

Jenny Julizza Alava Bolaños
CC. 094160972-9

1. Introducción

En la actualidad, la Inteligencia de Negocios (BI) se ha convertido en un componente esencial para las organizaciones que desean tomar decisiones basadas en datos. Este proyecto tiene como propósito implementar un flujo ETL (Extracción, Transformación y Carga) completo, partiendo de fuentes de datos heterogéneas, procesándolas en Python y cargando el resultado final en un Data Warehouse en Amazon Redshift Serverless, para habilitar análisis estratégicos.

2. Objetivos

Objetivo General

Implementar un modelo de datos bajo arquitectura de **modelo estrella** que permita integrar diferentes fuentes, transformarlas y cargarlas en Redshift, habilitando consultas analíticas y visualizaciones.

Objetivos Específicos

- Extraer información de múltiples fuentes: transacciones, clientes y dataset Olist.
- Generar claves primarias secuenciales en cada tabla para garantizar unicidad.
- Limpiar y transformar los datos aplicando normalización y funciones lambda.
- Construir dimensiones y una tabla de hechos bajo modelo estrella.
- Exportar los resultados a archivos .csv y cargarlos a **Amazon S3**.
- Conectar Redshift a S3 para cargar el modelo de datos.
- Validar el proceso mediante consultas y gráficos exploratorios en Python.

3. Marco Teórico

3.1 ETL (Extract, Transform, Load)

Proceso utilizado en BI para integrar datos desde diversas fuentes:

- **Extracción:** acceso a bases SQL, APIs, CSV/Excel, logs, etc.
- **Transformación:** limpieza, normalización, generación de nuevas variables.

Jenny Julizza Alava Bolaños

CC. 094160972-9

- **Carga:** envío de datos a un Data Warehouse optimizado para consultas.

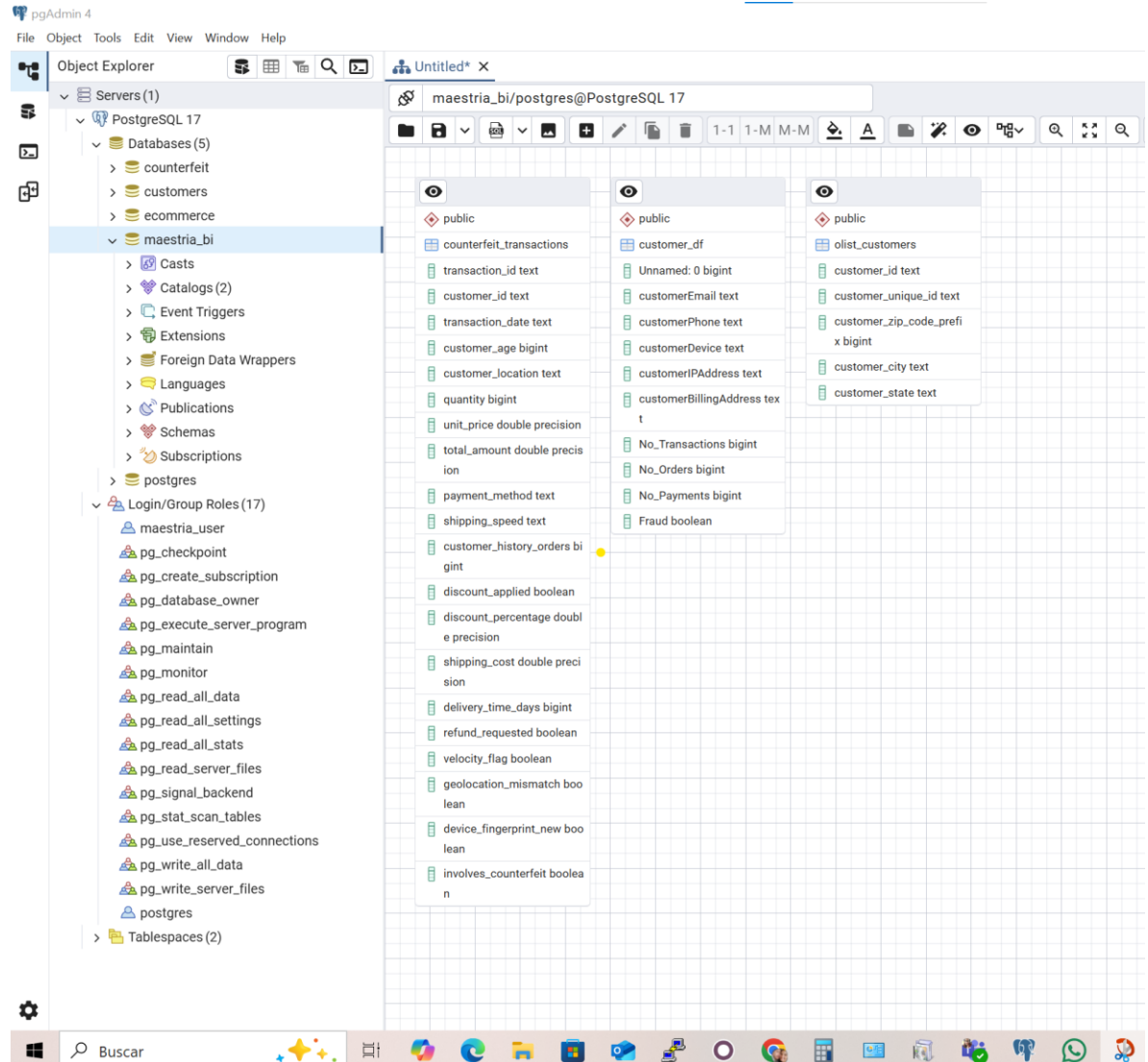


Figure 1 Nuestras bases de datos cargadas en Postgresql

3.2 Modelo Estrella

Esquema de modelado de datos donde:

- **Dimensiones:** tablas que describen entidades (cliente, tiempo, pago, envío, geografía).
- **Tabla de hechos:** centraliza las métricas y contiene claves foráneas a las dimensiones.

Jenny Julizza Alava Bolaños

CC. 094160972-9

Ventaja: optimiza consultas analíticas en entornos de BI como Power BI, Tableau o Redshift.

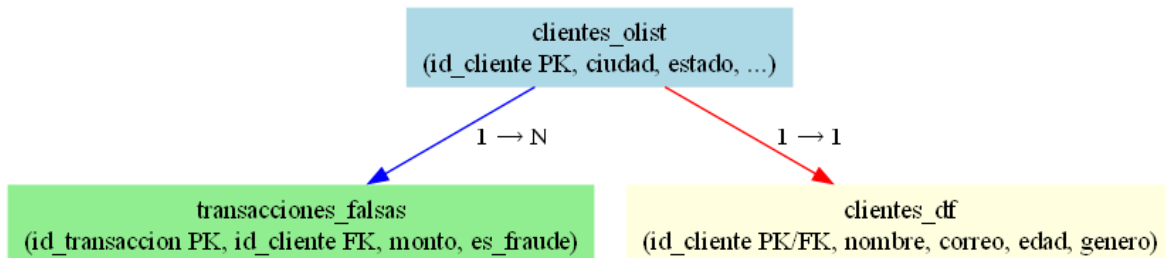


Figure 2 Modelo estrella en relacion a nuestras bases de datos

3.3 Amazon Redshift y S3

- **Amazon S3:** almacenamiento en la nube de objetos (archivos .csv).
- **Redshift Serverless:** Data Warehouse escalable para consultas SQL masivas.
- **COPY Command:** permite importar datos de S3 directamente a Redshift.

4. Desarrollo del Proyecto

4.1 Taller 1 – Extracción y exploración de datos

- Se cargaron tres datasets:
 - Counterfeit Transactions.
 - Customer DF.
 - Olist Customers.
- Se validaron columnas, duplicados y consistencia.
- Uso de `pandas.read_csv()` y exploración con `.head()` y `.info()`.

4.2 Taller 2 – Transformación y expansión de datos

- Se generaron claves únicas con `range()`.
- Se derivaron nuevas variables:
 - Año y mes desde `transaction_date`.
 - Longitud de email con funciones **lambda**.

Jenny Julizza Alava Bolaños

CC. 094160972-9

- Normalización de customer_city y customer_state.
- Se aplicó **merge** para consolidar información en un solo contenedor.

Ejemplo de lambda:

```
df_clientes["longitud_email"] = df_clientes["customeremail"].apply(lambda x: len(str(x)))
```

4.3 Taller 3 – Construcción del modelo estrella

- Se definieron dimensiones:
 - dim_clientes, dim_tiempo, dim_geografia, dim_pago, dim_envio.
- Se creó la tabla de hechos:
 - hechos_transacciones con métricas clave (quantity, total_amount, shipping_cost).
- Exportación a .csv para subirlos a S3.

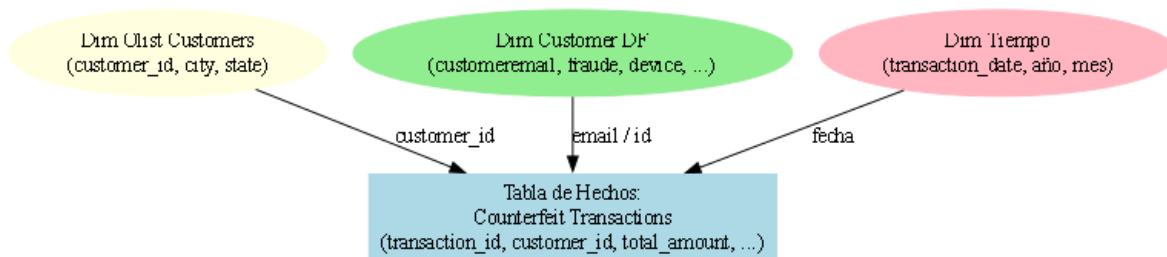


Figure 3 Nuestro modelo estrella completo

4.4 Taller 4 – Carga en Amazon Redshift

- Configuración de Redshift Serverless.
- Creación de bucket en S3.
- Subida de .csv a S3 desde Python con boto3.
- Uso de **COPY** para cargar datos a Redshift.

Jenny Julizza Alava Bolaños

CC. 094160972-9

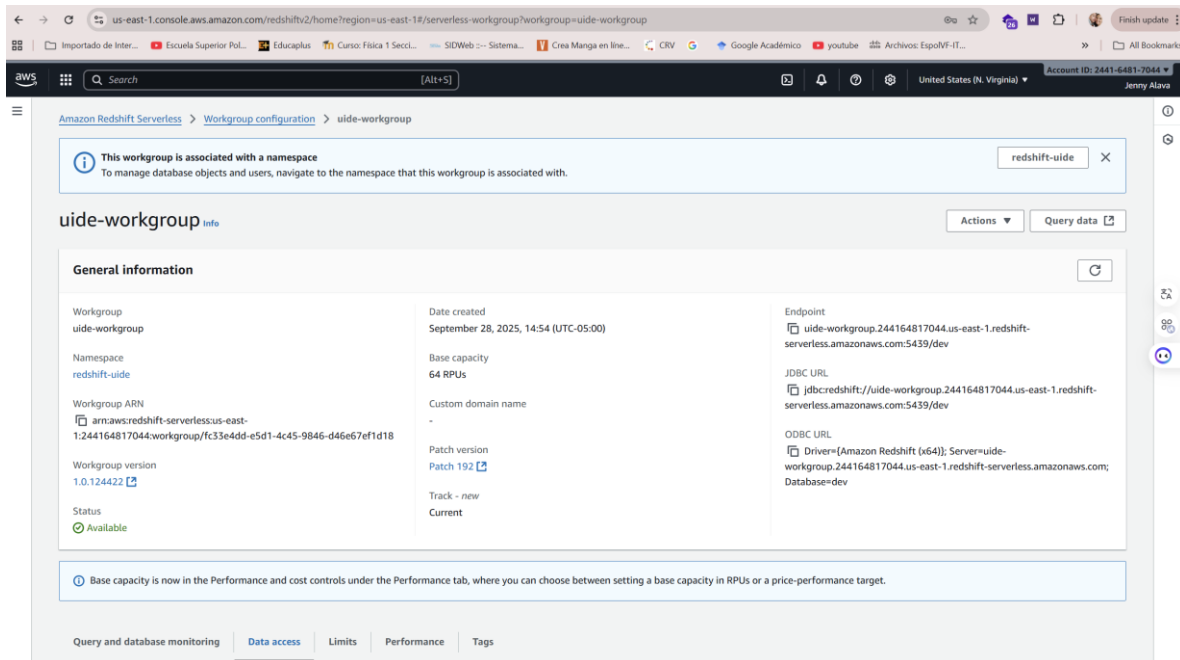


Figure 4 Workgroup de amazon redshift

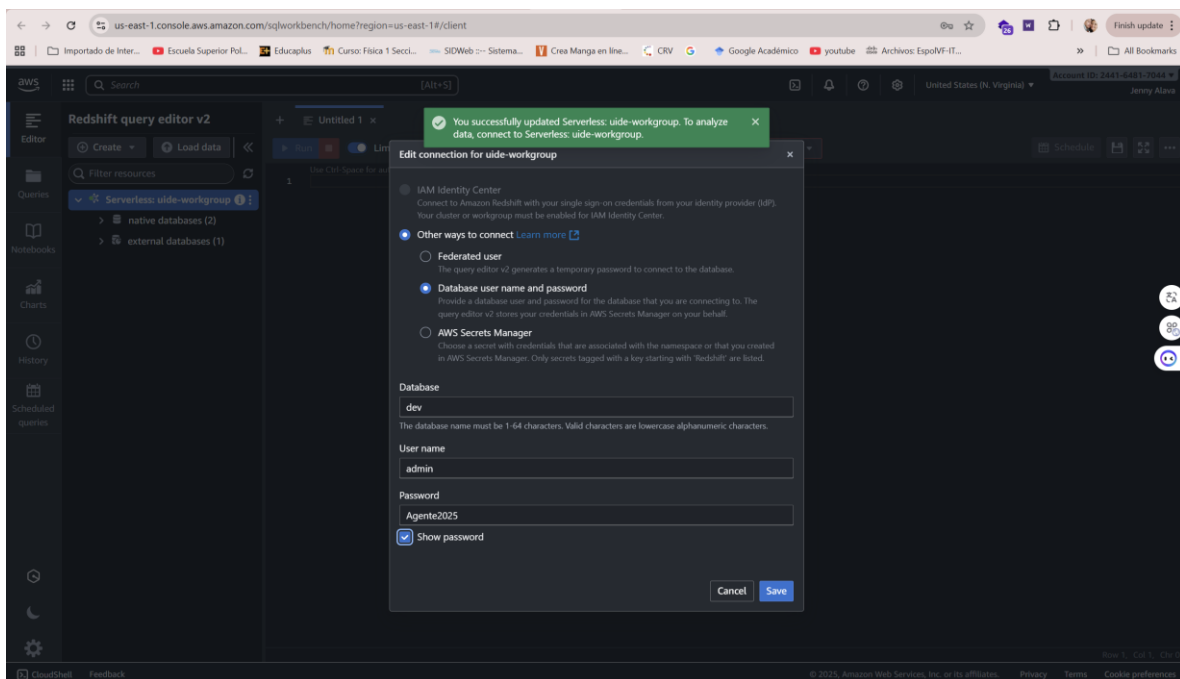


Figure 5 Credenciales de nuestra base de datos en amazon Redshift

Jenny Julizza Alava Bolaños
CC. 094160972-9

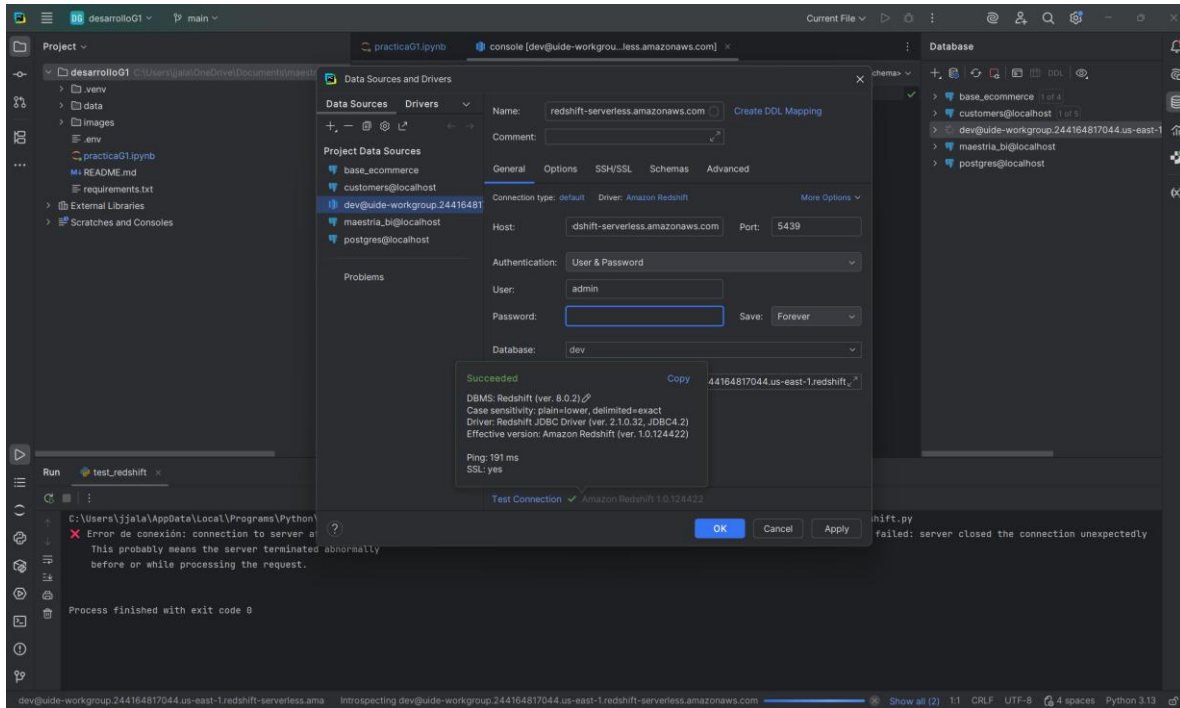


Figure 6 Conexión satisfactoria de la base de datos del Amazon Redshift de manera remota con Jetbrain

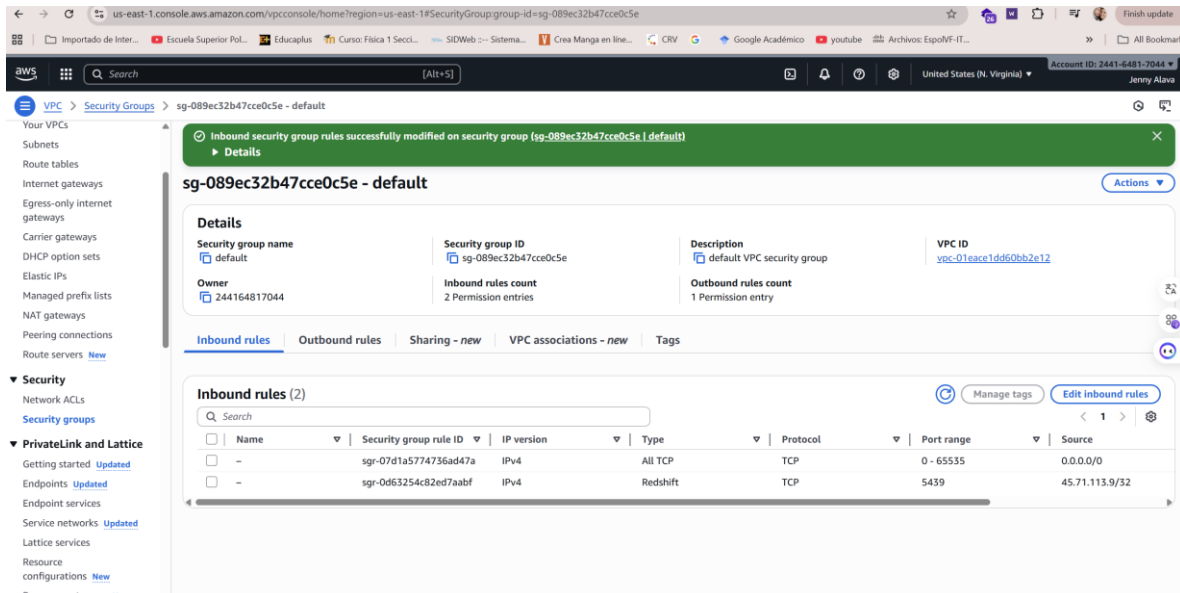


Figure 7 Reglas de seguridad entrantes y salientes para permitir la conexión remota

Jenny Julizza Alava Bolaños
CC. 094160972-9

5. Resultados

- Se obtuvo un **modelo estrella funcional** con 5 dimensiones y 1 tabla de hechos.
- Se logró consolidar fuentes heterogéneas en un solo modelo unificado.
- Se implementaron consultas exploratorias, por ejemplo:
 - Ventas por estado.
 - Distribución por método de pago.
 - Tendencias mensuales de transacciones.

Ejemplo de visualización:

```
import seaborn as sns
```

```
sns.barplot(data=hechos_transacciones, x="payment_method", y="total_amount")
```

Jenny Julizza Alava Bolaños

CC. 094160972-9

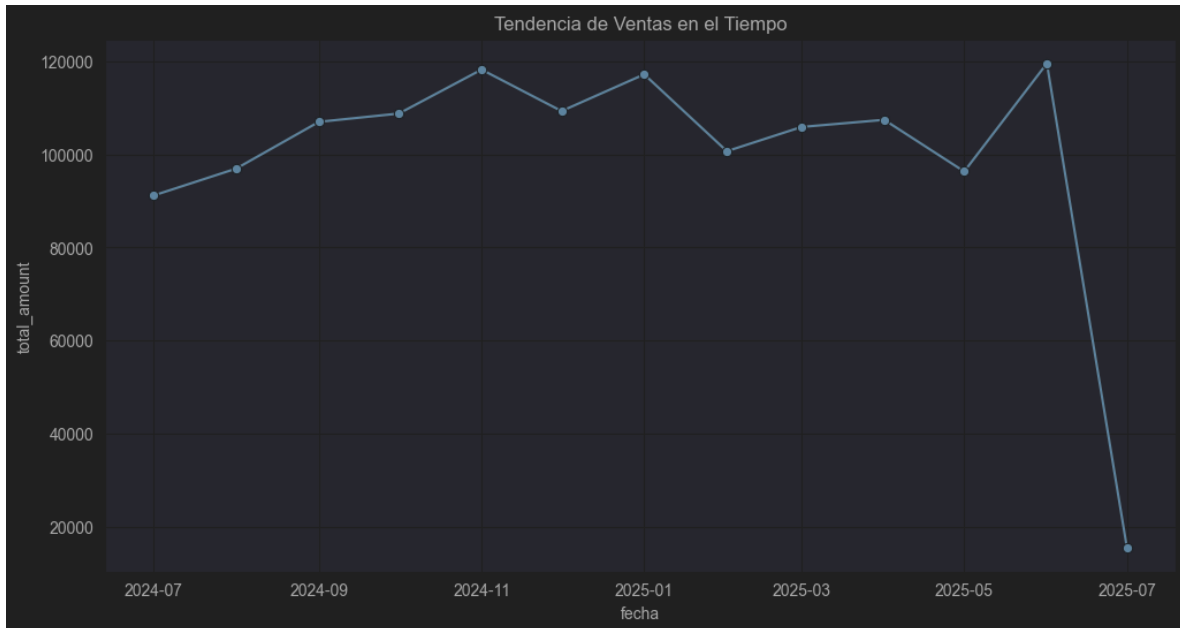


Figure 8 Grafico de la tabla de hechos_transacciones con referente al método de pago vs la cantidad transaccionada

6. Conclusiones

- El uso de identificadores secuenciales fue esencial para garantizar unicidad y evitar errores en los merges.
- La arquitectura de **modelo estrella** simplificó el análisis y habilita consultas rápidas en Redshift.
- Amazon Redshift y S3 permiten manejar grandes volúmenes de datos con escalabilidad.
- Python (pandas) fue fundamental para la limpieza y transformación previa a la carga.

7. Recomendaciones

- Automatizar el proceso ETL con Airflow o AWS Glue para producción.
- Integrar herramientas de visualización como Power BI o Tableau conectadas a Redshift.

Jenny Julizza Alava Bolaños
CC. 094160972-9

- Validar periódicamente la calidad de los datos en S3 antes de cargarlos al DWH.

8. Anexos

- Capturas de:
 - Creación de bucket en S3.
 - Configuración de Redshift Serverless.
 - Archivos exportados desde DataSpell.
 - Ejecución del COPY en Redshift.
- Fragmentos de código detallados en Python.

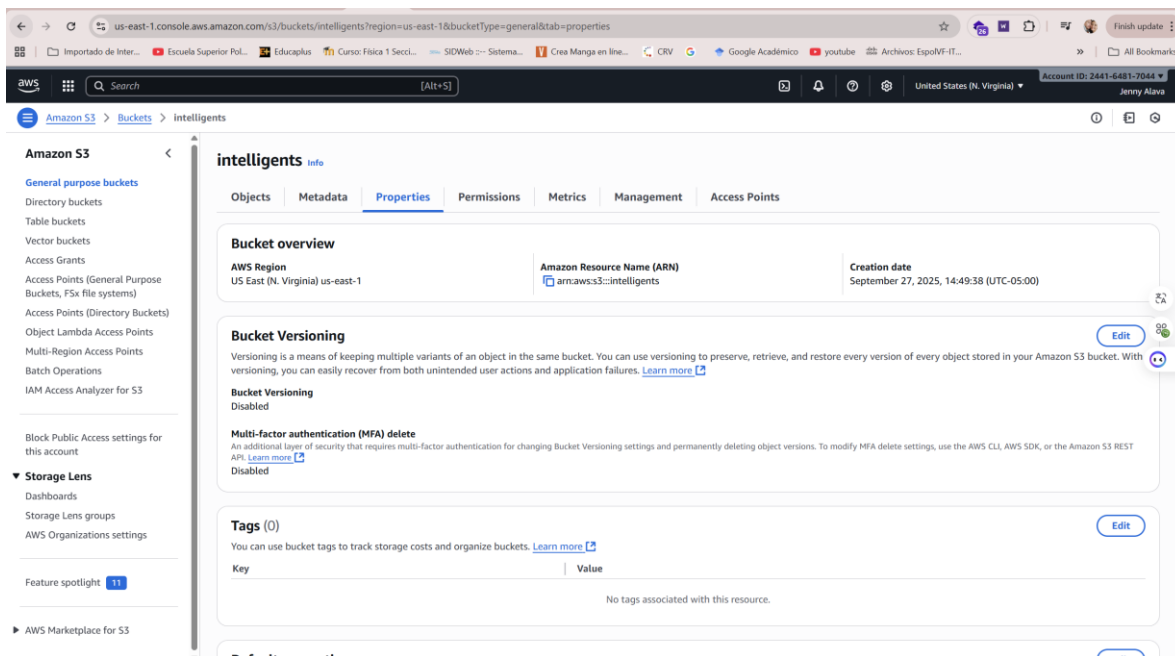


Figure 9 Creacion del bucket con Amazon S3

Jenny Julizza Alava Bolaños

CC. 094160972-9

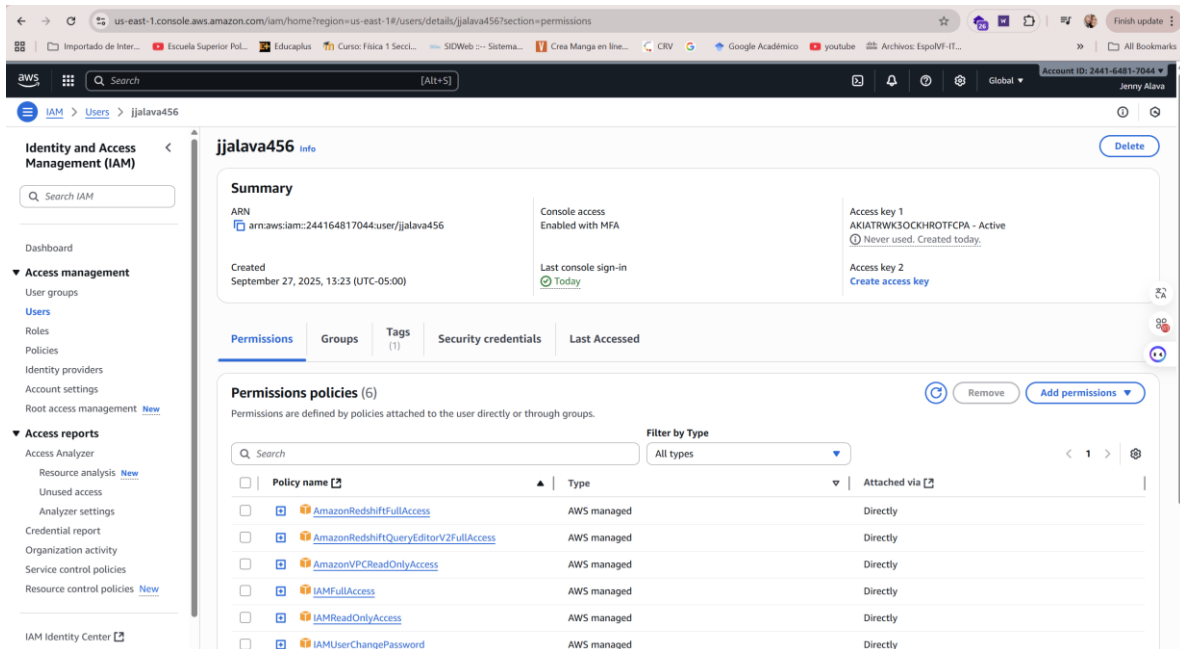


Figure 10 Configuración de mi IAM

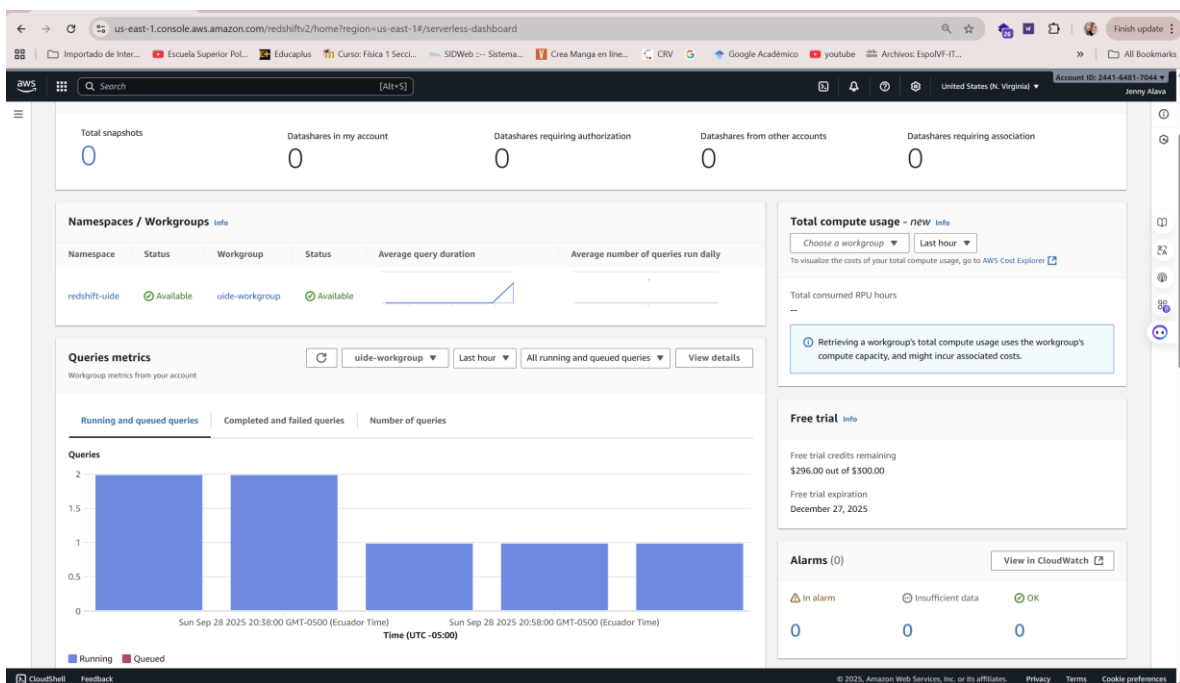
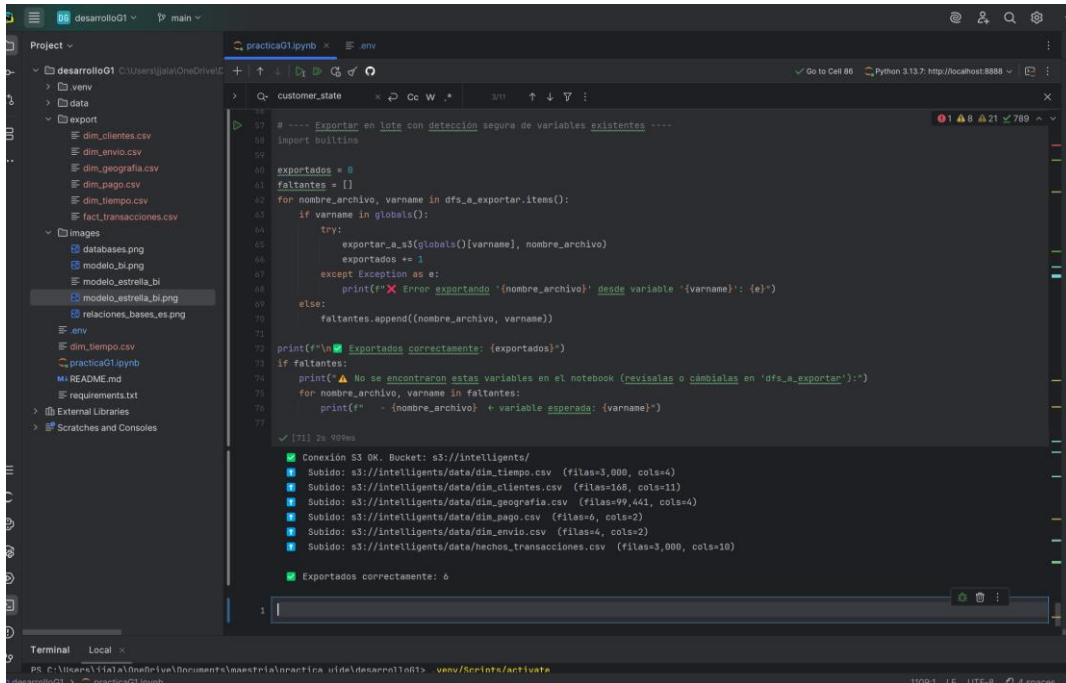


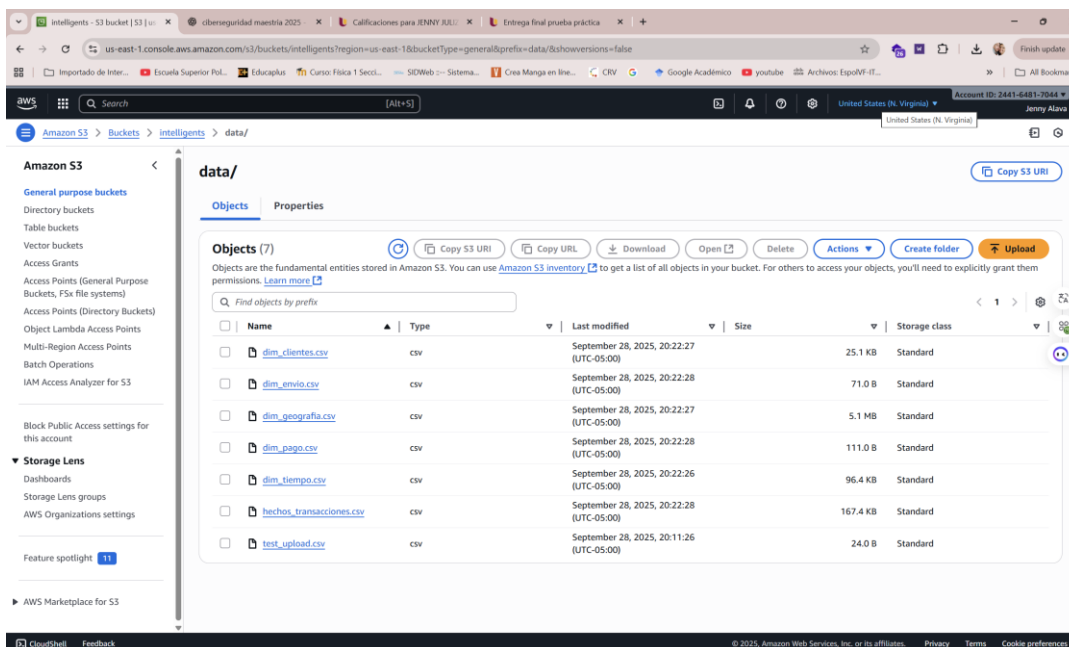
Figure 11 Panel del Amazon Redshift ya creado



```

57 # ---- Exportar en lote con detección segura de variables existentes ----
58 import builtins
59
60 exportados = 0
61 faltantes = []
62 for nombre_archivo, varname in dfs_a_exportar.items():
63     if varname in globals():
64         try:
65             exportar_a_s3(globals()[varname], nombre_archivo)
66             exportados += 1
67         except Exception as e:
68             print(f"Error exportando '{nombre_archivo}' desde variable '{varname}': {e}")
69     else:
70         faltantes.append((nombre_archivo, varname))
71
72 print(f"Exportados correctamente: {exportados}")
73 if faltantes:
74     print("⚠ No se encontraron estas variables en el notebook (revisalas o cámbialas en 'dfs_a_exportar'):")
75     for nombre_archivo, varname in faltantes:
76         print(f"  - {nombre_archivo} + variable esperada: {varname}")
77
78 ✓ [71] 2s 90ms
79
80 Conexión S3 OK. Bucket: s3://intelligents/
81 Subido: s3://intelligents/data/dim_tiempo.csv (filas=3,000, cols=4)
82 Subido: s3://intelligents/data/dim_clientes.csv (filas=168, cols=11)
83 Subido: s3://intelligents/data/dim_geografia.csv (filas=99,441, cols=4)
84 Subido: s3://intelligents/data/dim_pago.csv (filas=6, cols=2)
85 Subido: s3://intelligents/data/dim_envio.csv (filas=4, cols=2)
86 Subido: s3://intelligents/data/hechos_transacciones.csv (filas=3,000, cols=10)
87
88 Exportados correctamente: 6
  
```

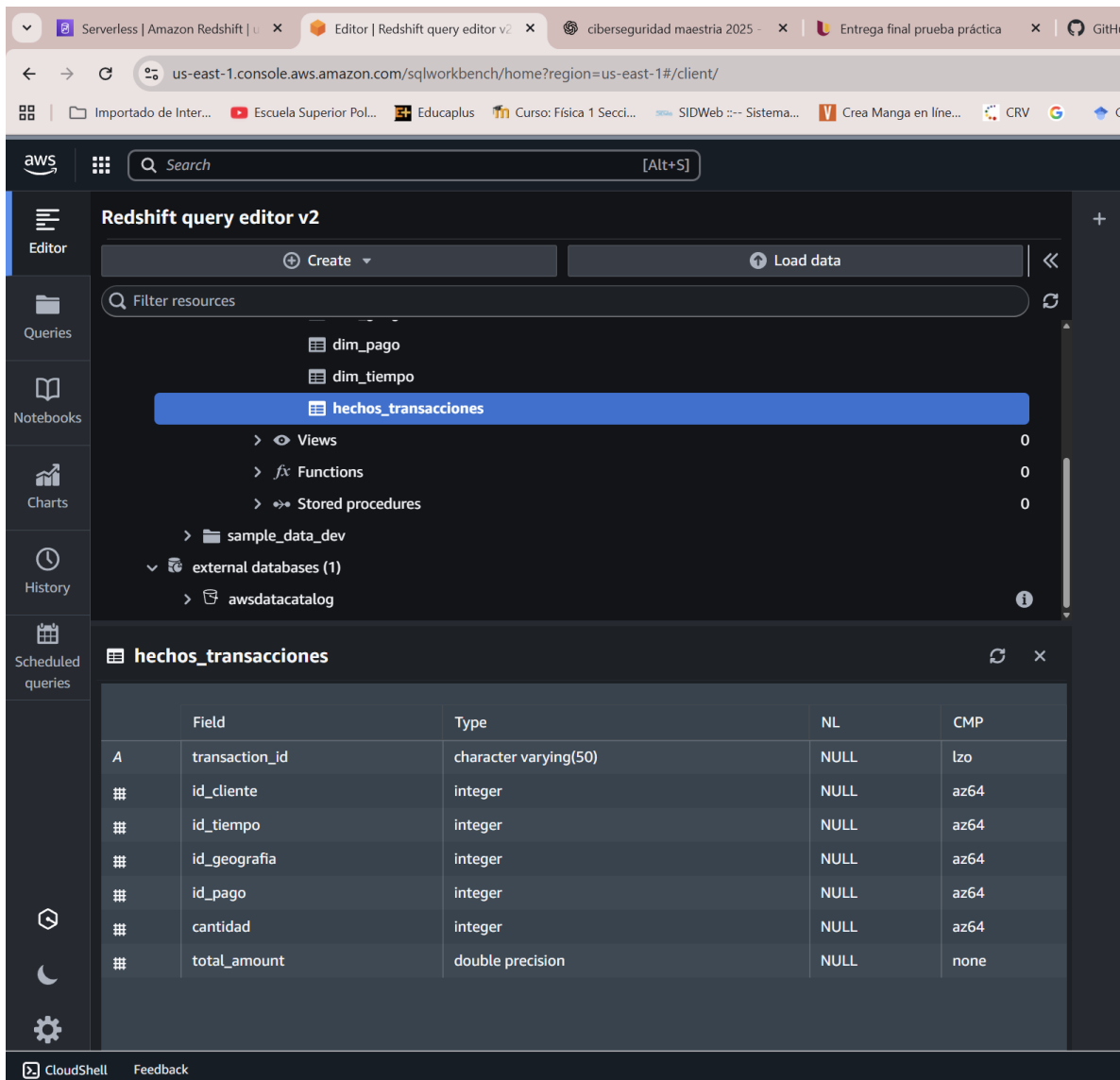
Figure 12 Archivos exportados al bucket S3



Name	Type	Last modified	Size	Storage class
dim_clientes.csv	csv	September 28, 2025, 20:22:27 (UTC-05:00)	25.1 KB	Standard
dim_envio.csv	csv	September 28, 2025, 20:22:28 (UTC-05:00)	71.0 B	Standard
dim_geografia.csv	csv	September 28, 2025, 20:22:27 (UTC-05:00)	5.1 MB	Standard
dim_pago.csv	csv	September 28, 2025, 20:22:28 (UTC-05:00)	111.0 B	Standard
dim_tiempo.csv	csv	September 28, 2025, 20:22:26 (UTC-05:00)	96.4 KB	Standard
hechos_transacciones.csv	csv	September 28, 2025, 20:22:28 (UTC-05:00)	167.4 KB	Standard
test_upload.csv	csv	September 28, 2025, 20:11:26 (UTC-05:00)	24.0 B	Standard

Figure 13 Archivos subidos exitosamente al S3

Jenny Julizza Alava Bolaños
CC. 094160972-9



The screenshot shows the Amazon Redshift console interface. The top navigation bar includes the AWS logo, a search bar, and a list of open tabs. The main content area is titled 'Redshift query editor v2' and features a sidebar with navigation options: Editor, Queries, Notebooks, Charts, History, and Scheduled queries. The 'Queries' section is active, displaying a list of resources. The 'hechos_transacciones' database is selected, and its schema details are shown in a table below.

	Field	Type	NL	CMP
A	transaction_id	character varying(50)	NULL	lzo
#	id_cliente	integer	NULL	az64
#	id_tiempo	integer	NULL	az64
#	id_geografia	integer	NULL	az64
#	id_pago	integer	NULL	az64
#	cantidad	integer	NULL	az64
#	total_amount	double precision	NULL	none

Figure 14 Visualización de la base de datos en Amazon Redshift