

## BASES DE DATOS II

DOCENTE:

Victor Hugo Mercado

ESTUDIANTE:

Gloria Yurleny Gómez González

Tarea

S30 -Evidencia de aprendizaje 3. Proceso de transformación de datos y carga en el data mart final

MARZO 2025 Medellín Antioquia

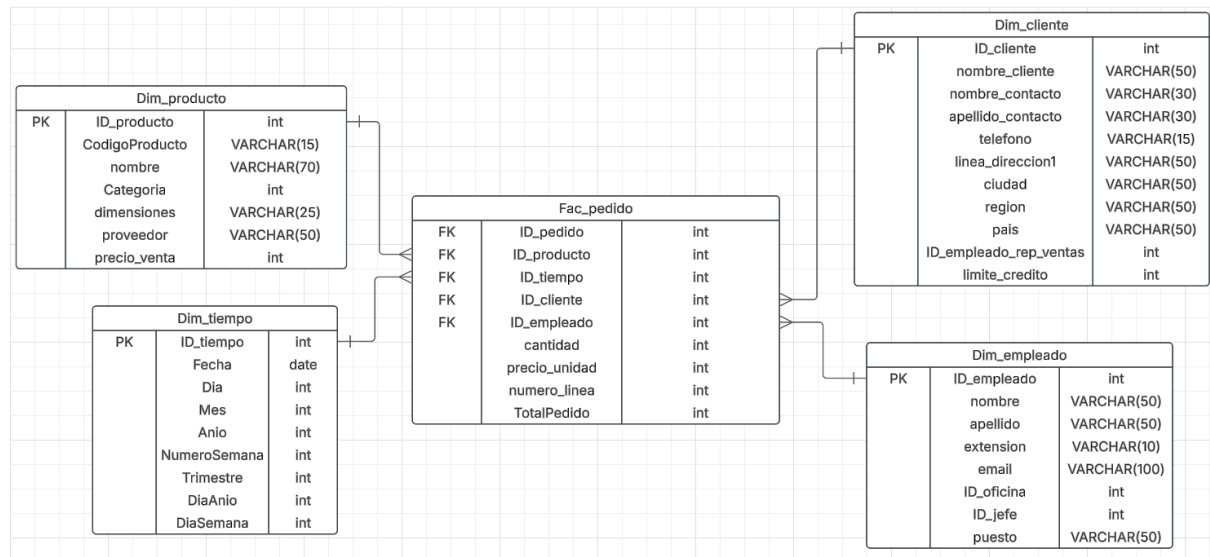


IU Digital  
de Antioquia

## Etapa de Preparación

Las preguntas planteadas al inicio de la actividad son: identificar el producto más vendido, la categoría con más productos y el año con más ventas. Las tablas que contienen los datos relevantes para responder son pedido, producto, detalle\_pedido, empleado, cliente y la tabla de creación de tiempo. El modelo estrella permite simplificar el análisis con estas tablas específicas.

A continuación, se muestra el diseño del modelo estrella donde se pueden observar las dimensiones, la tabla de hechos, sus campos, tipos de datos y relaciones.



## Etapa de Extracción

En este proceso se expone el análisis y diseño de una base de datos Staging para la base de datos Jardinería. Esta base de datos Staging tiene como propósito intermedio la limpieza, transformación y preparación de los datos antes de ser cargados en el modelo del data mart.

Se realizó un análisis de la base de datos de jardinería identificando cada tabla presente y sus posibles relaciones. Se determinó que la información de las tablas Pedido, Producto, Empleado, Cliente y detalle\_pedido son la clave para responder preguntas sobre las ventas, los productos y los clientes, serán tenidas en cuenta para la construcción de la base de datos Staging ya que proveen la información relevante para el caso de estudio.

A continuación se expone el proceso de creación de las tablas en la base de datos de staging.

Con el siguiente comando en SQL se crea la base de datos de Staging:

```
CREATE DATABASE [StagingBD2Gloria]
```

En la aplicación visual studio se crean flujos de datos independientes para cada una de las tablas de base de datos de jardinería que van a ser extraídos a la base de datos

staging a cada uno se le adiciona una tarea de SQL para que elimine los datos existentes una vez se ejecute la tarea y queden los datos actualizados que son obtenidos desde el ambiente de producción, se muestra en la siguiente imagen.



Tabla de tiempo:

Se seleccionan los campos de interés de la tabla pedido que es la que provee fechas para obtener los datos de tiempo, nos aseguramos que no hayan datos nulos en fecha de entrega para evitar errores en la transformación y serán combinados por medio de una unión de los datos con el siguiente script de SQL en el origen:

```
SELECT distinct [fecha_pedido] FROM [jardineria].[dbo].[pedido]
union all SELECT distinct [fecha_esperada] FROM [jardineria].[dbo].[pedido]
union all SELECT distinct [fecha_entrega] FROM [jardineria].[dbo].[pedido] WHERE
fecha_entrega IS NOT NULL;

order by [fecha_pedido]
```

Seguidamente se crea la tabla en la base de datos staging con el siguiente script de SQL en el destino:

```
CREATE TABLE "STtiempo" (
    "fecha_pedido" date)
```

Tabla pedido:

Se seleccionan los campos de interés de la tabla pedido con el siguiente script de SQL en el origen:

```
SELECT ID_pedido, fecha_esperada, fecha_pedido, fecha_entrega, estado, comentarios,  
ID_cliente
```

```
FROM pedido
```

Seguidamente se crea la tabla en la base de datos staging con el siguiente script de SQL en el destino:

```
CREATE TABLE "STpedido" ( "ID_pedido" int, "fecha_esperada" date, "fecha_pedido"  
date, "fecha_entrega" date, "estado" nvarchar(15), "comentarios" nvarchar(max),  
"ID_cliente" int)
```

Tabla de producto:

Se seleccionan los campos de interés de la tabla producto con el siguiente script de SQL en el origen:

```
SELECT ID_producto, CodigoProducto, nombre, Categoria, dimensiones, proveedor,  
precio_venta
```

```
FROM producto
```

Seguidamente se crea la tabla en la base de datos staging con el siguiente script de SQL en el destino:

```
CREATE TABLE "STproducto" ("ID_producto" int, "CodigoProducto" nvarchar(15),  
"nombre" nvarchar(70), "Categoria" int, "dimensiones" nvarchar(25), "proveedor"  
nvarchar(50), "precio_venta" numeric(15,2))
```

Tabla detalle\_pedido:

Se seleccionan los campos de interés de la tabla detalle\_pedido con el siguiente script de SQL en el origen:

```
SELECT detalle_pedido.*
```

```
FROM detalle_pedido
```

Seguidamente se crea la tabla en la base de datos staging con el siguiente script de SQL en el destino:

```
CREATE TABLE "STDetallePedido" ("ID_detalle_pedido" int, "ID_pedido" int,  
"ID_producto" int, "cantidad" int, "precio_unidad" numeric(15,2), "numero_linea" smallint)
```



#### Tabla de Cliente:

Se seleccionan los campos de interés de la tabla Cliente con el siguiente script de SQL en el origen:

```
SELECT ID_cliente, nombre_cliente, nombre_contacto, apellido_contacto, telefono,  
linea_direccion1, ciudad, region, pais, ID_empleado_rep_ventas, limite_credito  
  
FROM cliente
```

Seguidamente se crea la tabla en la base de datos staging con el siguiente script de SQL en el destino:

```
CREATE TABLE "STCliente" ("ID_cliente" int, "nombre_cliente" nvarchar(50),  
"nombre_contacto" nvarchar(30), "apellido_contacto" nvarchar(30), "telefono"  
nvarchar(15), "linea_direccion1" nvarchar(50), "ciudad" nvarchar(50), "region"  
nvarchar(50), "pais" nvarchar(50), "ID_empleado_rep_ventas" int, "limite_credito"  
numeric(15,2))
```

#### Tabla de Empleado:

Se seleccionan los campos de interés de la tabla Empleado con el siguiente script de SQL en el origen:

```
SELECT ID_empleado, nombre, apellido1, extension, email, ID_oficina, ID_jefe, puesto  
  
FROM empleado
```

Seguidamente se crea la tabla en la base de datos staging con el siguiente script de SQL en el destino:

```
CREATE TABLE "STEmpleado" ("ID_empleado" int, "nombre" nvarchar(50), "apellido1"  
nvarchar(50), "extension" nvarchar(10), "email" nvarchar(100), "ID_oficina" int, "ID_jefe"  
int, "puesto" nvarchar(50))
```

De esta manera ya se tienen configuradas en la base de datos de Staging las tablas necesarias para desarrollar la solución requerida.

## **Etapas de Transformación**

El proceso parte desde la base de datos de Staging donde se toma cada tabla y se determina qué campos se requiere modificar tales como, campos nulos, operaciones o modificaciones de texto que hacen que los datos queden legibles para la presentación al

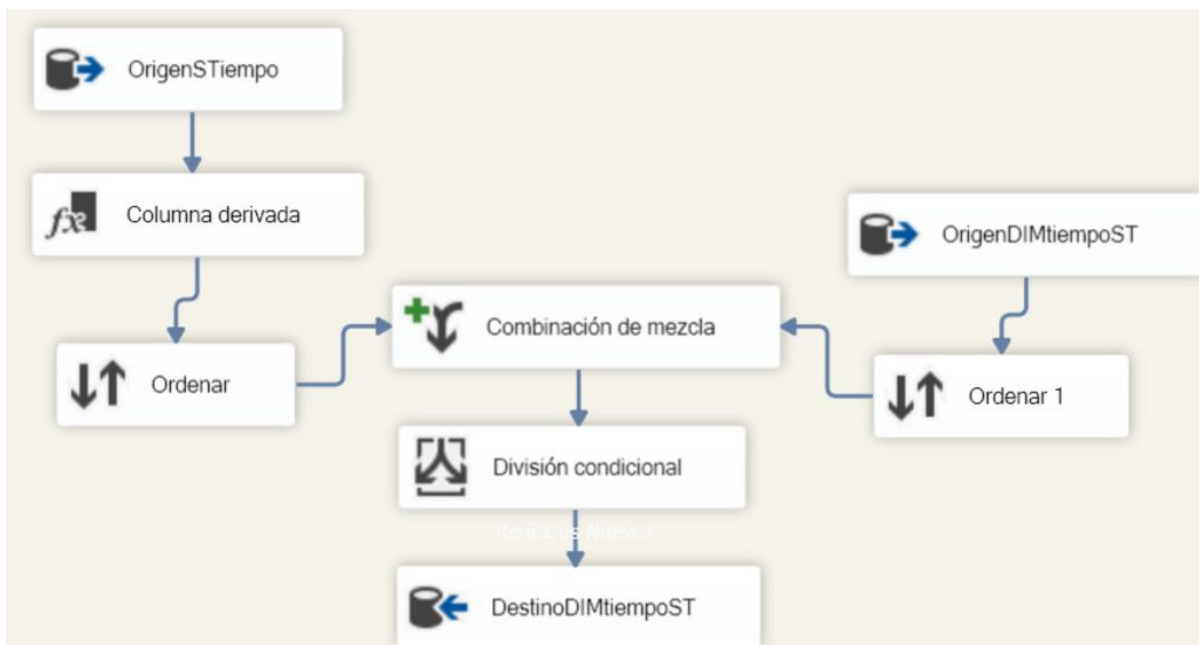
cliente. Se crean flujos de datos independientes para cada una de las tablas como se muestra en la siguiente imagen.



A continuación, se expone el proceso llevado a cabo, a cada una de las tablas en mención se le agrega su primary key al momento de crear su dimensión en la misma base de datos Staging.

Tabla DIMtiempo:

El origen será la tabla obtenida en la etapa de extracción y se transformará su estructura con el siguiente diagrama de flujo.



Se agregan nuevas columnas con la opción columna derivada como se muestra en la siguiente imagen.

Derived Column Name	Derived Column	Expression
Dia	<add as new column>	DAY(fecha_pedido)
Mes	<add as new column>	MONTH(fecha_pedido)
Anio	<add as new column>	YEAR(fecha_pedido)
NumeroSemana	<add as new column>	DATEPART("WW",fecha_pedido)
Trimestre	<add as new column>	DATEPART("quarter",fecha_pedido)
DiaAnio	<add as new column>	DATEPART("dy",fecha_pedido)
DiaSemana	<add as new column>	DATEPART("dw",fecha_pedido)

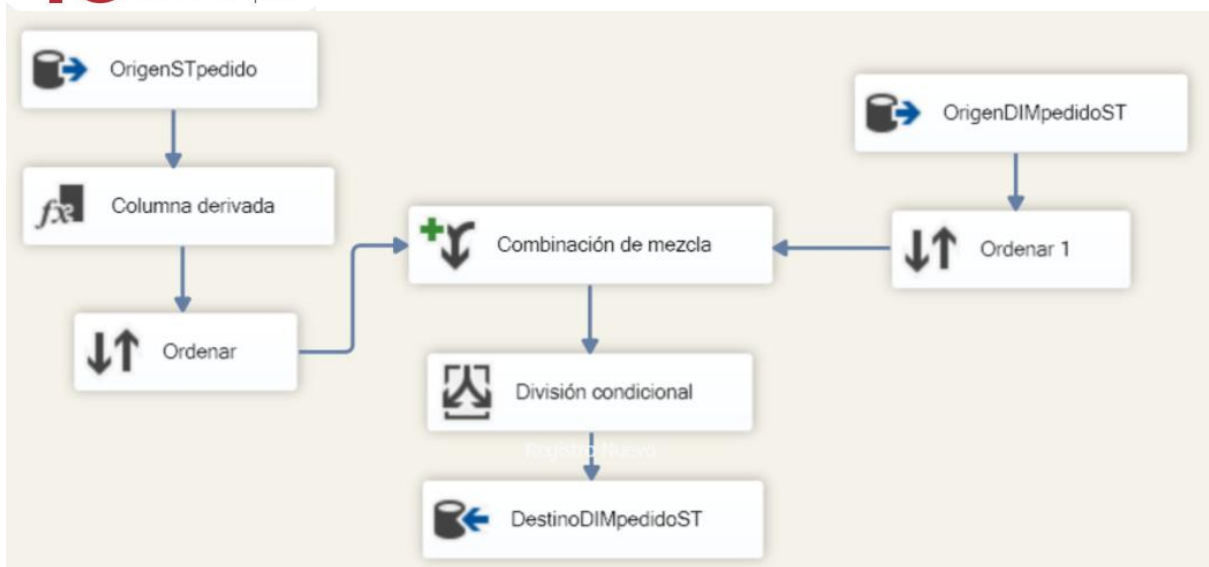
Seguidamente se ordena y se pasa por una combinación que contiene como origen la tabla de destino al final del diagrama de flujo, en el siguiente paso con la división condicional se evalúa si los datos que van a ser ingresados ya están en la tabla de destino, de lo contrario se agregarán solamente los nuevos registros.

Creación de la dimensión DIMtiempo: la tabla de destino resultado de la transformación se crea con el siguiente comando de SQL.

```
CREATE TABLE "DIMtiempoST" (IDDimTiempo int identity(1,1) primary key,
"fecha_pedido" date, "Dia" int, "Mes" int, "Anio" int, "NumeroSemana" int, "Trimestre" int,
"DiaAnio" int, "DiaSemana" int)
```

Tabla DIMpedido:

El origen será la tabla obtenida en la etapa de extracción y se transformará su estructura con el siguiente diagrama de flujo.



Se reemplazan columnas existentes con la opción columna derivada como se muestra en la siguiente imagen.

Derived Column Name	Derived Column	Expression
fecha_entrega	Reemplazar 'fecha_entrega'	REPLACENULL(fecha_entrega,"2030-12-31")
comentarios	Reemplazar 'comentarios'	REPLACENULL((DT_WSTR,4000)comentarios,"N/A")

Seguidamente se ordena y se pasa por una combinación que contiene como origen la tabla de destino al final del diagrama de flujo, en el siguiente paso con la división condicional se evalúa si los datos que van a ser ingresados ya están en la tabla de destino, de lo contrario se agregarán solamente los nuevos registros.

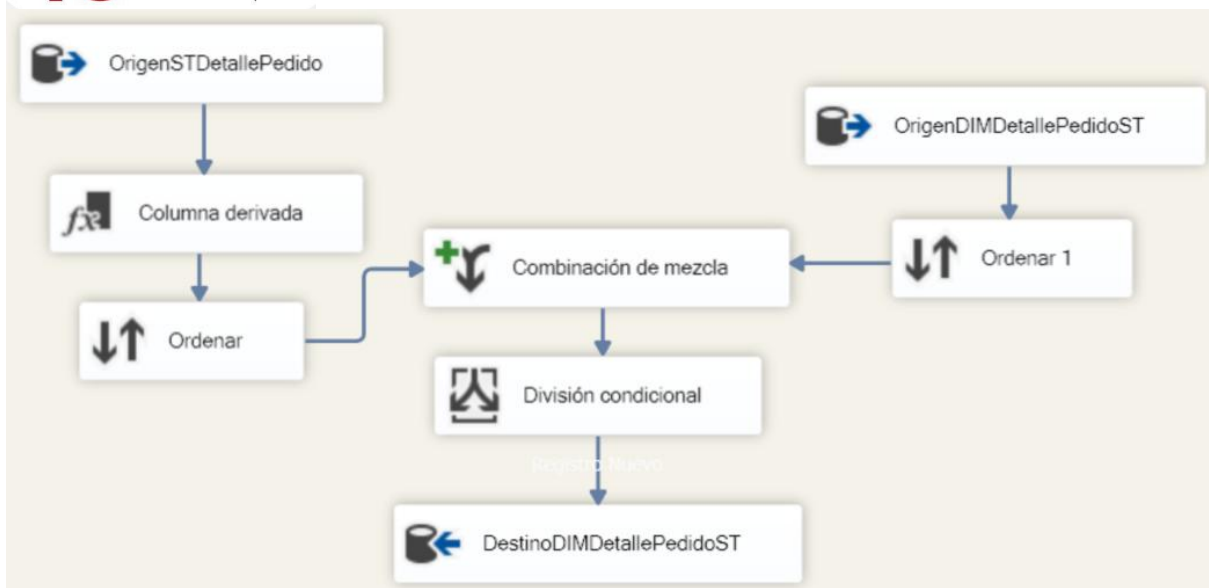
Creación de la dimensión DIMpedido: la tabla de destino resultado de la transformación se crea con el siguiente comando de SQL.

```
CREATE TABLE "DIMpedidoST" ("ID_pedido" int primary key, "fecha_esperada" date,
"fecha_pedido" date, "fecha_entrega" date, "estado" nvarchar(15), "comentarios"
nvarchar(max), "ID_cliente" int)
```

Tabla DIMdetallePedido:

El origen será la tabla obtenida en la etapa de extracción y se transformará su estructura con el siguiente diagrama de flujo.





Se agregan nuevas columnas con la opción columna derivada como se muestra en la siguiente imagen.

Derived Column Name	Derived Column	Expression
TotalPedido	<add as new column>	cantidad * precio_unidad

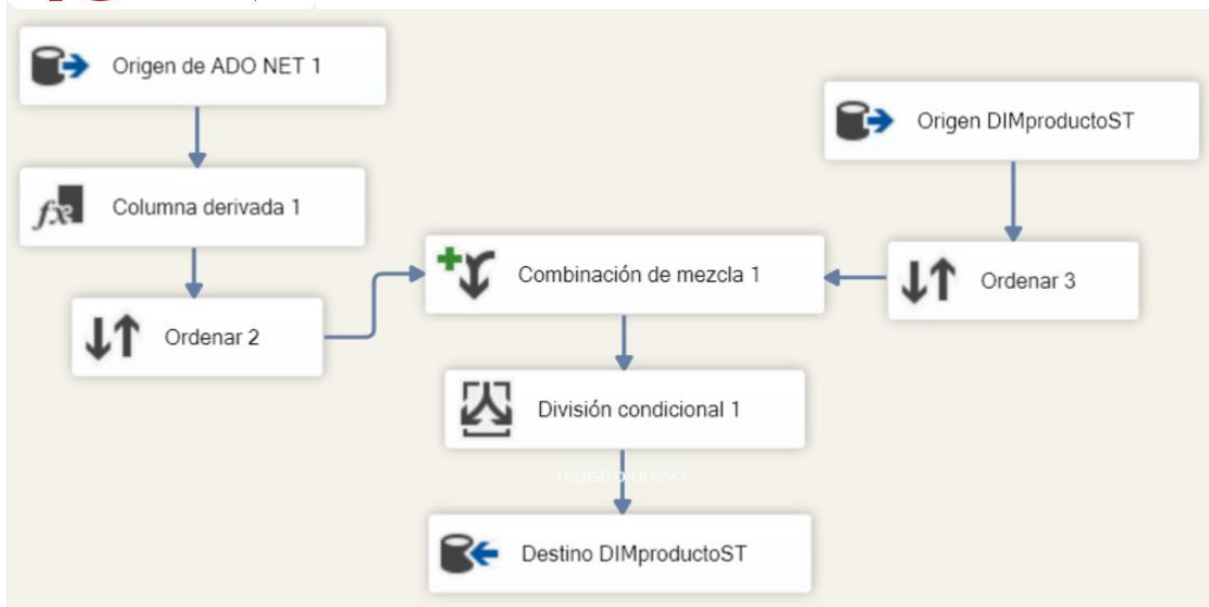
Seguidamente se ordena y se pasa por una combinación que contiene como origen la tabla de destino al final del diagrama de flujo, en el siguiente paso con la división condicional se evalúa si los datos que van a ser ingresados ya están en la tabla de destino, de lo contrario se agregarán solamente los nuevos registros.

Creación de la dimensión DIMdetallePedido: la tabla de destino resultado de la transformación se crea con el siguiente comando de SQL.

```
CREATE TABLE "DIMDetallePedidoST" ("ID_detalle_pedido" int primary key, "ID_pedido" int, "ID_producto" int, "cantidad" int, "precio_unidad" numeric(15,2), "numero_linea" smallint, "TotalPedido" numeric(26,2))
```

Tabla DIMproducto:

El origen será la tabla obtenida en la etapa de extracción y se transformará su estructura con el siguiente diagrama de flujo.



Se reemplazan columnas existentes con la opción columna derivada como se muestra en la siguiente imagen.

Derived Column Name	Derived Column	Expression
dimensiones	Reemplazar 'dimensiones'	dimensiones == "" ? "N/A" : dimensiones
proveedor	Reemplazar 'proveedor'	UPPER(proveedor)

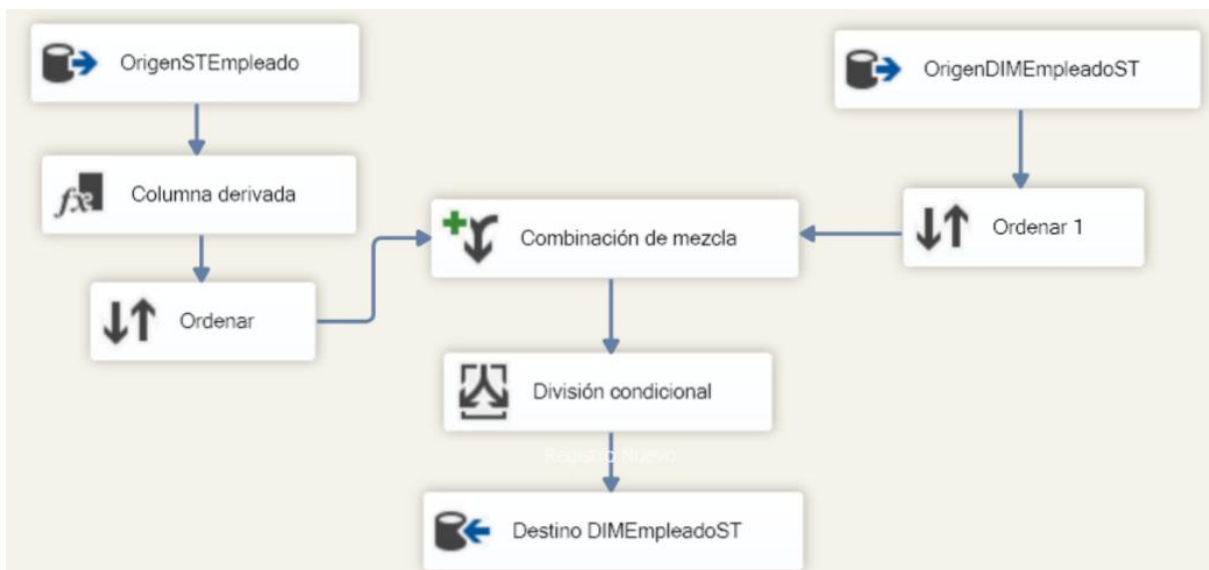
Seguidamente se ordena y se pasa por una combinación que contiene como origen la tabla de destino al final del diagrama de flujo, en el siguiente paso con la división condicional se evalúa si los datos que van a ser ingresados ya están en la tabla de destino, de lo contrario se agregarán solamente los nuevos registros.

Creación de la dimensión DIMproducto: la tabla de destino resultado de la transformación se crea con el siguiente comando de SQL.

```
CREATE TABLE "DIMproductoST" ("ID_producto" int primary key, "CodigoProducto"
nvarchar(15), "nombre" nvarchar(70), "Categoria" int, "dimensiones" nvarchar(25),
"proveedor" nvarchar(50), "precio_venta" numeric(15,2))
```

Tabla DIMEmpleado:

El origen será la tabla obtenida en la etapa de extracción y se transformará su estructura con el siguiente diagrama de flujo.



Se reemplazan columnas existentes con la opción columna derivada como se muestra en la siguiente imagen.

Derived Column Name	Derived Column	Expression
ID_jefe	Reemplazar 'ID_jefe'	REPLACENULL(ID_jefe,0)
puesto	Reemplazar 'puesto'	UPPER(puesto)

Seguidamente se ordena y se pasa por una combinación que contiene como origen la tabla de destino al final del diagrama de flujo, en el siguiente paso con la división condicional se evalúa si los datos que van a ser ingresados ya están en la tabla de destino, de lo contrario se agregarán solamente los nuevos registros.

Creación de la dimensión DIMEpleado: la tabla de destino resultado de la transformación se crea con el siguiente comando de SQL.

```
CREATE TABLE "DIMEpleadoST" ("ID_empleado" int primary key, "nombre"
nvarchar(50), "apellido1" nvarchar(50), "extension" nvarchar(10), "email" nvarchar(100),
"ID_oficina" int, "ID_jefe" int, "puesto" nvarchar(50))
```

### Tabla DIMCliente:

El origen será la tabla obtenida en la etapa de extracción y se transformará su estructura con el siguiente diagrama de flujo.



Se reemplazan columnas existentes con la opción columna derivada como se muestra en la siguiente imagen.

Derived Column Name	Derived Column	Expression
region	Reemplazar 'region'	REPLACENULL( [region] , "N/A")
ciudad	Reemplazar 'ciudad'	UPPER( [ciudad] )
pais	Reemplazar 'pais'	UPPER( [pais] )

Seguidamente se ordena y se pasa por una combinación que contiene como origen la tabla de destino al final del diagrama de flujo, en el siguiente paso con la división condicional se evalúa si los datos que van a ser ingresados ya están en la tabla de destino, de lo contrario se agregarán solamente los nuevos registros.

Creación de la dimensión DIMCliente: la tabla de destino resultado de la transformación se crea con el siguiente comando de SQL.

```

CREATE TABLE "DIMClienteST" ("ID_cliente" int primary key, "nombre_cliente"
nvarchar(50), "nombre_contacto" nvarchar(30), "apellido_contacto" nvarchar(30),
"telefono" nvarchar(15), "linea_direccion1" nvarchar(50), "ciudad" nvarchar(50), "region"
nvarchar(50), "pais" nvarchar(50), "ID_empleado_rep_ventas" int, "limite_credito"
numeric(15,2))
  
```

Tabla Fac\_pedido:

El origen será la unión de la tabla detalle\_pedido con la tabla pedido con el siguiente script de SQL:

SELECT

pe.[ID\_pedido], fes.IDDimTiempo as fecha\_esperada, fp.IDDimTiempo as fecha\_pedido,  
fen.IDDimTiempo as fecha\_entrega, pe.[estado], pe.[comentarios], pe.[ID\_cliente],

dp.[ID\_detalle\_pedido], pr.[ID\_producto], dp.[cantidad], dp.[precio\_unidad],  
dp.[numero\_linea], dp.[TotalPedido],

em.[ID\_empleado]

FROM [dbo].[DIMpedidoST] Pe inner join [dbo].[DIMDetallePedidoST] DP on  
PE.ID\_pedido=DP.ID\_pedido

inner join [dbo].[DIMtiempoST] FP on pe.fecha\_pedido=fp.fecha\_pedido

inner join [dbo].[DIMtiempoST] FES on pe.fecha\_esperada=fes.fecha\_pedido

inner join [dbo].[DIMtiempoST] FEN on pe.fecha\_esperada=FEN.fecha\_pedido

inner join [dbo].[DIMproductoST] Pr on pr.[ID\_producto]=DP.[ID\_producto]

inner join [dbo].[DIMClienteST] cl on pe.ID\_cliente=cl.ID\_cliente

inner join [dbo].[DIMEpleadoST] em on  
cl.[ID\_empleado\_rep\_ventas]=em.[ID\_empleado]

order by pe.ID\_pedido

El siguiente diagrama indica el flujo de los datos.





Seguidamente se ordena y se pasa por una combinación que contiene como origen la tabla de destino al final del diagrama de flujo, en el siguiente paso con la división condicional se evalúa si los datos que van a ser ingresados ya están en la tabla de destino, de lo contrario se agregarán solamente los nuevos registros.

Creación de la tabla de hechos Fac\_pedido: la tabla de destino resultado de la transformación se crea con el siguiente comando de SQL.

```
CREATE TABLE "Fac_pedidoST" ("ID_pedido" int, "fecha_esperada" int, "fecha_pedido" int, "fecha_entrega" int, "estado" nvarchar(15), "comentarios" nvarchar(max), "ID_cliente" int, "ID_detalle_pedido" int primary key, "ID_producto" int, "cantidad" int, "precio_unidad" numeric(15,2), "numero_linea" smallint, "TotalPedido" numeric(26,2), "ID_empleado" int,)
```

De esta manera ya se tienen configuradas en la base de datos de Staging las tablas necesarias para ser cargadas a la base de datos de Data Mart.

## **Etapas de Carga**

Con el siguiente comando en SQL se crea la base de datos de Data Mart:

```
CREATE DATABASE [DMJardineriaBD2Gloria]
```

El proceso parte desde la base de datos de Staging donde se toma cada tabla DIM creada en la etapa de transformación y serán cargadas a la base de datos del data mart. En la aplicación de visual studio se crean flujos de datos para cargar cada una de las tablas como se muestra en la siguiente imagen.



A continuación, se expone el proceso llevado a cabo para cada una de las tablas. Se agrega su primary key al momento de crear la tabla en la base de datos Data Mart.

Para cada operación de carga el origen será la tabla obtenida en la etapa de transformación, solamente se ordenarán sus campos, se pasa por una combinación que contiene como origen la tabla de destino al final del diagrama de flujo correspondiente, la cual estará ubicada en la base de datos de Data Mart, en este paso se evalúa si los datos que van a ser ingresados ya están en la tabla de destino, de lo contrario se agregarán solamente los nuevos registros para finalmente ser cargados a la base de datos de Data Mart.

Carga de tabla DIMEmpleado:

Diagrama de flujo.



Creación de la dimensión DIMEmpleado en la base de datos Data Mart: la tabla de destino resultado de la carga se crea con el siguiente comando de SQL.

```
CREATE TABLE "DIMEmpleado" ("ID_empleado" int primary key, "nombre" nvarchar(50),
"apellido1" nvarchar(50), "extension" nvarchar(10), "email" nvarchar(100), "ID_oficina" int,
"ID_jefe" int, "puesto" nvarchar(50))
```

Carga de tabla DIMCliente:

Diagrama de flujo.



Creación de la dimensión DIMCliente en la base de datos Data Mart: la tabla de destino resultado de la carga se crea con el siguiente comando de SQL.

```
CREATE TABLE "DIMCliente" ("ID_cliente" int primary key, "nombre_cliente"
nvarchar(50), "nombre_contacto" nvarchar(30), "apellido_contacto" nvarchar(30),
"telefono" nvarchar(15), "linea_direccion1" nvarchar(50), "ciudad" nvarchar(50), "region"
nvarchar(50), "pais" nvarchar(50), "ID_empleado_rep_ventas" int, "limite_credito"
numeric(15,2))
```

Carga de tabla DIMtiempo:



## Diagrama de flujo.



Creación de la dimensión DIMtiempo en la base de datos Data Mart: la tabla de destino resultado de la carga se crea con el siguiente comando de SQL.

```
CREATE TABLE "DIMtiempo" (IDDimTiempo int identity(1,1) primary key, "fecha_pedido" date, "Dia" int, "Mes" int, "Anio" int, "NumeroSemana" int, "Trimestre" int, "DiaAnio" int, "DiaSemana" int)
```

Carga de tabla DIMproducto:

## Diagrama de flujo.

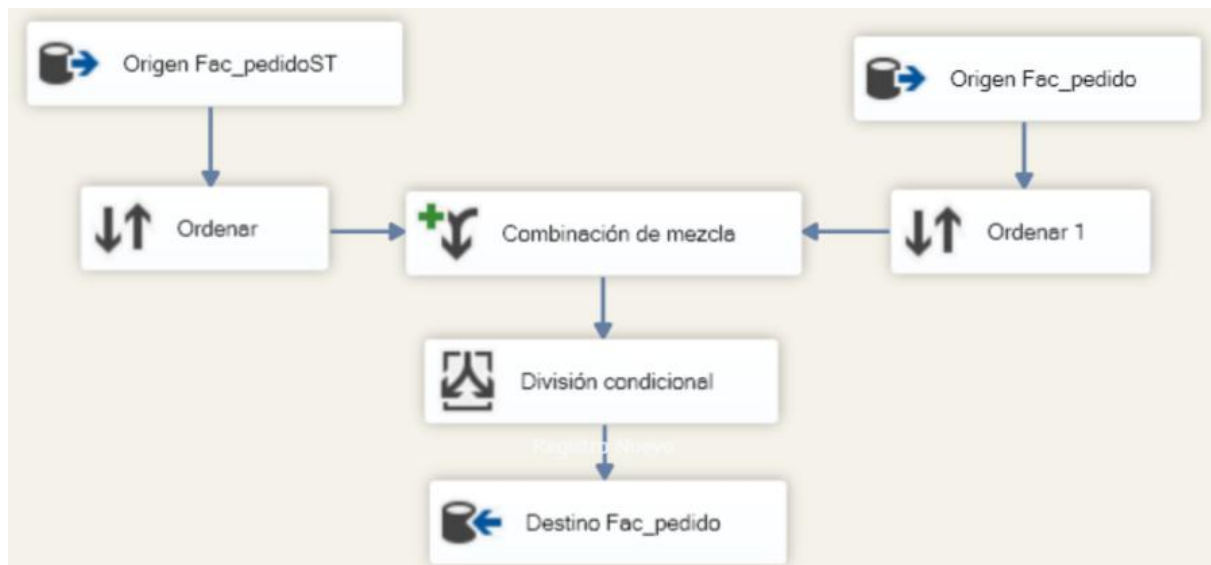


Creación de la dimensión DIMproducto en la base de datos Data Mart: la tabla de destino resultado de la carga se crea con el siguiente comando de SQL.

```
CREATE TABLE "DIMproducto" ("ID_producto" int primary key, "CodigoProducto"
nvarchar(15), "nombre" nvarchar(70), "Categoria" int, "dimensiones" nvarchar(25),
"proveedor" nvarchar(50), "precio_venta" numeric(15,2), "ID_productoDIM" int)
```

Carga de tabla Fac\_pedido:

Diagrama de flujo.



Creación de la tabla de hechos Fac\_pedido en la base de datos Data Mart: la tabla de destino resultado de la carga se crea con el siguiente comando de SQL.

```
CREATE TABLE "Fac_pedido" ("ID_pedido" int, "fecha_esperada" int, "fecha_pedido" int,
"fecha_entrega" int, "estado" nvarchar(15), "comentarios" nvarchar(max), "ID_cliente" int,
"ID_detalle_pedido" int primary key, "ID_producto" int, "cantidad" int, "precio_unidad"
numeric(15,2), "numero_linea" smallint, "TotalPedido" numeric(26,2), "ID_empleado" int)
```

