274	274	NULL	110	2009-04-24	30	67	10	70.00	30
275	275	NULL	105	NULL	16	67	10	70.00	16

Ahora como todo está en código en la extracción, vamos a realizar una transformación para poder leer los datos bien, y añadiremos unos datos importantes como el total de la venta, y demás y en id\_venta lo dejaremos como id\_pedido, para agilizar el proceso

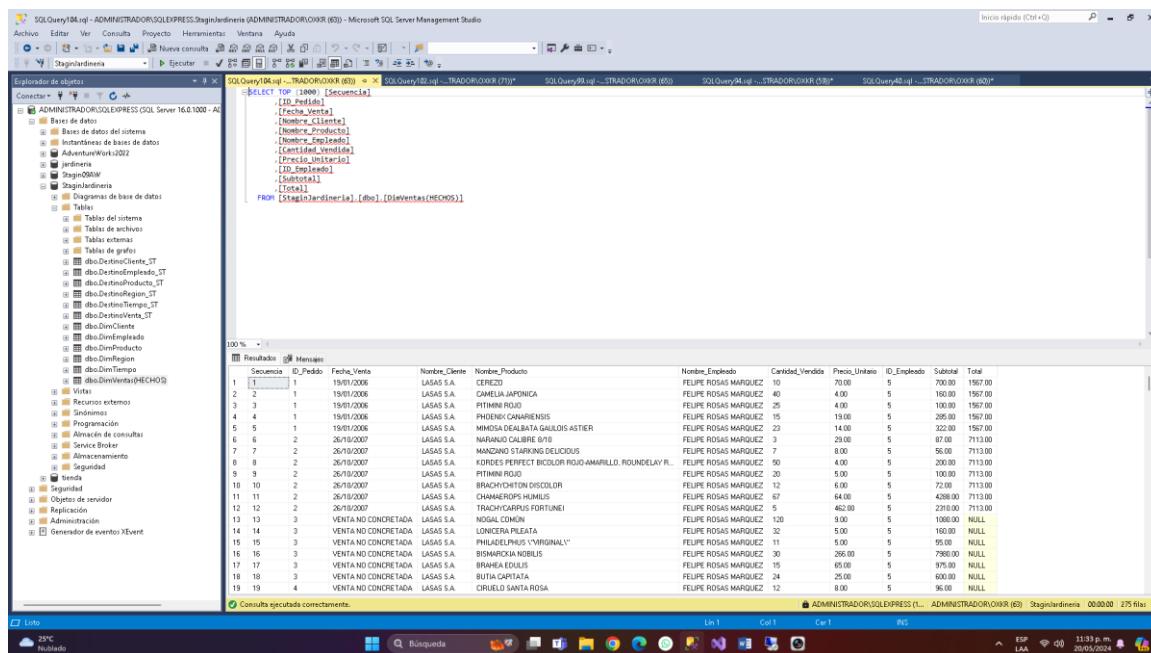
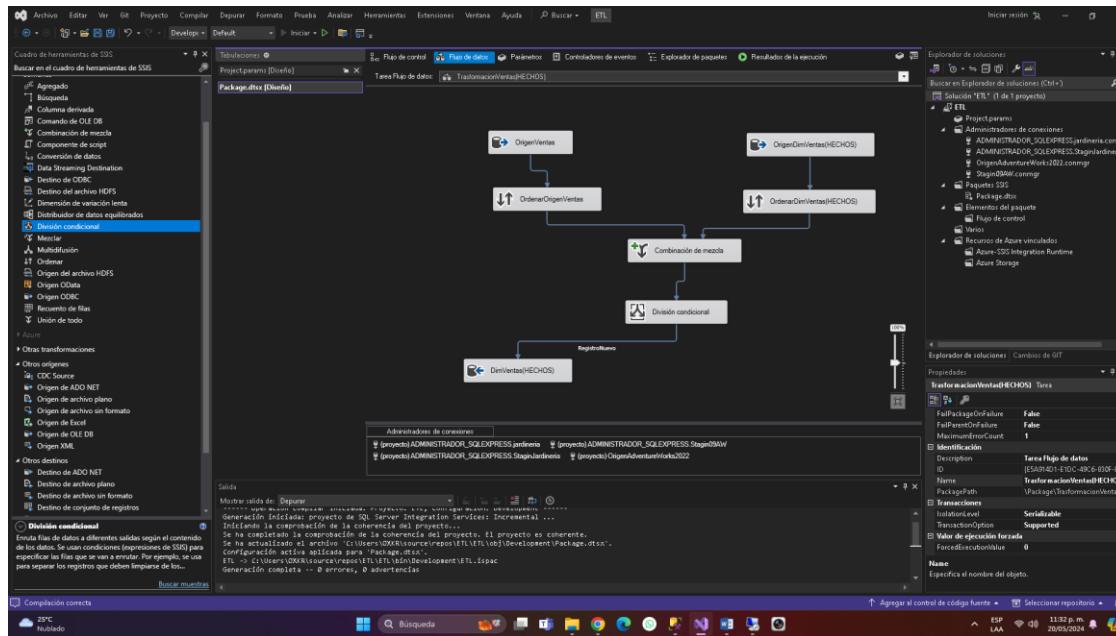
```

SELECT
    [Secuencia],
    [ID_Pedido],
    CASE
        WHEN dv.[Fecha_Venta] IS NULL THEN 'VENTA NO CONCRETADA'
        ELSE CONVERT(varchar, dv.[Fecha_Venta], 103)
    END AS [Fecha_Venta],
    ISNULL(dc.[nombre_cliente], '') AS [Nombre_Cliente],
    ISNULL(dp.[nombre], '') AS [Nombre_Producto],
    ISNULL(CONCAT(de.[nombre], ' ', de.[apellidos]), '') AS [Nombre_Empleado],
    [Cantidad_Vendida],
    [Precio_Unitario],
    dv.[ID_Empleado], -- Especificamos de qué tabla proviene 'ID_Empleado'
    [Cantidad_Vendida] * [Precio_Unitario] AS [Subtotal],
    CASE
        WHEN dv.[Fecha_Venta] IS NULL THEN NULL
        ELSE SUM([Cantidad_Vendida] * [Precio_Unitario]) OVER (PARTITION BY
            [ID_Pedido])
    END AS [Total]
FROM
    [StaginJardineria].[dbo].[DestinoVenta_ST] dv
LEFT JOIN
    [dbo].[DimCliente] dc ON dv.[ID_Cliente] = dc.[ID_cliente]
LEFT JOIN
    [dbo].[DimProducto] dp ON dv.[ID_Producto] = dp.[ID_producto]

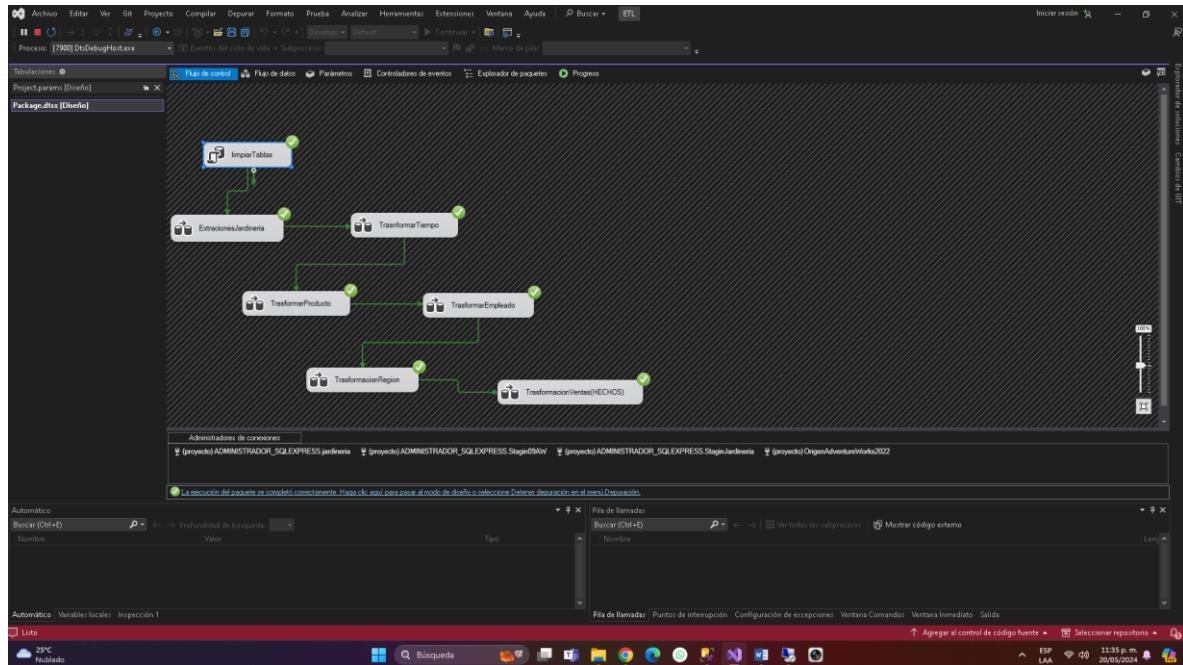
```

## LEFT JOIN

[dbo].[DimEmpleado] de ON dv.[ID\_Emppleado] = de.[id\_empleado];



Corroboramos que todas las trasformaciones y extracciones se ejecutan correctamente



#### CODIGO COMPLETO DE LAS TRASFORMACIONES

```

SELECT [Id_Categoría]
      ,[Desc_Categoría]
      ,[descripción_texto]
      ,[descripción_html]
      ,[imagen]
  FROM [dbo].[Categoria_producto]

GO

--Código transformación de producto
SELECT [ID_producto],
       [codigoBarras],
       UPPER([nombre]) AS [nombre],
       [Categoria],
       [dimensiones],
       UPPER([proveedor]) AS [proveedor],
       UPPER([descripción]) AS [descripción],
       [cantidad_en_stock],
       [precio_venta],
       [precio_proveedor],
       ([precio_venta] - [precio_proveedor]) AS [utilidad]
  FROM [dbo].[DestinoProducto_ST];

--código de transformación de cliente

```

```

        SELECT
        UPPER([ID_cliente]) AS [ID_cliente],
        UPPER([nombre_cliente]) AS [nombre_cliente],
        UPPER(CONCAT([nombre_contacto], ' ', [apellido_contacto])) AS
        [nombre_apellido_contacto],
        UPPER([telefono]) AS [telefono],
        UPPER([Direccion]) AS [direccion],
        UPPER([limite_credito]) AS [limite_credito]
    FROM
        [StaginJardineria].[dbo].[DestinoCliente_ST];

        --codigo trasformacion empleado
        SELECT
        [id_empleado],
        UPPER([nombre]) AS [nombre],
        UPPER(CONCAT([apellido1], ' ', [apellido2])) AS [apellidos],
        UPPER([descripcion_oficina]) AS [descripcion_oficina],
        [ID_jefe],
        UPPER([puesto]) AS [puesto]
    FROM
        [dbo].[DestinoEmpleado_ST];

        --codigo trasformacion de region
        SELECT
        [ciudad]AS[Codigo_Ciudad],
        UPPER(CONCAT([ciudad_0], ' ', [pais])) AS [ciudad_pais],
        UPPER([region]) AS [region]
    FROM
        [StaginJardineria].[dbo].[DestinoRegion_ST];

        --condigo de trasformacion de venta

        SELECT
        [Secuencia],
        [ID_Pedido],
        CASE
            WHEN dv.[Fecha_Venta] IS NULL THEN 'VENTA NO CONCRETADA'
            ELSE CONVERT(varchar, dv.[Fecha_Venta], 103)
        END AS [Fecha_Venta],
        ISNULL(dc.[nombre_cliente], '') AS [Nombre_Cliente],
        ISNULL(dp.[nombre], '') AS [Nombre_Producto],
        ISNULL(CONCAT(de.[nombre], ' ', de.[apellidos]), '') AS [Nombre_Empleado],
        [Cantidad_Vendida],
        [Precio_Unitario],
        dv.[ID_Empleado], -- Especificamos de qué tabla proviene 'ID_Empleado'
        [Cantidad_Vendida] * [Precio_Unitario] AS [Subtotal],
        CASE
            WHEN dv.[Fecha_Venta] IS NULL THEN NULL
            ELSE SUM([Cantidad_Vendida] * [Precio_Unitario]) OVER (PARTITION BY
            [ID_Pedido])
        END AS [Total]
    FROM
        [StaginJardineria].[dbo].[DestinoVenta_ST] dv
    LEFT JOIN
        [dbo].[DimCliente] dc ON dv.[ID_Cliente] = dc.[ID_cliente]
    LEFT JOIN

```

```

[dbo].[DimProducto] dp ON dv.[ID_Producto] = dp.[ID_producto]
LEFT JOIN
[dbo].[DimEmpleado] de ON dv.[ID_Emppleado] = de.[id_empleado];

```

## Actualización de código de Extracciones:

Consultas para la Extracción de Registros desde la Base de Datos Jardinería a la Base de Datos [Staging](#):

```

-- Extracción de datos de la tabla Empleado
SELECT id_empleado,nombre, apellido1, apellido2, ID_oficina, ID_jefe, puesto
FROM empleado;
--actualizamos la estracion de Empleado
SELECT
    e.id_empleado,
    e.nombre,
    e.apellido1,
    e.apellido2,
    o.descripcion AS descripcion_oficina,
    e.ID_jefe,
    e.puesto
FROM
    empleado e
JOIN
    oficina o ON e.ID_oficina = o.id_oficina;

-- Extracción de datos de la tabla Cliente
SELECT id_cliente,nombre_cliente, nombre_contacto, apellido_contacto, telefono,
linea_direccion1 as Direccion, codigo_postal, limite_credito
FROM cliente;

-- Extracción de datos de la tabla Producto
SELECT ID_producto, nombre, Categoria, dimensiones, proveedor, descripcion,
cantidad_en_stock, precio_venta, precio_proveedor
FROM producto;

--actualizamos la extracion de producto donde metimos tambien que añada el nombre de
la categoria y no el id SELECT

SELECT
    p.[ID_producto],
    p.[CodigoProducto] AS[CodigoBarras],
    p.[nombre],
    c.[Desc_Categoría] AS [Categoria],
    p.[dimensiones],
    p.[proveedor],
    p.[descripcion],
    p.[cantidad_en_stock],

```

```
p.[precio_venta],  
p.[precio_proveedor]  
FROM  
[jardineria].[dbo].[producto] p  
JOIN  
[dbo].[Categoria_producto] c  
ON  
p.[Categoria] = c.[Id_Categoría];  
  
--Extraccion de datos tabla tiempo  
select id_tiempo, Anio,Mes,Dia,Dia_Semana  
from tiempo;  
  
--Extraccion de datos tabal ventas  
SELECT  
ID_venta,Fecha_Venta,ID_Cliente,ID_Producto,Cantidad_Vendida,Precio_Unitario, ID_Empl  
eado  
FROM Ventas;
```

## BIBLIOGRAFÍA

- Kimball, R., & Ross, M. (2013). *The Data Warehouse Toolkit: The Definitive Guide to Dimensional Modeling*. John Wiley & Sons.
- Inmon, W. H., & Hackathorn, R. D. (1994). *Using the Data Warehouse*. John Wiley & Sons.
- Ralph Kimball et al., (2008). *The Data Warehouse ETL Toolkit: Practical Techniques for Extracting, Cleaning, Conforming, and Delivering Data*. John Wiley & Sons.
- García-Molina, H., Ullman, J. D., & Widom, J. (2008). *Database Systems: The Complete Book*. Pearson Education.