

# Proyecto Integral de Business Intelligence

De los Datos Crudos al Dashboard: SQL Server, SSIS y Power BI

Documentación Técnica del Proyecto

22 de noviembre de 2025

## Índice

<b>1. Análisis del Dataset de Origen y Definición de Requisitos de Negocio</b>	<b>3</b>
1.1. Selección y Justificación del Dataset . . . . .	3
1.2. Perfilado del Dataset de Origen (Diccionario de Datos) . . . . .	3
1.3. Definición de Requisitos de Negocio y KPIs . . . . .	4
<b>2. Diseño del Modelo Dimensional (Data Warehouse)</b>	<b>5</b>
2.1. Justificación del Esquema en Estrella . . . . .	5
2.2. Diseño de Tablas de Dimensiones (El Contexto) . . . . .	5
2.3. Diseño de Tablas de Hechos (Las Medidas) . . . . .	6
2.4. Diagrama del Esquema Físico . . . . .	6
<b>3. Implementación en SQL Server</b>	<b>8</b>
3.1. Configuración del Entorno . . . . .	8
3.2. Staging de Datos . . . . .	8
3.3. Script DDL (Creación de Tablas) . . . . .	9
<b>4. Proceso ETL con SSIS (Visual Studio)</b>	<b>12</b>
4.1. Configuración del Proyecto . . . . .	12
4.2. Diseño del Flujo de Datos (Data Flow) . . . . .	12
4.3. Transformaciones en Origen (Query SQL) . . . . .	13
4.4. Mapeo y Vista Previa de Datos . . . . .	14
4.5. Detalles Adicionales de Configuración . . . . .	14
<b>5. Dashboard en Power BI</b>	<b>16</b>
5.1. Desarrollo y Limpieza . . . . .	16
5.2. Dashboard Final . . . . .	16
5.3. Validación de Requisitos de Negocio . . . . .	17
5.4. Hoja de Ruta y Futuras Mejoras . . . . .	17

## Índice de figuras

1. Diagrama Entidad-Relación del Modelo Estrella (MarketingDW) . . . . .	7
2. Configuración inicial del servidor . . . . .	8
3. Secuencia completa de importación y validación de datos . . . . .	8

4.	Despliegue del esquema en SQL Server . . . . .	11
5.	Inicio del proyecto y configuración de conexiones OLE DB . . . . .	12
6.	Componentes del Data Flow y ejecución exitosa del paquete . . . . .	12
7.	Consultas SQL de extracción y transformación en SSIS . . . . .	13
8.	Vistas previas de datos y asignaciones de destino . . . . .	14
9.	Configuraciones adicionales y pasos intermedios del proceso ETL . . . . .	15
10.	Pasos de conexión, transformación y limpieza en Power BI . . . . .	16
11.	Análisis de datos y manejo de outliers . . . . .	16
12.	Vista de diseño del Dashboard . . . . .	17

## Índice de cuadros

1.	Diccionario de Datos y Agrupación Dimensional Propuesta . . . . .	3
2.	Mapeo Definitivo de Preguntas de Negocio a KPIs e Insights . . . . .	5
3.	Matriz de Cobertura de Requisitos . . . . .	17

# 1. Análisis del Dataset de Origen y Definición de Requisitos de Negocio

Esta sección fundamental establece las bases de todo el proyecto. Se selecciona la materia prima (los datos) y se define el propósito (los requisitos de negocio).

## 1.1. Selección y Justificación del Dataset

**Dataset Seleccionado:** 'Marketing Campaign Dataset'. **Fuente:** Kaggle (<https://www.kaggle.com/datasets/rodsaldanha/marketing-campaign/data>).

**Justificación de la Selección:** Este conjunto de datos se alinea perfectamente con los requisitos de la solicitud del proyecto:

- **Tabla Plana (CSV Único):** Se proporciona como un único archivo `marketing_campaign.csv`, lo que lo hace ideal para un escenario de ETL desde un origen de archivo plano.
- **Muchos Parámetros (Ancho):** Contiene 28 columnas, proporcionando una gran riqueza de atributos para extraer múltiples dimensiones (demografía del cliente, comportamiento de gasto, respuesta a campañas).
- **Valores Nulos:** La columna `Income` contiene valores nulos, cumpliendo con el requisito explícito de abordar un escenario de limpieza de datos del mundo real.

## 1.2. Perfilado del Dataset de Origen (Diccionario de Datos)

Antes de diseñar el almacén, es imperativo perfilar el archivo de origen. Un análisis de las 28 columnas revela que cada fila no representa una transacción individual. Columnas como `MntWines` (gasto en vinos en los últimos 2 años) y `Complain` (queja en los últimos 2 años) indican que cada fila es un snapshot periódico (posiblemente de 2 años) de un solo cliente.

Este es un punto crucial: no estamos cargando datos 1:1. El proceso ETL deberá desagregar y despivotar esta vista de cliente altamente denormalizada en un modelo transaccional normalizado (esquema en estrella). La siguiente tabla sirve como la 'piedra Rosetta' del proyecto, traduciendo las columnas de origen en bruto a su rol en el futuro Data Warehouse.

Cuadro 1: Diccionario de Datos y Agrupación Dimensional Propuesta

Columna (Origen)	Tipo	Descripción	Agrupación Propuesta
ID	int64	Identificador único	Dim_Cliente (Business Key)
Year_Birth	int64	Año de nacimiento	Dim_Cliente (Atributo)
Education	object	Nivel de educación	Dim_Cliente (Atributo)
Marital_Status	object	Estado civil	Dim_Cliente (Atributo)
Income	object	Ingreso anual (con nulos)	Dim_Cliente (Atributo)
Dt_Customer	object	Fecha de alta	Dim_Fecha (Clave Foránea)
MntWines	int64	Gasto en vinos	Fact_Gasto (Medida)
MntFruits	int64	Gasto en frutas	Fact_Gasto (Medida)
MntMeatProducts	int64	Gasto en carnes	Fact_Gasto (Medida)
NumWebPurchases	int64	Compras vía web	Fact_Compras (Medida)

Columna (Origen)	Tipo	Descripción	Agrupación Propuesta
NumStorePurchases	int64	Compras en tienda	Fact_Compras (Medida)
NumCatalogPurchases	int64	Compras por catálogo	Fact_Compras (Medida)
AcceptedCmp1-5	int64	Aceptación campañas 1-5	Fact_RespuestasCampana
Response	int64	Aceptación última campaña	Fact_RespuestasCampana
Complain	int64	Queja en últimos 2 años	Dim_Cliente (Atributo)
Country	object	País del cliente	Dim_Cliente (Atributo)

### 1.3. Definición de Requisitos de Negocio y KPIs

Tras el análisis del modelo dimensional y la calidad de los datos, se establecieron las siguientes reglas de negocio para garantizar la integridad del análisis en el Dashboard final:

- **Separación de Valor vs. Volumen:** Se detectó que el dataset vincula el gasto monetario (**Monto**) específicamente a las categorías de productos, mientras que vincula el volumen de transacciones (**NumeroDeCompras**) a los canales de venta. Por lo tanto, no es correcto calcular 'Ingresos por Canal', sino 'Tráfico por Canal'.
- **Limitación de Costos:** Dado que el dataset no incluye los costos operativos de las campañas, se descartaron métricas financieras como ROI o Margen para evitar datos ficticios, centrándose en Ingresos Brutos y Comportamiento.

A continuación, se presentan las preguntas clave de negocio (KBQ) definitivas y los KPIs implementados para resolverlas:

Cuadro 2: Mapeo Definitivo de Preguntas de Negocio a KPIs e Insights

Pregunta de Negocio (KBQ)	KPI / Métrica Implementada	Justificación Técnica
¿Cuánto dinero ingresa la compañía y qué productos lo generan?	<b>Ventas Totales</b> (Suma de Monto) y Gasto por Categoría.	Uso de Fact_Gasto vinculada a Dim_CategoríaProducto.
¿Qué canal de venta genera más tráfico de clientes?	<b>Volumen de Transacciones</b> y Tráfico por Canal.	Uso de Fact_Compras vinculada a Dim_CanalCompra.
¿Cuál es el valor promedio que aporta un cliente por interacción?	<b>Valor del Cliente</b> (Ticket Promedio Global).	División: $\frac{\text{Ventas Totales}}{\text{Volumen Transacciones}}$ .
¿Existe estacionalidad en las ventas?	<b>Tendencia Jerárquica</b> (Año/Mes).	Gráfico de líneas usando la jerarquía de Dim_Fecha.
¿Los clientes con mayores ingresos son los que más gastan realmente?	<b>Análisis de Dispersion</b> (Poder Adquisitivo vs. Gasto Real).	Scatter plot correlacionando Income y Monto (filtrando outliers).
¿Cuál es el Retorno de Inversión (ROI) de cada campaña?	<b>ROI</b> (Descartado en esta fase).	Inviabilidad técnica por falta de datos de costes reales en el dataset origen.

## 2. Diseño del Modelo Dimensional (Data Warehouse)

Aquí se traducen los requisitos y el perfilado de datos en un esquema técnico. El objetivo es pasar de una tabla plana denormalizada a un esquema en estrella optimizado para consultas analíticas.

### 2.1. Justificación del Esquema en Estrella

El modelo de origen (tabla plana única) es ineficiente para BI. Las consultas analíticas requerirían escaneos de tabla completa. Un esquema en estrella separa las medidas (números cuantitativos) en Tablas de Hechos y los atributos (contexto descriptivo) en Tablas de Dimensiones. Esto reduce la redundancia, mejora la integridad de los datos y ofrece un rendimiento de consulta superior para herramientas como Power BI.

### 2.2. Diseño de Tablas de Dimensiones (El Contexto)

- **Dim\_Cliente (El 'Quién')**: Almacena atributos demográficos.
  - *PK*: ClienteSK. *BK*: ClienteID\_Origen.
  - *Atributos*: AnioNacimiento, Educacion, EstadoCivil, Ingreso, Pais, FlagQueja.
- **Dim\_Fecha (El 'Cuándo')**: Tabla de fechas estándar para análisis temporal.

- *PK*: FechaSK.
- *Atributos*: FechaCompleta, Anio, Trimestre, Mes, NombreMes, DiaDeSemana.
- **Dim\_CategoríaProducto (El 'Qué' - Gasto)**: Transforma columnas de gasto en filas.
  - *PK*: CategoriaProductoSK.
  - *Valores*: 'Wines', 'Fruits', 'Meat', 'Fish', 'Sweets', 'Gold'.
- **Dim\_CanalCompra (El 'Dónde')**: Transforma columnas de compras en filas.
  - *PK*: CanalCompraSK.
  - *Valores*: 'Web', 'Catalog', 'Store', 'Deals'.
- **Dim\_Campana (El 'Por Qué')**: Almacena detalles de campañas.
  - *PK*: CampanaSK.
  - *Atributos*: NombreCampana, CostoCampana (inicializada en 0 debido a falta de datos para imputación).

### 2.3. Diseño de Tablas de Hechos (Las Medidas)

Como se identificó en el perfilado, una sola fila de cliente en el CSV contiene tres tipos de eventos de negocio distintos. El diseño correcto es crear tres tablas de hechos separadas (Esquema de Constelación).

- **Fact\_Gasto**: Gasto por Cliente por Categoría.
  - *FKs*: ClienteSK, FechaGastoSK, CategoriaProductoSK.
  - *Medida*: Monto.
- **Fact\_Compras**: Compras por Cliente por Canal.
  - *FKs*: ClienteSK, FechaCompraSK, CanalCompraSK.
  - *Medida*: NumeroDeCompras.
- **Fact\_RespuestasCampana**: Respuesta por Cliente por Campaña.
  - *FKs*: ClienteSK, FechaRespuestaSK, CampanaSK.
  - *Medida*: Respuesta (1/0).

### 2.4. Diagrama del Esquema Físico

El diseño resultante es un esquema de 'Constelación' (o 'Galaxy'), donde las tres tablas de hechos comparten dimensiones comunes.

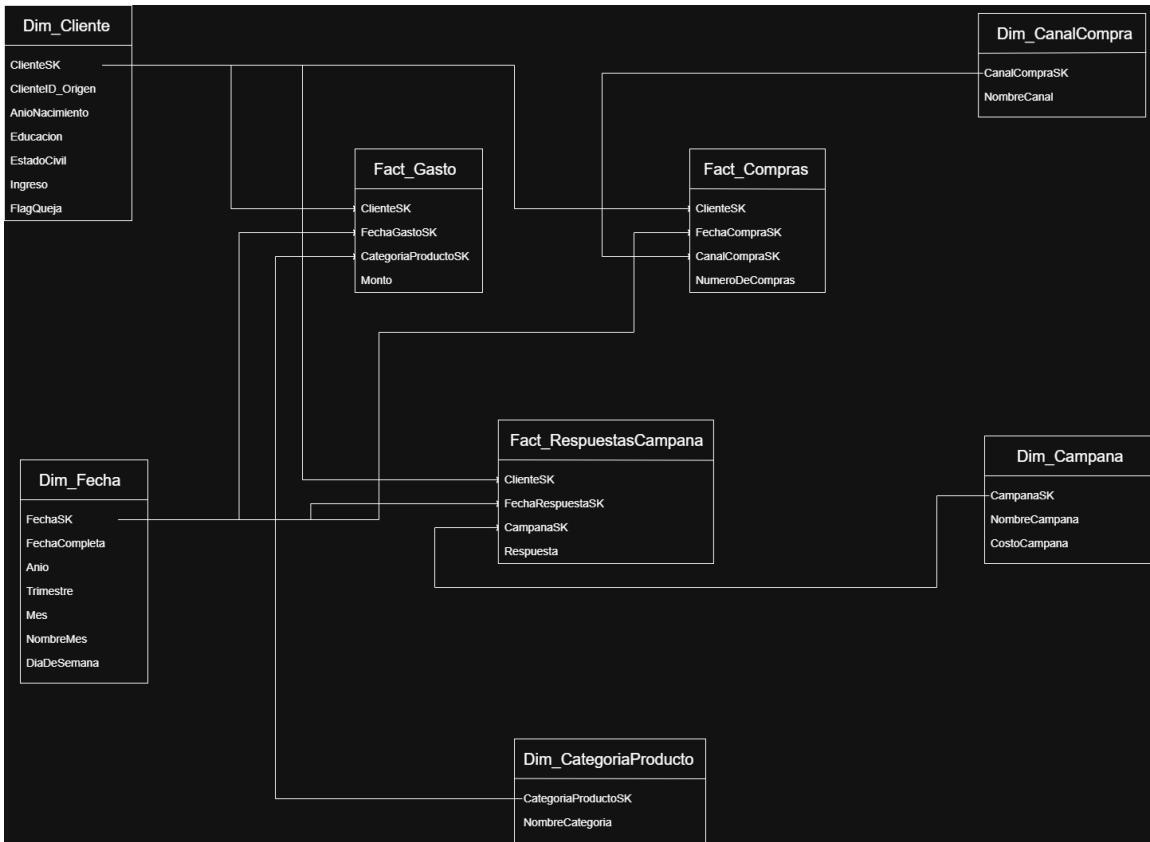


Figura 1: Diagrama Entidad-Relación del Modelo Estrella (MarketingDW)

### 3. Implementación en SQL Server

#### 3.1. Configuración del Entorno

Se configuró SQL Server creando dos bases de datos: **origen** (para la carga inicial del CSV) y **MarketingDW** (el destino final).

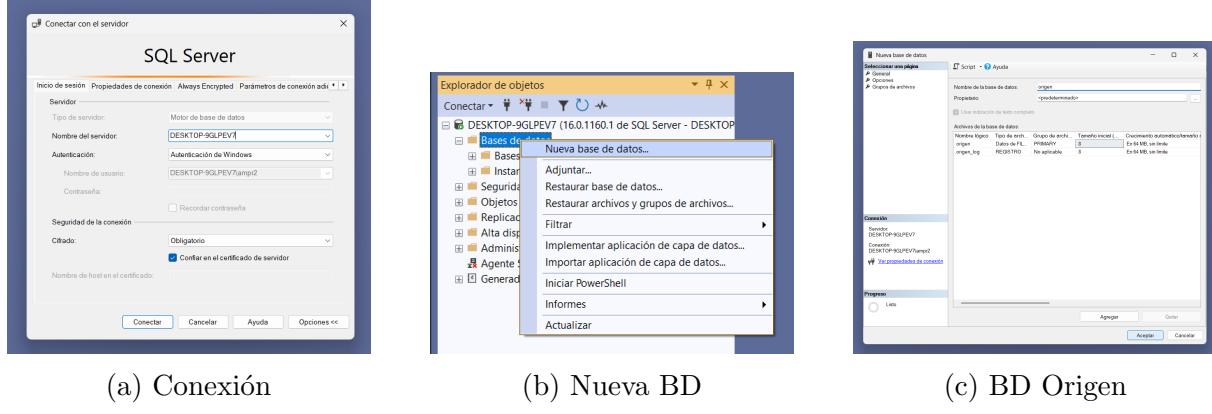


Figura 2: Configuración inicial del servidor

#### 3.2. Staging de Datos

Importación del archivo plano a la base de datos de staging.

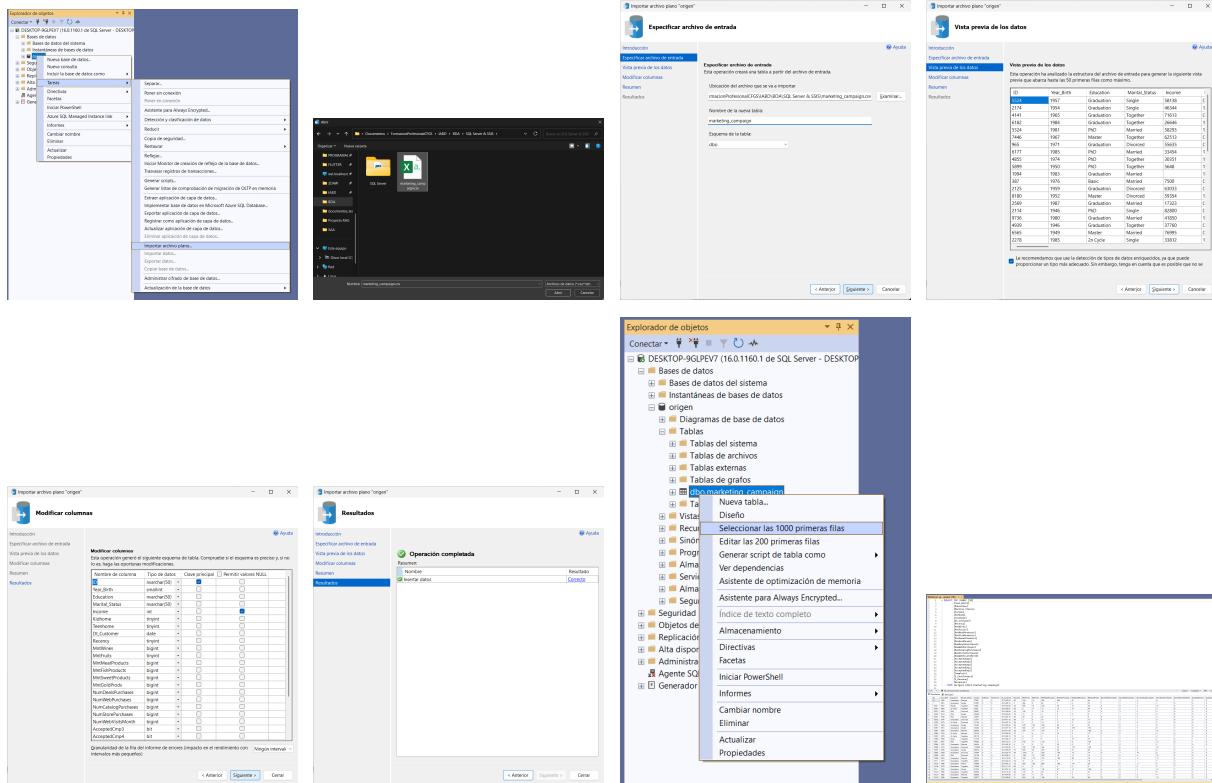


Figura 3: Secuencia completa de importación y validación de datos

### 3.3. Script DDL (Creación de Tablas)

El siguiente código SQL (SQLQuery3.sql) fue ejecutado para crear la estructura dimensional y poblar la dimensión fecha de manera programática.

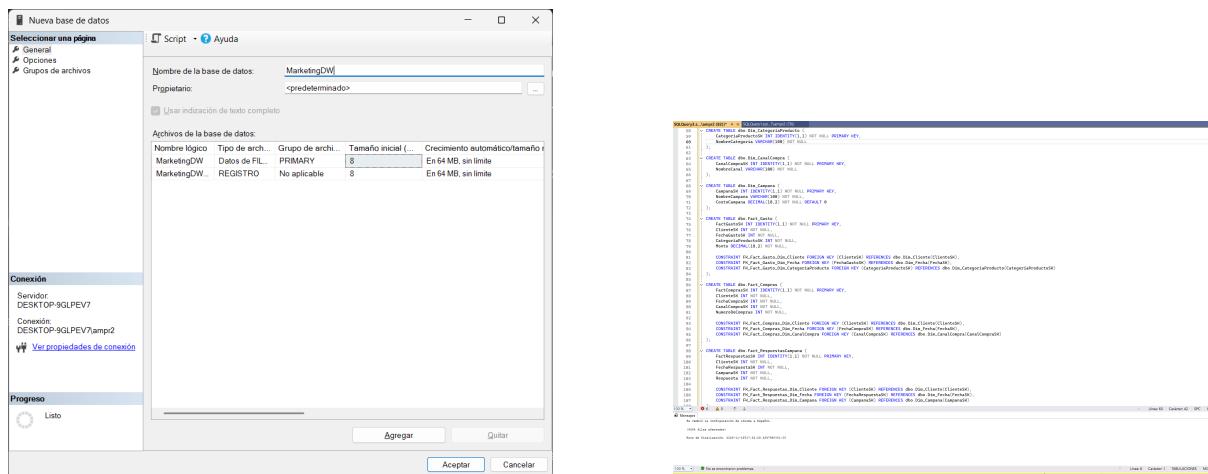
```
1 CREATE TABLE dbo.Dim_Cliente (
2     ClienteSK INT IDENTITY(1,1) NOT NULL PRIMARY KEY,
3     ClienteID_Origen INT NOT NULL ,
4     AnioNacimiento INT ,
5     Educacion VARCHAR(100) ,
6     EstadoCivil VARCHAR(100) ,
7     Ingreso DECIMAL(18,2) ,
8     FlagQueja INT NOT NULL DEFAULT 0 ,
9     FechaAlta DATE
10 );
11 CREATE NONCLUSTERED INDEX IX_Dim_Cliente_BK ON dbo.Dim_Cliente(
12     ClienteID_Origen);
13
14 CREATE TABLE dbo.Dim_Fecha (
15     FechaSK INT NOT NULL PRIMARY KEY ,
16     FechaCompleta DATE NOT NULL ,
17     Anio INT NOT NULL ,
18     Trimestre INT NOT NULL ,
19     Mes INT NOT NULL ,
20     NombreMes VARCHAR(50) NOT NULL ,
21     DiaDeSemana VARCHAR(50) NOT NULL
22 );
23 GO
24
25 DECLARE @StartDate DATE = '20100101';
26 DECLARE @EndDate DATE = '20251231';
27
28 SET LANGUAGE Spanish;
29
30 WITH DateSequence AS (
31     SELECT @StartDate AS Fecha
32     UNION ALL
33     SELECT DATEADD(DAY, 1, Fecha)
34     FROM DateSequence
35     WHERE Fecha < @EndDate
36 )
37 INSERT INTO dbo.Dim_Fecha (
38     FechaSK ,
39     FechaCompleta ,
40     Anio ,
41     Trimestre ,
42     Mes ,
43     NombreMes ,
44     DiaDeSemana
45 )
46 SELECT
47     ROW_NUMBER() OVER(ORDER BY Fecha) AS FechaSK ,
48     Fecha AS FechaCompleta ,
49     YEAR(Fecha) AS Anio ,
50     DATEPART(QUARTER, Fecha) AS Trimestre ,
51     MONTH(Fecha) AS Mes ,
52     DATENAME(MONTH, Fecha) AS NombreMes ,
53     DATENAME(WEEKDAY, Fecha) AS DiaDeSemana
54 FROM DateSequence
```

```

54 OPTION (MAXRECURSION 0);
55 GO
56
57 CREATE TABLE dbo.Dim_CategoríaProducto (
58     CategoríaProductoSK INT IDENTITY(1,1) NOT NULL PRIMARY KEY,
59     NombreCategoría VARCHAR(100) NOT NULL
60 );
61
62 CREATE TABLE dbo.Dim_CanalCompra (
63     CanalCompraSK INT IDENTITY(1,1) NOT NULL PRIMARY KEY,
64     NombreCanal VARCHAR(100) NOT NULL
65 );
66
67 CREATE TABLE dbo.Dim_Campana (
68     CampanaSK INT IDENTITY(1,1) NOT NULL PRIMARY KEY,
69     NombreCampana VARCHAR(100) NOT NULL,
70     CostoCampana DECIMAL(18,2) NOT NULL DEFAULT 0
71 );
72
73 CREATE TABLE dbo.Fact_Gasto (
74     FactGastoSK INT IDENTITY(1,1) NOT NULL PRIMARY KEY,
75     ClienteSK INT NOT NULL,
76     FechaGastoSK INT NOT NULL,
77     CategoríaProductoSK INT NOT NULL,
78     Monto DECIMAL(18,2) NOT NULL,
79
80     CONSTRAINT FK_Fact_Gasto_Dim_Cliente FOREIGN KEY (ClienteSK)
81 REFERENCES dbo.Dim_Cliente(ClienteSK),
82     CONSTRAINT FK_Fact_Gasto_Dim_Fecha FOREIGN KEY (FechaGastoSK)
83 REFERENCES dbo.Dim_Fecha(FechaSK),
84     CONSTRAINT FK_Fact_Gasto_Dim_CategoríaProducto FOREIGN KEY (
85     CategoríaProductoSK) REFERENCES dbo.Dim_CategoríaProducto(
86     CategoríaProductoSK)
87 );
88
89 CREATE TABLE dbo.Fact_Compras (
90     FactComprasSK INT IDENTITY(1,1) NOT NULL PRIMARY KEY,
91     ClienteSK INT NOT NULL,
92     FechaCompraSK INT NOT NULL,
93     CanalCompraSK INT NOT NULL,
94     NúmeroDeCompras INT NOT NULL,
95
96     CONSTRAINT FK_Fact_Compras_Dim_Cliente FOREIGN KEY (ClienteSK)
97 REFERENCES dbo.Dim_Cliente(ClienteSK),
98     CONSTRAINT FK_Fact_Compras_Dim_Fecha FOREIGN KEY (FechaCompraSK)
99 REFERENCES dbo.Dim_Fecha(FechaSK),
100    CONSTRAINT FK_Fact_Compras_Dim_CanalCompra FOREIGN KEY (
101     CanalCompraSK) REFERENCES dbo.Dim_CanalCompra(CanalCompraSK)
102 );
103
104 CREATE TABLE dbo.Fact_RespuestasCampana (
105     FactRespuestasSK INT IDENTITY(1,1) NOT NULL PRIMARY KEY,
106     ClienteSK INT NOT NULL,
107     FechaRespuestaSK INT NOT NULL,
108     CampanaSK INT NOT NULL,
109     Respuesta INT NOT NULL,
110
111 
```

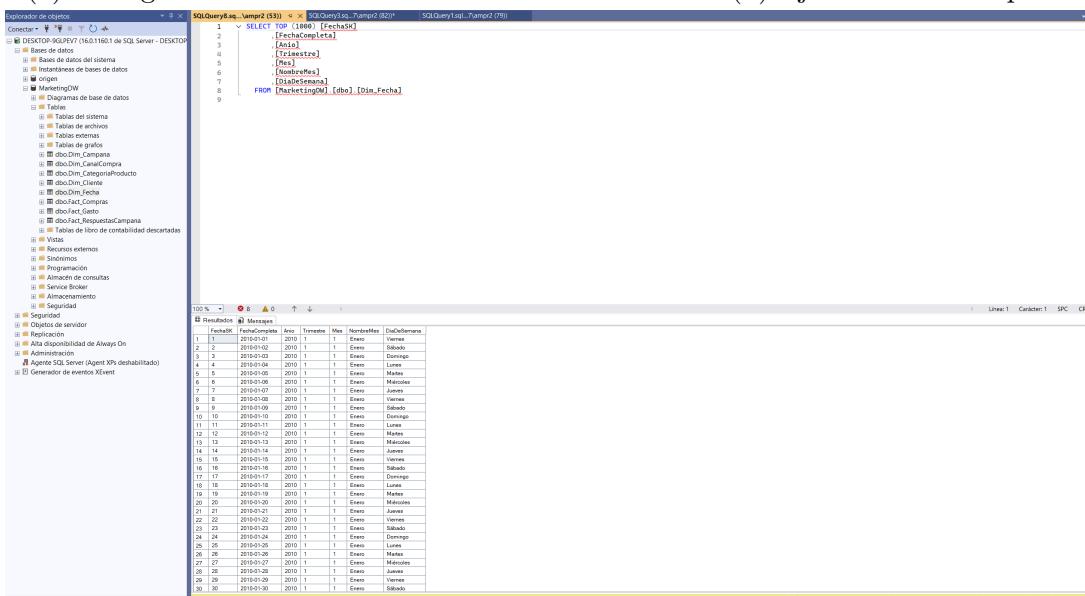
```
104     CONSTRAINT FK_Fact_Respuestas_Dim_Cliente FOREIGN KEY (ClienteSK)
105     REFERENCES dbo.Dim_Cliente(ClienteSK),
106     CONSTRAINT FK_Fact_Respuestas_Dim_Fecha FOREIGN KEY (
107     FechaRespuestaSK) REFERENCES dbo.Dim_Fecha(FechaSK),
108     CONSTRAINT FK_Fact_Respuestas_Dim_Campana FOREIGN KEY (CampanaSK)
109     REFERENCES dbo.Dim_Campana(CampanaSK)
110 );
```

Listing 1: SQLQuery3.sql: Creación del Data Warehouse



(a) Configuración BD Destino

#### (b) Ejecución del Script



(c) Tablas creadas en MarketingDW

Figura 4: Despliegue del esquema en SQL Server

## 4. Proceso ETL con SSIS (Visual Studio)

Se implementó un paquete ETL en Visual Studio para poblar el Data Warehouse.

### 4.1. Configuración del Proyecto

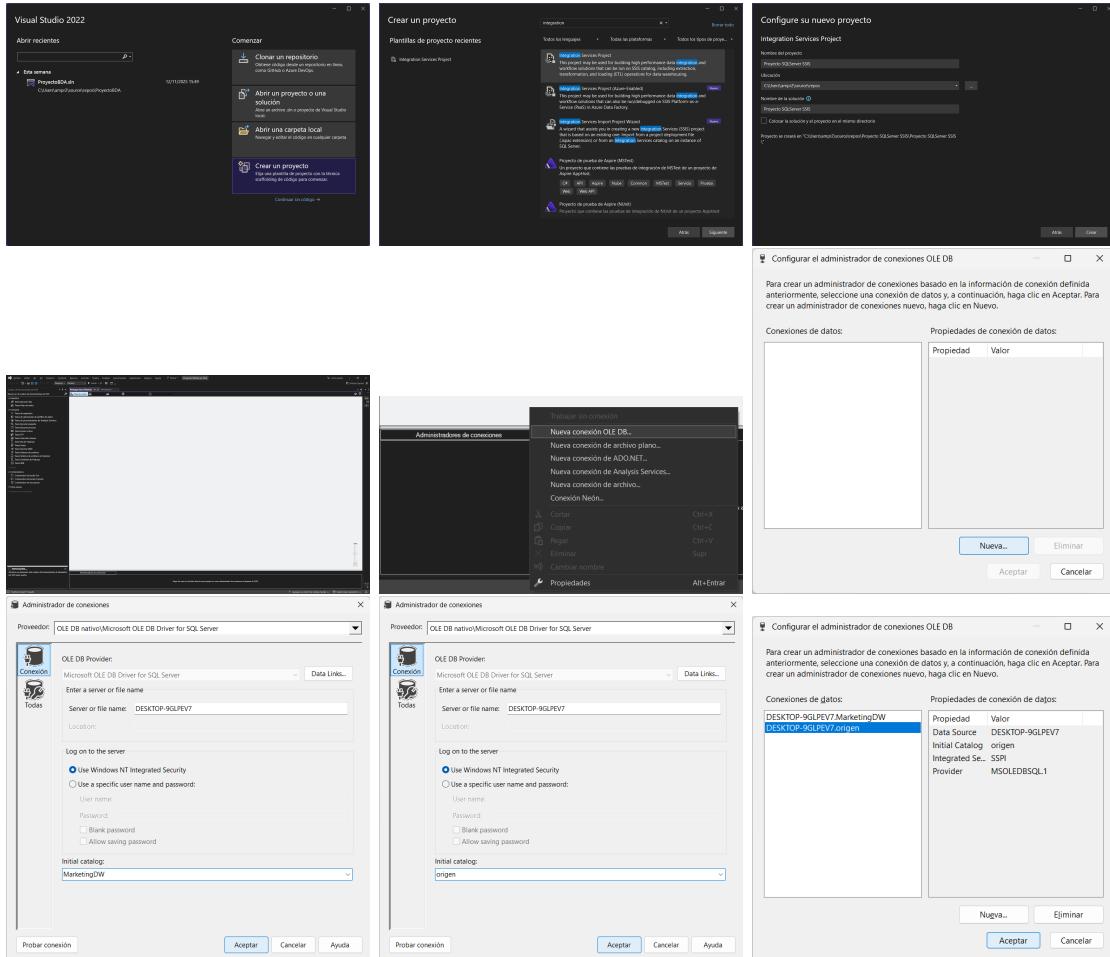


Figura 5: Inicio del proyecto y configuración de conexiones OLE DB

### 4.2. Diseño del Flujo de Datos (Data Flow)

Se crearon tareas para mover datos desde origen a MarketingDW, aplicando transformaciones para normalizar las tablas de hechos.

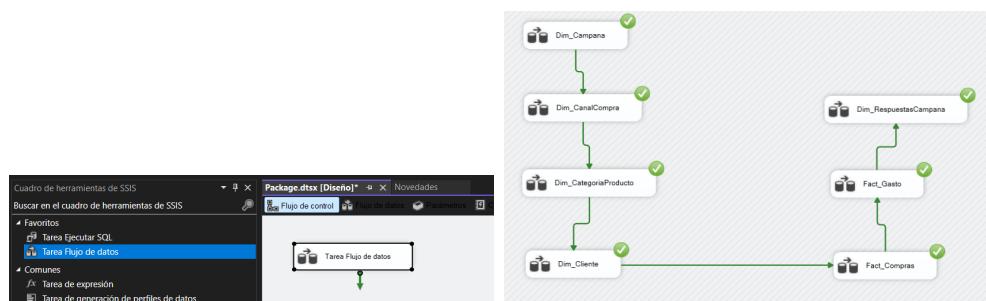


Figura 6: Componentes del Data Flow y ejecución exitosa del paquete

### 4.3. Transformaciones en Origen (Query SQL)

Se utilizaron consultas SQL con **CROSS APPLY** para transformar columnas (como los diferentes productos) en filas (Unpivot).

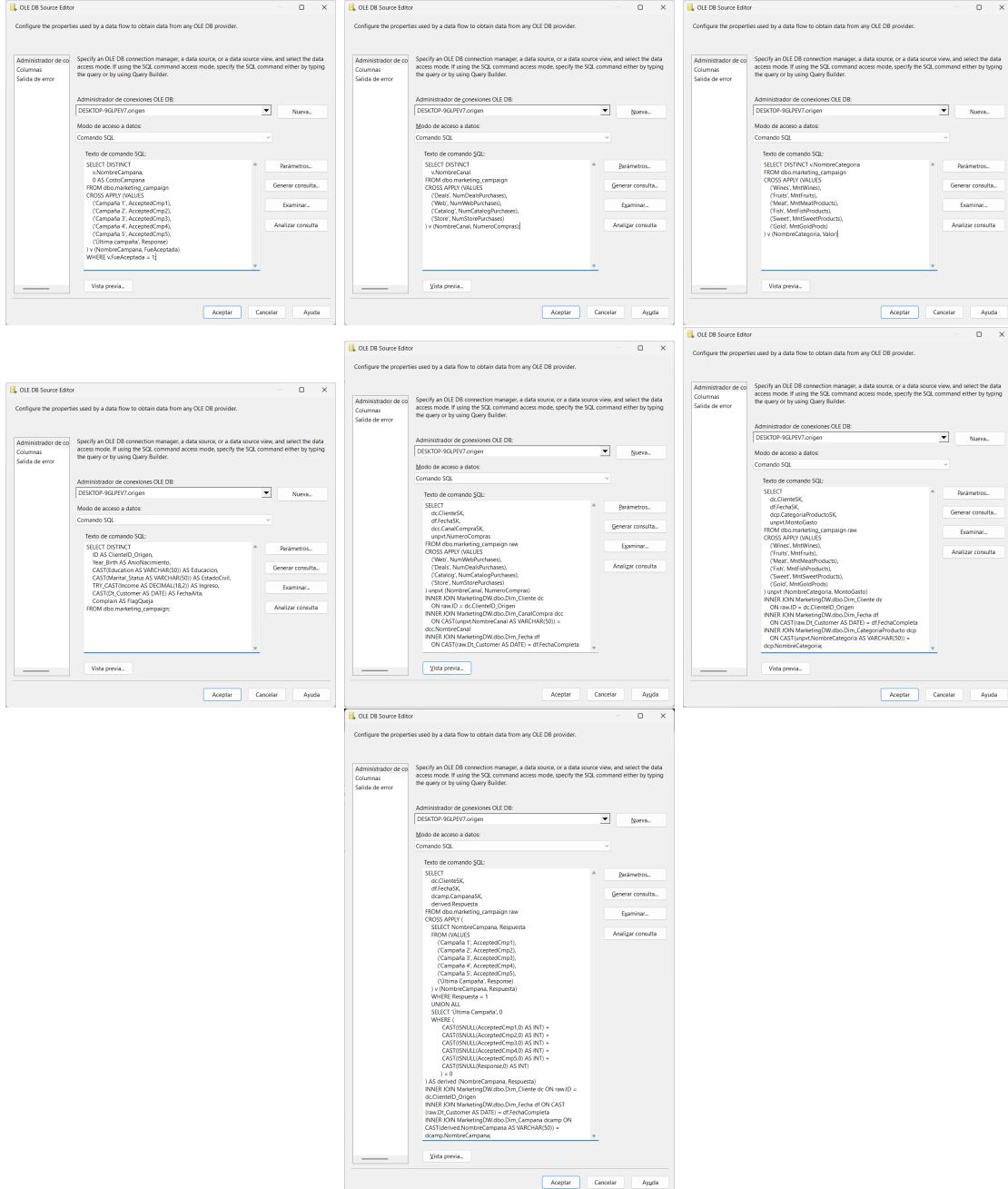


Figura 7: Consultas SQL de extracción y transformación en SSIS

## 4.4. Mapeo y Vista Previa de Datos

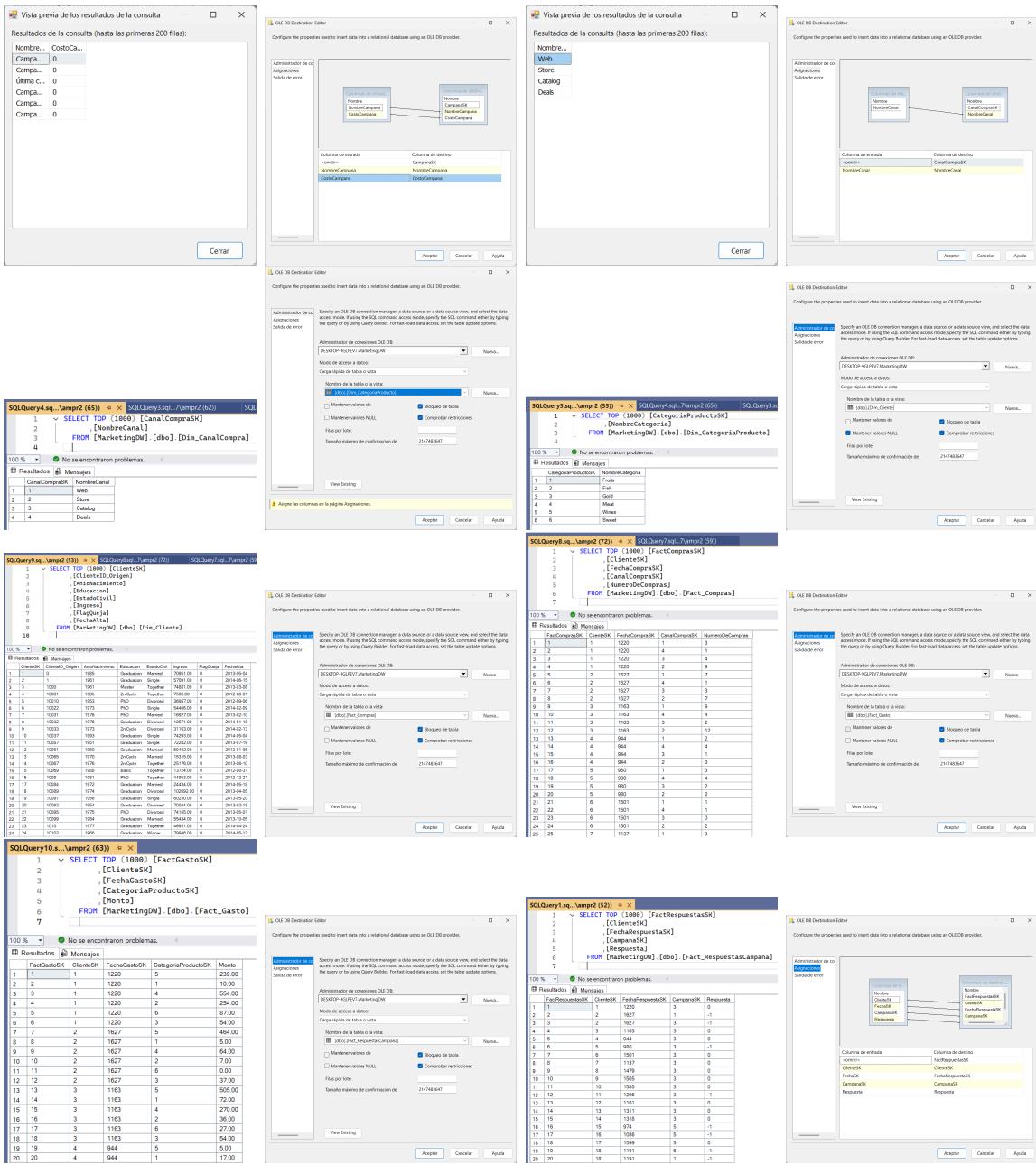


Figura 8: Vistas previas de datos y asignaciones de destino

## 4.5. Detalles Adicionales de Configuración

Se incluyen capturas adicionales del proceso de configuración de componentes, mapeo de columnas y ejecución del paquete.

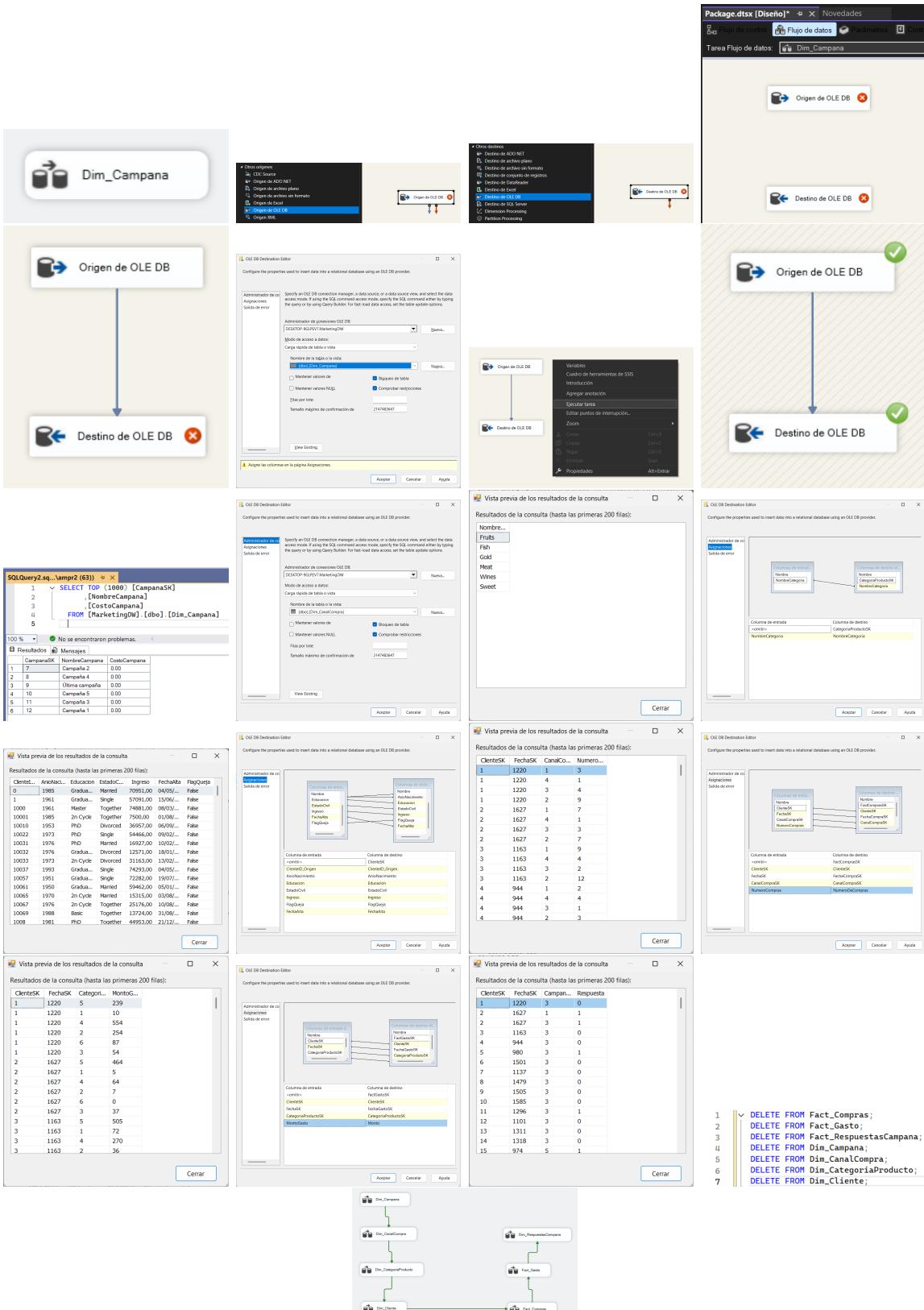


Figura 9: Configuraciones adicionales y pasos intermedios del proceso ETL

## 5. Dashboard en Power BI

### 5.1. Desarrollo y Limpieza

Se realizaron pasos de limpieza adicionales en Power Query y se creó el modelo de datos relacional.

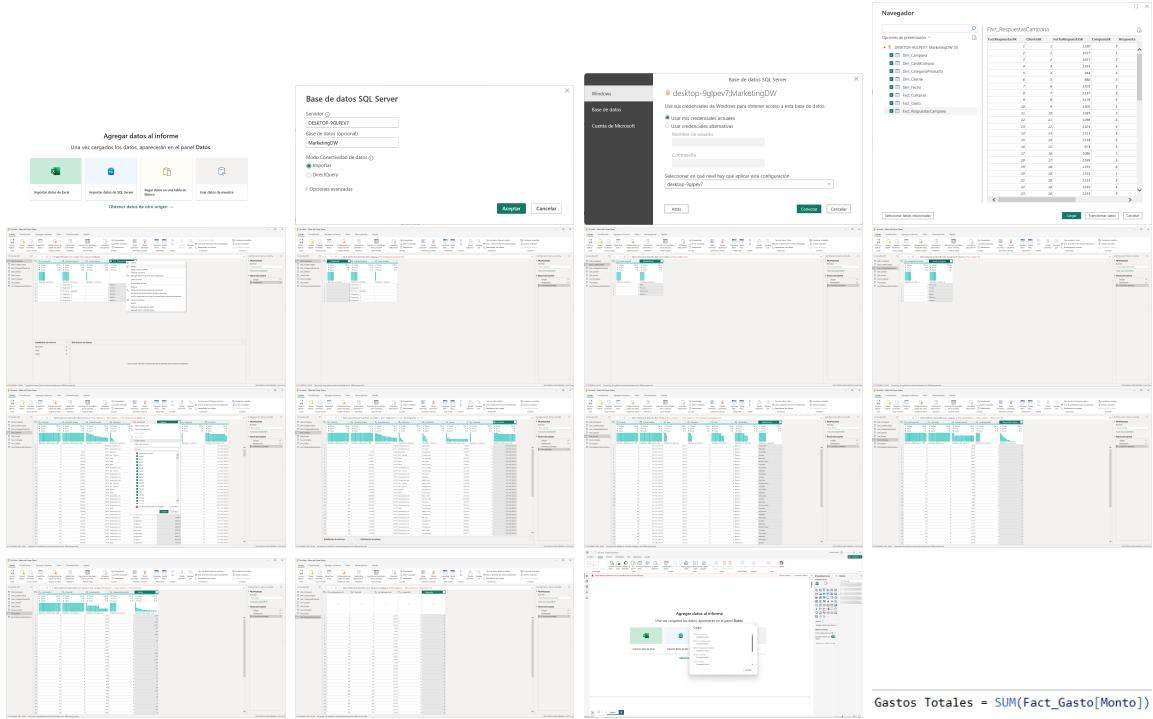


Figura 10: Pasos de conexión, transformación y limpieza en Power BI

### 5.2. Dashboard Final

El resultado es un cuadro de mando ejecutivo que permite el análisis cruzado de ventas por canal, categoría y perfil del cliente.

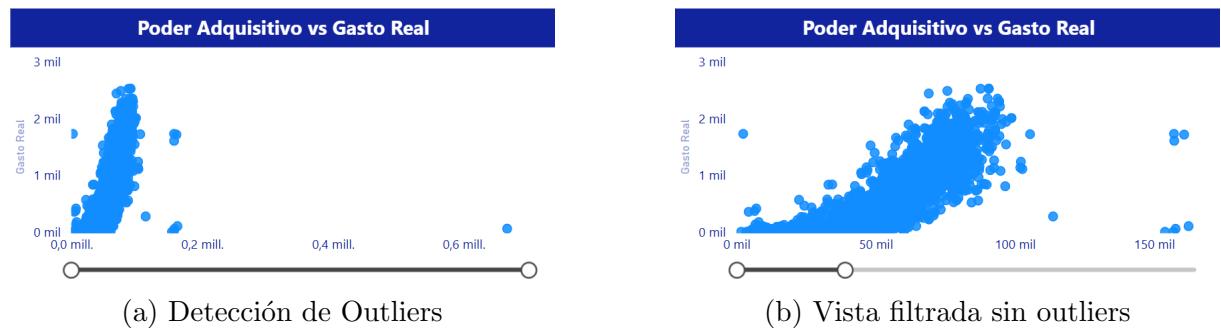


Figura 11: Análisis de datos y manejo de outliers

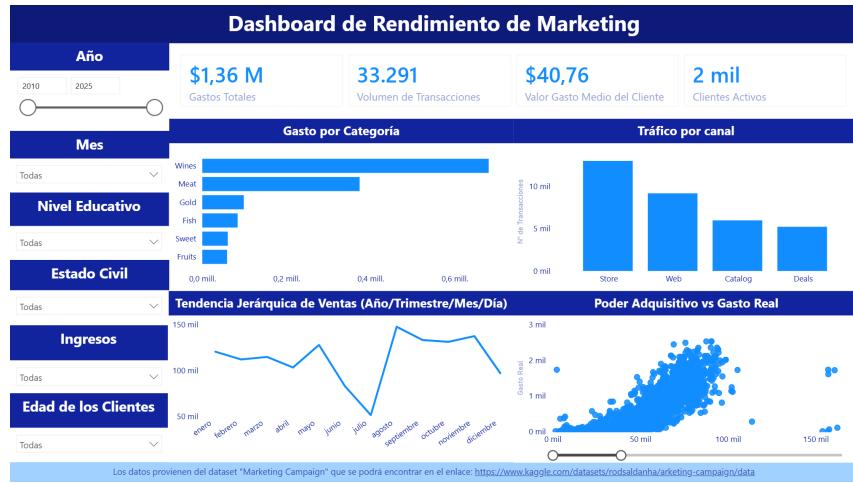


Figura 12: Vista de diseño del Dashboard

### 5.3. Validación de Requisitos de Negocio

Para garantizar la alineación del producto final con los objetivos estratégicos, se ha realizado una trazabilidad entre las Preguntas Clave de Negocio (KBQ) definidas en la fase de análisis y los componentes visuales implementados en el Dashboard.

Cuadro 3: Matriz de Cobertura de Requisitos

Pregunta de Negocio (KBQ)	Estado	Resolución en Dashboard
¿Cuánto dinero ingresa la compañía y qué productos lo generan?	Resuelto	KPI de <i>Ventas Totales</i> (\$1.36M) y Gráfico de barras <i>Gasto por Categoría</i> .
¿Qué canal de venta genera más tráfico de clientes?	Resuelto	KPI de <i>Volumen de Transacciones</i> (33K) y Gráfico de columnas <i>Tráfico por Canal</i> .
¿Cuál es el valor promedio que aporta un cliente?	Resuelto	KPI de <i>Valor Gasto Medio del Cliente</i> (\$40.76).
¿Existe estacionalidad en las ventas?	Resuelto	Gráfico de líneas <i>Tendencia Jerárquica</i> , permitiendo drill-down a nivel mensual para detectar picos.
¿Los clientes con mayores ingresos son los que más gastan?	Resuelto	Gráfico de dispersión <i>Poder Adquisitivo vs Gasto Real</i> con slider para filtrar outliers.
¿Cuál es el Retorno de Inversión (ROI) de cada campaña?	Pendiente	No implementado debido a la falta de datos de costes en el origen.

### 5.4. Hoja de Ruta y Futuras Mejoras

Basado en la auditoría anterior, se identifican áreas clave para la evolución del sistema en una segunda iteración:

1. **Incorporación de Datos de Costos (Prioridad Alta):** La pregunta de negocio sobre el ROI no ha podido ser resuelta en esta fase. Se requiere la ingestión de un nuevo fichero origen que contenga el presupuesto gastado por campaña (`Dim_Campana.CostoCampana`). Esto habilitará automáticamente el cálculo de métricas de rentabilidad sin modificar la estructura del Data Warehouse.
2. **Análisis Geográfico:** Aprovechar la columna `Country` disponible en el dataset (actualmente cargada pero no explotada) para añadir un mapa interactivo que permita filtrar el rendimiento de marketing por región geográfica.
3. **Modelo Predictivo:** Integrar un script de Python o R en Power BI para clusterizar a los clientes basándose en su comportamiento de compra (RFM Analysis), yendo más allá del análisis descriptivo actual hacia uno prescriptivo.