Pila de aplicaciones de código abierto para el manejo de datos meteorológicos masivos en serie de tiempo generados por la estación meteorológica autónoma Vaisala WXT536

Armando Almeida-Gómez y Heykel Hernández-Hernández

Instituto de Geofísica y Astronomía, Cuba
Email: almeida.iga@gmail.com, heykelh33@gmail.com

Recibido: julio 2022	Aceptado: diciembre 2022

Resumen

El trabajo presenta una pila de aplicaciones de código abierto para el manejo de los datos meteorológicos masivos en serie de tiempo generados diariamente por la estación meteorológica Vaisala Weather Transmitter WXT536, instalada en el Instituto de Geofísica y Astronomía de Cuba como parte de la cooperación cubano-ruso para la corrección del retardo de la señal por factores del sistema de Navegación Global (GNSS). Se describe la arquitectura del sistema y los procesos de acondicionamiento, almacenamiento y visualización de los datos, mediante el empleo de un script de Bash, un sistema de gestión de base de datos relacional y geoespacial con PostgreSQL + PostGIS y la plataforma Grafana, como interfaz web para el monitoreo y análisis de los datos. La pila de aplicaciones pone a disposición de los científicos y tomadores de decisión, una herramienta de vigilancia de variables meteorológicas en series de tiempo, conformada por un conjunto de datos meteorológicos espacio-temporales para el desarrollo de los proyectos asociados a las ciencias ambientales.

Palabras clave: estación meteorológica autónoma Vaisala Weather Transmitter WXT536, Big Data, PostgreSQL, PostGIS, Grafana, Ciudades inteligentes.

Open-source application stack for handling massive meteorological time series data generated by Vaisala Autonomous Weather Station WXT536

Abstract

The work presents a stack of open-source applications for massive meteorological time series data management daily generated by the autonomous weather station Vaisala Weather Transmitter WXT536 (23°04’09.04372 N; -82°27’35.30501 W), installed at the Institute of Geophysics and Astronomy of Cuba as part of the Cuban-Russian cooperation for delay signal correction from a spatial object to Global navigation System (GNSS) receiver due to meteorological factors. The architecture of the system and the process of conditioning, storage and visualization of the data are described, through the use of a Bash script, a relational and geospatial database management system with PostgreSQL + PostGIS and the Grafana metrics observability platform, as a web interface for monitoring and data analysis. The stack of applications makes available to scientists and decision makers, a meteorological time series variables surveillance tool, conformed by space-time meteorological dataset used in the massive data analytics systems and geographic information client’s applications associated with environmental sciences.

Keywords: Vaisala Weather Transmitter WXT536 autonomous weather station, Big Data, PostgreSQL, PostGIS, Grafana, Smart cities.

1. Introducción

Las ciudades desempeñan un papel central en el proceso de crecimiento económico de una nación. Las evidencias sugieren que más del 90 % de la población mundial vivirá en una ciudad para finales de este siglo (Batty, M. 2018), (Batty et. al. 2019). Como lugar de crecimiento demográfico, comercial e industrial, las ciudades consumen energía, recursos naturales y económicos en su búsqueda por brindar mejores servicios para satisfacer las necesidades de su población, pero también generan residuos en niveles que a menudo, los sistemas naturales y artificiales no son capaces de depurar, aparecen problemas naturales, económicos y sociales que requieren soluciones.

Para enfrentar estos desafíos es indispensable disponer de un conjunto de informaciones que permitan una correcta gestión y funcionamiento de las ciudades. En este sentido, el desarrollo de las Tecnologías de la Información y las Comunicaciones (TICs), tiene mucho que aportar en la superación de estos obstáculos. La disponibilidad y uso de datos masivos (grandes conjuntos de datos) indexados por fecha y tiempo (series temporales), que representen fehacientemente el comportamiento dinámico y complejo de los fenómenos que se desarrollan en las ciudades, contribuyen a que los tomadores de decisión puedan realizar una evaluación más rigurosa y multicriterio de las opciones de disponibles y puedan adoptar mejores estrategias de solución de los problemas ambientales.

La digitalización se ha convertido en parte integral de la vida cotidiana y la recopilación, almacenamiento y análisis de datos casi en tiempo real, ha devenido en la acumulación de enormes cantidades de datos en lo que se denomina la era del "Big Data". Las ciencias ambientales están presenciando un rápido aumento en la cantidad de información en series de tiempo relevante (Vitolo Claudia et. al. 2015), a través del monitoreo ambiental, ya sea *in situ* o mediante sensores remotos y la adopción generalizada de computadoras, redes de comunicación y sensores (Kogan, F. 2010). Una gran parte de estos datos se ponen a disposición por instituciones gubernamentales y científicas, empresas estatales y privadas y ciudadanos (Buytaert, W. 2012), con la esperanza de lograr redes de datos donde la información de valor se coloque en Internet en formatos legibles por humanos y máquinas (Nielsen, M. 2011), que la información generada sea más transparente y de libre acceso para mejorar la participación pública en el proceso de toma de decisiones y formulación de políticas.

Muchos gobiernos están considerando la adopción del concepto de "ciudad inteligente" a través de la implementación de aplicaciones de datos masivos o “Big Data” con el propósito de superar muchos de los problemas existentes en las ciudades, alcanzar un nivel superior de sostenibilidad y mejorar la calidad de vida de su población (Pantelis K, Aija L. 2013). El término Big Data a pesar de ser un término que se emplea desde hace más de dos décadas, es en los últimos años cuando su uso se ha ido extendiendo de forma vertiginosa y en ámbitos cada vez más amplios, gracias a que ha ido ganando en importancia en todos los sectores. Originalmente se utilizaba principalmente para realizar estudios centrados en campos como la meteorología, para realizar estudios y predicciones, pero en la actualidad, los campos en los que tiene relevancia se han ido multiplicando y cada vez abarca una mayor cantidad de ellos. Big Data hace referencia a un conjunto de tecnologías e iniciativas sólidas que permiten a las organizaciones almacenar inmensos volúmenes de datos y garantizar un rendimiento increíblemente rápido de extracción de información útil y significativa mediante el análisis de datos. El Big Data se puede descomponer en varias fases, como la captura de los datos, su posterior almacenamiento, la visualización de los mismos y el análisis posterior para sacar conclusiones y poder tomar decisiones fundamentadas. Estas tecnologías pueden contribuir a mejorar los servicios de salud, transporte, energía, educación y agua, mejorar los niveles de calidad de vida de sus ciudadanos, reducir los costos, los consumos de recursos naturales y la generación de residuos, además de propiciar la participación más activas y efectiva de la población. Entonces las "ciudades inteligentes" podrían definirse como una ciudad que explota la tecnología digital con el fin de solucionar sus problemas, mejorar su funcionamiento y su gestión.

Para lograr un nivel tan alto de rendimiento de procesamiento de datos se requiere una pila de análisis de datos sólida, a partir de la implementación de un conjunto diverso de tecnologías. Una pila de aplicaciones es un conjunto de programas de aplicación que ayudan a realizar una determinada tarea. Estas aplicaciones están estrechamente vinculadas entre sí y los datos se pueden exportar o importar entre ellas con pasos mínimos.

Ejemplos de cómo las tecnologías de "Big Data" se están aplicando a campos relacionados con el riesgo costero, se hace evidente que las soluciones de Big Data geoespaciales tienen un claro potencial para mejorar el proceso de toma de decisiones basada en el riesgo en la costa (Rumson A. G. et. al.2017).

En las ciencias ambientales un ejemplo de Datos Masivos (Big Data) lo constituyen los datos aportados por sensores meteorológicos de todo tipo, que necesitan ser procesados y preparados para uso de la comunidad científica y de especialistas dedicados a las geociencias. En nuestro caso, estos datos corresponden a los generados por la estación meteorológica Vaisala WXT536 ubicada en los predios del Instituto de Geofísica y Astronomía, los cuales son registrados en formato nativo propio del equipo.

2. Materiales y métodos

Para el presente trabajo se establecieron dos líneas de acción paralelas, una encaminada a la programación de un script en Batch, que permitiera el procesamiento y transformación de los datos en bruto primarios obtenidos directamente de la estación meteorológica a la estructura planificada en la base de datos escogida Postgresql y otra línea de trabajo relacionada con la creación de un servidor de base de datos Postgresql y su conexión con la plataforma de visualización Grafana, ambas plataformas montadas sobre contenedores de ubuntu 20.04.

Como plataforma de virtualización fue utilizado Proxmox (Ahmed, W. 2017) en su versión 6.4, con un sistema de respaldo automático montado sobre Openmediavault, este sistema de respaldo permite la recuperación de los contenedores y máquinas virtuales en caso de destrucción de los mismos de forma rápida y así estarían disponibles en un corto período de tiempo.

2.1 Arquitectura general del sistema

El sistema lo componen los siguientes elementos (***Figura 1***):

- Estación meteorológica Vaisala WXT536
- Script para la preparación y almacenamiento de los datos
- Sistema de gestión de base de datos relacional y geoespacial con PostgreSQL + PostGIS
- Plataforma de observabilidad de métricas Grafana

La arquitectura del sistema se muestra en la siguiente figura.

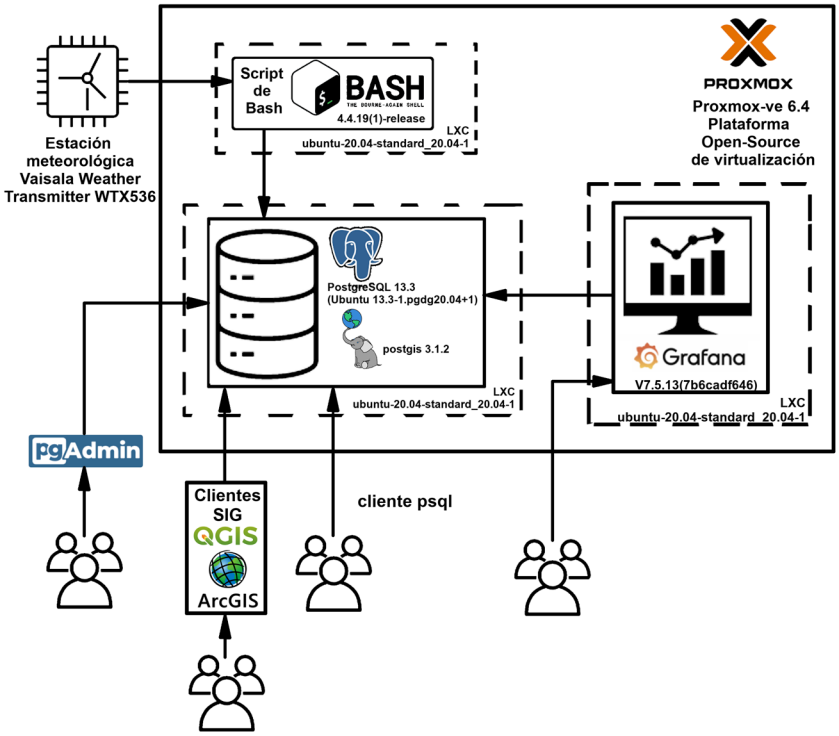


Fig.1. *Arquitectura general del sistema*

2.1.1 Estación meteorológica Vaisala WXT536

El Transmisor meteorológico Vaisala WXT536 (***Figura 2***) es una colección única de sensores que capturan una combinación de lecturas de viento, lluvia, temperatura, humedad y barométricas en un paquete compacto y asequible mediante el empleo de tecnologías de estado sólido para minimizar los costos de operación y mantenimiento. La integración es fácil y flexible a través de la interfaz digital y analógica, y también puede conectar sensores de terceros para crear su propio centro de parámetros meteorológicos.



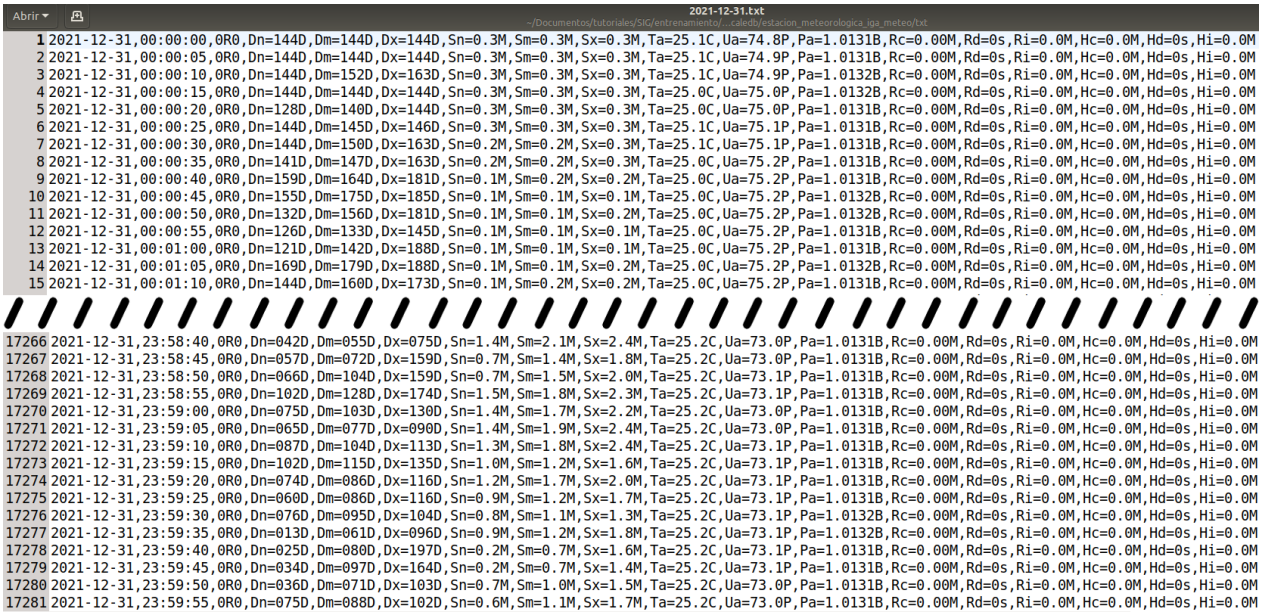
Fig. 2. *Estación meteorológica Vaisala WXT536*

Entre sus sensores cuenta con una tecnología exclusiva de sensor de estado sólido de Vaisala que incluye un sensor de viento WINDCAP® para determinar valores medios, máximos y mínimos para un tiempo de medición configurable entre 1-3600 s de dirección del viento entre 0 y 360^0, tiempo de respuesta de 0,25 s, precisión de ±3,0° a 10 m/s, resolución de 1^0 y velocidad del viento horizontal con un rango de 0-60 m/s, tiempo de respuesta de 0,25 s, precisión de ±3 % a 10 m/s y resolución de 0,1 m/s. El sensor capacitivo BAROCAP® para medir presión barométrica de 500-1100 hPa, precisión ±0,5 hPa de 0- 30 °C y ±1 hPa a +60 °C y una resolución de 0,1 hPa. El sensor HUMICAP® R2 se emplea para medir la humedad con un rango de medición de 0-100 %HR, una precisión de ±3 %HR de 0-90 %HR y ±5 %HR de 90-100 %HR y una resolución de 0,1 %HR. Se emplea un sensor resistivo de platino para medir la temperatura del aire con un rango de medición de −52- +60 °C, precisión de ±0,3 °C y resolución de 0,1 °C. Para medir la precipitación se emplea un sensor acústico RAINCAP® sin inundaciones, colmatación, humectación o pérdidas por evaporación. Este sensor cuenta con una área de captación de 60 cm², una resolución de 0,01 mm y un rango de medición de la intensidad de 0-200 mm/h. (fuente: ficha de datos vaisala estación meteorológica Vaisala WXT536 en sitio vaisala.com).

2.1.2 Preparación, almacenamiento y visualización de datos

Diariamente, la estación meteorológica genera un archivo de texto plano de datos estructurados cada cinco segundos para un total de 17281 registros/día, que no solo son difíciles de analizar por los especialistas, sino que no pueden ser procesados por las aplicaciones en las condiciones que se presentan (***Figura 3***).

Estos datos para ser analizados, necesitan ser acondicionados y convertidos en un formato adecuado, como los archivos CSV (del inglés *comma-separated values*). Este tipo de archivo es muy utilizado por los softwares de la ciencia de datos como Excel, o lenguajes de programación de código abierto como R o Python, junto a bibliotecas como Pandas, NumPy, modin y PandaPy, especializadas en el manejo y análisis de grandes conjuntos de datos. También se hace necesario importar los CSV a un sistema de gestión de base de datos relacional (RDBMS) como PostgreSQL (Schonig, Hans-Jurgen. 2020), unido a la extensión PostGIS, para convertir al RDBMS PostgreSQL en una base de datos espacial, herramienta muy empleada en los sistemas de información geográfica para análisis espaciales y por los visualizadores de datos en tiempo real como Grafana (Salituro, E. 2020), una herramienta de interfaz de usuario orientada al análisis y supervisión mediante obtención de datos a partir de consultas SQL.



Fecha: 31 de diciembre de 2021.

Fig. 3. *Fragmento de archivo de texto plano generado por la estación meteorológica autónoma Vaisala Weather Transmitter WXT536.*

Script para la preparación y almacenamiento de los datos

Para la preparación y almacenamiento de los datos se utiliza un script creado en Bash, como lenguaje de programación de comandos. Este *shell*, es el utilizado por el propio sistema operativo GNU/Linux y sus usuarios para automatizar tareas a través de instrucciones que se introducen directamente en un terminal, o usando archivos de texto con una serie de comandos (scripts) para ser ejecutados secuencialmente por el shell. Estos scripts son muy comunes dentro de GNU/Linux para la ejecución de programas que realizan tareas repetitivas, principalmente sobre archivos para el ahorro de tiempo, incluso pueden ser ejecutadas de manera programada mediante el Cron (encargados de la programación de tareas), proporciona una secuencia de actividades bien estructurada y modular, permite facilitar valores dinámicos a los comandos mediante el uso de argumentos de línea de comando y simplifica comandos complejos o múltiples comandos en una sola línea de comandos y gracias a ello es posible la ejecución de comandos como awk, sed, grep, tr head, tail, cat, split, junto a metacaracteres en una sola linea de código, lo que lo hace muy poderosos especialmente en el procesamiento de textos.

Los aspectos anteriores justifican el empleo del script para el acondicionamiento de los archivos TXT generados por la estación meteorológica, que cuentan un total de 17281 registros cada uno, e introducir las datas diarias en una base de datos de Postgresql “stations_metrics”, creada para el almacenamiento de las métricas ambientales, generadas por distintas fuentes de información. La ejecución del script es programada automáticamente con una frecuencia de dos veces al día a las 12 y 18 h de lunes a viernes (** 12,18 * * 1-5 user_name /path/script.sh 2>&1),* mediante el archivo de configuración Crontab y ejecutado por el demonio Cron. Esta configuración obedece a un proceso de copiado intermitente y manual de los archivos TXT hacia la carpeta donde se ejecuta el script, acumulándose los archivos sin procesar por políticas institucionales de seguridad informática.

Una vez obtenidas las credenciales para que el script se conecte al RDBS, se comprueba sí el servicio de PostgreSQL está en ejecución. Si la respuesta es afirmativa, se realiza una consulta SQL a la base de datos “stations_metrics”

(función *last_csv_inserted_query*), para obtener el último archivo CSV introducido y se devuelve su valor. Posteriormente se llama a la función *csv_file_db_inserting,* encargada de comprobar la existencia de nuevos archivos CSV y en caso positivo, registra en la base de datos (tabla "csv_file"), el nuevo archivo que será insertado para posteriormente insertar la información que contiene (tabla "metrics"). En el procesamiento de cada archivo TXT, se eliminan los caracteres {aA-zZ = [:blank:] # \r}, se eliminan y agregan columnas y se insertan nuevos datos.

El diagrama de flujo del script para la preparación de los datos TXT se muestra en la ***Figura 4***.

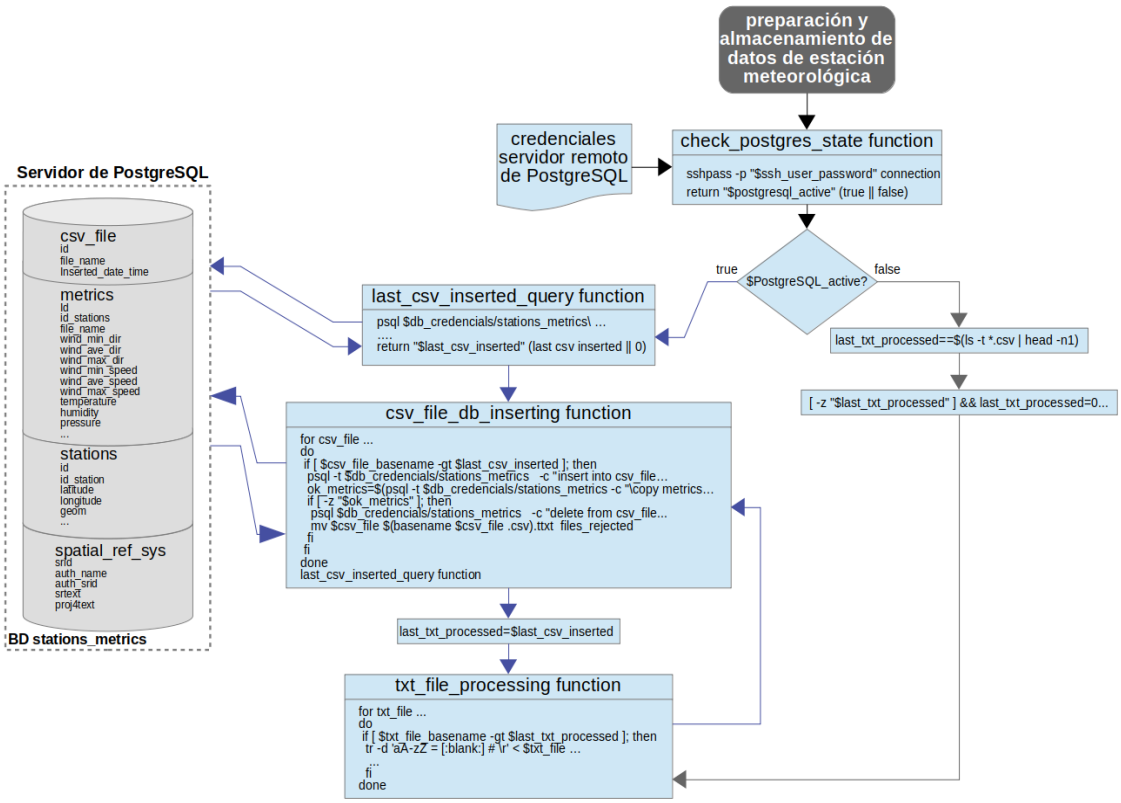


Fig. 4. *Diagrama de flujo del script para la preparación y almacenamiento de los datos*

En caso extraordinario de algún fallo de inserción de los datos, producido esencialmente por errores en la codificación del archivo CSV creado y su archivo de origen, el script mueve el CSV defectuoso y su archivo TXT correspondiente a otra carpeta, para ser revisado y corregido el TXT, convertido nuevamente en CSV e insertado manualmente. Consecuentemente, el script elimina de la base de datos el registro asociado al archivo que se deseaba insertar. Para evitar redundancia y/o repetición de datos, la base de datos está preparada para ser insertado uno y solo un archivo CSV según el nombre del archivo, que se corresponde con la fecha de medición (formato YYYY-MM-DD). Este ciclo se repite para todos los archivos CSV nuevos.

Terminada la inserción, se procede a comprobar la existencia de archivos TXT que no han sido convertidos a CSV mediante la función *txt_file_processing*, los convierte a CSV y seguidamente ejecuta nuevamente la función *csv_file_db_inserting* y se termina el script. Es de señalar que este proceso se realiza previendo la existencia diaria o no de nuevos archivos TXT y la posibilidad de caída del servidor de base de datos por alguna falla, lo cual de no preverse podría ocasionar un fallo en la ejecución del script.

Si la respuesta de la comprobación del estado de la ejecución del servicio de PostgreSQL es negativa, no se puede acceder a la base de datos "*stations_metrics*", por tanto, no se puede saber cuál es el último archivo CSV insertado. En tal caso, se toma como el último archivo insertado, el archivo CSV más recientemente creado (no tiene por qué coincidir con el último insertado en la base datos), se ejecuta la función txt*_file_processing* y se termina el script.

Un fragmento del archivo CSV obtenido por el script se muestra en la siguiente Figura (***Figura 5***).

Abrir — 2021-12-31.csv — ~/Documentos/tutoriales/SIG/entrenamiento/...caledb/estacion_meteorologica_iga_meteo/txt

```
 1 2021-12-31 00:00:00,meteo_IGA,144,144,144,0.3,0.3,0.3,25.1,74.8,1.0131,0.00,0,0.0,0.0,0,0.0,2021-12-31.txt
 2 2021-12-31 00:00:05,meteo_IGA,144,144,144,0.3,0.3,0.3,25.1,74.9,1.0131,0.00,0,0.0,0.0,0,0.0,2021-12-31.txt
 3 2021-12-31 00:00:10,meteo_IGA,144,152,163,0.3,0.3,0.3,25.1,74.9,1.0132,0.00,0,0.0,0.0,0,0.0,2021-12-31.txt
 4 2021-12-31 00:00:15,meteo_IGA,144,144,144,0.3,0.3,0.3,25.0,75.0,1.0132,0.00,0,0.0,0.0,0,0.0,2021-12-31.txt
 5 2021-12-31 00:00:20,meteo_IGA,128,140,144,0.3,0.3,0.3,25.0,75.0,1.0131,0.00,0,0.0,0.0,0,0.0,2021-12-31.txt
 6 2021-12-31 00:00:25,meteo_IGA,144,145,146,0.3,0.3,0.3,25.1,75.1,1.0131,0.00,0,0.0,0.0,0,0.0,2021-12-31.txt
 7 2021-12-31 00:00:30,meteo_IGA,144,150,163,0.2,0.2,0.3,25.1,75.1,1.0131,0.00,0,0.0,0.0,0,0.0,2021-12-31.txt
 8 2021-12-31 00:00:35,meteo_IGA,141,147,163,0.2,0.2,0.3,25.0,75.2,1.0131,0.00,0,0.0,0.0,0,0.0,2021-12-31.txt
 9 2021-12-31 00:00:40,meteo_IGA,159,164,181,0.1,0.2,0.2,25.0,75.2,1.0131,0.00,0,0.0,0.0,0,0.0,2021-12-31.txt
10 2021-12-31 00:00:45,meteo_IGA,155,175,185,0.1,0.1,0.1,25.0,75.2,1.0132,0.00,0,0.0,0.0,0,0.0,2021-12-31.txt
11 2021-12-31 00:00:50,meteo_IGA,132,156,181,0.1,0.1,0.2,25.0,75.2,1.0132,0.00,0,0.0,0.0,0,0.0,2021-12-31.txt
12 2021-12-31 00:00:55,meteo_IGA,126,133,145,0.1,0.1,0.1,25.0,75.2,1.0131,0.00,0,0.0,0.0,0,0.0,2021-12-31.txt
13 2021-12-31 00:01:00,meteo_IGA,121,142,188,0.1,0.1,0.1,25.0,75.2,1.0131,0.00,0,0.0,0.0,0,0.0,2021-12-31.txt
14 2021-12-31 00:01:05,meteo_IGA,169,179,188,0.1,0.1,0.2,25.0,75.2,1.0132,0.00,0,0.0,0.0,0,0.0,2021-12-31.txt
15 2021-12-31 00:01:10,meteo_IGA,144,160,173,0.1,0.2,0.2,25.0,75.2,1.0131,0.00,0,0.0,0.0,0,0.0,2021-12-31.txt
```



```
17266 2021-12-31 23:58:40,meteo_IGA,042,055,075,1.4,2.1,2.4,25.2,73.0,1.0131,0.00,0,0.0,0.0,0,0.0,2021-12-31.txt
17267 2021-12-31 23:58:45,meteo_IGA,057,072,159,0.7,1.4,1.8,25.2,73.0,1.0131,0.00,0,0.0,0.0,0,0.0,2021-12-31.txt
17268 2021-12-31 23:58:50,meteo_IGA,066,104,159,0.7,1.5,2.0,25.2,73.1,1.0131,0.00,0,0.0,0.0,0,0.0,2021-12-31.txt
17269 2021-12-31 23:58:55,meteo_IGA,102,128,174,1.5,1.8,2.3,25.2,73.1,1.0131,0.00,0,0.0,0.0,0,0.0,2021-12-31.txt
17270 2021-12-31 23:59:00,meteo_IGA,075,103,130,1.4,1.7,2.2,25.2,73.0,1.0131,0.00,0,0.0,0.0,0,0.0,2021-12-31.txt
17271 2021-12-31 23:59:05,meteo_IGA,065,077,090,1.4,1.9,2.4,25.2,73.0,1.0131,0.00,0,0.0,0.0,0,0.0,2021-12-31.txt
17272 2021-12-31 23:59:10,meteo_IGA,087,104,113,1.3,1.8,2.4,25.2,73.1,1.0131,0.00,0,0.0,0.0,0,0.0,2021-12-31.txt
17273 2021-12-31 23:59:15,meteo_IGA,102,115,135,1.0,1.2,1.6,25.2,73.1,1.0131,0.00,0,0.0,0.0,0,0.0,2021-12-31.txt
17274 2021-12-31 23:59:20,meteo_IGA,074,086,116,1.2,1.7,2.0,25.2,73.1,1.0131,0.00,0,0.0,0.0,0,0.0,2021-12-31.txt
17275 2021-12-31 23:59:25,meteo_IGA,060,086,116,0.9,1.2,1.7,25.2,73.1,1.0131,0.00,0,0.0,0.0,0,0.0,2021-12-31.txt
17276 2021-12-31 23:59:30,meteo_IGA,076,095,104,0.8,1.1,1.3,25.2,73.1,1.0132,0.00,0,0.0,0.0,0,0.0,2021-12-31.txt
17277 2021-12-31 23:59:35,meteo_IGA,013,061,096,0.9,1.2,1.8,25.2,73.1,1.0132,0.00,0,0.0,0.0,0,0.0,2021-12-31.txt
17278 2021-12-31 23:59:40,meteo_IGA,025,080,197,0.2,0.7,1.6,25.2,73.1,1.0131,0.00,0,0.0,0.0,0,0.0,2021-12-31.txt
17279 2021-12-31 23:59:45,meteo_IGA,034,097,164,0.2,0.7,1.4,25.2,73.1,1.0131,0.00,0,0.0,0.0,0,0.0,2021-12-31.txt
17280 2021-12-31 23:59:50,meteo_IGA,036,071,103,0.7,1.0,1.5,25.2,73.0,1.0131,0.00,0,0.0,0.0,0,0.0,2021-12-31.txt
17281 2021-12-31 23:59:55,meteo_IGA,075,088,102,0.6,1.1,1.7,25.2,73.0,1.0131,0.00,0,0.0,0.0,0,0.0,2021-12-31.txt
```

Fecha: 31 de diciembre de 2021.

Fig. 5. *Fragmento de archivo de CSV generado por el script de Bash*

2.1.3 Sistema de gestión de base de datos relacional y geoespacial con PostgreSQL + PostGIS + Timescaledb.

Como base de datos para este proyecto fue escogida la base de datos PostgreSQL en su versión 13.3, este tipo de base de datos aporta una gran flexibilidad dado que soporta múltiples lenguajes para programar funciones como PLpgSQL, PLPython, PLPHP o PLJava, entre otros.

Destacan también su seguridad, consistencia e integridad, así como el amplio espectro de tipos de datos soportados, entre los que destacan:

- Datos primitivos como enteros, numéricos, cadenas de texto, booleanos.
- Datos Estructurados como datos de fecha y hora, arrays o identificadores únicos.
- Documentos como JSON, XML, Key-value.

- Datos Geométricos: puntos, líneas, círculos y polígonos.
- Formatos personalizados

A esta base de datos se le incorporaron los Plugins PostGIS en su versión 3,1 y Timescaledb 1,7, los cuales agregan las siguientes funcionalidades:

PostGIS - La extensión PostGIS habilita el soporte para trabajar con objetos geográficos localizados en el espacio. En otras palabras: convierte PostgreSQL en una base de datos espacial, que en la práctica funciona (quitando el apartado gráfico) exactamente como un auténtico Sistema de Información Geográfica de escritorio.

Eso es posible porque, además de permitir el almacenamiento de información geográfica, el módulo en cuestión añade un enorme paquete de funciones para consultar, procesar, transformar y analizar información espacial almacenada.

Timescaledb - Es una base de datos de código abierto optimizada que se utiliza para almacenar datos de series temporales. Se implementa como una extensión de PostgreSQL y en ella se combinan la facilidad de uso de las bases de datos relacionales y la velocidad de las bases de datos NoSQL O sea, es una base de datos open-source para almacenamiento y análisis de series de tiempo con la potencia y ventajas de usar SQL.

Este contenedor fue creado con las siguientes características físicas: 2 procesadores, 2 Gb de RAM, 2 Gb de memoria de intercambio (SWAP) y 30 Gb de disco duro. En la ***Figura 6*** se muestran los consumos reales de este contenedor en producción

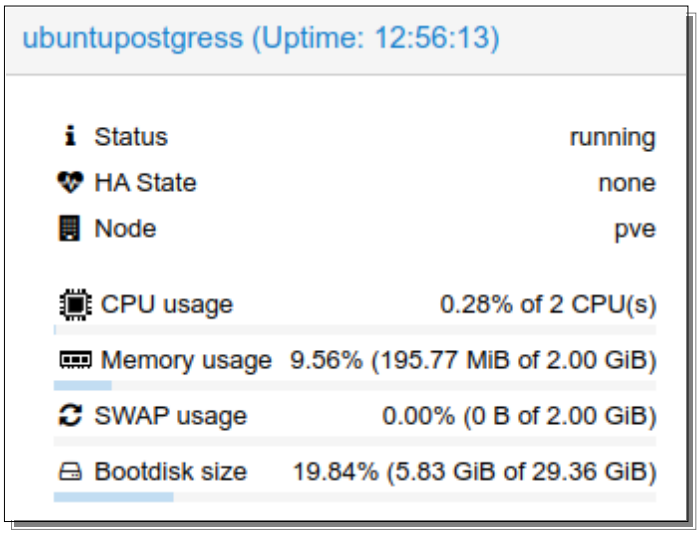


Fig. 6. *Estadísticas de funcionamiento del contenedor de PostgreSQL en producción*

2.1.4 Plataforma de observabilidad de métricas Grafana

Grafana es una solución de código abierto que sirve para ejecutar análisis de datos, extraer métricas que dan sentido ante enormes cantidades de datos y monitorear aplicaciones y recursos hardware con la ayuda de atractivos paneles de control personalizables. Entre sus características podemos destacar las siguientes:

- Se conecta con todas las fuentes de datos posibles, comúnmente conocidas como bases de datos como Graphite, Prometheus, Influx DB, ElasticSearch, MySQL, PostgreSQL, y otros.

- Es una solución de código abierto. Nos permite escribir complementos desde cero para su integración con varias fuentes de datos diferentes.
- La herramienta nos ayuda a estudiar, analizar y monitorear datos durante un período de tiempo, técnicamente llamado análisis de series de tiempo.
- En Grafana existen los paneles, donde contienen una gama de opciones de visualización, como mapas geográficos, mapas de calor, histogramas, toda la variedad de cuadros y gráficos que una empresa normalmente requiere para estudiar datos.
- Este modelo de código abierto se encarga de todos los análisis de nuestra aplicación. Podemos consultar, visualizar, configurar alertas y comprender los datos fácilmente con la ayuda de métricas.
- Es posible configurar alertas, que se envían por cualquier medio de comunicación cuando éstas cumplen las condiciones o parámetros programados.

En nuestro caso hemos decidido utilizar la versión 7.5.13 ya que precisamos del plugin Windrose para construir gráficos de rosas de viento que no trabaja con estabilidad en versiones superiores de Grafana. Hemos comunicado finalmente el servidor de Grafana con nuestra base de datos de PostgreSQL para que sirva de interface visual y filtrado a los datos de la estación meteorológica.

Al contenedor de Grafana le hemos asignado las siguientes características de hardware: 2 procesadores, 1 Gb de RAM, 1 Gb de área de intercambio y 40 Gb de disco duro. Las estadísticas de funcionamiento se muestran en la ***Figura 7***.

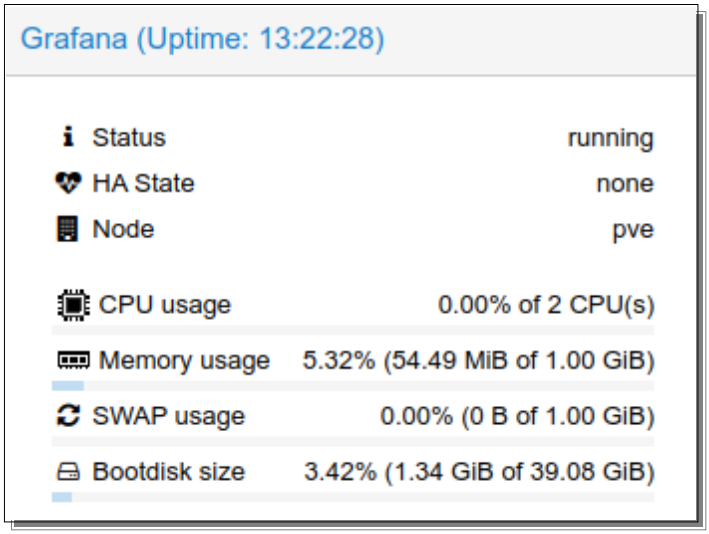


Fig. 7. ***Estadísticas de funcionamiento del contenedor de Grafana en producción.***

3. Resultados y discusión

Una vez enlazada la plataforma de análisis y métricas Grafana con la base de datos de PostgreSQL que contiene los datos de las variables medidas por la estación meteorológica se procedió a personalizar la pantalla de bienvenida de la aplicación (***Figura 8***).



Fig. 8. *Pantalla de bienvenida del portal WEB de la aplicación*

Posteriormente se elaboró un tablero, compuesto por las ventanas que muestran la distribución de las variables medidas (***Figuras 9, 10 y 11***).

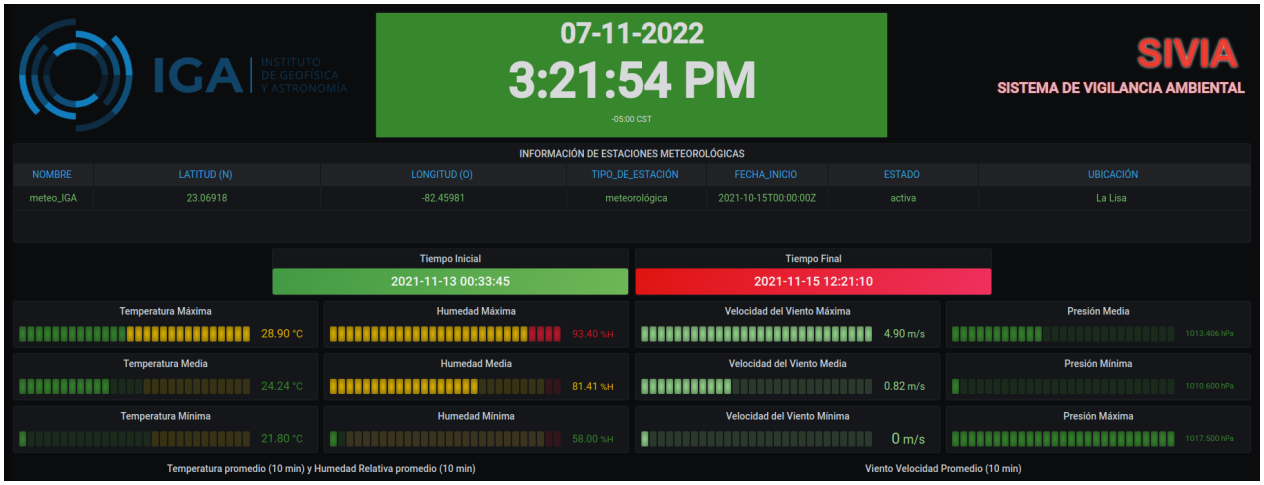


Fig. 9. *Parte superior del tablero de valores meteorológicos*

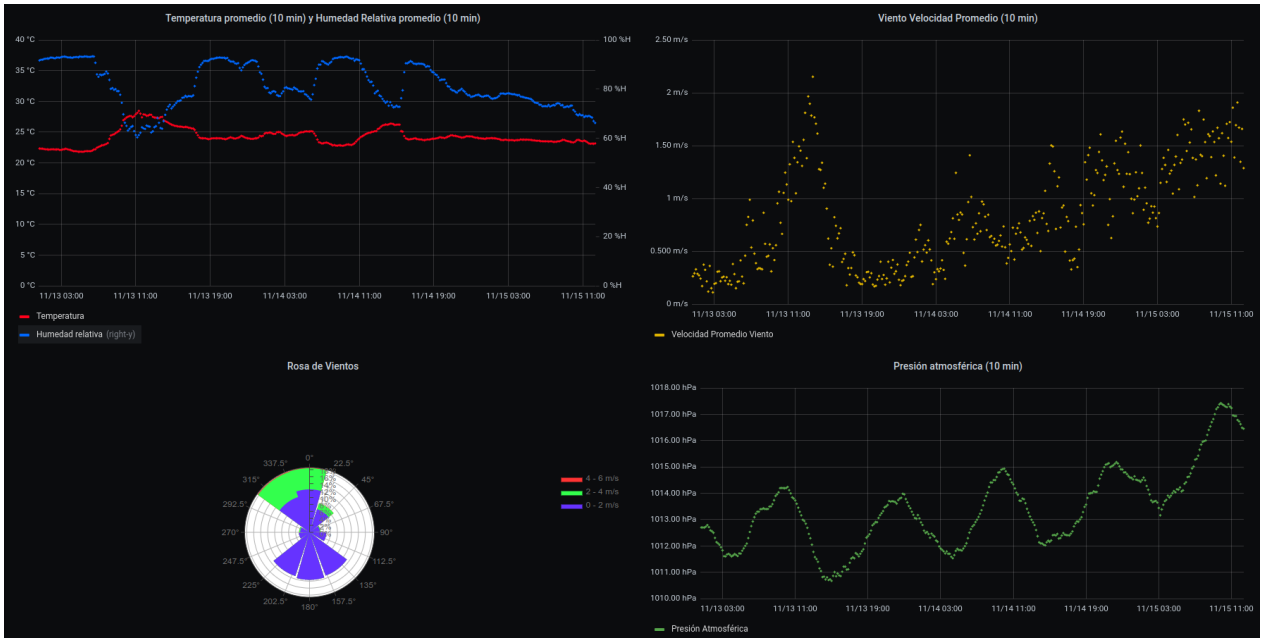


Fig.10. *Ventanas de datos en el tablero correspondiente a la información graficada de las variables meteorológicas incluyendo la rosa de vientos para el período seleccionado*



Fig.11. *Ventana donde se aprecia en la parte inferior los valores de eventos lluviosos en el período seleccionado*

Este tablero da la posibilidad de escoger el intervalo de tiempo deseado de la muestra, así como exportar los valores filtrados (en la ventana que muestra la tabla de estos valores) a archivos procesables en hojas de cálculo tales como

excel (***Figura 12***) a fin de realizar los procesamientos estadísticos que se necesiten acotados en intervalos de tiempo determinados. El tablero permite también configurar distintas vistas en dependencia de los usuarios registrados, lo que nos da una gran flexibilidad a la hora de representar los datos.

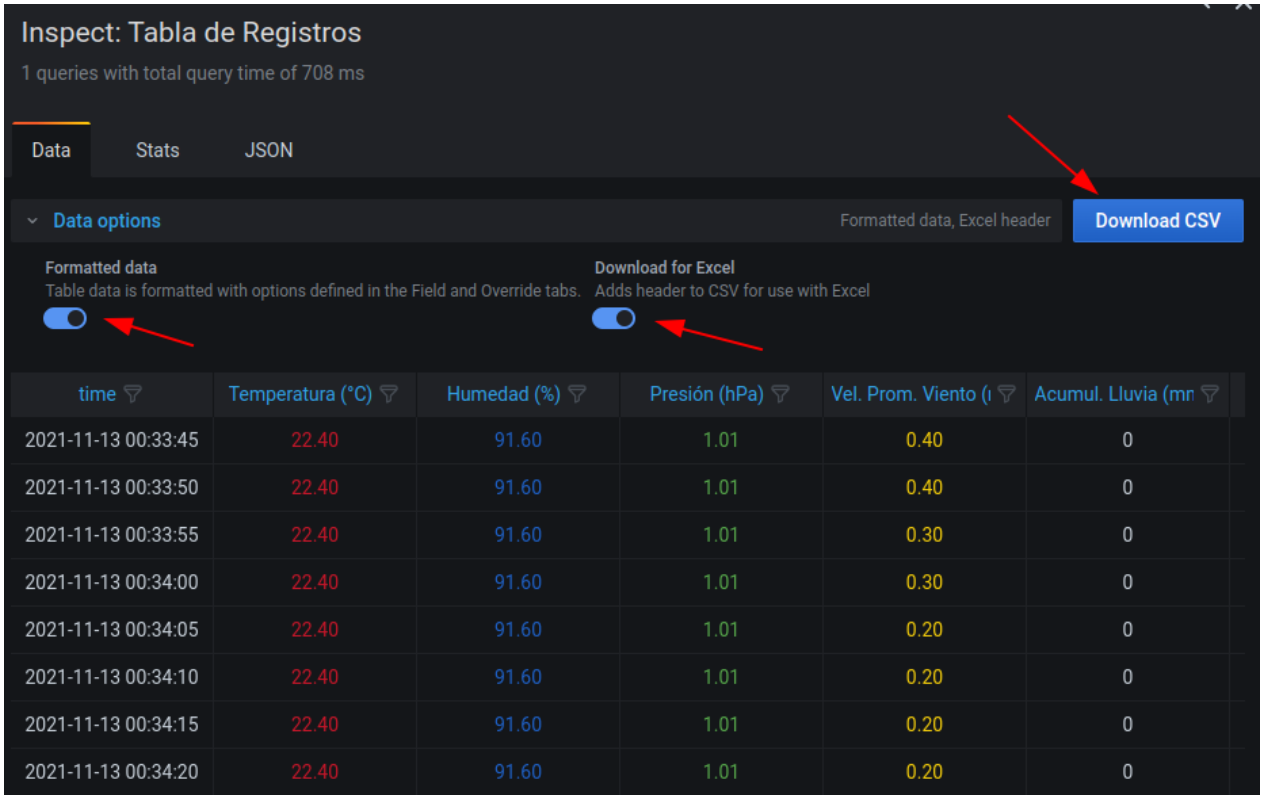


Fig.12. *Opciones de exportación de los datos seleccionados a excel en formato CSV para su ulterior procesamiento estadístico*

Conclusiones

- La metodología desarrollada en el presente trabajo es aplicable para cualquier data en bruto suministrada por cualquier equipamiento de mediciones de variables ambientales (López, H. R. 2019) tanto en tierra como marítimas (ejemplo: salinidad, turbidez, corrientes marinas, oleaje, elevación del nivel del mar y parámetros químicos del agua.
- Esta forma de tratamiento de la información ambiental brinda una gran facilidad a la hora de seleccionar y filtrar grandes cantidades de datos de una forma rápida y precisa e incorporar tantas estaciones de medición como sea posible en una misma base de datos, lo cual permite modelar espacialmente el comportamiento de las variables ambientales.
- Las bases (plataforma de virtualización) sobre las que se desarrolla esta metodología, presentan un impacto pequeño sobre la infraestructura informática, donde el consumo de recursos es mínimo tal y como fue demostrado, además de que permite su replicación en minutos hacia otros centros de datos en caso de ser necesario.

Recomendaciones.

- Para el trabajo con las variables ambientales se hace necesario tener la capacidad de poder procesar aquellos parámetros que dependen de la dirección en que influyen y su intensidad (Fernández-Gutiérrez, M. A. et. al. 2021), esto fue logrado en el caso del viento con el plugin de Grafana Windrose, el cual es un producto que ya no se sigue desarrollando. Este plugin al pasar Grafana de la versión 7 a la 8 dejo de funcionar por lo que se recomienda desarrollar un plugin similar que este soportado en las nuevas versiones de Grafana.

Referencias bibliográficas

Batty, M., 2018a. Inventing future cities. Cambridge, MA: The MIT Press, 282, Wenwen Li, Michael.

Batty & Michael F. Goodchild (2019): Real-time GIS for smart cities, International Journal of Geographical Information Science, DOI:10.1080/13658816.2019.1673397.

Claudia Vitolo, Yehia Elkhatib, Dominik Reusser, Christopher J.A. Macleod, Wouter Buytaert, 2015. Web technologies for environmental Big Data, Environmental Modelling & Software, Volume 63, 2015, Pages 185-198, ISSN 1364-8152, https://doi.org/10.1016/j.envsoft.2014.10.007.

Kogan, F., Powell, A., Fedorov, O., 2010. Use of Satellite and In-Situ Data to Improve Sustainability. Springer.

Buytaert, W., Baez, S., Bustamante, M., Dewulf, A., 2012. Web-based environmental simulation: bridging the gap between scientific modeling and decision-making. Environ. Sci. Technol. 46 (4), 1971e1976.

Nielsen, M., 2011. Reinventing Discovery: the New Era of Networked Science. Princeton University Press.

Pantelis K, Aija L. 2013. Understanding the value of (big) data. In Big Data, 2013 IEEE International Conference on IEEE; 2013. pp. 38–42.

Alexander G. Rumson, Stephen H. Hallett, Timothy R. Brewer. 2017. Coastal risk adaptation: the potential role of accessible geospatial Big Data, Marine Policy, Volume 83, Pages 100-110, https://doi.org/10.1016/j.marpol.2017.05.032.

Ahmed, Wasim. 2017.Mastering Proxmox - Third Edition: Build virtualized environments using the Proxmox VE hypervisor 3rd Revised edition Edición. Packt Publishing, 2017. ISBN 978-1788397605

Schonig, Hans-Jurgen. 2020. Mastering PostgreSQL 13: Build, administer, and maintain database applications efficiently with PostgreSQL 13, 4th Edition. Packt Publishing, 2020. ISBN 978-1800567498

Salituro, Eric. 2020. Learn Grafana 7.0: A beginner's guide to getting well versed in analytics, interactive dashboards, and monitoring. Packt Publishing, 2020. ISBN 978-1838826581

López, Héctor Ramón. 2019. "Técnica de extracción, transformación y carga de datos de estaciones meteorológicas". Revista de Ciencia y Tecnología, n.º 32 (4 de noviembre de 2019): 28–32. http://dx.doi.org/10.36995/j.recyt.2019.32.005.

Fernández-Gutiérrez, M. A., González Ramírez, C. M., & González Jardines, P. Cambios en los espectros de algunas variables meteorológicas y su relación con os índices NAO y AO. Revista Cubana De Meteorología, Vol. 27 Núm. 2 (2021): abril-junio. http://rcm.insmet.cu/index.php/rcm/article/view/562

Acerca de los autores.

Armando Almeida Gómez: Graduado de Ingeniería en Hidrografía y Oceanografía en el año 1986, en la Academia Naval Kirov del mar Caspio y Master en Medioambiente y Ordenamiento Territorial para el Turismo por la Universidad de la Habana-Cuba en 1997. Profesionalmente ha estado involucrado en proyectos de levantamientos batimétricos, inventario de territorios con uso de los SIG, estudios de impacto ambiental y proyectos de ingeniería de costas tanto en Cuba como en República Dominicana entre los años 1986 y 2012, actualmente se desempeña como especialista principal en el área de la informática en el Instituto de Geofísica y Astronomía de Cuba. No. ORCID: 0000-0003-2875-6689

Heykel Hernández-Hernández: Graduado de Ingeniería Química en el año 2002 en la Universidad Tecnológica de La Habana "José Antonio Echeverría". Master en Análisis y Tecnología Ambiental por la Instituto Superior de Tecnologías y Ciencias Aplicadas, La Habana, 2007. Líneas de Investigación: contaminación ambiental, estudios de impacto ambiental, tratamiento de aguas residuales, monitoreo ambiental, redes inalámbricas de sensores y Big Data. Servicios científico técnicos en Cuba y México en temas de salud ocupacional. Actualmente se desempeña como especialista ambiental en geología ambiental en el Instituto de Geofísica y Astronomía de Cuba. No. ORCID: 0000-0001-8141-5918