



## Informe 2. Segundo avance

Luisa Fernanda Buriticá & Marcela Echeverri

Grupo de investigación FACOM

21 de abril de 2022

Universidad de Antioquia

---

El presente informe tiene como objetivo profundizar en el trabajo realizado por las estudiantes Luisa Fernanda Buriticá y Marcela Echeverri hasta el 21 de abril de 2022.

En primer lugar se generaron 2 bases de datos con extensiones csv para las variables meteorológicas de temperatura en unidades de grados Celsius y precipitación en unidades de milímetros; estas variables tenían información de distintas estaciones meteorológicas ubicadas en varios departamentos y municipios de Colombia, y algunos en diferentes rangos de tiempo; se tiene como primer objetivo caracterizar ambos archivos csv, revisando sus información básica, verificando el orden, los formatos de cada fila y columna, y analizar los datos por medio de sus estadísticas básicas, principalmente por medio de la librería Pandas en python (Todas las funciones y códigos usados para alcanzar el objetivo se adjuntan en un archivo ubicado en el github al final del informe (ver anexo 1)). Finalmente, usando la librería de sqlalchemy, y a partir de los archivos csv suministrados, se generó una base de datos llamada "DATA.db" que permitió avanzar en el análisis de las dos variables con el lenguaje de SQL por medio de búsquedas localizadas que da como resultado un archivo csv, con información básica, como el nombre y número de columnas, estadística como el promedio, mediana y desviación estándar. A continuación se presenta la información inicial de ambos archivos:

### **Base de datos número 1:**

**Nombre del archivo:** Datos\_Hidrometeorol\_gicos\_Crudos\_-  
\_Red\_de\_Estaciones\_IDEAM\_\_\_Temperatura.csv

**Variable:** Temperatura

**Unidades:** Celsius [°C]

**Tamaño:** 8,7 G

**Número de columnas:** 12

**Número de filas:** 63'946.972

**Uso de memoria por columna:** 511'575.896 Bytes

El archivo contiene datos de temperatura en °C, registrados por 544 estaciones meteorológicas alrededor del país, con un muestreo temporal de aproximadamente 12 horas, además, se encuentra información acerca de la zona hidrográfica, longitud, latitud,

departamento y municipio. La fecha inicial del archivo completo es el **01 de Enero de 2001 a las 0:19:05 horas** y la fecha final para la muestra, es el **13 de Diciembre de 2021 a las 23:55:00 horas**, sin embargo, cada una de las estaciones meteorológicas tiene fechas iniciales, finales y pasos de tiempo diferentes, por lo que el análisis principal se realizará por estaciones. En el IDEAM se siguen actualizando los datos, por lo que en el futuro se buscará registrar y ordenar la información en tiempo real para mantener la base de datos actualizada.

### **Base de datos número 2:**

#### **Nombre del archivo:**

Precipitaci\_n.csv

**Variable:** Precipitación

**Unidades:** Milímetros [mm]

**Tamaño:** 21,1G

**Número de columnas:** 12

**Número de filas:** 160'665.056

**Uso de memoria por columna:** 160'665054 Bytes

El archivo contiene datos de precipitación en mm, registrados por 790 estaciones meteorológicas alrededor del país, se encuentra información acerca de la zona hidrográfica, longitud, latitud, departamento y municipio de la toma de la medición. La fecha inicial del archivo completo es el **02 de Enero de 2003 a las 15:20:00 horas** y la fecha final de la muestra de datos es el **13 de Diciembre de 2021 a las 23:59:00 horas**, sin embargo, al igual que con la primera base de datos, cada estación tiene intervalos de tiempo diferentes y el muestreo de los datos de precipitación también cambiar dependiendo de la estación, además en el IDEAM se siguen actualizando los datos, por lo que en el futuro se buscará registrar y ordenar la información en tiempo real para mantener la base de datos actualizada.

---

### **Columnas**

#### **Tabla 1.**

*Descripción de la muestra de temperatura*

C	Nombre	Tipo	Descripción
0	CodigoEstacion	int64	La Precipitación se registra en 550 estaciones y la Temperatura en 790; el filtro se genera con las estaciones existentes en ambas tablas.
1	CodigoSensor	int64	68
2	FechaObservacion	Object	Cada estación tiene un rango de fechas y un paso de tiempo (por eso algunos pasos de tiempo difieren, aún no se analiza si varía el paso de tiempo en alguna estación).

3	ValorObservado	float64	Los datos nulos se encuentran como <nil>, aún no se determina el rango que considere la naturaleza de la variable.
4	NombreEstacion	Object	Corrección antes de ingresar a la base de datos, por ejemplo de las comas, mayúsculas y diferentes palabras que se refieran a lo mismo.
5	Departamento	Object	Corrección antes de ingresar a la base de datos, por ejemplo de las comas, mayúsculas y diferentes palabras que se refieran a lo mismo.
6	Municipio	Object	Corrección antes de ingresar a la base de datos, por ejemplo de las comas, mayúsculas y diferentes palabras que se refieran a lo mismo.
7	ZonaHidrografica	Object	Corrección antes de ingresar a la base de datos, por ejemplo de las comas, mayúsculas y diferentes palabras que se refieran a lo mismo.
8	Latitud	float64	cada estación tiene una única latitud
9	Longitud	float64	cada estación tiene una única longitud
10	DescripcionSensor	Object	precipitación/temperatura
11	UnidadMedida	Object	milímetros

*C contiene el número de columnas, Nombre, tiene la información del nombre de la columna de la muestra de datos dentro del archivo csv y Descripción da un resumen de lo analizado hasta el momento para la columna.*



**Grafico 1. Estaciones**

---

## Lo que ya se hizo

- Se ordenaron los archivos csv para corregir errores de tipeo (como mayúsculas, tildes y ñ) en las columnas 4,5,6 y 7 correspondientes a NombreEstacion, Departamento, Municipio y ZonaHidrografica. En el proceso se encontró que los datos nulos están definidos como “<nil>”.
- Se generó una base de datos provisional en SQLite con la información completa de los archivos csv descargados desde la página del IDEAM y sin hacer ningún tipo de corrección (la base de datos definitiva con las correcciones pertinentes será generada al corregir el error mencionado más adelante). La base de datos, llamada DATA.db contiene 2 tablas, una de precipitación y la otra de temperatura, para hacer la generación de la base de datos se cargó el archivo csv por porciones, es decir, se cargó el 0,1% de las filas, luego el siguiente 0,1% y así sucesivamente hasta finalizar la carga de los datos, el porcentaje de subida se puede definir desde el código fuente.
- De la base de datos provisional se saca la información básica por estación como la fecha inicial y final, la cantidad de filas y columnas por estación, las primeras y últimas filas, la longitud y latitud, el municipio, departamento, zona hidrográfica, nombre de estación, unidades del valor observado y su descripción.
- Para el valor observado por estación se realizó un análisis de los estadísticos principales (valor máximo, mínimo y medio, desviación estándar y mediana para temperatura y precipitación), con ello se pretende iniciar la etapa de graficación y análisis de los valores físicos de la base de datos. Se generó el primer gráfico del ciclo medio anual por estación.

## Lo que falta por hacer

- Se pretende cambiar la definición de los datos nulos para fines prácticos de “<nil>” a “Nan”, debido a que nos resulta más intuitivo, también es necesario hacer un conteo de los datos nulos.
- Falta definir el límite de los datos atípicos para los valores observados de temperatura y precipitación, que serán corregidos como datos perdidos.
- Para generar finalmente la base de datos en SQLite desde python se debe solucionar un error de la máquina de la base de datos de la tabla temperatura y hacer la corrección de tipo de los nombres y los datos nulos para cada porcentaje del archivo csv, es decir, que cada trozo del archivo que se va a subir a SQLite sea corregido antes de ser añadido a la base de datos.
- Cuando la base de datos sea finalmente generada, ésta entrega información de cada una de las estaciones, dicha información debe ser guardada en conjunto en un archivo csv o similar, para poder acceder a ella fácilmente.

Teniendo la información básica de ambas bases de datos se comenzó con el análisis de las variables y esto es lo que se ha encontrado hasta el momento:

- Se buscó afirmar que no hubiera filas desplazadas a la izquierda o a la derecha (si las hubiera, significa que algún dato, en alguna columna, no está correctamente almacenado o no existe, por lo que se debería revisar en detalle esos casos particulares), se verificó identificando que el tipo de variable por columna fuera igual al tipo de variable dato a dato, y se hizo la suposición de que la primera fila de la base de datos tiene los formatos correctos para hacer una comparación, finalmente hasta ahora las columnas revisadas han cumplido con la suposición inicial.
- Se encontraron varias fechas y horas repetidas, esto se debe a que cada estación toma sus propias mediciones independientes las unas de las otras, por esa razón, en teoría, cada estación debe tener fechas y horas únicas; el muestreo también puede cambiar dependiendo del tipo de sensor que se use.
- Sobre el tiempo, la base de datos de precipitación tiene un muestreo de datos diferente para ciertos intervalos de tiempo, por lo que se busca agrupar los datos por estaciones para luego hacer un acople e identificar las fechas en las que el muestreo cambia, y abrir la posibilidad de que haya otros muestreos diferentes que en un primer momento, no fueron detectados. Luego, se busca ordenar la información de tal forma que todos los datos del archivo tengan el mismo muestreo temporal, identificando la mejor estrategia para llevarlos a un paso de tiempo de 10 min.

- Como se puede observar en la tabla 1 y 2, la columna de los departamentos tiene más elementos únicos que el número de departamentos en Colombia (32 departamentos), esto se debe a que hay datos repetidos que se diferencian con errores de gramática (tildes, mayúsculas, puntos, espacios, etc), la solución es crear listas de chequeo y de revisión que conviertan todas las posibilidades en una sola (esperamos que sea retroalimentativo) para guardar todos los errores que encontremos por región y corregirlos de forma automatizada. Esto también aplica para los municipios.

**Anexos:**

**Anexo 1. link de la carpeta de github**

<https://github.com/marcelaeyh/FACOM.git>