

**EVIDENCIA DE APRENDIZAJE 3:**  
**PROCESO DE TRANSFORMACIÓN DE DATOS Y CARGA EN EL DATA MART FINAL**

Felipe Fernández Rodríguez

**Docente:**

Antonio Jesús Valderrama Jaramillo

**Curso:**

Bases de Datos II

Ingeniería en Software y Datos

Institución Universitaria Digital de Antioquia

2025

## INTRODUCCIÓN

Este documento presenta la evidencia del proceso de transformación y carga de datos desarrollado a partir de la base de datos operacional Jardinería, pasando por una etapa de staging hasta la construcción del data mart final. El informe describe de forma ordenada las etapas del flujo ETL implementado, la verificación de la calidad y consistencia de los datos en la zona de staging, y la estructuración del data mart bajo un modelo dimensional en estrella.

A lo largo del proyecto se incluye: (i) la revisión del diseño del modelo estrella y las relaciones entre dimensiones y la tabla de hechos; (ii) las actividades de extracción de los datos desde la base origen hacia el staging; (iii) las transformaciones aplicadas para limpiar, normalizar y enriquecer la información; (iv) la carga final de los datos limpios en el data mart independiente; y (v) las comprobaciones y consultas de validación que garantizan la fiabilidad del análisis.

El data mart resultante está pensado para soportar análisis de negocio relevantes —por ejemplo, identificar los productos y categorías con mayor desempeño o analizar ventas por año— sin necesidad de acceder a las bases de datos operacionales. En secciones posteriores se detallarán las consultas SQL, las reglas de transformación aplicadas y la evidencia del correcto volcado de datos al data mart.

## OBJETIVOS

**Objetivo General:** Desarrollar un proceso de transformación y carga de datos desde la base de datos origen hasta un data mart en modelo estrella, garantizando información limpia, consistente y preparada para el análisis empresarial.

### Objetivos específicos:

- Revisar el modelo estrella definido para la comprensión de la estructura y las relaciones entre las tablas de dimensiones y de hechos.
- Verificar la disponibilidad y consistencia de los datos extraídos hacia la base de datos de staging.
- Aplicar técnicas de limpieza, normalización y enriquecimiento a los datos en staging para asegurar su calidad.
- Diseñar consultas SQL que permitan la carga de datos transformados en el data mart final.
- Comprobar la correcta inserción y coherencia de los registros en el data mart para la habilitación del análisis de negocio.

## PLANTEAMIENTO DEL PROBLEMA

Las organizaciones modernas generan grandes volúmenes de información en sus sistemas transaccionales, los cuales están diseñados principalmente para registrar operaciones diarias y asegurar la integridad de los datos. Sin embargo, estos sistemas no siempre están estructurados para responder de manera eficiente a preguntas estratégicas de negocio, como la identificación de los productos con mayor demanda, las categorías más relevantes o los periodos de tiempo con mayores ventas.

En el caso de la base de datos Jardinería, la información se encuentra distribuida en múltiples tablas relacionadas con clientes, productos, pedidos y pagos. Aunque este modelo resulta adecuado para el registro de las operaciones, dificulta el análisis directo, ya que los datos pueden contener inconsistencias, duplicidades o valores faltantes que limitan su uso confiable en procesos de toma de decisiones.

Asimismo, la falta de un modelo analítico estructurado impide contar con una fuente única y organizada de información que facilite la generación de reportes y la aplicación de herramientas de inteligencia de negocios. Sin un proceso de extracción, transformación y carga (ETL) que garantice la limpieza y estandarización de los datos, los análisis pueden arrojar resultados erróneos o incompletos.

Por ello, se hace necesario diseñar e implementar un data mart en modelo estrella, alimentado a través de una etapa previa de staging que permita depurar y transformar la información. Con este esquema, la empresa puede disponer de un entorno optimizado para el análisis de sus datos, asegurando que las decisiones estratégicas se fundamenten en información precisa, consistente y alineada con las necesidades del negocio.

## ANÁLISIS DEL PROBLEMA

El análisis de la base de datos Jardinería evidencia una serie de dificultades que afectan la confiabilidad y la utilidad de la información para fines analíticos y de apoyo a la toma de decisiones, ya que, está diseñada bajo un modelo relacional enfocado en la operación diaria. Aunque garantiza la integridad de los registros, no facilita consultas analíticas complejas ni consolidaciones de información.

Se identificaron valores nulos, precios inválidos, productos sin descripción, dimensiones mal formateadas y pedidos en estados no aptos para análisis de ventas. Estos problemas comprometen la calidad de los resultados obtenidos al consultar directamente la base operacional. Actualmente no existe un flujo sistemático que realice limpieza, normalización o enriquecimiento de los datos antes de utilizarlos en reportes de gestión. Esto genera riesgo de duplicidades y de errores en los cálculos al no contar con un data mart estructurado bajo un modelo dimensional, los reportes se ven limitados. No se dispone de una tabla de hechos que consolide las métricas clave ni de dimensiones que permitan analizar la información desde distintas perspectivas (producto, categoría, tiempo).

Estas limitaciones dificultan responder preguntas estratégicas como:

- ¿Cuál es el producto más vendido?
- ¿Qué categoría concentra mayor número de productos?
- ¿En qué año se registró el mayor volumen de ventas?

En conclusión, el problema radica en que la base de datos Jardinería, en su estado original, no está preparada para el análisis empresarial. Se requiere un proceso formal de extracción, transformación y carga (ETL) que permita crear un staging como espacio de depuración y un data mart en modelo estrella como entorno final de análisis confiable.

## PROPUESTA DE SOLUCIÓN

La solución propuesta para responder a las necesidades de análisis de la base de datos Jardinería se estructuró en tres etapas principales: el diseño de un modelo estrella inicial, la implementación de una base de datos de staging para cargar y transformar los datos, y la construcción de un data mart final en modelo estrella que permitió dar respuesta a las preguntas de negocio definidas.

Como punto de partida, se diseñó un modelo dimensional en estrella a partir de las tablas principales de la base de datos transaccional Jardinería. En este esquema, la tabla central de hechos, denominada FactVentas, concentra las métricas clave como cantidad, precio unitario y total de la venta, mientras que alrededor se disponen las tablas de dimensiones que aportan el contexto necesario para el análisis.

Las dimensiones definidas fueron:

### **1. Tabla de hechos: FactVentas:**

Es el núcleo del modelo y almacena los eventos de ventas realizados por la empresa. Contiene tanto métricas cuantitativas como las claves foráneas que enlazan con las dimensiones.

Claves foráneas:

- ID\_producto: identifica el producto vendido.
- ID\_cliente: identifica el cliente asociado a la venta.
- ID\_tiempo: indica el momento temporal de la transacción.
- ID\_empleado: relaciona al empleado que gestionó el pedido.
- ID\_categoria: permite el análisis agregado a nivel de categoría de producto.

Métricas de negocio:

- cantidad: número de unidades vendidas.
- precio\_unidad: precio por unidad del producto.
- total\_venta: métrica calculada de manera persistente como  $\text{cantidad} * \text{precio\_unidad}$ , que permite analizar los ingresos generados.

Esta tabla concentra la información transaccional de ventas, y gracias a las claves foráneas puede contextualizarse desde múltiples perspectivas de análisis.

## 2. Tabla de dimensión: DimProducto:

Contiene la descripción detallada de los productos. Su propósito es enriquecer los análisis de ventas con atributos que caracterizan a cada producto.

Atributos: código de producto, nombre, proveedor, dimensiones, precios de venta y proveedor.

Relación con DimCategoria: incluye la clave ID\_categoria para agrupar productos según su categoría.

Permite responder preguntas como:

- ¿Qué productos generan mayores ingresos?
- ¿Cuáles son los proveedores con mejor desempeño?

## 3. Tabla de dimensión: DimCategoria:

Describe las categorías a las que pertenecen los productos.

Atributos: identificador de categoría, descripción textual y opcionalmente un recurso gráfico (imagen).

Relación: se vincula con DimProducto y con FactVentas a través de ID\_categoria.

Permite análisis agregados como:

- ¿Qué categoría concentra la mayor cantidad de productos vendidos?
- ¿Qué categorías son más rentables?

#### **4. Tabla de dimensión: DimCliente:**

Contiene información detallada de los clientes.

Atributos: nombre de cliente, contactos, teléfono, fax, ciudad, región, país, código postal y límite de crédito.

Esta dimensión soporta análisis orientados al cliente, como:

- ¿En qué países se concentra la mayor parte de las ventas?
- ¿Qué clientes generan mayores ingresos?

#### **5. Tabla de dimensión: DimTiempo:**

Es fundamental para los análisis temporales, ya que permite desagregar las ventas en distintos niveles de granularidad.

Atributos: fecha completa, día, mes, trimestre y año.

Con esta dimensión se pueden responder preguntas como:



- ¿En qué año se registró el mayor volumen de ventas?
- ¿Cuál es la tendencia de ventas por trimestre o mes?

#### **6. Tabla de dimensión: DimEmpleado:**

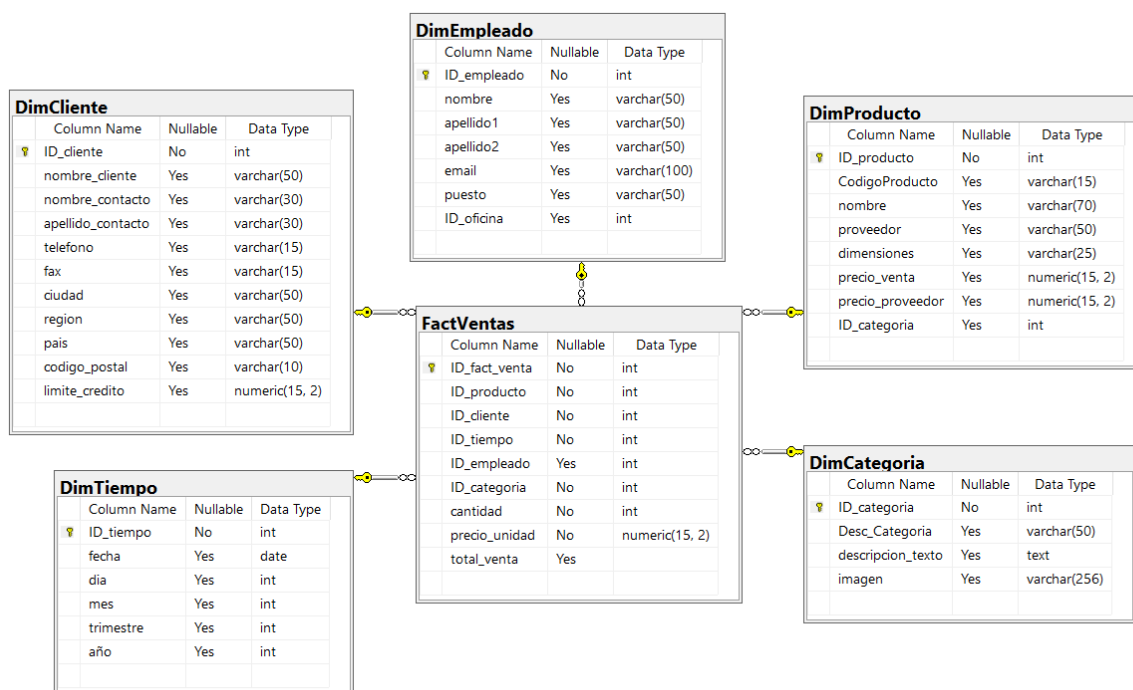
Permite asociar las ventas con los empleados de la empresa.

Atributos: nombre, apellidos, email, puesto e identificador de oficina.

Da soporte a análisis como:

- ¿Qué empleados han gestionado mayor volumen de ventas?
- ¿Qué oficinas concentran más transacciones?

Este modelo estrella permite analizar las ventas de Jardinería desde diferentes perspectivas: producto, categoría, cliente, tiempo y empleado, con métricas clave como la cantidad vendida y el total de ingresos generados.



El siguiente paso fue la creación de la base de datos `jardineria_staging`, cuyo propósito fue servir como un área intermedia de almacenamiento temporal para cargar los datos en su estado original (crudo) provenientes de la base de datos transaccional Jardinería. Esta fase no aplica aún procesos de limpieza o transformación, sino que garantiza que la información relevante esté disponible para posteriores análisis y depuración.

La estructura diseñada incluyó las tablas mínimas necesarias para responder a los requerimientos del negocio: productos, categorías, pedidos y detalles de pedidos.

### 1. Tabla `Stg_Categoria`:

Almacena la información de las categorías de productos. Incluye tanto descripciones textuales como recursos adicionales (texto, HTML e imagen) provenientes del sistema origen.

Clave primaria: Id\_Categoria.

Contenido: descripción corta, información textual y gráfica asociada a la categoría.

```
-- 1. CATEGORÍAS
CREATE TABLE Stg_Categoria (
    Id_Categoria INT PRIMARY KEY,
    Desc_Categoria VARCHAR(50),
    descripcion_texto VARCHAR(MAX),
    descripcion_html VARCHAR(MAX),
    imagen VARCHAR(256)
);
GO
```

## 2. Tabla Stg\_Producto:

Registra la información de los productos comercializados. Se estableció una relación con la categoría a la que pertenece cada producto.

Clave primaria: ID\_producto.

Clave foránea: Categoria, referenciada a Stg\_Categoria.

Contenido: código de producto, nombre, dimensiones, proveedor, descripción, cantidad en stock, precio de venta y precio de proveedor.

```
-- 2. PRODUCTOS
CREATE TABLE Stg_Producto (
    ID_producto INT PRIMARY KEY,
    CodigoProducto VARCHAR(15),
    nombre VARCHAR(70),
    Categoria INT NOT NULL,
    dimensiones VARCHAR(25),
    proveedor VARCHAR(50),
    descripcion VARCHAR(MAX),
    cantidad_en_stock SMALLINT,
    precio_venta NUMERIC(15,2),
    precio_proveedor NUMERIC(15,2),
    CONSTRAINT FK_StgProducto_Categoria FOREIGN KEY (Categoria)
        REFERENCES Stg_Categoria (Id_Categoria)
);
GO
```

### 3. Tabla Stg\_Pedido:

Concentra los pedidos realizados por los clientes. En esta etapa los datos se guardan tal como provienen del origen, sin filtros respecto al estado del pedido (entregado, pendiente o rechazado).

Clave primaria: ID\_pedido.

Contenido: fechas relevantes del pedido (pedido, esperada y entrega), estado del pedido, comentarios adicionales e identificador del cliente.

```
-- 3. PEDIDOS
CREATE TABLE Stg_Pedido (
    ID_pedido INT PRIMARY KEY,
    fecha_pedido DATE,
    fecha_esperada DATE,
    fecha_entrega DATE,
    estado VARCHAR(15),
    comentarios VARCHAR(MAX),
    ID_cliente INT
    -- No ponemos FK a cliente porque no lo cargamos en staging
);
GO
```

#### 4. Tabla Stg\_DetallePedido:

Registra los productos incluidos en cada pedido, con su cantidad y precio unitario.

Establece las relaciones necesarias tanto con los pedidos como con los productos.

Clave primaria: ID\_detalle\_pedido.

Claves foráneas:

ID\_pedido, referenciado a Stg\_Pedido.

ID\_producto, referenciado a Stg\_Producto.

Contenido: cantidad solicitada, precio por unidad y número de línea en el pedido.

```

-- 4. DETALLE DE PEDIDO
CREATE TABLE Stg_DetallePedido (
    ID_detalle_pedido INT PRIMARY KEY,
    ID_pedido INT NOT NULL,
    ID_producto INT NOT NULL,
    cantidad INT,
    precio_unidad NUMERIC(15,2),
    numero_linea SMALLINT,
    CONSTRAINT FK_StgDetallePedido_Pedido FOREIGN KEY (ID_pedido)
        REFERENCES Stg_Pedido (ID_pedido),
    CONSTRAINT FK_StgDetallePedido_Producto FOREIGN KEY (ID_producto)
        REFERENCES Stg_Producto (ID_producto)
);
GO

```

En estas tablas se insertaron directamente los registros provenientes de la base Jardinería sin aplicar transformaciones, asegurando así la conservación de los datos originales y la posibilidad de auditar el proceso.

```

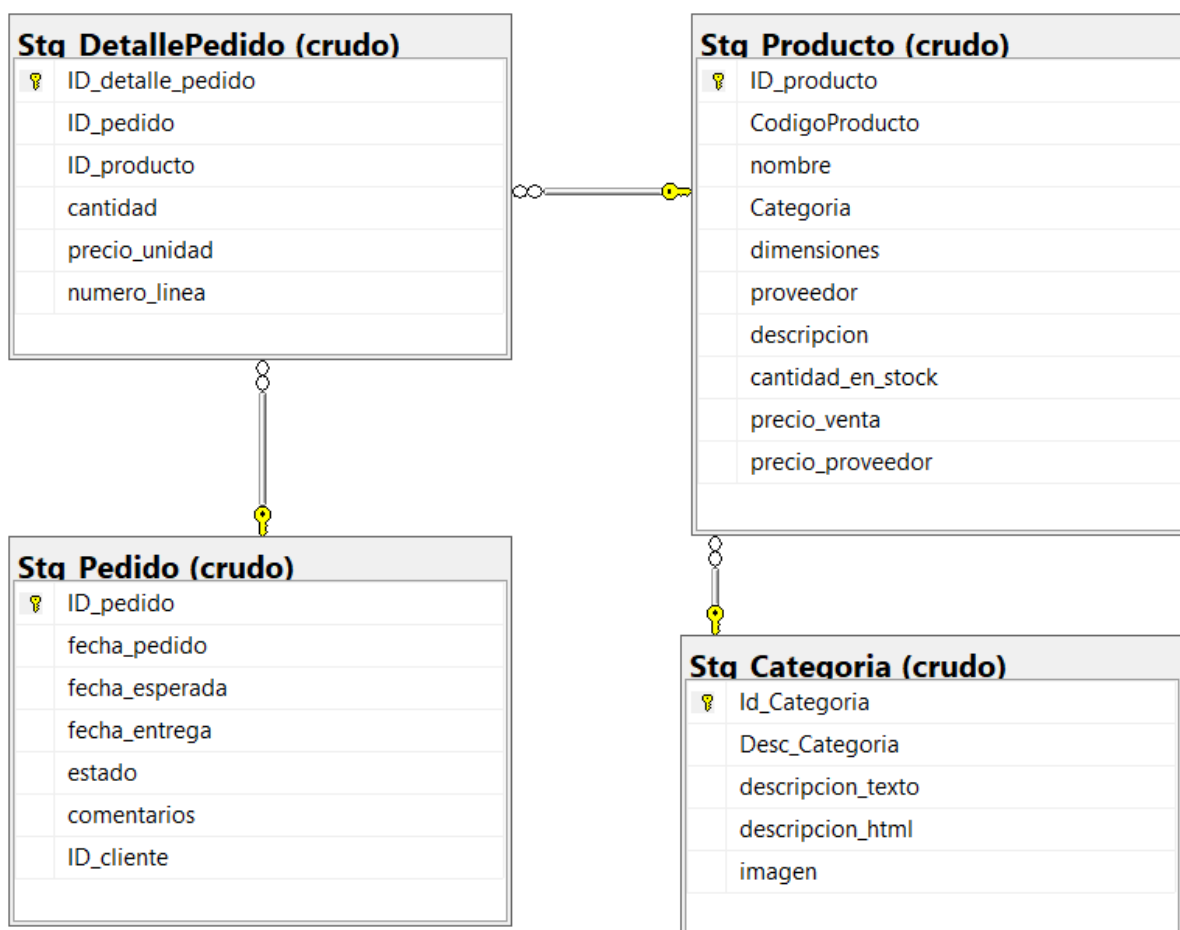
-- Carga en crudo desde la BD jardineria
-- Categorías
INSERT INTO Stg_Categoria
SELECT Id_Categoria, Desc_Categoria, descripcion_texto, descripcion_html, imagen
FROM jardineria.dbo.Categoria_producto;
GO

-- Productos
INSERT INTO Stg_Producto
SELECT ID_producto,CodigoProducto, nombre, Categoria, dimensiones, proveedor, descripcion,
       cantidad_en_stock, precio_venta, precio_proveedor
FROM jardineria.dbo.Producto;
GO

-- Pedidos
INSERT INTO Stg_Pedido
SELECT ID_pedido, fecha_pedido, fecha_esperada, fecha_entrega, estado, comentarios, ID_cliente
FROM jardineria.dbo.Pedido;
GO

-- Detalle de pedido
INSERT INTO Stg_DetallePedido
SELECT ID_detalle_pedido, ID_pedido, ID_producto, cantidad, precio_unidad, numero_linea
FROM jardineria.dbo.Detalle_pedido;
GO

```



Una vez cargados los datos en el staging, con el fin de evaluar la calidad e integridad de los datos cargados en crudo dentro del staging, se realizaron diversas consultas de validación. Estas se organizaron por cada tabla principal del modelo: categorías, productos, pedidos y detalle de pedido.

### 1. Tabla Stg\_Categoria:

- **Consulta: Categorías sin descripción.**

```

SELECT Id_Categoria
FROM Stg_Categoria
WHERE Desc_Categoria IS NULL OR LTRIM(RTRIM(Desc_Categoria)) = '';

```

100 %

Results Messages

Id_Categoria
--------------

Resultado: Sin resultados. Todas las categorías contaban con un valor en el campo Desc\_Categoria.

La nomenclatura de categorías es consistente.

- **Consulta: Categorías sin productos asociados.**

```

-- b. Categorías sin productos asociados
SELECT c.Id_Categoria, c.Desc_Categoria
FROM Stg_Categoria c
LEFT JOIN Stg_Producto p ON c.Id_Categoria = p.Categoria
WHERE p.ID_producto IS NULL;

```

100 %

Results Messages

	Id_Categoria	Desc_Categoria
1	1	Herbaceas

Resultado: Se detectó la categoría Herbáceas (Id\_Categoria = 1) sin ningún producto asignado.

Se trata de una categoría huérfana que debe ser depurada o excluida en la transformación, ya que no aporta valor al análisis.

## 2. Tabla Stg\_Producto:

- **Consulta: Productos sin categoría válida.**



```
-- a. Productos sin categoría válida
SELECT p.ID_producto, p.nombre
FROM Stg_Producto p
LEFT JOIN Stg_Categoria c ON p.Categoria = c.Id_Categoria
WHERE c.Id_Categoria IS NULL;
```

100 %

Results Messages

ID_producto	nombre
-------------	--------

Resultado: Sin resultados. Todos los productos tienen una categoría registrada en Stg\_Categoria.

No hay rupturas en la relación entre productos y categorías.

- **Consulta: Productos con precios inválidos (cero, nulos o negativos).**

```
-- b. Precios inválidos
SELECT ID_producto, nombre, precio_venta, precio_proveedor
FROM Stg_Producto
WHERE precio_venta IS NULL OR precio_venta <= 0
OR precio_proveedor IS NULL OR precio_proveedor <= 0;
```

100 %

Results Messages

	ID_producto	nombre	precio_venta	precio_proveedor
1	5	Ajedrea	1.00	0.00
2	6	Lavándula Dentata	1.00	0.00
3	7	Mejorana	1.00	0.00
4	8	Melissa	1.00	0.00
5	9	Mentha Sativa	1.00	0.00
6	10	Petrosilium Horte...	1.00	0.00
7	11	Salvia Mix	1.00	0.00
8	12	Thymus Citriodra ...	1.00	0.00
9	13	Thymus Vulgaris	1.00	0.00
10	14	Santolina Chama...	1.00	0.00

Resultado: 10 productos detectados, principalmente con precio\_proveedor = 0.00 y precio\_venta = 1.00.

Si precio\_proveedor está en 0 pero precio\_venta es válido, pueden mantenerse para ventas, ya que lo que interesa es el precio de venta.

- **Consulta: Productos sin descripción.**

```
-- c. Productos sin descripción
SELECT ID_producto, nombre
FROM Stg_Producto
WHERE descripcion IS NULL OR descripcion = '';
```

100 %

Results		Messages
	ID_producto	nombre
1	14	Santolina Chamaecyparys
2	15	Expositor Cítricos Mix
3	18	Nogal
4	32	Rosal bajo 1Å -En maceta-inicio brotación
5	33	ROSAL TREPADOR
6	34	Camelia Blanco, Chrysler Rojo, Soraya Naranja,
7	36	Landora Amarillo, Rose Gaujard bicolor blanco-rojo
8	37	Kordes Perfect bicolor rojo-amarillo, Roundelay ro...
9	38	Pitimini rojo
10	39	Rosal copa
11	61	Membrillero Gigante de Wranja
12	66	Nogal Común
13	67	Para Uva de Mesa

Resultado: 153 productos sin información en el campo descripcion.

Limita la calidad de los reportes y catálogos, aunque no afecta directamente los cálculos de ventas. Se puede mantener como “No disponible”, pero se debe registrar en un reporte de calidad.

- **Consulta: Dimensiones mal formateadas.**

```
-- d. Dimensiones mal formateadas
SELECT ID_producto, nombre, dimensiones
FROM Stg_Producto
WHERE dimensiones IS NULL
      OR dimensiones LIKE '%/%'
      OR dimensiones LIKE '%-%';
```

100 %

Results Messages

	ID_producto	nombre	dimensiones
1	5	Ajedrea	15-20
2	6	Lavándula Dentata	15-20
3	7	Mejorana	15-20
4	8	Melissa	15-20
5	9	Mentha Sativa	15-20
6	10	Petrosilium Hortense (Peregil)	15-20
7	11	Salvia Mix	15-20
8	12	Thymus Citriodra (Tomillo limón)	15-20
9	13	Thymus Vulgaris	15-20
10	14	Santolina Chamaecyparys	15-20
11	15	Expositor Cítricos Mix	100-120
12	17	Nectarina	8/10
13	18	Mozal	8/10

Resultado: 209 productos con valores nulos o formatos inconsistentes (ej. “15-20” o “8/10”).

Los datos carecen de estandarización, lo que puede complicar análisis técnicos o comparativos, no afecta ventas ni conteo de productos, pero afecta análisis de atributos, normalizar formatos de dimensiones si luego se usan para análisis de inventario, aunque no es crítico para el Data Mart actual.

### 3. Tabla Stg\_Pedido:

- **Consulta: Pedidos rechazados o pendientes.**

```
-- 3. Stg_Pedido
-- a. Pedidos rechazados o pendientes (no deberían cargarse en ventas)
SELECT ID_pedido, estado
FROM Stg_Pedido
WHERE estado IN ('Pendiente', 'Rechazado');
```

100 %

Results Messages

	ID_pedido	estado
1	3	Rechazado
2	4	Pendiente
3	7	Pendiente
4	8	Pendiente
5	9	Pendiente
6	11	Rechazado
7	16	Pendiente
8	17	Pendiente
9	18	Rechazado
10	20	Rechazado
11	22	Rechazado
12	23	Rechazado
13	26	Rechazado

Resultado: 54 pedidos encontrados en estado Pendiente o Rechazado.

Estos registros no deben cargarse en el análisis de ventas, ya que no representan ingresos reales.

- **Consulta: Fechas inconsistentes.**

```
-- b. Fechas inconsistentes
SELECT ID_pedido, fecha_pedido, fecha_esperada, fecha_entrega
FROM Stg_Pedido
WHERE fecha_esperada < fecha_pedido
      OR (fecha_entrega IS NOT NULL AND fecha_entrega < fecha_pedido);
```

100 %

Results Messages

	ID_pedido	fecha_pedido	fecha_esperada	fecha_entrega
1	42	2009-04-01	2009-03-04	2009-03-07
2	43	2009-04-03	2009-03-04	2009-03-05
3	44	2009-04-15	2009-03-17	2009-03-17
4	77	2009-02-07	2008-02-17	NULL

Resultado: 4 pedidos con problemas de fechas: fechas esperadas anteriores a la fecha de pedido o entregas registradas antes de la compra.

Inconsistencias temporales que requieren corrección o exclusión, pues afectan la dimensión tiempo y los análisis históricos. Para Stg\_Tiempo se usará fecha\_pedido como fecha de referencia y se excluirán registros que no tengan fecha\_pedido válida.

Las columnas fecha\_esperada y fecha\_entrega pueden mantenerse para auditoría, pero no se usarán en el análisis de ventas.

#### 4. Tabla Stg\_DetallePedido:

- **Consulta: Ventas con cantidades inválidas ( $\leq 0$ ).**

```
-- 4. Stg_DetallePedido

-- a. Ventas con cantidades inválidas
SELECT ID_detalle_pedido, ID_pedido, ID_producto, cantidad
FROM Stg_DetallePedido
WHERE cantidad <= 0;
```

100 %

Results Messages

ID_detalle_pedido	ID_pedido	ID_producto	cantidad
-------------------	-----------	-------------	----------

Resultado: Sin resultados. Todas las cantidades fueron positivas.

- **Consulta: Ventas con precios inválidos ( $\leq 0$ ).**

```
-- b. Ventas con precios inválidos
SELECT ID_detalle_pedido, ID_pedido, ID_producto, precio_unidad
FROM Stg_DetallePedido
WHERE precio_unidad <= 0;
```

100 %

Results Messages

ID_detalle_pedido	ID_pedido	ID_producto	precio_unidad
-------------------	-----------	-------------	---------------

Resultado: Sin resultados. Todos los precios unitarios asociados a ventas eran válidos.

- **Consulta: Detalles sin referencia a pedidos.**

```
-- c. Detalle sin referencia a pedido
SELECT d.ID_detalle_pedido, d.ID_pedido
FROM Stg_DetallePedido d
LEFT JOIN Stg_Pedido p ON d.ID_pedido = p.ID_pedido
WHERE p.ID_pedido IS NULL;
```

100 %

Results Messages

ID_detalle_pedido	ID_pedido
-------------------	-----------

Resultado: Sin resultados. Todos los detalles estaban asociados a pedidos válidos.

- **Consulta: Detalles sin referencia a productos.**

```
-- d. Detalle sin referencia a producto
SELECT d.ID_detalle_pedido, d.ID_producto
FROM Stg_DetallePedido d
LEFT JOIN Stg_Producto p ON d.ID_producto = p.ID_producto
WHERE p.ID_producto IS NULL;
```

100 %

Results Messages

ID_detalle_pedido	ID_producto
-------------------	-------------

Resultado: Sin resultados. Todos los detalles estaban correctamente vinculados a productos existentes.

Con base en estos hallazgos se definieron reglas de transformación, las cuales se aplicaron dentro del mismo staging, creando un esquema limpio paralelo al crudo. La transformación se diseñó para garantizar que únicamente los datos consistentes, completos y relevantes pasaran al entorno limpio, asegurando la calidad de la información que posteriormente se cargaría en el Data Mart. El proceso se organizó en cinco etapas clave:

## 1. Organización del entorno: separación entre crudo y limpio:

Se crearon dos esquemas:

- ❖ crudo: contiene las tablas originales con datos cargados directamente desde la base de datos de origen.
- ❖ limpio: alberga las tablas depuradas, estructuradas y preparadas para análisis.

Las tablas Stg\_Categoria, Stg\_Producto, Stg\_Pedido y Stg\_DetallePedido se movieron al esquema crudo, separando así los datos originales de los transformados.

Esto permitió trabajar con datos depurados sin alterar el histórico en crudo.

```
-- TRANSFORMACIÓN DE DATOS

-- Crear esquemas si no existen
IF NOT EXISTS (SELECT 1 FROM sys.schemas WHERE name = 'crudo')
    EXEC('CREATE SCHEMA crudo');
GO

IF NOT EXISTS (SELECT 1 FROM sys.schemas WHERE name = 'limpio')
    EXEC('CREATE SCHEMA limpio');
GO

-- Mover las tablas originales
ALTER SCHEMA crudo TRANSFER dbo.Stg_Categoria;
ALTER SCHEMA crudo TRANSFER dbo.Stg_Producto;
ALTER SCHEMA crudo TRANSFER dbo.Stg_Pedido;
ALTER SCHEMA crudo TRANSFER dbo.Stg_DetallePedido;
```



```
-- Crear tablas transformadas en limpio
```

```
-- Categorías limpias
```

```
CREATE TABLE limpio.Categoria (  
    Id_Categoria INT PRIMARY KEY,  
    Desc_Categoria VARCHAR(50)  
);
```

```
-- Productos limpios
```

```
CREATE TABLE limpio.Producto (  
    ID_producto INT PRIMARY KEY,  
    nombre VARCHAR(70),  
    Categoria INT NOT NULL,  
    precio_venta NUMERIC(15,2),  
    descripcion VARCHAR(MAX),  
    CONSTRAINT FK_Producto_Categoria FOREIGN KEY (Categoria) REFERENCES limpio.Categoria(Id_Categoria)  
);
```

```
-- Pedidos limpios
```

```
CREATE TABLE limpio.Pedido (  
    ID_pedido INT PRIMARY KEY,  
    fecha_pedido DATE,  
    estado VARCHAR(15)  
);
```

```
-- Detalle de pedidos limpios
```

```
CREATE TABLE limpio.DetallePedido (  
    ID_detalle_pedido INT PRIMARY KEY,  
    ID_pedido INT NOT NULL,  
    ID_producto INT NOT NULL,  
    cantidad INT,  
    precio_unidad NUMERIC(15,2),  
    CONSTRAINT FK_DetPedido_Pedido FOREIGN KEY (ID_pedido) REFERENCES limpio.Pedido(ID_pedido),  
    CONSTRAINT FK_DetPedido_Producto FOREIGN KEY (ID_producto) REFERENCES limpio.Producto(ID_producto)  
);
```

```

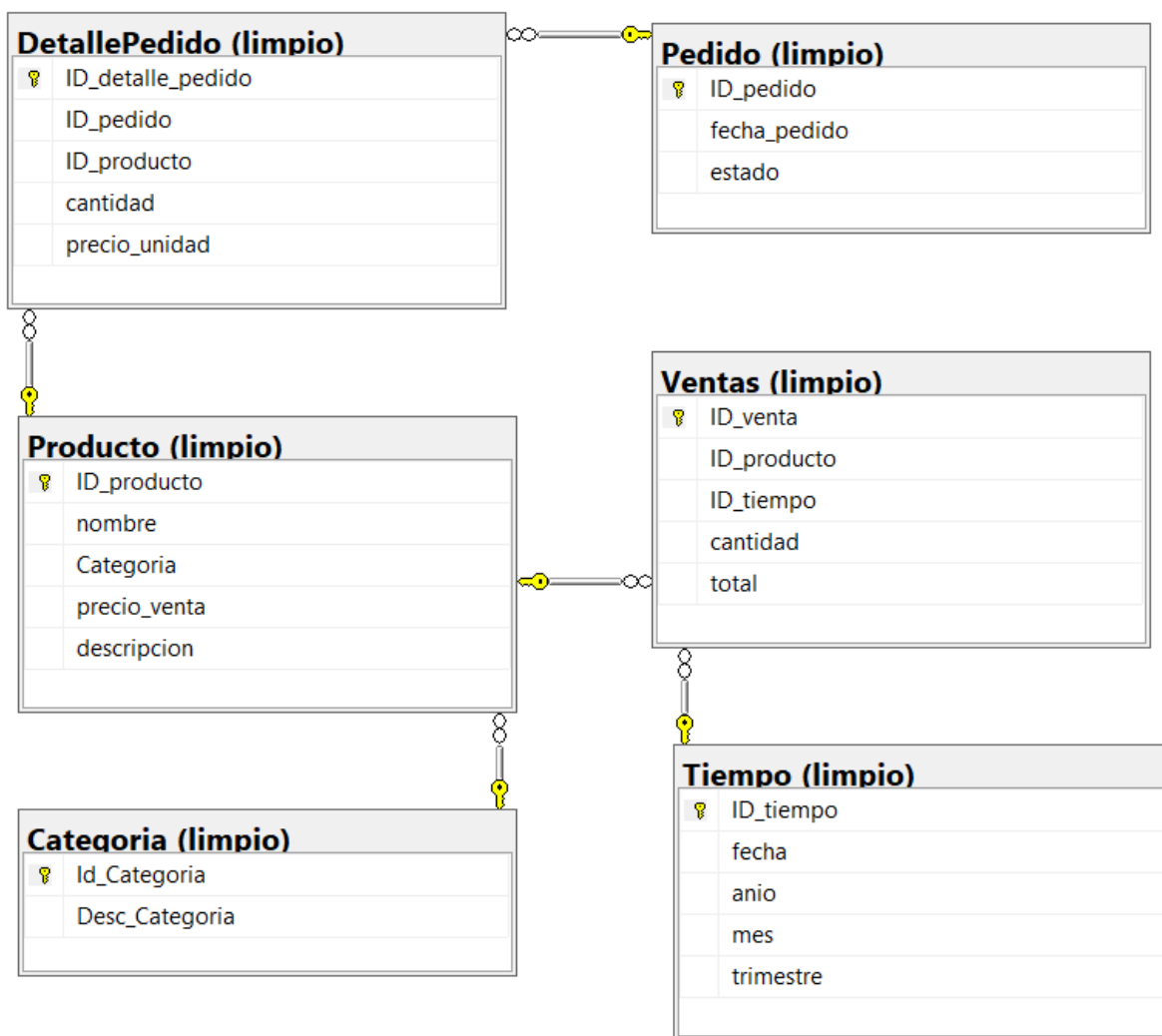
-- Dimensión tiempo
CREATE TABLE limpio.Tiempo (
    ID_tiempo INT PRIMARY KEY,
    fecha DATE NOT NULL,
    anio INT NOT NULL,
    mes INT NOT NULL,
    trimestre INT NOT NULL
);

```

```

-- Hechos de ventas
CREATE TABLE limpio.Ventas (
    ID_venta INT IDENTITY(1,1) PRIMARY KEY,
    ID_producto INT NOT NULL,
    ID_tiempo INT NOT NULL,
    cantidad INT NOT NULL,
    total NUMERIC(15,2) NOT NULL,
    CONSTRAINT FK_Ventas_Producto FOREIGN KEY (ID_producto) REFERENCES limpio.Producto(ID_producto),
    CONSTRAINT FK_Ventas_Tiempo FOREIGN KEY (ID_tiempo) REFERENCES limpio.Tiempo(ID_tiempo)
);

```



## 2. Transformación de categorías (limpio.Categoria):

- Se eliminaron categorías huérfanas, es decir, aquellas sin productos asociados (ejemplo: Herbáceas).
- Solo se cargaron las categorías con al menos un producto válido vinculado.

Con esto se garantiza que todas las categorías en el modelo tengan relevancia comercial.

```

-- Insertar datos transformados en limpio

-- Categorías limpias (excluyendo huérfanas como Herbaceas)
INSERT INTO limpio.Categoria (Id_Categoria, Desc_Categoria)
SELECT c.Id_Categoria, c.Desc_Categoria
FROM crudo.Stg_Categoria c
WHERE EXISTS (
    SELECT 1 FROM crudo.Stg_Producto p WHERE p.Categoria = c.Id_Categoria
);

```

### 3. Transformación de productos (limpio.Producto):

- Se descartaron productos con precios inválidos (precio de venta  $\leq 0$ ).
- Se incluyeron únicamente los productos con categoría válida en el staging limpio.
- Las descripciones vacías fueron reemplazadas con el texto estándar "No disponible".

```

-- Productos limpios (solo con precio_venta > 0 y categoría válida)
INSERT INTO limpio.Producto (ID_producto, nombre, Categoria, precio_venta, descripcion)
SELECT p.ID_producto,
       p.nombre,
       p.Categoria,
       p.precio_venta,
       ISNULL(NULLIF(LTRIM(RTRIM(p.descripcion)), ''), 'No disponible')
FROM crudo.Stg_Producto p
WHERE p.precio_venta > 0
      AND EXISTS (SELECT 1 FROM limpio.Categoria c WHERE c.Id_Categoria = p.Categoria);

```

Así, cada producto tiene un precio válido y una descripción, evitando valores nulos o inconsistentes en los reportes.

### 4. Transformación de pedidos y detalles:

- Pedidos (limpio.Pedido)
  - Se filtraron únicamente los pedidos en estado Entregado.
  - Se descartaron pedidos con fechas nulas o inconsistentes.

```
-- Pedidos limpios (solo entregados y con fecha válida)
INSERT INTO limpio.Pedido (ID_pedido, fecha_pedido, estado)
SELECT p.ID_pedido, p.fecha_pedido, p.estado
FROM crudo.Stg_Pedido p
WHERE p.estado = 'Entregado'
AND p.fecha_pedido IS NOT NULL;
```

- Detalle de pedidos (limpio.DetallePedido)
  - Se incluyeron solo registros de pedidos válidos (estado Entregado).
  - Se excluyeron líneas con cantidades  $\leq 0$  o precios unitarios no válidos.
  - Se validó que cada detalle tuviera referencias correctas a un producto y a un pedido.

```
-- Detalles de pedido limpios
INSERT INTO limpio.DetallePedido (ID_detalle_pedido, ID_pedido, ID_producto, cantidad, precio_unidad)
SELECT dp.ID_detalle_pedido,
       dp.ID_pedido,
       dp.ID_producto,
       dp.cantidad,
       dp.precio_unidad
FROM crudo.Stg_DetallePedido dp
JOIN limpio.Pedido pl ON dp.ID_pedido = pl.ID_pedido
JOIN limpio.Producto pr ON dp.ID_producto = pr.ID_producto
WHERE dp.cantidad > 0 AND dp.precio_unidad > 0;
```

Esto asegura que el análisis de ventas refleje únicamente operaciones efectivas, eliminando registros pendientes o rechazados.

## 5. Construcción de dimensiones de tiempo y hechos:

- Dimensión tiempo (limpio.Tiempo)
  - Se generó una clave surrogate (ID\_tiempo) con numeración secuencial.
  - Se cargaron las fechas distintas de pedidos entregados.
  - Se descompuso cada fecha en año, mes y trimestre, lo cual habilita análisis temporales detallados.

```
-- Dimensión tiempo
INSERT INTO limpio.Tiempo (ID_tiempo, fecha, anio, mes, trimestre)
SELECT ROW_NUMBER() OVER (ORDER BY p.fecha_pedido) AS ID_tiempo,
       p.fecha_pedido,
       YEAR(p.fecha_pedido) AS anio,
       MONTH(p.fecha_pedido) AS mes,
       DATEPART(QUARTER, p.fecha_pedido) AS trimestre
FROM (
  SELECT DISTINCT fecha_pedido
  FROM limpio.Pedido
  WHERE fecha_pedido IS NOT NULL
) p;
```

- Tabla de hechos (limpio.Ventas)
  - Se integraron las ventas a partir de los detalles de pedidos válidos.
  - Se calcularon métricas clave:
    - Cantidad vendida
    - Total de venta = cantidad × precio unidad
  - Se enlazaron las ventas con las dimensiones de producto y tiempo.

```
-- Ventas (hechos)
INSERT INTO limpio.Ventas (ID_producto, ID_tiempo, cantidad, total)
SELECT dp.ID_producto,
       t.ID_tiempo,
       dp.cantidad,
       dp.cantidad * dp.precio_unidad AS total
FROM limpio.DetallePedido dp
JOIN limpio.Pedido p ON dp.ID_pedido = p.ID_pedido
JOIN limpio.Tiempo t ON p.fecha_pedido = t.fecha;
```

Finalmente, se construyó la base de datos `jardineria_data_mart`, la cual fue diseñada bajo un modelo estrella simplificado, su estructura se enfocó únicamente en las entidades y métricas necesarias para responder a las tres preguntas principales del negocio:

- ¿Cuál es el producto más vendido?
- ¿Qué categoría concentra mayor número de productos?
- ¿En qué año se registró el mayor volumen de ventas?

### 1. Tablas de Dimensión:

Las dimensiones representan los ejes de análisis sobre los cuales se pueden explorar las métricas de ventas.

- **Dim\_Categoria:**

Contiene las categorías de productos.

Columnas principales:

- ❖ Id\_Categoria: clave primaria.
- ❖ Desc\_Categoria: descripción de la categoría.

Rol en el análisis: permite agrupar los productos y conocer qué categorías concentran mayor cantidad de artículos y ventas.

```
-- 1. DIMENSIONES
-- Dimensión Categoría
CREATE TABLE Dim_Categoria (
    Id_Categoria INT PRIMARY KEY,
    Desc_Categoria VARCHAR(50)
);
```

- **Dim\_Producto:**

Almacena la información de los productos comercializados.

Columnas principales:

- ❖ Id\_Producto: clave primaria.
- ❖ Nombre: nombre del producto.
- ❖ Id\_Categoria: clave foránea hacia Dim\_Categoria.
- ❖ Precio\_Venta: valor unitario de venta.
- ❖ Descripcion: detalle descriptivo.

Rol en el análisis: posibilita identificar cuáles productos generan mayor volumen de ventas.



```
-- Dimensión Producto
CREATE TABLE Dim_Producto (
    Id_Producto INT PRIMARY KEY,
    Nombre VARCHAR(70),
    Id_Categoria INT NOT NULL,
    Precio_Venta NUMERIC(15,2),
    Descripcion VARCHAR(MAX),
    CONSTRAINT FK_Producto_Categoria FOREIGN KEY (Id_Categoria)
        REFERENCES Dim_Categoria(Id_Categoria)
);
```

- **Dim\_Tiempo:**

Representa la dimensión temporal necesaria para el análisis histórico.

Columnas principales:

- ❖ Id\_Tiempo: clave primaria.
- ❖ Fecha: fecha de referencia.
- ❖ Año, Mes, Trimestre: jerarquías temporales.

Rol en el análisis: permite desglosar las ventas por año, mes o trimestre, identificando tendencias en el tiempo, como el año con mayor facturación.

```
-- Dimensión Tiempo
CREATE TABLE Dim_Tiempo (
    Id_Tiempo INT PRIMARY KEY,
    Fecha DATE NOT NULL,
    Año INT NOT NULL,
    Mes INT NOT NULL,
    Trimestre INT NOT NULL
);
```

## 2. Tabla de Hechos:

- **Fact\_Ventas:**

Es el núcleo del modelo estrella y concentra las métricas de negocio.

Columnas principales:

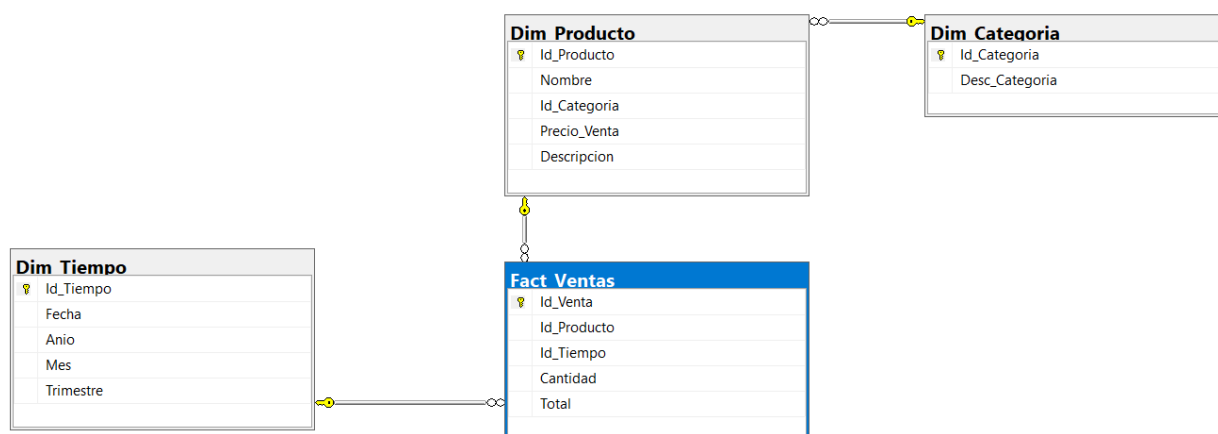
- ❖ Id\_Venta: clave primaria surrogate.
- ❖ Id\_Producto: referencia a la dimensión producto.
- ❖ Id\_Tiempo: referencia a la dimensión tiempo.
- ❖ Cantidad: número de unidades vendidas.
- ❖ Total: importe monetario total de la venta.

Relaciones:

- ❖ Se enlaza con Dim\_Producto para analizar el volumen de ventas por artículo.
- ❖ Se enlaza con Dim\_Tiempo para examinar la evolución temporal.

Rol en el análisis: concentra las métricas cuantitativas que sirven de base para responder las preguntas clave del negocio.

```
-- 2. TABLA DE HECHOS
CREATE TABLE Fact_Ventas (
  Id_Venta INT IDENTITY(1,1) PRIMARY KEY,
  Id_Producto INT NOT NULL,
  Id_Tiempo INT NOT NULL,
  Cantidad INT NOT NULL,
  Total NUMERIC(15,2) NOT NULL,
  CONSTRAINT FK_FactVentas_Producto FOREIGN KEY (Id_Producto) REFERENCES Dim_Producto(Id_Producto),
  CONSTRAINT FK_FactVentas_Tiempo FOREIGN KEY (Id_Tiempo) REFERENCES Dim_Tiempo(Id_Tiempo)
);
```



La carga de datos en este data mart se realizó directamente desde las tablas transformadas del esquema limpio en staging, asegurando que solo datos válidos y consistentes llegaran al modelo analítico.

```

-- Cargar Categorías
INSERT INTO Dim_Categoria (Id_Categoria, Desc_Categoria)
SELECT Id_Categoria, Desc_Categoria
FROM jardineria_staging.limpio.Categoria;

-- Cargar Productos
INSERT INTO Dim_Producto (Id_Producto, Nombre, Id_Categoria, Precio_Venta, Descripcion)
SELECT p.ID_producto, p.nombre, p.Categoria, p.precio_venta, p.descripcion
FROM jardineria_staging.limpio.Producto p
JOIN jardineria_staging.limpio.Categoria c ON p.Categoria = c.Id_Categoria;

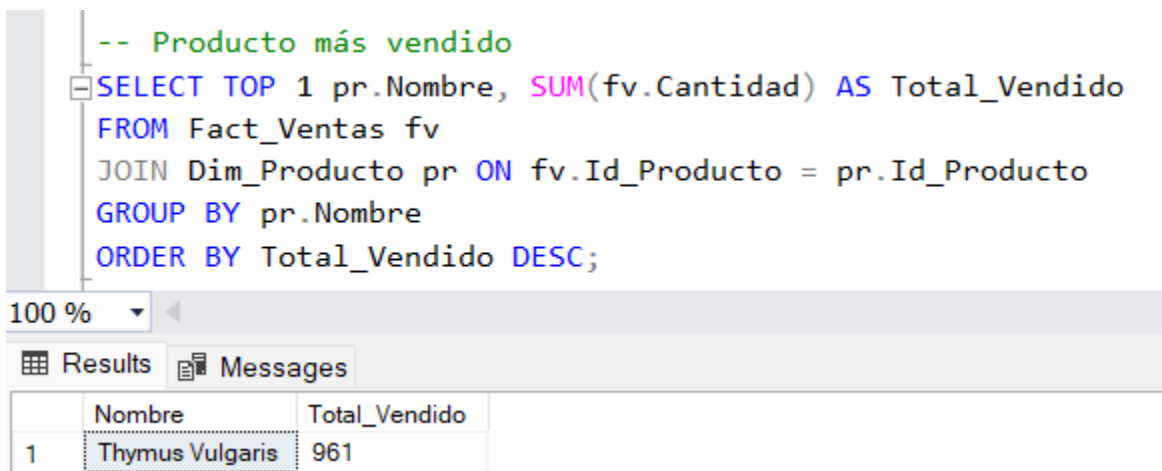
-- Cargar Tiempo
INSERT INTO Dim_Tiempo (Id_Tiempo, Fecha, Anio, Mes, Trimestre)
SELECT ID_tiempo, fecha, anio, mes, trimestre
FROM jardineria_staging.limpio.Tiempo;

-- Cargar Hechos de Ventas
INSERT INTO Fact_Ventas (Id_Producto, Id_Tiempo, Cantidad, Total)
SELECT dp.ID_producto,
       t.ID_tiempo,
       dp.cantidad,
       dp.cantidad * dp.precio_unidad AS total
FROM jardineria_staging.limpio.DetallePedido dp
JOIN jardineria_staging.limpio.Pedido p ON dp.ID_pedido = p.ID_pedido
JOIN jardineria_staging.limpio.Tiempo t ON p.fecha_pedido = t.fecha;

```

## CONSULTAS DEL NEGOCIO EN EL DATA MART

### 1. ¿Cuál es el producto más vendido?



The screenshot shows a SQL query editor with the following text:

```
-- Producto más vendido
SELECT TOP 1 pr.Nombre, SUM(fv.Cantidad) AS Total_Vendido
FROM Fact_Ventas fv
JOIN Dim_Producto pr ON fv.Id_Producto = pr.Id_Producto
GROUP BY pr.Nombre
ORDER BY Total_Vendido DESC;
```

Below the query, the results are displayed in a table with two columns: 'Nombre' and 'Total\_Vendido'. The first row shows 'Thymus Vulgaris' with a total of 961 units sold.

	Nombre	Total_Vendido
1	Thymus Vulgaris	961

Resultado: Thymus Vulgaris con un total de 961 unidades vendidas.

Este producto se posiciona como el de mayor rotación en la empresa. El alto volumen de ventas refleja una fuerte demanda, lo que puede deberse a factores como su popularidad en el mercado, precio competitivo o estacionalidad en su consumo. Desde el punto de vista de gestión, es importante garantizar niveles adecuados de inventario para evitar quiebres de stock y aprovechar la demanda sostenida. También podría evaluarse la rentabilidad de este producto: aunque sea el más vendido en cantidad, es necesario revisar si genera el mayor margen de ganancia.

### 2. ¿Qué categoría concentra mayor número de productos?

```
-- Categoría con más productos
SELECT TOP 1 c.Desc_Categoria, COUNT(p.Id_Producto) AS Cantidad_Productos
FROM Dim_Producto p
JOIN Dim_Categoria c ON p.Id_Categoria = c.Id_Categoria
GROUP BY c.Desc_Categoria
ORDER BY Cantidad_Productos DESC;
```

100 %

Results Messages

	Desc_Categoria	Cantidad_Productos
1	Ornamentales	154

Resultado: Ornamentales con 154 productos registrados.

Esta categoría agrupa la mayor diversidad de artículos dentro del catálogo de la empresa.

Un portafolio amplio en esta categoría indica que la empresa ha enfocado sus esfuerzos en atender la demanda del mercado ornamental, posiblemente por su atractivo comercial o por ser un segmento en crecimiento. Sin embargo, tener más productos no necesariamente significa mayores ventas; será necesario contrastar esta diversidad con la facturación total de la categoría para evaluar su rentabilidad. La amplitud también puede implicar mayores costos de gestión (almacenamiento, marketing, logística), por lo que conviene analizar si todos los productos dentro de la categoría son rentables o si existe un exceso de variedad.

### 3. ¿En qué año se registró el mayor volumen de ventas?

```
-- Año con más ventas
SELECT TOP 1 t.Anio, SUM(fv.Total) AS Ventas_Totales
FROM Fact_Ventas fv
JOIN Dim_Tiempo t ON fv.Id_Tiempo = t.Id_Tiempo
GROUP BY t.Anio
ORDER BY Ventas_Totales DESC;
```

100 %

Results Messages

	Anio	Ventas_Totales
1	2008	92854.00

Resultado: 2008 con un total de 92,854.00 unidades monetarias en ventas.

El año 2008 representó el mejor desempeño de la empresa en términos de facturación.

Esto puede estar relacionado con factores externos (tendencias del mercado, mayor demanda en ese año) o internos (mejor estrategia comercial, incorporación de nuevos clientes, políticas de precios). Comparar 2008 con otros años permitirá identificar si este fue un pico aislado o parte de una tendencia creciente en las ventas. La empresa debería analizar qué estrategias implementadas en ese periodo contribuyeron al éxito, con el fin de replicarlas en años posteriores.

## CONCLUSIONES

La construcción del modelo estrella permitió definir la estructura analítica de la base de datos, identificando de manera clara las tablas de dimensiones y la tabla de hechos necesarias para responder a los requerimientos del negocio. Esto proporcionó una base sólida para organizar la información y facilitar su explotación en el Data Mart.

La creación de la base de datos staging con datos en crudo fue un paso fundamental, ya que permitió separar el entorno transaccional de la fase de preparación de datos. En este espacio fue posible realizar un análisis exhaustivo de las inconsistencias, tales como productos con precios inválidos, descripciones faltantes, categorías sin productos asociados, pedidos rechazados o pendientes, y fechas incoherentes.

El proceso de transformación dentro del staging garantizó la limpieza, normalización y depuración de los datos. Gracias a este paso, se filtraron registros no aptos para el análisis (como pedidos no entregados o productos sin precios válidos) y se estandarizaron campos, asegurando la calidad y coherencia de la información antes de cargarla en el Data Mart.

La carga de datos limpios en el Data Mart consolidó la información en un modelo estrella eficiente y orientado al análisis, donde se integraron las dimensiones de producto, categoría y tiempo con la tabla de hechos de ventas. Esto permitió contar con un repositorio confiable y optimizado para consultas estratégicas.

El proceso completo demostró la importancia de aplicar un flujo ETL (Extract, Transform, Load) en proyectos de inteligencia de negocios. Extraer datos en crudo, transformarlos para garantizar su calidad

y cargarlos en un esquema analítico no solo asegura confiabilidad en la información, sino que también mejora significativamente la capacidad de respuesta a las necesidades del negocio.

Finalmente, la empresa pudo responder de manera clara a preguntas clave: identificar el producto más vendido (*Thymus Vulgaris*), la categoría con más productos (Ornamentales) y el año con mayores ventas (2008). Estos resultados evidencian cómo un proceso bien diseñado de integración y análisis de datos puede convertirse en una herramienta estratégica para la toma de decisiones empresariales.



## BIBLIOGRAFÍA

Informatica. (s. f.). What is ETL? (Extract, Transform, Load). Recuperado de

<https://www.informatica.com/resources/articles/what-is-etl.html>

IBM. (s. f.). What is ETL (Extract, Transform, Load)? Recuperado de

<https://www.ibm.com/think/topics/etl>

Microsoft. (2016). Data Warehouse Fast Track Reference Guide for SQL Server 2016. Recuperado de

<https://download.microsoft.com>

WhereScape. (2023). Delivering Data Warehouses on Microsoft SQL Server. Recuperado de

<https://www.wherescape.com>

Oracle. (2011). Oracle Data Integrator Best Practices for a Data Warehouse. Recuperado de

<https://www.oracle.com>

Obisesan, A. (2021). Implementing Data Warehouse On-premises. Theseus.fi. Recuperado de

<https://www.theseus.fi>

Cuzzocrea, A., et al. (2021). BIcenter: A collaborative Web ETL solution. Journal of Computer

Languages, ScienceDirect. Recuperado de <https://www.sciencedirect.com>

Choudhary, S., et al. (2019). ETL Framework for Real-Time Business Intelligence. Journal of Big Data,

PMC. <https://pmc.ncbi.nlm.nih.gov/articles/PMC6737132>

- Vášek, J. (2014). Process of Transformation, Storage and Data Analysis for Data Mart Enlargement. Acta Informatica Pragensia. Recuperado de <https://www.researchgate.net>
- Vassiliadis, P., et al. (2014). Optimizing ETL Dataflow Using Shared Caching and Parallelization Methods. arXiv. <https://arxiv.org/abs/1409.1639>
- Akhtar, F. (2022). METL: a modern ETL pipeline with a dynamic mapping matrix. arXiv. <https://arxiv.org/abs/2203.10289>
- Ramírez, D., et al. (2025). FlowETL: An Autonomous Example-Driven Pipeline for Data Engineering. arXiv. <https://arxiv.org/abs/2507.23118>
- Zhao, J., et al. (2018). On-Demand Big Data Integration: A Hybrid ETL Approach for Reproducible Scientific Research. arXiv. <https://arxiv.org/abs/1804.08985>
- Systems Personnel. (2018). ETL Professional. Recuperado de <https://systemspersonnel.com>
- Squared. (2023). ETL vs ELT: Explained. Recuperado de <https://squared.ai>
- Airbyte. (2023). ETL Use Cases: 8 Industry-Specific Challenges ETL Solves. Recuperado de <https://airbyte.com>
- Oracle. (2010). Data Mart Setup Guide. Recuperado de <https://docs.oracle.com>
- SAS Institute. (2009). Building a Data Warehouse with SAS DI Studio and MS SQL Server. Recuperado de <https://support.sas.com>

Integrate.io. (2021). Understanding the Necessity of ETL in Data Integration. Recuperado de <https://www.integrate.io>

Matillion. (2022). The Ultimate Guide to ETL. Recuperado de <https://www.matillion.com>

Singh, K. (2023). Metadata-driven ETL frameworks: A paradigm shift in real-time analytics architecture. ResearchGate. <https://www.researchgate.net>

Wikipedia. (2023). Extract, transform, load. Recuperado de [https://en.wikipedia.org/wiki/Extract%2C\\_transform%2C\\_load](https://en.wikipedia.org/wiki/Extract%2C_transform%2C_load)

Wikipedia. (2023). Área de stage (datos). Recuperado de [https://es.wikipedia.org/wiki/%C3%81rea\\_de\\_stage\\_%28datos%29](https://es.wikipedia.org/wiki/%C3%81rea_de_stage_%28datos%29)

Wikipedia. (2023). Extract, load, transform (ELT). Recuperado de [https://en.wikipedia.org/wiki/Extract%2C\\_load%2C\\_transform](https://en.wikipedia.org/wiki/Extract%2C_load%2C_transform)

WhereScape. (2023). Delivering Data Warehouses on Microsoft SQL Server (White Paper). Recuperado de <https://www.wherescape.com>

Obisesan, A. (2021). Implementing Data Warehouse On-premises (tesis). Theseus.fi. Recuperado de <https://www.theseus.fi>

Microsoft. (2016). Data Warehouse Fast Track Reference Guide for SQL Server. Recuperado de <https://download.microsoft.com>

Oracle. (2011). Oracle Data Integrator Best Practices. Recuperado de <https://www.oracle.com>

Airbyte. (2023). ETL Use Cases: Industry-Specific Challenges ETL Solves. Recuperado de <https://airbyte.com>

Vassiliadis, P., et al. (2014). Optimizing ETL Dataflow Using Shared Caching and Parallelization. arXiv. <https://arxiv.org/abs/1409.1639>