iimas

Universidad Nacional Autónoma de México

Instituto de Investigaciones en Matemáticas Aplicadas y en Sistemas

Licenciatura en Ciencia de Datos

Bases de Datos Estructuradas

Proyecto final

Alumnos:
Aguilar Martinez Erick Yair
Ahuatzi Pichardo Mariano Josué
Martinez Muñoz Alan Magno
Mendoza Hernandez Carlos Emiliano
Semestre: 2025-1
Fecha: 22/11/2024
Profesor: Ing. Fernando Arreola

Índice

1 Introducción

Como parte de nuestra formación en la materia de **Bases de Datos Estructuradas**, desarrollaremos el presente proyecto, cuyo propósito es aplicar los conocimientos adquiridos en el curso para resolver un problema real relacionado con la gestión y análisis de datos climáticos en México.

El Gobierno de México, como parte de su política pública, ofrece acceso a información relativa al clima del país a través de un servicio web. No obstante, la estructura y formato de estos datos no están diseñados para facilitar un análisis profundo o automatizado, lo que representa un desafío importante en términos de estandarización y gestión eficiente de la información.

Nuestro objetivo principal es **extraer, transformar y cargar** los datos proporcionados por este servicio web, con el fin de automatizar y estandarizar el flujo de información. Esto permitirá convertir los datos crudos en información estructurada y procesable, facilitando su uso para análisis meteorológicos, elaboración de reportes y aplicaciones basadas en datos climáticos.

Aspectos clave del análisis preliminar

1. Periodicidad de la información

- Los datos proporcionados tienen una periodicidad específica; es decir, se actualizan cada cierto intervalo de tiempo. En este caso, se reciben temperaturas para cada municipio desde el momento de la consulta hasta tres días en el futuro.
- Un desafío importante es que los datos históricos no son recuperables desde el propio servicio web una vez que han sido reemplazados, lo que subraya la necesidad de un sistema que capture y almacene los datos a medida que se generan.

2. Estructura de la información

- Los datos se presentan en un formato estándar, `JSON`, lo que facilita su procesamiento.
- Sin embargo, estos datos están comprimidos para optimizar la transmisión por la red, lo que implica una etapa adicional de descompresión durante el proceso ETL.

Con base en estas características y aprovechando los conocimientos adquiridos durante el curso, hemos identificado que un proceso **ETL** es la solución más adecuada para este problema. Este enfoque nos permitirá gestionar la información de manera eficiente y resolver los retos asociados al análisis y almacenamiento de los datos climáticos.

Impacto esperado

Esta solución permitirá no solo automatizar la gestión de los datos climáticos, sino también facilitar la toma de decisiones y desarrollar herramientas que utilicen esta información como insumo. Con ello, nuestro proyecto busca contribuir al uso más eficiente y efectivo de los datos proporcionados por el Gobierno de México.

2 Plan de trabajo

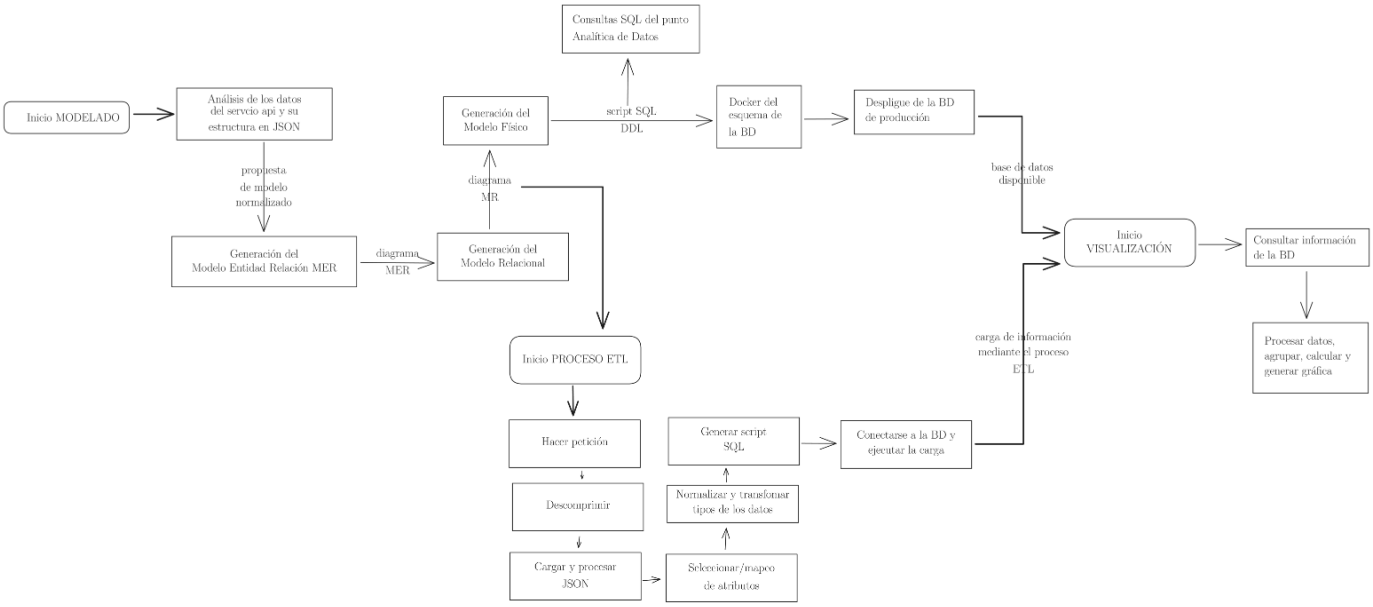


Figure 1: Diagrama de flujo general del proyecto.

2.1 Descripción General de las Actividades

El proyecto consiste en el desarrollo de un sistema de análisis y procesamiento de información meteorológica. Las actividades están organizadas en las siguientes fases:

1. **Análisis de Información**
   - Estudiar la documentación de la API del Servicio Meteorológico Nacional (SMN).
   - Examinar el formato de datos en JSON y sus atributos principales.
   - Generar preguntas clave sobre la información y planificar las respuestas a través de consultas SQL y visualizaciones.
2. **Modelado de Datos**
   - Diseñar un modelo de datos que comprenda:
     - Modelo Entidad-Relación (MER).
     - Modelo Relacional.
     - Modelo Físico (SQL).
   - Normalizar la estructura de datos para su correcto manejo en bases de datos relacionales.
3. **Procesamiento de Datos (ETL)**
   - Configurar un flujo ETL utilizando Pentaho que permita:
     - Descargar datos en formato JSON mediante una petición GET.
     - Transformar el formato JSON en uno estructurado.
     - Normalizar atributos y asignar tipos de datos correctos.
     - Cargar los datos procesados en las tablas de la base de datos.

4. **Analítica de Datos**
    - Resolver consultas analíticas, tales como:
        - Municipios con los mayores descensos de temperatura.
        - Cuarto municipio con la temperatura más alta por estado.
        - Plantear un caso de estudio que aporte valor a una problemática real.
    - Justificar ventajas y desventajas del formato de datos inicial.
5. **Visualización**
    - Crear un dashboard interactivo que permita:
        - Visualizar temperaturas máximas por estado y fecha.
        - Filtrar por estado y fecha seleccionados.
        - Reflejar actualizaciones en la base de datos automáticamente.
6. **Documentación**
    - Redactar un documento en LaTeXque incluya:
        - Introducción al proyecto.
        - Análisis de datos.
        - Diseño de la base de datos.
        - Descripción del flujo ETL.
        - Explicación de las consultas analíticas.
        - Detalles del dashboard.
        - Conclusiones y recomendaciones.

2.2 Roles

2.2.1 Erick

- **Modelado de Datos**: Diseño del modelo Entidad-Relación (MER), modelo relacional y modelo físico, asegurando la normalización de los datos y su representación estructurada.
- **Configuración de la Base de Datos Remota**: Creación de un contenedor Docker y configuración de la base de datos remota para asegurar la accesibilidad y disponibilidad de los datos procesados.
- **Desarrollo del Dashboard Interactivo**: Implementación de un sistema visual que consume datos de la base de datos, permitiendo filtrar por municipio y estado y fecha y reflejando actualizaciones en tiempo real.

2.2.2 Emiliano

- **Desarrollo del Flujo ETL en Pentaho**: Configuración de un proceso ETL para descargar datos del servicio API, transformar su formato JSON en datos estructurados, realizar normalizaciones necesarias y cargarlos en la base de datos diseñada.

2.2.3 Magno

- **Diseño y Ejecución de Consultas SQL Analíticas**: Creación de consultas que respondan a preguntas clave, identificando patrones y comportamientos en los datos meteorológicos.
- **Elaboración de la Documentación en LaTeX**: Redacción del reporte final en formato LaTeX, consolidando las actividades realizadas, los resultados obtenidos y las conclusiones del proyecto.

2.2.4 Mariano

- **Análisis de Datos y Definición de Preguntas Clave**: Estudio detallado de los datos del servicio API para identificar áreas de interés, definir preguntas clave y planificar las respuestas mediante consultas y visualizaciones.
- **Creación y Redacción de la Presentación**: Desarrollo de una presentación formal que detalla el proceso, los resultados y el impacto del proyecto, destinada a su exposición.

3 Análisis

El análisis de los datos meteorológicos se llevó a cabo utilizando la librería `pandas` en Python para procesar el archivo JSON obtenido del servicio web. A continuación, se presentan las actividades realizadas y los resultados obtenidos:

3.1 Cargar y transformar los datos

```python
import pandas as pd
import numpy as np

clima_df = pd.read_json('./dato_servicio_web.json')
clima_df['dloc'] = pd.to_datetime(clima_df['dloc'])

# Validación del tipo de datos de las variables
clima_df.info()
clima_df.head()
```

El resultado muestra que las variables se distribuyen de la siguiente manera:

- Variables categóricas: `ides`, `idmun`, `nes`, `nmun`, etc.
- Variables numéricas: `lat`, `lon`, valores predictivos, entre otras.
- Fecha: `dloc`, convertida al tipo `datetime`.

3.2 Definición de funciones para evaluar dependencias funcionales

```python
def validar_dependencia_funcional(df, columnas_izquierda, n_variables):
    valores_previos = {}
    for index, row in df.iterrows():
        key = tuple(row[col] for col in columnas_izquierda)
        if key not in valores_previos:
            valores_previos[key] = {var: {'v': row[var], 'i': [index]} for var in n_variables}
        else:
            for var in n_variables:
                if valores_previos[key][var]['v'] != row[var]:
                    valores_previos[key][var]['i'].append(index)
    return valores_previos

def hayDependenciaFuncional(valores_previos, df, columnas_izquierda, n_variables, log=True):
    for _, value in valores_previos.items():
        for var in n_variables:
            if len(value[var]['i']) > 1:
                if log:
                    idx1, idx2 = value[var]['i'][0], value[var]['i'][1]
```

```
                    print(f"Filas {idx1} y {idx2} no cumplen con la dependencia funcional para la variable {var}:")
                    print(df.loc[idx1, columnas_izquierda + n_variables])
                    print(df.loc[idx2, columnas_izquierda + n_variables])
                return False
    return True
```

3.3 Validación de dependencias funcionales

```
cols_izquierda = ['ides']
cols_derecha = ['nes']
valores_previos = validar_dependencia_funcional(clima_df, cols_izquierda, cols_derecha)
hayDP = hayDependenciaFuncional(valores_previos, clima_df, cols_izquierda, cols_derecha, log=False)
if hayDP:
    print(f"({','.join(cols_izquierda)}) -> ({','.join(cols_derecha)})")

cols_izquierda = ['ides']
cols_derecha = ['nes','dh']
valores_previos = validar_dependencia_funcional(clima_df, cols_izquierda, cols_derecha)
hayDP = hayDependenciaFuncional(valores_previos, clima_df, cols_izquierda, cols_derecha, log=True)
if hayDP:
    print(f"({','.join(cols_izquierda)}) -> ({','.join(cols_derecha)})")
else:
    print(f"({','.join(cols_izquierda)}) no determina a ({','.join(cols_derecha)})")

cols_izquierda = ['ides','idmun']
cols_derecha = ['lat','lon','nmun','dh']
valores_previos = validar_dependencia_funcional(clima_df, cols_izquierda, cols_derecha)
hayDP = hayDependenciaFuncional(valores_previos, clima_df, cols_izquierda, cols_derecha, log=False)
if hayDP:
    print(f"({','.join(cols_izquierda)}) -> ({','.join(cols_derecha)})")
```

Se implementaron funciones para verificar la existencia de dependencias funcionales entre ciertas columnas clave. Las dependencias evaluadas fueron:

1. `ides` $\rightarrow$ `nes`:
    - Resultado: Cumple la dependencia funcional.
2. `ides` $\rightarrow$ `nes, dh`:
    - Resultado: No cumple completamente la dependencia funcional. Se encontraron conflictos en las filas analizadas.
3. `ides, idmun` $\rightarrow$ `lat, lon, nmun, dh`:
    - Resultado: Cumple la dependencia funcional.

3.4 Grupo y valores únicos

Las columnas relacionadas exclusivamente con las predicciones meteorológicas se identificaron como:

```python
v, c = np.unique(clima_df['dloc'], return_counts=True)
print(len(v))
print(v)
x = clima_df.groupby(by=['ides','idmun','lat','lon','nmun','dh']).size().
    reset_index(name='count')
print(x)

# Datos de la entidad predicción
datos = list(set(clima_df.columns) - set(['dloc', 'ides', 'idmun', 'lat', 'lon',
    'nmun', 'dh', 'nes']))
print(datos)
```

De los anteriores resultados se puede decir que hay una medición por cada municipio al dia. Por consiguiente, hay cuatro registros que corresponden al dia actual, al siguiente, dos y tres días después

3.5 Validación de restricciones

```python
# Validación de constrains, valores negativos, cotas máximas, precisión de los
    valores numéricos
clima_df.describe()
```

Se realizó una exploración estadística de las variables numéricas para identificar posibles valores negativos, límites máximos y precisión en los valores numéricos:

3.6 Resultados y Análisis

- La estructura de los datos permite identificar relaciones clave, como la dependencia funcional entre `ides` y las entidades asociadas al municipio.
- Cada municipio cuenta con registros correspondientes a cuatro días consecutivos.
- Los datos meteorológicos, como temperaturas máximas y mínimas, están listos para ser utilizados en consultas analíticas y visualizaciones.

4 Diseño

4.1 Diseño del Modelo Entidad-Relación (MER)

Se identifican tres entidades principales: `Estado`, `Municipio` y `Predicción`, junto con sus relaciones y atributos. A continuación, se desglosan los componentes del modelo:

4.2 Entidades

- **Estado**: Contiene información sobre los estados donde están ubicados los municipios.
- **Municipio**: Almacena los detalles de cada municipio, incluyendo su ubicación geográfica, como latitud, longitud y nombre.
- **Predicción**: Registra datos climáticos y predicciones meteorológicas para cada municipio, incluyendo temperaturas máximas y mínimas, probabilidad de precipitación y otras variables relevantes.

4.3 Relaciones

- **ESTA**: Relación uno a muchos entre `Estado` y `Municipio`, donde un estado puede contener múltiples municipios.
- **TIENE**: Relación uno a muchos entre `Municipio` y `Predicción`, donde cada municipio tiene varias predicciones asociadas a diferentes fechas y tiempos.

4.4 Atributos Principales de las Entidades

- **Estado**
  - `id_estado`: Identificador único del estado.
  - `nombre_estado`: Nombre del estado.
- **Municipio**
  - `id_municipio`: Identificador único del municipio.
  - `nombre_municipio`: Nombre del municipio.
  - `latitud`: Coordenada geográfica (latitud) del municipio.
  - `longitud`: Coordenada geográfica (longitud) del municipio.
  - `id_estado`: Identificador del estado al que pertenece.
- **Predicción**
  - `id_prediccion`: Identificador único de la predicción.
  - `fecha`: Fecha de la predicción.
  - `tmin`: Temperatura mínima esperada.
  - `tmax`: Temperatura máxima esperada.
  - `precipitacion`: Probabilidad de lluvia o cantidad esperada.
  - `id_municipio`: Identificador del municipio asociado.

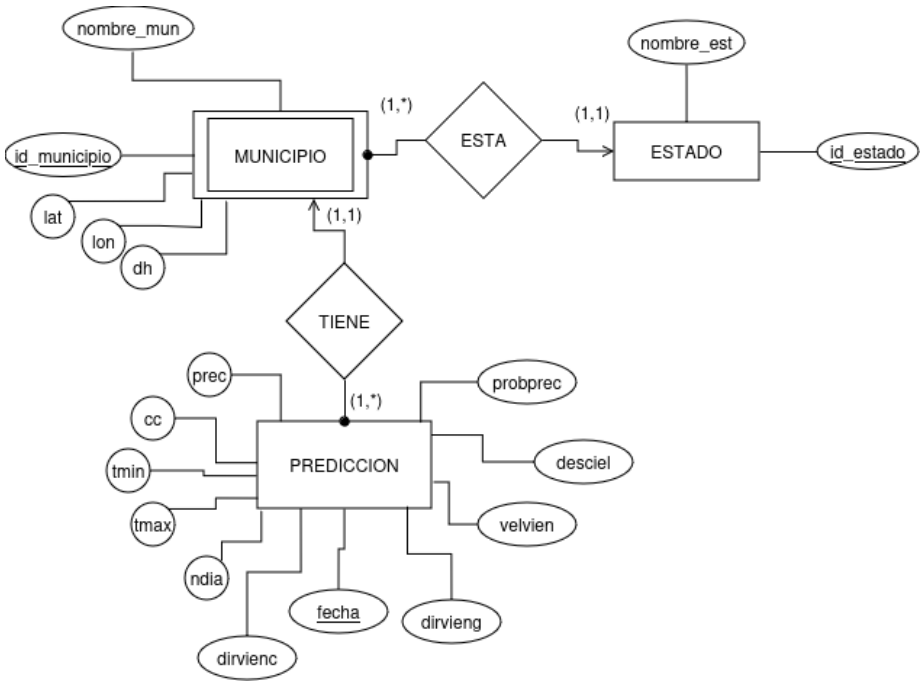


Figure 2: Modelo Entidad-Relación

4.5 Diseño del Modelo Relacional

El modelo entidad-relación fue transformado en un modelo relacional para facilitar su implementación en SQL. A continuación, se describe cada tabla del modelo junto con sus atributos y restricciones.

```
ESTADO = {
    id_estado INT PK,
    nombre_est VARCHAR(100)
}

MUNICIPIO = {
    [
        id_estado INT FK,
        id_municipio INT
    ] PK,
    nombre_mun VARCHAR(150),
    lat FLOAT,
    lon FLOAT,
    dh INT,
}


PREDICCION = {
    [
        id_estado INT FK,
        id_municipio INT FK,
        fecha TIMESTAMP
    ] PK,
    cc FLOAT C,
    desciel VARCHAR(255),
    dirvienc VARCHAR(60),
    dirvieng INT,
    ndia VARCHAR(10) C,
    prec FLOAT C,
    probprec INT C,
    tmax FLOAT,
    tmin FLOAT,
    velvien FLOAT C
}
```

4.5.1 Claves y Restricciones

- **Claves Primarias (PK)**:
  - ESTADO: `id_estado`.
  - MUNICIPIO: `[id_estado, id_municipio]`.
  - PREDICCION: `[id_estado, id_municipio, fecha]`.
- **Claves Foráneas (FK)**:
  - `MUNICIPIO.id_estado` referencia a `ESTADO.id_estado`.
  - `PREDICCION.id_estado` referencia a `ESTADO.id_estado`.
  - `PREDICCION.id_municipio` referencia a `MUNICIPIO.id_municipio`.
- **Restricciones**:
  - Los atributos marcados como `C` (constrain) deben cumplir con restricciones específicas, como no permitir valores negativos (e.g., `prec`, `probprec`, `tmin`, etc.).
  - Las claves primarias y foráneas deben garantizar unicidad e integridad referencial.

4.6 Diseño del Modelo Físico

El modelo físico implementa el modelo relacional en SQL, definiendo tipos de datos adecuados, restricciones de integridad y claves.

La siguiente figura ilustra la representación gráfica del modelo físico, destacando las relaciones y los atributos definidos.

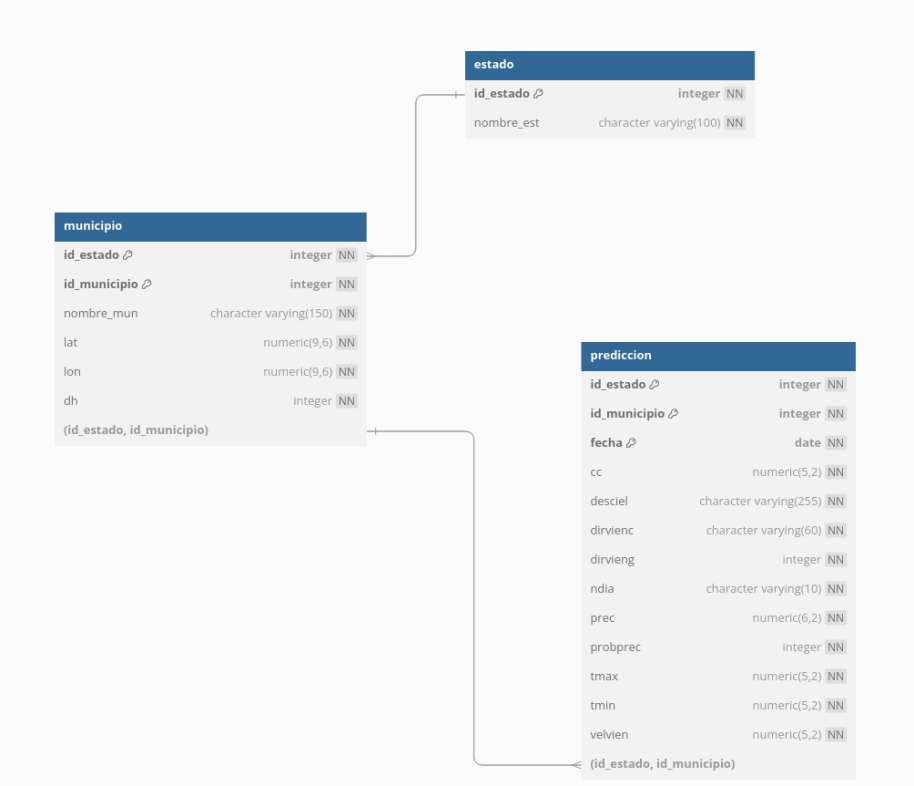


Figure 3: Modelo Físico

El modelo físico se presenta mediante la siguiente representación SQL:

```sql

CREATE TABLE estado (
    id_estado    INT,
    nombre_est   VARCHAR(100) NOT NULL,
    CONSTRAINT pk_estado PRIMARY KEY (id_estado)
);

CREATE TABLE municipio (
    id_estado      INT,
    id_municipio   INT,
    nombre_mun     VARCHAR(150)  NOT NULL,
    lat            DECIMAL(9, 6) NOT NULL,
    lon            DECIMAL(9, 6) NOT NULL,
    dh             INT           NOT NULL,
    CONSTRAINT pk_municipio PRIMARY KEY (id_estado, id_municipio),
    CONSTRAINT fk_municipio_estado FOREIGN KEY (id_estado) REFERENCES estado(
    id_estado)
        ON DELETE CASCADE
        ON UPDATE CASCADE
);

CREATE TABLE prediccion (
    id_estado      INT,
```

```
    id_municipio  INT,
    fecha         DATE,
    cc            DECIMAL(5, 2) NOT NULL,
    desciel       VARCHAR(255)  NOT NULL,
    dirvienc      VARCHAR(60)   NOT NULL,
    dirvieng      INT           NOT NULL,
    ndia          VARCHAR(10)   NOT NULL,
    prec          DECIMAL(6, 2) NOT NULL,
    probprec      INT           NOT NULL,
    tmax          DECIMAL(5, 2) NOT NULL,
    tmin          DECIMAL(5, 2) NOT NULL,
    velvien       DECIMAL(5, 2) NOT NULL,
    CONSTRAINT pk_prediccion PRIMARY KEY (id_estado, id_municipio, fecha),
    CONSTRAINT fk_prediccion_municipio FOREIGN KEY (id_estado, id_municipio)
    REFERENCES municipio(id_estado, id_municipio)
        ON DELETE CASCADE
        ON UPDATE CASCADE,
    CONSTRAINT chk_cc_positive CHECK (cc >= 0),
    CONSTRAINT chk_ndia_valid CHECK (
        ndia IN ('lunes', 'martes', 'miercoles', 'jueves', 'viernes', 'sabado', '
    domingo')
    ),
    CONSTRAINT chk_prec_positive CHECK (prec >= 0),
    CONSTRAINT chk_probprec_positive CHECK (probprec >= 0),
    CONSTRAINT chk_velvien_positive CHECK (velvien >= 0)
);
```

4.6.1 Detalles Clave

- Las tablas `estado`, `municipio` y `prediccion` están relacionadas mediante claves foráneas, asegurando la integridad referencial.
- Las restricciones `CHECK` garantizan que:
  - Los valores numéricos sean positivos (`cc`, `prec`, `probprec`, `velvien`).
  - Los días de la semana sean válidos (`ndia`).

4.7 Validación del Modelo

4.7.1 Descripción

Se realizaron pruebas iniciales para validar la integridad referencial y la funcionalidad del modelo implementado. Estas pruebas incluyeron la inserción de datos de prueba en las tablas `estado`, `municipio` y `prediccion`, asegurando que se cumplieran las restricciones definidas en el diseño físico.

4.7.2 Resultados

- **Integridad Referencial**: Las claves primarias y foráneas funcionaron correctamente, garantizando la relación entre las tablas.
- **Consistencia de Datos**: Los datos de prueba se integraron sin inconsistencias ni violaciones de las restricciones definidas, como valores negativos o claves foráneas no existentes.
- **Restricciones de Validación**: Todas las restricciones `CHECK` definidas (e.g., valores positivos, días válidos de la semana) se verificaron con éxito al intentar insertar datos erróneos.

4.7.3 Pruebas Realizadas

Las pruebas realizadas incluyeron los siguientes escenarios:

1. Inserción de datos en la tabla `estado` con valores válidos y duplicados.
2. Inserción de municipios asociados a un estado no existente, lo que resultó en un error debido a la restricción de clave foránea.
3. Inserción de predicciones con valores fuera de rango o días de la semana no válidos, lo que activó las restricciones `CHECK`.
4. Actualización y eliminación de registros en cascada para validar las relaciones definidas en las claves foráneas.

5 Procesamiento

5.1 Flujo de Predicciones

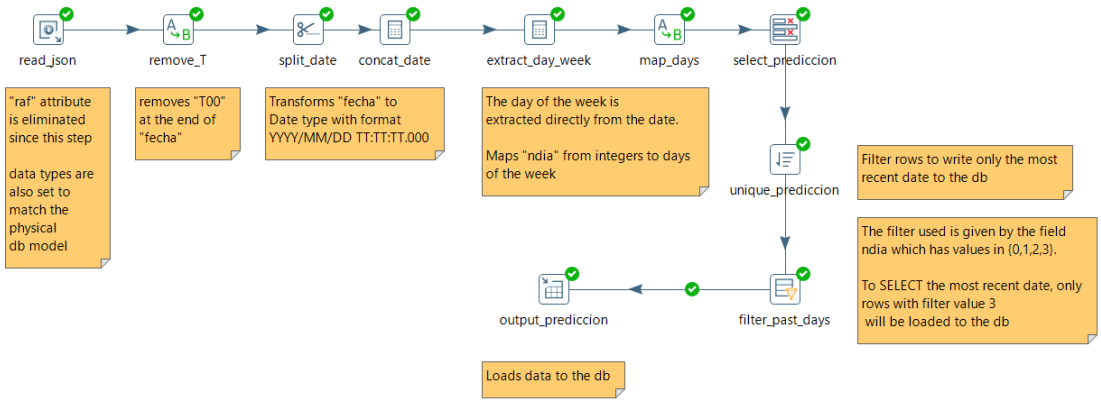


Figure 4: Flujo ETL para predicciones.

read_json

Figure 5: Step **read_json**.

- **Función**: Este paso lee datos desde un archivo JSON que contiene las predicciones.
- **Acciones específicas**:
  - Elimina un atributo innecesario llamado `raf`, que no será utilizado en el proceso.
  - Convierte y ajusta los tipos de datos de cada campo del JSON para alinearlos con el esquema físico de la base de datos destino.

- **Resultado**: Los datos JSON son importados y preparados para ser manipulados.

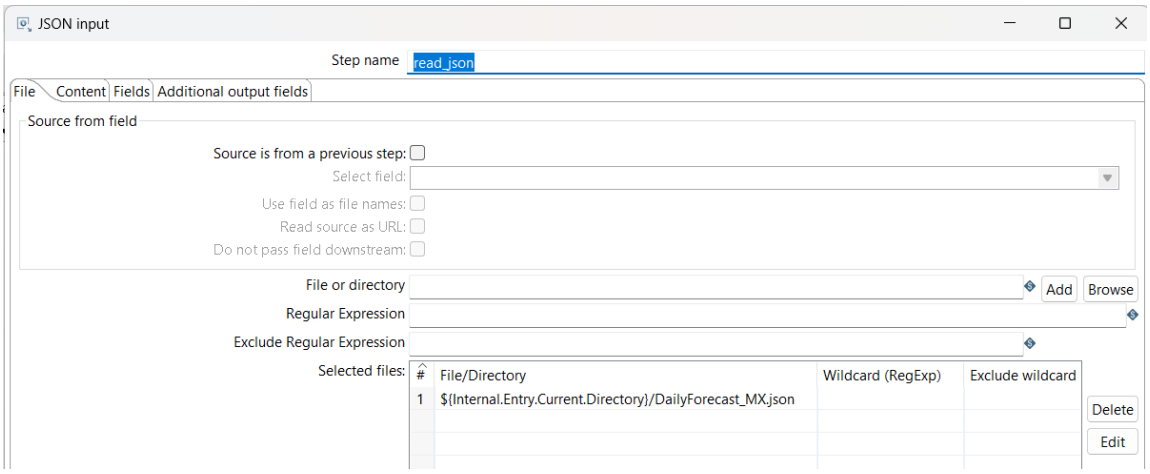
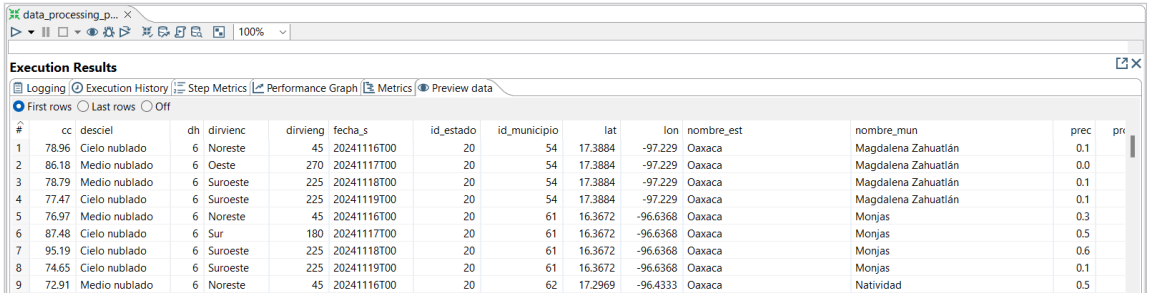


Figure 6: JSON input



#	cc	desciel	dh	dirvienc	dirvieng	fecha_s	id_estado	id_municipio	lat	lon	nombre_est	nombre_mun	prec	pr
1	78.96	Cielo nublado	6	Noreste	45	20241116T00	20	54	17.3884	-97.229	Oaxaca	Magdalena Zahuatlán	0.1	
2	86.18	Medio nublado	6	Oeste	270	20241117T00	20	54	17.3884	-97.229	Oaxaca	Magdalena Zahuatlán	0.0	
3	78.79	Medio nublado	6	Suroeste	225	20241118T00	20	54	17.3884	-97.229	Oaxaca	Magdalena Zahuatlán	0.1	
4	77.47	Cielo nublado	6	Suroeste	225	20241119T00	20	54	17.3884	-97.229	Oaxaca	Magdalena Zahuatlán	0.1	
5	76.97	Medio nublado	6	Noreste	45	20241116T00	20	61	16.3672	-96.6368	Oaxaca	Monjas	0.3	
6	87.48	Cielo nublado	6	Sur	180	20241117T00	20	61	16.3672	-96.6368	Oaxaca	Monjas	0.5	
7	95.19	Cielo nublado	6	Suroeste	225	20241118T00	20	61	16.3672	-96.6368	Oaxaca	Monjas	0.6	
8	74.65	Cielo nublado	6	Suroeste	225	20241119T00	20	61	16.3672	-96.6368	Oaxaca	Monjas	0.1	
9	72.91	Medio nublado	6	Noreste	45	20241116T00	20	62	17.2969	-96.4333	Oaxaca	Natividad	0.5	

Figure 7: Preview resultante

remove_T

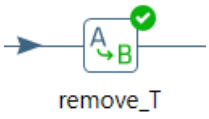
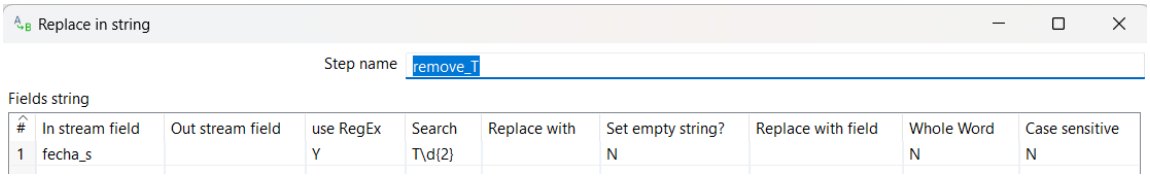


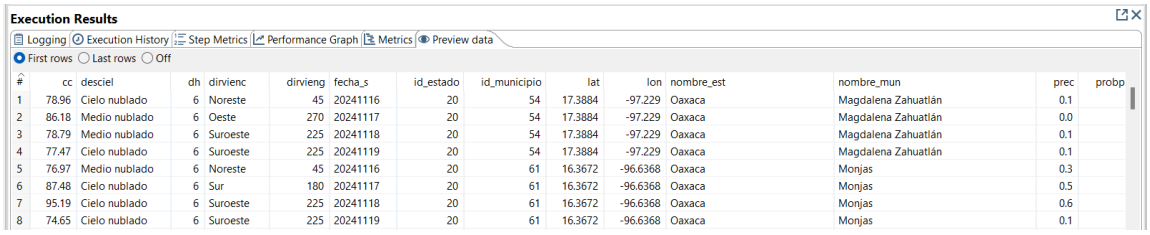
Figure 8: Step **remove_T**.

- **Función**: Limpia el campo `fecha` para eliminar información innecesaria.
- **Acciones específicas**:
  - Elimina el sufijo `"T00"` que aparece al final de las fechas, simplificando su formato.
- **Resultado**: El campo `fecha` ahora tiene un formato más limpio y está listo para su transformación.



#	In stream field	Out stream field	use RegEx	Search	Replace with	Set empty string?	Replace with field	Whole Word	Case sensitive
1	fecha_s		Y	T\d{2}		N		N	N

Figure 9: Replace in string



#	cc	desciel	dh	dirvienc	dirvieng	fecha_s	id_estado	id_municipio	lat	lon	nombre_est	nombre_mun	prec	probp
1	78.96	Cielo nublado	6	Noreste	45	20241116	20	54	17.3884	-97.229	Oaxaca	Magdalena Zahuatlán	0.1	
2	86.18	Medio nublado	6	Oeste	270	20241117	20	54	17.3884	-97.229	Oaxaca	Magdalena Zahuatlán	0.0	
3	78.79	Medio nublado	6	Suroeste	225	20241118	20	54	17.3884	-97.229	Oaxaca	Magdalena Zahuatlán	0.1	
4	77.47	Cielo nublado	6	Suroeste	225	20241119	20	54	17.3884	-97.229	Oaxaca	Magdalena Zahuatlán	0.1	
5	76.97	Medio nublado	6	Noreste	45	20241116	20	61	16.3672	-96.6368	Oaxaca	Monjas	0.3	
6	87.48	Cielo nublado	6	Sur	180	20241117	20	61	16.3672	-96.6368	Oaxaca	Monjas	0.5	
7	95.19	Cielo nublado	6	Suroeste	225	20241118	20	61	16.3672	-96.6368	Oaxaca	Monjas	0.6	
8	74.65	Cielo nublado	6	Suroeste	225	20241119	20	61	16.3672	-96.6368	Oaxaca	Monjas	0.1	

Figure 10: Preview resultante

split_date

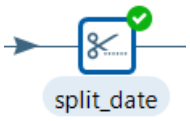
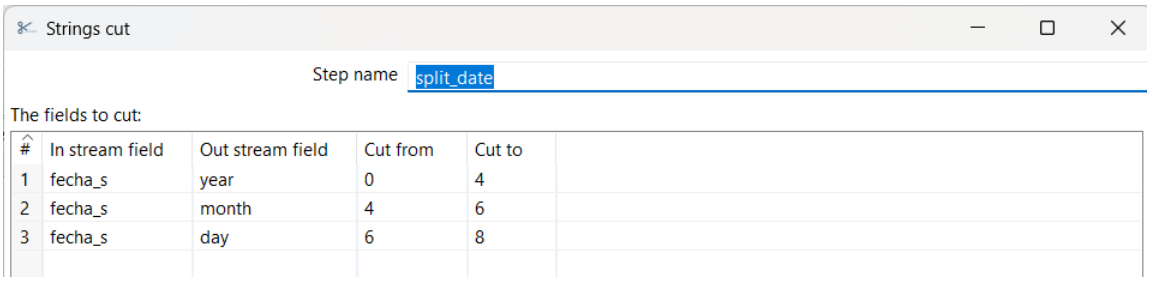


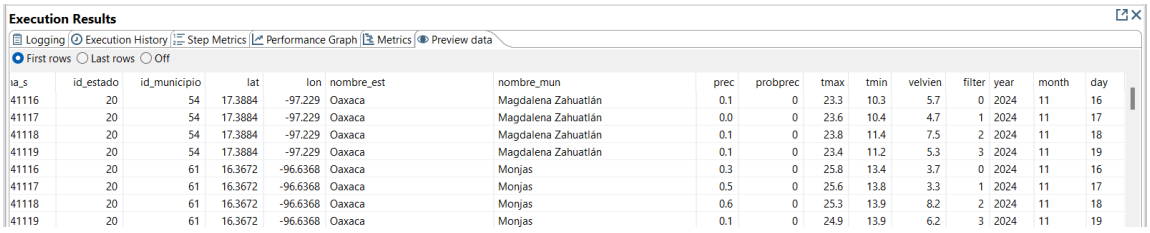
Figure 11: Step **split_date**

- **Función**: Divide el campo `fecha` en sus componentes fundamentales (día, mes, año, hora).
- **Resultado**: Cada componente de la fecha es accesible para procesos específicos.



#	In stream field	Out stream field	Cut from	Cut to
1	fecha_s	year	0	4
2	fecha_s	month	4	6
3	fecha_s	day	6	8

Figure 12: Strings cut



ia_s	id_estado	id_municipio	lat	lon	nombre_est	nombre_mun	prec	probprec	tmax	tmin	velvien	filter	year	month	day
41116	20	54	17.3884	-97.229	Oaxaca	Magdalena Zahuatlán	0.1	0	23.3	10.3	5.7	0	2024	11	16
41117	20	54	17.3884	-97.229	Oaxaca	Magdalena Zahuatlán	0.0	0	23.6	10.4	4.7	1	2024	11	17
41118	20	54	17.3884	-97.229	Oaxaca	Magdalena Zahuatlán	0.1	0	23.8	11.4	7.5	2	2024	11	18
41119	20	54	17.3884	-97.229	Oaxaca	Magdalena Zahuatlán	0.1	0	23.4	11.2	5.3	3	2024	11	19
41116	20	61	16.3672	-96.6368	Oaxaca	Monjas	0.3	0	25.8	13.4	3.7	0	2024	11	16
41117	20	61	16.3672	-96.6368	Oaxaca	Monjas	0.5	0	25.6	13.8	3.3	1	2024	11	17
41118	20	61	16.3672	-96.6368	Oaxaca	Monjas	0.6	0	25.3	13.9	8.2	2	2024	11	18
41119	20	61	16.3672	-96.6368	Oaxaca	Monjas	0.1	0	24.9	13.9	6.2	3	2024	11	19

Figure 13: Preview resultante

concat_date

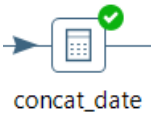
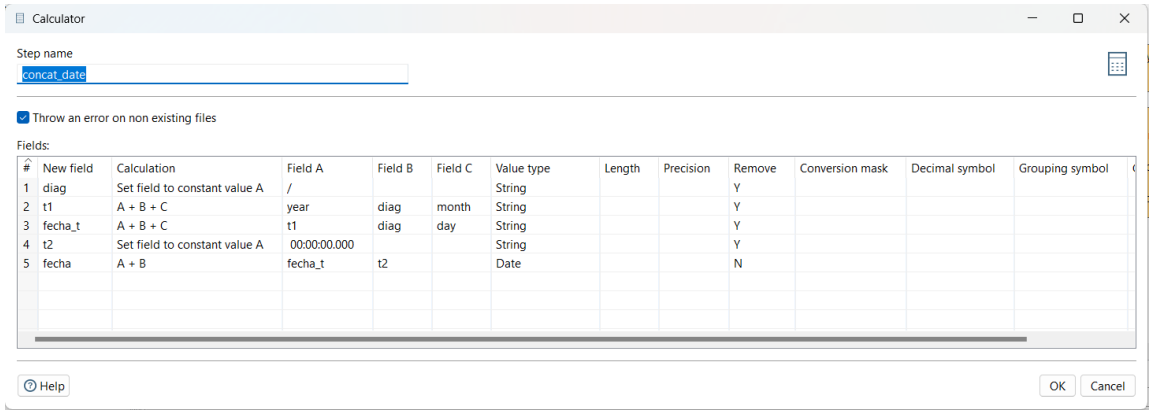


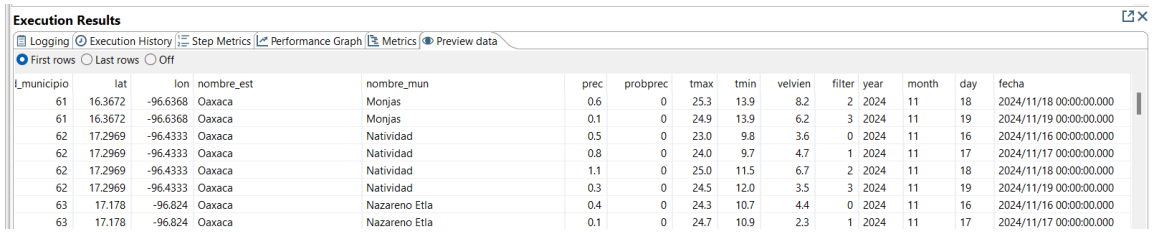
Figure 14: Step **concat_date**.

- **Función**: Reconstruye el campo `fecha` utilizando un formato estándar.
- **Acciones específicas**:
  - Transforma la fecha en el formato `YYYY/MM/DD`, que es adecuado para análisis y almacenamiento.
- **Resultado**: El campo de fecha ahora está en un formato uniforme y estandarizado.



#	New field	Calculation	Field A	Field B	Field C	Value type	Length	Precision	Remove	Conversion mask	Decimal symbol	Grouping symbol
1	diag	Set field to constant value A	/			String			Y			
2	t1	A + B + C	year	diag	month	String			Y			
3	fecha_t	A + B + C	t1	diag	day	String			Y			
4	t2	Set field to constant value A	00:00:00.000			String			Y			
5	fecha	A + B	fecha_t	t2		Date			N			

Figure 15: Concat date



l_municipio	lat	lon	nombre_est	nombre_mun	prec	probprec	tmax	tmin	velvien	filter	year	month	day	fecha
61	16.3672	-96.6368	Oaxaca	Monjas	0.6	0	25.3	13.9	8.2	2	2024	11	18	2024/11/18 00:00:00.000
61	16.3672	-96.6368	Oaxaca	Monjas	0.1	0	24.9	13.9	6.2	3	2024	11	19	2024/11/19 00:00:00.000
62	17.2969	-96.4333	Oaxaca	Natividad	0.5	0	23.0	9.8	3.6	0	2024	11	16	2024/11/16 00:00:00.000
62	17.2969	-96.4333	Oaxaca	Natividad	0.8	0	24.0	9.7	4.7	1	2024	11	17	2024/11/17 00:00:00.000
62	17.2969	-96.4333	Oaxaca	Natividad	1.1	0	25.0	11.5	6.7	2	2024	11	18	2024/11/18 00:00:00.000
62	17.2969	-96.4333	Oaxaca	Natividad	0.3	0	24.5	12.0	3.5	3	2024	11	19	2024/11/19 00:00:00.000
63	17.178	-96.824	Oaxaca	Nazareno Etla	0.4	0	24.3	10.7	4.4	0	2024	11	16	2024/11/16 00:00:00.000
63	17.178	-96.824	Oaxaca	Nazareno Etla	0.1	0	24.7	10.9	2.3	1	2024	11	17	2024/11/17 00:00:00.000

Figure 16: Preview resultante

extract_day_week

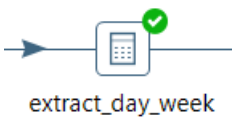


Figure 17: Step **extract_day_week**.

- **Función**: Extrae el día de la semana basado en la fecha.

- **Resultado**: El día de la semana (como un valor numérico, e.g., 1 = Domingo) queda disponible para su uso.

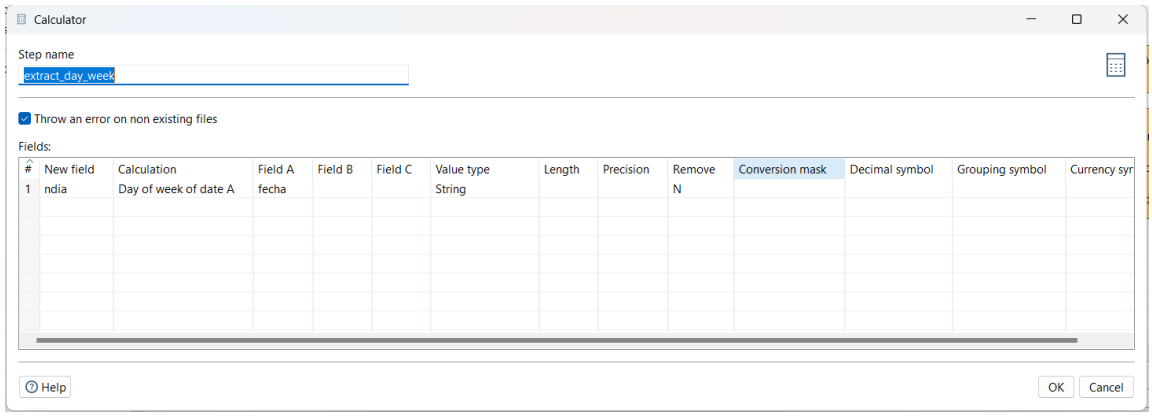


Figure 18: Calculator

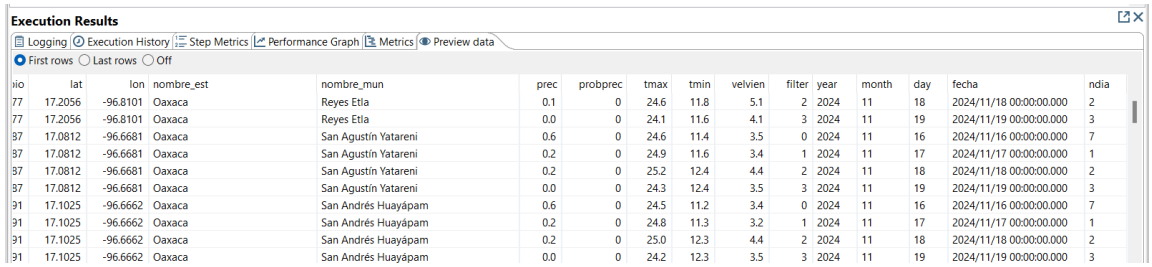


Figure 19: Preview resultante

map_days

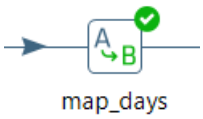
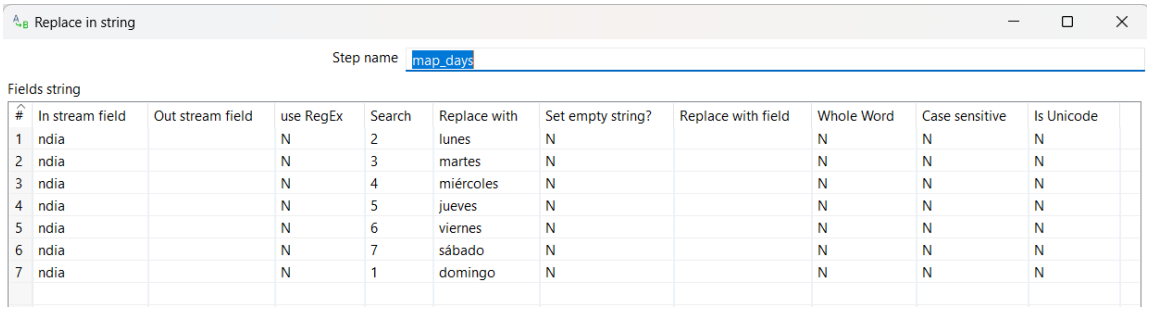


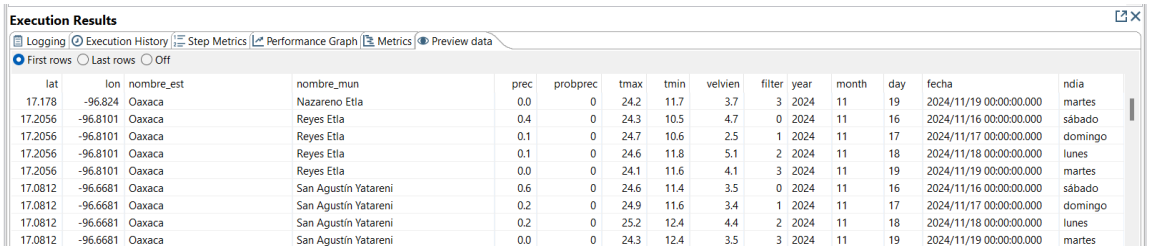
Figure 20: Step **map_days**.

- **Función**: Mapea los valores numéricos de días de la semana a nombres descriptivos.
- **Acciones específicas**:
  - Convierte valores como 0, 1, 2, etc., en sus equivalentes textuales (`Domingo`, `Lunes`, `Martes`, etc.).
- **Resultado**: Los días de la semana se representan en un formato más entendible.



#	In stream field	Out stream field	use RegEx	Search	Replace with	Set empty string?	Replace with field	Whole Word	Case sensitive	Is Unicode
1	ndia		N	2	lunes	N		N	N	N
2	ndia		N	3	martes	N		N	N	N
3	ndia		N	4	miércoles	N		N	N	N
4	ndia		N	5	jueves	N		N	N	N
5	ndia		N	6	viernes	N		N	N	N
6	ndia		N	7	sábado	N		N	N	N
7	ndia		N	1	domingo	N		N	N	N

Figure 21: Calculator



lat	lon	nombre_est	nombre_mun	prec	probprec	tmax	tmin	velvien	filter	year	month	day	fecha	ndia
17.178	-96.824	Oaxaca	Nazareno Etla	0.0	0	24.2	11.7	3.7	3	2024	11	19	2024/11/19 00:00:00.000	martes
17.2056	-96.8101	Oaxaca	Reyes Etla	0.4	0	24.3	10.5	4.7	0	2024	11	16	2024/11/16 00:00:00.000	sábado
17.2056	-96.8101	Oaxaca	Reyes Etla	0.1	0	24.7	10.6	2.5	1	2024	11	17	2024/11/17 00:00:00.000	domingo
17.2056	-96.8101	Oaxaca	Reyes Etla	0.1	0	24.6	11.8	5.1	2	2024	11	18	2024/11/18 00:00:00.000	lunes
17.2056	-96.8101	Oaxaca	Reyes Etla	0.0	0	24.1	11.6	4.1	3	2024	11	19	2024/11/19 00:00:00.000	martes
17.0812	-96.6681	Oaxaca	San Agustín Yatareni	0.6	0	24.6	11.4	3.5	0	2024	11	16	2024/11/16 00:00:00.000	sábado
17.0812	-96.6681	Oaxaca	San Agustín Yatareni	0.2	0	24.9	11.6	3.4	1	2024	11	17	2024/11/17 00:00:00.000	domingo
17.0812	-96.6681	Oaxaca	San Agustín Yatareni	0.2	0	25.2	12.4	4.4	2	2024	11	18	2024/11/18 00:00:00.000	lunes
17.0812	-96.6681	Oaxaca	San Agustín Yatareni	0.0	0	24.3	12.4	3.5	3	2024	11	19	2024/11/19 00:00:00.000	martes

Figure 22: Preview resultante

select_prediccion

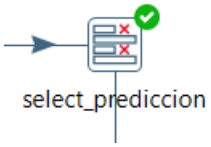


Figure 23: Step **select_prediccion**.

- **Función**: Seleccionar las columnas y variables necesarias que servirán como entrada para los pasos posteriores, asegurando la compatibilidad con el modelo físico de la base de datos.
- **Acciones específicas**:
  - Se seleccionan únicamente las columnas que coinciden con el esquema del modelo físico de la base de datos, descartando cualquier dato adicional o redundante.
  - La columna `fecha` se transforma al formato previamente definido (`YYYY/MM/DD`) para asegurar uniformidad y compatibilidad.
  - Se eliminan columnas que no sean requeridas para los pasos posteriores, reduciendo la complejidad del conjunto de datos.
- **Resultado**: Un subconjunto de datos que contiene únicamente las columnas relevantes, con los formatos ajustados según las especificaciones del modelo físico de la base de datos.

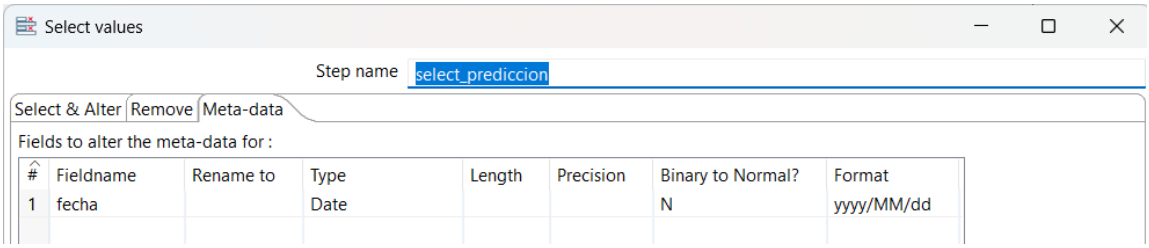
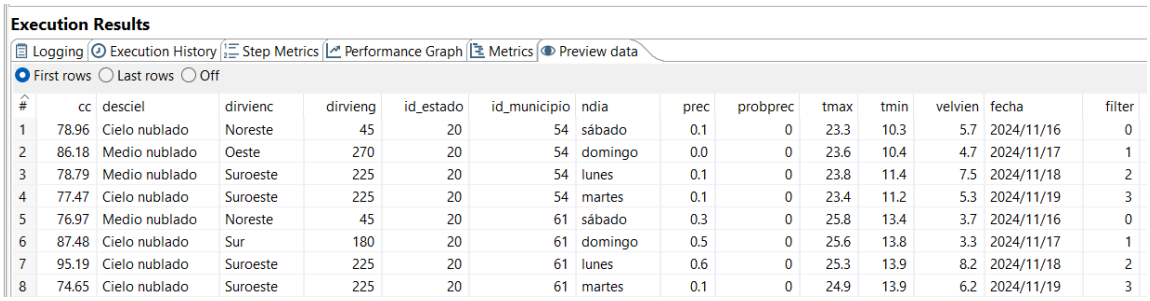


Figure 24: Filter rows



#	cc	desciel	dirvienc	dirvieng	id_estado	id_municipio	ndia	prec	probprec	tmax	tmin	velvien	fecha	filter
1	78.96	Cielo nublado	Noreste	45	20	54	sábado	0.1	0	23.3	10.3	5.7	2024/11/16	0
2	86.18	Medio nublado	Oeste	270	20	54	domingo	0.0	0	23.6	10.4	4.7	2024/11/17	1
3	78.79	Medio nublado	Suroeste	225	20	54	lunes	0.1	0	23.8	11.4	7.5	2024/11/18	2
4	77.47	Cielo nublado	Suroeste	225	20	54	martes	0.1	0	23.4	11.2	5.3	2024/11/19	3
5	76.97	Medio nublado	Noreste	45	20	61	sábado	0.3	0	25.8	13.4	3.7	2024/11/16	0
6	87.48	Cielo nublado	Sur	180	20	61	domingo	0.5	0	25.6	13.8	3.3	2024/11/17	1
7	95.19	Cielo nublado	Suroeste	225	20	61	lunes	0.6	0	25.3	13.9	8.2	2024/11/18	2
8	74.65	Cielo nublado	Suroeste	225	20	61	martes	0.1	0	24.9	13.9	6.2	2024/11/19	3

Figure 25: Preview resultante

unique_prediccion

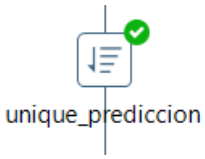


Figure 26: Step **unique_prediccion**.

- **Función**: Elimina duplicados para garantizar que cada predicción sea única.
- **Resultado**: Una colección de predicciones únicas y limpias.

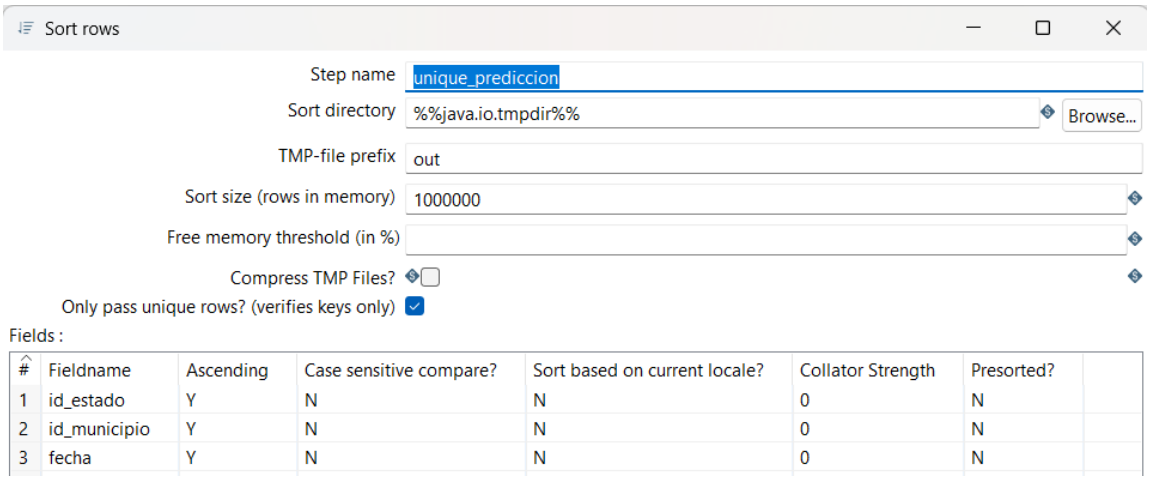


Figure 27: Sort rows

Execution Results

Logging | Execution History | Step Metrics | Performance Graph | Metrics | Preview data

First rows | Last rows | Off

#	cc	desciel	dirvienc	dirvieng	id_estado	id_municipio	ndia	prec	probprec	tmax	tmin	velvien	fecha	filter
1	28.01	Medio nublado	Sur	180	1	1	sábado	0.0	0	27.3	10.7	8.7	2024/11/16	0
2	52.12	Cielo nublado	Sur	180	1	1	domingo	0.1	0	26.1	12.1	16.4	2024/11/17	1
3	77.81	Poco nuboso	Suroeste	225	1	1	lunes	0.0	0	27.3	10.9	12.7	2024/11/18	2
4	89.55	Despejado	Suroeste	225	1	1	martes	0.0	0	27.4	9.9	9.5	2024/11/19	3
5	19.03	Cielo nublado	Sur	180	1	2	sábado	0.0	0	26.4	10.9	9.6	2024/11/16	0
6	36.56	Cielo nublado	Sur	180	1	2	domingo	0.0	0	25.6	12.1	21.7	2024/11/17	1
7	65.62	Poco nuboso	Sur	180	1	2	lunes	0.0	0	26.4	10.9	15.3	2024/11/18	2
8	91.25	Despejado	Suroeste	225	1	2	martes	0.0	0	26.4	10.0	11.5	2024/11/19	3

Figure 28: Preview resultante

filter_past_days

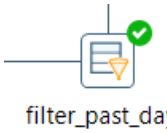


Figure 29: Step **filter_past_days**.

- **Función**: Selecciona registros basados en fechas recientes.
- **Acciones específicas**:
  - Usa un campo llamado `ndia` para identificar y mantener únicamente los registros recientes (`ndia = 3`).
- **Resultado**: Solo los datos más recientes se conservan para el almacenamiento.

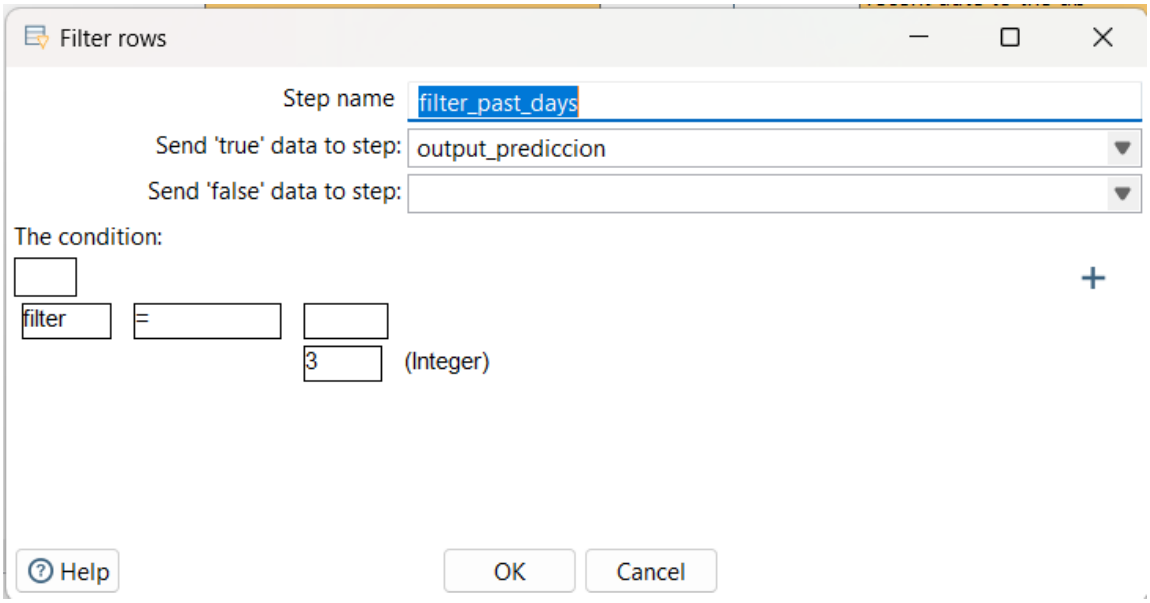


Figure 30: Filter rows

Execution Results

Logging | Execution History | Step Metrics | Performance Graph | Metrics | Preview data

First rows | Last rows | Off

#	cc	desciel	dirvienc	dirvieng	id_estado	id_municipio	ndia	prec	probprec	tmax	tmin	velvien	fecha	filter
1	89.55	Despejado	Suroeste	225	1	1	martes	0.0	0	27.4	9.9	9.5	2024/11/19	3
2	91.25	Despejado	Suroeste	225	1	2	martes	0.0	0	26.4	10.0	11.5	2024/11/19	3
3	88.73	Despejado	Suroeste	225	1	3	martes	0.0	0	26.8	9.1	7.1	2024/11/19	3
4	89.18	Despejado	Suroeste	225	1	4	martes	0.0	0	26.1	9.7	11.3	2024/11/19	3
5	89.91	Despejado	Suroeste	225	1	5	martes	0.0	0	26.8	9.9	9.6	2024/11/19	3
6	90.89	Despejado	Suroeste	225	1	6	martes	0.0	0	26.8	10.0	10.9	2024/11/19	3
7	90.11	Despejado	Suroeste	225	1	7	martes	0.0	0	25.9	9.8	11.3	2024/11/19	3
8	89.49	Despejado	Suroeste	225	1	8	martes	0.0	0	25.1	9.5	10.7	2024/11/19	3
9	91.36	Despejado	Suroeste	225	1	9	martes	0.0	0	26.7	10.0	11.5	2024/11/19	3

Figure 31: Preview resultante

output_prediccion

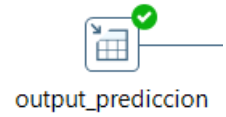


Figure 32: Step **output_prediccion**.

- **Función**: Escribe los datos procesados y filtrados en la base de datos.
- **Resultado**: Las predicciones limpias y relevantes están disponibles para su análisis futuro.

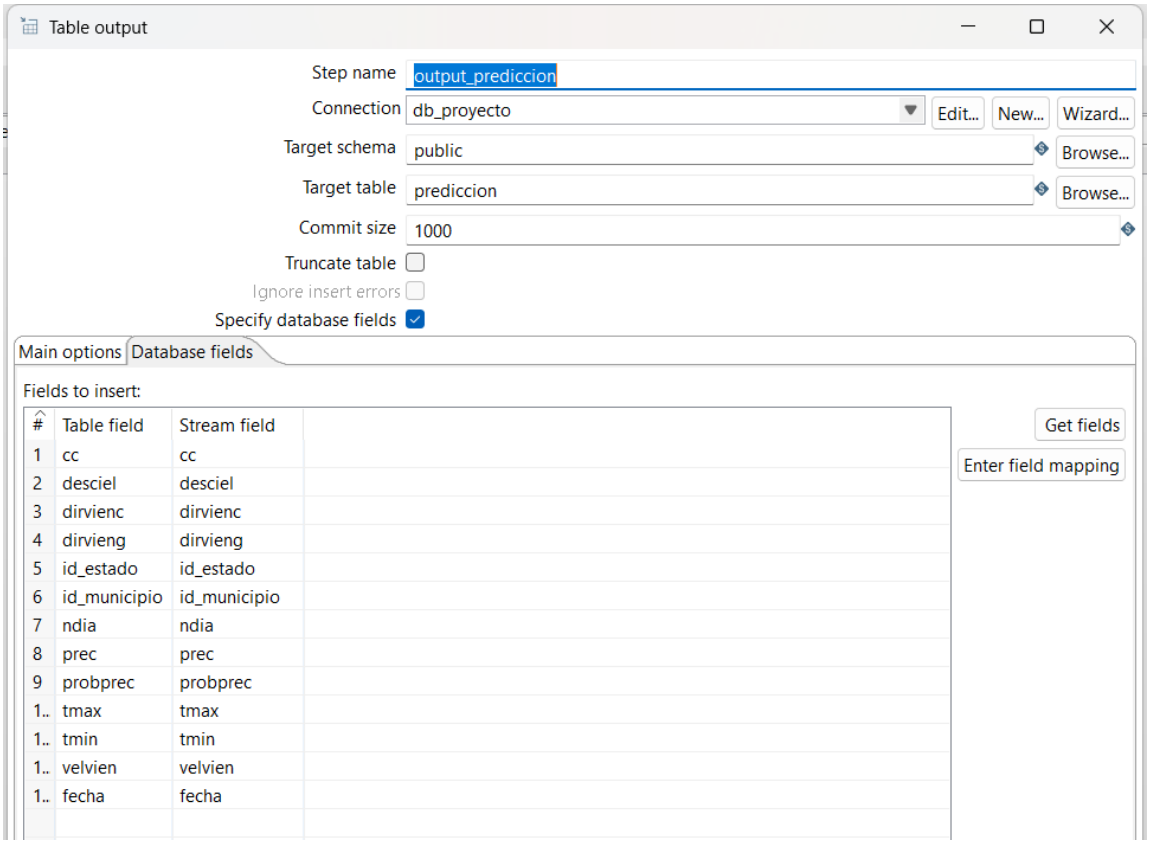


Figure 33: Table output

Propósito

Este flujo se utiliza para procesar datos de predicciones, asegurando consistencia y relevancia antes de almacenarlos.

5.2 Flujo de Estados

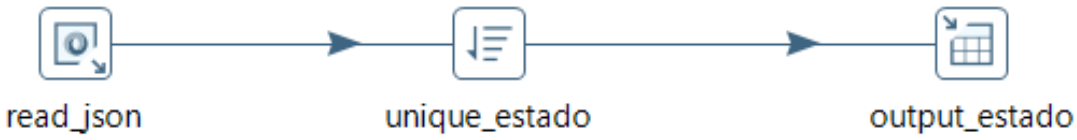


Figure 34: Flujo ETL para estados.

read_json

Figure 35: Step **read_json**.

- **Función**: Extrae datos desde un archivo JSON que contiene registros de estados.
- **Resultado**: Los datos son importados para ser transformados.

Execution Results

Logging | Execution History | Step Metrics | Performance Graph | Metrics | Preview data

First rows | Last rows | Off

#	id_estado	nombre_est
1	20	Oaxaca
2	20	Oaxaca
3	20	Oaxaca
4	20	Oaxaca
5	20	Oaxaca

Figure 36: Preview resultante

unique_estado

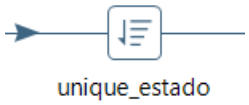


Figure 37: Step **unique_estado**.

- **Función**: Elimina duplicados en los datos de estados.
- **Resultado**: Una lista única de estados está disponible para ser almacenada.

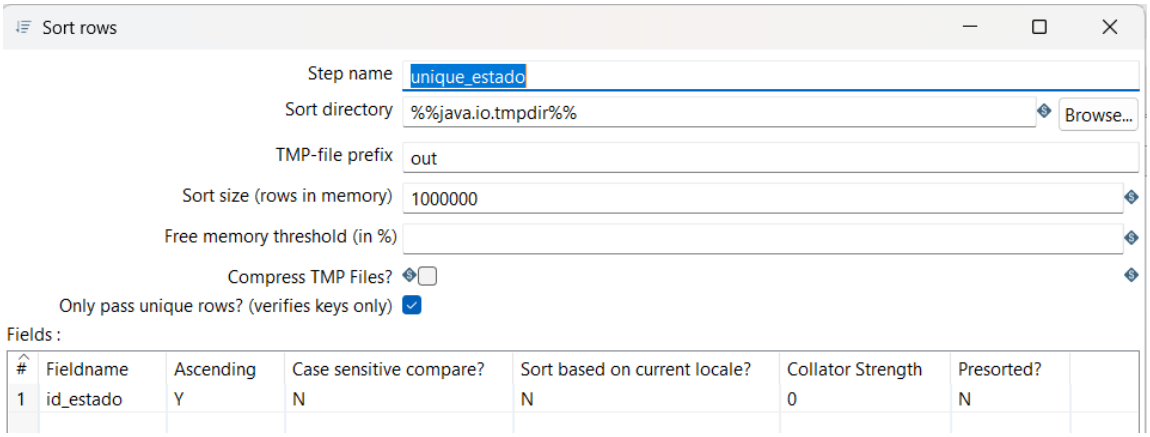


Figure 38: Sort rows

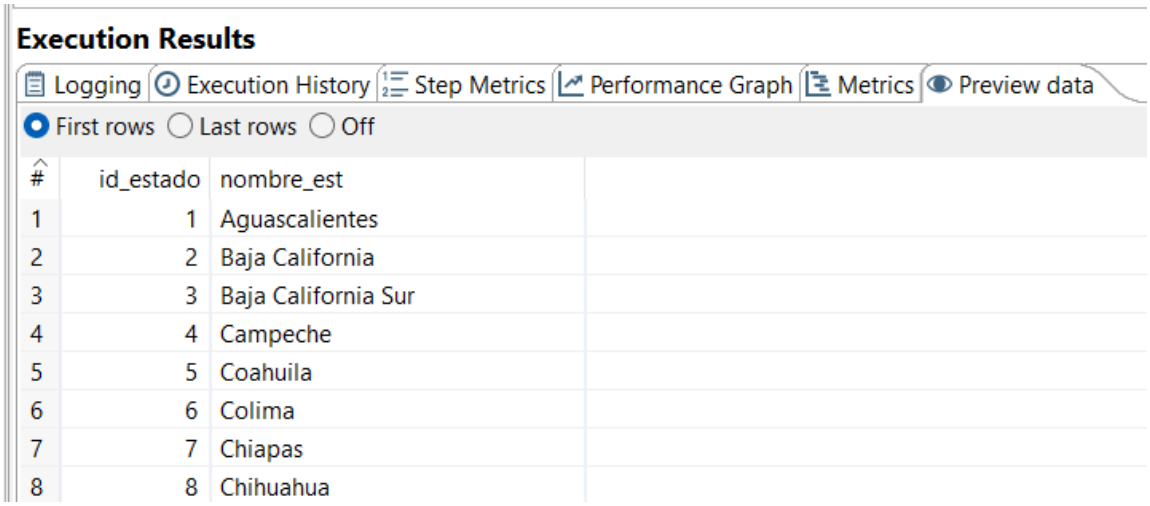


Figure 39: Preview resultante

output_estado

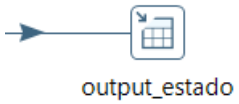


Figure 40: Step **output_estado**.

- **Función**: Carga los datos únicos a una base de datos o sistema de almacenamiento.
- **Resultado**: Los registros limpios de estados son almacenados.

Propósito

Este flujo asegura que los datos relacionados con estados sean únicos y consistentes.

5.3 Flujo de Municipios

Figure 41: Flujo ETL para municipios.

read_json

Figure 42: Step **read_json**.

- **Función**: Extrae datos desde un archivo JSON que contiene registros de municipios.
- **Resultado**: Los datos son importados para ser procesados.

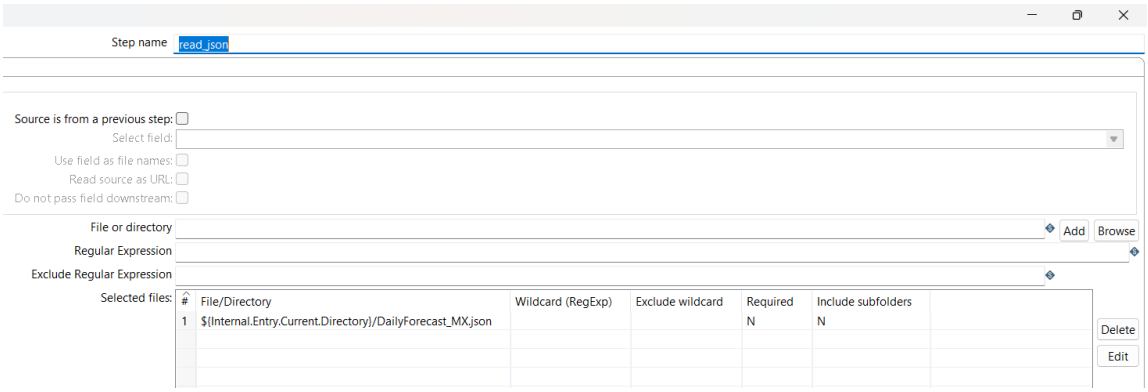


Figure 43: JSON input

Execution Results

Logging | Execution History | Step Metrics | Performance Graph | Metrics | Preview data

First rows | Last rows | Off

#	dh	id_estado	id_municipio	lat	lon	nombre_mun
1	6	20	54	17.3884	-97.229	Magdalena Zahuatlán
2	6	20	54	17.3884	-97.229	Magdalena Zahuatlán
3	6	20	54	17.3884	-97.229	Magdalena Zahuatlán
4	6	20	54	17.3884	-97.229	Magdalena Zahuatlán
5	6	20	61	16.3672	-96.6368	Monjas
6	6	20	61	16.3672	-96.6368	Monjas
7	6	20	61	16.3672	-96.6368	Monjas
8	6	20	61	16.3672	-96.6368	Monjas

Figure 44: Preview resultante

unique_municipio

Figure 45: Step **unique_municipio**.

- **Función**: Elimina duplicados en los datos de municipios.
- **Resultado**: Una lista única de municipios está disponible.

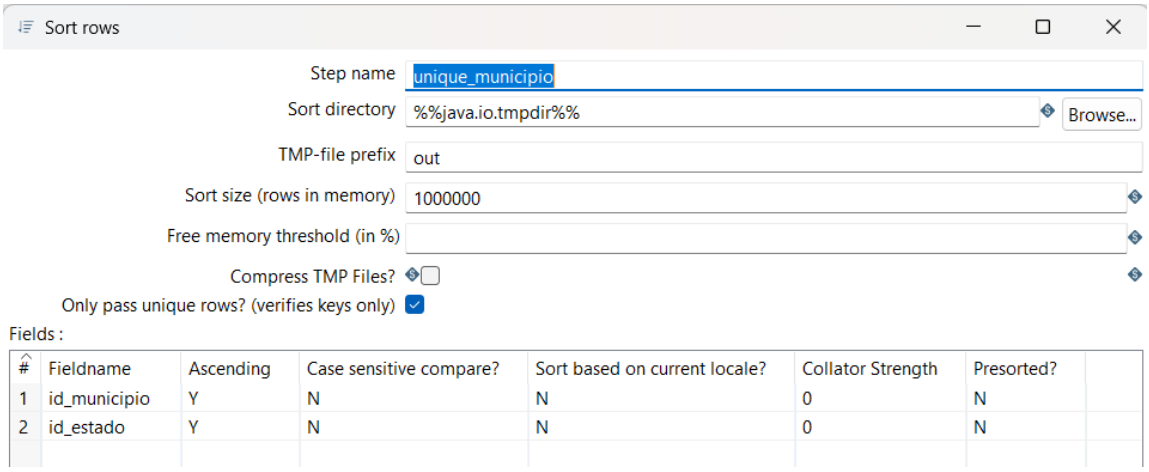


Figure 46: Sort rows

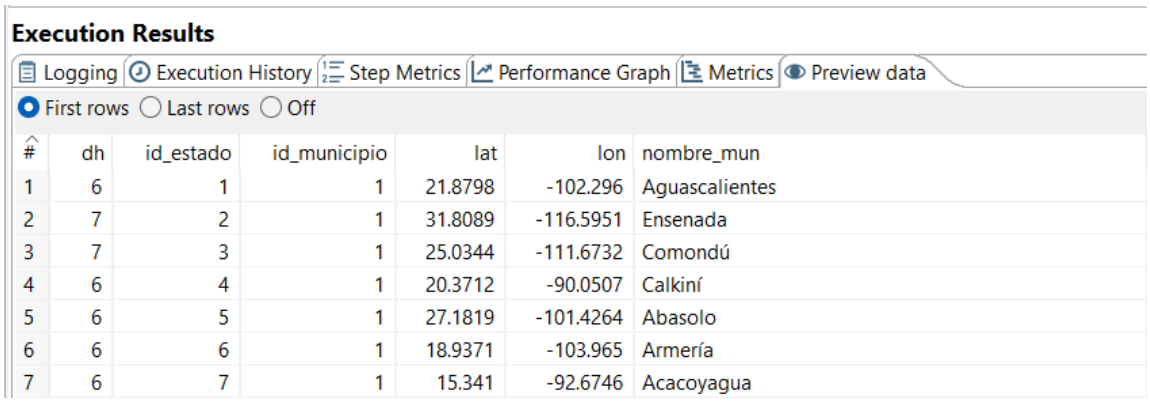


Figure 47: Preview resultante

output_municipio

- **Función**: Carga los datos únicos a una base de datos o sistema de almacenamiento.
- **Resultado**: Los registros limpios de municipios están almacenados.

Propósito

Este flujo garantiza que los datos relacionados con municipios estén libres de duplicados antes de almacenarse.

6 Analítica de datos

6.1 Ventajas y desventajas de la representación inicial de la información

Ventajas

1. **GNU ZIP:** El servicio web de la CONAGUA devuelve el archivo DailyForecast_MX.gz en formato GNU ZIP. Este tipo de compresión es eficiente para reducir el espacio necesario para almacenar los datos y permite que los tiempos de transferencia

por red sean menores al descargar el archivo. Esto es especialmente útil cuando se manejan grandes volúmenes de datos meteorológicos que se descargarán de manera frecuente como en este caso.

2. **JSON:** El formato de escritura del archivo de texto es JSON. El uso de JSON permite que la información sea fácilmente legible por humanos y máquinas. Tiene una estructura clave-valor que hace que los datos sean intuitivos de entender y manipular. El formato JSON es altamente manipulable con distintos lenguajes de programación y herramientas. Por ejemplo, se puede acceder directamente a los datos de un municipio o estado específico usando Python.

Desventajas

1. **Descomprimir el archivo .gz:** Antes de poder trabajar con los datos, es necesario descomprimir el archivo .gz y luego procesar el contenido JSON. Para tales operaciones fue necesario introducir un paso adicional, lo cual se traduce en tiempo y recursos, especialmente si el archivo es grande.

2. **Complejidad en las Consultas:** El formato JSON no está optimizado para consultas complejas como lo están los sistemas de bases de datos. Para realizar filtrados (como encontrar la tmin de hoy y mañana), los datos primero deben cargarse en memoria y procesarse con herramientas adicionales. Por ello es mucho más conveniente tener los datos normalizados en bases de datos y usar consultas SQL.

6.2 Consulta para obtener los 5 municipios con descensos de temperatura más marcados

Consulta SQL

```sql
SELECT
    e.nombre_est,
    m.nombre_mun,
    p_hoy.tmin AS tmin_hoy,
    p_manana.tmin AS tmin_manana,
    p_hoy.tmin - p_manana.tmin AS descenso_temp
FROM
    estado e
JOIN
    municipio m ON e.id_estado = m.id_estado
JOIN
    prediccion p_hoy ON m.id_estado = p_hoy.id_estado AND
    m.id_municipio = p_hoy.id_municipio
JOIN
    prediccion p_manana ON m.id_estado = p_manana.id_estado AND
    m.id_municipio = p_manana.id_municipio
WHERE
    p_hoy.fecha = CURRENT_DATE
    AND p_manana.fecha = CURRENT_DATE + INTERVAL '1 day'
ORDER BY descenso_temp DESC LIMIT 5;
```

Explicación

- **Propósito:** Esta consulta identifica los cinco municipios donde se espera que ocurra el mayor descenso de temperatura mínima entre hoy y mañana.

- **Detalles de la consulta:**

- Se realizan uniones (`JOIN`) entre las tablas de `estado`, `municipio` y las predicciones para hoy (`p_hoy`) y mañana (`p_manana`) basándose en identificadores comunes.
- Se calculan las diferencias entre la temperatura mínima de hoy y mañana utilizando `p_hoy.tmin - p_manana.tmin`.
- Se filtran las predicciones para fechas específicas (`CURRENT_DATE` y `CURRENT_DATE + INTERVAL '1 day'`).
- La consulta ordena los resultados en orden descendente según la magnitud del descenso de temperatura y selecciona los cinco mayores valores con `LIMIT 5`.

· **Resultado esperado:** Una lista de cinco municipios junto con su estado correspondiente, mostrando las temperaturas mínimas de hoy y mañana, y la magnitud del descenso.

nombre_est	nombre_mun	tmin_hoy	tmin_manana	descenso_temp
Coahuila	Parras	16.20	10.50	5.7
Coahuila	Viesca	16.30	10.90	5.4
Durango	Peñón Blanco	12.20	7.20	5.0
Durango	San Juan de Guadalupe	14.40	9.40	5.0
Durango	Cuencamé	13.70	8.90	4.8

Table 1: Descensos de temperatura en municipios seleccionados

6.3 Consulta para obtener por estado el municipio con la cuarta temperatura más alta

Consulta SQL

```sql
WITH RankedPredictions AS (
    SELECT
        e.nombre_est,
        p.tmax,
        DENSE_RANK() OVER (PARTITION BY e.nombre_est ORDER BY p.tmax DESC) AS
    rank
    FROM
        estado e
    JOIN
        municipio m ON e.id_estado = m.id_estado
    JOIN
        prediccion p ON m.id_estado = p.id_estado AND m.id_municipio = p.
    id_municipio
)
SELECT DISTINCT
    nombre_est,
    tmax
FROM
    RankedPredictions
WHERE
    rank = 4;
```

Explicación

· **Propósito:** Esta consulta identifica, para cada estado, el municipio con la cuarta temperatura máxima más alta.

- **Detalles de la consulta:**
  - Se utiliza una consulta común con nombre (`WITH`) llamada `RankedPredictions` para generar un ranking (`DENSE_RANK`) de las temperaturas máximas (`tmax`) dentro de cada estado (`PARTITION BY e.nombre_est`).
  - En el ranking, las temperaturas máximas se ordenan en orden descendente (`ORDER BY p.tmax DESC`).
  - Se filtran los resultados para seleccionar únicamente los registros con `rank = 4`.
- **Resultado esperado:** Una lista que muestra el estado y la temperatura máxima correspondiente al municipio que ocupa el cuarto lugar dentro de ese estado.

nombre_est	tmax
Aguascalientes	29.00
Baja California	26.50
Baja California Sur	30.30
Campeche	36.00
Chiapas	33.00
Chihuahua	32.70
Ciudad de México	25.60
Coahuila	33.30
Colima	33.00
Durango	32.70
Estado de México	28.50
Guanajuato	31.10
Guerrero	34.40
Hidalgo	30.90
Jalisco	32.70
Michoacán de Ocampo	33.80
Morelos	30.90
Nayarit	33.30
Nuevo León	33.60
Oaxaca	35.30
Puebla	32.40
Querétaro de Arteaga	30.70
Quintana Roo	33.10
San Luis Potosí	33.90
Sinaloa	35.50
Sonora	34.10
Tabasco	33.20
Tamaulipas	35.60

Table 2: Temperaturas máximas por estado

6.4 Mejora de una situación práctica basada en los datos procesados

6.4.1 Caso de Estudio: Optimización del Riego y Prevención de Daños Agrícolas en Chiapas Mediante Análisis Meteorológico

Descripción del Problema

Chiapas es un estado con gran diversidad climática y uno de los mayores productores agrícolas de México. Cultivos como café, maíz, plátano y cacao dependen en gran

medida de las condiciones climáticas. Sin embargo, los agricultores enfrentan desafíos como:

1. **Eventos climáticos extremos:**
    - *Lluvias intensas:* Pueden causar inundaciones y erosión del suelo, afectando cultivos como el maíz y el plátano.
    - *Sequías estacionales:* Reducen la producción de cultivos como el café en regiones como la Sierra Madre de Chiapas.
2. **Acceso limitado a datos precisos:**
    - Los agricultores pequeños y medianos carecen de herramientas accesibles para interpretar datos meteorológicos y planificar acciones.
3. **Sostenibilidad del agua:**
    - Chiapas posee recursos hídricos importantes, pero su gestión eficiente es crucial para mantener la productividad agrícola sin comprometer el ecosistema.

Objetivo

Diseñar un sistema basado en datos meteorológicos para optimizar la gestión agrícola en Chiapas, permitiendo a los agricultores:

1. Anticiparse a lluvias torrenciales o períodos secos prolongados.
2. Planificar el riego según las predicciones diarias de lluvia, temperatura y humedad.
3. Proteger cultivos sensibles frente a vientos fuertes o cambios bruscos de temperatura.
4. Mejorar la sostenibilidad de los recursos naturales.

Propuesta de Solución

1. **Análisis de datos meteorológicos específicos del estado:** Utilizando datos meteorológicos como los proporcionados en el archivo JSON comprimido (`tmax`, `tmin`, `prec`, `probprec`, `velvien`), el sistema podrá identificar patrones locales en regiones específicas de Chiapas:
    - *Zona del Soconusco:* Predominantemente productora de café, vulnerable a lluvias intensas y derrumbes.
    - *Selva Lacandona:* Áreas con alta humedad que necesitan monitoreo de lluvias para evitar encharcamientos.
    - *Altos de Chiapas:* Clima más seco, donde la gestión del riego es crucial.
2. **Implementación de predicciones y alertas en tiempo real:**
    - Análisis de los datos meteorológicos para detectar eventos extremos.
    - Emisión de alertas automatizadas a través de mensajes de texto o aplicaciones móviles. Ejemplo:
        - "Probabilidad de lluvia alta (85%) en las próximas 24 horas en Tuxtla Chico. Se recomienda suspender el riego."
        - "Viento fuerte (ráfagas de 20 km/h) en la región de Comitán. Proteja cultivos sensibles."
    - Proyecciones a corto plazo (48 horas) para ajustar tareas agrícolas.

3. **Herramienta de visualización para agricultores:**
    - Predicciones regionales diarias y semanales de temperatura, precipitación y viento.
    - Gráficos de evolución de lluvias y temperaturas por región.
    - Indicadores de riesgo climático, como heladas o sequías.
4. **Capacitación y adopción tecnológica:**
    - Implementar talleres locales para capacitar a los agricultores sobre el uso del sistema.
    - Generar reportes en lenguaje sencillo, adaptados a las necesidades de pequeños productores.

Implementación Técnica

1. **Procesamiento de datos:**
    - Los datos del archivo `.gz` se descomprimirán y procesarán en un entorno de Python.
    - Crear un flujo ETL para procesar y almacenar los datos en una base de datos relacional (PostgreSQL, por ejemplo) para permitir consultas dinámicas.
2. **Análisis predictivo:**
    - Modelos estadísticos o de *machine learning* (por ejemplo, regresión o redes neuronales) para predecir probabilidades de lluvia, temperaturas y eventos extremos.
3. **Infraestructura tecnológica:**
    - Integración de un servidor web para ofrecer el sistema como un servicio accesible desde cualquier dispositivo.
    - Uso de herramientas para crear visualizaciones interactivas.

Impacto Esperado

1. **Incremento en la productividad agrícola:**
    - Ajustes en el riego con base en predicciones precisas.
    - Reducción de pérdidas por lluvias inesperadas o sequías.
2. **Ahorro de recursos naturales:**
    - Uso eficiente del agua, protegiendo cuencas importantes como la del Grijalva.
3. **Empoderamiento de agricultores locales:**
    - Acceso a información climática que antes estaba fuera de su alcance.
    - Mejora de la toma de decisiones en comunidades rurales.
4. **Mitigación de riesgos climáticos:**
    - Protección de cultivos frente a heladas, sequías y vientos fuertes.
    - Mayor resiliencia frente al cambio climático.

Ejemplo Práctico Aplicado a Chiapas

Un productor de café en la región del Soconusco podría:

- Recibir una alerta de probabilidad de lluvias intensas (>80%) en las próximas 24 horas, permitiéndole suspender el riego y proteger el suelo con coberturas.
- Consultar un gráfico en su teléfono que indique un período seco de 3 días, ajustando el riego de manera más eficiente.

7 Visualización de datos

La visualización de datos meteorológicos en este proyecto se diseñó para presentar de manera interactiva y comprensible los datos relacionados con la temperatura en los municipios de México. Esta sección detalla las herramientas y metodologías utilizadas para crear una interfaz gráfica que facilite la exploración de datos climáticos por parte del usuario.

7.1 Descripción General

El sistema de visualización permite a los usuarios consultar datos meteorológicos diarios de todos los municipios de México, enfocados en temperaturas máximas y mínimas. Además, se ofrece un contexto temporal mostrando las variaciones de temperatura en un rango que incluye dos días previos, el día seleccionado y dos días posteriores.

La plataforma se desarrolla como una página web interactiva, integrando múltiples funcionalidades para garantizar una experiencia de usuario intuitiva y eficiente.

7.2 Componentes Principales

7.2.1 Página Web Interactiva

La visualización se implementa en una página web que permite:

- Seleccionar la fecha específica para la cual se desean consultar los datos.
- Elegir entre temperatura máxima y mínima para visualizar la información deseada.
- Navegar a través de un mapa interactivo para seleccionar estados y municipios.

Figure 48: Ejemplo de la página principal mostrando el mapa interactivo y las opciones de filtro.

7.2.2 Mapa Interactivo con Escala de Colores

El mapa interactivo representa las temperaturas de manera geográfica, utilizando una escala de colores para identificar visualmente los rangos de temperatura:

- Los municipios están coloreados de acuerdo con su temperatura promedio en la fecha seleccionada.
- Los usuarios pueden hacer clic en un estado y posteriormente en un municipio para obtener detalles adicionales.

Figure 49: Ejemplo de la visualización de un estado específico en el mapa interactivo.

7.2.3 Gráfica de Tendencias Temporales

Al seleccionar un municipio, el sistema genera automáticamente una gráfica que muestra la evolución de las temperaturas:

- Se incluyen datos de dos días previos, el día seleccionado y dos días posteriores.

- La gráfica permite observar patrones o tendencias en la variación de temperaturas.

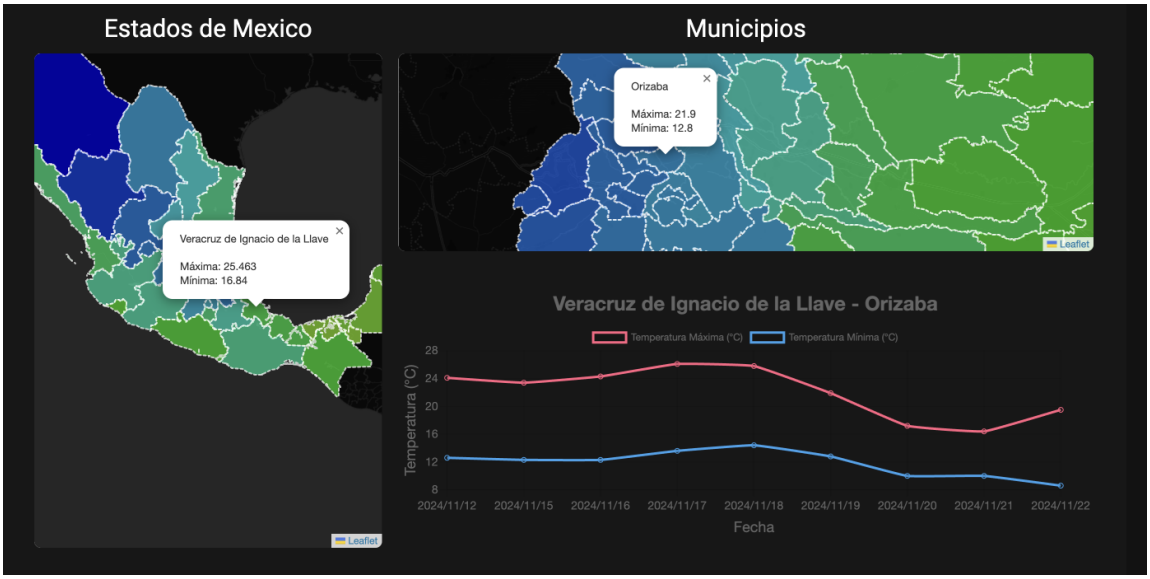


Figure 50: Ejemplo de la visualización de la tendencia temporal de temperaturas en un municipio específico.

7.3 Interacción del Usuario

El sistema está diseñado para garantizar que el usuario pueda interactuar fácilmente con los datos, permitiéndole:

1. Seleccionar el estado y municipio deseado a través del mapa interactivo, con una vista dinámica.
2. Explorar la temperatura en diferentes fechas con un simple ajuste de filtros.
3. Observar cambios de temperatura en un contexto temporal a través de gráficas detalladas.

7.4 Implementación Técnica

La visualización se implementó utilizando herramientas modernas de desarrollo web y bibliotecas especializadas para gráficos interactivos.

- El mapa interactivo se construyó utilizando `Leaflet.js`, una librería de JavaScript para mapas dinámicos.
- Las gráficas de tendencias se generaron con `Charts.js`, una herramienta versátil para visualización de datos.
- La integración de datos meteorológicos se gestionó con un flujo ETL que asegura la actualización diaria de la información.

8 Conclusiones

Conclusión 1: Aguilar Martinez Erick Yair

La implementación de este proyecto demuestra cómo un enfoque estructurado y basado en procesos ETL puede transformar datos crudos en información valiosa y procesable.

Al automatizar la extracción, transformación y carga de los datos climáticos, logramos estandarizar un flujo de información que anteriormente era difícil de gestionar. Esta solución no solo mejora la accesibilidad y organización de los datos, sino que también habilita nuevas oportunidades para análisis más profundos y aplicaciones prácticas en sectores como la planificación urbana, la agricultura y la gestión de riesgos climáticos. De esta manera, el proyecto no solo refuerza nuestra comprensión de conceptos técnicos, sino que también ilustra su aplicabilidad en la resolución de problemas reales.

Conclusión 2: Ahuatzi Pichardo Mariano Josué

Durante el desarrollo de este proyecto se implementaron diferentes herramientas en el análisis de información estructurada. Se aprovechó el servicio web de una institución de dependencia gubernamental en la que se ofrece información climatológica por municipio de toda la República Mexicana. Con esta información, al analizarla, se realizó un modelo entidad-relación del que se pudo derivar un modelo relacional y un modelo físico para implementarse en una base de datos. Conjuntamente, se realizó un proceso ETL para la limpieza y la estructuración de la información, para posteriormente visualizarla. En todo este proceso se pusieron en práctica diferentes herramientas de software, así como conceptos fundamentales para su buen funcionamiento, como la normalización en el modelado de datos o la estructuración para su consulta posterior. No está de más mencionar que estos conocimientos se adquirieron durante el curso de BDE. De esta forma, se logró la aplicación y la implementación de todas las herramientas adquiridas, con las que se pudo apreciar el flujo de la información y su estructuración.

Adicionalmente, el trabajo en equipo con diferentes puntos de vista y con conocimientos en distintos campos de conocimiento derivó en una estructura técnica sólida que mostró el desarrollo de un proyecto cooperativo. Además, se evidenciaron los retos y complicaciones durante el desarrollo, como la organización y el ritmo de trabajo.

Conclusión 3: Martinez Muñoz Alan Magno

Este proyecto pone de manifiesto cómo la tecnología puede ser un puente entre las políticas públicas y la sociedad. Al optimizar y estructurar los datos climáticos proporcionados por el Gobierno de México, hacemos accesible una fuente de información crucial que puede ser utilizada por comunidades, empresas y organismos públicos para tomar decisiones más informadas. Además, la estandarización de estos datos fomenta una sostenibilidad tecnológica, asegurando que la información sea almacenada y gestionada de manera eficiente, minimizando la pérdida de datos históricos. Así, el proyecto no solo representa un ejercicio técnico, sino una contribución tangible hacia un mejor uso de los recursos públicos en beneficio de la sociedad.

Conclusión 4: Mendoza Hernandez Carlos Emiliano

El desarrollo de este proyecto ha sido una oportunidad para consolidar los conocimientos adquiridos en la materia de Bases de Datos Estructuradas, aplicándolos a un problema real y relevante. Desde el análisis de los datos hasta la implementación de un proceso ETL con Pentaho, hemos experimentado todo el ciclo de vida de un proyecto de integración de datos. Este aprendizaje práctico nos prepara para abordar desafíos similares en entornos laborales, fortaleciendo nuestras habilidades en modelado de datos, automatización de flujos y gestión de información. En resumen, el proyecto no solo resuelve un problema técnico, sino que también enriquece nuestro desarrollo profesional, sentando las bases para futuros proyectos de mayor escala y complejidad.