

AGROSENSE: SOFTWARE PARA MONITOREO Y ALMACENAMIENTO DE DATOS DEL SUELO

Michel Estrada Castro ¹, Elio Manuel Castañeda González ², Jorge Esteban Santos Tournal ³

¹Etecsa Santiago, Carnicería S/N % Enramadas y Aguilera, Santiago de Cuba, Cuba, ²Cubatur, Avenida Victoriano Garzón S/N % 3ra y 4ta, ³Departamento de Telecomunicaciones, FITIB, Universidad de Oriente, Avenida de Las Américas S/N esquina Casero, Santiago de Cuba, Cuba

¹e-mail: miche.estrada.castro@gmail.com

²e-mail: elio@cabaturstgo.tur.cu ³e-mail: jsantos@uo.edu.cu

RESUMEN

El presente trabajo describe al software AgroSense, de aplicación en la Agricultura de Precisión. Permite almacenar en una base de datos los valores de humedad y temperatura del suelo. Para ello el *software* se comunica con una red de *hardware* de medición localizados en diferentes áreas de cultivo, y almacena la información en una base de datos. La comunicación se realiza por medio del protocolo TCP/IP, bajo la concepción de Internet de las Cosas. Para su implementación se utiliza Qt y base de datos SQLite. El *hardware* se diseña por medio de la plataforma Arduino. El entorno de desarrollo Qt incorpora las herramientas necesarias para la comunicación TCP/IP y la conexión con la base de datos sin necesidad de instalar otras herramientas. El funcionamiento del *software* se realiza por medio de la conexión con los dispositivos de medición implementados con Arduino. Se logra un *software* servidor que logra establecer conexión con dos dispositivos Arduino que actúan como clientes. Los resultados, comprobados a nivel de laboratorio, muestran que no existe colapso del sistema, los datos se almacenan de forma correcta en la base de datos, y que la interfaz gráfica es amigable e intuitiva al usuario. Su implementación permite caracterizar los suelos lo que permite tomar decisiones relacionadas con el riego y la plantación de semillas.

PALABRAS CLAVES: Agricultura de Precisión, TCP/IP, IoT, Qt, Arduino.

AGROSENSE: SOFTWARE FOR MONITORING AND STORING SOIL DATA

ABSTRACT

This paper describes the AgroSense software, a Precision Agriculture application. It allows soil moisture and temperature values to be stored in a database. To do this, the software communicates with a network of measurement hardware located in different growing areas, and stores the information in a database. Communication is carried out through the TCP/IP protocol, under the conception of the Internet of Things. For its implementation, Qt and SQLite database are used. The hardware is designed using the Arduino platform. The Qt development environment incorporates the necessary tools for TCP/IP communication and connection to the database without the need to install other tools. The implemented software is tested by connecting to the measuring devices implemented with Arduino. As result a server software is achieved that manages to establish a connection with two Arduino devices that act as clients. The results, verified at the laboratory level, show that there is no system collapse, the data is stored correctly in the database, and that the graphical interface is user-friendly and intuitive. Its implementation makes it possible to characterize soils, which allows decisions to be made related to irrigation and seed planting.

INDEX TERMS: Precision Agriculture, Qt, TCP/IP, Qt, Arduino.

1. INTRODUCCIÓN

La Agricultura de Precisión (AP) puede ser vista como un sistema de gestión. En ella se recolectan, procesan, y analizan datos temporales para una posterior toma de decisiones [1]. Se fundamenta en un uso extensivo de las Tecnologías de la Información [2, 3] para recolectar datos como características de los suelos, variables meteorológicas,

demanda de riego entre otros. Lo anterior permite hacer un uso eficiente de recursos, disminuir costos, optimizar el tiempo de cosecha e incrementar beneficios a partir de la productividad [2, 4]. Para poder realizar de forma efectiva el monitoreo de variables mencionado, en especial en extensas áreas de cultivo, se requiere del uso de redes de sensores. Una de las vías para lograr este objetivo es el uso del Internet de las Cosas (IoT) [5, 6]. IoT es definido por el Diccionario Oxford como la interconexión por medio de Internet de objetos de cómputo embebidos en objetos de uso diario con la correspondiente característica de enviar y recibir datos [7].

En Cuba existen antecedentes del uso de la AP. Se conoce del uso de sensores inteligentes, geolocalización, plataformas Web, uso de drones y monitoreo de cultivos por medio de técnicas multispectrales [1]. Esta demostrado por medio de estudios el uso ineficiente del agua en el regadío [8], y se reconoce la importancia de un riego y fertilización inteligente [9].

Entre las aplicaciones de la AP se encuentran el mapeo de suelos, así como la estimación de la temperatura y humedad de estos. El mapeo permite determinar la calidad de los suelos para decidir por el tipo de cultivo a sembrar. Otra importante aplicación es conocer si es preciso realizar el regadío, ya que este depende de las capacidades de drenaje del suelo [10, 11]. Un aplicación que permita el uso óptimo del regadío consiste una urgencia a nivel mundial ante la creciente necesidad de ahorrar el recurso [9, 12].

Es en este contexto que el Departamento de Telecomunicaciones de la Universidad de Oriente participa en un proyecto de colaboración con centros productivos agropecuarios llamado InteRiego. En este proyecto se trabaja en dos objetivos fundamentales: resolver problemas de automatización de los pivotes de riego central, y establecer un riego basado en los requerimientos del cultivo, las condiciones climáticas y las características del suelo.

El presente trabajo se enmarca dentro del segundo aspecto. Para ello realiza la medición de la temperatura y humedad del suelo de forma remota, y almacena los valores en una base de datos por medio de un *software* de almacenamiento. Dicha base de datos puede ser consultada con posterioridad para tomar decisiones de riego o siembra de determinados tipos de cultivo. Aprovecha el uso de Internet para establecer una lectura de las mediciones libre del accionar humano. La adquisición de los parámetros del suelo se realiza por medio de un Arduino UNO y una placa Ethernet (en lo adelante *hardware* de medición) que le provee conectividad. La base de datos es administrada por medio de un *software* de almacenamiento elaborado en el entorno de desarrollo Qt en su versión de *software* libre. Dicho *software* en lo adelante será llamado AgroSense. Como motor de base de datos se utiliza SQLite.

MATERIALES Y MÉTODOS

Sistema de monitoreo y almacenamiento

Para una mejor comprensión del trabajo se comienza por presentar la concepción general del sistema para la gestión del riego (Figura 1). El área productiva está formada por varias zonas de cultivo, cada una con un sistema de riego por pivote central. Dichos pivotes se comunican por medio de Internet, utilizando el protocolo TCP/IP, con un *software* servidor que provee la información de riego para cada pivote.

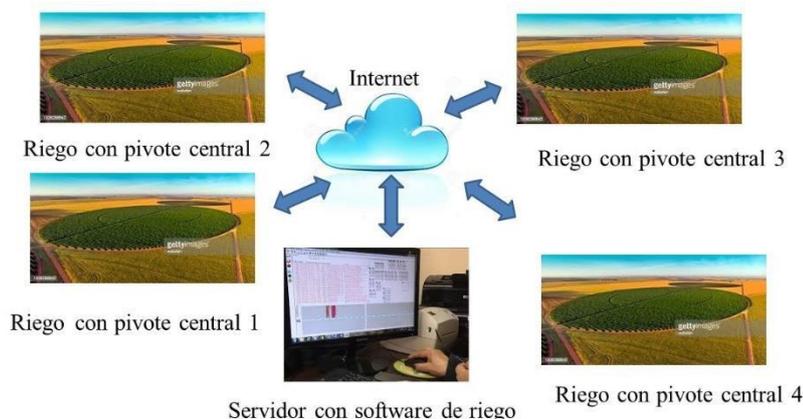


Figura 1: Concepción del sistema para la gestión del riego.

Para que el sistema funcione correctamente, además de otras variables como tipo de cultivo y condiciones climáticas, es necesario conocer la temperatura y la humedad del suelo. Por consiguiente, en cada zona de cultivo se planea colocar una *hardware* de medición que mida dichos parámetros. Este, luego de realizar la medición se comunica por Internet con AgroSense, el que administra una base de datos. Vale destacar que, aunque se propone un sistema de medición completo, el mayor énfasis de este trabajo se pone en el software AgroSense.

En la Figura 2 se muestra el esquema en bloques del sistema propuesto. La placa de Arduino adquiere los datos de los sensores de humedad y temperatura. El *Shield* Ethernet garantiza conectividad por medio del protocolo TCP/IP. El *hardware* Arduino al implementar la comunicación TCP/IP actúa como una aplicación cliente. AgroSense actúa como una aplicación servidor. Se implementa en Qt, y utiliza un motor de base de datos SQLite. Por simplicidad se muestra un solo elemento de *hardware*. Como se mencionó con anterioridad el sistema consta de tantos *hardware* de medición como áreas de cultivo existan. En el desarrollo del sistema se utiliza la interfaz de desarrollo Arduino 2.1, el *framework* QtCreator 4.6 y el motor de base de datos SQLite.

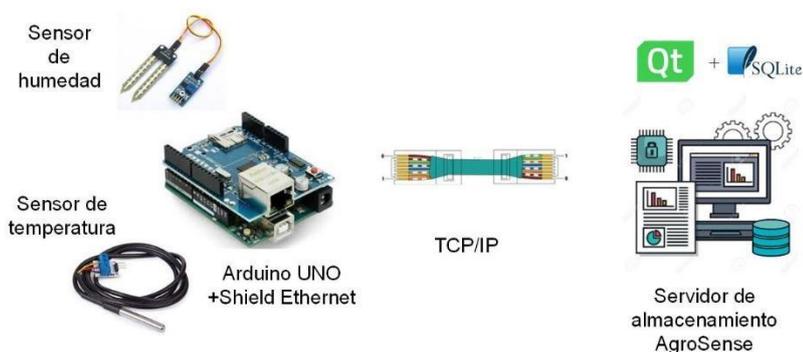


Figura 2: Sistema de monitoreo y almacenamiento. Diagrama en bloques.

Arduino 2.1 y comunicación TCP/IP

El entorno de desarrollo de Arduino en su versión 2.1 se basa en la programación en lenguaje C. Posee un amplio soporte de librerías que permiten implementar diferentes aplicaciones, y emplear con eficiencia los diferentes *Shields*, sensores y módulos electrónicos compatibles con Arduino. También posee una amplia comunidad de desarrolladores que facilita la implementación de aplicaciones [13]. Para la comunicación TCP/IP por medio del *Shield* Ethernet es

necesario utilizar las librerías Ethernet e SPI (*Serial Peripheral Interface*). La librería Ethernet permite implementar aplicaciones cliente o servidor, cliente o servidor Web, cliente telnet o cliente UDP. Soporta los protocolos TCP/IP y UDP [14, 15].

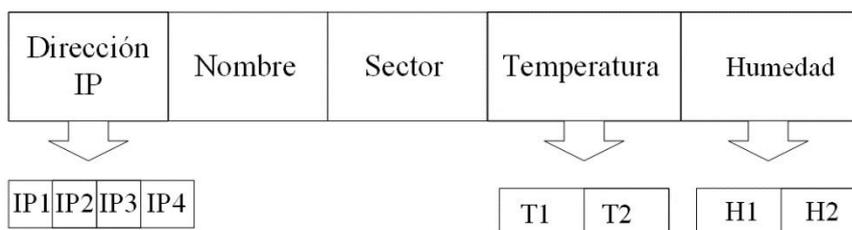
QtCreator 4.6. Conexión con la base de datos y comunicación TCP/IP.

Qt es un *framework* de C++ que agrega una amplia gama de funcionalidades a este lenguaje de programación. Uno de los aspectos más destacados de Qt es su capacidad para simplificar la programación en C++. Es creado por la empresa noruega Trolltech y actualmente es propiedad de Nokia. Es ampliamente utilizado por la Agencia Espacial Europea, *Dreamworks*, Samsung, Philips entre otros [16]. Posee una versión libre de pago que es la utilizada en este proyecto.

El uso de Qt permite una programación basada en eventos y una comunicación eficiente entre objetos sin necesidad de bucles de eventos complicados. Para ello introduce conceptos como meta objetos y el sistema de señales y *slots*. Esta característica simplifica la programación de interfaces de usuario receptivas y sistemas interactivos y es la causa de que sea ampliamente utilizada para crear interfaces visuales a otros lenguajes de programación como Python [16]. Permite la gestión de bases de datos sin grandes dificultades. Para la comunicación con SQL, posee las clases *QSqlDatabase* y *QSqlQuery*. Hace uso de las librerías *QtSql/QSqlDatabase*, *QtSql/QSqlQuery*, *QtSql/QSqlError*. Estas son suficientes para crear, abrir, cerrar, eliminar, consultar y manejar errores con bases de datos de tipo *SQLite* [17].

La comunicación TCP/IP requiere del uso de *sockets*. En el presente trabajo se utilizan las librerías *QTcpServer*, *QTcpSocket*, *QHostAddress* y *QAbstractSocket*. Estas garantizan el establecimiento de una aplicación servidor, la identificación de los clientes, la transmisión y recepción de datos, el establecimiento y cierre de conexiones. La sincronización de eventos TCP/IP se realiza por medio de tres *slots* que se activan ante: una nueva conexión, ante una desconexión, y ante una recepción de datos TCP/IP [18].

Para poder establecer una comunicación entre el *hardware* de medición y el *software* fue necesario establecer una trama de comunicación (Figura 3). Las librerías mencionadas garantizan la comunicación por medio TCP/IP, pero no establecen un formato de comunicación. Una vez que los datos son medidos por el *hardware* de medición, este construye la trama y la envía al servidor. Una vez que el servidor recibe el dato por alguno de los clientes, la transmisión se decodifica y se extraen los datos medidos, así como otros datos de identificación.



IP1-4: Cuarteta de la dirección IP

T1: Parte entera del valor de temperatura

T2: Parte decimal del valor de temperatura

H1: Parte entera del valor de humedad

H2: Parte decimal del valor de humedad

Figura. 3: Trama de datos.

El campo Dirección IP contiene la dirección IP asignada al *hardware*. Está formado por cuatro subcampos, uno para cada cuarteta de una dirección IPv4. El campo Nombre contiene el nombre asignado al *hardware*. Sector es el nombre del sector donde fue colocado. Los campos Temperatura y Humedad contienen los datos. Cada uno cuenta con dos subcampos para transmitir la parte entera y la parte decimal de la medición.

SQLite.

SQLite es un motor de base de datos ampliamente reconocido y utilizado. Su principal ventaja es que, a diferencia de otros motores de base de datos, funciona como un servidor propio e independiente, con lo que se evita tener que realizar consultas externas en procesos separados. Es una herramienta multiplataforma, compatible con varios sistemas operativos incluidos Windows, macOS, Linux o Android.

En el presente trabajo se crea una base de datos relacional. Una tabla contiene la dirección IP, nombre y sector de cada *hardware* de medición. A esta tabla se relaciona una segunda con los valores de temperatura y humedad medidos por cada *hardware* de medición.

2. RESULTADOS

La Figura 4 muestra el algoritmo de trabajo en el Arduino. Luego de la inicialización de variables relacionadas con la medición y la comunicación TCP/IP, se pasa a configurar la conexión como cliente y a establecer la comunicación con el servidor. En caso de que no se establezca la conexión se muestra un error. De esta forma un operario puede percatarse de que existe un error de comunicación. La acción de conectarse al servidor se repite mientras no sea exitosa.

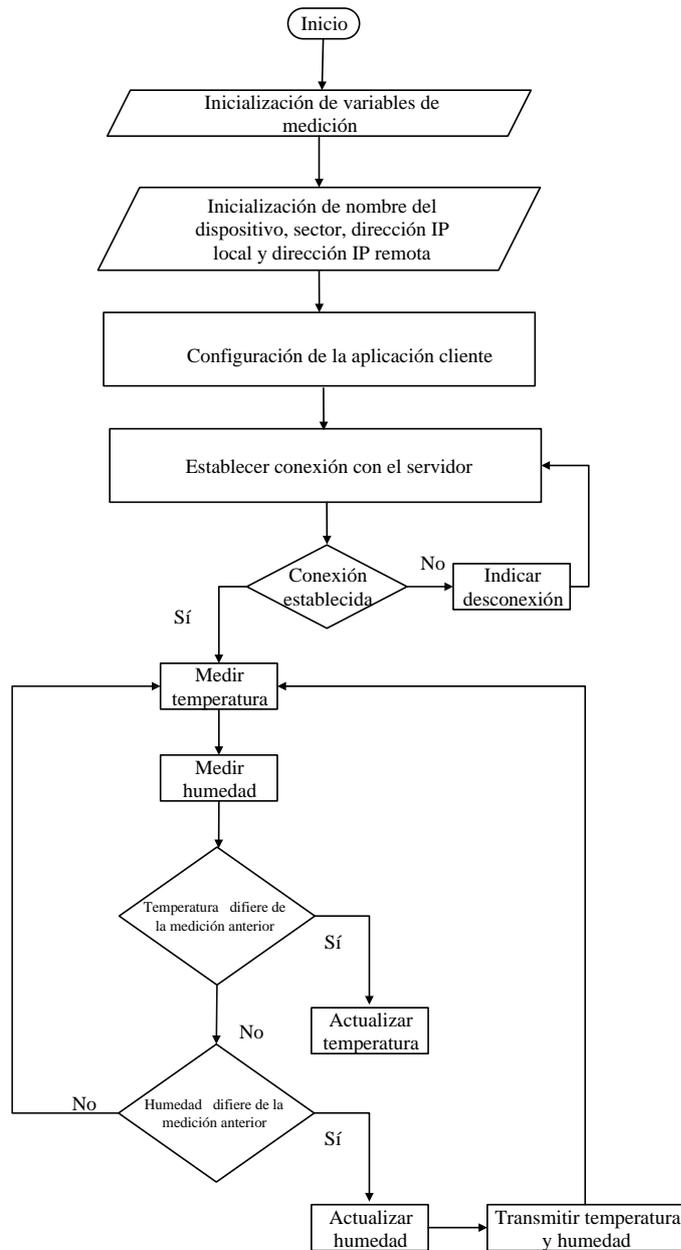


Figura. 4: Algoritmo de trabajo en el Arduino.

Una vez establecida la conexión, se pasa a medir los valores de temperatura y humedad. Estos se comparan con el valor anterior. En caso de que difiera alguno de ellos, se actualizan las mediciones y se transmiten los nuevos valores. El ciclo de medición se repite mientras el dispositivo se encuentre energizado. Los valores de temperatura y humedad se actualizan siempre que existan cambios de temperatura de $\pm 0,5^{\circ}\text{C}$ o $\pm 5,0\%$ de humedad con respecto a la última medición.

La Figura 5 muestra el algoritmo del *software* AgroSense. Al iniciarse el programa se comprueba si la base de datos existe. De no ser así esta se crea. Luego se pasa a configurar la aplicación servidor, las señales y los *slots*. En caso de existir la base de datos esta se abre y se pasa a realizar la configuración de la aplicación como servidor.

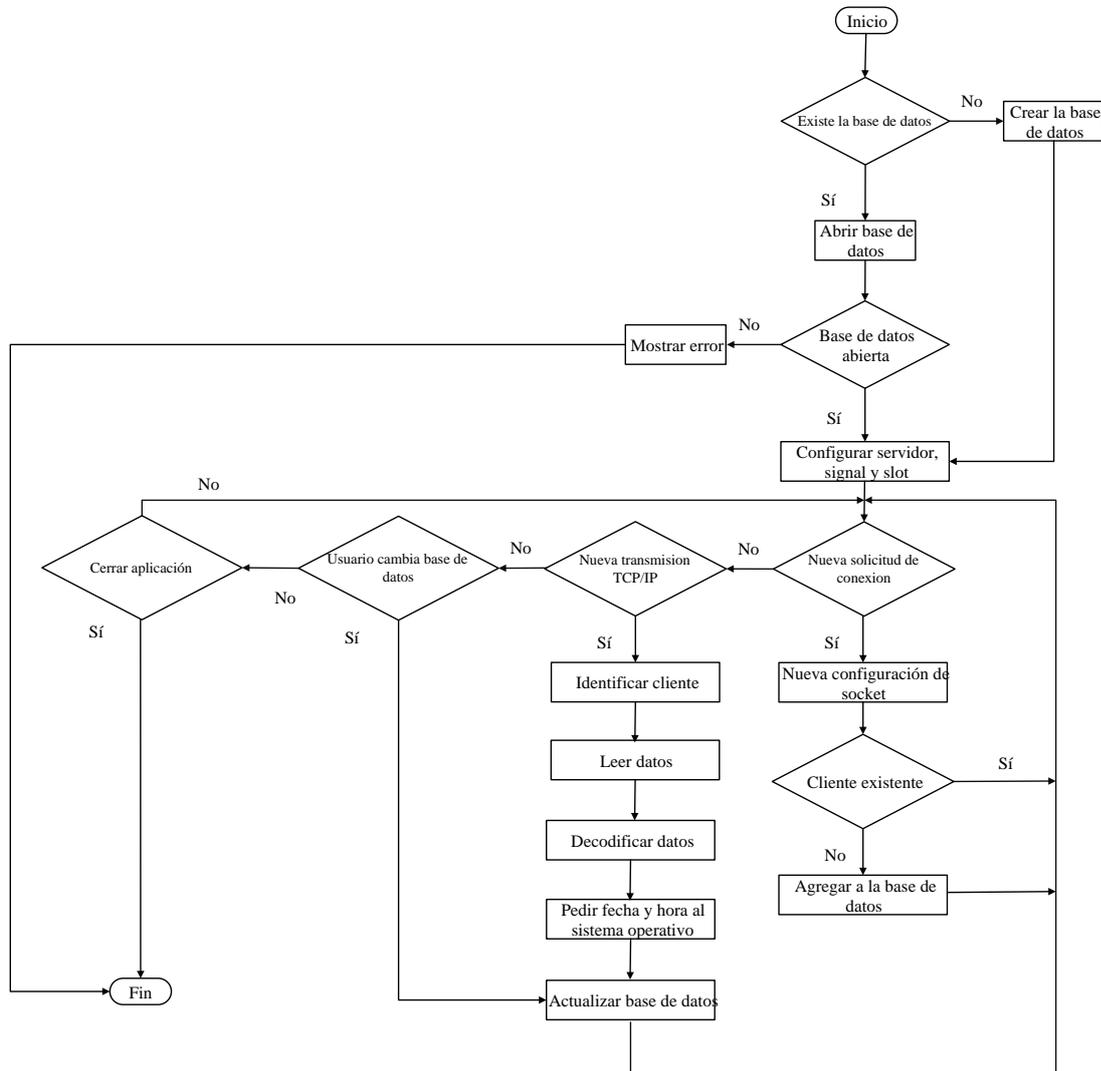


Figura. 5: Algoritmo de trabajo del software de almacenamiento.

Una vez configurado, la aplicación espera que ocurra una señal asociada a una solicitud de conexión, una señal asociada a una transmisión de un cliente, o que se active algún *slot* que indique que el usuario modificó la base de datos o desea cerrar la aplicación.

En caso de ocurrir una solicitud de conexión se configura un socket, se comprueba que el cliente se encuentre en la base de datos. Si se encuentra se espera que ocurra una de las señales o *slots* mencionados. Si no existe se agrega a la base de datos y se pasa a esperar la ocurrencia de una señal o slot.

En caso de que ocurra una transmisión de un cliente se identifica el cliente, se leen y decodifican los datos. Se determina la fecha y la hora y se guardan valores de temperatura y humedad transmitidos en la base de datos. Posteriormente se regresa a la condición de espera ya descrita. Si el usuario modifica la base de datos esta se actualiza y si desea cerrar la aplicación esta se termina. De igual forma si ocurre un error en la conexión con la base de datos se muestra un mensaje de error y se cierra la aplicación.

La Figura 6 muestra la interfaz gráfica creada. Se muestra la pestaña Dispositivos. En esta pestaña aparece el identificador del dispositivo, el sector al que se encuentra asignado, el nombre del dispositivo y la dirección IP. En la Figura 7 se muestra la pestaña Mediciones de la interfaz gráfica. En ella se muestra el identificador del dispositivo, el sector asignado, la fecha y hora en que se realizó la medición, así como los valores de temperatura y humedad.

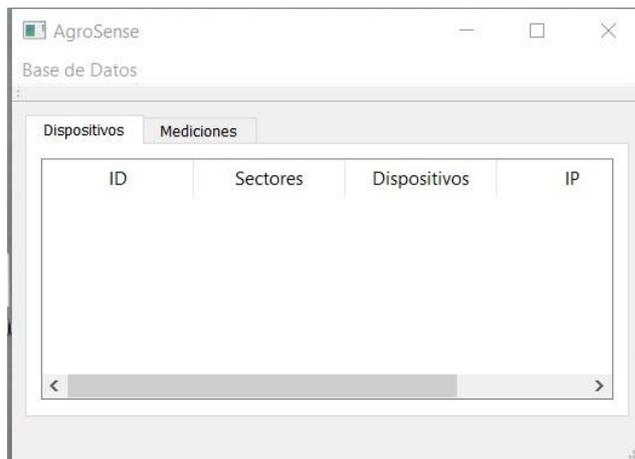


Figura. 6: Interfaz gráfica del software AgroSense. Pestaña de Dispositivos.

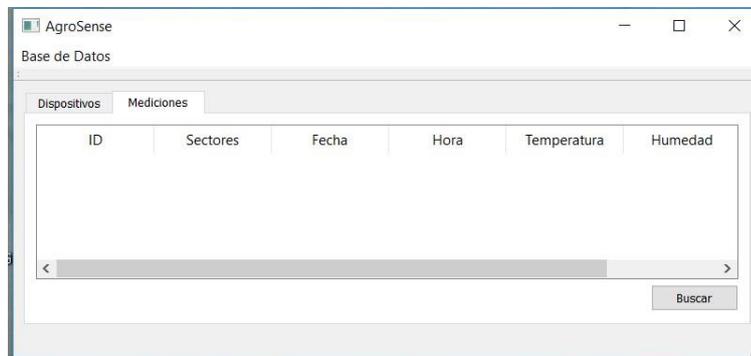


Figura. 7: Interfaz gráfica del software AgroSense. Pestaña de Mediciones.

La Figura 8 muestra el montaje en el laboratorio. Para la prueba se conectaron dos dispositivos utilizando un *router* Planet SW-801. La prueba se realizó durante ocho horas ininterrumpidas. Las Figuras 9 y 10 muestran los datos registrados en AgroSense.



Figura. 8: Montaje de prueba en el laboratorio.

ID	Sectores	Dispositivos	IP
1	2	Arduino_Mega	10, 30, 62, 148
2	1	Arduino_Uno	10, 30, 62, 150

Figura. 9: Datos en la pestaña Dispositivos.

ID	Sectores	Fecha	Hora	Temperatura	Humedad
1	2	2023-10-31	12:57:32	32.5	50
2	1	2023-10-31	12:57:33	31.5	45
3	2	2023-10-31	12:57:35	32.5	50
4	1	2023-10-31	12:57:35	31.5	45
5	2	2023-10-31	12:57:38	32.5	50

Figura. 10: Datos en la pestaña Mediciones.

3. CONCLUSIONES

En este artículo, se presenta un sistema de monitoreo y almacenamiento de datos en tiempo real para la agricultura de precisión utilizando comunicación TCP/IP. El *software* AgroSense permite registrar de forma remota y automática parámetros clave del suelo, como la temperatura y la humedad, a través de una red de sensores basada en la plataforma Arduino y con comunicación mediante el protocolo TCP/IP. Además, se garantiza el almacenamiento de los datos en una base de datos SQLite, lo que facilita su consulta para la toma de decisiones relacionadas con el riego y la gestión de cultivos.

A partir de los resultados experimentales, se puede afirmar que AgroSense ofrece una solución eficiente, basada en *software* y *hardware* libre para la caracterización del suelo, con el objetivo final de facilitar la toma de decisiones asociada a la siembra y riego de productos agrícolas. Tiene la ventaja de que puede ser adaptado para medir otras variables con ligeras modificaciones al ser un *software* de código abierto.

Los resultados mostraron que la integración de Qt y Arduino con *Shield* Ethernet permite realizar una aplicación compleja, con una interfaz gráfica amigable al usuario, que garantiza una comunicación basada en TCP/IP, y el almacenamiento de información en una base de datos. Se demostró que dicha comunicación es estable, en tiempo real y con una demora solo delimitada por la latencia de la red.

Como trabajo futuro se pretende crear un escenario de prueba donde se compruebe la comunicación a grandes distancias por medios inalámbricos. Además, lograr su integración con un *software* de toma de decisiones en el riego a partir de la evapotranspiración basado en la recomendación FAO56.

REFERENCIAS

- [1] R. R. Hernández, "La agricultura de precisión. Una necesidad actual," *Revista Ingeniería Agrícola*, vol. 11, no. 1, pp. 67-74, 2021.
- [2] I. Cisternas, I. Velásquez, A. Caro, and A. Rodríguez, "Systematic literature review of implementations of precision agriculture," *Computers and Electronics in Agriculture*, vol. 176, p. 105626, 2020.
- [3] B. Nowak, "Precision agriculture: Where do we stand? A review of the adoption of precision agriculture technologies on field crops farms in developed countries," *Agricultural Research*, vol. 10, no. 4, pp. 515-522, 2021.
- [4] A. Antúnez Sanchez, *La agricultura de precisión en la empresa agraria de Cuba*. Universidad Nacional, Costa Rica, 2019.
- [5] S. K. Roy, S. Misra, N. S. Raghuvanshi, and S. K. Das, "AgriSens: IoT-based dynamic irrigation scheduling system for water management of irrigated crops," *IEEE Internet of Things Journal*, vol. 8, no. 6, pp. 50235030, 2020.
- [6] R. Akhter and S. A. Sofi, "Precision agriculture using IoT data analytics and machine learning," *Journal of King Saud University-Computer and Information Sciences*, vol. 34, no. 8, pp. 5602-5618, 2022.
- [7] R. A. Mouha, "Internet of things (IoT)," *Journal of Data Analysis and Information Processing*, vol. 9, no. 2, pp. 77-101, 2021.
- [8] O. González-Cueto *et al.*, "Irrigation water productivity in selected crops from central region of Cuba," *Revista Ciencias Técnicas Agropecuarias*, vol. 29, no. 1, 2020.
- [9] E. A. Maldonado, L. E. C. Barreiro, and C. E. S. Toca, "Intelligent fertigation, pillar of sustainable agriculture," *Advances in Modern Agriculture*, vol. 3, no. 1, p. 12, 2022.
- [10] R. Allen, M. Smith, L. Pereira, D. Raes, and J. Wright, "Revised FAO procedures for calculating evapotranspiration," *Revised FAO procedures for calculating evapotranspiration-Irrigation and Drainage paper, N° 56 with testing in Idaho*, 2000.
- [11] M. Y. P Waller, *Irrigation and drainage engineering*. New York: Springer, 2016.

- [12] A. Domínguez, J. A. Martínez-López, H. Amami, R. Nsiri, F. Karam, and M. Oueslati, "Adaptation of a Scientific Decision Support System to the Productive Sector—A Case Study: MOPECO Irrigation Scheduling Model for Annual Crops," *Water*, vol. 15, no. 9, p. 1691, 2023.
- [13] A. Bouraiou *et al.*, "Temperature display on web browser using ethernet shield and LM35 sensor based on arduino board," in *2022 19th International Multi-Conference on Systems, Signals & Devices (SSD)*, 2022, pp. 421-424: IEEE.
- [14] Arduino. (2023, septiembre). *Ethernet*. Available: <https://www.arduino.cc/reference/en/libraries/ethernet>
- [15] P. Seneviratne, *Building Arduino PLCs: The essential techniques you need to develop Arduino-based PLCs*. Apress, 2017.
- [16] Q. C. Ltd. (2023, septiembre). *Qt documentation*. Available: https://wiki.qt.io/About_Qt
- [17] Q. C. Ltd. (2023, septiembre 2023). *Using the SQL Model Classes*. Available: <https://doc.qt.io/qt-6/sqlmodel.html>
- [18] Q. C. Ltd. (2023, septiembre). *Network Programming with Qt*. Available: <https://doc.qt.io/qt-6/qtnetworkprogramming.html#using-tcp-with-qtsocket-and-qtserver>

SOBRE LOS AUTORES

Michel Estrada Castro: Graduado de Ingeniero en Telecomunicaciones y Electrónica en la Universidad de Oriente en 2023. Trabaja en Planta Exterior en ETECSA Santiago de Cuba. Su labor investigativa se encuentra asociada al desarrollo de aplicaciones de Hardware/Software basadas en TCP/IP. **ORCID: 0009-0006-6533-3852**

Elio Manuel Castañeda González: Graduado de Ingeniero en Telecomunicaciones y Electrónica en la Universidad de Oriente en 2003. Máster en Sistemas de Telecomunicaciones en 2018 por la Universidad de Oriente. Especialista en Seguridad Informática y Administrador de Redes de Cubatur Santiago de Cuba. Trabaja en Planta Exterior en ETECSA Santiago de Cuba. Profesor Asistente. Imparte docencia a tiempo parcial en la Universidad de Oriente en las asignaturas de Redes, Dispositivos Integrados Programables y Electrónica Digital. Su labor investigativa se encuentra asociada al desarrollo de sistemas de Hardware. **ORCID: 0009-0005-1163-7693**

Jorge Esteban Santos Toural: Graduado de Ingeniero en Telecomunicaciones y Electrónica en la Universidad de Oriente en 2003. Máster en Sistemas de Telecomunicaciones en 2009 por la Universidad de Oriente. Dr.C. técnicas por la Universidad de Oriente en 2021. Profesor Titular. Ha impartido docencia en Electrónica Analógica, Mediciones Electrónicas, Instrumentación Virtual, Dispositivos Integrados Programables entre otras. Posee varios registros de *software*. Su labor investigativa se encuentra asociada al desarrollo de sistemas de Hardware/Software basados en IoT, la domótica y el procesamiento digital de señales médicas. Miembro de la Asociación Cubana de Reconocimiento de Patrones. **ORCID: 0000-0002-6249-550X**

CONFLICTO DE INTERESES

No existen conflictos de intereses

CONTRIBUCIONES DE LOS AUTORES

- ✦ **Michel Estrada Castro:** Desarrollo de la aplicación de software. Conceptualización del sistema. Redacción.
- ✦ **Elio Manuel Castañeda González:** Desarrollo de la aplicación de hardware. Pruebas. Redacción. Sugerencias acertadas para la conformación de la versión final
- ✦ **Jorge Esteban Santos Toural:** Conceptualización del sistema. Desarrollo de la aplicación de software. Redacción. Revisión crítica de cada una de las versiones del borrador del artículo y aprobación de la versión final a publicar.

CONTACTOS

Si tiene alguna pregunta u observación por favor diríjase a la siguiente dirección de correo electrónico jsantos@uo.edu.cu

TELEMÁTICA

REVISTA DIGITAL de las Tecnologías
de la Información y las Telecomunicaciones

Revista Telemática. Vol.22 , Año.2023, p.148- 159

ISSN 1729-3804

Esta revista provee acceso libre inmediato a su contenido bajo el principio de hacer disponible gratuitamente investigación al público. Los contenidos de la revista se distribuyen bajo una licencia *Creative Commons Attribution-NonCommercial 4.0 Unported License*. Se permite la copia y distribución de sus manuscritos por cualquier medio, siempre que mantenga el reconocimiento de sus autores y no se haga uso comercial de las obras.

