

# Caso IATA - Gestión de Datos

## Información del Proyecto

**Maestría:** Ciencia de Datos  
**Universidad:** Pontificia Universidad Javeriana  
**Materia:** Gestión de Datos  
**Estudiantes:** Edwin Silva Salas, Carlos Preciado Cárdenas, Cristian Restrepo Zapata  
**Fecha de inicio:** Noviembre 2025



## Descripción del proyecto

Este proyecto corresponde al **Caso IATA** de la materia Gestión de Datos, en el cual se trabaja con una base de datos relacional de un sistema de reservas y vuelos de aerolíneas. El objetivo principal es realizar operaciones de gestión de datos, análisis y potencialmente crear un **Data Mart** para facilitar consultas analíticas sobre el negocio de vuelos.

## Objetivos del Proyecto

- 1. **Configurar la base de datos transaccional IATA** con las tablas y datos proporcionados
- 2. **Realizar consultas y análisis** sobre los datos de vuelos, usuarios, aerolíneas e itinerarios
- 3. **Diseñar e implementar un Data Mart dimensional** (Star Schema) para análisis de negocio
- 4. **Generar reportes y métricas** clave del negocio de aerolíneas
- 5. **Visualizar la información** interactiva y dinamica para la toma de decisiones

## Modelo de Datos - Base Transaccional (OLTP)

La base de datos IATA contiene las siguientes entidades:

### Tablas Principales

Tabla	Descripción	Registros
<b>AEROLINEAS</b>	Aerolíneas disponibles (Avianca, Latam, Wingo)	3
<b>MODELOS</b>	Modelos de aviones (Airbus 320, Boeing 747)	2
<b>AVIONES</b>	Flota de aviones de cada aerolínea	10
<b>CIUDADES</b>	Ciudades origen/destino de los vuelos	12
<b>AEROPUERTOS</b>	Aeropuertos internacionales	12
<b>ITINERARIOS</b>	Rutas y horarios de vuelos	35
<b>USUARIOS</b>	Pasajeros registrados en el sistema	20
<b>VUELOS</b>	Reservas y compras de vuelos (Fact table)	170

# Proceso de Implementación (OLTP)

## Fase 1: Preparación del Entorno

- ☒ Configurar conexión a Oracle Database local del estudiante
- ☒ Crear usuario IATA en la base de datos local
- ☒ Otorgar privilegios básicos al usuario IATA
- ☒ Preparar scripts de eliminación y verificación

### Detalles de Configuración Inicial

Como primera actividad del caso, se procedió a crear el **usuario IATA** en la base de datos Oracle local del computador del estudiante. Esta decisión se tomó debido a que el script `IATA.sql` proporcionado como material de la actividad utiliza la nomenclatura `IATA.OBJECT` para todos los objetos de base de datos (tablas, constraints, índices, etc.).

### Pasos ejecutados:

1. **Conexión como usuario SYS:** Se estableció conexión a la base de datos local utilizando el usuario administrador SYS con rol SYSDBA
2. **Creación del usuario IATA:** Se ejecutó el comando `CREATE USER IATA` con contraseña segura
3. **Asignación de privilegios:** Se otorgaron los privilegios básicos necesarios:
  - `CONNECT`: Para conectarse a la base de datos
  - `RESOURCE`: Para crear objetos (tablas, vistas, secuencias, etc.)
  - `QUOTA UNLIMITED ON USERS`: Para utilizar espacio ilimitado en el tablespace USERS

```
-- Comandos ejecutados como SYS
CREATE USER IATA IDENTIFIED BY "passIATA123";
GRANT CONNECT, RESOURCE TO IATA;
ALTER USER IATA QUOTA UNLIMITED ON USERS;
```

Ver detalles completos del script de creación: `Creacion_IATA_OLTP.sql`

### Conexión al Esquema IATA

Una vez completada la configuración del usuario IATA, se estableció una nueva conexión a la base de datos utilizando las credenciales del usuario IATA. Esta conexión permite trabajar directamente sobre el esquema transaccional, garantizando que todos los objetos de base de datos se creen bajo el propietario correcto.

### Datos de conexión utilizados:

- **Username:** IATA
- **Password:** passIATA123
- **Role:** Default (ninguno, a diferencia de SYS que requiere SYSDBA)
- **Hostname:** localhost
- **Port:** 1521
- **Service Name:** FREEPDB1

Con la conexión establecida como usuario IATA, se procede a la siguiente fase: la carga de datos transaccionales mediante la ejecución del script proporcionado.

## Fase 2: Carga de Datos Fuente

- ☒ Ejecutar script **IATA.sql** para crear las tablas transaccionales
- ☒ Verificar la carga de datos (170 vuelos, 20 usuarios, 35 itinerarios)
- ☒ Validar integridad referencial y constraints

### Ejecución del Script IATA.sql

Posterior a la creación del usuario IATA, se procedió a ejecutar el script **IATA.sql** proporcionado como insumo de la actividad. Este script contiene:

- **8 tablas del modelo relacional:** AEROLINEAS, AEROPUERTOS, AVIONES, CIUDADES, ITINERARIOS, MODELOS, USUARIOS, VUELOS
- **Inserción de datos de prueba:** 170 registros de vuelos, 20 usuarios, 35 itinerarios, 12 ciudades, entre otros
- **Constraints y Foreign Keys:** Definición completa de integridad referencial
- **Índices únicos:** Para optimización de consultas

**Resultado de la ejecución:** El script se ejecutó **con cero errores**, creando exitosamente toda la estructura de base de datos y cargando los datos en el schema IATA

## Proceso de Implementación (OLAP)

### Fase 1: Preparación del Entorno OLAP

- ☒ Crear esquema IATA\_OLAP en la base de datos
- ☒ Otorgar privilegios básicos al esquema OLAP
- ☒ Configurar permisos de lectura sobre esquema OLTP (IATA)
- ☒ Validar conectividad entre esquemas

### Detalles de Creación del Esquema OLAP

Para implementar una arquitectura de datos robusta y siguiendo las mejores prácticas de separación entre sistemas transaccionales (OLTP) y analíticos (OLAP), se procedió a crear un **esquema independiente llamado IATA\_OLAP** que albergará el Data Mart dimensional.

### Justificación arquitectónica:

La separación de esquemas OLTP y OLAP permite:

- **Independencia operacional:** Las consultas analíticas no afectan el rendimiento transaccional
- **Seguridad por capas:** Control de acceso diferenciado entre operaciones y análisis
- **Optimización específica:** Cada esquema puede optimizarse para su propósito (escritura vs lectura)
- **Mantenibilidad:** Cambios en el modelo dimensional no impactan el sistema operacional

### Pasos ejecutados (ver script completo en **Creacion\_IATA\_OLAP.sql**):

1. **Conexión como usuario SYS:** Se estableció conexión con privilegios SYSDBA

2. Creación del esquema IATA\_OLAP:

```
CREATE USER IATA_OLAP IDENTIFIED BY "passIATAOLAP123";
```

3. Asignación de privilegios básicos:

```
GRANT CONNECT, RESOURCE TO IATA_OLAP;  
ALTER USER IATA_OLAP QUOTA UNLIMITED ON USERS;  
GRANT CREATE MATERIALIZED VIEW TO IATA_OLAP;
```

- **CONNECT**: Permite conectarse a la base de datos
- **RESOURCE**: Permite crear objetos dimensionales (tablas, vistas, procedimientos)
- **QUOTA UNLIMITED**: Espacio ilimitado en tablespace USERS
- **CREATE MATERIALIZED VIEW**: Para crear vistas materializadas (opcional para optimización)

4. Configuración de permisos de lectura sobre IATA (OLTP):

```
GRANT SELECT ON IATA.AEROLINEAS TO IATA_OLAP;  
GRANT SELECT ON IATA.AEROPUERTOS TO IATA_OLAP;  
GRANT SELECT ON IATA.AVIONES TO IATA_OLAP;  
GRANT SELECT ON IATA.CIUDADES TO IATA_OLAP;  
GRANT SELECT ON IATA.ITINERARIOS TO IATA_OLAP;  
GRANT SELECT ON IATA.MODELOS TO IATA_OLAP;  
GRANT SELECT ON IATA.USUARIOS TO IATA_OLAP;  
GRANT SELECT ON IATA.VUELOS TO IATA_OLAP;
```

Estos permisos son esenciales para que el proceso ETL pueda extraer datos del sistema transaccional.

Verificación de permisos otorgados:

```
SELECT  
  grantee,  
  owner,  
  table_name,  
  privilege  
FROM dba_tab_privs  
WHERE grantee = 'IATA_OLAP'  
ORDER BY table_name;
```

Resultado de la consulta:

GRANTEE	OWNER	TABLE_NAME	PRIVILEGE
IATA_OLAP	IATA	AEROLINEAS	SELECT

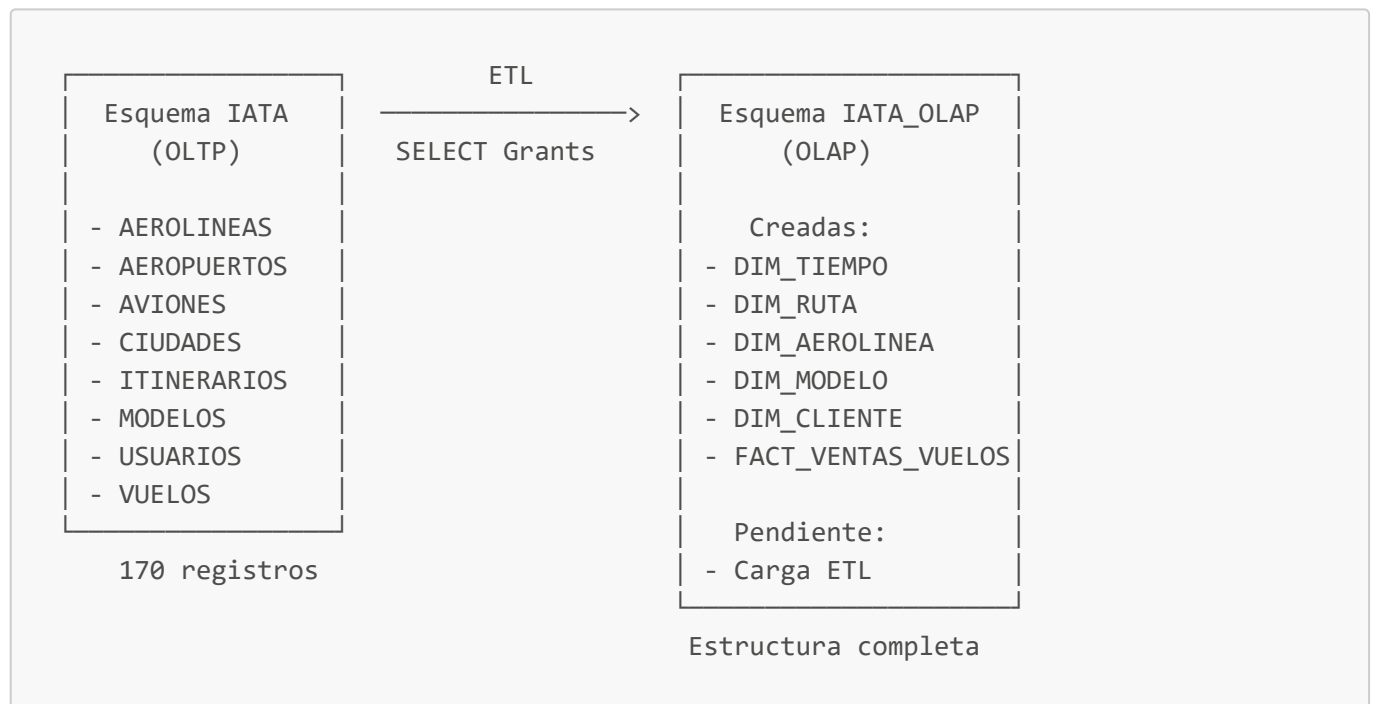
GRANTEE	OWNER	TABLE_NAME	PRIVILEGE
IATA_OLAP	IATA	AEROPUERTOS	SELECT
IATA_OLAP	IATA	AVIONES	SELECT
IATA_OLAP	IATA	CIUDADES	SELECT
IATA_OLAP	IATA	ITINERARIOS	SELECT
IATA_OLAP	IATA	MODELOS	SELECT
IATA_OLAP	IATA	USUARIOS	SELECT
IATA_OLAP	IATA	VUELOS	SELECT

Se confirmó que el usuario IATA\_OLAP tiene permisos SELECT sobre las 8 tablas del esquema IATA.

#### 5. Verificación de la configuración:

- Se validó la creación del usuario/esquema
- Se verificaron los privilegios otorgados
- Se comprobaron las cuotas asignadas en tablespace

#### Arquitectura resultante:



**Resultado:** El esquema IATA\_OLAP se creó exitosamente con todos los privilegios necesarios y permisos de lectura configurados sobre las tablas del esquema IATA (OLTP).

**Ver detalles completos del script de creación:** [Creacion\\_IATA\\_OLAP.sql](#)

#### Conexión al Esquema IATA\_OLAP

Una vez completada la configuración del esquema IATA\_OLAP (ver script anterior), se procedió a establecer una nueva conexión a la base de datos utilizando las credenciales del usuario IATA\_OLAP. Esta conexión

permite trabajar directamente sobre el esquema analítico, garantizando que todos los objetos dimensionales (tablas de dimensiones y hechos) se creen bajo el propietario correcto.

#### Datos de conexión utilizados:

- **Username:** IATA\_OLAP
- **Password:** passIATAOLAP123
- **Role:** Default (ninguno, a diferencia de SYS que requiere SYSDBA)
- **Hostname:** localhost
- **Port:** 1521
- **Service Name:** FREEPDB1

Con la conexión establecida como usuario IATA\_OLAP, se procede a la siguiente fase: el diseño e implementación del **Data Mart dimensional** utilizando el modelo de **Esquema en Estrella (Star Schema)**, que permitirá realizar análisis multidimensionales sobre las operaciones de vuelos de la aerolínea.

#### Fase 2: Diseño del Data Mart

- ☒ Diseñar modelo estrella (5 dimensiones + 1 tabla de hechos)
- ☒ Crear tablas de dimensiones y hechos

#### Diseño del Modelo Estrella (Star Schema)

Se ha diseñado e implementado un modelo dimensional tipo **Esquema en Estrella** para el Data Mart de análisis de ventas de vuelos. Este modelo se encuentra documentado y ejecutable en el script [Creacion\\_DATA\\_MART.sql](#).

#### Componentes del modelo:

##### Dimensiones (5):

- **DIM\_TIEMPO:** Análisis temporal (fecha, año, semestre, trimestre, mes, día)
- **DIM\_RUTA:** Análisis geográfico (ciudad origen, ciudad destino, aeropuertos)
- **DIM\_AEROLINEA:** Análisis por aerolínea
- **DIM\_MODELO:** Análisis por modelo de avión (Airbus 320, Boeing 747)
- **DIM\_CLIENTE:** Análisis demográfico de pasajeros (ciudad de residencia)

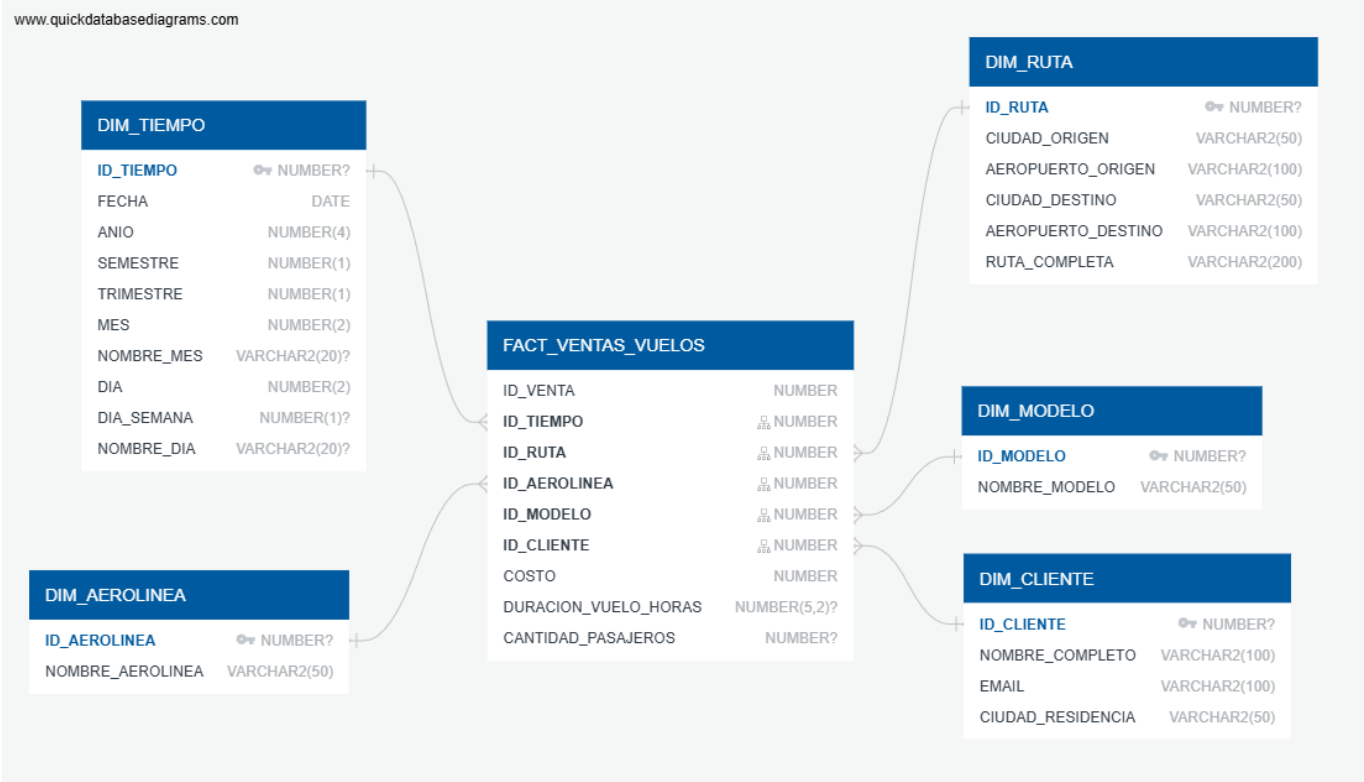
##### Tabla de Hechos (1):

- **FACT\_VENTAS\_VUELOS:** Métricas de negocio (costo, duración, cantidad de pasajeros) con claves foráneas a las 5 dimensiones

El diseño contempla las 4 preguntas analíticas clave del caso:

1. Aerolíneas con más vuelos a ciudades específicas por año
2. Recaudación por aerolínea por semestre
3. Modelos de avión con más vuelos por año
4. Ciudades cuyos habitantes viajaron más por año

#### Diagrama del Modelo Estrella:



Ejecución del Script de Creación

Una vez conectados como usuario **IATA\_OLAP**, se ejecutó el script **Creacion\_DATA\_MART.sql** que contiene todas las sentencias DDL para crear la estructura dimensional del Data Mart.

Objetos creados:

Tabla	Tipo	Estado	Propósito
DIM_TIEMPO	DIMENSION	Creada	Dimensión temporal para análisis por fecha, año, semestre, mes
DIM_RUTA	DIMENSION	Creada	Dimensión geográfica con origen-destino de vuelos
DIM_AEROLINEA	DIMENSION	Creada	Dimensión de aerolíneas (Avianca, Latam, Wingo)
DIM_MODELO	DIMENSION	Creada	Dimensión de modelos de avión (Airbus 320, Boeing 747)
DIM_CLIENTE	DIMENSION	Creada	Dimensión de clientes/pasajeros con ciudad de residencia
FACT_VENTAS_VUELOS	FACT	Creada	Tabla de hechos con métricas de ventas y FKs a 5 dimensiones

```
=====
Data Mart (Esquema Estrella) creado exitosamente
Tablas de dimensiones y hechos listas para carga ETL
=====
```

El Data Mart se creó exitosamente con todas las tablas dimensionales y de hechos. El modelo está estructuralmente completo y listo para iniciar el **proceso ETL** de carga de datos desde el esquema transaccional IATA (OLTP).

### Fase 3: Proceso ETL

- ☒ Diseñar queries de extracción desde esquema IATA (OLTP)
- ☐ Implementar transformaciones de datos
- ☐ Cargar dimensiones (DIM\_TIEMPO, DIM\_RUTA, DIM\_AEROLINEA, DIM\_MODELO, DIM\_CLIENTE)
- ☐ Cargar tabla de hechos (FACT\_VENTAS\_VUELOS)
- ☐ Validar integridad referencial y calidad de datos

### Diseño de Queries de Extracción (ETL)

Se han diseñado las consultas SQL de extracción, transformación y carga (ETL) para poblar todas las tablas del Data Mart desde el esquema transaccional IATA (OLTP). Estas queries se encuentran documentadas en el archivo [CodigosETL-Input\\_Table.sql](#).

### Herramienta ETL utilizada: Pentaho Data Integrator (PDI/Kettle)

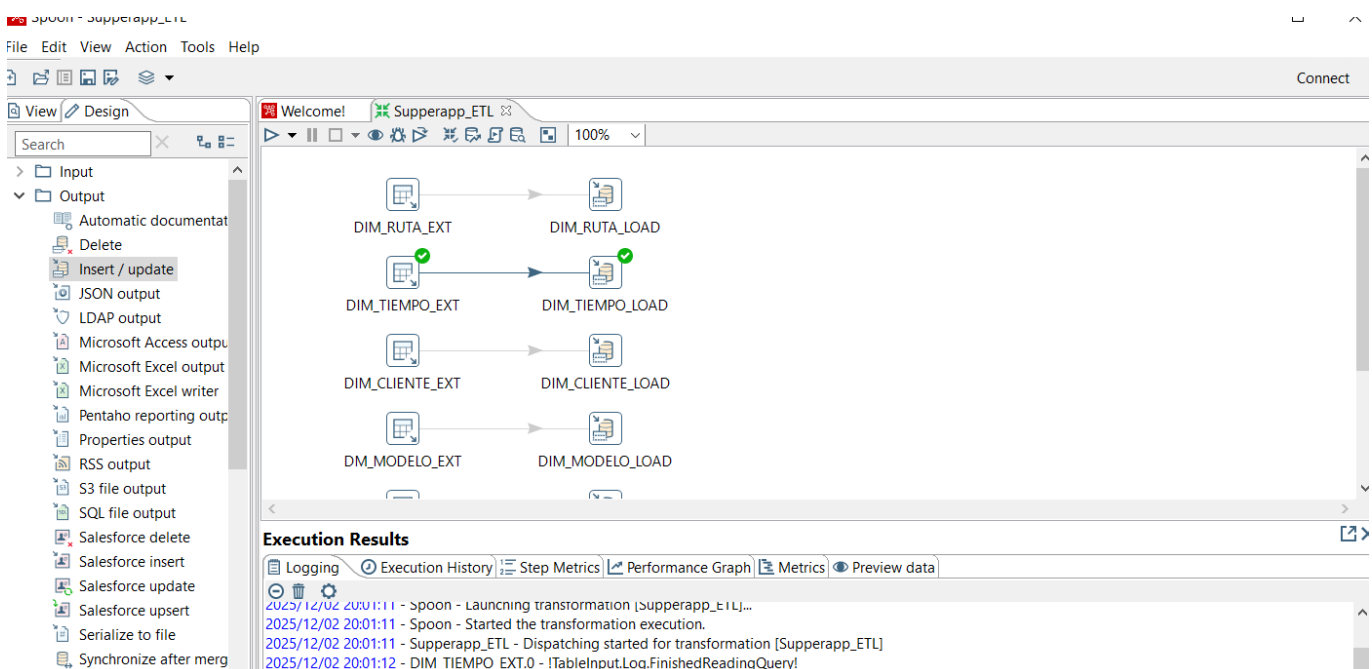
Los queries de extracción fueron implementados en **Pentaho Data Integrator**, una herramienta visual de ETL que permite diseñar flujos de integración de datos mediante transformaciones gráficas. Cada query se configuró como un componente "Table Input" dentro de una transformación PDI, facilitando la carga automatizada desde el esquema OLTP (IATA) hacia el Data Mart OLAP (IATA\_OLAP).

Las capturas de pantalla de cada transformación ETL implementada en Pentaho se encuentran en la carpeta [/images/](#) del repositorio y se muestran a continuación para cada tabla dimensional y de hechos.

### Queries de extracción por tabla:

#### 1. DIM\_TIEMPO - Extracción de dimensión temporal

#### Implementación en Pentaho Data Integrator:



Query de extracción:

```
SELECT
  ROW_NUMBER() OVER (ORDER BY f)          AS ID_TIEMPO,
  f                                         AS FECHA,
  EXTRACT(YEAR FROM f)                    AS ANIO,
  CASE
    WHEN EXTRACT(MONTH FROM f) BETWEEN 1 AND 6 THEN 1
    ELSE 2
  END                                     AS SEMESTRE,
  TO_NUMBER(TO_CHAR(f, 'Q'))              AS TRIMESTRE,
  EXTRACT(MONTH FROM f)                   AS MES,
  TO_CHAR(f, 'fmMonth', 'NLS_DATE_LANGUAGE=SPANISH') AS NOMBRE_MES,
  EXTRACT(DAY FROM f)                     AS DIA,
  TO_NUMBER(TO_CHAR(f, 'D'))              AS DIA_SEMANA,
  TO_CHAR(f, 'fmDay', 'NLS_DATE_LANGUAGE=SPANISH') AS NOMBRE_DIA
FROM (
  SELECT DISTINCT TRUNC(fecha_salida) AS f FROM IATA.ITINERARIOS
  UNION
  SELECT DISTINCT TRUNC(fecha_llegada) AS f FROM IATA.ITINERARIOS
)
ORDER BY f;
```

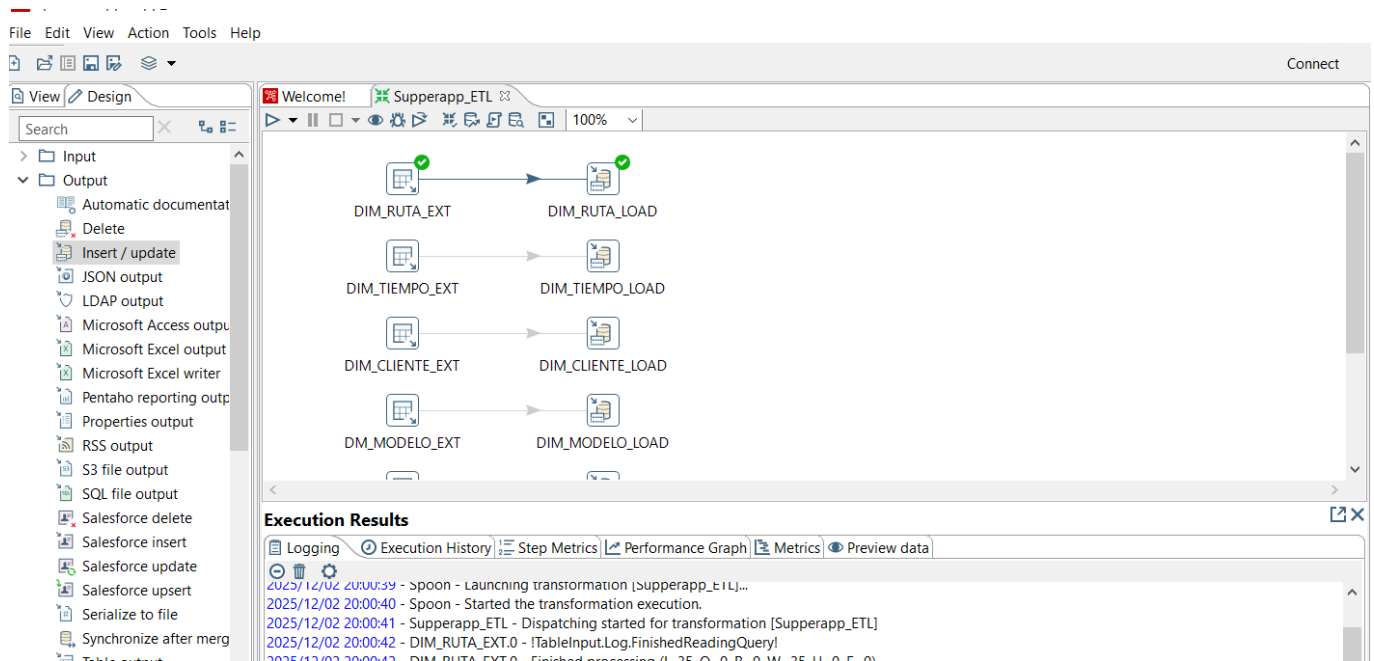
Resultado de carga (primeros 5 registros):

ID_TIEMPO	FECHA	ANIO	SEMESTRE	NOMBRE_MES
1	2019-01-15	2019	1	Enero
2	2019-02-20	2019	1	Febrero
3	2019-03-10	2019	1	Marzo
4	2019-07-22	2019	2	Julio
5	2019-10-25	2019	2	Octubre

**Descripción:** Extrae todas las fechas únicas de salida y llegada de vuelos desde `IATA.ITINERARIOS`, generando automáticamente los atributos temporales (año, semestre, trimestre, mes, día, nombres en español).

2. DIM\_RUTA - Extracción de dimensión geográfica

Implementación en Pentaho Data Integrator:



### Query de extracción:

```
SELECT
    i.id_itinerario                AS ID_RUTA,
    c_o.nombre                     AS CIUDAD_ORIGEN,
    a_o.nombre                     AS AEROPUERTO_ORIGEN,
    c_d.nombre                     AS CIUDAD_DESTINO,
    a_d.nombre                     AS AEROPUERTO_DESTINO,
    c_o.nombre || ' - ' || c_d.nombre AS RUTA_COMPLETA
FROM IATA.ITINERARIOS i
JOIN IATA.AEROPUERTOS a_o ON i.id_aeropuerto_origen = a_o.id_aeropuerto
JOIN IATA.CIUDADES c_o ON a_o.id_ciudad = c_o.id_ciudad
JOIN IATA.AEROPUERTOS a_d ON i.id_aeropuerto_destino = a_d.id_aeropuerto
JOIN IATA.CIUDADES c_d ON a_d.id_ciudad = c_d.id_ciudad;
```

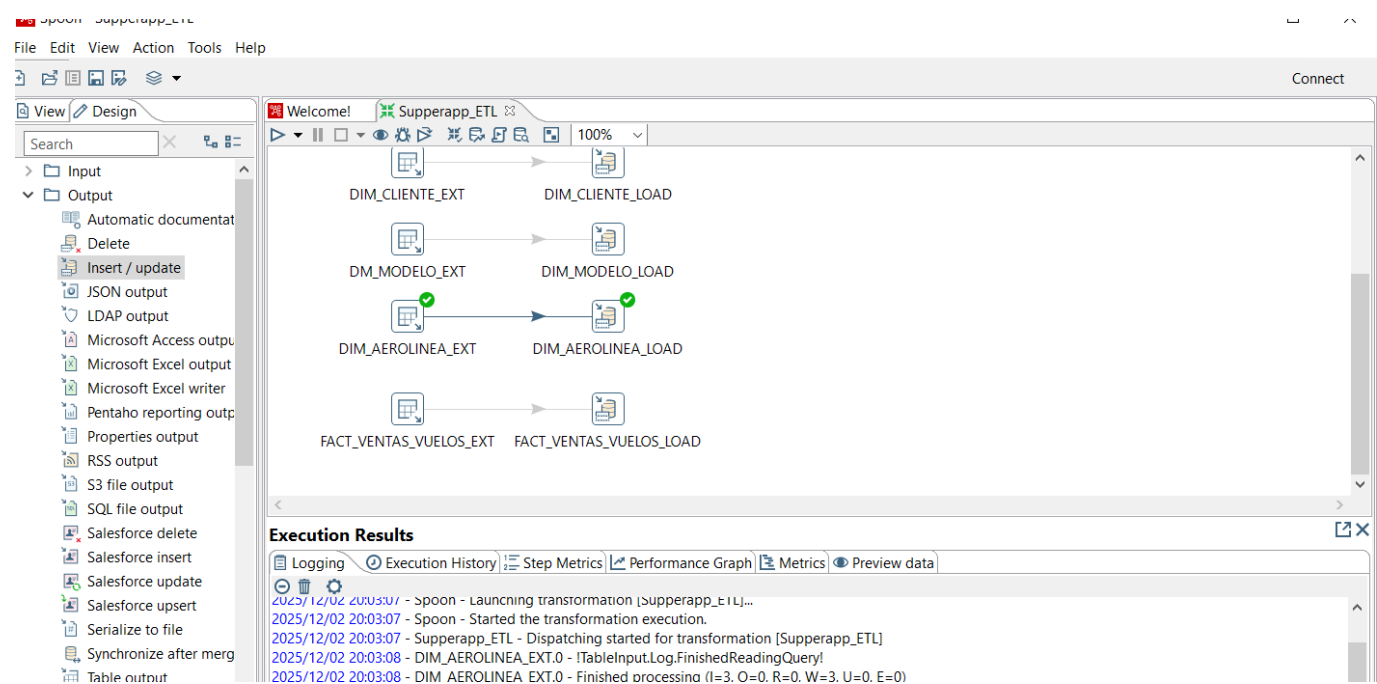
**Resultado de carga (primeros 5 registros):**

ID_RUTA	CIUDAD_ORIGEN	CIUDAD_DESTINO	RUTA_COMPLETA
1	Bogotá	Madrid	Bogotá - Madrid
2	Bogotá	Roma	Bogotá - Roma
3	Medellín	Madrid	Medellín - Madrid
4	Cali	Barcelona	Cali - Barcelona
5	Cartagena	Londres	Cartagena - Londres

**Descripción:** Combina itinerarios con aeropuertos y ciudades para generar rutas completas origen-destino con información descriptiva.

### 3. DIM AEROLINEA - Extracción de dimensión de aerolíneas

Implementación en Pentaho Data Integrator:



Query de extracción:

```
SELECT
  a.id_aerolinea AS ID_AEROLINEA,
  a.nombre      AS NOMBRE_AEROLINEA
FROM IATA.AEROLINEAS a;
```

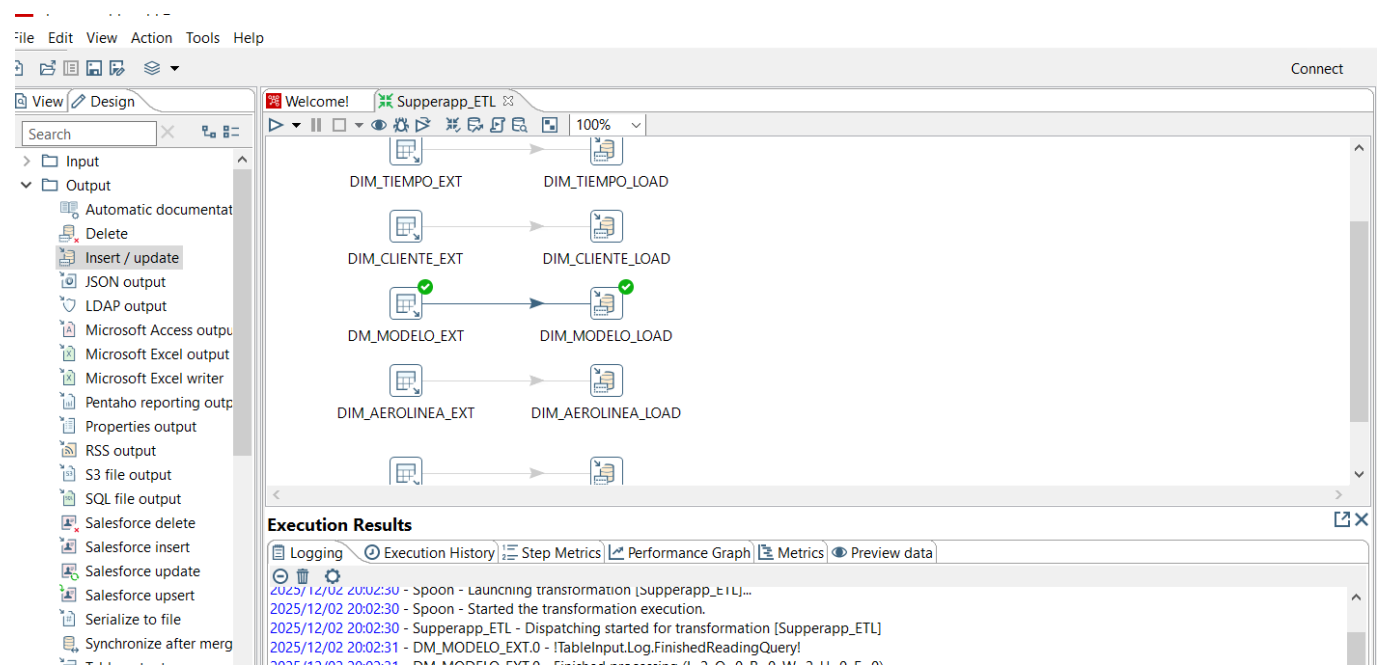
Resultado de carga (todos los registros):

ID_AEROLINEA	NOMBRE_AEROLINEA
1	Avianca
2	Latam
3	Wingo

**Descripción:** Extracción directa de todas las aerolíneas desde la tabla maestra IATA.AEROLINEAS.

4. DIM\_MODELO - Extracción de dimensión de modelos de avión

Implementación en Pentaho Data Integrator:



Query de extracción:

```
SELECT
    m.id_modelo AS ID_MODELO,
    m.nombre AS NOMBRE_MODELO
FROM IATA.MODELOS m;
```

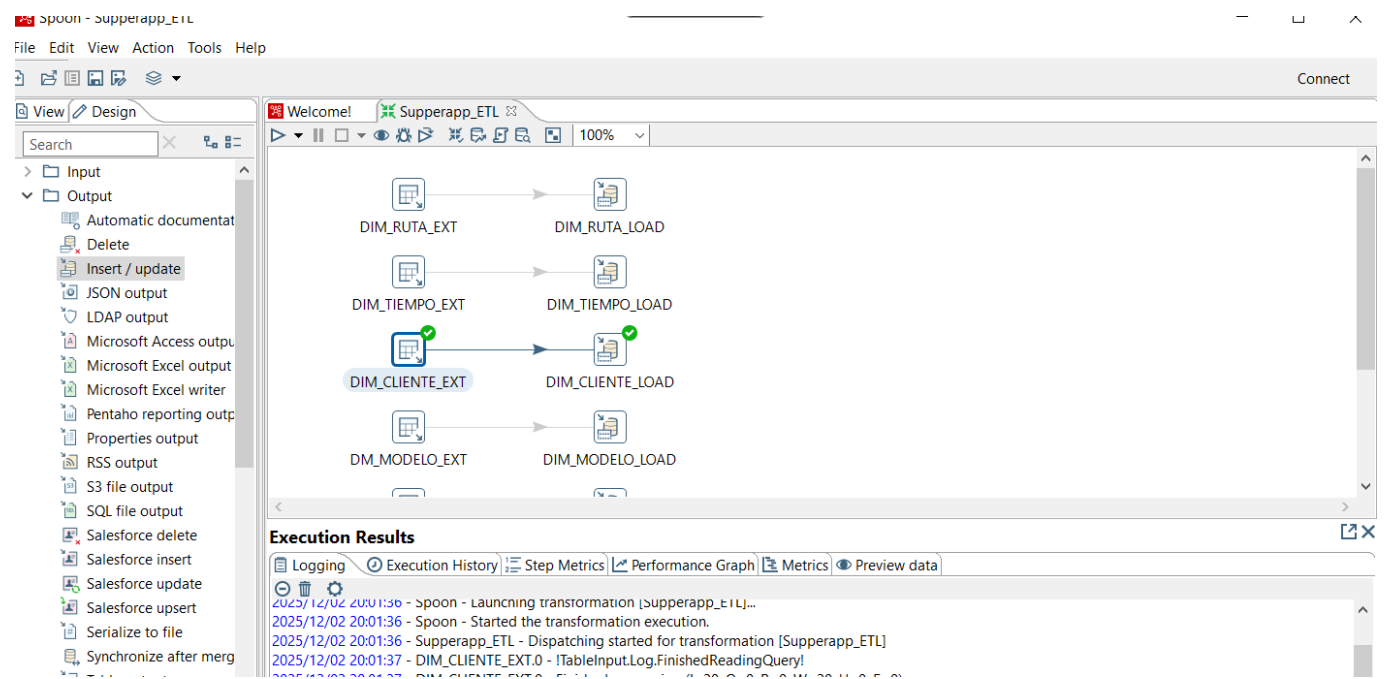
Resultado de carga (todos los registros):

ID_MODELO	NOMBRE_MODELO
1	Airbus 320
2	Boeing 747

**Descripción:** Extracción directa de todos los modelos de avión desde la tabla maestra IATA.MODELOS.

5. DIM\_CLIENTE - Extracción de dimensión de clientes/pasajeros

Implementación en Pentaho Data Integrator:



Query de extracción:

```
SELECT
    u.cedula AS ID_CLIENTE,
    u.nombre || ' ' || u.apellido AS NOMBRE_COMPLETO,
    u.email AS EMAIL,
    c.nombre AS CIUDAD_RESIDENCIA
FROM IATA.USUARIOS u
JOIN IATA.CIUDADES c ON u.id_ciudad = c.id_ciudad;
```

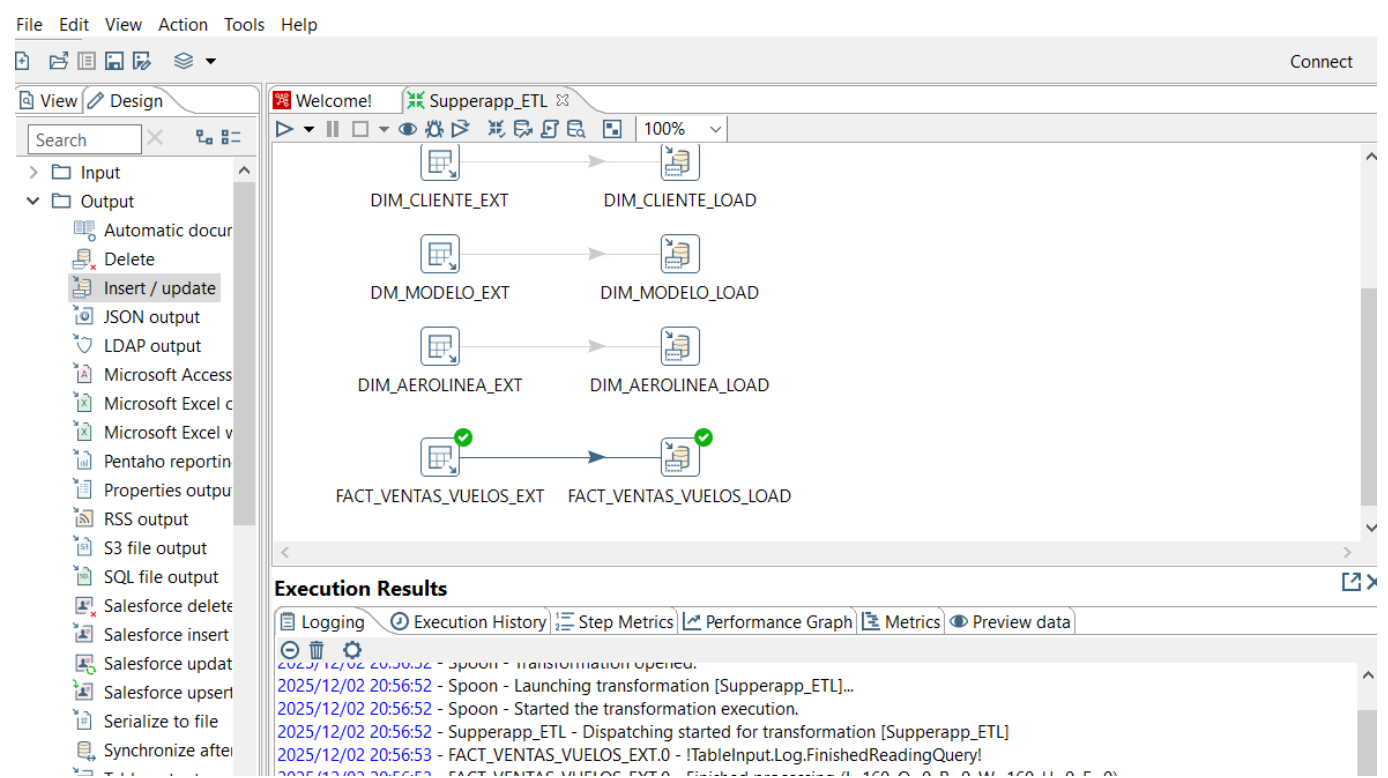
Resultado de carga (primeros 5 registros):

ID_CLIENTE	NOMBRE_COMPLETO	CIUDAD_RESIDENCIA
1001234567	Juan Pérez	Bogotá
1002345678	María García	Medellín
1003456789	Carlos López	Cali
1004567890	Ana Martínez	Barranquilla
1005678901	Luis Rodríguez	Cartagena

**Descripción:** Combina información de usuarios con sus ciudades de residencia, concatenando nombre y apellido en un solo campo.

6. FACT\_VENTAS\_VUELOS - Extracción de tabla de hechos

Implementación en Pentaho Data Integrator:



Query de extracción:

```
SELECT
    ROW_NUMBER() OVER (
        ORDER BY v.id_itinerario, v.id_avion, v.id_usuario
    ) AS ID_VENTA,
    dt.id_tiempo AS ID_TIEMPO,
    i.id_itinerario AS ID_RUTA,
    a.id_aerolinea AS ID_AEROLINEA,
    av.id_modelo AS ID_MODELO,
    TO_NUMBER(u.cedula) AS ID_CLIENTE,
    v.costo AS COSTO,
    (i.fecha_llegada - i.fecha_salida) * 24 AS DURACION_VUELO_HORAS,
    COUNT(*) OVER (
        PARTITION BY i.id_itinerario, av.id_avion
    ) AS CANTIDAD_PASAJEROS
FROM IATA.VUELOS v
JOIN IATA.ITINERARIOS i ON v.id_itinerario = i.id_itinerario
JOIN IATA.AVIONES av ON v.id_avion = av.id_avion
JOIN IATA.AEROLINEAS a ON av.id_aerolinea = a.id_aerolinea
JOIN IATA.USUARIOS u ON v.id_usuario = u.cedula
JOIN DIM_TIEMPO dt ON dt.fecha = TRUNC(i.fecha_salida);
```

Resultado de carga (primeros 5 registros):

ID_VENTA	ID_TIEMPO	ID_RUTA	ID_AEROLINEA	COSTO	CANTIDAD_PASAJEROS
1	1	1	1	850000.00	15
2	2	2	2	920000.00	12

ID_VENTA	ID_TIEMPO	ID_RUTA	ID_AEROLINEA	COSTO	CANTIDAD_PASAJEROS
3	3	3	1	880000.00	18
4	4	5	2	910000.00	14
5	5	6	1	930000.00	16

**Descripción:** Query complejo que integra datos de múltiples tablas transaccionales (VUELOS, ITINERARIOS, AVIONES, AEROLINEAS, USUARIOS) con la dimensión temporal ya cargada (DIM\_TIEMPO). Calcula métricas como duración del vuelo en horas y cantidad de pasajeros por vuelo.

Ver código completo de todos los queries ETL: [CodigosETL-Input\\_Table.sql](#)

Ver capturas de pantalla completas: [Carpeta /images/](#) en [GitHub](#)

#### Secuencia de ejecución ETL:

1. Cargar **DIM\_TIEMPO** (sin dependencias)
2. Cargar **DIM\_RUTA** (sin dependencias)
3. Cargar **DIM\_AEROLINEA** (sin dependencias)
4. Cargar **DIM\_MODELO** (sin dependencias)
5. Cargar **DIM\_CLIENTE** (sin dependencias)
6. Cargar **FACT\_VENTAS\_VUELOS** (requiere que DIM\_TIEMPO esté poblada)

**Validación:** Verificar integridad referencial y conteo de registros en cada paso.

#### Fase 4: Diseño del Cubo OLAP Multidimensional

- ☒ Diseñar esquema OLAP con Schema Workbench de Pentaho
- ☐ Implementar cubo en servidor LinceBI
- ☐ Publicar cubo para análisis multidimensional
- ☐ Validar conectividad y consultas MDX

#### Creación de Cubos OLAP con Schema Workbench

Para habilitar análisis multidimensional interactivo sobre el Data Mart, se han diseñado **dos cubos OLAP** utilizando la herramienta **Pentaho Schema Workbench**. Esta herramienta permite definir esquemas multidimensionales que luego son utilizados por el motor Mondrian para ejecutar consultas MDX.

#### Herramientas utilizadas:

- **Pentaho Schema Workbench:** Editor visual para diseñar esquemas OLAP
- **LinceBI:** Plataforma de Business Intelligence para visualización y análisis interactivo
- **Mondrian:** Motor OLAP que interpreta el esquema XML

#### ¿Por qué dos cubos OLAP?

Se crearon dos cubos con configuraciones distintas debido a que las **jerarquías de las dimensiones** deben ajustarse según la prioridad analítica de cada pregunta de negocio. En particular, la dimensión **DIM\_RUTA**

requiere diferentes ordenamientos jerárquicos según si se analiza desde el punto de vista del **destino** o del **origen** de los vuelos.

Archivos generados:

Cubo	Archivo	Uso
Cubo 4.1	Superapp_Cube4.1_Schema.xml	Para resolver las <b>preguntas 1, 2 y 3</b> (análisis enfocado en destinos)
Cubo 4.2	Superapp_Cube4.2_Schema.xml	Para resolver la <b>pregunta 4</b> (análisis enfocado en origen/residencia)


Regla de uso:

- **Cubo 4.1 (Superapp\_Cube4.1\_Schema):** Para los tres primeros requerimientos de análisis
- **Cubo 4.2 (Superapp\_Cube4.2\_Schema):** Para el último requerimiento de análisis

Esta separación garantiza que las jerarquías dimensionales estén optimizadas para cada tipo de consulta analítica, facilitando el analisis y la navegación intuitiva en LinceBI.

Diferencias Clave entre Cubos

Cubo 4.1 (Superapp\_Cube4.1\_Schema):


Diseño del Cubo OLAP 4.1 en Schema Workbench

```
DIM_RUTA Jerarquía:
└ Ciudad_destino (nivel 1 - más alto)
  └ Ciudad_origen (nivel 2)
    └ Aeropuerto_origen (nivel 3)
      └ Aeropuerto_destino (nivel 4)
        └ Ruta_completa (nivel 5 - más bajo)
```

**Uso:** Análisis que priorizan el **destino** como punto de partida del análisis:

- Pregunta 1: ¿Qué aerolíneas vuelan más a **ciudades destino** específicas?
- Pregunta 2: ¿Cuánto recaudan las aerolíneas por semestre?
- Pregunta 3: ¿Qué modelos de avión realizan más vuelos?

Cubo 4.2 (Superapp\_Cube4.2\_Schema):

Diseño del Cubo OLAP 4.2 en Schema Workbench

```
DIM_RUTA Jerarquía:
└ Ciudad_origen (nivel 1 - más alto)
  └ Ciudad_destino (nivel 2)
    └ Aeropuerto_origen (nivel 3)
      └ Aeropuerto_destino (nivel 4)
        └ Ruta_completa (nivel 5 - más bajo)
```

**Uso:** Análisis que priorizan el **origen/residencia** como punto de partida del análisis:

- Pregunta 4: ¿Desde qué **ciudades de residencia** los habitantes viajan más?

**Ver archivos completos de los cubos:**

- [Superapp\\_Cube4.1\\_Schema.xml](#) - Preguntas 1, 2, 3
- [Superapp\\_Cube4.2\\_Schema.xml](#) - Pregunta 4

Fase 5: Análisis y Reporting (Pendiente)

- ☐ Ejecutar análisis OLAP en LinceBI
- ☐ Crear dashboards interactivos
- ☐ Generar reportes de negocio
- ☐ Documentar insights y hallazgos

---

## Autores

**Estudiante:** Edwin Silva Salas, Carlos Preciado Cárdenas, Cristian Restrepo Zapata

**Programa:** Maestría en Ciencia de Datos

**Universidad:** Pontificia Universidad Javeriana

**Repositorio GIT:** [IATA\\_CASE\\_MCD](#)