

Software libre para la recolección y publicación de datos de estaciones meteorológica automáticas

Free software for collecting and publishing data from automatic weather stations

José Luís Mengarelli, Exequiel Aramburu, Mario Martín Sbarbaro y
Cristhian Federico Bonnet*



Fecha de recepción: 29/08/2024

Fecha de aceptación: 15/11/2024

Resumen

Se exponen los resultados alcanzados en el desarrollo de un sistema distribuido, íntegramente realizado con software libre, que permita recolectar datos de diferentes estaciones meteorológicas automáticas, almacenarlos tales datos de manera centralizada y publicarlos a través de un sitio web. Un sistema distribuido es un conjunto de dispositivos y programas separados físicamente e interconectados por una red de comunicación. Los programas clientes son los nodos finales de este sistema y cada uno se vincula a una estación meteorológica automática, obtienen datos de sus sensores y los envía a un servidor central a través de esta red de comunicación. El servidor central, otro software diferente del sistema, recibe datos de los programas clientes y los almacena en una base de datos. Por último, un sitio web se vincula con el servidor central, obtiene los datos y los exhibe haciendo uso de gráficos, tablas y otros recursos visuales. Todo el sistema se interconecta a través de una red segura, habiéndose vinculando hasta el momento tres estaciones meteorológicas (dos en la ciudad de Paraná y una en Oro Verde) que envían datos al servidor central. Si bien estos datos se están almacenando, se deja para una etapa siguiente la validación de los mismos a fin de obtener mayor calidad e información.

Palabras Clave: *datos meteorológicos; programación; sistemas distribuidos; software libre; Linux.*

Abstract

The results achieved in the development of a distributed system are presented, entirely made with free software, which allows data to be collected from different automatic weather stations, stored in a centralized manner and published through a website. A distributed system

* Laboratorio de Investigación Gugler, FCyT (UADER), Oro Verde, Provincia de Entre Ríos, Argentina. Dirección de contacto: gugler_contacto@uader.edu.ar

is a set of devices and programs physically separated and interconnected by a communication network. The client programs are the end nodes of this system and each one links to an automatic weather station, obtains data from its sensors and sends it to a central server through this communication network. The central server, separate software from the system, receives data from client programs and stores it in a database. Finally, a website links to the central server, obtains the data and displays it using graphs, tables and other visual resources. The entire system is interconnected through a secure network, with three weather stations having been linked so far (two in the city of Paraná and one in Oro Verde) that send data to the central server. Although this data is being stored, its validation is left for a next stage in order to obtain greater quality and information.

Keywords: *meteorological data; programming; distributed systems; free software; Linux.*

Introducción

La información meteorológica es una herramienta muy importante para la toma de decisiones en diversas disciplinas. La reducción de riesgo de desastres, la protección de la salud humana, la gestión de recursos hídricos, el monitoreo de glaciares, la producción de alimentos, las actividades turísticas y las de transporte, se hacen más eficientes gracias a la constante información del estado del tiempo y de su predicción. Toda esta información aplicada, por ejemplo en la ingeniería robótica, permite iniciar automáticamente sistemas de riego ante determinados valores de humedad en el suelo, regular sistemas de refrigeración en cámaras frigoríficas y automatizar la alimentación de animales de manera autónoma, entre otras cosas. Las estaciones meteorológicas automáticas son los instrumentos que utilizamos para recolectar los datos necesarios para obtener ésta información.

Descripción del proyecto

Una EMA (Estación Meteorológica Automática) es un dispositivo destinado a medir y registrar regularmente las diferentes variables meteorológicas, que utiliza instrumentos adecuados para tal fin. Cada estación cuenta con distintos instrumentos de medición o sensores; comúnmente una estación posee sensores de temperatura, humedad, velocidad y dirección del viento, presión atmosférica, precipitación y radiación solar. Para la realización de este proyecto se utilizó una estación meteorológica marca DAZA modelo DZWH 3104, la cual cuenta con dos componentes principales: una unidad externa (Figura 1) y una estación de recepción (Figura 2).

La unidad externa está formada por un mástil principal al cual se le anexan distintos sensores encargados de realizar lecturas correspondientes a los valores de las variables me-

teorológicas. Los sensores (anemómetro, veleta, pluviómetro, placa solar y termómetro) se encuentran conectados a una controladora alimentada por dos pilas AA la cual sincroniza la lectura, genera un registro con los datos y los envía a la estación de recepción para su procesamiento. Estos pasos son realizados cada cierto período de tiempo (48 segundos por defecto) utilizando una señal de radio frecuencia a 433 Mhz como canal de comunicación.



Figura 1. Unidad externa de la estación meteorológica



Figura 2. Estación de recepción

La estación de recepción recibe los datos de la unidad externa, los almacena en una memoria (hasta un máximo de 4080 registros) y visualiza en pantalla. Además posee sus propios sensores con los que realiza lecturas de las condiciones meteorológicas del ambiente en el que se encuentra, por lo que gestiona los datos de la unidad externa además de la propia. Mediante un puerto USB (Bus Universal en Serie) se conecta la estación de recepción a una computadora (Figura 3) para que ésta envíe datos meteorológicos o reciba información de configuración. Si bien la estación trae un software de gestión que permite configurar ciertos valores y leer los datos que registra, es en este punto donde comienza el desarrollo objeto del presente artículo.

Las estaciones que se analizan incluyen programas de computadora o software que permiten vincularlas a la computadora, obtener sus datos, visualizarlos y en algunos casos publicarlos en algún sitio de meteorología existente en Internet. Estos programas son desarrollados bajo el concepto de software privativo o propietario, lo que significa, fundamentalmente, no tener control sobre lo que se hace, debido a que no puede estudiarse el código fuente ni modificarse (Stallman, 2004), lo cual representa una limitante si se quiere obtener alguna funcionalidad extra a la que ya poseen.

Mediante este proyecto se desarrolló un conjunto de programas denominado Proyecto EMA Libre que interconecta estaciones meteorológicas automáticas a través de una

red de comunicación, obtiene y almacena los datos en un servidor central para posteriormente brindarlos a través de un visualizador. Estos programas son licenciados bajo el concepto de software libre, lo que permite a cualquier interesado descargar, estudiar, modificar y utilizar libremente cualquiera de ellos pudiendo publicarse las mejoras o cambios que se hayan realizado (Stallman, 2004). En este escenario las estaciones meteorológicas automáticas se instalan en diferentes puntos geográficos por lo que debemos asociar a cada una un programa que recolecte datos de sus sensores, los formatee y los envíe a un servidor. Esta orquestación es la que define a los sistemas distribuidos.



Figura 3. Unidad de recepción conectada a una computadora

En la región existen organismos que ya poseen sus propias redes de estaciones meteorológicas recolectando y transmitiendo sus datos. En efecto, el Servicio Meteorológico Nacional, la Bolsa de Cereales de Entre Ríos, la Dirección de Hidráulica de Entre Ríos, la cátedra de Climatología Agrícola de la Universidad Nacional de Entre Ríos (UNER.) y el Instituto Nacional de Tecnología Agropecuaria (INTA) poseen distintos programas mediante los cuales obtienen estos datos, los almacenan y luego los procesan a fin de obtener información. Todos tienen un denominador en común: que estos programas no están licenciados bajo el concepto de *software libre*. Esto exime la posibilidad de que cualquier interesado los pueda utilizar, mejorarlos o modificarlos a fin de adaptarlos a sus necesidades. Este trabajo es el primero en su tipo llevado a cabo en la región dado que cada una de las herramientas de software desarrolladas ha sido licenciada con software libre y publicadas en internet con el propósito de que cualquier persona, organismo o universidad interesada pueda hacer uso de ella y de su código fuente.

Un sistema distribuido es una colección de componentes de software autónomos localizados en distintas computadoras que dan al usuario la impresión de constituir un único sistema coherente (Tanenbaum y Van Steen, 2008). Por otro lado, dentro de los sistemas distribuidos existen distintos modelos arquitectónicos que tratan sobre la colocación de cada uno de sus componentes y las relaciones entre ellos. Algunos ejemplos son el modelo cliente servidor y el modelo de procesos de igual a igual (Coulouris et al., 2011).

Sobre la base de un escenario ideal definido, se determinó que el modelo cliente-servidor sería el adecuado para cumplir los requerimientos descriptos, y se decidió la creación de tres tipos de programas diferentes: (a) los clientes denominados EMA Libre Cliente, (b) el servidor llamado EMA Libre Center y (c) un programa visualizador llamado EMA Libre Visor, interconectados a través de una red de comunicación (Figura 4). Cada cliente se vincula a una estación meteorológica automática diferente (existirá uno por cada estación existente en la red), obtiene sus datos y genera un registro meteorológico (estructura estandarizada que define la ubicación y el formato de los mismos). Luego este registro es enviado al servidor quien se encarga de validarlo y almacenarlo en una base de datos. Por último, EMA Libre Visor se conectará con el servidor, consultará los datos almacenados, los procesará y mostrará haciendo uso de distintos recursos gráficos.

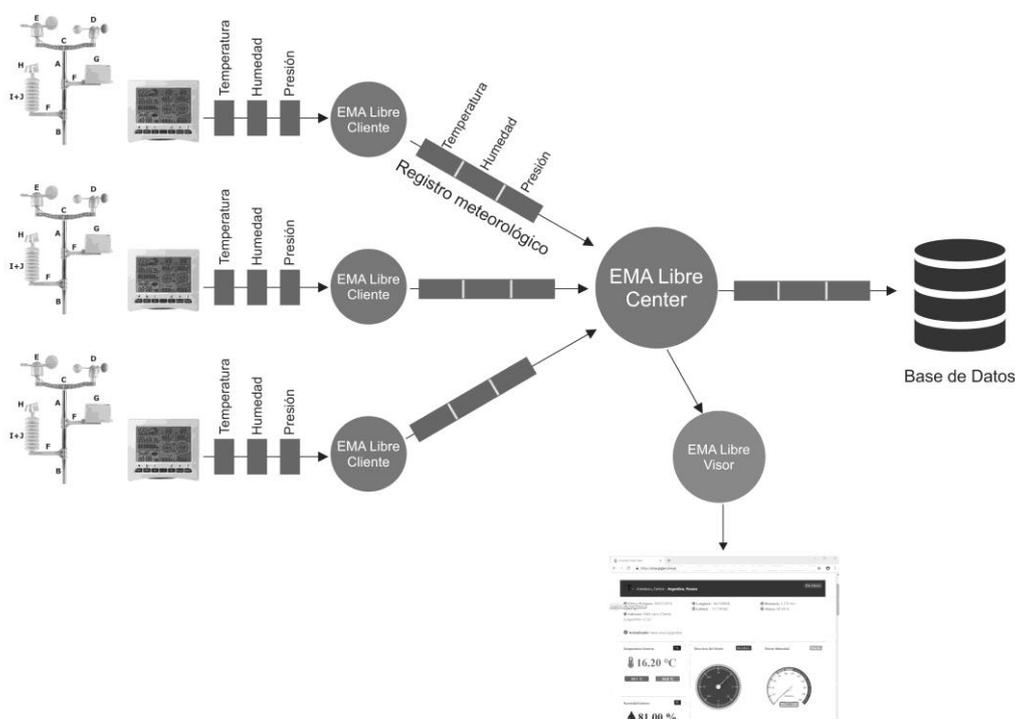


Figura 4. Esquema del Sistema Distribuido EMA Libre

El Cliente: EMA Libre Cliente (Carpincho)

EMA Libre Cliente denominado Carpincho se desarrolló utilizando *Node.js* con el objetivo que sea multiplataforma y que pueda ejecutarse en dos modos diferentes: uno de consola y otro con interfaz gráfica. Carpincho en modo consola (Figura 5) contiene una interfaz minimalista que permite la ejecución en una terminal, sin la necesidad de contar con un entorno gráfico, ideal para servidores o computadoras de placa única. Un ejemplo de estas últimas es el caso de las *raspberry pi* que son desarrolladas y promovidas por la Fundación Raspberry Pi, ya que consumen menos recursos. Carpincho, en modo gráfico (Figura 6) incluye una interfaz más amigable y visual utilizando el proyecto *NW.js*, siendo éste un *framework* que combina *Node.js* con el motor del navegador Web de *Chromium* para crear aplicaciones gráficas de escritorio con lenguajes HTML, CSS y JavaScript.

Figura 5. EMA Libre Carpincho Consola

```
EMA Libre                                     Logs                                     Salir
Modos: Web/Consola                           Modo Actual: web
Version: 1.25v                                 Recolecta: 30 s
Estado: Encendida                             Ult. Reg1: 23/11/2019 20:28:01

DATOS EXTERNOS                                DATOS INTERNOS
Temperatura: 24.7°C                           Temperatura: 29.7°C
Humedad: 40%                                  Humedad: 29%
Presion R.: 1013.2 hpa
Punto de rocío: 10.17°C
Iluminancia: 0 lux
UV: 0 uv
Viento V.: 2.52 Km/h
Viento R.: 5.04 Km/h
Viento N.: Noroeste

TANQUES DEL SISTEMA
Tanque Diario: 4.72 Mb
Tanque Pendiente: 0.00 Mb
Tanque Debug: 4.99 Mb

LLUVIA ACUMULADA
Actual: 0 mm   Hora: 0 mm   Diaria: 0 mm   Total: 810.3 mm

REDES ENVIADAS
emacentergugler:23/11/2019 20:27:31 - true
pws:23/11/2019 20:27:31 - true
windguru:23/11/2019 20:27:31 - true
wunderground:23/11/2019 20:27:31 - true
```

Figura 6. EMA Libre Carpincho GUI



Se definió una misma estructura de directorios en ambos modos de funcionamiento para alojar los archivos según su naturaleza y función dentro del sistema, quedando conformada de la siguiente manera:

- **/src:** contiene todo el código fuente del desarrollo.
- **/conf:** contiene dos archivos con configuraciones que el sistema lee para su funcionamiento:
 - **ConfiguracionGeneral.js:** se establecen parámetros de configuración general, configuración de correo electrónico para notificaciones, parámetros para la publicación de información a distintas redes meteorológicas, entre otras configuraciones posibles.
 - **ConfiguracionGlobal.js:** parámetros de la versión del programa (se utili-

- za esta información para las actualizaciones de versión).
- **/core:** contiene los archivos que brindan funcionalidad al software:
 - **ema.js:** se ejecuta en segundo plano y es el encargado, de obtener, registrar, calcular y publicar los registros, incluyendo los módulos necesarios para realizar estas tareas.
 - **mail.js:** contiene un conjunto de funciones de envío de correo electrónico para las notificaciones.
 - **recolector.js:** brinda la funcionalidad de recolección de datos de una estación USB.
 - **redes_meteorologicas.js:** brinda la funcionalidad de envío de datos a una red meteorológica. Contiene un método que recibe como parámetros una red, y los datos a enviar, y la configuración requerida por la misma (la cual es incluida desde los archivos alojados en **/redes**).
 - **validaciones.js:** contiene métodos que validan si cada registro recolectado (ej.: temperatura, humedad, etc) está dentro de los rangos estipulados como correctos, indicando error en el caso que la validación no sea superada.
 - **actualizar.js:** contiene las funciones de verificación de versión y actualización del sistema.
 - **/gui:** contiene los archivos correspondientes a la interfaz del sistema. Este directorio solamente se incluye en el modo gráfico.
 - **/logs:** directorio destinado a almacenar los distintos registros generados por el sistema:
 - **debug.txt:** guarda información sobre la carga de módulos, estado de conexión de la estación, información del sistema operativo, versión del cliente, configuración de zona horaria y lectura de los parámetros de la estación.
 - **registro_diario.txt:** información de los registros diarios leídos por la estación.
 - **tanquePendientes.log:** este archivo se utiliza cuando el cliente pierde conexión con EMA Libre Center y no puede publicar datos. Los registros se almacenan aquí y al momento de restablecerse la comunicación son leídos y enviados uno a uno.

Una vez instalado, el programa admite dos modos de operación diferentes: visor y recolector:

- **Visor:** no requiere una EMA vinculada al puerto USB. Se conecta a través de una red de comunicación con el servidor EMA Libre Center y obtiene datos de la estación meteorológica automática configurada. Este modo solo visualiza los datos almacenados en el servidor y en tiempo real.
- **Recolector:** requiere una EMA vinculada al puerto USB. Se conecta a la misma, lee sus datos, los formatea y los publica al servidor de EMA Libre Center como así también a otras redes meteorológicas abiertas existentes en Internet.

El proceso principal es **ema.js**, el cual se encarga gestionar el funcionamiento total del cliente. La configuración del modo de trabajo se almacena en el archivo `ConfiguracionGeneral.js`. Al momento de cargarse en memoria este proceso, lee la configuración del archivo y determina su modo de funcionamiento.

Carpincho en modo visor, fue desarrollado para usuarios que no poseen una EMA. En este modo consume, procesa y muestra la información en tiempo real de una estación que se encuentra publicando a un servidor EMA *Libre Center*. Para obtener los registros de una estación particular es necesario utilizar una API (*Application Programming Interface*) que fue desarrollada y publicada en EMA *Libre Center*, la cual permite comunicar e intercambiar los datos con cualquier cliente Carpincho que, siendo un cliente, consume esta API en forma continua según el tiempo de recolección configurado, por defecto 40 segundos, obteniendo los registros de una estación meteorológica automática. Es importante señalar que para poder consumir esta API es necesario autenticarse en ella. Esto se realiza a través de una llave de lectura, única y privada para cada estación. En el caso de las estaciones del laboratorio, éstas son públicas, y por ello viene una de ellas previamente configurada en el cliente Carpincho, que se corresponde con la llave de lectura de la EMA instalada en la ciudad Oro Verde, Entre Ríos, Argentina.

En modo recolector, se vincula a una EMA conectada al puerto USB, en forma continua según el tiempo de recolección configurado, por defecto 40 segundos, y realiza las siguientes acciones:

1. **Leer datos desde el puerto USB.** Las estaciones utilizadas en este proyecto poseen una memoria que almacena dos bloques de datos claramente diferenciables: uno fijo y uno dinámico. Los primeros 256 bytes del mapa de memoria corresponden al blo-

que fijo y contiene datos como registros de temperatura máxima, mínima y configuraciones de alarma. El diseño detallado de registro se presenta en la Tabla 1 (véase el Anexo I). El bloque dinámico guarda un historial de registros con los datos de las variables meteorológicas leídas por la estación. El tamaño y estructura de los registros es de 20 bytes, los cuales se detallan en la Tabla 2 del Anexo I. Toda esta información se obtiene desde el puerto USB en fragmentos de 32 bytes hexadecimales (Figura 7), para lo cual se adaptó un driver de licencia libre: **whid.js**, perteneciente al proyecto (<https://github.com/haricotbean/WeatherStation>), que obtiene el bloque hexadecimal, identifica los datos, los categoriza y estandariza en un formato que hace más sencilla su interpretación. Este driver fue desarrollado para el modelo de estación **ws1080** lo que nos obligó a adaptar el mismo para que pueda reconocer e interpretar el bloque recibido desde nuestro modelo de estación (el cual incluía la información de sensores de luminiscencia y de UV que el original no contemplaba).

| Offset (h) | 00 | 01 | 02 | 03 | 04 | 05 | 06 | 07 | 08 | 09 | 0A | 0B | 0C | 0D | 0E | 0F |
|------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| 0000007DB0 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| 0000007DC0 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| 0000007DD0 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| 0000007DE0 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| 0000007DF0 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| 0000007E00 | EB | 52 | 90 | 4E | 54 | 46 | 53 | 20 | 20 | 20 | 20 | 00 | 02 | 08 | 00 | 00 |
| 0000007E10 | 00 | 00 | 00 | 00 | 00 | F8 | 00 | 00 | 3F | 00 | FF | 00 | 3F | 00 | 00 | 00 |
| 0000007E20 | 00 | 00 | 00 | 00 | 80 | 00 | 80 | 00 | C5 | EA | FF | 0F | 00 | 00 | 00 | 00 |
| 0000007E30 | 00 | 00 | 0C | 00 | 00 | 00 | 00 | 00 | AC | FE | FF | 00 | 00 | 00 | 00 | 00 |
| 0000007E40 | F6 | 00 | 00 | 00 | 01 | 00 | 00 | 00 | A1 | 89 | 4C | 88 | 9D | 4C | 88 | 8A |

Figura 7. Parte del fragmento de 32 bytes recibido desde la estación

2. **Generar un arreglo con los datos.** Adaptado, el *driver* define un arreglo cuya estructura se detalla en la Tabla 3 (Anexo I) con los valores obtenidos desde los sensores.
3. **Validar cada parámetro del arreglo.** Definido el arreglo, se procede a realizar validaciones sobre cada uno de los datos. Si alguna de estas validaciones no se cumple, el registro es descartado. Las validaciones se exponen en la Tabla 4 del Anexo I.
4. **Publicar los datos en las redes meteorológicas configuradas.** Existen en internet redes meteorológicas que permiten publicar sus datos a usuarios particulares. EMA Libre Cliente lo hace en las siguientes redes meteorológicas: *EMA Libre Center*, *pws*, *windguru*, *wunderground* y *wow*. Todas estas redes difieren en la forma en que reciben los datos, por lo que se programó un controlador distinto para cada una de ellas, los cuales se encuentran dentro del directorio **/src/core/redes**. Además los paráme-

tros de autenticación y envío también difieren, por lo que se guardan dentro del archivo de configuración **ConfiguracionGeneral.js** al momento de dar de alta la estación. Periódicamente el proceso principal (ema.js) ejecuta el método **EviarRedesMeteorologicas** del archivo **redes_meteorologicas.js** enviando como parámetros la red a la cual se quiere publicar y los datos meteorológicos. Según la red, el método elegirá el controlador necesario, los parámetros de autenticación y envío y publicará los datos. El resultado de éxito o fracaso será devuelto al proceso principal para su posterior procesamiento.

Requerimientos de autenticación y envío de cada red meteorológica

1. EMA Libre Center

Para este tipo de red meteorológica, el software provee dos opciones “EMA Libre Center Gugler” y “EMA Libre Center Particular”. La diferencia entre ambas redes es que para la primera no hay que configurar el host, ya que apunta directamente al servidor EMA Libre Center del Laboratorio Investigación Gugler (<https://emacenter.gugler.com.ar>). La segunda posibilita a que un usuario cree su propia red meteorológica, utilizando el proyecto EMA Libre Center, por lo que esta opción permite parametrizar el equipo (host), donde se le indica la URL o dirección IP del servidor EMA Libre Center Particular. En ambos casos se utiliza una llave de escritura para la autenticación de los clientes, con permiso de escritura de nuevos registros. A continuación se detallan los parámetros del controlador: **Host:** emacenter.gugler.com.ar o URL/IP (en caso de Particular); **Puerto:** 443; **Path:** /post; **Autenticación:** llave de escritura; **Método:** POST.

| | | |
|----------------------------|-------------------------------|---|
| 'fecha', | Fecha actual | -> En ISO |
| 'temperatura_externa', | Temperatura externa | -> En celsius |
| 'temperatura_interna', | Temperatura interna | -> En celsius |
| 'humedad_externa', | Humedad externa | -> En % |
| 'humedad_interna', | Humedad interna | -> En % |
| 'punto_de_rocio', | Punto de rocío | -> En celsius |
| 'luxer_intencidad', | Radiación Solar | -> En en w/m ² (watt por metro cuadrado) |
| 'luxer_uv', | Uv | -> En en w/m ² (watt por metro cuadrado) |
| 'viento_velocidad', | Velocidad | -> En kilómetros por hora |
| 'viento_rafagas', | Velocidad de ráfagas | -> En kilómetros por hora |
| 'viento_direccion', | Dirección del viento | -> En grados desde el norte |
| 'presion_relativa', | Presión atmosférica relativa | -> En hectopascal |
| 'presion_absoluta', | Presión atmosférica absoluta | -> En hectopascal |
| 'lloviendo', | Estado de lluvia | -> En milímetros |
| 'lluvia_actual', | Lluvia actual | -> En milímetros |
| 'lluvia_acumulado_hora', | Lluvia acumulado por hora | -> En milímetros |
| 'lluvia_acumulado_diario', | Lluvia acumulado por día | -> En milímetros |
| 'software', | Nombre y versión del software | -> En texto |

Figura 8. Estructura de registro meteorológico requerida por EMA Libre Center

2. Red Pws (<http://www.pwsweather.com>)

Esta red brinda una API la cual se autentica con usuario y contraseña, por lo que el modulo recibe estos valores junto con los datos a publicar. A continuación se detallan los parámetros del controlador: **Host:** www.pwsweather.com; **Puerto:** 80; **Path:** /pwsupdate/pwsupdate.php; **Autenticación:** usuario y contraseña; **Método:** get.

```
[
  'id',           | Nombre de usuario      -> En texto
  'password',    | Contraseña            -> En texto
  'ateutc',      | Fecha actual          -> Formato UTC
  'tempf',       | Temp. Externa        -> En fahrenheit
  'humidity',    | Humedad Externa      -> En %
  'dewptf',     | Punto de Rocío       -> En fahrenheit
  'uv',          | UV                   -> En en w/m² (watt por metro cuadrado)
  'solarradiation', | Solar radiacion     -> En en w/m² (watt por metro cuadrado)
  'windspeedmph', | Viento               -> En en metros por hora
  'winddir',     | Direccion del Viento -> En grados desde el norte
  'windgustmph', | Rafagas              -> En metros por hora
  'rainin',      | Lluvia Acumulado Hora -> En pulgadas
  'dailyrainin', | Lluvia Acumulado Dia -> En milímetros
  'baromin',     | Presion Relativa     -> En inches
  'softwaretype', | Software              -> En texto
  'action',      | Tipo de actualizacion -> En texto
]
```

Figura 9. Estructura de registro meteorológico requerida por Pws.

3. Red Windguru (<https://www.windguru.cz>)

Esta red brinda una API la cual se autentica con usuario y contraseña, por lo que el modulo recibe estos valores junto con los datos a publicar. A continuación se detallan los parámetros del controlador: **Host:** www.windguru.cz; **Puerto:** 80; **Path:** /upload/api.php; **Autenticación:** usuario y contraseña; **Control:** https://www.windguru.cz/int/wgsapi.php; **Método:** get.

```
[
  'interval',    | Intervalo             -> En enteros
  'precip_interval', | Intervalo de precipitación -> En enteros
  'wind_avg',    | Viento en nudos      -> En nudos
  'wind_max',    | Viento en nudos. Ráfagas -> En nudos
  'wind_direction', | Dirección del viento -> En grados desde el norte
  'temperature', | Temperatura          -> En celsius
  'rh',          | Humedad              -> En %
  'mslp',        | Presión relativa     -> En hectopascal
  'precip',      | Lluvia acumulado por hora -> En milímetros
]
```

Figura 10. Estructura de registro meteorológico requerida por Windguru

4. Red Wunderground (<https://www.wunderground.com>)

Esta red brinda una API la cual se autentica con usuario y contraseña, por lo que el modulo recibe estos valores junto con los datos a publicar. A continuación se detallan los parámetros del controlador: **Host:** rtupdate.wunderground.com; **Puerto:** 443; **Path:** /weatherstation/updateweatherstation.php; **Autenticación:** usuario y contraseña; **Método:** get.

| | | | |
|---|-------------------|--|---|
| [| | | |
| | 'dateutc', | | Fecha actual -> En UTC |
| | 'tempf', | | Temperatura externa -> En fahrenheit |
| | 'indoortempf', | | Temperatura interna -> En fahrenheit |
| | 'indoorhumidity', | | Humedad interna -> En % |
| | 'humidity', | | Humedad externa -> En % |
| | 'dewptf', | | Punto de Rocío -> En fahrenheit |
| | 'uv', | | UV -> En en w/m ² (watt por metro cuadrado) |
| | 'solarradiation', | | Solar radiacion -> En en w/m ² (watt por metro cuadrado) |
| | 'winddir', | | Direccion del Viento -> En grados desde el norte |
| | 'windgustmph', | | Rafagas -> En metros por hora |
| | 'windspeedmph', | | Viento -> En metros por hora |
| | 'rainin', | | Lluvia Acumulado Hora -> En pulgadas |
| | 'dailyrainin', | | Lluvia Acumulado Dia -> En pulgadas |
| | 'baromin', | | Presion Relativa -> En inches |
| | 'softwaretype', | | Software -> En texto |
|] | | | |

Figura 11. Estructura de registro meteorológico requerida por WunderGround

5. Red Wow (<http://wow.metoffice.gov.uk>)

Esta red brinda una API la cual se autentica con usuario y contraseña, por lo que el módulo recibe estos valores junto con los datos a publicar. A continuación se detallan los parámetros del controlador: **Host:** wow.metoffice.gov.uk; **Puerto:** 80; **Path:** /automaticreading; **Autenticación:** usuario y contraseña; **Método:** get.

| | | | |
|---|-----------------|--|--|
| [| | | |
| | 'dateutc', | | Fecha actual -> En UTC |
| | 'tempf', | | Temperatura externa -> En fahrenheit |
| | 'humidity', | | Humedad externa -> En % |
| | 'dewptf', | | Punto de Rocío -> En fahrenheit |
| | 'winddir', | | Direccion del Viento -> En grados desde el norte |
| | 'windgustmph', | | Rafagas -> En metros por hora |
| | 'windspeedmph', | | Viento -> En metros por hora |
| | 'rainin', | | Lluvia Acumulado Hora -> En milímetros |
| | 'baromin', | | Presion Relativa -> En inches |
| | 'softwaretype', | | Software -> En texto |
|] | | | |

Figura 12. Estructura de registro meteorológico requerida por Wow

6. El servidor: EMA Libre Center (Capibara)

El servidor EMA Libre Center denominado Capibara, es un software que permite registrar programas clientes asociados a estaciones meteorológicas con el fin de obtener, validar, catalogar y almacenar sus datos, siendo el núcleo de una **Red Abierta de EMA Libre** (Red Abierta de Estaciones Meteorológicas Automáticas Libre) ya que aquí es donde se almacenan los registros de todas las estaciones meteorológicas en una Base de datos Libre. Para este desarrollo se seleccionó y utilizó PostgreSQL como Sistema Gestor de Base de Datos. Capibara brinda la posibilidad de consultar toda esta información en un formato estándar y abierto para que cualquier persona pueda hacer uso de la misma. Se encuentra

publicado mediante la siguiente url: <https://emacenter.gugler.com.ar>. Al acceder se presentará el portal web que ofrece información sobre el desarrollo, un formulario web de registro de estaciones y un medio de contacto.

Para el desarrollo de Capibara se utilizó python como lenguaje de programación incorporando las siguientes librerías y/o frameworks: Django, djangorestframework, gunicorn, Pillow, psycopg2, setproctitle y django-cors-headers.

Una de las funciones principales desarrolladas en EMA Libre Center es proveer una API (application programming interface), que permita comunicar e intercambiar información con cualquier cliente, especialmente con EMA Libre Carpincho, esta API se desarrolló bajo la tecnología API REST (**RE**presentational **St**ate **T**ransfer) y brinda los métodos necesarios para: Dar de alta programas clientes asociados a estaciones meteorológicas para que envíen sus datos. Obtener, validar, catalogar y almacenar todos estos datos. Publicar la información almacenada.

Dar de alta estaciones meteorológicas para que envíen sus datos

Cualquier persona que posea una estación meteorológica (siempre que ésta implemente la API declarada por el servidor) podrá configurarla para que envíe sus datos Ema Libre Center. El interesado deberá completar un formulario web (Figura 13) donde se solicitan datos de contacto e información específica de su estación.

EMA Center Registrar estación Estaciones disponibles Contacto Ingresar

Registrar estación

Nombre de la estación:

Modelo de la estación:

País:

Ciudad:

Calle:

Latitud:

Longitud:

Altura sobre nivel del mar:

Nombre:

Apellido:

Dirección de email:

Nombre de usuario:

Obligatorio. Longitud máxima de 100 caracteres. Solo puede estar formado por letras, números y los caracteres @/./+/_

Password:

Enviar datos de registro

Figura 13. Formulario para el registro de estaciones meteorológicas

Finalizado el registro obtendrá un usuario y contraseña con el cual puede acceder a la gestión de sus datos.

Autenticación de clientes en el servidor

Se utilizó el concepto de llaves como método de autenticación entre los programas clientes y el servidor. Al registrarse, el nuevo usuario accede a un par de llaves las cuáles deberá agregar a su programa cliente para que éste se autentique en EMA Libre Center antes de requerir cualquier acción. Por cada usuario se crean dos llaves diferentes: una de lectura y una de escritura (Figura 14).

```
LLAVE DE LECTURA: e530tns3-2de3-4dgh-90rw-1c8367d83e1d  
LLAVE DE ESCRITURA: t830rab3-2awb-4r42-90ab-1c8dhn7dd5x4
```

Figura 14. Llaves de lectura y escritura

Estas llaves son almacenadas junto con la información del usuario en la base de datos. Las llaves de lectura se utilizan para autenticar a cualquier cliente que quiera obtener información del servidor, mientras que las llaves de escritura las deberán utilizar aquellos que quieran enviar sus datos para que EMA Libre Center los almacene. Siempre que un cliente quiera leer datos o prepare un registro a ser almacenado deberá enviar la llave que corresponda, el servidor la validará con las almacenadas en su base de datos y permitirá, o no, la acción solicitada. Además, la conexión entre el cliente y el programa servidor se realiza cifrada mediante mecanismos SSL/TLS, por lo que ningún dato viaja en texto plano aumentando la privacidad en el envío.

Envío de información de clientes al servidor

Para el envío de información desde el cliente se definió la estructura del objeto JSON que el servidor espera recibir. Se debe empaquetar toda la información requerida y enviarla mediante el método post a la url: <https://emacenter.gugler.com.ar/>, agregando la clave de escritura. La petición queda como se presenta en la Figura 15.

```
POST /HTTP/1.1  
Host: emacenter.gugler.com.ar  
Content-Type: application / json  
Content-Length: 375  
  
t830rab3-2awb-4r42-90ab-1c8dhn7dd5x4
```

Figura 15. Envío de datos desde el cliente al servidor - método POST

```
[
  'fecha',
  'temperatura_externa',
  'temperatura_interna',
  'humedad_externa',
  'humedad_interna',
  'punto_de_rocio',
  'luxer_intencidad',
  'luxer_uv',
  'viento_velocidad',
  'viento_rafagas',
  'viento_direccion',
  'presion_relativa',
  'presion_absoluta',
  'lloviendo',
  'lluvia_actual',
  'lluvia_acumulado_hora',
  'lluvia_acumulado_diario',
  'software',
  |
  Fecha actual
  Temperatura externa
  Temperatura interna
  Humedad externa
  Humedad interna
  Punto de rocío
  Radiación Solar
  Uv
  Velocidad
  Velocidad de ráfagas
  Dirección del viento
  Presión atmosférica relativa
  Presión atmosférica absoluta
  Estado de lluvia
  Lluvia actual
  Lluvia acumulado por hora
  Lluvia acumulado por día
  Nombre y versión del software
  -> En ISO
  -> En celsius
  -> En celsius
  -> En %
  -> En %
  -> En celsius
  -> En en w/m² (watt por metro cuadrado)
  -> En en w/m² (watt por metro cuadrado)
  -> En kilómetros por hora
  -> En kilómetros por hora
  -> En grados desde el norte
  -> En hectopascal
  -> En hectopascal
  -> En hectopascal
  -> En milímetros
  -> En texto
]
```

Figura 16. Estructura de registro meteorológico requerida por el servidor

Cuando el servidor recibe los datos desde un cliente, realiza unas validaciones (Tabla 5, véase Anexo I) que determinarán si son almacenados o no. Se creó la base de datos de nombre *EmaCenter* compuesta de las tablas descritas en la Figura 17.

| <input type="checkbox"/> | Name | Owner | Partitioned table? |
|--------------------------|----------------------------|-----------|--------------------------------------|
| <input type="checkbox"/> | auth_group | emacenter | <input type="button" value="False"/> |
| <input type="checkbox"/> | auth_group_permissions | emacenter | <input type="button" value="False"/> |
| <input type="checkbox"/> | auth_permission | emacenter | <input type="button" value="False"/> |
| <input type="checkbox"/> | auth_user | emacenter | <input type="button" value="False"/> |
| <input type="checkbox"/> | auth_user_groups | emacenter | <input type="button" value="False"/> |
| <input type="checkbox"/> | auth_user_user_permissions | emacenter | <input type="button" value="False"/> |
| <input type="checkbox"/> | climapp_estaciones | emacenter | <input type="button" value="False"/> |
| <input type="checkbox"/> | climapp_registros | emacenter | <input type="button" value="False"/> |
| <input type="checkbox"/> | django_admin_log | emacenter | <input type="button" value="False"/> |
| <input type="checkbox"/> | django_content_type | emacenter | <input type="button" value="False"/> |
| <input type="checkbox"/> | django_migrations | emacenter | <input type="button" value="False"/> |
| <input type="checkbox"/> | django_session | emacenter | <input type="button" value="False"/> |

Figura 17. Base de datos EmaCenter

La tabla **auth_user** (véase Tabla 6, Anexo I) almacena información de las cuentas de los usuarios registradas en el sistema. Se utiliza en el proceso de registro y de autenticación cuando un usuario quiere acceder al portal de administración de su estación. La información de las estaciones meteorológicas se guarda en la tabla **climapp_estaciones** (Tabla 7, Anexo I) y se relaciona con la tabla **auth_user** mediante el campo **id_usuario_id**

(dado que cada estación requiere la existencia de un usuario en la tabla `auth_user`). Los registros que cada estación envía se almacenan en la tabla `climapp_registros`. Éstos se relacionan con los datos de la estación que los envió (almacenada en la tabla `climapp_estaciones`) mediante el campo `id_estacion_id`. La descripción completa de la tabla `climapp_registros` se detalla en la Tabla 8 (Anexo I). Por su parte, EMA Libre Center brinda la posibilidad de que cualquier persona, previo registro, pueda consultar y hacer uso de todos los datos almacenados. Para esto se desarrolló una serie de APIS de consulta que brinda información de distinta índole. En la Tabla 9 (Anexo I) se detallan las rutas a consultar. Cada consulta devolverá un objeto JSON diferente con la información solicitada. En la Tabla 10 (Anexo I) se detalla la estructura del objeto según la petición realizada.

El visualizador: EMA Libre Visor (Ronsoco)

Por último se hizo imprescindible contar con una aplicación que visualice todos los datos meteorológicos de manera más amigable e intuitiva y que además permita descargar todos los desarrollos del proyecto EMA Libre detallando la funcionalidad de cada uno de ellos. Esto determinó el desarrollo **EMA Libre Visor** bajo el nombre de **Ronsoco**, un sitio web público con tecnología HTML, CSS y Javascript que brinda información del proyecto **EMA Libre**, permite la descarga de los programas, su código fuente y brinda una interfaz interactiva para visualizar datos meteorológicos en tiempo real por medio de herramientas gráficas y tablas. Estos datos podrán obtenerse de cualquier servidor EMA Libre Center (sea particular o público) configurando solamente la URL o IP del mismo.

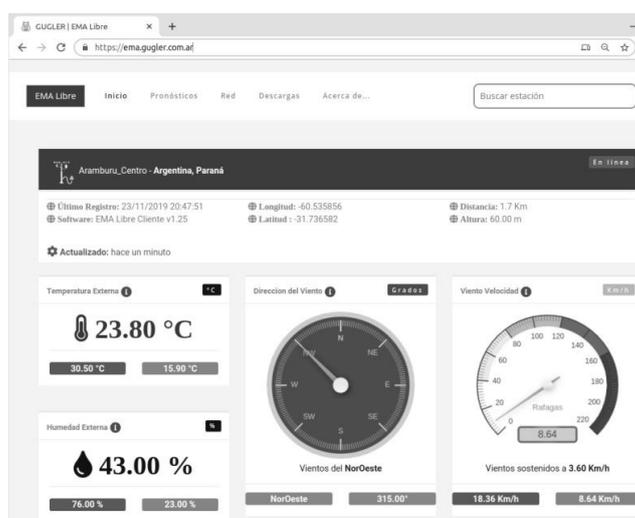


Figura 18. EMA Libre Visor

Resultados

Como resultado del presente trabajo se obtuvo un sistema distribuido libre, conformado por tres tipos diferentes de aplicaciones complementadas entre sí, cada una de ellas sea igualmente importante y necesaria. Todos los programas que conforman el proyecto EMA Libre están bajo la licencia de software GPL versión 3, permitiendo que cualquier persona u organismo pueda descargar y hacer uso de ellos sin limitaciones, además de poder modificar el modo de operar de sus componentes. El proyecto no se limita a la utilización de la red meteorológica tal y como se encuentra implementada en la actualidad sino que cualquier interesado puede generar su propia red meteorológica al descargar todos los programas y configurarlos, pudiendo de esta forma gestionar su propia información y decidiendo el hacerla pública o no.

Discusión

Durante el desarrollo del sistema se obtuvo un sistema informático robusto que, haciendo uso de una red, permitió la conexión y el envío de datos de las estaciones meteorológicas ubicadas en puntos equidistantes a un servidor central. Si bien una gran parte de las tareas se encuentra finalizada, este trabajo forma parte de un proyecto mayor que tomará estos resultados como base para su continuación. Hoy en día se posee una base de datos con más de dos millones de registros meteorológicos, aunque aún no se ha avanzado totalmente en mecanismos de control que examinen y ponderen la calidad de estos datos, a fin de establecer el grado de validez requerido para poder utilizarlos en aspectos como la predicción del estado del tiempo, entre otros. Por tal motivo, se continuará avanzando en este aspecto hasta lograr un grado más de eficiencia para el proyecto EMA Libre.

Conclusiones

Este trabajo es el primero en su tipo llevado a cabo en la región. Se presentan a continuación, las siguientes conclusiones:

- La información meteorológica presenta gran interés en diversos ámbitos universitarios, de colegios técnicos y profesionales del agro.
- El desarrollo basado en el software libre posibilita la reutilización de código, permitiendo utilizar un driver elaborado para una estación diferente a la adquirida, realizando las modificaciones necesarias para que se adapte a las nuevas exigencias.

- El desarrollo modular de EMA Libre Cliente permite, de una manera sencilla, incluir nuevos *driver* que posibiliten el envío a la red meteorológica que se requiera.
- El diseño distribuido del proyecto EMA Libre permite escalar una Red EMA Libre, además de extender las funcionalidades de cada software de manera independiente.
- Al utilizar un formato estandarizado de comunicación, como lo es el REST/JSON, cualquier persona podrá desarrollar los drivers necesarios para leer o escribir datos en EMA Libre Center.

Agradecimientos. A la FCyT – UADER por impulsar la conformación del equipo interdisciplinario de profesionales, alumnos y colaboradores que llevó a cabo este desarrollo, posibilitando la difusión de las bondades y cualidades del Software Libre. El financiamiento se obtuvo por el proyecto PIDIN titulado “Desarrollo de un software libre para la administración y publicación de datos de manera automática de distintas estaciones meteorológicas”. De allí se obtuvieron los fondos para la adquisición de una estación meteorológica, material bibliográfico e insumos correspondientes a bienes de consumo. Especial agradecimiento al Dr. Ing. César Aguirre y al Dr. Ernesto Brunetto, como así también a los colegas y amigos que han apoyado esta iniciativa y contribuido con información, referencias y lectura crítica del presente material.

Bibliografía citada

- ❖ Coulouris, G., Dollimore, J, y T. Kindberg, 2001. *Sistemas Distribuidos. Conceptos y Diseño*. Tercera Edición. Pearson Educación, S. A. Madrid, 726 pp.
- ❖ Stallman, R. M., 2004. *Software libre para una sociedad libre*. Traficantes de Sueños, Madrid, 318 pp.
- ❖ Tanenbaum, A. S. y M. Van Steen, 2008. *Sistemas Distribuidos. Principios y Paradigmas*. Segunda Edición. Pearson Educación, Estado de México, 686 pp.
- ❖ Tanenbaum, A. y D. Wetheral, 2012. *Redes de Computadoras*. Quinta Edición. Pearson Educación, Estado de México, 791 pp.

ANEXO I

Tabla 1. Bloque fijo de 256 bytes

| Dirección | Bytes | Item | Comentarios |
|-----------|-------|--------------------------------------|---|
| 0 | 2 | Bandera inicializada de EEPROM | Debe ser 0x55AA. |
| 2 | 2 | (2080 solamente) modelo | Se espera que sea 0x8010 |
| 2 | 2 | (3080 solamente) factor de lluvia | Divide por 8192 |
| 4 | 1 | (2080 solamente) versión | Se espera que sea 0x20 |
| 4 | 2 | (3080 solamente) factor del viento | |
| 5 | 2 | (2080 solamente) ID | |
| 6 | 5 | (3080 solamente) fecha de máximo UV | |
| 7 | 2 | (2080 solamente) factor de lluvia | Divide por 8192 |
| 9 | 2 | (2080 solamente) factor del viento | Divide por 8192 |
| 11 | 2 | (2080 solamente) factor de lluvia | Factor de lluvia ^ 0xffff |
| 11 | 5 | (3080 solamente) máximo, iluminancia | |
| 13 | 2 | (2080 solamente) factor de viento | Factor del viento ^ 0xffff |
| 16 | 1 | periodo de lectura | Minutos entre cada lectura almacenada |
| 17 | 1 | ajustes de la unidad 1 | bit 0: temperatura interior: 0 = °C, 1 = °F bit 1: temperatura exterior: 0 = °C, 1 = °F bit 2: lluvia: 0 = mm, 1 = pulgada bit 5: presión: 1 = hPa bit 6: presión: 1 = inHg bit 7: presión: 1 = mmHg |
| 18 | 1 | ajustes de la unidad 2 | bit 0: velocidad del viento: 1 = m / s bit 1: velocidad del viento: 1 = km / h bit 2: velocidad del viento: 1 = nudo bit 3: velocidad del viento: 1 = m / h bit 4: velocidad del viento: 1 = bft |
| 19 | 1 | opciones de visualización 1 | bit 0: presión: 0 = absoluta, 1 = relativa bit 1: velocidad del viento: 0 = promedio, 1 = ráfaga bit 2: tiempo: 0 = 24 horas, 1 = 12 horas bit 3: fecha: 0 = día-mes-año, 1 = mes-día-año bit 4: escala de tiempo (?): 0 = 12 horas, 1 = 24 horas bit 5: fecha: 1 = mostrar año bit 6: fecha: 1 = mostrar el nombre del día bit 7: fecha: 1 = hora de alarma |
| 20 | 1 | opciones de visualización 2 | bit 0: temperatura exterior: 1 = temperatura bit 1: temperatura exterior: 1 = sensación térmica bit 2: temperatura exterior: 1 = punto de rocío bit 3: lluvia: 1 = hora bit 4: lluvia: 1 = día bit 5: lluvia: 1 = semana bit 6: lluvia: 1 = mes bit 7: lluvia: 1 = total |

Artículos

Mengarelli, J. L. et al., Software libre para la recolección y publicación de datos de estaciones meteorológica...

| | | | |
|----|---|---|--|
| 21 | 1 | alarma habilitar 1 | bit 1: tiempo bit 2: dirección del viento bit 4: humedad interior baja bit 5: humedad interior alta bit 6: humedad exterior baja bit 7: humedad exterior alta |
| 22 | 1 | alarma habilitar 2 | bit 0: media del viento bit 1: ráfaga de viento bit 2: lluvia por hora bit 3: lluvia diaria bit 4: presión absoluta baja bit 5: presión absoluta alta bit 6: presión relativa baja bit 7: presión relativa alta |
| 23 | 1 | alarma habilitada 3 | bit 0: temperatura interior baja bit 1: temperatura interior alta bit 2: temperatura exterior baja bit 3: temperatura exterior alta bit 4: viento frío bajo bit 5: viento frío alto bit 6: punto de rocío bajo bit 7: punto de rocío alto |
| 24 | 1 | zona horaria | |
| 25 | | | |
| 26 | 1 | datos actualizados | |
| 27 | 2 | recuento de datos | Número de lecturas almacenadas. |
| 29 | | | |
| 30 | 2 | posición actual | |
| 32 | 2 | presión relativa | La presión atmosférica relativa actual |
| 34 | 2 | presión absoluta | Presión atmosférica absoluta actual. |
| 36 | 2 | (3080 solamente) coeficiente de Lux | Divide por 10. |
| 38 | 2 | (2080) Factor de corrección del viento | Porcentaje en rango 75..125 |
| 40 | 2 | (2080) Compensación de temperatura exterior | Multiplícala por 0.1 para obtener ° C |
| 42 | 2 | (2080) Compensación de temperatura interior | Multiplícala por 0.1 para obtener ° C? |
| 43 | 5 | (1080) Fecha y hora actual | |
| 44 | 2 | (2080) Compensación de humedad exterior | Unidades desconocidas |
| 46 | 2 | (2080) Compensación de humedad interior | Unidades desconocidas |
| 48 | 1 | alarma, humedad interior, alta | |
| 49 | 1 | alarma, humedad interior, baja | |
| 50 | 2 | alarma, temperatura interior, alta | Multiplícala por 0.1 para obtener ° C. |
| 52 | 2 | alarma, temperatura interior, baja | Multiplícala por 0.1 para obtener ° C. |
| 55 | 1 | alarma, humedad exterior, baja | |
| 56 | 2 | alarma, temperatura exterior, alta | Multiplícala por 0.1 para obtener ° C. |

| | | | |
|-----|---|--|---|
| 58 | 2 | alarma, temperatura exterior, baja | Multiplica por 0.1 para obtener ° C. |
| 60 | 2 | alarma, viento frío, alto | Multiplica por 0.1 para obtener ° C. |
| 62 | 2 | alarma, viento frío, bajo | Multiplica por 0.1 para obtener ° C. |
| 64 | 2 | alarma, punto de rocío, alto | Multiplica por 0.1 para obtener ° C. |
| 66 | 2 | alarma, punto de rocío, bajo | Multiplica por 0.1 para obtener ° C. |
| 68 | 2 | alarma, presión absoluta, alta | Multiplica por 0.1 para obtener hPa. |
| 70 | 2 | alarma, presión absoluta, baja | Multiplica por 0.1 para obtener hPa. |
| 72 | 2 | alarma, presión relativa, alta | Multiplica por 0.1 para obtener hPa. |
| 74 | 2 | alarma, presión relativa, baja | Multiplica por 0.1 para obtener hPa. |
| 76 | 1 | alarma, velocidad media del viento, Beaufort | |
| 77 | 2 | alarma, velocidad media del viento, m / s | Multiplica por 0.1 para obtener m / s. |
| 79 | 1 | alarma, ráfaga de viento, Beaufort | |
| 80 | 2 | alarma, ráfaga de viento, m / s | Multiplica por 0.1 para obtener m / s. |
| 82 | 1 | alarma, direccion viento | Multiplica por 22.5 para obtener ° del norte. |
| 83 | 2 | alarma, lluvia, por hora | Multiplica por 0.1 para obtener mm. |
| 85 | 2 | alarma, lluvia, diario | Multiplica por 0.1 para obtener mm. |
| 87 | 2 | hora de alarma | Hora y minuto. |
| 89 | 3 | (3080) alarma, iluminancia | |
| 92 | 1 | (3080 solamente) alarma, UV | |
| 93 | 1 | (3080) máximo, UV, valor | |
| 94 | 3 | (3080) máximo, iluminancia, valor | Multiplica por 0.1 para obtener lux. |
| 97 | | | |
| 98 | 1 | máxima, humedad interior | |
| 99 | 1 | Mínimo, humedad interior | |
| 100 | 1 | máxima, humedad exterior | |
| 101 | 1 | Mínimo, humedad exterior | |
| 102 | 2 | máximo, temperatura interior | Multiplica por 0.1 para obtener ° C. |
| 104 | 2 | mínimo, temperatura interior | Multiplica por 0.1 para obtener ° C. |
| 106 | 2 | máximo, temperatura exterior | Multiplica por 0.1 para obtener ° C. |
| 108 | 2 | mínimo, temperatura exterior | Multiplica por 0.1 para obtener ° C. |
| 110 | 2 | máximo, viento | Multiplica por 0.1 para obtener ° C. |
| 112 | 2 | mínimo, sensación térmica | Multiplica por 0.1 para obtener ° C. |
| 114 | 2 | máximo, punto de rocío | Multiplica por 0.1 para obtener ° C. |
| 116 | 2 | mínimo, punto de rocío | Multiplica por 0.1 para obtener ° C. |
| 118 | 2 | máximo, presión absoluta | Multiplica por 0.1 para obtener hPa. |
| 120 | 2 | mínimo, presión absoluta | Multiplica por 0.1 para obtener hPa. |

Artículos

Mengarelli, J. L. et al., Software libre para la recolección y publicación de datos de estaciones meteorológica...

| | | | |
|-----|---|--|--|
| 122 | 2 | máximo, presión relativa | Multiplica por 0.1 para obtener hPa. |
| 124 | 2 | mínimo, presión relativa | Multiplica por 0.1 para obtener hPa. |
| 126 | 2 | máximo, velocidad media del viento | Multiplica por 0.1 para obtener m / s. |
| 128 | 2 | máximo, ráfaga de viento | Multiplica por 0.1 para obtener m / s. |
| 130 | 2 | máximo, lluvia por hora | Multiplica por 0.1 para obtener mm. |
| 132 | 2 | máximo, lluvia diaria | Multiplica por 0.1 para obtener mm. |
| 134 | 2 | máximo, lluvia semanal | Multiplica por 0.1 para obtener mm. |
| 136 | 2 | máximo, lluvia mensual | Multiplica por 0.1 para obtener mm. |
| 138 | 2 | máximo, lluvia total | Multiplica por 0.1 para obtener mm. |
| 140 | 1 | Picadillo alto por mes y lluvia total. | |
| 141 | 5 | máxima, humedad interior | |
| 146 | 5 | mínimo, humedad interior | |
| 151 | 5 | máxima, humedad exterior | |
| 156 | 5 | mínimo, humedad exterior | |
| 161 | 5 | máximo, temperatura interior | |
| 166 | 5 | mínimo, temperatura interior | |
| 171 | 5 | máxima, temperatura exterior | |
| 176 | 5 | mínimo, temperatura exterior | |
| 181 | 5 | máximo, viento frío | |
| 186 | 5 | mínimo, viento frío | |
| 191 | 5 | máximo, punto de rocío | |
| 196 | 5 | mínimo, punto de rocío | |
| 201 | 5 | máxima, presión absoluta | |
| 206 | 5 | presión mínima, absoluta | |
| 211 | 5 | máxima, presión relativa | |
| 216 | 5 | presión mínima, relativa | |
| 221 | 5 | máxima, velocidad media del viento | |
| 226 | 5 | máximo, ráfaga de viento | |
| 231 | 5 | máximo, lluvia por hora | |
| 236 | 5 | máximo, lluvia diaria | |
| 241 | 5 | máximo, lluvia semanal | |
| 246 | 5 | máximo, lluvia mensual | |
| 251 | 5 | máximo, lluvia total | |

Tabla 2. Diseño de registro almacenado por la estación

| Dirección | Bytes | Item | Comentario |
|-----------|-------|--------------------------------|--|
| 0 | 1 | delay | Minutos desde la última lectura almacenada |
| 1 | 1 | Humedad interna | |
| 2 | 2 | Temperatura interna | Multiplícala por 0.1 para obtener ° C. |
| 4 | 1 | Humedad externa | |
| 5 | 2 | Temperatura externa | Multiplícala por 0.1 para obtener ° C. |
| 7 | 2 | Presión absoluta | Multiplícala por 0.1 para obtener hPa. |
| 9 | 1 | Velocidad media del viento | Multiplícala por 0.1 para obtener m / s. |
| 10 | 1 | Velocidad del viento en ráfaga | Multiplícala por 0.1 para obtener m / s. |
| 11 | 1 | Velocidad del viento | Los 4 bits más bajos son los bits altos de velocidad del viento promedio, los 4 bits superiores son los bits altos de velocidad del viento de ráfaga |
| 12 | 1 | Dirección del viento | Multiplícala por 22.5 para obtener ° del norte |
| 13 | 2 | lluvia total | Multiplícala por 0.3 para obtener mm. |
| 15 | 1 | Estado | bit 6: 1 = pérdida de contacto con los sensores. bit 7: 1 = desbordamiento del contador de lluvia |
| 16 | 3 | Iluminancia | Multiplícala por 0.1 para obtener lux. |
| 19 | 1 | uv | |

Tabla 3. Diseño de registro creado con los datos obtenidos de la estación

| Índice | Valor | Descripción | Ejemplo |
|--------|----------------------|------------------------------|---------------------|
| 0 | fecha | Fecha de lectura | 05/06/2019 00:47 |
| 1 | fechaISO | Fecha en formato ISO | 2019-06-05T00:47:24 |
| 2 | datos['rf.temp_out'] | Temperatura externa | 16.2 |
| 3 | datos['rf.temp_in'] | Temperatura interna | 21.1 |
| 4 | datos['rf.hum_out'] | Humedad externa | 81 |
| 5 | datos['rf.hum_in'] | Humedad interna | 62 |
| 6 | datos['rf.lux'] | Intensidad luxer | 15 |
| 7 | datos['rf.uv'] | UV | 35 |
| 8 | VientoEnKm | Velocidad del viento en k/h. | 25 |

| | | | |
|----|------------------------------------|----------------------------------|---------------------|
| 9 | RafagaEnKm | Velocidad máxima de ráfaga km/h. | 38 |
| 10 | datos['rf.wind_dir'] | Dirección del viento | 45 |
| 11 | VientoDireccion | Dirección del viento en nombre | Nordeste |
| 12 | datos['rf.abs_pressure'] | Presión absoluta | 1008 |
| 13 | configuraciones['fb.rel_pressure'] | Presión relativa | 1013.2 |
| 14 | datos['rf.rain'] | Lluvia total | 0 |
| 15 | punto_de_rocio | Punto de rocío | 12.91 |
| 16 | fechaUTC | Fecha en UTC | 2019-06-05+03:47:24 |
| 17 | datos['rf.status'] | Estado de los sensores | {0,0,0,0,0,0,0} |

Tabla 4. Validaciones sobre los datos obtenidos de la estación

| Campo | Valor válido |
|------------------------------------|-------------------------------------|
| fecha | No se realiza validación |
| fechaISO | No se realiza validación |
| datos['rf.temp_out'] | mayor a -50 y menor que 100 |
| datos['rf.temp_in'] | mayor a -50 y menor que 100 |
| datos['rf.hum_out'] | mayor/igual a 0 y menor/igual a 100 |
| datos['rf.hum_in'] | mayor/igual a 0 y menor/igual a 100 |
| datos['rf.lux'] | mayor a 0. Acepta nulos |
| datos['rf.uv'] | mayor a 0 y menor a 20 |
| VientoEnKm | mayor a 0 y menor a 1000 |
| RafagaEnKm | mayor a 0 y menor a 1000 |
| datos['rf.wind_dir'] | mayor a 0 y menor/igual a 360 |
| VientoDireccion | No se realiza validación |
| datos['rf.abs_pressure'] | mayor a 0 |
| configuraciones['fb.rel_pressure'] | mayor a 0 |
| datos['rf.rain'] | mayor a 0 |
| punto_de_rocio | mayor a -50 y menor que 100 |
| fechaUTC | No se realiza validación |
| datos['rf.status'] | No se realiza validación |

Tabla 5. Validaciones sobre los datos antes de ser almacenados

| Campo | Tipo | Mínimo | Máximo |
|---------------------|-----------|--------|--------|
| fecha | Timestamp | - | Ahora |
| temperatura_externa | Decimal | 0 | 100 |
| temperatura_interna | Decimal | 0 | 100 |
| humedad_externa | Decimal | 0 | 10 |
| humedad_interna | Decimal | 0 | 100 |
| luxer_uv | Decimal | 0 | 100 |
| viento_direccion | Decimal | 0 | 360 |

Tabla 6. Campos de la tabla auth_user

| Columnas | Tipo de Dato | Descripción |
|--------------|-------------------------|----------------------------|
| id | integer | id del usuario |
| password | character varying (128) | Contraseña |
| last_login | timestamp | Último login |
| is_superuser | boolean | Superusuario (true/false) |
| username | character varying (30) | Cuenta del usuario |
| first_name | character varying (150) | Nombre de usuario |
| last_name | character varying (150) | Apellido de usuario |
| email | character varying (254) | Correo electrónico |
| is_staff | boolean | Sin uso |
| is_active | boolean | Cuenta activa (true/false) |
| date_joined | timestamp | Fecha de alta de la cuenta |

Tabla 7. Campos de la tabla climapp_estaciones

| Columnas | Tipo de Dato | Descripción |
|--------------------|-------------------------|--------------------------------------|
| id_estacion | integer | Identificador de estación |
| fecha_alta | timestamp | Fecha de alta de la estación |
| fecha_baja | timestamp | Fecha de baja de la estación |
| fecha_modificación | timestamp | Fecha de modificación de los datos |
| activa | boolean | Activa (true/false) |
| clave_lectura | uuid | Clave de identificación de lectura |
| clave_escritura | uuid | Clave de identificación de escritura |
| modelo | character varying (100) | Marca y modelo de la estación |
| país | character varying (100) | País |
| ciudad | character varying (100) | Ciudad |
| calle | character varying (100) | Calle |

| | | |
|---------------------------|------------------------|---|
| latitud | numeric(10,6) | Latitud de ubicación |
| longitud | numeric(10,6) | Longitud de ubicación |
| altura | numeric(10,6) | Altura respecto al mar |
| fecha_ultimo_registro | timestamp | Fecha en que envió el último registro |
| lluvia_acumulado_hora | numeric(10,2) | Lluvia acumulada por hora en mm |
| lluvia_acumulado_diario | numeric(10,2) | Lluvia acumulada por día en mm |
| lluvia_acumuladio_semanal | numeric(10,2) | Lluvia acumulada por semana en mm |
| lluvia_acumulado_anual | numeric(10,2) | Lluvia acumulada por año en mm |
| lluvia_acumulado_total | numeric(10,2) | Lluvia acumulada por total en mm |
| id_usuario_id | integer | Identificador del usuario que la registró |
| nombre | character varyin (100) | Nombre de usuario |

Tabla 8. Campos de la tabla climapp_registros

| Columnas | Tipo de Dato | Descripción |
|-------------------------|-------------------------|--|
| id_registro | integer | Identificador de registro |
| fecha | timestamp | Fecha en que se creo el registro (en el cliente) |
| fecha_ingreso | timestamp | Fecha en que llego a EmaCenter |
| fecha_alta | timestamp | Fecha en que se agregó a la tabla |
| fecha_baja | timestamp | Fecha en que se quitó de la tabla |
| software | character varying (100) | Programa que envió el registro |
| viento_direccion_nombre | character varying (100) | Dirección del viento expresada con nombre |
| temperatura_externa | numeric(10,2) | Temperatura externa |
| temperatura_interna | numeric(10,2) | Temperatura interna |
| humedad_externa | numeric(10,2) | Humedad interna (en %) |
| humedad_interna | numeric(10,2) | Humedad interna (en %) |
| punto_de_rocio | numeric(10,2) | Punto de rocío |
| luxer_intensidad | numeric(10,2) | Luxer intensidad |
| luxer_uv | numeric(10,2) | Luxer uv |
| viento_velocidad | numeric(10,2) | Velocidad del viento |
| viento_rafagas | numeric(10,2) | Velocidad de ráfaga más fuerte |
| viento_direccion | numeric(10,2) | Dirección del viento expresada en grados |
| presion_absoluta | numeric(10,2) | Presión absoluta |
| presion_relativa | numeric(10,2) | Presión relativa |
| lluvia_actual | numeric(10,2) | Lluvia actual en mm |
| lluvia_acumulado_hora | numeric(10,2) | Lluvia acumulada por hora en mm |

| | | |
|--------------------------|---------------|--|
| lluvia_acumulado_diario | numeric(10,2) | Lluvia acumulada por día en mm |
| lluvia_acumulado_semanal | numeric(10,2) | Lluvia acumulada por semana en mm |
| lluvia_acumulado_mensual | numeric(10,2) | Lluvia acumulada por mes en mm |
| lluvia_acumulado_anual | numeric(10,2) | Lluvia acumulada por año en mm |
| lluvia_acumulado_total | numeric(10,2) | Lluvia acumulada total en mm |
| id_estacion_id | numeric(10,2) | Identificador de la estación que envió el registro |
| sin_acumulado_lluvia | boolean | Si posee acumulado (true/false) |

Tabla 9. Rutas a consultar en la API EMA Center

| URL | Auth | Descripción |
|--|------|--|
| /api/user | Si | Listado de usuarios |
| /api/user/<id> | Si | Información detallada del usuario |
| /api/station | No | Listado de estaciones |
| /api/station/<id> | No | Información detallada de la estación |
| /api/station/<id>/registros | No | Listado de los últimos registros de la estación |
| /api/station/<id>/estadisticas | No | Listado de los últimos registros de la estación |
| /api/estacion | No | Listado de estaciones |
| /api/estacion/<id> | No | Información detallada de la estación |
| /api/estacion/<id>/registros | No | Listado de los últimos registros de la estación |
| /api/estacion/<id>/estadisticas | No | Listado de los últimos registros de la estación |
| /api/record/<id> | Si | Información detallada del registro meteorológico |
| /api/data/<clave_lectura> | No | Información detallada de la estación |
| /api/data/<clave_lectura>/registros | No | Listado de los últimos registros de la estación |
| /api/data/<clave_lectura>/estadisticas | No | Listado de los últimos registros de la estación |

Tabla 10. Información devuelta según la petición

| Ruta | Información devuelta |
|-----------------------------|---|
| /api/station/<id> | {id_estacion, nombre, modelo, pais, ciudad, calle, latitud, longitud, altura, fecha_ultimo_registro, registros, estadisticas} |
| /api/station/<id>/registros | {id_registro, fecha, fecha_ingreso, software, viento_direccion_nombre, temperatura_externa, temperatura_interna, humedad_externa, humedad_interna, punto_de_rocio, luxer_intencidad, luxer_uv, viento_velocidad, viento_rafagas, viento_direccion, presion_absoluta, presion_relativa, lluvia_actual, lluvia_acumulado_hora, lluvia_acumulado_diario, lluvia_acumulado_semanal, lluvia_acumulado_mensual, lluvia_acumulado_anual, lluvia_acumulado_total} |

| | |
|--|---|
| /api/station/<id>/estadisticas | {hora, temperatura_externa_max, temperatura_externa_min, humedad_externa_max, lluvia_acumulado_hora_max, lluvia_acumulado_hora_min, luxer_intencidad_max, luxer_intencidad_min, viento_velocidad_max, viento_velocidad_min, presion_absoluta_max, presion_absoluta_min, luxer_uv_max, luxer_uv_min, presion_relativa_max, presion_relativa_min, humedad_externa_min, viento_rafagas_max} |
| /api/estacion | {id_estacion, nombre, modelo, pais, ciudad, calle, latitud, longitud, altura, fecha_ultimo_registro, registros, estadisticas} |
| /api/estacion/<id> | {id_estacion, nombre, modelo, pais, ciudad, calle, latitud, longitud, altura, fecha_ultimo_registro, registros, estadisticas} |
| /api/estacion/<id>/registros | {id_registro, fecha, fecha_ingreso, software, viento_direccion_nombre, temperatura_externa, temperatura_interna, humedad_externa, humedad_interna, punto_de_rocio, luxer_intencidad, luxer_uv, viento_velocidad, viento_rafagas, viento_direccion, presion_absoluta, presion_relativa, lluvia_actual, lluvia_acumulado_hora, lluvia_acumulado_diario, lluvia_acumulado_semanal, lluvia_acumulado_mensual, lluvia_acumulado_anual, lluvia_acumulado_total} |
| /api/estacion/<id>/estadisticas | {hora, temperatura_externa_max, temperatura_externa_min, humedad_externa_max, lluvia_acumulado_hora_max, lluvia_acumulado_hora_min, luxer_intencidad_max, luxer_intencidad_min, viento_velocidad_max, viento_velocidad_min, presion_absoluta_max, presion_absoluta_min, luxer_uv_max, luxer_uv_min, presion_relativa_max, presion_relativa_min, humedad_externa_min, viento_rafagas_max} |
| /api/record/<id> | {id_registro, fecha, fecha_ingreso, software, viento_direccion_nombre, temperatura_externa, temperatura_interna, humedad_externa, humedad_interna, punto_de_rocio, luxer_intencidad, luxer_uv, viento_velocidad, viento_rafagas, viento_direccion, presion_absoluta, presion_relativa, lluvia_actual, lluvia_acumulado_hora, lluvia_acumulado_diario, lluvia_acumulado_semanal, lluvia_acumulado_mensual, lluvia_acumulado_anual, lluvia_acumulado_total} |
| /api/data/<clave_lectura> | {id_estacion, nombre, modelo, pais, ciudad, calle, latitud, longitud, altura, fecha_ultimo_registro, registros, estadisticas} |
| /api/data/<clave_lectura>/registros | {id_registro, fecha, fecha_ingreso, software, viento_direccion_nombre, temperatura_externa, temperatura_interna, humedad_externa, humedad_interna, punto_de_rocio, luxer_intencidad, luxer_uv, viento_velocidad, viento_rafagas, viento_direccion, presion_absoluta, presion_relativa, lluvia_actual, lluvia_acumulado_hora, lluvia_acumulado_diario, lluvia_acumulado_semanal, lluvia_acumulado_mensual, lluvia_acumulado_anual, lluvia_acumulado_total} |
| /api/data/<clave_lectura>/estadisticas | {hora, temperatura_externa_max, temperatura_externa_min, humedad_externa_max, lluvia_acumulado_hora_max, lluvia_acumulado_hora_min, luxer_intencidad_max, luxer_intencidad_min, viento_velocidad_max, viento_velocidad_min, presion_absoluta_max, presion_absoluta_min, luxer_uv_max, luxer_uv_min, presion_relativa_max, presion_relativa_min, humedad_externa_min, viento_rafagas_max} |

Cita: Mengarelli, J. L.; Aramburu, E.; Sbarbaro, M. M. y C. F. Bonnet, 2024. “Software libre para la recolección y publicación de datos de estaciones meteorológica automáticas” (pp. 11-38), *@archivos de Ciencia y Tecnología* N° 5, FCyT-UADER, Oro Verde.