

Integración de Datos IoT con GIS usando Python

Documentación del proceso para realizar una aplicación IoT con GIS (TELEMÁTICA).

1. **Desarrollo de un Script en Python:**
2. **Integración GIS:**

Para cumplir el requerimiento número 1, se decidió simular los datos de sensores de calidad de aire. Por simplicidad consideraré las mediciones de ozono a 8 h(O3) en ppm y monóxido de carbono a 8h en ppm.

Ahora, para poder clasificar los datos en función de que tan bueno o malo es la calidad, he tomado como referencia una tabla proporcionada por Solectroshop (<https://solectroshop.com/es/blog/sensores-de-calidad-del-aire-parametros-niveles-y-caracteristicas-n79>)

ICA	COLOR	CLASIFICACIÓN	O3 8h ppm	O3 1h ppm	PM ₁₀ 24h µg/m ³	PM _{2.5} 24h µg/m ³	CO 8h ppm	SO ₂ 24h ppm	NO ₂ 1h ppm
0 - 50	Verde	Buena	0.000 0.059	-	0 54	0 12	0 4.4	0 0.035	0 0.053
51 - 100	Amarillo	Moderada	0.060 0.075	-	55 154	12.1 35.4	4.5 9.4	0.036 0.075	0.054 0.100
101 - 150	Naranja	Dañina a la salud para grupos sensibles	0.076 0.095	0.125 0.164	155 254	35.5 55.4	9.5 12.4	0.076 0.185	0.101 0.360
151 - 200	Rojo	Dañina a la salud	0.096 0.115	0.165 0.204	255 354	55.5 150.4	12.5 15.4	0.186 0.304	0.361 0.649
201 - 300	Púrpura	Muy Dañina a la salud	0.116 0.374	0.205 0.404	355 424	150.5 250.4	15.5 30.4	0.305 0.604	0.650 1.249
301 - 400	Marrón	Peligrosa	-	0.405 0.504	425 504	250.5 350.4	30.5 40.4	0.605 0.804	1.250 1.649
401 - 500	Marrón	Peligrosa	-	0.505 0.604	505 604	350.5 500.4	40.5 50.4	0.805 1.004	1.650 2.049

Una vez entendido el 'modelo de negocio' o la clasificación de los datos, realizaré un script que pueda generar valores aleatorios de data (O3 y CO ambos a 8h) para 10 sensores. La idea es crear un archivo CSV donde se encuentren los datos con el siguiente formato

SensorID	Timestamp (ISO format)	CO (entre 0 y 50.5 ppm)	O3 (entre 0 y 0.4 ppm)
----------	------------------------	-------------------------	------------------------

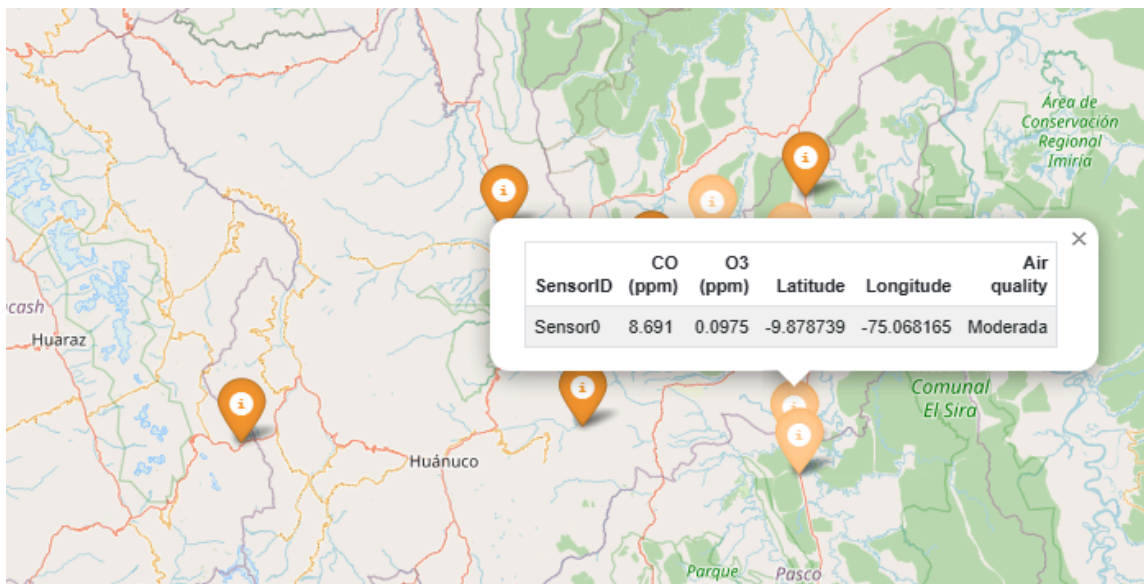
Para ello se crearon dos funciones, una encargada de generar los valores aleatorios dado un rango (`gen_random_value`) y otro que genera el archivo CSV (`gen_sensor_data`).

Postulante Ricardo Bravo Wong

Una vez se tiene el archivo CSV se procede a leer la información en un dataframe de pandas. La idea inicial es poder clasificar los valores de O3 y CO encontrados en función a la tabla mostrada anteriormente y para ello se agrupa los valores de cada sensor para luego calcular su mediana (Opté por este valor puesto que si los valores en sí son muy dispersos el promedio quedaría bastante alejado de los valores más comunes).

	SensorID	CO (ppm)	O3 (ppm)	Latitude	Longitude
0	Sensor0	34.6015	0.2370	-9.884788	-76.284972
1	Sensor1	14.5415	0.1675	-9.390164	-76.845535
2	Sensor10	29.5900	0.0610	-9.765414	-76.855378
3	Sensor2	33.4350	0.2040	-9.663599	-76.142199
4	Sensor3	13.7020	0.1570	-9.098146	-76.254271
5	Sensor4	21.4520	0.3220	-9.885419	-75.184618
6	Sensor5	36.1500	0.2580	-9.865092	-75.770412
7	Sensor6	33.9445	0.2980	-9.517724	-75.964428
8	Sensor7	20.5560	0.2305	-9.704672	-76.273561
9	Sensor8	17.8460	0.1710	-9.127129	-76.780155
10	Sensor9	30.6265	0.2340	-9.944748	-75.476525

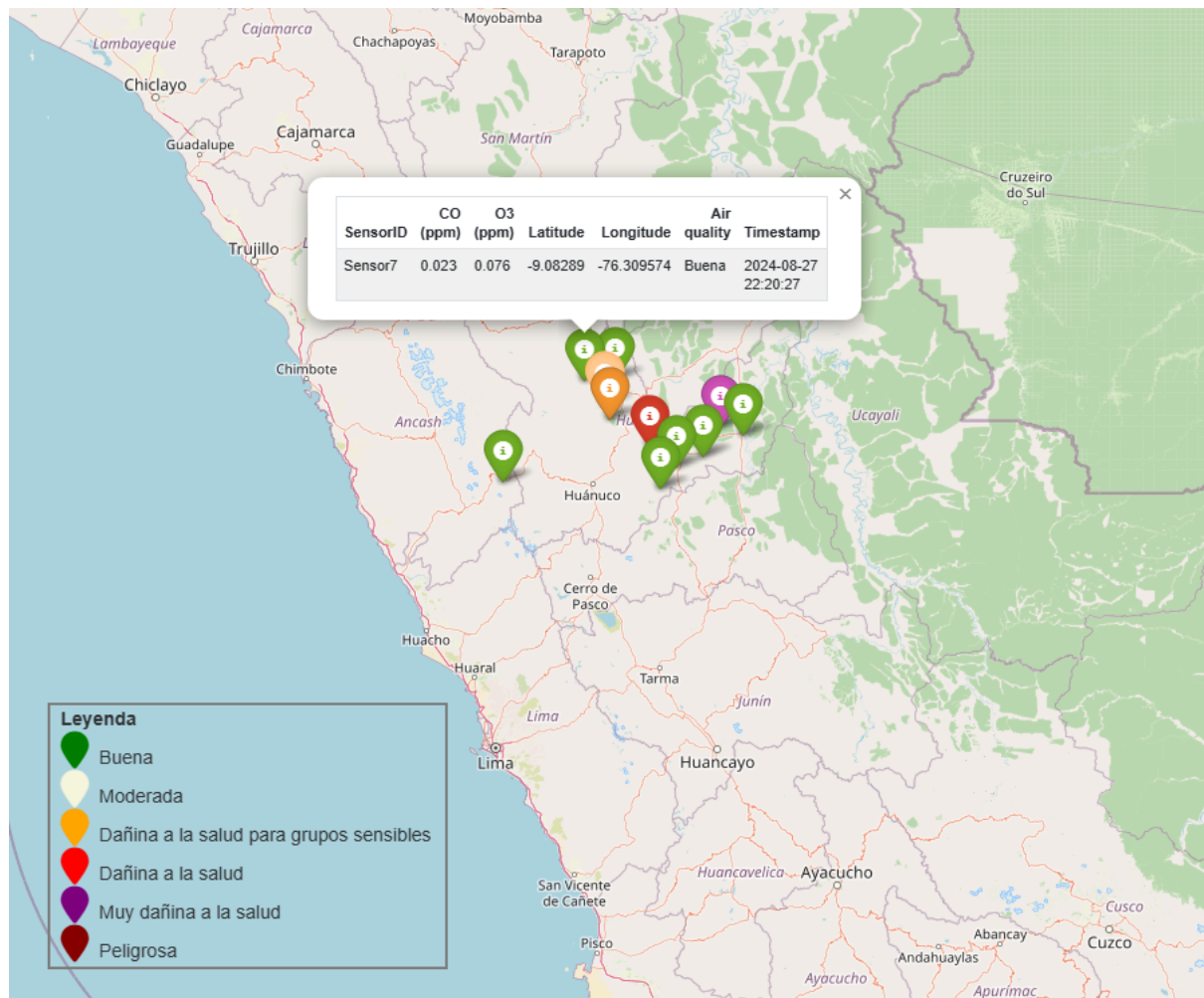
Una vez se tiene un dataframe con información procesada, se procede a clasificar y mostrar los valores en el mapa. Para ello, cada fila pasaba por una función llamada (`classify_air_quality`) en donde se retornaba información como el estado (Bueno, Malo, Dañino, etc) y el color con el que se graficaría el marcador en el mapa (Los colores están basados en la tabla de referencia). Asimismo, en Folium existe la posibilidad de agregar información a los marcadores, por lo que el marcador y mapa queda de la siguiente forma:



Finalmente, la idea era agregar una leyenda. Sin embargo, de acuerdo a lo leído en foros, necesita de plugins adicionales para hacerlo más sencillo y por temas de no instalar

Postulante Ricardo Bravo Wong

librerías adicionales, opté por hacer html con ayuda de copilotos como chat gpt. El resultado queda de la siguiente manera:



Cambio de la propuesta original:

La primera idea se basaba en generar un archivo estático con valores aleatorios de tamaño fijo e interpretar los datos. Sin embargo, una de las sugerencias del equipo del Semillero fue el hecho de que el archivo “estático”, que desde mi punto de vista simulaba una base de datos, sea variable y automáticamente se le agreguen valores en un intervalo de tiempo X.

Para realizar esto se buscó la misma lógica, pero ahora usando hilos donde uno de los hilos se encarga de mandar datos al archivo CSV (funciona como una tarea en segundo plano). Por otro lado, existe un loop en el script que genera un mapa cada 60 segundos y se encarga de actualizarlo.