



Institución Universitaria Digital de Antioquia IUDIGITAL  
Faculta de Ingeniería y Ciencias Agropecuarias  
Ingeneria de Software y datos

## **EVIDENCIA DE APRENDIZAJE 2. CREACIÓN DE UNA BASE DE DATOS DE STAGING**

**Omar Aleiser Alvarez Laverde**  
**PREICA2401B010094**

Bucaramanga, Santander  
12 Mayo 2024

## Introducción

El propósito de este estudio es registrar el desarrollo de una base de datos de puesta en escena (StagingJardineria.bak) modelada según la estructura relacional de la base de datos Jardinería (jardineria.bak). Este proyecto está diseñado para simplificar el análisis y la manipulación de datos para la empresa ficticia Jardinería, con el objetivo de mejorar la comprensión de sus operaciones y facilitar la toma de decisiones informadas.

## Objetivo general

Implementar una base de datos de puesta en escena (StagingJardineria.bak) basada en el modelo relacional de la base de datos Jardinería (jardineria.bak), con el fin de mejorar la eficiencia en el análisis y procesamiento de datos para la empresa Jardinería.

## Objetivos especificados

- Diseñar un esquema de base de datos de puesta en escena (StagingJardineria.bak) que refleje de manera precisa y eficiente la estructura relacional de la base de datos Jardinería.
- Desarrollar un proceso automatizado para la carga inicial de datos desde la base de datos Jardinería a la base de datos de puesta en escena, garantizando la integridad y consistencia de los datos.

## Planteamiento del problema

La base de datos Jardinería contiene información sobre productos, pedidos, clientes, empleados y otros datos relevantes. Sin embargo, su estructura actual no está optimizada para realizar análisis de ventas de manera eficiente. Es necesario diseñar un modelo estrella que organice los datos de forma adecuada para facilitar consultas analíticas y obtener información valiosa sobre las ventas.

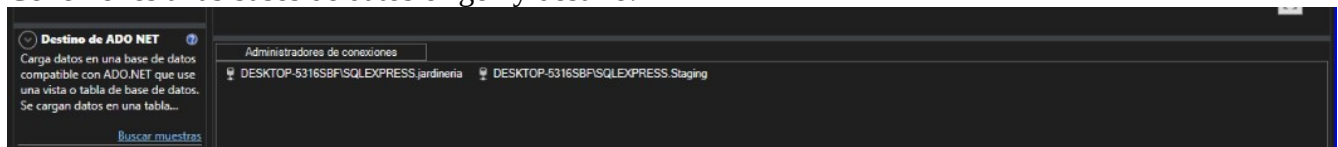
## Análisis del problema

Para abordar el problema, se analizó la estructura de la base de datos Jardinería y se identificaron las tablas relevantes, como: detalle\_pedido, producto, pedido y cliente. Estas tablas contienen información clave para el análisis de ventas, pero se encuentran normalizadas y dispersas, lo que dificulta el rendimiento de las consultas analíticas.

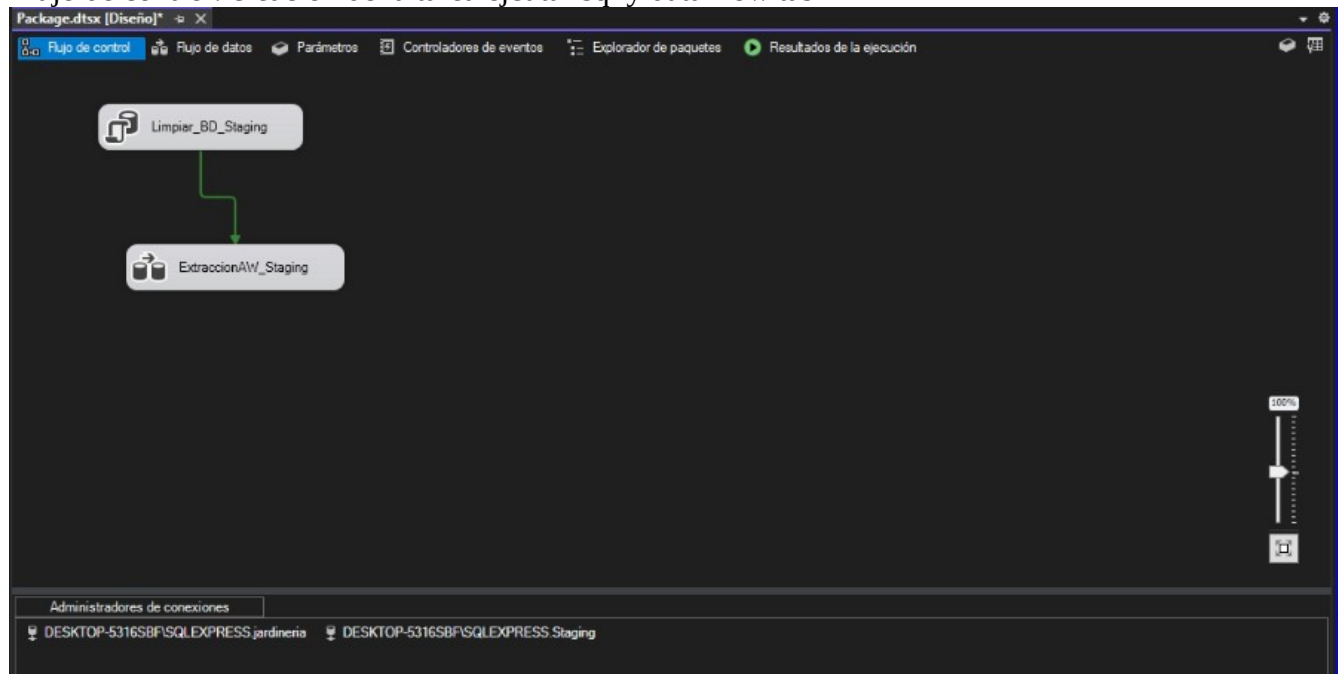
## Propuesta de solución

### Microsoft Integration Services Project:

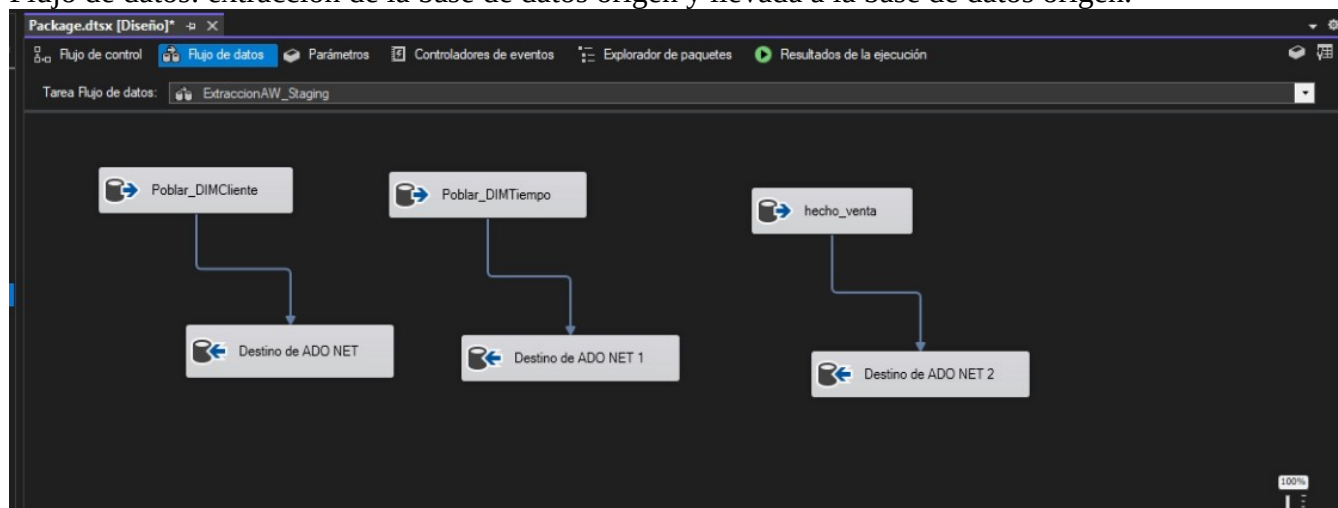
Conexiones a las bases de datos origen y destino.



Flujo de control: creación de la tarea ejecutar sql y data flow task



Flujo de datos: extracción de la base de datos origen y llevada a la base de datos origen.



Consultas de la extracción de la tabla origen (jardineria)

**Poblar la\_DIMCliente**

```

SELECT      ID_cliente, nombre_cliente, nombre_contacto, apellido_contacto, telefono, fax,
linea_direccion1, linea_direccion2, ciudad, region, pais, codigo_postal,
ID_empleado_rep_ventas, limite_credito
FROM        cliente

```

### Poblar\_DIMTiempo

```

SELECT      ID_producto, CodigoProducto, nombre, Categoria, dimensiones, proveedor,
descripcion, cantidad_en_stock, precio_venta, precio_proveedor
FROM        producto

```

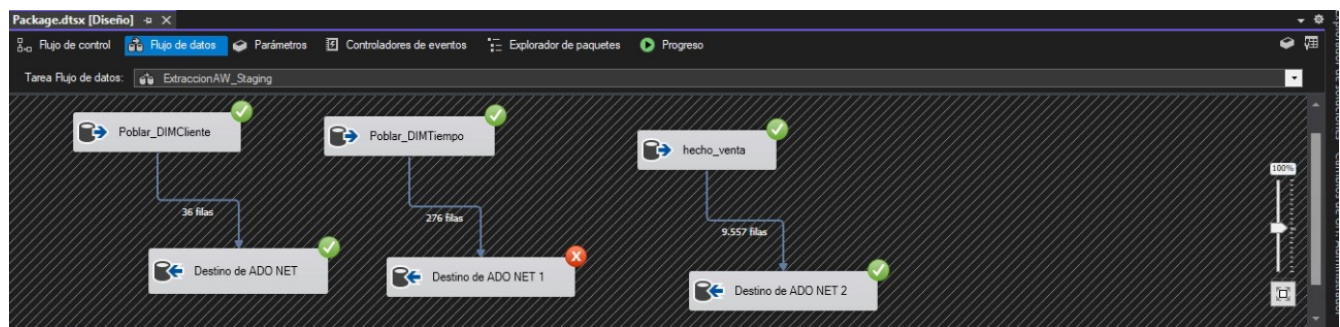
### Poblar\_hecho\_venta

```

SELECT      cliente.ID_cliente, producto.ID_producto, detalle_pedido.cantidad,
detalle_pedido.precio_unidad
FROM        detalle_pedido INNER JOIN
           producto ON detalle_pedido.ID_producto = producto.ID_producto CROSS
JOIN
           cliente

```

### Prueba de ejecución



### Correcciones a la Entrega 1

Se realiza corrección con relación a la primera entrega. Se excluye las entidades DIMCategoria, DIMPedidos por comentario sugerido por el profesor. Se genera un nuevo script con un nuevo modelo entidad relación MER.

### Conclusiones

Este proyecto tuvo como objetivo principal simplificar la manipulación de datos y mejorar la comprensión de las operaciones de Jardinería, lo que a su vez facilitaría la toma de decisiones informadas. Tras un meticuloso proceso de extracción y comprensión de la estructura de la base de datos original, se diseñó un esquema de base de datos de puesta en escena que reflejara con precisión dicha estructura, pero con mejoras para optimizar el análisis y la manipulación de datos.

Una de las claves del éxito fue el desarrollo de un proceso automatizado para la carga inicial de datos desde la base de datos original a la de puesta en escena, asegurando la integridad y consistencia de los mismos. Además, se implementaron transformaciones durante este proceso para mejorar la calidad y coherencia de los datos en la base de datos de puesta en escena.

Una vez implementada, esta base de datos de puesta en escena permitió una mayor eficiencia operativa al simplificar el acceso y análisis de datos para los usuarios de Jardinería. Además, facilitó la toma de decisiones informadas al proporcionar una visión clara y estructurada de las operaciones de la empresa.

## **Anexos**

Script sql server staging

-- Crear la tabla DIMCliente

```
CREATE TABLE DIMCliente (  
    ID_Cliente INT IDENTITY(1,1) PRIMARY KEY,  
    Nombre_Cliente VARCHAR(50),  
    Nombre_Contacto VARCHAR(30),  
    Apellido_Contacto VARCHAR(30),  
    Telefono VARCHAR(15),  
    Fax VARCHAR(15),  
    Direccion_Linea1 VARCHAR(15),  
    Direccion_Linea2 VARCHAR(15),  
    Ciudad VARCHAR(50),  
    Region VARCHAR(50),  
    Pais VARCHAR(50),  
   Codigo_Postal VARCHAR(50),  
    ID_Empleado_Rep_Ventas INT,  
    Limite_Credito INT  
);
```

-- Crear la tabla DIMTiempo

```
CREATE TABLE DIMTiempo (  
    ID_Tiempo INT IDENTITY(1,1) PRIMARY KEY,  
    Fecha DATE NOT NULL,  
    Anio INT,  
    Trimestre INT,  
    Semana INT,  
    Mes INT  
);
```

-- Crear la tabla DIMProductos

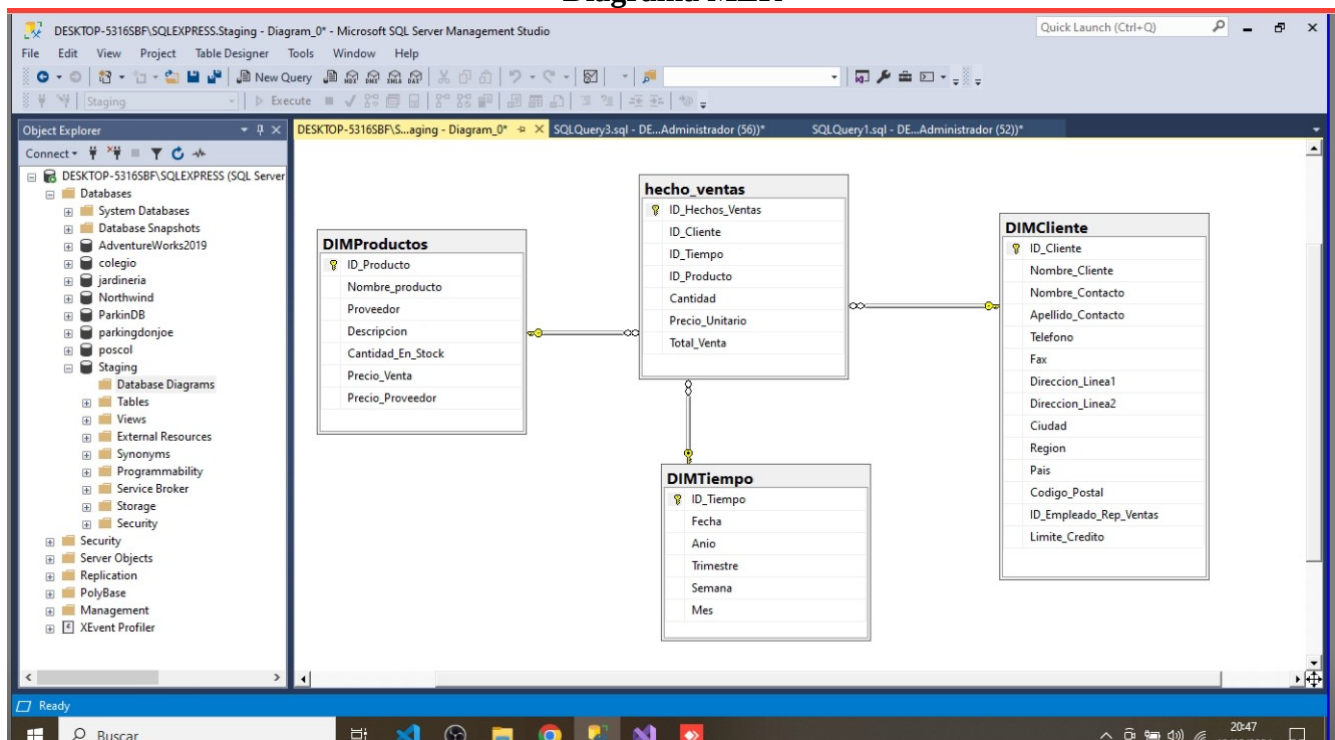
```
CREATE TABLE DIMProductos (  
    ID_Producto INT IDENTITY(1,1) PRIMARY KEY,  
    Nombre_producto VARCHAR(70),  
    Proveedor VARCHAR(50),  
    Descripcion VARCHAR(200),  
    Cantidad_En_Stock INT,  
    Precio_Venta INT,
```

```

Precio_Proveedor INT
);
-- Crear la tabla hecho_ventas
CREATE TABLE hecho_ventas (
    ID_Hechos_Ventas INT IDENTITY(1,1) PRIMARY KEY,
    ID_Cliente INT FOREIGN KEY REFERENCES DIMCliente(ID_Cliente),
    ID_Tiempo INT FOREIGN KEY REFERENCES DIMTiempo(ID_Tiempo),
    ID_Producto INT FOREIGN KEY REFERENCES DIMProductos(ID_Producto),
    Cantidad INT,
    Precio_Unitario INT,
    Total_Venta INT
);

```

## Diagrama MER



## Bibliografía

P. Muñoz. “Desarrollo de una arquitectura de Big Data para registros mercantiles”. Trabajo de grado, UCV, Caracas, 2016.

## Link acceso a los archivos .bak

<https://drive.google.com/drive/folders/1x7d3lnkUQ3ZPoX05XnP92IxOoSs1r13W?usp=sharing>