

# Jornadas de Automática

## Trazador inteligente en tiempo real para productos perecederos

Garrido, J.\*, Jiménez, M., Toledo, A., Soto, F., Torres, R.

*Dpto. de Automática, Ingeniería Eléctrica y Tecnología Electrónica, ETSII, Universidad Politécnica de Cartagena, 30202, Cartagena, España.*

**To cite this article:** Garrido, J., Jiménez, M., Toledo, A., Soto, F., Torres, R. 2024. Real-time intelligent tracker for perishable products. *Jornadas de Automática*, 45. <https://doi.org/10.17979/ja-cea.2024.45.10860>

### Resumen

Durante la fase de poscosecha y antes del consumo, los productos perecederos pueden presentar pérdidas de calidad debido a malas condiciones de conservación. La monitorización de las condiciones ambientales clave durante este periodo puede ser utilizada para predecir estas pérdidas. En este proyecto, se presenta el diseño, construcción y ensayos de un datalogger autónomo, portátil y compacto para la monitorización inalámbrica en tiempo real de parámetros ambientales durante la cadena de suministro. El trazador incorpora en un único dispositivo sensores para medir temperatura, humedad relativa, concentración de CO<sub>2</sub>, luminosidad y vibraciones, así como capacidad de comunicación inalámbrica para la transmisión de los datos, lo cual simplifica las soluciones existentes que emplean más de un dispositivo y presentan costes prohibitivos. Se han empleado estrategias para optimizar el consumo de energía, logrando un mes de autonomía para cubrir periodos enteros de transporte de alimentos. El datalogger utiliza NB-IoT y se apoya en otras tecnologías inalámbricas en caso de no estar disponible para enviar los datos a una plataforma en la nube. Se han realizado ensayos de autonomía, comparación con otros dataloggers comerciales y ensayos de campo satisfactorios para validar su funcionamiento y transmisión de datos en tiempo real durante un transporte refrigerado.

*Palabras clave:* Poscosecha, Monitorización en tiempo real, Datalogger, Productos perecederos, Vida útil.

### Real-time intelligent tracker for perishable products

#### Abstract

During postharvest and before consumption, perishable products may show quality losses due to poor storage conditions. The monitoring of key environmental conditions during this period can be used to predict these losses. In this project, the design, construction and testing of an autonomous, portable and compact datalogger for real-time wireless monitoring of environmental parameters during the supply chain is presented. The tracker incorporates in a single device sensors to measure temperature, relative humidity, CO<sub>2</sub> concentration, luminosity and vibrations, as well as wireless communication capability for data transmission, which simplifies existing solutions that employ more than one device and are cost prohibitive. Strategies have been employed to optimize energy consumption, achieving one month of autonomy to cover entire periods of food transport. The datalogger uses NB-IoT and relies on other wireless technologies in case it is not available to send data to a cloud platform. Autonomy tests, comparison with other commercial dataloggers and satisfactory field tests have been performed to validate its performance and real-time data transmission during a refrigerated transport.

*Keywords:* Post-harvest, Real-time monitoring, Datalogger, Perishable commodities, Shelf-life.

### 1. Introducción

Según la FAO (Organización de las Naciones Unidas para la Alimentación y la Agricultura), a lo largo del año se desperdicia aproximadamente un tercio de los alimentos.

Estas pérdidas pueden ocurrir durante el almacenamiento, el transporte o el procesamiento, y es esencial preservar la frescura de estos productos y reducir su desperdicio durante la cadena de suministro (FAO, 2019, 2022; FAO et al., 2023).

Para ello, es necesario mantener unas condiciones ambientales adecuadas, entre las que destacan la temperatura, la humedad relativa, la concentración de gases como el CO<sub>2</sub>, la luminosidad y las vibraciones, y también es primordial monitorizarlas en tiempo real para tener información y retroalimentación continua del estado de estas variables, lo cual permitirá llevar a cabo una gestión eficiente de la cadena de suministro y la preservación de la calidad de los productos (De Venuto & Mezzina, 2018; Depetris et al., 2014).

Actualmente, no se han encontrado sistemas comerciales de bajo coste que midan esos parámetros y que, además, los transmitan en tiempo real (Swift Sensors, s. f.; Tempmate, s. f.). Teniendo en cuenta la necesidad de monitorizar las condiciones ambientales de los productos durante la etapa de poscosecha, este proyecto busca realizar el diseño y fabricación de un sistema de instrumentación portátil, tamaño reducido, bajo coste y bajo consumo para uso en transportes refrigerados. Además, este sistema tendrá capacidad de transmisión de datos en tiempo real. El dispositivo es un trazador o *datalogger* capaz de medir las variables mencionadas anteriormente y que presenta alta autonomía gracias a estrategias de optimización de consumo de hardware y software, entre las que se encuentran el uso de redes móviles de comunicación inalámbrica de baja potencia (LPWAN).

Adicionalmente, se ha realizado la programación de una plataforma en la nube para registrar y visualizar los datos enviados de manera inalámbrica en tiempo real, y se han realizado ensayos con el prototipo funcional para validar su funcionamiento, analizando los resultados posteriormente.

El objetivo principal del proyecto es demostrar que el sistema diseñado basado en tecnología IoT (Internet of Things) es capaz de monitorizar en tiempo real los parámetros más importantes de los productos perecederos durante la etapa de poscosecha.

## 2. Materiales y métodos

### 2.1. Arquitectura del sistema

La arquitectura de la solución completa diseñada se puede observar en el diagrama de bloques de la Figura 1. El trazador o nodo se monta en la zona donde se van a medir los parámetros ambientales, como la cámara climática de un camión frigorífico, y mide dichos parámetros a través de los sensores que incorpora.

El cerebro del dispositivo es un ESP32 (Espressif Systems, 2023), un System-On-Chip (SoC) flexible y económico muy utilizado en desarrollos electrónicos. Los sensores se comunican con el SoC, el cual almacena una copia de los datos en su memoria flash y los envía inalámbricamente al operador INCE (INCE, 2023) a través del módulo de comunicación SIM7000G (SIMCom, 2018), que incorpora una tarjeta SIM, mediante comunicación NB-IoT, una red LPWAN de baja potencia. La comunicación entre el SoC y el SIM7000G se realiza mediante comandos AT.

Entre las distintas alternativas de tecnologías de comunicación en Europa, se ha seleccionado NB-IoT porque

minimiza el consumo y se ha empleado UDP como protocolo por su ligereza, operando sobre el protocolo IP.

Posteriormente, se accede a los datos del operador desde un servidor VPS a través de una interfaz programada en Node-RED (OpenJS Foundation, 2023), el cual permite utilizar su REST API, y los datos se guardan en una base de datos creada con InfluxDB (InfluxData, 2023) y Grafana (Grafana Labs, 2023), a los cuales se puede acceder y visualizar en tiempo real desde cualquier dispositivo a través de Internet.

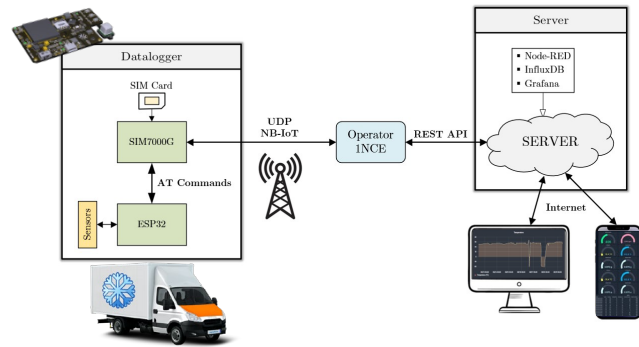


Figura 1: Diagrama de bloques de la arquitectura del sistema.

### 2.2. Hardware

El sistema de instrumentación diseñado consiste en un datalogger que incorpora el SoC para gestionar todo el sistema, el cual es muy versátil y presenta un bajo coste y un alto rendimiento, así como bajo consumo de energía, permitiendo operar en diferentes modos de bajo consumo para alargar la vida de la batería. Además, incorpora una memoria flash de 8 MB para almacenar el programa y los datos medidos por los sensores.

Junto al SoC, se encuentra el módulo de comunicación SIM7000G, que permite la transmisión inalámbrica de los datos a través de las redes LPWAN NB-IoT y LTE-M, además de las tecnologías GSM y EGPRS. También presenta bajo consumo, alto rendimiento, es económico y tiene una alta flexibilidad, permitiendo el uso de protocolos como TCP y UDP, además de ofrecer un modo de ahorro de energía para extender aún más la vida de la batería.

El trazador es capaz de monitorizar temperatura, humedad relativa, luminosidad, vibraciones y concentración de CO<sub>2</sub>, mediante el uso de sensores que presentan un bajo coste y bajo consumo a la vez que ofrecen una buena precisión en sus mediciones. La comunicación de los sensores con el SoC se realiza a través de I<sup>2</sup>C y SPI.

En la Figura 2, se puede observar el diagrama de bloques que detalla el funcionamiento del circuito del datalogger, el cual consiste en una etapa de alimentación, comunicación con el USB, los sensores, el SoC y la parte de comunicación inalámbrica. Del mismo modo, se puede ver el diseño de la PCB en la Figura 3, donde los componentes se han colocado de la forma más óptima posible para obtener una placa de un tamaño muy reducido. Además, se ha realizado la impresión 3D de una carcasa en PLA (ácido poliláctico) mediante la tecnología de fabricación aditiva FDM (modelado por deposición fundida) para protegerla de las condiciones externas.

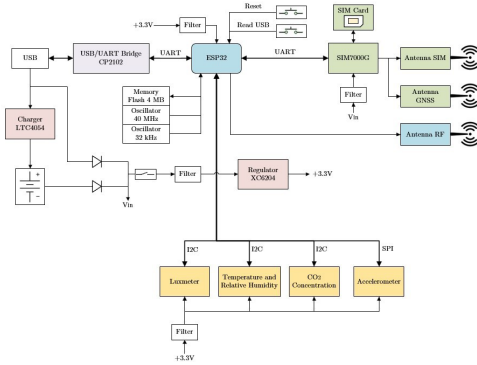


Figura 2: Diagrama de bloques del funcionamiento del circuito del datalogger.

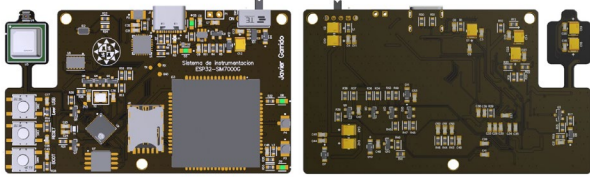


Figura 3: Vista 3D de la PCB del datalogger, 79 × 45 mm.

2.3. Software, nube y base de datos

La programación del firmware del SoC del datalogger se ha realizado en C++ utilizando el entorno Visual Studio Code con la extensión PlatformIO. En el flujograma de la Figura 4 se resume su funcionamiento.

Los datos se miden y se registran en la memoria flash cada 15 minutos y se envían inalámbricamente cada menos frecuencia en función del nivel de batería, el número de vibraciones y la diferencia con respecto a la medida anterior de forma que, si esta es muy pequeña, no se envían los datos, para alargar más la vida de la batería.

El SoC entra en modo de sueño profundo o Deep Sleep mientras no tiene actividad para ahorrar energía. Los datos almacenados en la memoria flash son una copia de seguridad para recuperar los datos por USB en caso de fallo en las comunicaciones o cualquier otro incidente.

En primer lugar, se comprueba la conexión del USB con el propósito de recuperar los datos almacenados en la memoria flash. En caso de estar conectado, se procede a la lectura y ejecución de una serie de comandos para recuperar o eliminar los datos presentes en la flash, a través de la Shell programada del sistema.

En su funcionamiento normal, sin la conexión del USB, durante el modo Deep Sleep, se ha programado una interrupción externa para registrar las vibraciones que superen un umbral de 2 g, dado que estas pueden ocurrir en cualquier momento, volviendo al modo de sueño profundo inmediatamente después. De este modo, se registran la aceleración máxima, media y el número de vibraciones durante los 15 minutos.

Cuando el SoC se despierta, primero, se inicializa el acelerómetro y se comprueba si se ha despertado por la interrupción externa. En caso contrario, tras los 15 minutos, se alimenta el resto de los sensores a través de un MOSFET, se inicializan, se registran sus mediciones y se vuelven a apagar. Después, se lee el voltaje de la batería y se guardan todos los datos en formato CSV, junto a un contador de paquete.

Cuando se cumplen las condiciones mencionadas anteriormente, los datos se envían inalámbricamente a través de NB-IoT por defecto, en formato CayenneLPP (Cayenne Low Power Payload) para optimizar el tamaño del mensaje, sincronizando el timestamp (fecha y hora), y entra de nuevo en Deep Sleep, repitiéndose el ciclo.

Cuando el datalogger se enciende por primera vez, se sincroniza el timestamp y se calibra el sensor de CO<sub>2</sub> con el ambiente. Además, cuando se envían los datos de manera inalámbrica, el dispositivo podrá conectarse por LTE-M, GSM o EGPRS en caso de que NB-IoT no esté disponible.

Los niveles de batería que determinan la frecuencia de envío se pueden ver en la Tabla 1. Asimismo, si hay un exceso de vibraciones, definido como que el sistema se ha despertado 20 veces o más por vibración la última vez, la interrupción del acelerómetro que mide las vibraciones no se habilita hasta que han transcurrido 30 minutos desde que ocurrió el evento. También, los umbrales que se tienen que superar con respecto a la medida anterior para que los datos se envíen inalámbricamente se muestran en la Tabla 2. Es suficiente con que uno de estos umbrales sea superado para el envío de los datos. En cualquier caso, los datos siempre se envían cuando transcurren 2 horas aunque estos umbrales no se hayan superado, para mantener un seguimiento de la monitorización de los parámetros.

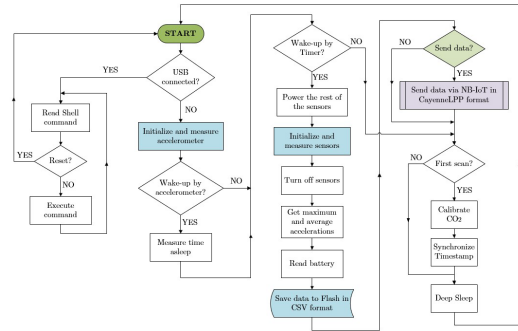


Figura 4: Flujograma del funcionamiento del firmware del datalogger.

Tabla 1: Nivel de voltaje de la batería y frecuencias de envío.

Voltaje batería	Frecuencia de envío
Batería cargada (> 3,9 V)	45 minutos
Batería < 3,9 V	1 hora
Batería < 3,8 V	1 hora 30 minutos
Batería descargada (< 3,7 V)	2 horas y deshabilitar interrupción acelerómetro

Tabla 2: Umbrales que se deben superar con respecto a la medida anterior.

Parámetro	Umbral
Luminosidad	50 lux
Concentración de CO <sub>2</sub>	200 ppm
Temperatura	1 °C
Humedad relativa	∞
Aceleración	0,5 g

Las frecuencias de muestreo y de envío se rigen por un equilibrio entre el consumo de energía y la dinámica del comportamiento de los alimentos perecederos. Es decir, un mayor muestreo no sería útil porque no tiene influencia en el alimento.

Por su parte, el valor para cada medida de aceleración ( $a_i$ ) en fuerzas  $g$  se obtiene como la aceleración total resultante de los 3 ejes relativo a la gravedad, como se muestra en (1). Asimismo, si se ha medido un total de  $n_a$  aceleraciones durante los 15 minutos, las aceleraciones media ( $a_{med}$ ) y máxima ( $a_{m\acute{a}x}$ ) se obtienen como se muestra en (2), donde  $\max(a_1, \dots, a_{n_a})$  devuelve el máximo valor del conjunto de valores  $a_i$  introducidos.

$$a_i = \sqrt{a_x^2 + a_y^2 + a_z^2} - 1 \quad (1)$$

$$a_{med} = \frac{1}{n_a} \sum_{i=1}^{n_a} a_i \quad a_{m\acute{a}x} = \max(a_1, \dots, a_{n_a}) \quad (2)$$

Para la lectura de temperatura, humedad y CO<sub>2</sub>, se toman 5 medidas  $x_i$  cada 500 ms para cada sensor y se calcula una media ponderada ( $x_{med}^{pond}$ ), que consiste en ordenar los valores de más bajo a más alto, eliminar el máximo y el mínimo y hacer la media con los datos restantes, como se muestra en (3), donde  $n_x = 5$  y  $\max(x_1, \dots, x_{n_x})$ ,  $\min(x_1, \dots, x_{n_x})$  devuelven el máximo y el mínimo valor del conjunto de valores  $x_i$  introducidos, respectivamente.

$$x_{med}^{pond} = \frac{1}{n_x - 2} \sum_{i=1}^{n_x-2} x_i \quad (3)$$

donde  $x_i \neq \max(x_1, \dots, x_{n_x}), \min(x_1, \dots, x_{n_x})$

Para utilizar la API del operador y almacenar y visualizar los datos, se han utilizado algunas de las herramientas de código abierto más empleadas en desarrollos IoT: (i) una base de datos de series temporales (TSDB) con InfluxDB, diseñada especialmente para tratar grandes series de datos adquiridas en instantes determinados, con la posibilidad de etiquetado, (ii) Grafana como plataforma de visualización, que puede realizar consultas de grandes volúmenes de datos de TSDB y mostrarlos en paneles avanzados de forma altamente eficiente, y (iii) Node-RED como herramienta rápida y visual para la integración sencilla de servicios (consultas REST API, automatización del etiquetado e inserción en la TSDB, etc.) (Radia et al., 2023).

Node-RED se encarga de guardar los datos medidos y la geolocalización en el servidor en una base de datos creada con InfluxDB. Además, los datos se pueden visualizar desde la plataforma Grafana en tiempo real. El funcionamiento se muestra en el flujograma de la Figura 5. Asimismo, el servidor envía una alarma al móvil a través de Telegram cuando se supera un determinado umbral en los parámetros medidos, tal y como se muestra en la Tabla 3.

El operador proporciona una actualización de la geolocalización cada 24 horas por lo que, si se requiere una mayor monitorización, existe la posibilidad de conectar una antena GPS externa al dispositivo y enviar los datos de localización también al servidor, lo cual ya está contemplado en el código, aunque esto implicaría un mayor consumo. Asimismo, el datalogger registra los códigos de las antenas a las que se ha conectado cada vez en su memoria flash y las envía al servidor, de modo que se puede obtener la localización utilizando cualquier base de datos de antenas de telefonía.

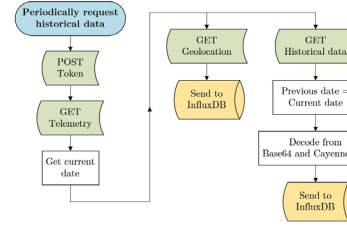


Figura 5: Flujograma del funcionamiento de Node-RED.

Tabla 3: Umbrales que se deben superar para enviar una alarma.

Parámetro	Umbral máximo	Umbral mínimo
Luminosidad	10.000 lux	-
Concentración CO <sub>2</sub>	1.000 ppm	100 ppm
Temperatura	30 °C	-10 °C
Humedad relativa	80%	10%
Aceleración	0,5 g	-

#### 2.4. Ensayo de autonomía

Debido a la importancia de alcanzar la máxima autonomía posible con el sistema, se ha llevado a cabo un ensayo para determinar el consumo del datalogger durante su funcionamiento normal. Para ello, se ha empleado la tarjeta de adquisición de datos USB-6008 de National Instruments (National Instruments, 2023), con la que se ha medido el voltaje ( $V_S$ ) a través de una resistencia shunt de 220 mΩ ( $R_S$ ) colocada en serie con la batería. La tarjeta se ha configurado en modo diferencial para medir una entrada analógica en un rango de  $\pm 1 V$  con una resolución de 12 bits, lo cual equivale a una resolución de 0,488 mV. La tarjeta utiliza una frecuencia de muestreo de 1 kHz, obteniendo medias cada 250 ms. La expresión que relaciona la caída de tensión medida por la tarjeta con la corriente consumida por el sistema ( $I_{BAT}$ ) es la que se muestra en (4), aplicando la Ley de Ohm.

$$I_{BAT} (mA) = \frac{V_S}{R_S} = \frac{V_S (mV)}{220 m\Omega} \times 10^{-3} \quad (4)$$

#### 2.5. Pruebas preliminares

Como estudio preliminar, se ha comprobado la correcta operación del datalogger monitorizando los parámetros ambientales de un frigorífico y un congelador durante dos semanas, para unas temperaturas de referencia de 6 °C y -19 °C, respectivamente, colocándolo en una zona intermedia y utilizando una batería de 3000 mAh.

#### 2.6. Ensayo comparativo con otros dataloggers

Para validar el correcto funcionamiento del sistema y de los datos obtenidos, se han realizado ensayos comparativos con dos dataloggers comerciales, que son los dispositivos Parkside PKDL A1 (Parkside, 2022) y Tinytag Ultra 2 modelo TGU-4500 (Gemini Dataloggers, 2019), ambos de temperatura y humedad, que son las variables más críticas para la calidad de los alimentos. Para ello, se introducen los trazadores en un frigorífico durante 24 horas.

El objetivo de este ensayo es verificar que las tendencias de todos los dataloggers son las mismas. La similitud entre las tendencias se ha cuantificado obteniendo los errores absolutos y relativos máximo, mínimo y medio, así como sus medianas y desviaciones típicas. También se ha calculado la raíz del error cuadrático medio (RMSE) como indicador.

### 2.7. Ensayo de campo

Se ha llevado a cabo un ensayo de campo con el sistema completo para verificar que funciona correctamente en cuanto a transmisión inalámbrica de los datos al servidor y autonomía, además de realizar correctamente las mediciones de los parámetros ambientales.

Para ello, el trazador ha participado en la etapa de poscosecha de transporte y distribución de los productos instalándolo en el interior de la cámara climática de un camión frigorífico que recorre Europa, estando sujeto a sus condiciones reales de temperatura y humedad durante dos semanas. Se ha utilizado una batería de 3000 mAh.

El transporte de este ensayo se realizó con la empresa ESP Solutions (ESP Solutions, 2024). Actualmente, se están haciendo más ensayos con más destinos para comprobar el correcto funcionamiento y la robustez del sistema.

## 3. Resultados y discusión

### 3.1. Nube y alarmas

La interfaz desarrollada en la nube para móvil y ordenador se muestra en la Figura 6, que muestra los resultados de los parámetros ambientales y la geolocalización durante dos semanas. Muestra los valores de los parámetros, gráficas de su evolución y un mapa con la localización. Se trata de interfaces sencillas e intuitivas que muestran toda la información necesaria. También se observa una captura de pantalla de las alarmas recibidas a través de Telegram en el móvil. De esta forma, se verifica el correcto funcionamiento del envío de los datos a la nube junto al envío de alarmas.



Figura 6: Interfaz en la nube para móvil y ordenador, junto al envío de alarmas al móvil.

### 3.2. Autonomía

La Figura 7 muestra la evolución del consumo de corriente del datalogger. Si ampliamos las gráficas, se puede ver que hay tres tipos de consumo, tal y como se observa en la Figura 8. La primera gráfica, correspondiente a las secciones más anchas, se corresponde con periodos de actividad donde el sistema mide los sensores y envía los datos inalámbricamente. La gráfica central, de tamaño medio, se refiere únicamente a la medida de sensores, sin comunicación inalámbrica. Por último, los picos estrechos como el que se muestra en la tercera gráfica se corresponden a los eventos del acelerómetro.

Para el cálculo de la corriente media  $I_{med}$  en las distintas secciones, se ha utilizado (5), donde  $T$  es el periodo de la señal,  $i(t)$  es la corriente instantánea,  $i_k$  son los valores instantáneos en tiempo discreto, tras aplicar el muestreo con periodo de muestreo  $T_s$  y  $N = T/T_s$  es el número de

muestras durante ese periodo. Las corrientes medias junto a otros valores representativos del consumo se pueden ver en la Tabla 4. Se puede determinar la autonomía del sistema  $t_A$  utilizando (6), para una batería de  $Q_{BAT} = 6000 \text{ mAh}$  y un consumo medio de  $\bar{I} = 7,91 \text{ mA}$ , obteniendo una autonomía de, aproximadamente, un mes. Además, con (7), se puede calcular el valor del consumo de potencia ( $P$ ) y de energía ( $E$ ) durante un mes, para una tensión de batería  $V_{BAT} = 3,7 \text{ V}$ .

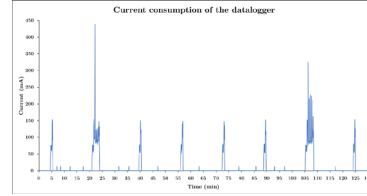


Figura 7: Evolución del consumo de corriente del datalogger.

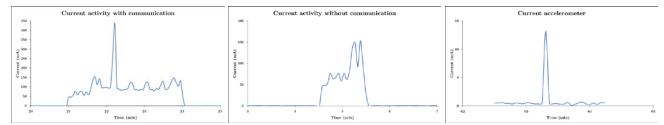


Figura 8: Consumo de corriente del datalogger en las distintas fases.

$$I_{med} = \frac{1}{T} \int_0^T i(t) dt \approx \frac{1}{T} \sum_{k=1}^N i_k T_s = \frac{1}{N} \sum_{k=1}^N i_k \quad (5)$$

$$t_A = \frac{Q_{BAT}}{\bar{I}} = 31,61 \text{ días} \quad (6)$$

$$P = V_{BAT} \cdot \bar{I} \quad E = P \cdot t = P \cdot 1 \text{ mes} \quad (7)$$

Tabla 4: Valores de consumo del datalogger.

Tipo de consumo	Valor
Corriente media	7,91 mA
Corriente máxima	567,25 mA
Corriente media Deep Sleep	379,87 $\mu$ A
Corriente media actividad con comunicación	113,42 mA
Corriente media actividad sin comunicación	77,16 mA
Corriente media evento acelerómetro	13,18 mA
Consumo de potencia	29,26 mW
Consumo de energía en un mes	21,07 Wh

### 3.3. Resultados de las pruebas preliminares

Durante los ensayos preliminares, los datos se almacenaron en la nube correctamente, tal y como se muestra en la Figura 9. También se observa que los datos medidos de temperatura se corresponden con las temperaturas de referencia de 6 °C y -19 °C. Los resultados presentan valores razonables, y se puede ver que la batería proporciona suficiente autonomía al dispositivo durante el ensayo.

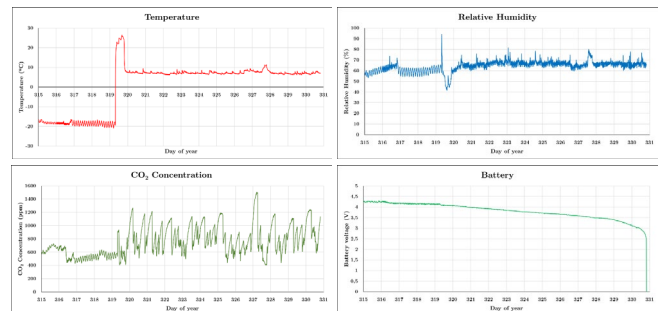


Figura 9: Resultados durante las pruebas preliminares.

### 3.4. Validación mediante comparación con otros dataloggers

Los resultados del ensayo comparativo con los otros dos dataloggers comerciales se muestran en la Figura 10. Con esto, se verifica que las tendencias de todos los dataloggers son las mismas, que era el objetivo principal de este ensayo.

Tras el cálculo de los errores, destaca un RMSE medio de 0,68 °C para la temperatura y de 4,5% para la humedad relativa, así como una mediana de error absoluto de 0,54 °C de temperatura y 3,5% de humedad, lo cual son valores muy bajos.

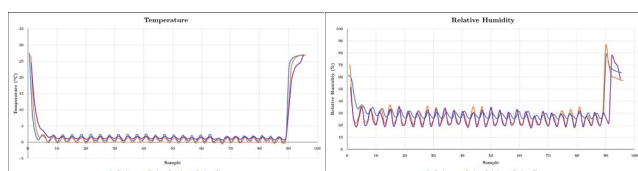


Figura 10: Resultados comparativos con los dataloggers comerciales.

### 3.5. Resultados del ensayo real de transporte

El sistema completo fue probado en condiciones reales dentro de un camión frigorífico en movimiento. En la Figura 11, se puede observar el montaje del datalogger en el camión junto a los resultados de los parámetros enviados al servidor en tiempo real.

En caso de no haber tenido disponibilidad de conexión inalámbrica, el sistema ha guardado las medidas en su memoria flash para enviarlas al servidor en cuanto ha recuperado la conexión. Además, el trazador ha tenido autonomía suficiente durante todo el ensayo.



Figura 11: Resultados del ensayo de campo.

## 4. Conclusiones

Reducir las pérdidas de productos perecederos en la cadena de suministro es una necesidad urgente. Para evitar el desperdicio de estos productos y mantener la máxima calidad durante la poscosecha, es esencial mantener unas condiciones ambientales adecuadas, lo cual extenderá su vida útil. Una correcta monitorización de parámetros ambientales que influyen en la calidad de los alimentos es fundamental para identificar posibles deficiencias y reducir el deterioro durante el transporte y almacenamiento.

Ante la falta de dispositivos asequibles que permitan la transmisión de estos datos en tiempo real, el datalogger compacto que se ha diseñado y probado en los ensayos es una solución innovadora y funcional que transmite los datos a un servidor mediante tecnología IoT con redes LPWAN de baja potencia y que presenta una alta autonomía gracias a las estrategias de optimización de consumo hardware y software empleadas. Además, es muy fácil de utilizar y se pueden visualizar los datos en cualquier momento desde cualquier dispositivo en la nube, incluyendo también el envío de

alarmas para mejorar la monitorización y control de calidad. La arquitectura de un solo dispositivo portátil que integra los sensores y la comunicación inalámbrica lo convierte en un sistema más rápido y fácil de usar en comparación con otras soluciones existentes que emplean varios dispositivos para lograrlo.

Este dispositivo se integra en un proyecto cuyo objetivo es predecir la vida útil de los productos perecederos, integrando un modelo predictivo que utilice los datos de los sensores para predecir su calidad y ajustar su vida comercial cuando han sufrido perturbaciones en su conservación.

Este desarrollo forma parte del Programa AGROALNEXT que ha sido financiado por MICIU con fondos NextGenerationEU (PRTR-C17.I1), por la Fundación Séneca con fondos de la Comunidad Autónoma de la Región de Murcia (CARM) (<https://fseneca.es/agroalnext/>) y por el programa “Prueba de concepto” con código 22267PDC-23.

## Referencias

- INCE. (2023). *INCE, IoT software and connectivity provider*. <https://ince.com/es-es/>
- De Venuto, D., & Mezzina, G. (2018). Spatio-Temporal Optimization of Perishable Goods' Shelf Life by a Pro-Active WSN-Based Architecture. *Sensors* 2018, Vol. 18, Page 2126, 18(7), 2126. <https://doi.org/10.3390/S18072126>
- Depetris, M. C., González, S., & Rodríguez, M. de los Á. (2014). *Recomendaciones para el Almacenamiento de Alimentos Perecederos*.
- ESP Solutions. (2024). *International Transport and Logistics Company*. <https://espsolutions.eu/>
- Espressif Systems. (2023). *ESP32 Series Datasheet*. [https://www.espressif.com/sites/default/files/documentation/esp32\\_atasheet\\_en.pdf](https://www.espressif.com/sites/default/files/documentation/esp32_atasheet_en.pdf)
- FAO. (2019). *The state of food and agriculture. 2019. Moving forward on food loss and waste reduction*.
- FAO. (2022). The State of Food and Agriculture 2022. Leveraging automation in agriculture for transforming agrifood systems. En *The State of Food and Agriculture 2022*. FAO. <https://doi.org/10.4060/cb9479en>
- FAO, IFAD, UNICEF, WFP, & WHO. (2023). The State of Food Security and Nutrition in the World 2023. Urbanization, agrifood systems transformation and healthy diets across the rural-urban continuum. En *The State of Food Security and Nutrition in the World 2023*. <https://doi.org/10.4060/cc3017en>
- Gemini Dataloggers. (2019). *Tinytag TGU-4500, Temperature & humidity datalogger*. <https://www.gemindataloggers.com/dataloggers/tinytag-ultra-2/tgu-4500>
- Grafana Labs. (2023). *Grafana: The open observability platform*. <https://grafana.com/>
- InfluxData. (2023). *InfluxDB Times Series Data Platform*. <https://www.influxdata.com/>
- National Instruments. (2023). *USB-6008*. <https://www.ni.com/es-es/support/model.usb-6008.html>
- OpenJS Foundation. (2023). *Node-RED*. <https://nodered.org/>
- Parkside. (2022). *Datalogger PKDL A1*. <https://www.lidl.de/p/parkside-klimadatenlogger-pkdl-a1-mess-aufzeichnungsintervall-einstellbar/p100361442>
- Radia, M. A. A., Nimr, M. K. E., & Atlam, A. S. (2023). IoT-based wireless data acquisition and control system for photovoltaic module performance analysis. *e-Prime - Advances in Electrical Engineering, Electronics and Energy*, 6, 100348. <https://doi.org/10.1016/J.PRIME.2023.100348>
- SIMCom. (2018). *SIM7000X Datasheet*. [https://simcom.ee/documents/SIM7000E/SIM7000%20Hardware%20Design\\_V1.05.pdf](https://simcom.ee/documents/SIM7000E/SIM7000%20Hardware%20Design_V1.05.pdf)
- Swift Sensors. (s. f.). *Wireless Sensors*. Recuperado 4 de noviembre de 2023, de <https://www.swiftsensors.com/wireless-sensors/>
- Tempmate. (s. f.). *tempmate.®-GS2, Real-time Supply Chain Data Logger*. Recuperado 4 de noviembre de 2023, de <https://www.tempmate.com/solutions/tempmate-gs2/>