UNIVERSIDAD TECNOLÓGICA DE CHIHUAHUA

Tecnologías de la Información: Desarrollo y Gestión de Software

II.3. Reporte de limpieza de datos

IDGS91N - Kevin Iván Aguirre Silva (1123250020)
Extracción de Conocimiento en Bases de Datos - Ing. Luis Enrique Mascote Cano

Chihuahua, Chih., 12 de octubre de 2025

ÍNDICE

1. INTRODUCCIÓN

El análisis y gestión eficiente de la información constituye un elemento esencial en la toma de decisiones dentro de cualquier organización o entidad gubernamental. En el ámbito de la migración internacional, disponer de datos confiables y estructurados permite comprender mejor los flujos de personas, las tendencias por nacionalidad, los tipos de visa y las dinámicas entre países de origen y destino.

El presente trabajo tiene como objetivo procesar, limpiar y modelar el conjunto de datos “International Migration – March 2021”, el cual contiene estimaciones de movimientos migratorios clasificados por ciudadanía, tipo de visa y país de última residencia. A través del uso del lenguaje Python y la librería Pandas, se realizaron tareas de limpieza de datos para eliminar duplicados, estandarizar formatos y preparar el archivo para su posterior análisis.

Posteriormente, se desarrolló un modelo de datos orientado a la construcción de un Data Warehouse, definiendo una tabla de hechos centrada en las estimaciones de migración y diversas tablas de dimensiones que aportan contexto temporal, geográfico y categórico. Este modelo fue complementado con un script SQL que especifica la estructura del sistema, sus relaciones y claves primarias y foráneas.

En conjunto, este proceso permitió transformar un conjunto de datos plano en una base analítica organizada, capaz de soportar consultas multidimensionales, generar reportes estratégicos y servir como punto de partida para proyectos de inteligencia de negocios (BI) o análisis predictivo sobre fenómenos migratorios.

1.1. Objetivos

1.1.1. Objetivo general:

Diseñar e implementar un modelo de datos tipo Data Warehouse que permita analizar la información de migración internacional de manera estructurada, confiable y escalable.

1.1.2. Objetivos específicos:

- Realizar un proceso de limpieza y depuración de datos utilizando Python y la librería Pandas.
- Identificar las tablas de hechos y dimensiones que conforman el modelo analítico.
- Desarrollar un modelo de datos en esquema estrella, adecuado para el análisis multidimensional.
- Elaborar un script SQL que defina la estructura de las tablas, sus claves primarias, foráneas y relaciones.

Establecer las bases para la integración del modelo en herramientas de inteligencia de negocios (BI) y visualización de datos.

2. LIMPIEZA DE DATOS

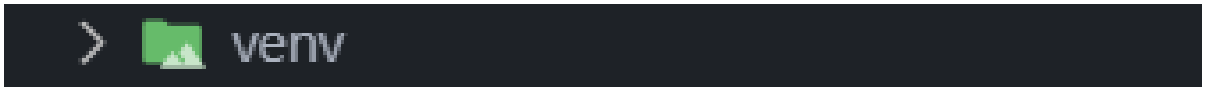
2.1. Configuración inicial del proyecto

2.1.1. Creación y activación de entorno virtual

La limpieza de datos será realizada con el lenguaje de programación Python en conjunto de la librería Pandas. Lo primero que se debe realizar antes que nada es la correcta configuración del proyecto mediante la creación de un entorno virtual, de esta manera al instalar la librería esta no se instala de manera global en la PC si no que sólo será contenida en la carpeta raíz del proyecto. Para ello se utiliza el siguiente comando:

```
python -m venv venv
```

Como resultado se crea una carpeta llamada venv en la raíz del proyecto:



El siguiente paso es ejecutar el siguiente comando (en el caso de Windows), este lo que hará es activar el entorno virtual recién creado:

```
venv\Scripts\activate
```

Una vez activado si se utiliza VisualStudio Code se tendrá que seleccionar el intérprete para Python, escribiendo en la barra de búsqueda *>Python: Select Interpreter*.

Y como último paso seleccionar el que dice *(venv)*.

2.1.2. Instalación de Pandas

Una vez realizados los pasos anteriores ya es posible instalar librerías en el proyecto dentro del entorno virtual. Para instalar Pandas se utiliza el comando:

```
pip install pandas
```

Esos serían todos los pasos para la correcta configuración del proyecto, de esta manera ya es posible comenzar a trabajar con la librería y desarrollar el código necesario para la limpieza de datos.

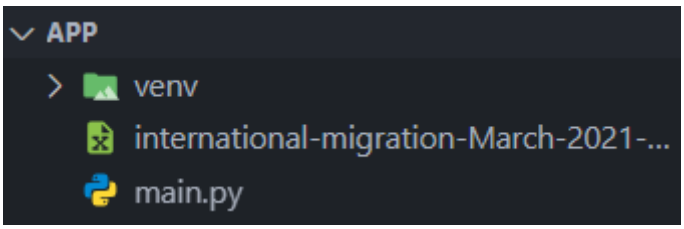
2.2. Identificación de valores faltantes, inconsistencias de formato y duplicados

2.2.1. Importación de Pandas y carga de archivo CSV

La primera línea de código del script importa la librería bajo el alías de pd para escribir código de manera más simple sin tener que llamar al nombre completo de la librería cada vez que se utiliza.

```
1   import pandas as pd
```

Para cargar el archivo CSV es necesario que éste se encuentre dentro de la carpeta raíz donde se creó el entorno virtual, esto para que sea más fácil ubicarlo.



Ahora mediante las siguientes líneas de código se reemplaza el nombre del archivo por la ruta correcta si es necesario y se lee dicho archivo.

```
5    file_path =
     "international-migration-March-2021-citizenship-by-visa-by-country-of-last-permanen
     t-residence.csv"
6    df = pd.read_csv(file_path)
```

```
13   print("Valores faltantes por columna:")
14   print(df.isnull().sum(), "\n")
```

2.2.2. Identificación de valores faltantes

- **df.isnull()** - Crea una máscara booleana del DataFrame donde True indica valores nulos/missing.
- **.sum()** - Agrega por columnas, contando True (valores nulos) como 1 y False como 0.

```
Valores faltantes por columna:
year_month              0
month_of_release        0
passenger_type          0
direction               0
citizenship             0
visa                    0
country_of_residence    0
estimate                0
standard_error          0
status                  0
dtype: int64
```

2.2.3. Trata de valores faltantes

```
18   df = df.dropna(how='all')
```

- **dropna()** - Remueve filas/columnas con valores faltantes.

- **how='all'** - Parámetro que especifica que solo elimina filas donde todas las columnas son nulas.
- **df =** - Reasigna el resultado al DataFrame original (modificación in-place).

2.2.4. Normalización de formatos

```
df.columns = df.columns.str.strip()
df.columns = df.columns.str.lower()
```

- Se eliminan espacios en blanco en nombres de columnas.
- Se convierten a minúsculas los nombres de columnas.

2.2.5. Detección y eliminación duplicados

```
duplicates = df.duplicated()
print(f"Número de filas duplicadas: {duplicates.sum()}")
df = df.drop_duplicates()
```

1. Se crea una Serie booleana donde True indica filas duplicadas (excepto la primera ocurrencia)
2. Se cuenta y muestra el total de filas duplicadas (suma los True)
3. Se eliminan las filas duplicadas, manteniendo solo la primera ocurrencia de cada conjunto duplicado.

3. HECHOS Y DIMENSIONES

A partir del conjunto de datos “International Migration – March 2021 (Citizenship by Visa by Country of Last Permanent Residence)”, se identificaron las entidades necesarias para el diseño de un modelo de datos orientado a un Data Warehouse.
El objetivo principal de este modelo es permitir el análisis de los flujos migratorios internacionales según la nacionalidad, tipo de visa, país de residencia anterior, dirección del movimiento y periodo temporal.

3.1. Tabla de Hechos

3.1.1. Hechos_Migracion

- La tabla concentra las medidas cuantitativas del sistema, correspondientes a las estimaciones de migración internacional.
- Cada registro representa un evento de migración clasificado por mes, ciudadanía, tipo de visa, país de residencia y dirección del flujo (arribo o salida).

- Contiene los indicadores estimate (estimación del número de migrantes) y standard_error (error estándar de la estimación).
- La tabla se relaciona con las dimensiones de tiempo, país, ciudadanía, tipo de visa, tipo de pasajero y dirección del movimiento.

3.2. Tablas de Dimensiones

3.2.1. Dim_Tiempo

- Describe la dimensión temporal de los eventos migratorios. Permite analizar la información por año y mes, así como identificar la fecha de publicación o actualización de los datos.
- Campos relevantes: year_month, month_of_release.

3.2.2. Dim_Pais

- Representa los países de residencia anterior de los migrantes. Facilita el análisis geográfico de los movimientos.
- Campo principal: country_of_residence.

3.2.3. Dim_Ciudadania

- Contiene la información relacionada con la nacionalidad o ciudadanía de las personas migrantes, permitiendo diferenciar entre neozelandeses y no neozelandeses.
- Campo principal: citizenship.

3.2.4. Dim_Visa

- Describe los tipos de visa o permisos migratorios asociados a cada movimiento (por ejemplo, residente, visitante, ciudadano NZ/Australia).
- Campo principal: visa.

3.2.5. Dim_Pasajero

- Clasifica los movimientos según el tipo de pasajero, como migrantes de largo plazo o corto plazo.

- Campo principal: passenger_type.

3.2.6. Dim_Direccion

- Indica la dirección del flujo migratorio, especificando si se trata de llegadas (Arrivals) o salidas (Departures).
- Campo principal: direction.

4. MODELO RELACIONAL

4.1. Tablas

4.1.1. Tiempo

```sql
CREATE TABLE Dim_Tiempo (
    id_tiempo SERIAL PRIMARY KEY,
    year_month VARCHAR(7) NOT NULL,
    month_of_release VARCHAR(10) NULL
);
```

4.1.2. País

```sql
CREATE TABLE Dim_Pais (
    id_pais SERIAL PRIMARY KEY,
    country_of_residence VARCHAR(100) NOT NULL
);
```

4.1.3. Ciudadanía

```sql
CREATE TABLE Dim_Ciudadania (
    id_ciudadania SERIAL PRIMARY KEY,
    citizenship VARCHAR(50) NOT NULL
);
```

4.1.4. Tipo de visa

```sql
CREATE TABLE Dim_Visa (
    id_visa SERIAL PRIMARY KEY,
    visa VARCHAR(100) NOT NULL
);
```

4.1.5. Pasajero

```sql
CREATE TABLE Dim_Pasajero (
    id_tipo_pasajero SERIAL PRIMARY KEY,
    passenger_type VARCHAR(100) NOT NULL
);
```

4.1.6. Dirección

```sql
CREATE TABLE Dim_Direccion (
    id_direccion SERIAL PRIMARY KEY,
    direction VARCHAR(20) NOT NULL  -- Ej: 'Arrivals', 'Departures'
);
```

4.1.7. Hechos

```sql
CREATE TABLE Hechos_Migracion (
    id_hecho SERIAL PRIMARY KEY,

    -- Claves foráneas hacia las dimensiones
    id_tiempo INT NOT NULL,
    id_pais INT NOT NULL,
    id_ciudadania INT NOT NULL,
    id_visa INT NOT NULL,
    id_tipo_pasajero INT NOT NULL,
    id_direccion INT NOT NULL,

    -- Medidas
    estimate INT NOT NULL,
    standard_error INT,
    status VARCHAR(50),

    -- Relaciones
    FOREIGN KEY (id_tiempo) REFERENCES Dim_Tiempo(id_tiempo),
    FOREIGN KEY (id_pais) REFERENCES Dim_Pais(id_pais),
    FOREIGN KEY (id_ciudadania) REFERENCES Dim_Ciudadania(id_ciudadania),
    FOREIGN KEY (id_visa) REFERENCES Dim_Visa(id_visa),
    FOREIGN KEY (id_tipo_pasajero) REFERENCES Dim_Pasajero(id_tipo_pasajero),
    FOREIGN KEY (id_direccion) REFERENCES Dim_Direccion(id_direccion)
);
```

5. CONCLUSIONES

El proceso de análisis y modelado de datos permitió comprender la estructura y el contenido del conjunto “International Migration – March 2021”. A partir de la exploración inicial, se determinó que los datos se encontraban en buen estado, sin valores nulos ni duplicados, lo que facilitó su posterior procesamiento. No obstante, fue necesario realizar tareas de estandarización de nombres y formatos para asegurar la consistencia del dataset.

El diseño del modelo de datos en esquema estrella permitió organizar la información en una tabla de hechos (Hechos_Migracion) y diversas tablas de dimensiones (Tiempo, País, Ciudadanía, Visa, Pasajero y Dirección).
Esta estructura mejora la eficiencia de las consultas analíticas y facilita la creación de reportes dinámicos orientados a la toma de decisiones sobre flujos migratorios.

El resultado final constituye una base sólida para implementar un Data Warehouse enfocado en la migración internacional, con potencial de integrarse a herramientas de Business Intelligence o visualización interactiva.

5.1. Aprendizajes

Durante el desarrollo de este proyecto se reforzaron conocimientos en:

- Limpieza y preparación de datos utilizando Python y la librería Pandas, abordando detección de valores faltantes, normalización de formatos y eliminación de duplicados.
- Diseño lógico de modelos analíticos, diferenciando claramente entre datos de hechos y dimensiones.
- Uso de SQL para definir estructuras de datos normalizadas, incluyendo claves primarias, foráneas y vistas consolidadas.
- Comprensión del valor que tiene un modelo bien estructurado para mejorar el rendimiento y la interpretabilidad de la información en entornos analíticos.

5.2. Recomendaciones

- Automatizar el proceso de carga y actualización de datos, integrando los scripts de limpieza y carga en un flujo ETL (Extract, Transform, Load).
- Ampliar el modelo incorporando nuevas dimensiones, como género, edad o motivo de migración, en caso de disponer de datos adicionales.
- Implementar controles de calidad de datos que verifiquen la consistencia de los registros antes de su inserción en el Data Warehouse.
- Utilizar herramientas de visualización (Power BI, Tableau o Metabase) para generar paneles interactivos que permitan explorar las tendencias migratorias de forma intuitiva.
- Documentar las transformaciones realizadas, manteniendo un registro reproducible de las etapas del proceso para garantizar trazabilidad y transparencia analítica.

6. REFERENCIAS

Torres, A. (9 de diciembre de 2021). *Limpieza de datos en Pandas: Explicado con ejemplos*. freeCodeCamp.org. https://www.freecodecamp.org/espanol/news/limpieza-de-datos-en-pandas-explicado-con-ejemplos/

Daniel. (30 de octubre de 2023). *Data Warehouse: ¿qué es y cómo utilizarlo?* DataScientest. https://datascientest.com/es/data-warehouse-que-es-y-como-utilizarlo

MacDonald, L. (16 de septiembre de 2025). *Fact vs. dimension tables explained*. Monte Carlo Data. https://www.montecarlodata.com/blog-fact-vs-dimension-tables-in-data-warehousing-explained/

pandas - Python Data Analysis Library. (s. f.). https://pandas.pydata.org/

12. Entornos virtuales y paquetes. (s. f.). Python Documentation. https://docs.python.org/es/3.13/tutorial/venv.html