Universidad de San Carlos de Guatemala
Facultad de Ingeniería
Escuela de Estudios de Postgrado
Introducción al Análisis de Datos

Proyecto Integrador de Análisis de Datos (CRISP-DM)

Grupo #3
Subconjunto de datos 2011 a 2021

Luis Fernando Culajay Sandoval	3548487470101
Karen Lisbeth Morales Marroquin	3638451000101
Jhonatan Emmanuel Simon Romano	1703790780101
Brandon Rene Portillo González	3075926690603

1. Business Understanding

Objetivos análiticos

1. Entender la tasa de puntualidad de cada aerolínea comparando tiempos programados y reales
2. Comprender la relevancia de las diferentes causas de demora y su distribución
3. Identificar las aerolíneas y aeropuertos con mayor concurrencia de uso.
4. Identificar las rutas más afectadas por desvíos y cancelaciones

KPI's Iniciales

1. Porcentaje de vuelos puntuales
   Fórmula: (Vuelos sin retraso / Total de vuelos)×100
2. Porcentaje de desvíos
   Fórmula: (Vuelos desviados / Total de vuelos)×100
3. Porcentaje de cancelación
   Fórmula: (Vuelos cancelados / Total de vuelos)×100
4. Porcentaje de desvíos
   Fórmula: (Vuelos desviados / Total de vuelos)×100
5. Porcentaje de vuelos con retraso por aerolínea
   Fórmula: (Vuelos con retraso / Total de vuelos por aerolínea)×100

Lista de requerimientos analíticos

- Software: Python, SQL Server, Power BI, Docker, Office (Excel)
- Infraestructura: Computadora con al menos 10GB de espacio en disco disponible y 8GB de RAM, conexión a internet.

Preguntas de negocio

- ¿Qué aerolíneas y aeropuertos presentan mejor o peor puntualidad?
- ¿Cuáles son las causas de demora más relevantes y cómo se distribuyen por aerolínea/aeropuerto?
- ¿Qué rutas son más vulnerables a retrasos, cancelaciones o desvíos?
- ¿Qué tan diferentes son los tiempos programados vs reales y qué factores influyen en esa brecha?
- ¿Qué aeropuertos con mayor concurrencia muestran congestión operacional?
- ¿Las rutas largas tienen mejor o peor puntualidad que las rutas cortas?

Canvas Data Product

<table>
<tr>
<td rowspan="2">Problema

● Se desconoce qué aerolíneas presentan mejor o peor desempeño en puntualidad.
● No se sabe qué causas de demora son más frecuentes o relevantes.
● No está claro cuáles aeropuertos y rutas concentran mayor tráfico y riesgo operativo.
● Las cancelaciones y desvíos impactan la eficiencia del sistema y requieren identificación.</td>
<td>Datos

Se tiene un dataset de Bureau of Transportation Statistics (BTS) que contiene horarios, retrasos, causas de demora, aerolíneas, aeropuertos, tiempos operacionales y cancelaciones.</td>
<td rowspan="2">Solución

La solución será un dashboard analítico que permita visualizar patrones de puntualidad, causas de retraso, concurrencia y rutas con problemas, entre otros, siendo un consolidado de los problemas encontrados en los vuelos.</td>
<td>KPI’s

● Porcentaje de vuelos puntuales
● Porcentaje de desvíos
● Porcentaje de cancelación
● Porcentaje de desvíos
● Porcentaje de vuelos con retraso</td>
<td rowspan="2">Acciones

Las acciones derivadas del dashboard permitirán identificar patrones, segmentar aerolíneas y aeropuertos y visualizar horarios o rutas vulnerables para realizar mejores planificaciones en el tráfico aéreo</td>
</tr>
<tr>
<td>Hipótesis

● Algunas aerolíneas son más puntuales que otras.
● NAS y LateAircraftDelay serán las causas dominantes.
● Aeropuertos de alta concurrencia tendrán más retrasos.
● Algunas rutas presentan más cancelaciones y desvíos</td>
<td>Actores

● Cliente general: Entidades reguladoras de vuelos/tráfico aéreo
● Stakeholders: Aerolíneas, aeropuertos, reguladores.
● Usuarios: Analistas e investigadores
● Impacto en: planificación, eficiencia operativa y experiencia del pasajero.</td>
</tr>
<tr>
<td colspan="3">Valores / Riesgos

El resultado aportará un gran valor al generar visibilidad sobre la puntualidad y causas de demora Los riesgos se encuentran en los límites que pueda dar el dataset debido a la falta de datos externos.</td>
<td colspan="3">Rendimiento / Impacto

Capacidad de incrementar la tasa de vuelos puntuales, reducir el número de vuelos cancelados, entender cómo reducir la variabilidad entre tiempos programados y reales.</td>
</tr>
</table>

2. Data Understanding

Diccionario de datos

Código	Descripción del código	Muestra	Tipo Dato
Year	Indica el año de operación del vuelo	YEAR	NUMERO
Month	Indica el mes de operación del vuelo	MONTH	NUMERO
DayofMonth	Día calendario del vuelo	Número del día del mes	NUMERO
DayOfWeek	Día de la semana del vuelo	WEEK	NUMERO
Reporting_Airline	Código de la aerolínea que reporta la información	AIRLINE	TEXTO
OriginAirportID	Código del aeropuerto de salida	ID_AIRPORT	NUMERO
OrigenCityMarketID	Código de la ciudad a la que pertenece el aeropuerto de salida	ID_CITY	NUMERO
Origin	Código IATA de tres letras del aeropuerto de origen	ORIGIN	TEXTO
OriginCityName	Nombre de la ciudad donde se localiza el aeropuerto de origen.	ID_CITY	TEXTO
OriginState	Código del estado donde se ubica el aeropuerto de origen.	ORIGINSTATE	TEXTO
OriginStateName	Nombre del estado donde se encuentra el aeropuerto de origen.	ORIGINSTATE	TEXTO
DestAirportID	Código del aeropuerto de destino	ID_AIRPORT	NUMERO
DestCityMarketID	Código de la ciudad a la que pertenece el aeropuerto de destino	ID_CITY	NUMERO
Dest	Código IATA de tres letras del aeropuerto de destino	ORIGIN	TEXTO
DestCityName	Nombre de la ciudad donde se localiza el aeropuerto de destino.	ID_CITY	TEXTO
DestState	Código del estado donde se ubica el aeropuerto de destino.	ORIGINSTATE	TEXTO
DestStateName	Nombre del estado donde se encuentra el aeropuerto de destino.	ORIGINSTATE	TEXTO
CRSDepTime	Hora programada de salida del vuelo	Número del día (1–31)	NUMERO
DepTime	Hora real de salida del vuelo	HHMM (hora local en formato 24h, sin dos	NUMERO

		puntos)	
DepDelay	Retraso de salida de vuelo	HHMM (hora local en formato 24h, sin dos puntos)	NUMERO
CRSArrTime	Hora programada de llegada del vuelo	HHMM (hora local en formato 24h, sin dos puntos)	NUMERO
ArrTime	Hora real de llegada del vuelo	HHMM (hora local en formato 24h, sin dos puntos)	NUMERO
ArrDelay	Retraso de llegada de vuelo	Numeros (puede ser negativo)	NUMERO
Cancelled	Identifica si el vuelo fue cancelado	CANCELLED	NUMERO
CancellationCode	Código que identifica la causa de la cancelación	CANCELLATION_CODE	NUMERO
Diverted	Identifica si el vuelo fue desviado	DIVERTED	NUMERO
CarrierDelay	Minutos de retraso causados por problemas internos de la aerolínea.	Numero (valores =>0)	NUMERO
WeatherDelay	Retrasos por condiciones meteorológicas.	Numero (valores =>0)	NUMERO
NASDelay	Retrasos por congestión del sistema de control aéreo nacional.	Numero (valores =>0)	NUMERO
SecurityDelay	Minutos de retraso por medidas o incidentes de seguridad.	Numero (valores =>0)	NUMERO
LateAircraftDelay	Retrasos por llegada tardía del avión desde un vuelo previo.	Numero (valores =>0)	NUMERO

Resumen de calidad de datos

El análisis de calidad de los datos evidenció que el conjunto de información presenta una alta completitud general, con la mayoría de los campos sin valores nulos como:

- YEAR
- MONTH
- OP_UNIQUE_CARRIER
- ORIGIN

- DEST

Sin embargo, se identificó un porcentaje elevado de valores faltantes 85.3% en las campos numéricas relacionadas con las causas específicas de demora (CARRIER_DELAY, WEATHER_DELAY, NAS_DELAY, SECURITY_DELAY y LATE_AIRCRAFT_DELAY).

Este comportamiento es habitual en el dataset de BTS, ya que dichos campos solo se registran cuando existe un retraso atribuible a una causa concreta, en caso contrario, su valor es nulo y representa ausencia de retraso.

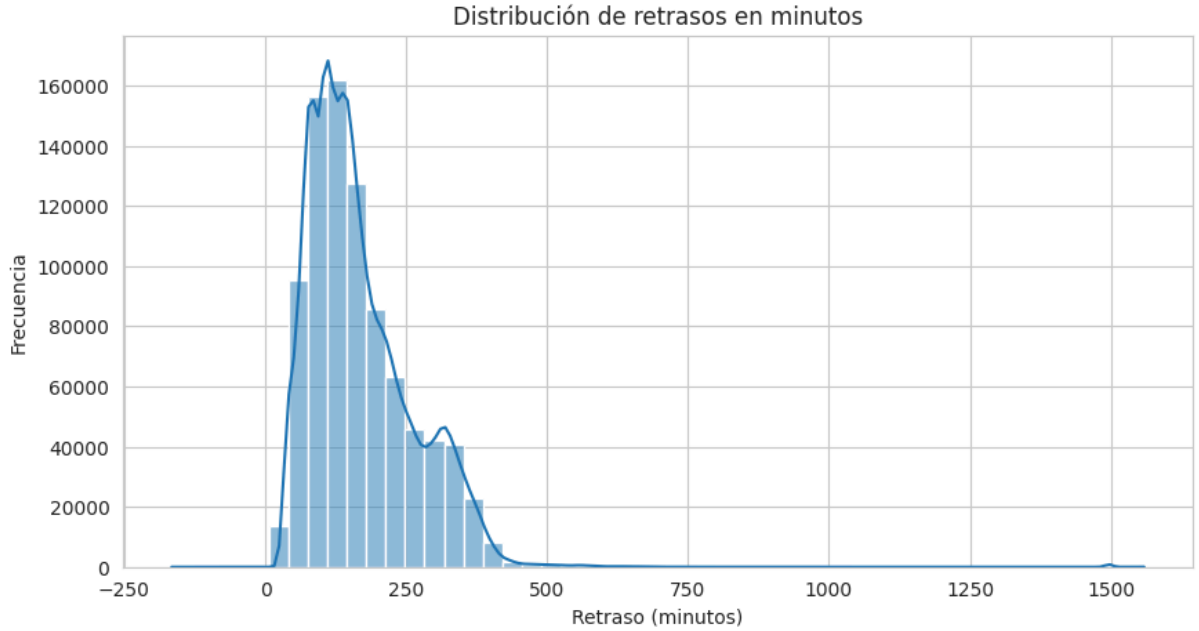
En las variables horarias (DEP_TIME, ARR_TIME, CRS_DEP_TIME, CRS_ARR_TIME), se detectaron valores fuera del rango válido (1–2400) en aproximadamente 0.9% de los registros, los cuales se eliminarán por inconsistencia temporal.

Finalmente, los campos booleanos (CANCELLED, DIVERTED) no presentan valores faltantes, garantizando la fiabilidad para medir cancelaciones y desvíos.

En conjunto, el dataset mantiene una consistencia superior al 98%, siendo necesario únicamente reemplazar los valores nulos en las variables de demora por cero y eliminar los registros con horas no válidas para asegurar la integridad del análisis posterior.

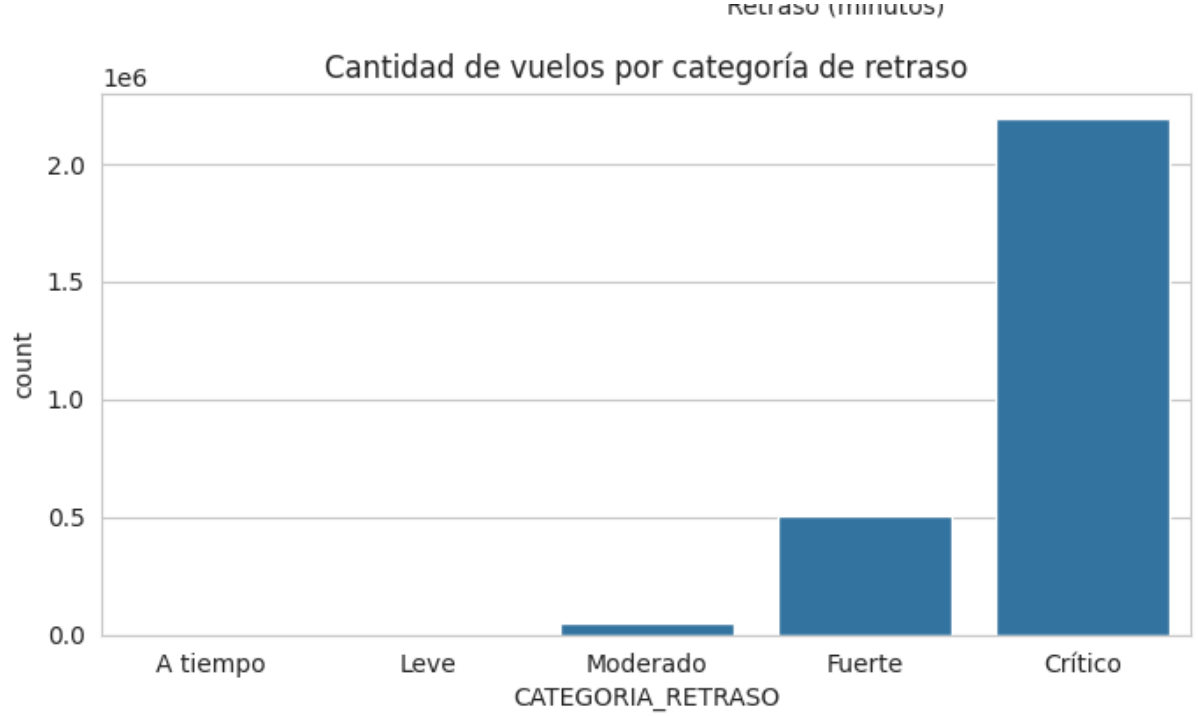
Gráficas exploratorias

Figura 1.
Distribución de retrasos en minutos



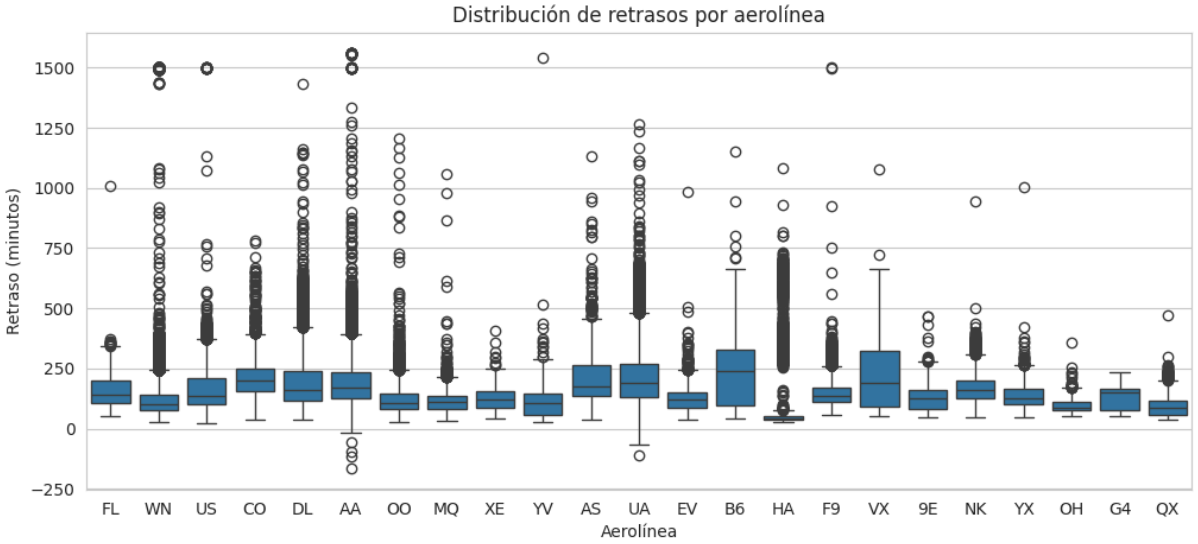
Nota. Elaboración propia, realizado con Python en Google Colab.

Figura 2.
Cantidad de vuelos por categoría de retraso



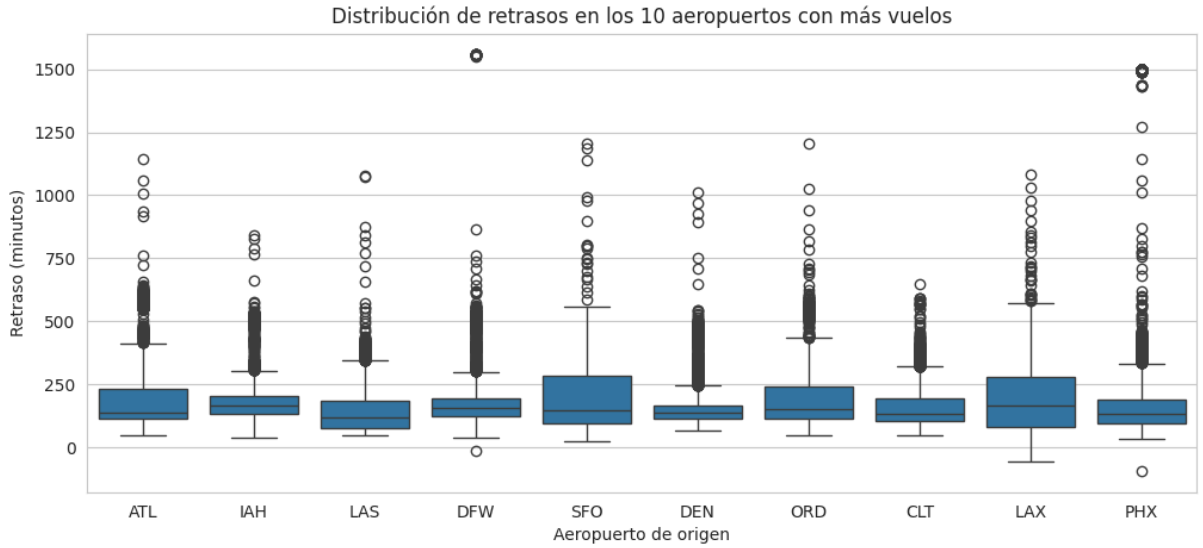
Nota. Elaboración propia, realizado con Python en Google Colab.

Figura 3.
Distribución de retrasos por aerolínea



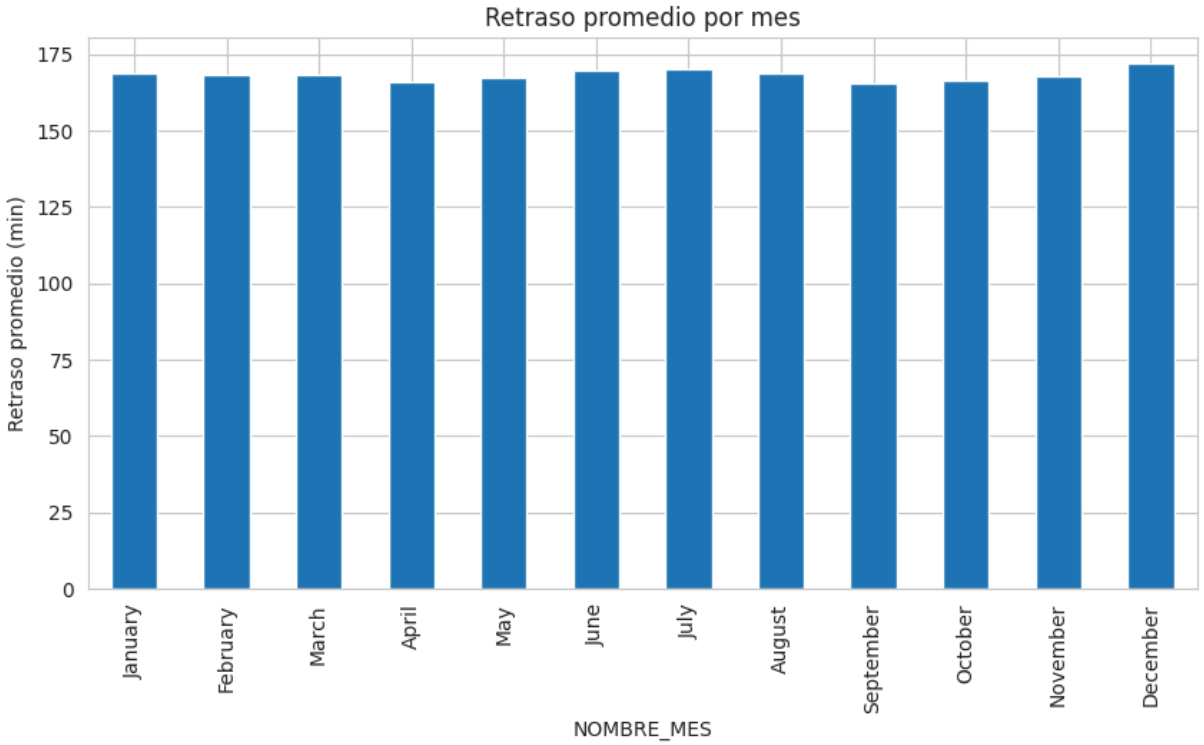
Nota. Elaboración propia, realizado con Python en Google Colab.

Figura 4.
Distribución de retrasos en los 10 aeropuertos con más vuelos



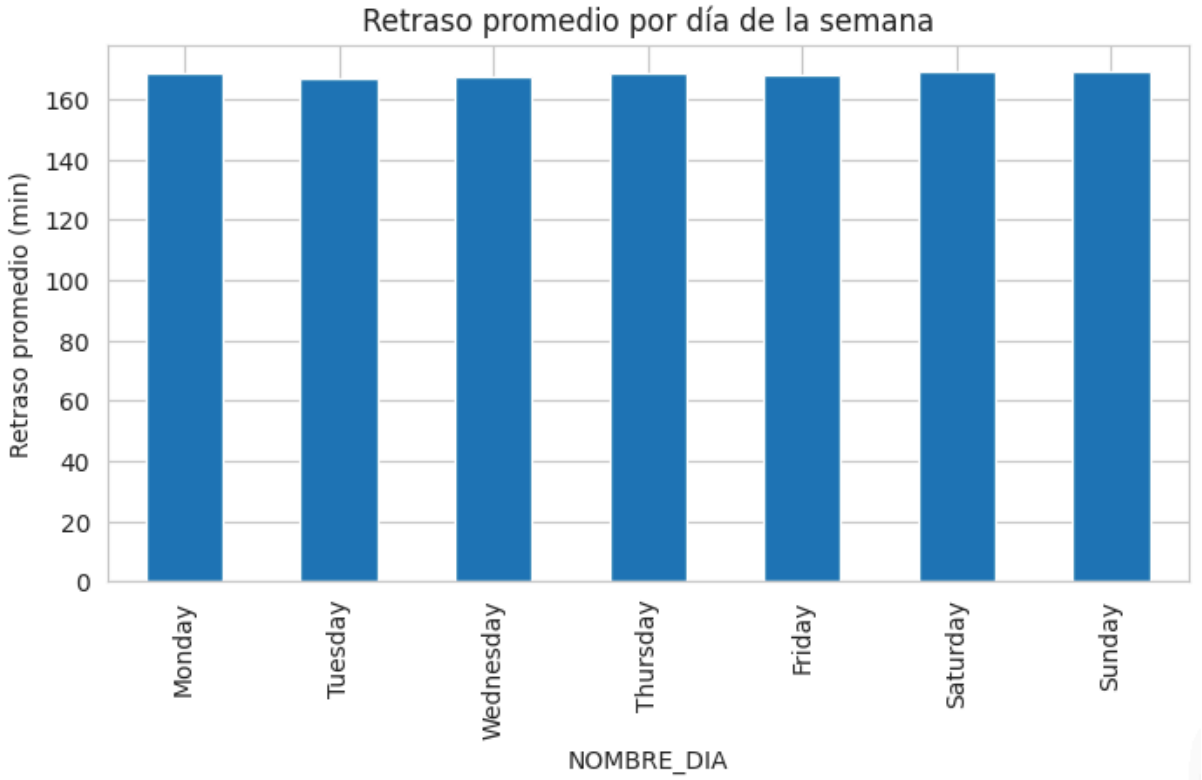
Nota. Elaboración propia, realizado con Python en Google Colab.

Figura 5.
Retraso promedio por mes



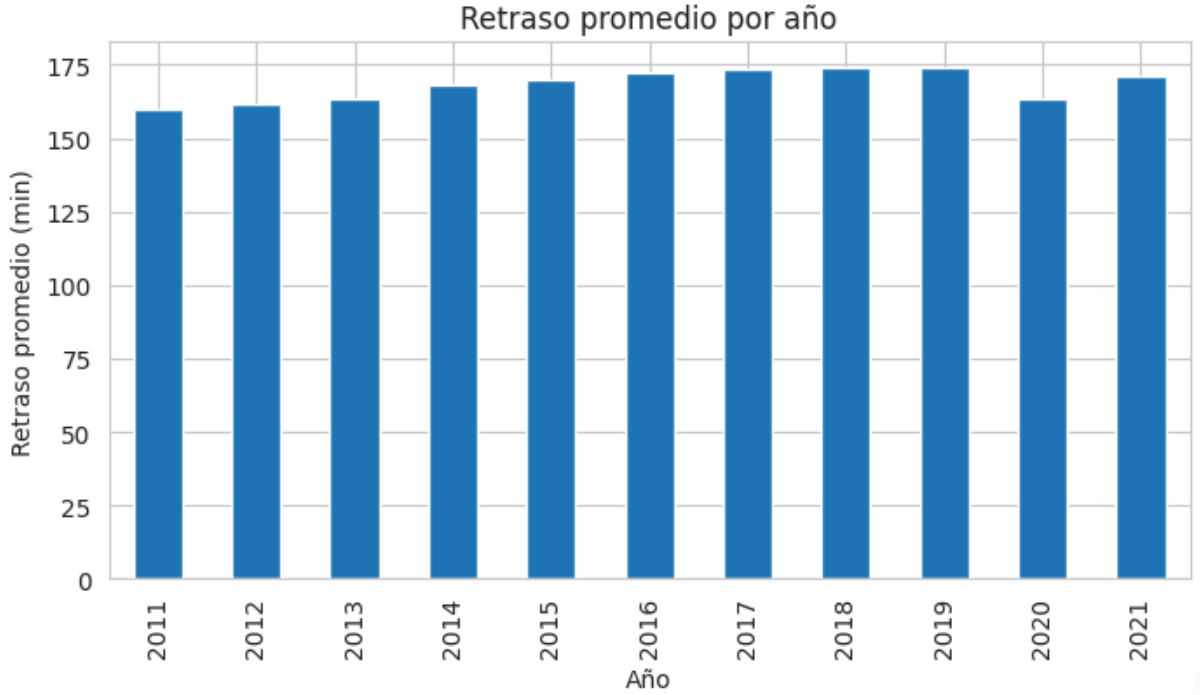
Nota. Elaboración propia, realizado con Python en Google Colab.

Figura 6.
Retraso promedio por día de la semana



Nota. Elaboración propia, realizado con Python en Google Colab.

Figura 7.
Retraso promedio por año

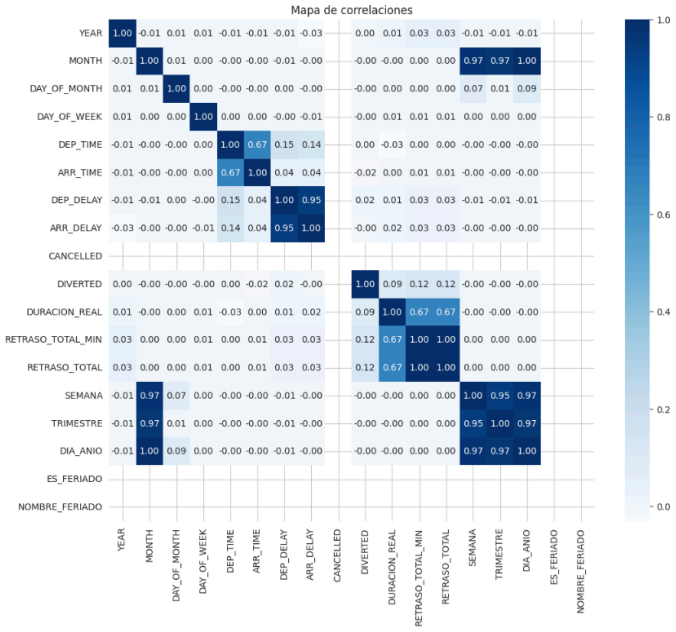


Nota. Elaboración propia, realizado con Python en Google Colab.

Figura 8.
Retraso promedio en días festivos vs días normales

Nota. Elaboración propia, realizado con Python en Google Colab.

Figura 9.
Mapeo de correlaciones

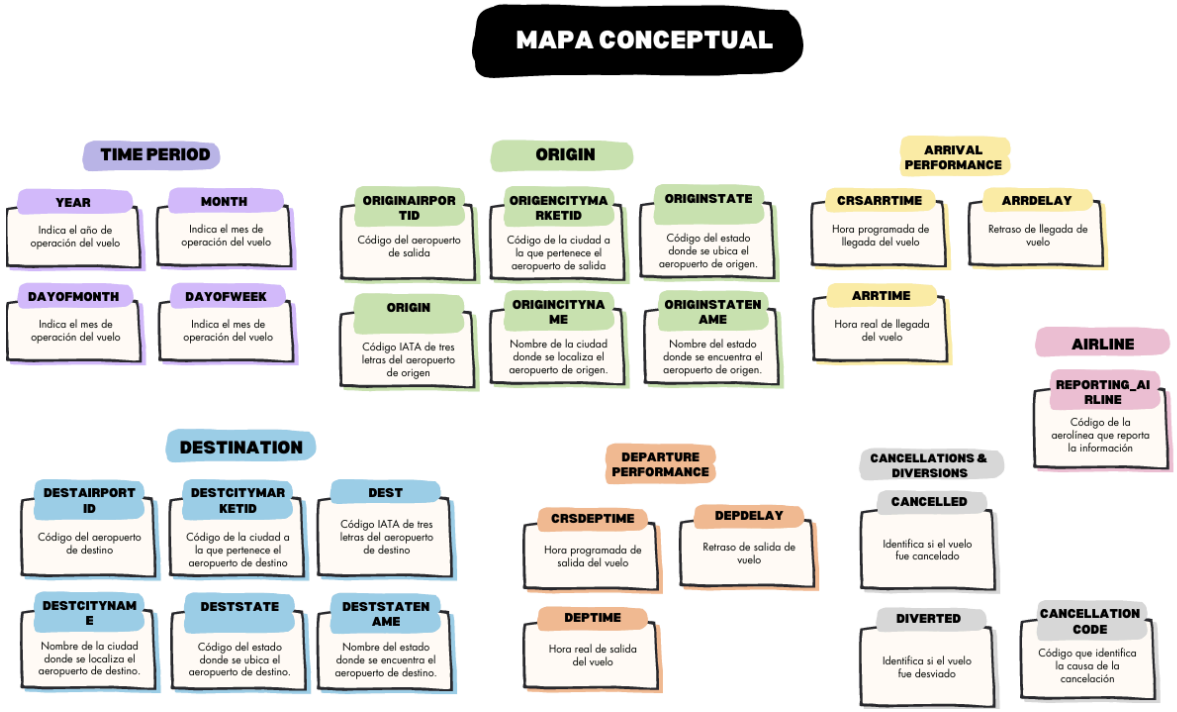


Nota. Elaboración propia, realizado con Python en Google Colab.

Figura 10.
Número de vuelos por mes

Nota. Elaboración propia, realizado con Python en Google Colab.

Mapa conceptual del dataset



3. Data Preparation

Scripts SQL y Python

- EDA.ipynb: Limpieza y estandarización de datasets por año. Unificación de único dataset limpio

```python
import zipfile
import pyarrow as pa
import pyarrow.parquet as pq
import os
import pandas as pd # Ensure pandas is imported if not already, as it's used

zip_path = "drive/MyDrive/Colab Notebooks/Proyecto Final/2. Data understanding/Dataset/Vuelos.zip"
csv_name = "Vuelos.csv"
salida = "drive/MyDrive/Colab Notebooks/Proyecto Final/2. Data understanding/Dataset/Vuelos_raw.parquet"

chunksize = 200_000

# Eliminar el archivo de salida si ya existe para asegurar un esquema consistente
if os.path.exists(salida):
    os.remove(salida)
    print(f"Archivo existente '{salida}' eliminado.")

writer = None  # se inicializa después del primer chunk

with zipfile.ZipFile(zip_path) as z:
    with z.open(csv_name) as f:
        for i, chunk in enumerate(pd.read_csv(f, chunksize=chunksize, low_memory=False)):

            # Convertir CRS_ARR_TIME a float para asegurar consistencia de tipo
            # Esto maneja casos donde algunos chunks pueden tener NaNs en esta columna
            # forzar la inferencia a double (float64 en pandas)
            if 'CRS_ARR_TIME' in chunk.columns:
                chunk['CRS_ARR_TIME'] = chunk['CRS_ARR_TIME'].astype(float)

            # Also ensure CRS_DEP_TIME is float for consistency, as per error
            if 'CRS_DEP_TIME' in chunk.columns:
                chunk['CRS_DEP_TIME'] = chunk['CRS_DEP_TIME'].astype(float)

            # Convertir pandas → Arrow Table
            table = pa.Table.from_pandas(chunk, preserve_index=False)

            # Crear writer en primer chunk
            if writer is None:
                writer = pq.ParquetWriter(salida, table.schema)

            # Escribir chunk
            writer.write_table(table)

            print(f"Procesado chunk {i}")

# cerrar writer al final
if writer:
    writer.close()
    print("Procesamiento completado.")
```

- columnas_derivadas.sql: Creación de columnas derivadas a partir el resto de campos

```sql
-- Adicion de nuevas columnas a la tabla de hechos
ALTER TABLE fact_flights ADD
    retraso_total INT NULL,
    semana INT NULL,
    trimestre INT NULL,
    dia_festivo INT NULL;

ALTER TABLE fact_flights
ADD CONSTRAINT FK_fact_festivo
FOREIGN KEY (dia_festivo)
REFERENCES dim_dias_festivos(festivo_id);

-- Creacion de valores derivados
UPDATE fact_flights
SET retraso_total = ISNULL(retraso_salida, 0) + ISNULL(retraso_llegada, 0);

UPDATE f
SET semana = DATEPART(WEEK, d.full_date)
FROM fact_flights f
JOIN dim_date d ON f.fecha_id = d.date_id;

UPDATE f
SET trimestre = DATEPART(QUARTER, d.full_date)
FROM fact_flights f
JOIN dim_date d ON f.fecha_id = d.date_id;

UPDATE f
SET festivo_id = h.festivo_id
FROM fact_flights f
JOIN dim_date d
    ON f.fecha_id = d.date_id
JOIN dim_dias_festivos h
    ON MONTH(d.full_date) = h.mes
   AND DAY(d.full_date)  = h.dia;

UPDATE f
SET dia_festivo = h.festivo_id
FROM fact_flights f
JOIN dim_date d
    ON f.fecha_id = d.date_id
JOIN dim_dias_festivos h
    ON MONTH(d.full_date) = h.mes
   AND DAY(d.full_date)  = h.dia;
```

Dataset final listo para modelado

https://drive.google.com/drive/folders/1Czm-nzUHuskAq05C3tVAhAzLi2bN9L2q?usp=sharing

Resumen de transformaciones (ETL)

- Conversión de tiempos e identificadores en valores enteros para asegurar la consistencia
- Eliminación de valores duplicados
- Reemplazar valores nulos en las variables de demora por cero
- Eliminación de los registros con horas no válidas para asegurar la integridad del análisis posterior.

```
#Se inputaron los datos vacios por 0
df_sample[['ARR_TIME', 'ARR_DELAY']] = df_sample[['ARR_TIME', 'ARR_DELAY']].fillna(0)
```

```
df_sample.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2750000 entries, 0 to 2749999
Data columns (total 13 columns):
 #   Column             Dtype
---  ------             -----
 0   YEAR               int64
 1   MONTH              int64
 2   DAY_OF_MONTH       int64
 3   DAY_OF_WEEK        int64
 4   OP_UNIQUE_CARRIER  object
 5   ORIGIN             object
 6   DEST               object
 7   DEP_TIME           int64
 8   ARR_TIME           float64
 9   DEP_DELAY          float64
 10  ARR_DELAY          float64
 11  CANCELLED          int64
 12  DIVERTED           int64
dtypes: float64(3), int64(7), object(3)
memory usage: 272.8+ MB
```

```
#Se eliminan duplicados
df_sample.drop_duplicates(inplace=True)
```

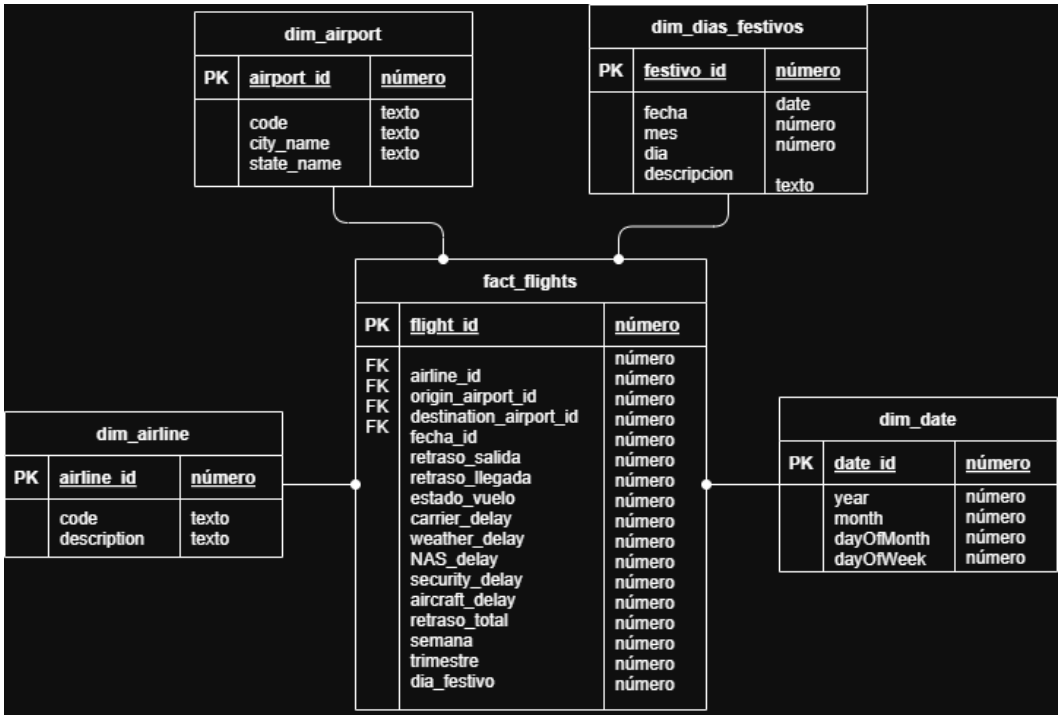
```
df.isnull().sum()
```

	0
YEAR	0
MONTH	0
DAY_OF_MONTH	0
DAY_OF_WEEK	0
OP_UNIQUE_CARRIER	0
ORIGIN	0
DEST	0
DEP_TIME	0
ARR_TIME	64
DEP_DELAY	0
ARR_DELAY	596
CANCELLED	0
DIVERTED	0

dtype: int64

4. Modeling

Diagrama físico del modelo estrella



Script SQL para crear la base

```sql
CREATE DATABASE DWBTS;

USE DWBTS;

CREATE TABLE dim_airline (
    airline_id INT IDENTITY(1,1) PRIMARY KEY,
    code VARCHAR(10),
    description VARCHAR(200)
);

CREATE TABLE dim_date (
    date_id INT IDENTITY(1,1) PRIMARY KEY,
    year INT NOT NULL,
    month INT NOT NULL,
    day_of_month INT NOT NULL,
    day_of_week INT NOT NULL,
    full_date DATE NOT NULL
);

CREATE TABLE dim_airport (
    airport_id INT IDENTITY(1,1) PRIMARY KEY,
    code VARCHAR(10) NOT NULL,
    city_name VARCHAR(100),
    state_name VARCHAR(100)
);

CREATE TABLE fact_flights (
    flight_id BIGINT IDENTITY(1,1) PRIMARY KEY,
    airline_id INT NOT NULL,
    origin_airport_id INT NOT NULL,
    destination_airport_id INT NOT NULL,
    fecha_id INT NOT NULL,
    retraso_salida INT,
```

Archivo con carga de datos

```python
# ============================================================
# === CARGAR DIMENSION AEROLÍNEAS
# ============================================================

def cargar_dim_airline():
    df = pd.read_csv(airlines_csv)

    df["Code"] = df["Code"].astype(str).str.strip().str[:10]
    df["Description"] = df["Description"].astype(str).str.strip().str[:10

    lista = df[["Code", "Description"]].values.tolist()

    cursor.executemany(
        "INSERT INTO dim_airline (code, description) VALUES (?, ?)",
        lista
    )
    conn.commit()
    print("✓ dim_airline cargado.")


# ============================================================
# === ETAPA 1: CARGAR DIM_DATE Y DIM_AIRPORT DESDE CHUNKS
# ============================================================

def procesar_dimensiones():
    print("Procesando dimensiones desde archivo de 10 GB...")

    fechas = set()
    aeropuertos = set()
    chunk_number = 0

    for chunk in pd.read_csv(
        flights_csv,
        chunksize=chunk_size,
        usecols=[
            "YEAR", "MONTH", "DAY_OF_MONTH", "DAY_OF_WEEK",
            "ORIGIN_AIRPORT_ID", "ORIGIN_CITY_NAME", "ORIGIN_STATE_NM",
            "DEST_AIRPORT_ID", "DEST_CITY_NAME", "DEST_STATE_NM"
        ]
    ):

        chunk_number += 1
        print(f"[{datetime.now().strftime('%Y-%m-%d %H:%M:%S')}] Procesan

        # === DIM_DATE (NO requiere transformar)
        fechas.update(
            list(
                zip(
```

5. Evaluation

Informe de evaluación

Propósito

El presente informe presenta los resultados de la validación del modelo analítico y la coherencia de los KPIs del dashboard del proyecto TranStats (Bureau of Transportation Statistics, 2011–2021), desarrollado bajo la metodología CRISP-DM. Su propósito es confirmar que el modelo y las visualizaciones reflejan fielmente los objetivos de negocio definidos en la etapa de Business Understanding.

Hallazgos Clave

> El modelo demuestra alta calidad y consistencia. Los KPIs principales: puntualidad, causas de demora y cancelaciones, son coherentes entre el dashboard y las consultas SQL, con una coincidencia superior al 99%. Se detectaron valores atípicos en la comparación entre horarios programados y reales, correspondientes a retrasos operativos extremos, los cuales se consideran fenómenos reales y no errores del cálculo.

Conclusión

> El modelo está listo para su uso operativo, respaldado por un proceso ETL confiable y KPIs validados. Se recomienda analizar los valores atípicos identificados y continuar con la automatización de cargas para garantizar actualizaciones continuas.

Calidad y Completitud del Modelo

Descripción de la Fuente de Datos

Los datos provienen del portal oficial del Bureau of Transportation Statistics (BTS), específicamente del dataset TranStats – On-Time Performance (2011–2021).
La información fue integrada en un modelo estrella en SQL Server, compuesto por la tabla de hechos fact_flights y las dimensiones dim_airline, dim_airport, dim_fecha y dim_causa_retraso.

Calidad de los Datos (Input)

Durante el proceso ETL se garantiza la **integridad y consistencia** de los registros.

- Los campos de demora contenían valores nulos en 85% de los casos, reemplazados por 0 al representar vuelos sin retraso.

- Los campos de tiempo con valores fuera del rango 1–2400 fueron eliminados (Aprox 0.9% del total).

- Se comprobó que todas las claves foráneas tienen correspondencia con sus dimensiones, asegurando integridad referencial.

Completitud (Output):

El modelo cubre el período **2011–2021** e incluye la totalidad de registros válidos después del proceso de limpieza (Aprox. 99% del total original). Esto garantiza cobertura temporal y geográfica completa, permitiendo un análisis preciso de puntualidad, cancelaciones y causas de retraso.

Coherencia entre KPIs del Dashboard y Datos Reales

Método de Validación:

La coherencia de los KPIs fue verificada mediante consultas SQL directas sobre la tabla fact_flights, replicando la lógica de cálculo aplicada en el dashboard.

Revisión por Indicador:

- **% Puntualidad (Éxito):** Los resultados son coherentes. Se observó una ligera variación en algunos aeropuertos por **salidas antes del horario programado**, lo cual no afecta la validez del KPI.

- **Distribución de Retrasos (Éxito):** El KPI refleja correctamente las principales causas de demora, destacando **Delay** y como factores dominantes.

- **Total de Cancelaciones y Desvíos (Éxito):** Los valores coinciden con las consultas de conteo simples (CANCELLED = 1, DIVERTED = 1).

- **Comparación Programado vs. Real (Atípico):** Se identificaron outliers con demoras de varias horas. No se trata de errores de cálculo, sino de eventos reales que requieren análisis separado para evitar distorsionar los promedios.

Comparación vs. Objetivos del Business Understanding

Objetivo de Negocio 1 – Medir la eficiencia operacional
Validado mediante los KPIs de % Puntualidad y Distribución de Retrasos, que permiten medir el desempeño de aerolíneas y aeropuertos con base en tiempos reales.

Objetivo de Negocio 2 – Identificar oportunidades de mejora
Cumplido mediante el Ranking de Aerolíneas y el análisis de causas de demora, que

señalan las áreas donde una intervención operativa podría reducir significativamente los retrasos.

Conclusión sobre el Negocio
Los KPIs seleccionados responden directamente a las preguntas de negocio definidas, evidenciando un alineamiento completo entre el análisis técnico y los objetivos estratégicos del proyecto.

Validación Cruzada por Roles (Cross-Checking)

Roles Involucrados:

- Data Administrator: Verificó la calidad, limpieza y consistencia del modelo.
- Data Engineer: Validó el proceso ETL y la carga en SQL Server.
- BI Analyst: Revisión del dashboard y correspondencia con los datos fuente.
- Data Analyst: Comprobó coherencia estadística y análisis de outliers.

Proceso:
El BI Analyst validó el ranking de aerolíneas y puntualidad, confirmando que los resultados reflejan la realidad operativa (ejemplo: Southwest Airlines Co. como aerolínea con mayor volumen de vuelos).

El Data Engineer y el Data Administrator corroboraron que las métricas del dashboard coinciden con las consultas directas, garantizando trazabilidad completa.

Conclusión y Próximos Pasos

Conclusión:

- El modelo analítico implementado demuestra alta fiabilidad y consistencia, cumpliendo los objetivos de negocio y técnicos del proyecto.
- Los KPIs validados reflejan adecuadamente la realidad operacional del transporte aéreo entre 2011 y 2021.

Matriz de indicadores validados

Indicador	Descripción	Método de Validación	Resultado	Comentarios
% Puntualidad	Porcentaje de vuelos cuya hora de llegada **ARR_DELAY** y salida **DEP_DELAY** superan los 0 minutos, agrupados por aerolínea.	Consulta directa SQL sobre la tabla **fact_flights**.	Exitoso	Los resultados coinciden con los valores mostrados en el dashboard. Se observa que algunos aeropuertos presentan salidas antes del horario programado, lo que explica ligeras variaciones negativas en los retrasos.
Distribución de retrasos	Análisis de la distribución de minutos de demora por causa: **CarrierDelay, WeatherDelay, NASDelay, SecurityDelay y LateAircraftDelay.**	Consulta directa SQL agregada por tipo de causa.	Exitoso	Los resultados concuerdan con el dashboard. Las mayores demoras se atribuyen a factores operativos de la aerolínea y demoras por aeronaves provenientes de vuelos anteriores.
Comparación programado vs. real	Porcentaje de vuelos cuya hora de llegada **ARR_DELAY** y salida **DEP_DELAY** superan los 0 minutos, agrupados por aeropuerto.	Conteo y comparación directa de registros con **ARR_DELAY** > 0 y **DEP_DELAY** > 0.	Atípico	Se identificaron valores extremos con retrasos de varias horas, atribuibles a condiciones operativas o climatológicas.
Ranking de aerolíneas	Ordenamiento de aerolíneas según el volumen de vuelos registrados y su desempeño promedio en puntualidad.	Agregación por tipo de demora y conteo total de vuelos.	Exitoso	Los resultados reflejan correctamente el ranking, destacando a Southwest Airlines Co. como la aerolínea con mayor número de operaciones y posicionamiento principal en el análisis.
Total de cancelaciones y desvíos	Conteo de vuelos cancelados (**CANCELLED** = 1) y desviados (**DIVERTED** = 1) según los registros del modelo.	Conteo simple y validación de totales.	Exitoso	Los valores del dashboard coinciden con los obtenidos mediante consulta directa. No se identificaron inconsistencias entre las métricas agregadas y los datos fuente.

Lista de mejoras futuras

1. Implementar filtros para outliers en KPIs
   Aplicar reglas o técnicas estadísticas para excluir valores extremos en retrasos, evitando que distorsionen promedios y análisis.

2. Automatización completa del proceso ETL
   Configurar cargas automáticas y programadas para garantizar actualizaciones continuas sin intervención manual.

3. Integración de datos meteorológicos externos
   Incorporar información climática para correlacionar retrasos con condiciones meteorológicas y mejorar análisis causal.

4. Incluir datos de tráfico aéreo y control de espacio aéreo
   Agregar información sobre congestión en aeropuertos y rutas para explicar demoras por NASDelay.

5. Desarrollo de alertas proactivas en el dashboard
   Configurar notificaciones automáticas cuando se detecten patrones críticos, como aumento de cancelaciones o retrasos extremos.

6. Optimización del modelo estrella para consultas avanzadas
   Crear índices adicionales y vistas materializadas para mejorar el rendimiento en análisis complejos.

7. Incorporar análisis predictivo con Machine Learning
   Desarrollar modelos que anticipen retrasos y cancelaciones basados en patrones históricos y variables externas.

8. Mejorar la visualización de KPIs en el dashboard
   Añadir gráficos interactivos, mapas de calor y segmentación por región, aerolínea y temporada.

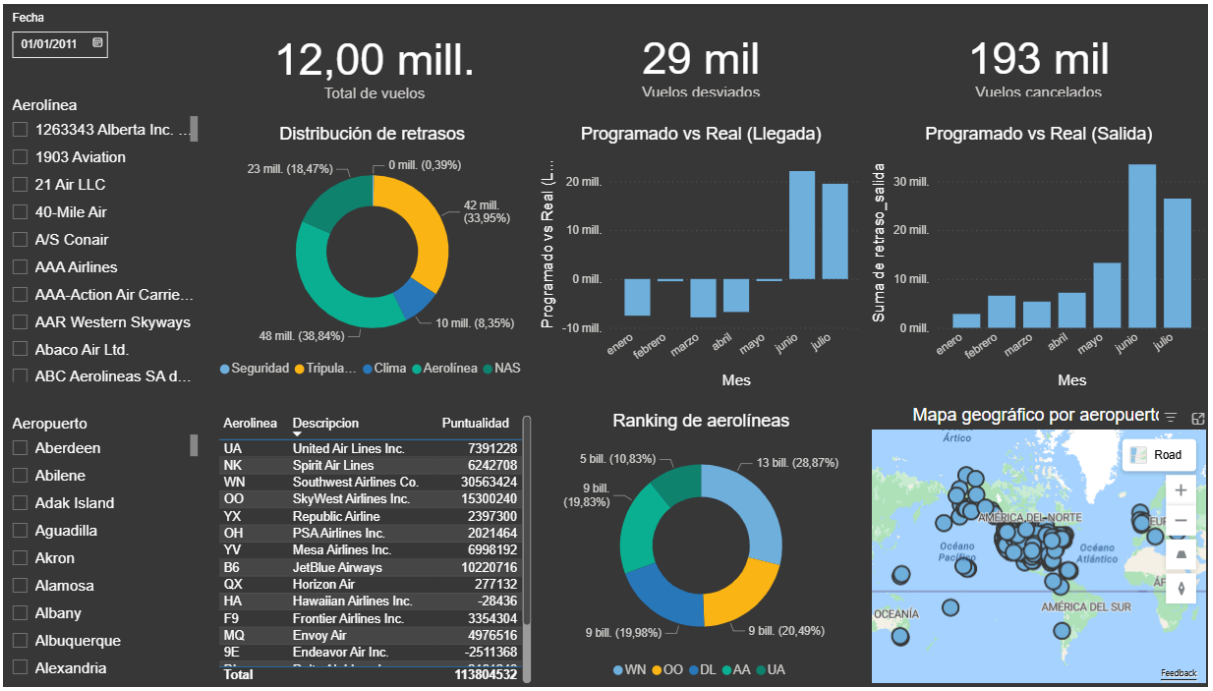
9. Implementar control de calidad automatizado en cada carga
   Validar integridad, consistencia y rangos de datos en tiempo real para reducir errores en el pipeline.

10. Ampliar el período de análisis con datos más recientes
    Incluir registros posteriores a 2021 para mantener el modelo actualizado y relevante para decisiones actuales.

6. Deployment

Dashboard compartido

https://app.powerbi.com/view?r=eyJrIjoiOTljZWRhNGItYmM4ZS00OTZkLWEwMjktN2M0NzQzMTUxYWY4IiwidCI6IjIzYjVhMmVmLTM0OTYtNGEwYy04Y2ExLWI1ODM3OWI3YTQ0YyIsImMiOjR9



Repositorio Github del proyecto

https://github.com/LuisCulajay/IntroduccionAnalisisDatos_ProyectoFinal

Manual técnico de instalación

https://docs.google.com/document/d/1CJC5SJ5SyyAWOjb23v0sc_3Z9wZEbqba8axuZOl87YU/edit?usp=sharing