

Desarrollo y evolución de sistema OES incluyendo datalogger y software asociado

Development and evolution of OES system, datalogger equipment and associated software

Pablo Bertinat^{†1}, Juan Salerno^{†2}, Marcelo Castello^{†3}, Rafael Oliva^{*4}

[†]*Observatorio de Energía y Sustentabilidad (OES), UTN FRRO
Zeballos 1341 - Rosario - Santa Fe (Argentina)*

¹pbertinat@frro.utn.edu.ar

²jsalerno@frro.utn.edu.ar

³mcastello@frro.utn.edu.ar

^{*}*Energías Alternativas / Instituto de Tecnología Aplicada*

Avda Gregores y Lero Rivera - Río Gallegos - Santa Cruz (Argentina)

⁴roliva@uarg.unpa.edu.ar

Abstract—This paper delineates the different stages in the design and evolution of a datalogger system and its associated software. It was developed within the framework of a Project aimed at the study and processing of wind data for power generation, and carried out with open source tools.

This project has focused on the development of in-house technology, with capabilities and performance comparable to equipment available on the market. Commercial off-the-shelf systems have been analysed, as well as associated software in order to set the general features and requirements for the prototype.

An important number of industry-standard features in the processing of wind data for electric power generation have been considered. Hardware was developed and a web-based system located in an in-house server have been deployed, both of which work together to fulfill the proposed functions.

Keywords: datalogger; wind energy; measurements; software

Resumen— El presente trabajo detalla las etapas en el diseño y la evolución de un datalogger y su software asociado, desarrollados en el marco de un Proyecto orientado al relevamiento y procesamiento de datos de potencial eólico con fines energéticos mediante herramientas de acceso libre.

El presente proyecto se ha enfocado en el desarrollo y ensayos de desempeño de un equipo de tecnología propia, que busca equiparar en prestaciones a los disponibles en plaza. Para el diseño del equipo, se han examinado las capacidades y el funcionamiento general de equipos disponibles comercialmente con la finalidad de establecer las características generales que debe tener el prototipo.

Se analizaron las particularidades requeridas en el procesamiento de la información del potencial eólico para la producción de energía eléctrica. Se ha desarrollado un conjunto de hardware y sistema web alojado en un servidor propio que operan coordinadamente para cumplir las funciones propuestas.

Palabras clave: datalogger; energía eólica; mediciones; software.

I. INTRODUCCIÓN

El énfasis del desarrollo que se presenta, ha sido puesto en el diseño de una cadena completa de medición -que incluye hardware y software- para un sistema de adquisición de datos orientado a mediciones meteorológicas para aplicaciones de energía eólica según el estándar IEC [1]. Las

señales más importantes, según el análisis realizado en [2] son las de intensidad y dirección de viento, temperatura y presión atmosférica. En la normativa IEC se analizan también aspectos relativos a la calibración y tratamiento de incertidumbres asociadas a la medición, cuya aplicación para equipos similares ha sido evaluada en [3] y en [4]. Varias adaptaciones a condiciones locales se sugieren en [5].

La producción de equipamiento apto para mediciones de viento es una tarea compleja, y a tal efecto se han analizado equipos y paquetes de software aceptados y utilizados habitualmente en la industria a nivel internacional. Se ha trabajado con equipos de origen norteamericano NRG Symphonie [6], provisto con el software básico de reportes Data Retriever [7], los Vaisala-Nomad2 (antes de Secondwind) [8] y su software asociado Nomad Desktop (todavía obtenible a través de Vaisala en [9]), y con equipos de origen alemán Ammonit [10]. En el campo de los dataloggers de precisión se ha trabajado en la interfaz y programación de los modelos CR850, CR1000 y CR1000X de Campbell Scientific [11] y su software asociado Loggernet [12]. Asimismo se ha trabajado con el software Argentina Map desarrollado en su momento por el CREE / Dr. H. Mattio [13] y sucesivas versiones académicas del software Windographer, hoy propiedad de UL [14]. Si bien cada sistema se adapta a aplicaciones específicas, una característica común en estos productos es que se trata de desarrollos propietarios, algo que se da en casi toda la industria de relevamiento eólico. Esto hace que los equipos y software disponibles en Argentina sean de origen extranjero, su utilización onerosa y en gran medida atada a criterios técnico-económicos externos, que requieren adaptación y entrenamiento especial en caso de cambios o actualizaciones. El impulso a una alternativa de desarrollo de tecnología propia destinada a este fin, que pueda ser adaptada a las necesidades de usuarios locales, y que utilice mayormente licencias abiertas (con la consiguiente disminución de costos) evitando la dependencia tecnológica en equipos y en software, es lo que ha generado la presente propuesta de trabajo.

La posibilidad de generar energía mediante el viento implica el previo conocimiento de una serie de condi-

ciones locales que son propias del sitio que se pretende elegir para las instalaciones. Estas condiciones son diversas y heterogéneas, y es posible mencionar entre ellas: la disponibilidad de los terrenos para el emplazamiento de los molinos, la cercanía y accesibilidad a las líneas de transmisión eléctrica donde se insertarán los generadores, las condiciones locales de estas líneas -tensión, capacidad, potencia de cortocircuito, etc.- la accesibilidad de caminos para la logística de traslado de equipos de construcción, personal y los propios molinos, entre muchas otras. Pero las de mayor incidencia para decidir la elección de un sitio en particular, son las condiciones del recurso eólico disponible. La evaluación precisa y ajustada de este recurso y el efecto multiplicador que sus posibles variaciones e incertidumbres imprimen en los resultados energéticos y económicos de la futura instalación, remarcan la importancia asignada y los recursos empleados en lograr una alta previsibilidad en los resultados. Mediante este aporte se espera poder mejorar las condiciones para el aprovechamiento de esta fuente de energía renovable así como la generación de capacidades locales, tanto en cadenas de valor tecnológico en el área como en el desarrollo del recurso humano necesario.

II. OBJETIVOS DEL PROYECTO

Partiendo de las premisas indicadas, puede mencionarse como *objetivo general*, ya explicitado en el mencionado trabajo de tesis [2] es el siguiente: alcanzar el desarrollo de un prototipo de datalogger y un sistema de procesamiento de información de licencias abiertas, que permitan adquirir, transmitir, almacenar y analizar la información proveniente de diversos tipos de sensores asociados a la medición de potencial eólico con fines energéticos. Los *objetivos específicos* se pueden enumerar como sigue:

- Evaluar el desempeño de un equipo de tecnología propia, equivalente en prestaciones y calidad a los disponibles en plaza, con las capacidades y adaptaciones necesarias para estudios locales.
- Desarrollar un sistema para la visualización en tiempo real de las variables capturadas por el equipo datalogger.
- Explorar, relacionar y desarrollar herramientas de procesamiento de la información elaborada en base a software no propietario con el hardware desarrollado. Estos sistemas estarán basados en tecnología web y podrán estar disponibles a nivel global previa autenticación de los usuarios.

El desarrollo de un datalogger es una tarea compleja, más aún cuando el objetivo implica crear un sistema completo que permita conocer las características y el potencial energético de los vientos locales, con posibilidad de tratamiento de la información recogida. Se entiende por sistema en este caso al conjunto de equipo, el software y una metodología de trabajo. Dicho conjunto debe ser capaz de analizar variables y crear gráficos que relacionen estas variables, determinen potencial energético y orienten hacia la concreción de proyectos eólicos.

III. ANTECEDENTES

El trabajo se ha llevado adelante en el marco del Proyecto de Investigación Homologado PID "Desarrollo de

equipamiento y procesamiento de datos de potencial eólico con fines energéticos median herramientas de software libre", código ENUTIRO0002136TC, dirigido por Pablo Bertinat. Antes de abordar el diseño, se han examinado las prestaciones y el funcionamiento general de equipos disponibles en plaza como los citados en I.) con la finalidad de determinar las características generales que se deben proveer. Se buscó de esta forma encontrar los puntos importantes y comunes que deben atenderse para establecer funciones a incorporar en el diseño.

IV. DISCUSIÓN SOBRE LAS ALTERNATIVAS DEL HARDWARE

Las premisas planteadas (consulta en tiempo real de las mediciones, almacenamiento interno y configuración online), implican que se debía optar para la construcción del sistema, por una arquitectura de hardware. Dentro de las opciones disponibles se consideraron desde las MicroPC (RaspberryPi, PandaBoard, Beagleboard, etc.) con variantes de Linux hasta los microcontroladores de diversas prestaciones (8 bits, 32 bits). Se le dió especial importancia al objetivo del equipo: la medición de las variables meteorológicas por sobre las otras premisas. Los tiempos de procesamiento relativamente bajos y requerimientos de simplicidad llevaron a elegir una arquitectura basada en microcontrolador sin RTOS, trabajando en modalidad de lazo infinito con temporización por interrupciones. Se optó por una plataforma LPC1769 [15] (ARM Cortex M3) de 32 bits, específicamente utilizando la placa de desarrollo LPCXpresso 1769 [16] de amplia disponibilidad. En la figura 1 se aprecia la plataforma seleccionada. La combinación de firmware y controlador seleccionada permite:

- Medición de variables en tiempo real.
- Utilización de bibliotecas de bajo nivel en C escritas y mantenidas por el fabricante, con licencias flexibles, que facilitan la escritura de la aplicación.
- Acceso a herramientas de desarrollo gratuitas o de bajo costo: IDE basado en Eclipse [17] compilador gcc [18], programador y debugger LPC Link2 [19].
- Posibilidad a futuro de incorporar FreeRTOS [20] como sistema operativo, dada la existencia de un port a dicho microcontrolador y una importante base de usuarios.

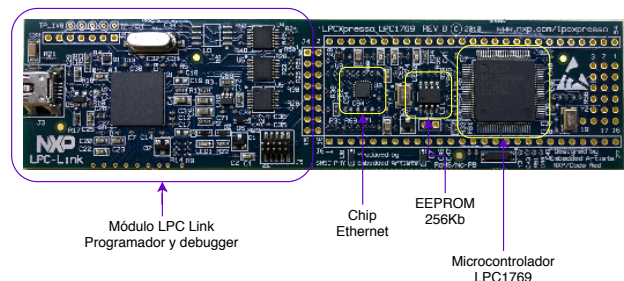


Fig. 1. LPCXpresso 1769 con microcontrolador Cortex M3, 32 bits

V. ARQUITECTURA DEL SISTEMA

El diseño del sistema completo requiere la coordinación del funcionamiento de diversos componentes. Para su explicación puede ser dividido en dos partes fundamentales:

- Hardware y firmware del datalogger

- Comunicación con servidor remoto
- Sistema de visualización y persistencia de datos

A. Hardware y firmware del datalogger

Observando el diagrama en bloques de la figura 2, puede discriminarse su funcionamiento en los siguientes pasos:

- 1) Circuitos de entrada de señales
- 2) Digitalización
- 3) Filtrado digital (en desarrollo)
- 4) Muestreo y procesamiento
- 5) Almacenamiento en RAM
- 6) Sincronización con SD
- 7) Procesador de comandos

1) *Circuitos de entrada de señales:* Los circuitos desarrollados realizan un tratamiento diferenciado para dos tipos básicos de señal a procesar: en primer lugar la señal de los anemómetros que proporcionan una tensión de CA de amplitud y frecuencia variable en un rango de 0 a 100 Hz, con información de velocidad de viento contenida en la frecuencia, y por otro lado el resto de las señales que son del tipo analógico y de variación lenta. Los criterios de diseño del hardware han contemplado la posibilidad de agregar más canales tanto de frecuencia como analógicos en futuros desarrollos, para el caso de requerirse su empleo en una torre con mayor cantidad de alturas de medición. El modelo actual se desarrolló para 2 alturas.

Las etapas de entrada de todas las señales incorporan filtrado analógico activo. En el caso de los anemómetros se realiza un filtrado del tipo Sallen-Key de 2° orden, y posteriormente se agrega una etapa de conformación de la onda necesaria para su tratamiento posterior. En la figura 3 se puede apreciar el circuito del filtro mencionado con su conformador de señal. En el caso de las variables analógicas sólo se incorporó un filtro activo de primer orden. En la figura 4 se aprecia la sección analógica de la placa del datalogger.

2) *Digitalización:* Las señales analógicas provenientes de los filtros, se aplican al subsistema de conversión analógico digital. Se utilizó el circuito integrado MCP3208 [21], de Microchip, un conversor A/D de 12 bits de aproximaciones sucesivas. Su tecnología CMOS de muy bajo consumo (500 nA típicos) permite su utilización en circuitos alimentados por baterías. Su rango de temperaturas de trabajo, de -40 a +85°C lo hacen apto para aplicaciones industriales y de campo.

Se prefirió utilizar un conversor externo y no el integrado en el microcontrolador por razones de escalabilidad para futuras implementaciones que impliquen medición de mayor cantidad de variables analógicas. Al conectarse este chip al microcontrolador a través de un bus SPI es relativamente simple el agregado de más dispositivos conversores. En la figura 5 se pueden apreciar las dos unidades MCP3208 utilizadas para este proyecto.

3) *Filtrado digital (en desarrollo):* Se encuentra en desarrollo una etapa de filtrado digital que permitirá analizar las señales ingresadas, evaluando su pendiente. Luego, en base a dicho valor el microcontrolador decide para cada medición en base a las pendientes encontradas

históricamente si incorpora o descarta ese valor. Por el momento, sólo se utiliza el filtro analógico de la sección de acondicionamiento de señal.

4) *Muestreo y Procesamiento:* Se ha considerado importante incorporar en el firmware la posibilidad de configurar los tiempos de muestreo y el de producción de registros. Dichos tiempos se almacenan en dos variables:

- *im:* intervalo de muestreo.
Intervalo de tiempo entre cada medición (analógica y digital). Su valor por defecto es 1 seg.
- *cr:* cantidad de mediciones por registro.
Es la cantidad de mediciones necesarias para la producción de un registro. Su valor por defecto es 600.

Los valores por defecto mostrados, son los que provienen de la recomendación de las mencionadas normas [1] y [5] para equipos eólicos en conexión a red.

El proceso realiza una lectura de cada canal de entrada en el intervalo de tiempo determinado por *im*. A esta medición se la denomina dato crudo, cuya estructura puede apreciarse en la figura 6. Cuando la cantidad de mediciones llega a *cr*, tiene lugar el procesamiento, que consiste en el cálculo del promedio, desvío standard, máximo y mínimo a partir de los datos crudos leídos y para cada variable muestreada. Se forma así un registro, que está compuesto por los valores procesados para cada canal de entrada, según la estructura que se aprecia en la figura 7.

5) *Almacenamiento en RAM:* Cada *cr* segundos, según la normativa utilizada se graba en memoria RAM el registro de los datos generados.

6) *Sincronización con SD:* Se denomina así al proceso mediante el cual los registros de la RAM pasan a la memoria SD extraíble. Si bien -como ya se ha dicho- el equipo cuenta con transmisión de los datos medidos para que sean guardados en una base de datos remota, también se incorpora una tarjeta SD donde se almacenan los mismos. Se le llama sincronización ya que mediante algoritmos de control este proceso hace que el contenido de la RAM y de la memoria externa SD sean idénticos, no permitiendo la falta ni la duplicación de datos. Si por alguna razón se interrumpe la escritura en SD, en el siguiente ciclo se reanuda desde el último dato no almacenado.

7) *Procesador de comandos:* El firmware posee un procesador de comandos accesible a través de dos capas físicas posibles:

- Puerto USB
- Ethernet (Telnet)

Los comandos están agrupados según su función, siendo ellas:

- Configuración/Información
- Actuación
- Gestión de SD

La sintaxis de los comandos es:

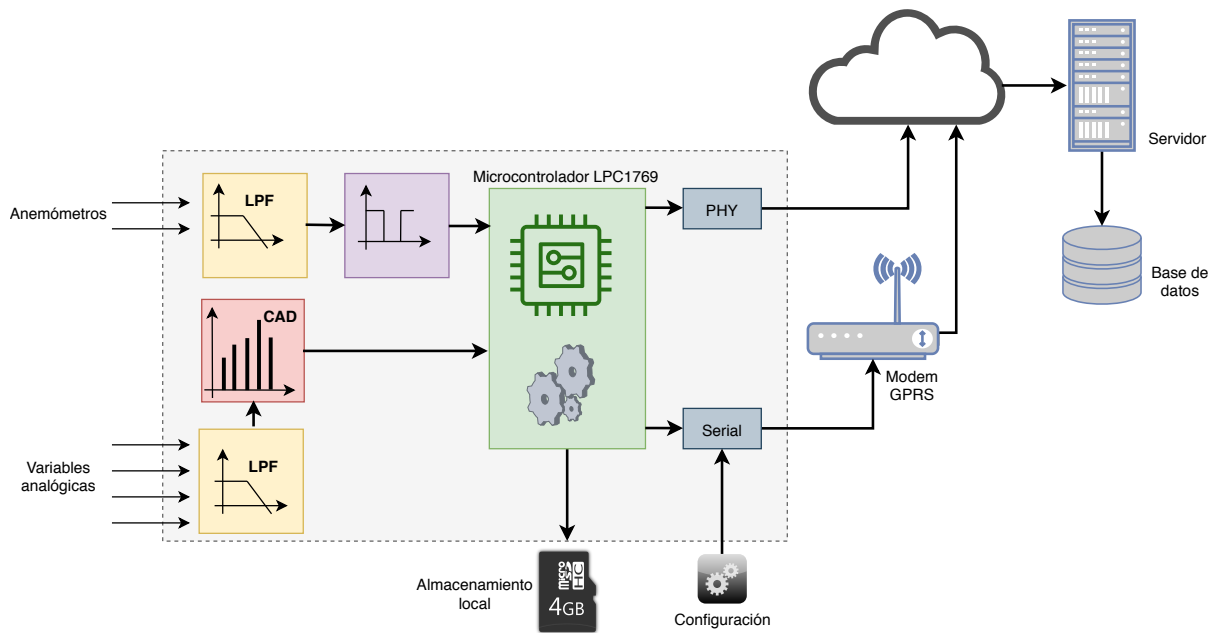


Fig. 2. Diagrama de bloques del datalogger.

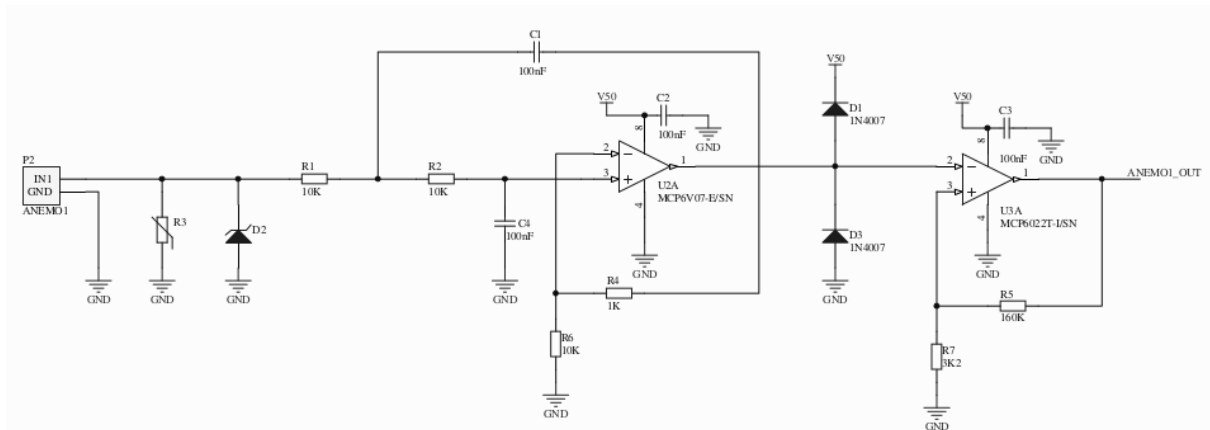


Fig. 3. Filtro y conformador de onda.

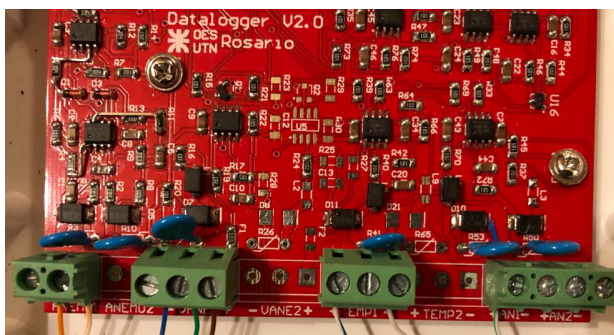


Fig. 4. Vista de la sección analógica del PCB



Fig. 5. Conversores A/D MCP3208

comando [parámetros][-h]

donde:

parámetros es la lista de parámetros con sus valores.

-h ayuda, cuando se especifica este parámetro se muestra una ayuda del comando.

Cuando no se especifica ningún parámetro el co-

mando actúa como comando de información, si pertenece al grupo Configuración/Información. Algunos comandos tienen parámetros obligatorios: *sms*, *ls*, *cd*, *tail*

Los parámetros van separados por espacios y precedidos



Fig. 6. Dato crudo

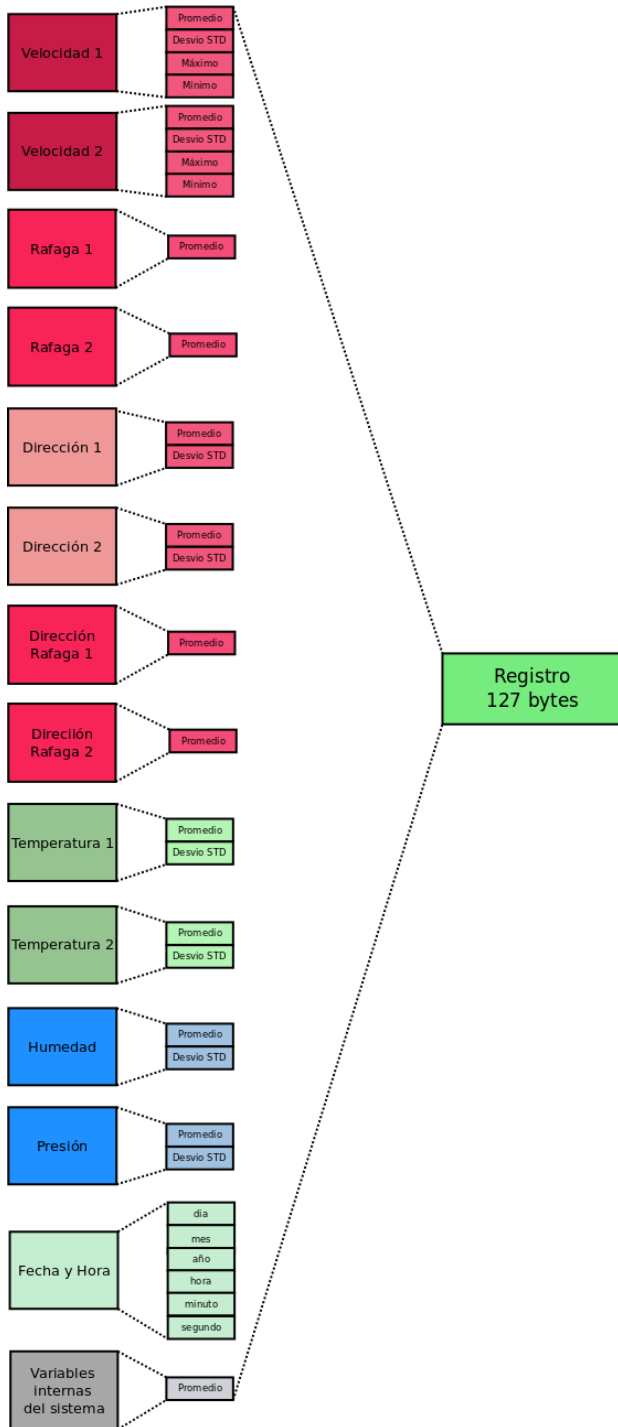


Fig. 7. Estructura de un registro

por el signo -, por ejemplo el comando *rtc* cambia/muestra el estado del reloj de tiempo real, y la secuencia:

```
rtc -A 2020 -M 10 -D 12 -h 20 -m 15
```

ajusta la fecha y hora del reloj de tiempo real a: 12/10/2020 20:15

Las constantes de calibración de los anemómetros y otros sensores se configuran a través del comando *sensconf* y se almacenan en EEPROM. En el caso de los anemómetros, estas constantes vienen frecuentemente certificadas por un laboratorio externo de calibración. Además de ello, se realiza una calibración de bajo nivel en frecuencia para los canales de viento del datalogger siguiendo un procedimiento similar al indicado en [22], y en tensión para los canales analógicos utilizando instrumental de baja incertidumbre.

En las figuras 8 y 9 se pueden ver las salidas por terminal de los comandos *ifconfig* y *read* respectivamente.

```
mcastello@marcelo-desktop:~$ ifconfig -h
File Modifica Visualizza Cerca Terminale Aiuto

WINDHOSE

datalogger 1.1>
ifconfig -h
Uso:
  ifconfig [-ip nueva ip
            [-tcp puerto TCP]
            [-p] muestra configuracion en formato paquete
Ejemplos:
  ifconfig -ip 192.168.1.250 -tcp 995 Ajusta ip y puerto TCP
  ifconfig -h Muestra este help

Ethernet
IP:192.168.1.201
Gateway:192.168.1.1
Puerto TCP Telnet:996
Puerto serie
Velocidad Uart1 (USB):115200
Velocidad Uart2 (GPRS):9600

datalogger 1.1>
```

Fig. 8. Ayuda del comando de configuración de red

```
mcastello@marcelo-desktop:~$ read
File Modifica Visualizza Cerca Terminale Aiuto
Datos en tiempo real

Vientos
-----
Velocidad 1 [m/s]: 3.56
Velocidad 1 max[m/s]: 6.33
Velocidad 1 min[m/s]: 1.84
Angulo veleta 1 [°]: 141.4
Rafaga 1 [m/s]: 0.00
Angulo rafaga 1 [°]: 0.0
Temperatura 1 [°C]: 20.93

Velocidad 2 [m/s]: 2.35
Velocidad 2 max[m/s]: 5.54
Velocidad 2 min[m/s]: 1.82
Angulo veleta 2 [°]: 126.2
Rafaga 2 [m/s]: 0.00
Angulo rafaga 2 [°]: 0.0
Temperatura 2 [°C]: -49.44

Canales analógicos
-----
Analog 1 [V]: 0.01
Analog 2 [V]: 0.01
```

Fig. 9. Resultado de la ejecución del comando read

En la Tabla 1 se ven los comandos disponibles y sus funciones.

VI. COMUNICACIÓN CON SERVIDOR REMOTO

Para que se puedan visualizar las mediciones en tiempo real, es necesario que el datalogger se comunique con un

Tabla 1: Comandos disponibles		
Comando	Función	Descripción
help	Info	Muestra los comandos disponibles
ver	Info	Version del hardware/firmware
datastat	Info	Muestra estadísticas de la RAM y SD
read	Info	Muestra datos en tiempo real
sysstat	Info	Muestra el estado del sistema
rtc	Info/config	Muestra/configura reloj
sensconf	Info/config	Muestra/configura la función de transferencia de los sensores
storeconf	Info/config	Muestra/configura almacenamiento de datos
ifconfig	Info/config	Muestra/configura la ethernet
id	Info/config	Muestra/configura las id origen-destino
gprs	Info/config	Muestra/configura el GPRS
exit	Actuación	Exit shell
freset	Actuación	Carga configuraciones por defecto
sms	Actuación	Inicia envío de SMS
ls	Gestión de SD	Muestra el directorio
cd	Gestión de SD	Cambia el directorio
tail	Gestión de SD	Muestra últimas n líneas del archivo
mount	Gestión de SD	Monta el sistema de archivos en la SD
umount	Gestión de SD	Desmonta el sistema de archivos de la SD

TABLE I
COMANDOS DISPONIBLES

servidor remoto donde se almacenan los datos medidos. Para esta comunicación se utilizó el modelo cliente-servidor, lo cual requirió el desarrollo una capa de software que se encargara de las comunicaciones de ambos lados: datalogger y lado servidor.

Se analizaron las alternativas disponibles para lograr la comunicación:

- Servidor dentro de la LAN de medición
- Servidor fuera de la LAN de medición y datalogger con redundancia de enlace
- Servidor fuera de la LAN de medición y datalogger vía GPRS

Finalmente se seleccionó esta última opción que incorpora *General Packet Radio Service*(GPRS), por considerarla la más adecuada. En la figura 10 se muestra un esquema de aplicación real del sistema, que reúne las ventajas del GPRS en cuanto a simplicidad de configuración, economía de desarrollo y su disponibilidad y gran difusión en Argentina (lo cual aumenta la posibilidad de instalar el sistema sin repetidoras en buena parte del país).

Para asegurar que no haya duplicación ni pérdida de datos en el proceso de transmisión, la aplicación implementa procesos de verificación y validación. En caso de interrupción de la comunicación, los datos persisten para ser enviados en el próximo ciclo de transmisión. Un algoritmo de marcas y comprobaciones se ha implementado para tener un registro de los datos efectivamente guardados en la base de datos. Así, ante una falla en la transmisión, el sistema retoma la sincronización sin pérdida de datos ni duplicación.

Existe también la posibilidad de recuperar los datos concurriendo al sitio del emplazamiento y extrayendo la tarjeta SD del datalogger y efectuando su lectura desde una PC compatible. En la figura 11 se ven dos casos de comunicación entre el datalogger y el servidor. El de la izquierda ilustra una comunicación exitosa, mientras que en el de la derecha se aprecia una falta de respuesta por parte del servidor, en este caso el datalogger no registra la marca de dato grabado, con lo cual en la próxima sesión de transmisión lo hará desde

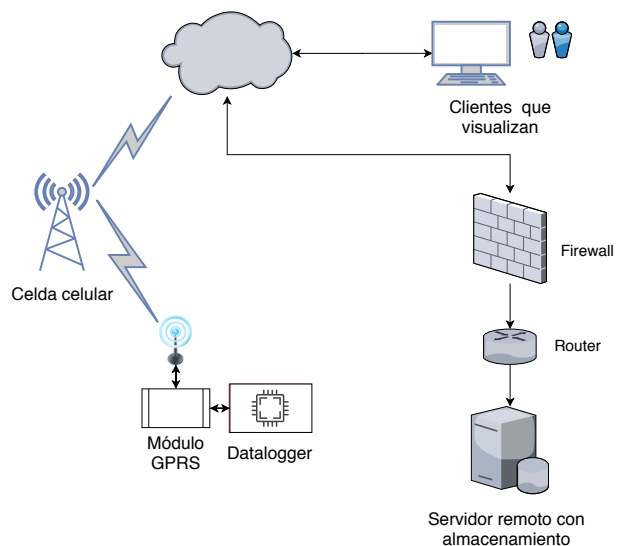


Fig. 10. Diagrama de enlace con GPRS

el dato que no pudo concretar la grabación.

Como se mencionó previamente, la figura 2 muestra el esquema general del sistema. En el bloque nominado como "servidor" se aloja el sistema web que permite realizar las representaciones que se detallarán en los párrafos siguientes.

VII. SISTEMA DE VISUALIZACIÓN Y PERSISTENCIA DE DATOS

Se realizará la explicación de esta sección con una discriminación en *capas*, para su mejor comprensión. La figura 12 muestra el diagrama conceptual de las capas de software que componen el sistema.

Siempre pensando que cada capa es una porción de software, se observa que esta disposición no discrimina en qué plataforma se está ejecutando cada capa. Esto se debe a que el diagrama sugiere interpretar cual es la lógica puesta en juego en todo el proceso de medición, hasta la

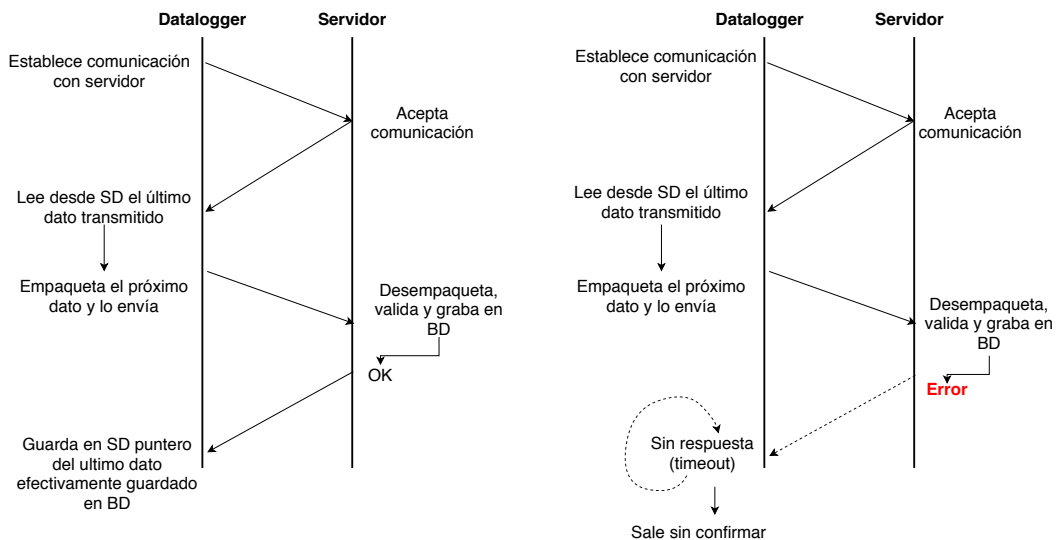


Fig. 11. Sincronización con servidor remoto

visualización de los resultados. Mediante la aplicación de este concepto, se concluye que la entrada al sistema son las variables físicas y la salida o resultado es la visualización de estas variables.

A. Capa 1

Esta capa se refiere al firmware que corre en el microcontrolador del datalogger, que se encarga de manejar el convertor A/D, procesar las variables para obtener promedios, máximos, mínimos y desvío estándar y gestiona la grabación de la SD (con sistema de archivos FAT). Asimismo administra las comunicaciones ya sea por capa física ethernet (vía protocolo TCP/IP) o GPRS. También posee un procesador de comandos utilizado para la configuración y comunicación. Está escrita en lenguaje C.

B. Capa 2

Esta capa, escrita en lenguaje Python [23], es la encargada de la interface entre el datalogger y el servidor de base de datos donde se almacenarán las variables procesadas (Diálogo M2M = diálogo máquina a máquina). Se encarga de gestionar la comunicación desde el lado fijo (servidor), validar los paquetes (datos formateados) que llegan desde el datalogger y guardarlos en la base de datos. Posee un algoritmo de confirmación para evitar la pérdida de datos por falla en la capa física.

C. Capa 3

Es el motor de base de datos MySQL [24] que se encarga de recibir los datos de la capa anterior y efectivizar el almacenamiento en el hardware del servidor.

D. Capa 4

Esta capa constituye el sistema web, y es la encargada de mostrar los datos medidos, ya sea en forma de gráficos temporales (velocidad del viento, temperatura, etc.) o gráficos específicos (rosa de los vientos, distribución estadística de Weibull, etc.). Muestra también de manera simple e intuitiva los últimos datos medidos por el datalogger, y la posición geográfica del mismo (o los mismos, ya que el sistema se ha

diseñado para albergar varios equipos). Posee formularios para la configuración del sistema en general. También es posible elegir los entornos de tiempo de graficación y cálculo de resultados especiales. Está desarrollada en los siguientes lenguajes de programación: PHP [25], Javascript [26], HTML [27]. Para las representaciones gráficas se han utilizado las bibliotecas de acceso libre Highcharts [28], escritas en Javascript.

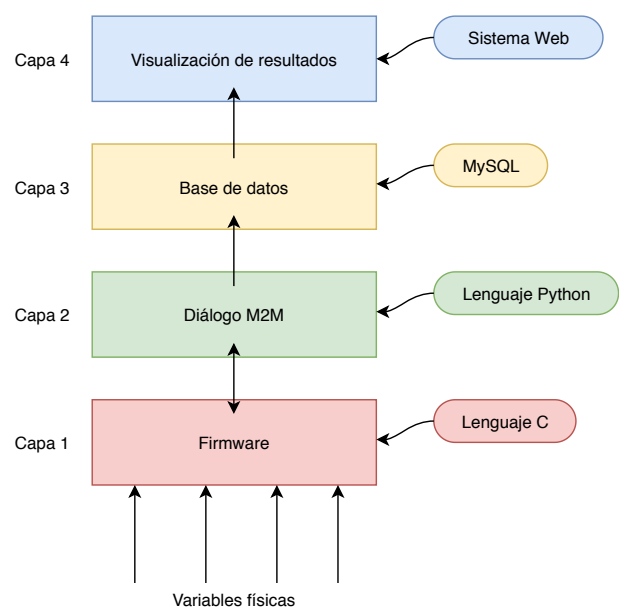


Fig. 12. Diagrama de capas del software del sistema.

VIII. RESULTADOS Y PRESENTACIÓN DE LA INFORMACIÓN

Para presentar la información obtenida de las mediciones que procesa el datalogger, se ha creado un sitio web. El acceso al mismo se logra a través de autenticación con usuario y contraseña.

La intención es que el usuario habituado al uso de aplicaciones en dataloggers comerciales encuentre familiares

OES - Sistema de medición y supervisión

Equipo Seleccionado: Datalogger V2.0 en CEMAR II

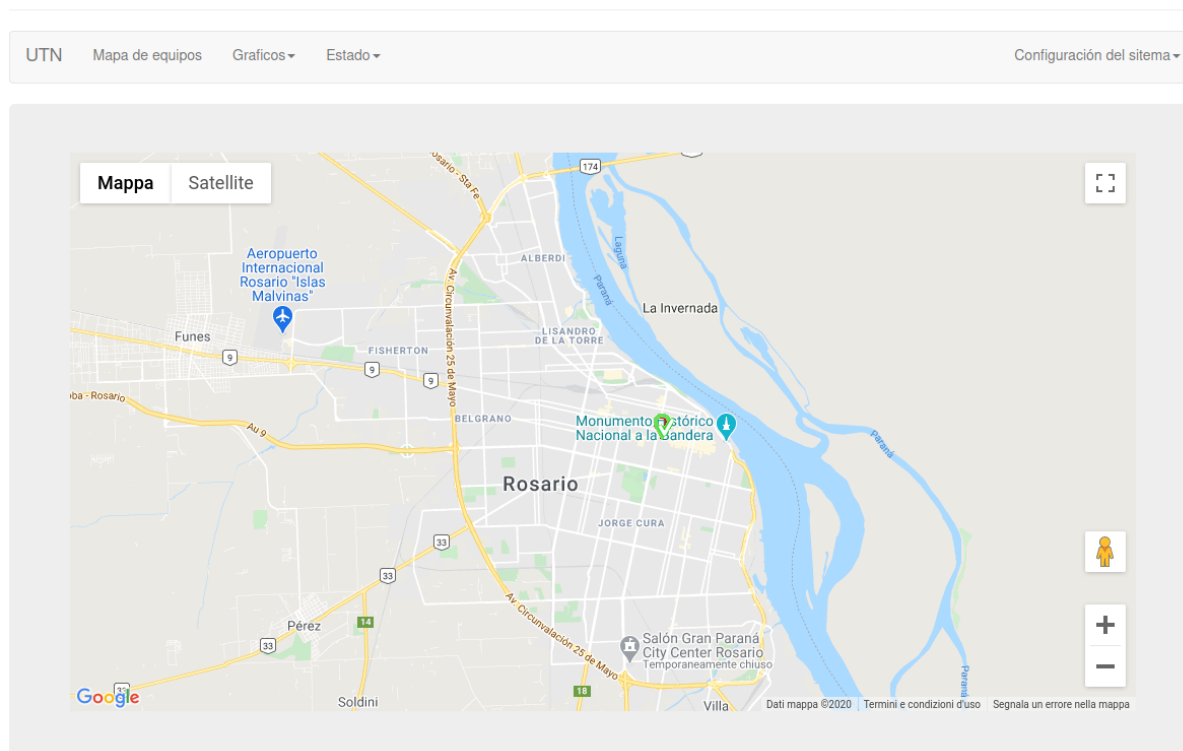


Fig. 13. Página inicial del sistema de visualización

los gráficos e informaciones contenidas y se maneje con solvencia con una mínima adaptación. Dado que el datalogger se encontrará en servicio activo en el sistema, la idea básica es que pueda accederse a la medición online, en tiempo real, puedan observarse los valores que se están midiendo, además de obtener los datos y a partir de ellos trabajar en el reporte que se necesite. También, como eventualmente habrá varios equipos operativos dentro del sistema, se ha establecido una identificación de los mismos de modo inequívoco.

Menú de inicio

Al ingresar al sitio se observa un mapa de ubicación de los diferentes dataloggers (o eventualmente estaciones meteorológicas, u otros equipos) disponibles, cuyas coordenadas geográficas se pueden ingresar. Esta instancia se ha elaborado aprovechando las facilidades de Google Maps, y la idea es que allí puedan observarse las diferentes mediciones para los que se ha pensado el sitio. El proyecto integral contempla la capacidad de brindar operatividad al funcionamiento de un conjunto de estaciones que volcarán las mediciones en este sitio. Se ha trabajado en este diseño para ser empleado en el Proyecto PRIER [29]. La figura 13 muestra la página inicial del sitio web.

Los puntos de medición se señalan mediante un marcador, y cierta información acerca del mismo se anota en una señal tipo burbuja u otra de uso habitual en maps (por ejemplo coordenadas, denominación del sitio u otro que sirva de referencia).

Representaciones gráficas de las salidas del sistema

Velocidades de vientos: Examinando la pestaña de gráficos, lo primero que se observa en este menú es la gráfica de vientos, esto es, la representación de los promedios de velocidades cada 10 minutos (o intervalo seleccionado), de los anemómetros instalados en el sitio. En este caso son dos, y las gráficas son en dos colores, permitiendo mediante la localización del puntero obtener el detalle de las dos velocidades y la hora del correspondiente registro, como se puede apreciar en la figura 14.

Pueden seleccionarse las fechas de inicio y finalización de la gráfica. Además existe la posibilidad de hacer un zoom temporal en el gráfico (mediante el eje inferior), desde una hora hasta un año, o mostrar toda la información disponible.



Fig. 14. Gráfico de series de vientos.

Rosa de vientos: La rosa de vientos de frecuencia se obtiene en la siguiente opción de la pestaña de gráficos. Se

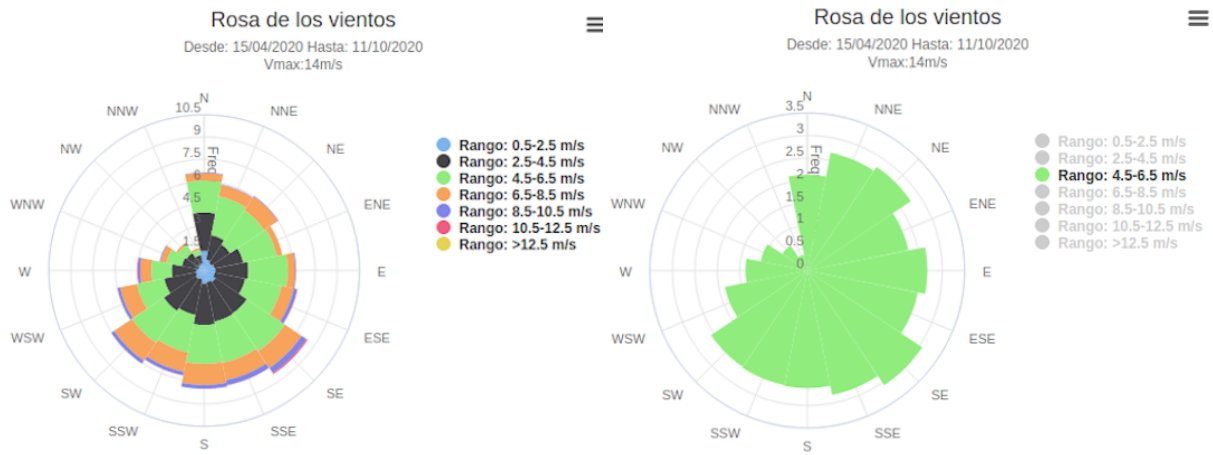


Fig. 15. Gráfico de rosa de vientos.

observa que además de definir el rango temporal se puede asignar un entorno de velocidades, y una Vmin (velocidad mínima) a representar. Este entorno en [m/s] implica la subdivisión del total de mediciones que corresponde a cada sector de la rosa. La Vmin es el valor de la velocidad registrada por encima del cual se realizará el gráfico, es decir, se descartarán las velocidades por debajo de ella. Existe la posibilidad de obtener una rosa de vientos tradicional, mediante el recurso de asignar un entorno que abarque a todas las velocidades medidas (el dato de velocidad máxima Vmax siempre aparece explicitado). Se observa por ejemplo en la figura 15 (derecha), que la ocurrencia total para el sector ESE ha sido del 8,44%. Se han descartado los vientos menores a 0,5 m/s.

Además de esta rosa de vientos estándar, es posible configurar el entorno, y obtener una representación que resulte más útil para la aplicación deseada. Por ejemplo, ajustando el entorno en 2 m/s (figura 15 izquierda), se observa en la pantalla que para los valores de ocurrencia porcentual en cada sector de dirección del viento, el segmento representativo se subdivide a su vez en sectores diferenciados por colores que representan el porcentaje de viento de ese rango que ha sido configurado. En pantalla puede apreciarse (ubicando el cursor) que para el sector predominante, SE, la participación de los vientos de entre 4,5 y 6,5 m/s es del 3.073%

Condiciones meteorológicas: temperatura, presión y humedad: Se muestra en la figura 16, la temperatura promedio cada diez minutos. El cursor permite ver los valores particulares con exactitud.

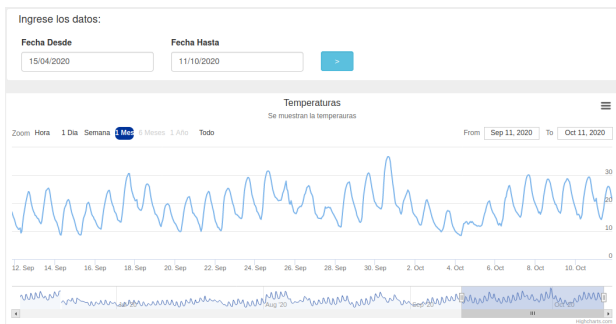


Fig. 16. Gráfico de series de tiempos de temperatura.

Variables físicas en tiempo real: En esta pestaña el sistema permite visualizar las variables en tiempo real, a partir de la última medición efectuada, esto es, el último dato leído y calculado. Se ha elegido desde Highcharts [28] mostrar una representación de rápido impacto visual (speedometer), mostrada en la figura 17. Se trata de una visualización alternativa para las mediciones de la velocidad y la dirección del viento representada según un eje temporal, que se han expuesto arriba, y que tiene la utilidad de ubicar el estado general de la medición de modo inmediato.



Fig. 17. Variables en tiempo real anemómetro 1.

Intensidad de turbulencia: La intensidad de turbulencia, descrita en la ecuación 1, es un dato de importancia, que debe atenderse ya que junto a los vientos extremos afecta dos aspectos claves en el diseño de un parque eólico: el aprovechamiento energético del recurso y la fatiga mecánica de los materiales de los aerogeneradores. Es posible a través del sistema realizar una adecuada estimación (en una dimensión) de estos valores, mediante el cálculo:

$$IT = \frac{\sigma_{10}}{\bar{v}_{10}} \quad (1)$$

donde:

σ_{10} : desvío standard en un período de 10 minutos

\bar{v}_{10} : velocidad promedio en un período de 10 minutos.

Tanto los promedios de velocidades de viento como el

desvío estándar para cada período de 10 minutos han sido calculados, con lo cual se puede construir la gráfica (figura 18) que se muestra en la pestaña gráficos-intensidad de turbulencia. Puede seleccionarse allí el rango temporal que se desea para la representación.

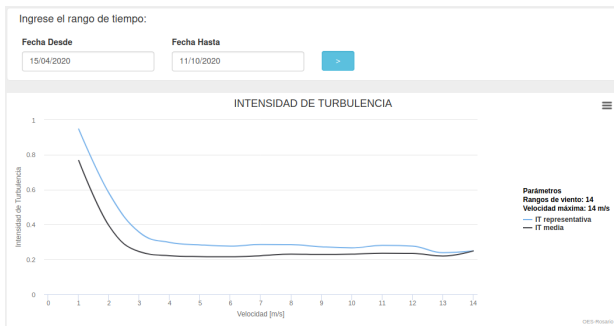


Fig. 18. Intensidad de turbulencia.

Distribución estadística de Weibull: Siendo el viento una variable aleatoria continua, para su estudio deben organizarse los datos medidos. La función de densidad de probabilidad de Weibull II, de valores positivos, asimétrica y biparamétrica, es la que mejor se ajusta para este tipo de estudio según lo visto en [2]. En el sistema OES es posible seleccionar en la pestaña de gráficos esta función, y elegir el rango de fecha para los datos. Se calculan automáticamente los parámetros c [m/s] y k [] de dicha distribución, con los datos de cada anemómetro, como se muestra en la figura 19. De los métodos habitualmente utilizados, se ha seleccionado el de los mínimos cuadrados.



Fig. 19. Gráfica de Weibull.

IX. CONCLUSIONES Y TRABAJOS FUTUROS

La evaluación del recurso eólico de un sitio, como presupuesto para el desarrollo de parques eólicos, se encuentra en el inicio de toda política de inserción de energías renovables. El propósito del presente trabajo es que tienda a facilitar herramientas para lograr un cambio en la matriz energética del país, que se oriente -tal como las políticas nacionales impulsadas actualmente lo proclaman- a una inserción significativa y sostenida de energías renovables en dicha matriz. Existe una frontera imprecisa entre las funciones que debe brindar un datalogger y su sistema asociado de gestión de la información que se obtiene en un sitio de inspección del recurso eólico, y las funciones de análisis profundo de dicha información con la finalidad de evaluar la calidad del sitio de generación eólica. Al

avanzar en el Proyecto se transitó sobre esta frontera, y por momentos no quedaba claro el límite del desarrollo y qué debería reservarse para futuros Proyectos. Se ha buscado que el resultado fuese similar al que se obtiene de otros dataloggers, y poder de alguna manera comparar estos resultados, para evaluar los propios logros en el trabajo realizado. Analizando los objetivos propuestos en el presente trabajo, se observa que muchos de ellos se pudieron alcanzar satisfactoriamente y que el proyecto ha permitido poner en práctica conocimientos adquiridos en varios Proyectos anteriores y exitosos del Grupo OES. Como continuación del trabajo presentado, se ha propuesto profundizar el estudio y la evaluación de las incertidumbres introducidas en la medición del sistema diseñado, tarea indispensable para la validación del equipo, y en la que el Grupo se encuentra trabajando en la actualidad.

REFERENCES

- [1] IEC Standard (2017) IEC 61400-12-1 Wind energy generation systems - Part 12-1: Power performance measurements of electricity producing wind turbines. <https://webstore.iec.ch/publication/26603>
- [2] Salerno, Juan (2017), Tesis de Maestría: Diseño de herramientas propias para evaluar el potencial eólico local - Desarrollo de equipamiento y procesamiento de datos software libre. Defensa 08/2019 ResGate DOI: 10.13140/RG.2.2.17863.73128
- [3] Oliva, R. (2014), Evaluación de incertidumbre en mediciones de potencia eléctrica en registradores automáticos. <http://www.sase.com.ar/2014/congreso-argentino-de-sistemas-embebidos-case-2014/>
- [4] Oliva, R.; Martín, G.; Duzdevich, J.; Zappa, A. (2013) Evaluación de Incertidumbres de medición en Plataforma de Ensayo para pequeños aerogeneradores, AVERMA Vol. 17, pp.06.11-06.20, ISSN 2314-1433 <https://bit.ly/2UTnx8C>
- [5] Mattio, H. T. (2009). Recomendaciones para mediciones de velocidad y dirección de viento con fines de generación eléctrica, y medición de potencia eléctrica generada por aerogeneradores. Buenos Aires: Secretaría de Energía.
- [6] <https://www.nrgsystems.com/products/data-loggers/detail/symphonieplus3-data-logger>
- [7] <https://www.nrgsystems.com/support/product-support/software/symphonie-data-retriever-software>
- [8] <http://wind-think.com/nomad-2-data-logger/>
- [9] <http://forms.vaisala.com/nomad-desktop>
- [10] <https://www.ammonit.com/en/support/9-ammonit-produkte/datalogger/109-data-logger-wind>
- [11] <https://www.campbellsci.com/data-loggers>
- [12] <https://www.campbellsci.com/loggernet-admin>
- [13] <https://core.ac.uk/download/pdf/325984495.pdf>
- [14] <https://www.ul.com/resources/apps/windographer>
- [15] LPC1769 NXP Semic.: <https://www.nxp.com/products/processors-and-microcontrollers/arm-microcontrollers/general-purpose-mcus/lpc1700-cortex-m3/512kb-flash-64kb-sram-ethernet-usb-lqfp100-package:LPC1769FBD100>
- [16] <https://www.embeddedartists.com/products/lpc1769-lpcxpresso/>
- [17] Integrated Developing Environment. Herramienta de desarrollo basada en Eclipse: <https://www.eclipse.org/ide/>
- [18] GNU C Compiler, compilador de código C licencia GPL abierta: <https://developer.arm.com/tools-and-software/open-source-software/developer-tools/gnu-toolchain/gnu-rm/downloads>
- [19] Programador-depurador integrado LPC-Link2: <https://www.nxp.com/design/microcontrollers-developer-resources/lpc-microcontroller-utilities/lpc-link2:OMI3054>
- [20] FreeRTOS Sistema operativo de tiempo real de licencia abierta: <https://www.freertos.org/>
- [21] MCP3208, Microchip: <https://www.microchip.com/wwwproducts/en/MCP3208>
- [22] Oliva, R. (2014) Tesis Maestría: Estacion meteorológica de construcción modular orientada a la prospección eólica en Argentina. ResGate DOI: 10.13140/RG.2.2.24096.33289
- [23] Python (1991) - lenguaje de programación de alto nivel ampliamente utilizado, autor Guido Van Rossum: <https://www.python.org/>
- [24] MySQL Sistema de gestión de base de datos de código abierto. Propiedad de Oracle Corp. <https://www.mysql.com/>

- [25] PHP lenguaje de scripting de lado servidor/cliente, orientado a objetos. Creado por Rasmus Lerdorf en 1994<https://www.php.net/>
- [26] JavaScript Implementación de ECMAScript, primera version 1995. <https://www.ecma-international.org/publications/standards/Ecma-262.htm>
- [27] HTML5 Versión mas reciente de Hypertext Markup Language: <https://html.spec.whatwg.org/multipage/>
- [28] Bibliotecas de graficación HighCharts: <https://www.highcharts.com/>
- [29] PRIER (Proyecto de Generación Distribuida con Energías Renovables) un Proyecto piloto de generación eléctrica con paneles fotovoltaicos promovido por la UTN, la Cooperativa de Provisión de Obras y Servicios Públicos de Armstrong y el INTI con el apoyo de la Agencia Nacional de Promoción Científica y Tecnológica, CAMMESA y la Subsecretaría de Energía de la Nación. <http://www.celar.com.ar/index.php/prier>