



**INSTITUTO POLITÉCNICO  
NACIONAL**



**"ESCUELA SUPERIOR DE  
CÓMPUTO"**

**Integrantes:**

- Velasco Huerta Ángel Eduardo
- Hernández Clemente Samantha

**Práctica No. 4**

**Proceso de ETL, caso "precipitación pluvial en CDMX"**

**Materia:**

Data Mining

**Profesor:**

**Zagal Flores Roberto Eswart**

**Grupo:**

3CV18

## Introducción:











En la presente práctica, se aplicarán los conocimientos teóricos acerca del proceso de extracción, transformación y carga (ETL) de los metadatos obtenidos de la red de depósito atmosférico, en específico, los datos recabados de precipitación pluvial de 2010 a 2019. El objetivo es extraer la información presente en los archivos de terminación PPH. XLS, para después procesarla en una base de datos con una estructura más entendible y fácil de procesar, esto se realizará procesando la información presente en los archivos XLS, finalmente, se cargará la información para su posterior estudio.

Lo anterior mencionado, se realizará por medio de programación, se diseñará el ETL en lenguaje Python, para poder acceder a los archivos de Excel y poder establecer conexión con SQL Server para poder procesar y cargar información. Adicionalmente, se usarán 2 catálogos para darle estructura a la tabla principal de registros procesados.

## Desarrollo:

### 1 Y 2.- Revisión y análisis de los metadatos

Dentro de la página <http://www.aire.cdmx.gob.mx/default.php?opc=%27aKBk%27>, seleccionaremos los años del 2010 al 2019, y descargamos el archivo .zip correspondiente a cada año, posteriormente, extraemos solamente el correspondiente a PP, quedándonos solo 10 archivos de entrada:

Nombre	Fecha de modificación	Tipo	Tamaño
 2010PPH.xlsx	01/11/2021 09:09 p. m.	Hoja de cálculo d...	14 KB
 2011PPH.xlsx	01/11/2021 09:09 p. m.	Hoja de cálculo d...	14 KB
 2012PPH.xlsx	01/11/2021 09:18 p. m.	Hoja de cálculo d...	24 KB
 2013PPH.xlsx	01/11/2021 09:10 p. m.	Hoja de cálculo d...	23 KB
 2014PPH.xlsx	01/11/2021 09:10 p. m.	Hoja de cálculo d...	23 KB
 2015PPH.xlsx	01/11/2021 09:10 p. m.	Hoja de cálculo d...	23 KB
 2016PPH.xlsx	01/11/2021 09:11 p. m.	Hoja de cálculo d...	23 KB
 2017PPH.xlsx	01/11/2021 09:11 p. m.	Hoja de cálculo d...	23 KB
 2018PPH.xlsx	01/11/2021 09:11 p. m.	Hoja de cálculo d...	13 KB
 2019PPH.xlsx	01/11/2021 09:11 p. m.	Hoja de cálculo d...	13 KB

### 3.-Definición de flujo de trabajo del ETL

Ya con los archivos, es importante empezar a explorarlos manualmente, para poder definir correctamente las reglas a seguir en el ETL y el flujo de trabajo en general. Al abrir el archivo, veremos un formato como este:

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R
1	FECHA	LOM	TEC	DIC	MCM	TLA	XAL	EDL	IBM	NEZ	MON	EAJ	AJU	MPA	SNT	COR	LAA	
2	06/01/2014	-99	-99	-99	-99	-99	-99	-99	-99	-99	-99	-99	-99	-99	-99	-99	-99	
3	13/01/2014	-99	-99	-99	-99	-99	-99	-99	-99	-99	-99	-99	-99	-99	-99	-99	-99	
4	20/01/2014	-99	-99	-99	-99	-99	-99	-99	-99	-99	-99	-99	-99	-99	-99	-99	-99	
5	27/01/2014	-99	-99	-99	-99	-99	-99	-99	-99	-99	-99	-99	-99	-99	-99	-99	-99	
6	03/02/2014	-99	-99	-99	-99	-99	-99	-99	-99	-99	-99	-99	-99	-99	-99	-99	-99	
7	10/02/2014	-99	-99	-99	-99	-99	-99	-99	-99	-99	-99	-99	-99	-99	-99	-99	-99	
8	17/02/2014	-99	-99	-99	-99	-99	-99	-99	-99	-99	-99	-99	-99	-99	-99	-99	-99	
9	24/02/2014	-99	-99	-99	-99	-99	-99	-99	-99	-99	-99	-99	-99	-99	-99	-99	-99	
10	03/03/2014	-99	-99	-99	-99	-99	-99	-99	-99	-99	-99	-99	-99	-99	-99	-99	-99	
11	10/03/2014	-99	-99	-99	-99	-99	-99	-99	-99	-99	-99	-99	-99	-99	-99	-99	-99	
12	17/03/2014	-99	-99	-99	-99	-99	-99	-99	-99	-99	-99	-99	-99	-99	-99	-99	-99	
13	24/03/2014	-99	-99	-99	-99	-99	-99	-99	-99	-99	-99	-99	-99	-99	-99	-99	-99	
14	31/03/2014	-99	-99	-99	-99	-99	-99	-99	-99	-99	-99	-99	-99	-99	-99	-99	-99	
15	07/04/2014	-99	-99	-99	-99	-99	-99	-99	-99	-99	-99	-99	-99	-99	-99	-99	-99	
16	14/04/2014	-99	-99	-99	-99	-99	-99	-99	-99	-99	-99	-99	-99	-99	-99	-99	-99	
17	21/04/2014	-99	-99	-99	-99	-99	-99	-99	-99	-99	-99	-99	-99	-99	-99	-99	-99	
18	28/04/2014	-99	12.89	26.71	22.10	16.12	13.20	-99	14.28	10.75	13.66	23.64	-99	25.33	-99	8.44	-99	
19	05/05/2014	-99	11.67	16.58	10.13	14.12	12.59	45.13	23.33	8.90	31.78	23.03	10.28	8.14	-99	6.45	9.82	
20	12/05/2014	-99	3.53	-99	11.21	-99	3.38	19.03	-99	9.06	8.75	-99	-99	12.89	-99	7.83	5.68	
21	19/05/2014	86.58	5.37	31.93	22.41	25.02	11.67	46.36	86.73	9.21	3.99	51.58	111.45	11.05	30.09	11.05	19.96	
22	26/05/2014	52.96	17.65	15.20	28.09	24.41	13.82	84.27	45.44	70.77	23.03	30.39	-99	49.58	51.89	36.07	28.71	
23	02/06/2014	64.63	16.43	44.82	30.70	60.63	28.86	66.62	31.47	12.43	13.51	60.79	-99	15.04	58.79	27.78	22.87	
24	09/06/2014	36.23	21.95	9.21	39.14	56.80	27.17	29.78	71.23	1.38	9.82	4.61	34.85	34.23	-99	26.40	10.28	
25	16/06/2014	137.54	111.29	54.96	115.90	142.30	109.14	79.06	145.98	98.55	76.14	66.47	77.21	54.80	-99	38.68	109.60	
26	23/06/2014	33.46	110.83	28.71	58.33	65.85	74.14	46.05	31.47	34.85	18.27	48.20	37.92	29.78	31.93	48.66	173.46	
27	30/06/2014	55.88	52.04	84.74	53.42	60.33	53.73	65.70	65.55	46.67	96.71	73.53	83.35	34.23	88.57	31.16	44.06	
28	07/07/2014	54.03	50.66	13.51	24.41	36.38	36.53	48.35	30.39	-99	27.94	58.49	20.42	15.66	57.26	25.79	-99	
29	14/07/2014	-99	92.87	8.44	55.57	26.40	73.68	9.67	40.22	-99	-99	31.47	23.33	15.81	44.21	-99	84.89	

Analizando el formato de los archivos, vemos que se nos presenta una fecha, y las lecturas correspondientes a cada estación de la red, por ello, es que crearemos una tupla de información por cada uno de estas mediciones, pues es más fácil de trabajar posteriormente si cada registro cuenta con su propia información.

Definiendo el flujo de trabajo del ETL, se puede definir lo siguiente:

- Del nombre del archivo, podemos extraer el año, aminorará el proceso en el código pues para cada archivo es un año.
- De la fecha, podemos extraer el día y el mes de la medición.
- Ya contando con la fecha segmentada en partes, podemos obtener la semana mediante funciones integradas de calendario.
- Tenemos la abreviatura de la estación, para darle una buena estructura a nuestra base, se asociará a un catálogo de estaciones, de esa manera solo tendremos un ID de la estación correspondiente.
- La medición se registrará normalmente, sin embargo, en las reglas definidas por la misma red de depósito, los registros con medición -99 representan valores nulos, por lo cual no se registraron dentro de la base.
- Para todas las mediciones, el elemento es el mismo, y se puede obtener del nombre del archivo, de la misma manera, se puede estructurar mejor con un catálogo.

#### 4.- Desarrolla la estructura de la tabla de hechos principal

Antes de empezar a codificar el ETL, crearemos la base de datos correspondiente, así como los catálogos necesarios para el procesamiento de la información.

Primero crearemos un catálogo de estaciones, la información puede ser obtenida de la siguiente liga: <http://www.aire.cdmx.gob.mx/default.php?opc=%27ZaBhnml=&dc=%27ZA==> el catálogo contará con:

- ID
- Estación
- Nombre

- Ubicación

quedará de la siguiente forma:

	ID	Estacion	Nombre	Ubicacion
1	1	ACO	Acolman	Acolman
2	2	LOM	Lomas	Miguel Hidalgo
3	3	TEC	Cerro del Tepeyac	Gustavo A. Madero
4	4	DIC	Diconsa	Tlalpan
5	5	MCM	Museo de la Ciudad de México	Cuauhtémoc
6	6	TLA	Tlalnepantla	Tlalnepantla de Baz
7	7	XAL	Xalostoc	Ecatepec de Morelos
8	8	EDL	Ex Convento Desierto de los Leones	Cuajimalpa de Morelos
9	9	IBM	Legaria	Miguel Hidalgo
10	10	NEZ	Nezahualcóyotl	Nezahualcóyotl
11	11	MON	Montecillo	Texcoco
12	12	EAJ	Ecoguardas Ajusco	Tlalpan
13	13	AJU	Ajusco	Tlalpan
14	14	MPA	Milpa Alta	Milpa Alta
15	15	SNT	San Nicolás Totolapan	La Magdalena Contreras
16	16	COR	CORENA	Xochimilco

También se elaborará un catálogo de elementos de medición, definido así:

- ID
- Elemento de medición
- Abreviatura
- Concentración

quedará de la siguiente forma:

	ID	Elemento_medicion	Abreviatura	Concentracion
1	1	Precipitacion Pluvial	PP	Milímetros de lluvia (mm)


De momento solo se registró el elemento de precipitación pluvial, pues para el diseño del ETL a medida, solo se requiere este, sin embargo, si en un futuro se quiere ampliar este ETL para poder extraer información de los archivos correspondientes a otros elementos de desecho ambiental, solo habría que ampliar este catálogo.

Finalmente, la tabla de hechos principales se diseñó de la siguiente manera:

- ID\_Registro: No importa, es un ID con auto-incremento
- ID\_Elemento: Representa el ID del elemento en el catálogo
- Año\_Registro: Año obtenido del archivo
- Mes\_Registro: Mes obtenido de la fecha de registro
- Dia: día obtenido de la fecha de registro.
- Semana\_Registro: Número de semana del año calculada a partir de la fecha
- ID\_Estacion: Representa el ID de la estación en el catálogo

- Valor\_medicion: la medición registrada.

Quedará de la siguiente forma:

	ID_Registro	int	<input type="checkbox"/>
	ID_Elemento	int	<input checked="" type="checkbox"/>
	Año_Registro	int	<input checked="" type="checkbox"/>
	Mes_Registro	int	<input checked="" type="checkbox"/>
	Dia	int	<input checked="" type="checkbox"/>
	Semana_Registro	int	<input checked="" type="checkbox"/>
	ID_Estacion	varchar(200)	<input checked="" type="checkbox"/>
	Valor_Medicion	int	<input checked="" type="checkbox"/>

ID_Registro	ID_Elemento	Año_Registro	Mes_Registro	Dia	Semana_Registro	ID_Estacion	Valor_Medicion
-------------	-------------	--------------	--------------	-----	-----------------	-------------	----------------

Una vez implementado el ETL, (Más adelante se explica el funcionamiento y estructura del ETL programado) tendremos los registros de la siguiente forma:

	ID_Registro	ID_Elemento	Año_Registro	Mes_Registro	Dia	Semana_Registro	ID_Estacion	Valor_Medicion
1	1	1	2010	5	3	18	6	5.59
2	2	1	2010	5	3	18	7	7.06
3	3	1	2010	5	3	18	8	17.76
4	4	1	2010	5	3	18	12	9.98
5	5	1	2010	5	3	18	13	0.31
6	6	1	2010	5	3	18	15	2.61
7	7	1	2010	5	3	18	17	0.71
8	8	1	2010	5	10	19	3	7.58
9	9	1	2010	5	10	19	6	5.53
10	10	1	2010	5	10	19	7	5.89
11	11	1	2010	5	10	19	8	53.18
12	12	1	2010	5	10	19	9	10.75
13	13	1	2010	5	10	19	10	5.37
14	14	1	2010	5	10	19	12	17.46
15	15	1	2010	5	10	19	13	15.66
16	16	1	2010	5	10	19	14	8.48
17	17	1	2010	5	10	19	15	26.44
18	18	1	2010	5	10	19	16	2.68

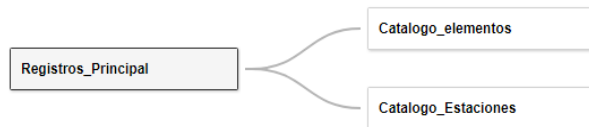
Ya con los datos transformados y cargados, podemos realizar la exploración de los mismos.

## 5- Exploración de datos integrados

Aprovecharemos que los datos ya están cargados correctamente, y realizaremos una exploración básica con Tableau.

Primero, cargamos las tablas para establecer relación entre los catálogos y la tabla principal

PPH\_10-19



Ordenar campos <span>Orden de fuente de datos</span>								
# Registros_Principal ID Registro	# Registros_Principal ID Elemento	# Registros_Principal Año Registro	# Registros_Principal Mes Registro	# Registros_Principal Día	# Registros_Principal Semana Registro	# Registros_Principal ID Estacion	# Registros_Principal Valor Medicion	
1	1	2010	5	3	18	6	5.590	
2	1	2010	5	3	18	7	7.060	
3	1	2010	5	3	18	8	17.760	
4	1	2010	5	3	18	12	9.980	
5	1	2010	5	3	18	13	0.310	
6	1	2010	5	3	18	15	2.610	
7	1	2010	5	3	18	17	0.710	
8	1	2010	5	10	19	3	7.580	
9	1	2010	5	10	19	6	5.530	
10	1	2010	5	10	19	7	5.890	
11	1	2010	5	10	19	8	53.180	
12	1	2010	5	10	19	9	10.750	

Ahora, podemos empezar con los libros:

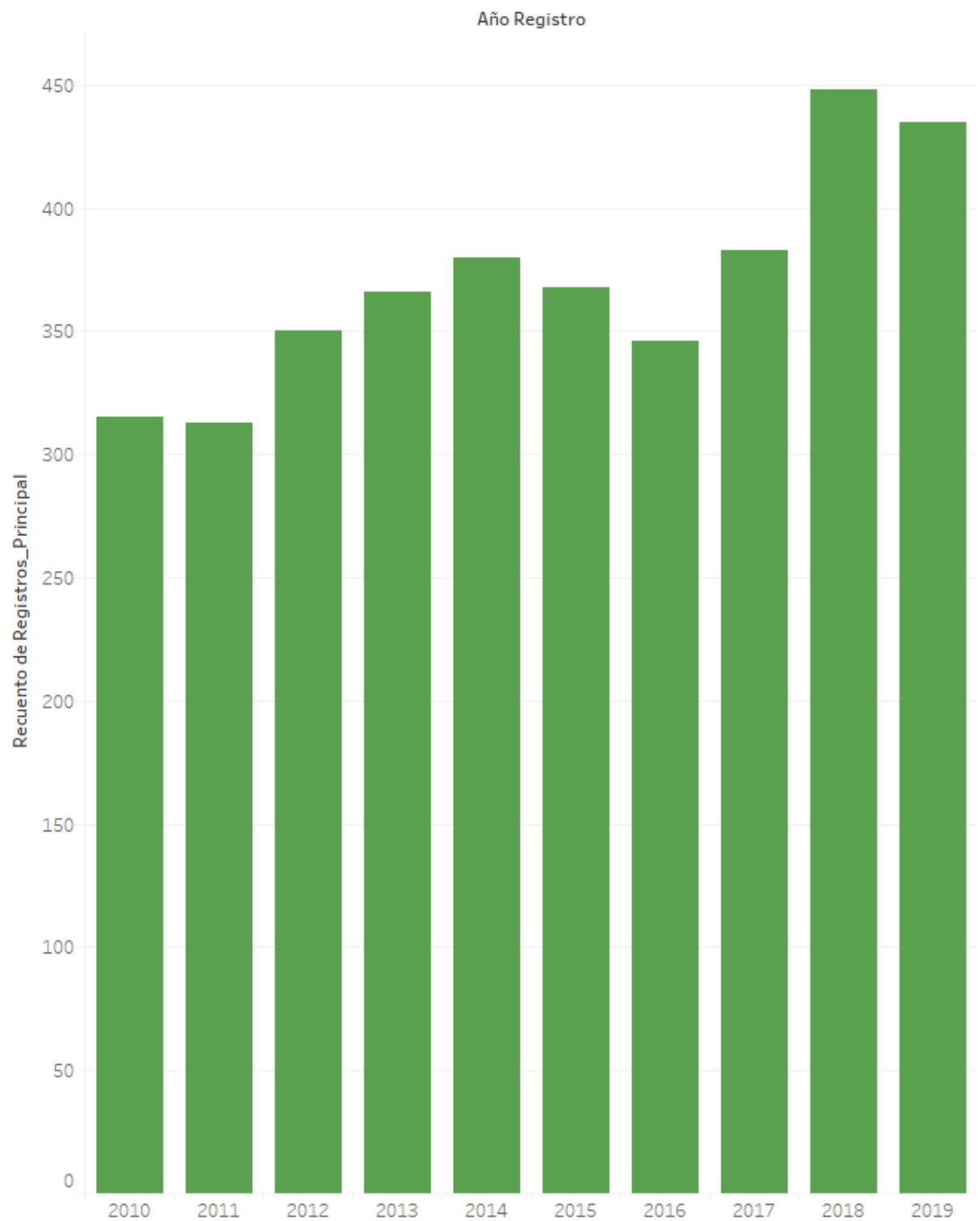
1. Cantidad de registros totales y por año

```
SELECT COUNT(*) AS TOTAL FROM REGISTROS_PRINCIPALES
```

	TOTAL
1	3704

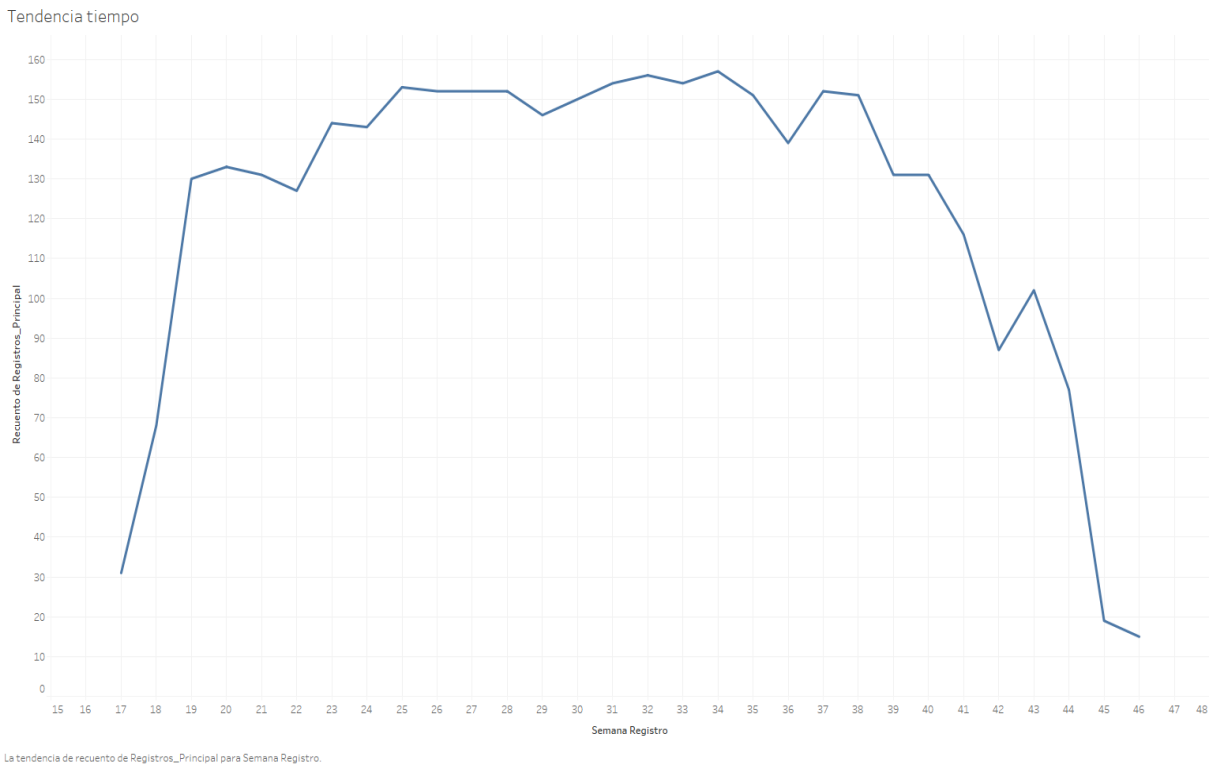
2. Cantidad de registros por año:

Registros\_Año

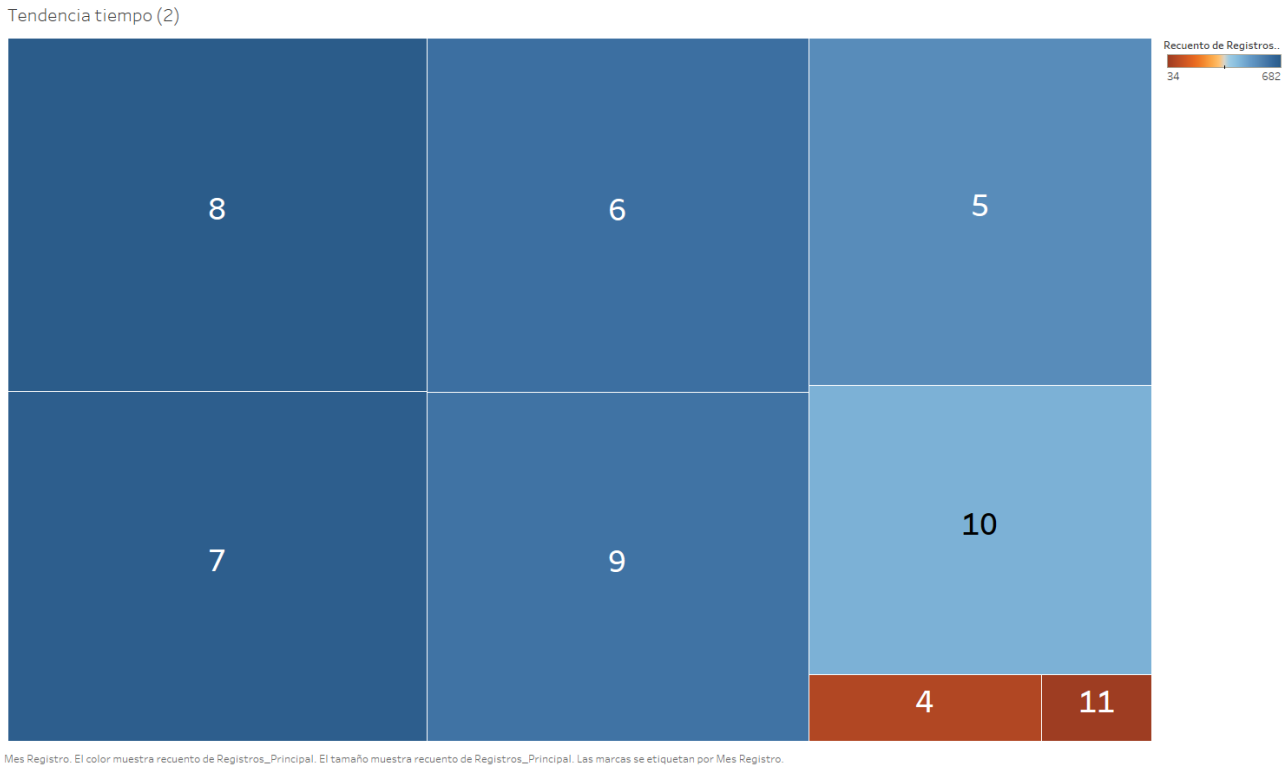


Recuento de Registros\_Principal para cada Año Registro.

b- Tendencia en el tiempo de la precipitación pluvial



por semana

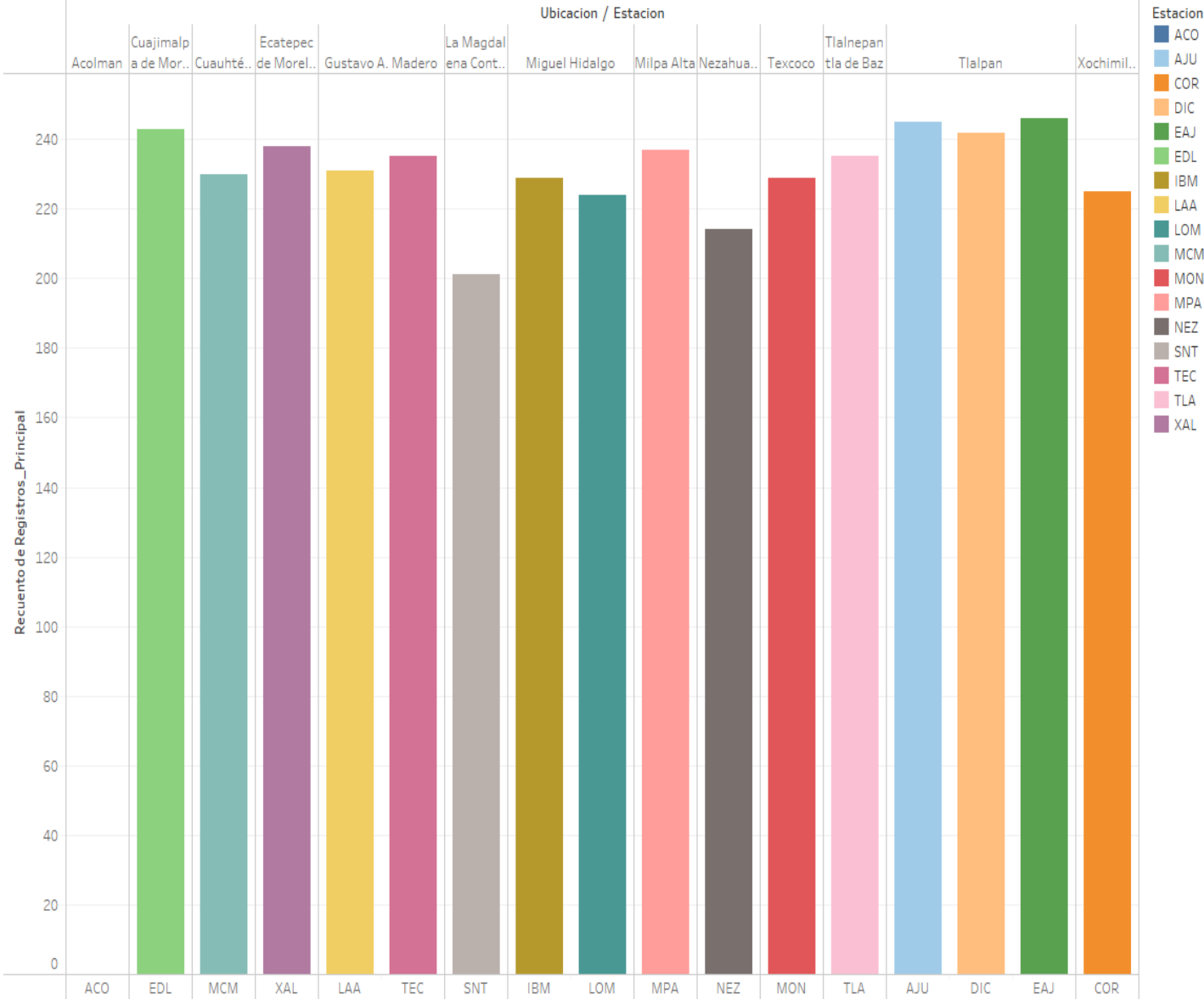


por mes



C-Indique los lugares con mayor precipitación durante todo el periodo de estudio

LUGAR



Recuento de Registros\_Principal para cada Estacion desglosado por Ubicacion. El color muestra detalles acerca de Estacion.

## 6- Documentación del código del ETL:

El ETL, se realizará en Python, pues ofrece muchas funciones integradas para el manejo de cadenas, así como muchas librerías para integración de datos.

Comenzaremos agregando las siguientes Librerías:

```
import os
from openpyxl import load_workbook
from openpyxl.descriptors import base
from openpyxl.descriptors.base import DateTime
import pyodbc
import datetime
```

openpyxl nos permite abrir y manejar libros XLS Y XLSX

pyodbc nos permite establecer una conexión segura con SQL

```
def getId_estacion(estacion):
    cursor.execute(
        f"select ID from Catalogo_Estaciones where Estacion = '{estacion}';")
    while 1:
        row = cursor.fetchone()
        if not row:
            break

        variable = row[0]
    return variable

# Encontramos el ID del elemento

def getId_elemento(elemento):
    cursor.execute(
        f"SELECT ID FROM Catalogo_elementos where Abreviatura = '{elemento}';")
    while 1:
        row = cursor.fetchone()
        if not row:
            break

        variable = row[0]
    return variable

# Buscamos la cantidad de archivos
```

Se definen tres funciones, dos de ellas son para transformar los datos, nos es más fácil registrar solo enteros en la tabla, por eso es que las funciones buscan en los respectivos catálogos para encontrar el ID e insertar ese en la tupla.

```
def buscar_archivos(ruta):
    archivos_texto = []
    archivos = os.listdir(ruta)
    for archivo in archivos:
        if archivo[-5:] == '.xlsx':
            archivos_texto.append(archivo)
    return archivos_texto
```

La tercera función nos sirve para realizar la exploración de los archivos, se le introduce una ruta, e inserta en una lista los archivos que encontró.

```
# CREAMOS LA CONEXION SEGURA CON SQL
connection = pyodbc.connect(
    'DRIVER={ODBC Driver 17 for SQL Server};SERVER=ANGEL\SQLEXPRESS;DATABASE=PPH_10-19;Trusted_Connection=yes;')
cursor = connection.cursor()
```

Ya dentro del programa principal, lo primero que se hace, es establecer una conexión con SQL Server, y crear un cursor para ejecutar sentencias SQL.

```
# recorremos los archivos uno por uno.
for archivo in buscar_archivos(folder):
    print(f"año: {archivo[0:4]}")
    wb = load_workbook(folder+"/"+archivo, data_only=True)
    ws = wb[archivo[0:7]]
    año = int(archivo[0:4])
    IDelemento = getId_elemento(str(archivo[4:6]))
    for x in range(53):
        if (año < 2018):
            mes = str(ws['A'+str(x+2)].value)[5:7]
            dia = str(ws['A'+str(x+2)].value)[8:10]
        else:
            mes = str(ws['A'+str(x+2)].value)[3:5]
            dia = str(ws['A'+str(x+2)].value)[0:2]

        if(dia == '' or dia == 'No' or dia == ' '):
            dia = 11
            mes = 11
        semana_registro = datetime.date(
            año, int(mes), int(dia)).isocalendar()[1]
        for letra in 'BCDEFGHIJKLMNOPQ':
            IDestacion = getId_estacion(str(ws[letra+'1'].value))
            medicion = str(ws[letra+str(x+2)].value)
            if(medicion == '-99' or medicion == '' or medicion == 'None' or medicion == ' '):
                pass
            else:
                tupla = (int(IDelemento), int(año), int(mes), int(dia),
                    int(semana_registro), int(IDestacion), float(medicion))
```

El siguiente paso representa la fase de extracción y transformación de los datos, primero iteramos los archivos encontrados, anidamos un for con el número máximo de filas en los

archivos, y dentro, empezamos a descomponer las fechas y los nombres del archivo, esto mediante slicing, para poder separar año, mes, día y encontrar la semana, así como para encontrar que tipo de elemento es, esto se obtiene del archivo.

Ahora iteramos sobre las columnas, y sobre cada registro no nulo (! = -99), creamos una tupla con todos los elementos.

```
with open('PPH_10-19', 'a') as f:
    f.write(str(tupla)[1:-1]+"\\n")
```

Abrimos un archivo plano, y escribimos línea por línea, la tupla obtenida y formateada.

```
print("-----")
f = open("PPH_10-19", "r")
print("Insertando los valores en la base de datos...")
contador = 0
with connection.cursor() as cursor2:
    for linea in f:
        print(linea)
        cursor2.execute(f"INSERT INTO Registros_Principal
                           VALUES({linea});")
        contador = contador + 1
f.close()
print(f"Todos los registros fueron insertados {contador} lineas afectadas")
# CERRAMOS LA CONEXIÓN CON SQL

cursor.close()
connection.close()
```

Finalmente, entramos en la etapa de carga, leyendo todos los registros del archivo previamente creado, insertamos uno por uno con un nuevo cursor, como la tupla ya está formateada para que la tabla principal la acepte, la inserción es fácil.

Para terminar el programa, arrojamus los resultados cargados así como el número de registros completados.

Ejecución del programa:

Para este caso, no se realizó una interfaz, sin embargo, el programa notifica partes importantes del proceso.

```
PS C:\Users\Angel\Desktop\Data-mining> python ETL.py
año: 2010
año: 2011
año: 2012
año: 2013
año: 2014
año: 2015
año: 2016
año: 2017
año: 2018
```

Al ejecutarlo, nos muestra los resultados de la extracción de datos, imprimiendo el año de cada archivo leído

The screenshot shows a Jupyter Notebook environment with a code cell and a terminal window. The code in the notebook is as follows:

```
81     tupla = (int(IDelemento), int(año), int(mes),
82             int(semestre_registro), int(IDestacion))
83     with open('PPH_10-19', 'a') as f:
84         f.write(str(tupla)[1:-1]+"\n")
85
86 # una vez que se creo el archivo, sigue la carga de datos den
87
88 print("-----")
89 f = open("PPH_10-19", "r")
90 print("Insertando los valores en la base de datos...")
91 contador = 0
92 with connection.cursor() as cursor2:
93     for linea in f:
94         print(linea)
95         cursor2.execute(f"INSERT INTO Registros_Principal
96                         VALUES({linea});")
97         contador = contador + 1
98 f.close()
99 print(f"Todos los registros fueron insertados {contador} linea
100 # CERRAMOS LA CONEXIÓN CON SQL
101
102 cursor.close()
```

The terminal window displays the output of the code, showing the first few lines of the data file and the total number of records inserted:

```
1, 2019, 11, 11, 46, 4, 3.5
1, 2019, 11, 11, 46, 6, 2.2
1, 2019, 11, 11, 46, 8, 3.8
1, 2019, 11, 11, 46, 9, 2.5
1, 2019, 11, 11, 46, 12, 1.8
1, 2019, 11, 11, 46, 15, 1.6

Todos los registros fueron insertados 3704 lineas afectadas
PS C:\Users\Angel\Desktop\Data-mining>
```

Finalmente, así es como se visualiza, llena el archivo plano, e imprime en pantalla cada tupla correctamente insertada, así como el resultado de inserciones.

## VALORES ATÍPICOS Y PROBLEMAS:

- El primer problema al momento de realizar este ETL, es unificar el formato de la fecha, a partir de 2018, el archivo contiene un formato de fecha diferente al momento de extraerlo, lo cual creaba muchas inconsistencias en la transformación de datos, afortunadamente, al notar esto, fue sencillo realizar una verificación extra en el código para unificar el formato sin tener que modificar los archivos base.
- Parece haber muchos nulos en los primeros registros de cada año.
- No existen registros en diciembre.
- Son muchos los valores -99 de cada año (Nulos)

## Conclusiones

**Velasco Huerta Ángel Eduardo:** Creo que el desarrollo de esta práctica, nos ayudó demasiado a reforzar esta parte teórica del proceso de ETL, si bien considero que la solución presentada cumple bien a la problemática inicial, definir varias partes del proceso se nos complicó, principalmente la parte de cómo definir la estructura de la tabla principal, posteriormente la unificación del formato de la fecha causó problemas, desconocemos si es por la biblioteca usada o no, pero estos problemas nos ayudaron a comprender más esta etapa de transformación, pues vemos que no necesariamente todos los datos están representados de la misma manera, y si queremos realizar algún análisis posterior, debemos limpiar los datos.

La exploración posterior al ETL fue muy sencilla, el tener la base estructurada de esa manera, nos ayudó a encontrar toda la información que requeríamos para resolver ciertas preguntas que pudieran surgir de la información, asimismo, el programa podría funcionar

bien para otros depósitos ambientales, pues por la misma estructura, permite registrar más cosas.

**Hernández Clemente Samantha:** Al desarrollar esta práctica, nos permitió entender en mayor medida el proceso de llevar a cabo el desarrollo de un ETL, pese a tener los conocimientos teóricos y comprender todo el proceso que se tiene que seguir, a llevarlo a la práctica, se nos presentaron algunas complicaciones, a mi en especial en la parte de definir unas partes del proceso de ETL para que el programa extrajera los datos de todos los archivos.

Ya que logramos extraer, transformar y añadir esos datos de forma correcta, fue todo mucho más sencillo y los puntos posteriores no presentaron una gran dificultad a comparación de al principio.