

EVIDENCIA DE APRENDIZAJE 2
CREACIÓN DE UNA BASE DE DATOS STAGING

DAVID ALEJANDRO AGUDELO MENESES
FERNEY DE JESUS ECHEVERRI ECHEVERRI
ESTEBAN URRUTIA BERMUDEZ

Asesor

ANTONIO JESÚS VALDERRAMA JARAMILLO

Institución Universitaria Digital de Antioquia

Facultad de Ingenierías y Ciencias

Ingeniería Software y Datos

2025

Resumen

La gestión de datos en los diferentes entornos de data Warehouse requiere de procesos intermedios que garanticen la calidad, consistencia y organización de la información antes de su análisis. En este trabajo se aborda la construcción de una base de datos Staging a partir de la base de datos Jardinería, con el fin de estructurar, depurar y trasladar datos relevantes que permitan soportar futuros procesos analíticos.

El desarrollo de esta actividad incluye un análisis de la información disponible, la creación de consultas SQL para migrar los registros y la validación de la integridad de los datos.

De igual manera se busca desarrollar competencias en el diseño, construcción y documentación de soluciones de bases de datos bajo un enfoque práctico y analítico.

Tabla de Contenido

Introducción.....	4
Justificación.....	5
Objetivos.....	6
Objetivo General.....	6
Objetivos Específicos.....	6
Planteamiento del problema.....	7
Análisis del problema.....	7
Propuesta de la solución:.....	9
Descripción	7
Diseño (Imagen)	9
Modelo estrella actualizado.....	10
Análisis realizado a los datos Jardinería, consultas y cómo éstos se trasladaron a la base de datos Staging.....	11
Validación de la extracción de datos de Jardinería a la Base de Datos Staging.....	13
Conclusiones.....	16
Referencias Bibliográficas.....	17

Introducción

Actualmente los datos se han convertido en un recurso fundamental para la toma de decisiones estratégicas, pero antes de que la información pueda ser utilizada en sistemas de inteligencia de negocios o procesos analíticos, debe pasar por ciertas etapas que garanticen su confiabilidad. Una de estas etapas es el Staging, la cual es un espacio intermedio donde los datos se depuran y organizan para posteriormente ser utilizados en almacenes de datos o sistemas de reporte.

El presente documento se desarrolla una base de datos Staging a partir del modelo relacional Jardinería, lo que permite aplicar conocimientos de diseño, normalización y construcción de consultas SQL, de tal forma se combina la teoría aprendida con un ejercicio práctico que nos preparen para desafíos reales en el ámbito de la gestión de datos.

Justificación

La creación de una base de datos Staging constituye un paso esencial en los procesos de integración y análisis de datos. En entornos de bases de datos, la información suele encontrarse dispersa, incompleta o poco depurada, lo que genera riesgos en la calidad de los resultados obtenidos en análisis posteriores.

En este caso la base de datos Jardinería, contiene información heterogénea que no se ajusta directamente a las necesidades de análisis. Contar con un Staging permite garantizar la coherencia y limpieza de los datos, facilitando su uso en futuros proyectos de inteligencia de negocios, análisis estadístico o sistemas de apoyo a la toma de decisiones.

Objetivos

Objetivo General

Construir una base de datos Staging basada en el modelo relacional de la base de datos Jardinería, aplicando consultas SQL que permitan la migración y validación de los datos.

Objetivos Específicos

Analizar los datos de la base de datos Jardinería para identificar la información relevante que debe trasladarse.

Implementar consultas SQL que permitan la carga de datos de manera organizada y precisa.

Validar la correcta migración de los registros asegurando su integridad y consistencia.

Planteamiento del problema.

La empresa Jardinería gestiona sus operaciones comerciales a través de una base de datos transaccional que almacena información sobre clientes, pedidos, pagos, productos, oficinas y empleados. Esta base de datos es funcional para el día a día de la compañía, pero no está optimizada para responder preguntas de carácter estratégico ni para realizar análisis empresariales complejos.

El reto surge cuando se requiere obtener indicadores de negocio, como ventas mensuales, clientes más rentables, productos más vendidos o niveles de cobranza. Al consultar directamente la base transaccional, las consultas resultan lentas, difíciles de mantener y pueden afectar el rendimiento operativo del sistema.

Por ello, es necesario diseñar una base de datos Staging (jardineria_stg) que actúe como una zona intermedia para la carga y transformación de datos. Esta permitirá organizar, limpiar y preparar la información proveniente de Jardinería para ser utilizada en procesos de análisis, reportes y construcción de modelos de inteligencia de negocio.

Análisis del problema.

Limitaciones de la base transaccional (Jardinería)

- A) Está orientada a registrar operaciones (altas, bajas, modificaciones) y no a responder consultas analíticas.
- B) La estructura normalizada hace más complejo realizar reportes que integren múltiples tablas.
- C) Consultas frecuentes sobre ventas, clientes o productos pueden degradar el rendimiento.

Necesidad de un entorno de Staging (jardineria_stg)

- A) Centralizar en tablas espejo los datos relevantes (clientes, productos, pedidos, pagos, empleados, oficinas).
- B) Agregar metadatos de control (batch_id, fecha de carga, fuente) para asegurar trazabilidad.
- C) Facilitar procesos ETL (Extract, Transform, Load) que limpien y consoliden información antes de llegar a un Data Warehouse.

Objetivos que resuelve el Staging

- Velocidad: mejorar tiempos de respuesta en las consultas analíticas.
- Calidad de datos: asegurar consistencia, detectar registros incompletos o inválidos.

- Seguridad: evitar impactar la base operativa al ejecutar reportes pesados.
- Escalabilidad: sentar la base para un modelo de Data Warehouse en estrella o copo de nieve.

Impacto esperado

- Directivos y analistas contarán con reportes confiables y oportunos.
- Se podrán generar indicadores clave de gestión (KPI), como ingresos mensuales, rentabilidad por categoría de producto o desempeño de oficinas y empleados.
- Se establece un ciclo de mejora continua en la toma de decisiones.

Propuesta de la solución

Se busca superar las limitaciones de la base de datos transaccional de Jardinería mediante la creación de un entorno Staging (jardineria_stg). Este entorno permitirá centralizar, transformar y depurar los datos antes de ser utilizados en análisis empresariales y en la posible construcción de un Data Warehouse.

La base de datos jardineria_stg se diseña como un conjunto de tablas espejo de la base de datos Jardinería, que contienen únicamente los datos relevantes para análisis. Estas tablas mantienen la estructura básica del origen, pero se complementan con columnas de auditoría:

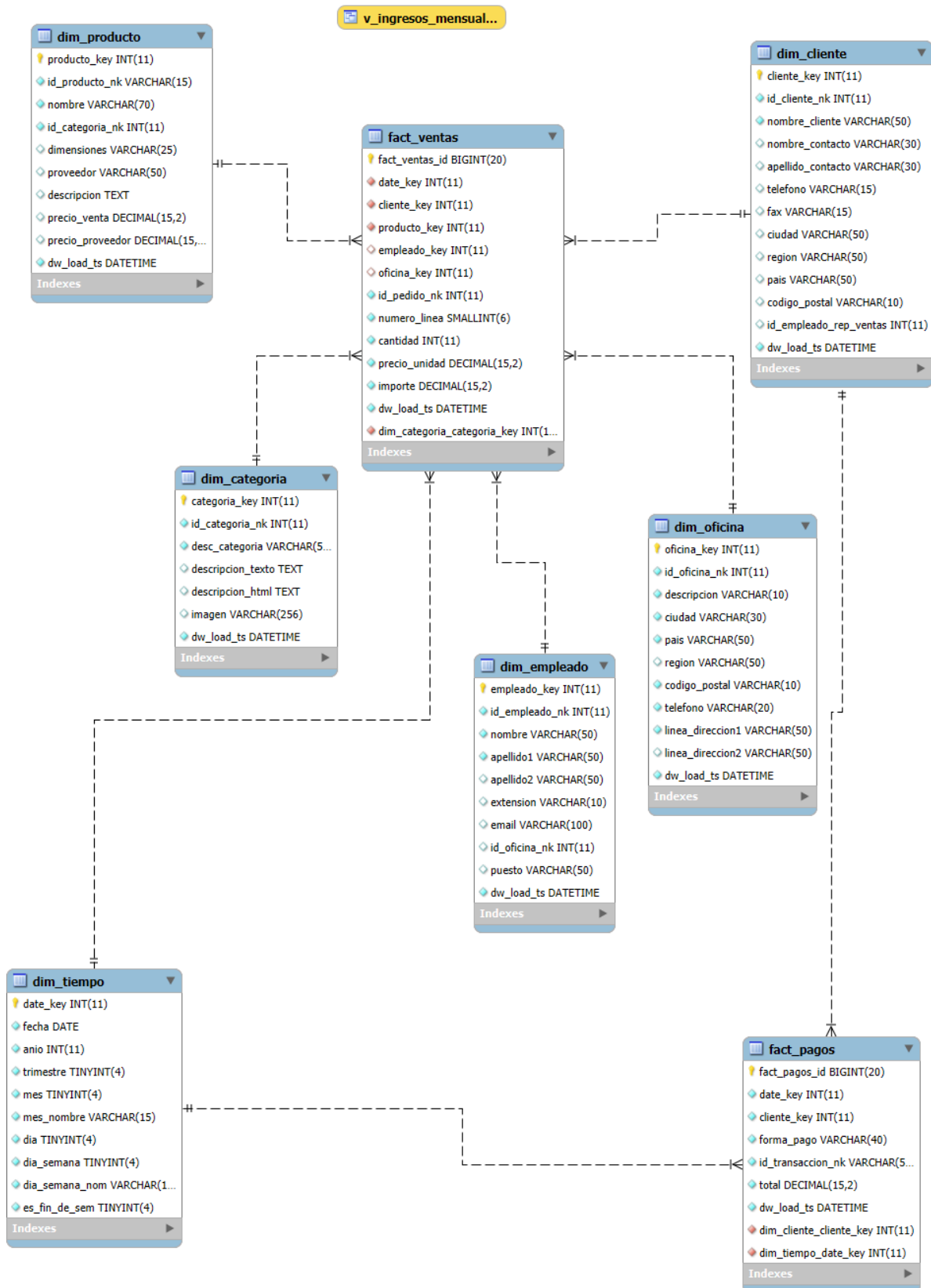
stg_batch_id: Identificador del lote de carga.

stg_load_ts: Fecha y hora en que se insertaron los datos.

stg_source: Origen de los datos (ejemplo: “Jardinería”).

Corrección entrega 1.

Modelo estrella actualizado



Análisis realizado a los datos Jardinería, consultas y cómo estos se trasladaron a la base de datos Staging.

En primer lugar, se realizó la creación del entorno de Staging mediante la instrucción `DROP DATABASE IF EXISTS` y la posterior construcción de un esquema nuevo (`jardineria_stg`). Este entorno funciona como una copia temporal de los datos del sistema de origen, pero enriquecido con metadatos de control que permiten dar trazabilidad a cada carga. Posteriormente, se implementó una tabla central de control (`stg_control_lote`) en la cual se registran los procesos de carga por lote incluyendo la fecha y hora de inicio y fin, el estado de ejecución (`RUNNING`, `OK`, `ERROR`), el número de filas cargadas y notas adicionales. Esta tabla permite auditar y supervisar los procesos de integración. Cada una de las tablas del sistema de origen (`oficina`, `empleado`, `categoria_producto`, `producto`, `cliente`, `pedido`, `detalle_pedido`, `pago`) fue replicada en Staging mediante la sentencia `CREATE TABLE ... LIKE`. Adicionalmente, a todas ellas se les agregaron campos de control como:

`stg_batch_id`: identifica el lote de carga al cual pertenece el registro.

`stg_load_ts`: marca temporal que indica cuándo se insertó el dato.

`stg_source`: registra la fuente de origen (en este caso, `jardineria`).

Para garantizar la consistencia, se aplicó una limpieza previa de las tablas con `TRUNCATE TABLE`, asegurando que cada carga fuera un full refresh y evitando duplicados o residuos de ejecuciones anteriores.

La fase de carga se realizó mediante sentencias `INSERT ... SELECT`, copiando los datos de las tablas de origen hacia sus equivalentes en Staging, enriquecidos con la metadata de lote y timestamp.

Una vez terminada la carga, se actualizaron los metadatos en stg_control_lote para marcar el lote como finalizado con éxito (OK).

Finalmente, se efectuó un análisis de calidad de datos y validaciones típicas, entre ellas:

Comparación de conteos entre tablas de origen y Staging para confirmar que los volúmenes coincidieran.

Verificación de rangos de fechas en pedidos y pagos, garantizando la coherencia temporal.

Revisión de integridad referencial lógica, por ejemplo, que todos los pedidos tuvieran clientes válidos y que todos los detalles de pedidos apuntaran a productos existentes.

Perfilado de datos críticos, como detectar productos con precios de venta nulos o negativos, y analizar la presencia o ausencia de comentarios en los pedidos.

Este proceso permitió trasladar los datos de manera controlada, auditable y confiable, preparando así la información para su posterior integración en el Data Warehouse bajo un esquema dimensional.

Validación de la extracción de datos de Jardinería a la Base de Datos Staging.

Imagen 1. Consulta de tabla temporal `jardineria_stg.stg_detalle_pedido` en la base de datos Staging.

The screenshot displays the SQL Server Enterprise Manager interface. On the left, the 'SCHEMAS' pane shows the 'jardineria_stg' database with various tables. The 'stg_detalle_pedido' table is selected. The 'Table: stg_detalle_pedido' information pane shows the following columns and data types:

Column	Data Type
ID_detalle_pedido	int AI PK
ID_pedido	int
ID_producto	int
cantidad	int
precio_unidad	decimal(15,2)
numero_linea	smallint
stg_batch_id	bigint
stg_load_ts	datetime
stg_source	varchar(50)

The 'Result Grid' shows the following data:

ID_detalle_pedido	ID_pedido	ID_producto	cantidad	precio_unidad	numero_linea	stg_batch_id	stg_load_ts	stg_source
1	1	87	10	70.00	3	1757804639	2025-09-13 18:03:59	jardineria
2	1	151	40	4.00	1	1757804639	2025-09-13 18:03:59	jardineria
3	1	165	25	4.00	2	1757804639	2025-09-13 18:03:59	jardineria
4	1	265	15	19.00	4	1757804639	2025-09-13 18:03:59	jardineria
5	2	57	3	29.00	6	1757804639	2025-09-13 18:03:59	jardineria
6	2	58	7	8.00	7	1757804639	2025-09-13 18:03:59	jardineria
7	2	164	50	4.00	3	1757804639	2025-09-13 18:03:59	jardineria
8	2	165	20	5.00	2	1757804639	2025-09-13 18:03:59	jardineria

The 'Output' pane shows the following actions:

#	Time	Action	Message	Duration / Fetch
264	18:04:25	SELECT d.ID_detalle_pedido FROM jardineria_stg.stg_detalle_pedido d LEF...	0 row(s) returned	0.000 sec / 0.000 sec
265	18:04:25	SELECT COUNT(*) AS productos_precio_venta_no_valido FROM jardineria_...	1 row(s) returned	0.000 sec / 0.000 sec
266	18:04:25	SELECT * FROM jardineria_stg.stg_detalle_pedido	8 row(s) returned	0.000 sec / 0.000 sec

Imagen 2. Consulta de tabla temporal `jardineria_stg.stg_pedido` en la base de datos Staging.

The screenshot displays the SQL Server Enterprise Manager interface. On the left, the 'SCHEMAS' pane shows the 'jardineria_stg' database with various tables. The 'stg_pedido' table is selected. The 'Table: stg_pedido' information pane shows the following columns and data types:

Column	Data Type
ID_pedido	int AI PK
fecha_pedido	date
fecha_esperada	date
fecha_entrega	date
estado	varchar(15)
comentarios	text
ID_cliente	int
stg_batch_id	bigint
stg_load_ts	datetime
stg_source	varchar(50)

The 'Result Grid' shows the following data:

ID_pedido	fecha_pedido	fecha_esperada	fecha_entrega	estado	comentarios	ID_cliente	stg_batch_id	stg_load_ts
1	2006-01-17	2006-01-19	2006-01-19	Entregado	Pagado a plazos	5	1757804639	2025-09-13 18:03:59
2	2007-10-23	2007-10-28	2007-10-26	Entregado	La entrega llegó antes de lo esperado	5	1757804639	2025-09-13 18:03:59
3	2008-06-20	2008-06-25	NULL	Rechazado	Limite de credito superado	5	1757804639	2025-09-13 18:03:59
4	2009-01-20	2009-01-26	NULL	Pendiente	NULL	5	1757804639	2025-09-13 18:03:59
5	2008-11-09	2008-11-14	2008-11-14	Entregado	El cliente paga la mitad con tarjeta y la otra mit...	1	1757804639	2025-09-13 18:03:59
6	2008-12-22	2008-12-27	2008-12-28	Entregado	El cliente comprueba la integridad del paquete, ...	1	1757804639	2025-09-13 18:03:59
7	2009-01-15	2009-01-20	NULL	Pendiente	El cliente llama para confirmar la fecha - Esoera...	3	1757804639	2025-09-13 18:03:59

The 'Output' pane shows the following actions:

#	Time	Action	Message	Duration / Fetch
267	21:16:08	SELECT * FROM jardineria_stg.stg_detalle_pedido LIMIT 0, 2000	314 row(s) returned	0.032 sec / 0.000 sec
268	21:27:07	SELECT * FROM jardineria_stg.stg_pedido LIMIT 0, 2000	115 row(s) returned	0.000 sec / 0.000 sec

Análisis de calidad de datos y validaciones típicas

Imagen 3. Consulta de SQL Verificación de rangos de fechas en pedidos y pagos, garantizando la coherencia temporal.

The screenshot shows a SQL query in a development tool. The query is as follows:

```
19 • SELECT 'pago', (SELECT COUNT(*) FROM jardineria.pago),
20 (SELECT COUNT(*) FROM jardineria_stg.stg_pago);
21
22 -- 4.2 Fechas min/max (sanidad)
23 • SELECT MIN(fecha_pedido) min_fec, MAX(fecha_pedido) max_fec
24 FROM jardineria_stg.stg_pedido;
25
```

The result grid shows the following data:

min_fec	max_fec
2006-01-17	2009-10-18

On the left, the table structure for `stg_pedido` is displayed:

Table: stg_pedido

Columns:

- `ID_pedido`: int AI PK
- `fecha_pedido`: date
- `fecha_esperada`: date
- `fecha_entrega`: date
- `estado`: varchar(15)
- `comentarios`: text

Imagen 4 . Consulta de SQL Consulta de pedidos por comentarios.

The screenshot shows a SQL query in a development tool. The query is as follows:

```
49
50 -- ¿Comentarios NULL vs no NULL en pedidos?
51 • SELECT estado,
52 SUM(comentarios IS NULL) AS sin_comentarios,
53 SUM(comentarios IS NOT NULL) AS con_comentarios
54 FROM jardineria_stg.stg_pedido
55 GROUP BY estado;
```

The result grid shows the following data:

estado	sin_comentarios	con_comentarios
Entregado	28	33
Rechazado	2	22
Pendiente	16	14

Imagen 5. Consulta de SQL Validación de ventas en valor negativo.

```
41 WHERE pr.ID_producto IS NULL
42 LIMIT 10;
43
44 -- 4.4 Perfil básico de campos críticos (ejemplos)
45 -- ¿Hay precios de venta en cero o negativos?
46 • SELECT COUNT(*) AS productos_precio_venta_no_valido
47 FROM jardineria_stg.stg_producto
```

Result Grid

productos_precio_venta_no_valido
0

Result 78 x

Output

Imagen 6. Consulta de SQL Selección de la tabla stg_categoria_producto.

```
1 • SELECT * FROM jardineria_stg.stg_categoria_producto;
2
3
4 -- 4.1 conteos origen vs staging
5 • SELECT 'oficina' tabla, (SELECT COUNT(*) FROM jardineria.oficina) origen,
6     (SELECT COUNT(*) FROM jardineria_stg.stg_oficina) staging;
7 • SELECT 'empleado', (SELECT COUNT(*) FROM jardineria.empleado),
```

Result Grid

	Id_Categoria	Desc_Categoria	descripcion_texto	descripcion_html	imagen	stg_batch_id	stg_load_ts	stg_source
▶	1	Herbaceas	Plantas para jardín decorativas	NULL	NULL	1757804639	2025-09-13 18:03:59	jardineria
	2	Herramientas	2 para todo tipo de acción	NULL	NULL	1757804639	2025-09-13 18:03:59	jardineria
	3	Aromaticas	Plantas aromáticas	NULL	NULL	1757804639	2025-09-13 18:03:59	jardineria
	4	Frutales	Árboles pequeños de producción frutal	NULL	NULL	1757804639	2025-09-13 18:03:59	jardineria
	5	Ornamentales	Plantas vistosas para la decoración del jardín	NULL	NULL	1757804639	2025-09-13 18:03:59	jardineria
•	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

stg_categoria_producto 82 x

Output

Conclusiones

El ejercicio permitió la construcción y análisis de una base de datos (base de datos Staging), basada en un base de datos previamente construida, la cual se encontraba basada en un modelo relacional (base de datos Jardinería).

La realización de esta actividad, más allá de tratarse de una simple migración, implicó identificar que en casos reales es necesario identificar la información relevante que debe trasladarse o migrar, teniendo presente que se pueden migrar todas la base de datos o solamente la parte de la información que es requerida para analizar, de acuerdo con el negocio.

Una vez migrada la base de datos , se hizo necesario la realización de consultas SQL, para evaluar que los datos fueron cargados de manera organizada y precisa. La realización de las consultas SQL permitieron validar la correcta migración, con lo cual se asegura la integridad y consistencia de la base de datos construida.

Finalmente se pudo establecer que contar con un Staging permite garantizar la coherencia y limpieza de los datos, facilitando su uso en futuros proyectos de inteligencia de negocios, análisis estadístico o sistemas de apoyo a la toma de decisiones, así como facilitar procesos ETL (Extract, Transform, Load) que limpien y consoliden información antes de que ésta haga parte de un Data Warehouse.

Anexos

BK de las dos bases de datos y el documento del script de las consultas para crear la base de datos Staging.

<https://drive.google.com/file/d/1fKYzkjltZO0T-ase14ARVec-WBs07a7X/view?usp=sharing>

Backup Jardinería.

<https://drive.google.com/file/d/1i7oIGjycTmFZor0AfV1vIToynCeFC4qS/view?usp=sharing>

Script de Base de datos Staging.

<https://drive.google.com/file/d/1K04VT0VN9IV9281qOSnABzSNyRGONfA8/view?usp=sharing>

Referencias Bibliográficas

IUDigital de Antioquia (2025). *Programación orientada a objetos II. Unidad 2.*

Star Data Engineering (15 sep 2025). *What is the staging area?* [Video]. Youtube.

<https://www.youtube.com/watch?v=4Q4FT594ZLA>