

Desarrollo de una plataforma de bajo coste y código abierto para sistemas de riego de precisión

Francisco Puig Pérez-Barquero¹, Juan Antonio Rodríguez Díaz², Emilio Camacho Poyato³ y Rafael González Perea⁴

¹Universidad de Córdoba; Edificio Leonardo da Vinci; Campus de Rabanales 14071-Córdoba, g32pupef@uco.es

² ma2rodij@uco.es

³ ag1capoe@uco.es

⁴ g72goper@uco.es

En este trabajo se ha desarrollado un sistema IoT que facilita la integración con otras plataformas y es compatible con cualquier tipo de sensor. El sistema incorpora estrategias de riego de precisión, que se calculan automáticamente haciendo uso de los datos climáticos y de los sensores. Además, se ha desarrollado un nodo de comunicación, compatible con varios sistemas de comunicación y sensores comerciales.

El cambio climático, unido al incremento de la demanda de agua en la agricultura y otros sectores, está produciendo que el balance entre el agua requerida y el agua disponible se encuentre en niveles cada vez más críticos. El internet de las cosas (IoT) emerge como una elección natural para aplicaciones de riego de precisión (Kamienski et al 2019). El uso de sistemas de riego de precisión basados en sensores que envían datos en tiempo real, acompañados de estrategias que permitan optimizar la productividad del agua, se esté convirtiendo en algo indispensable.

Debido a esta tendencia, se está observando un aumento en el número de plataformas IoT en el sector agrícola. Sin embargo, estas plataformas se caracterizan por ser sistemas cerrados, que solo permiten la conexión de unos pocos sensores y se limitan a mostrar los valores de forma gráfica y sin dar información adicional sobre las necesidades de la planta, lo que limita su potencial para uso en la toma de decisiones.



soluciones de código abierto destinadas a tratar problemas específicos de IoT (Viola et al, 2019).

El componente principal en la segunda capa es el bróker de contexto llamado Orion-LD, cuya función es administrar, consultar y actualizar la información de contexto. En FIWARE el contexto hace referencia a todo lo que ocurre a nuestro alrededor y puede ser medido o alterado por los sensores. La comunicación con el bróker se realiza a través de la API NGSI-LD (Cantera et al, 2019). Esta API tiene la ventaja de que está estandarizada y como consecuencia permite la conexión con otras plataformas que la utilicen.

Los demás componentes de la segunda capa, llamados microservicios, se conectan al bróker a través de su API y cada uno de ellos realiza una función específica. El microservicio IoT Agent es el encargado de controlar los sensores y recibir sus datos. QuantumLeap es otro microservicio que tiene como objetivo almacenar los valores de los sensores de forma persistente en una base de datos llamada CrateDB. El sistema de autenticación es un microservicio que permite la interacción segura con el usuario y se conecta con la base de datos MongoDB para almacenar toda la información. Por último, el microservicio estrategias de riego se basa en un algoritmo de riego que calcula la dosis de agua que hay que aplicar. Estos dos últimos se han desarrollado haciendo uso del framework ExpressJS (<https://expressjs.com>).

Para calcular las necesidades hídricas, el algoritmo de riego utiliza la metodología propuesta por la FAO para realiza un balance de agua en el suelo (Allen et al, 2006). Este balance de agua hace uso de los valores de los sensores, las características de la explotación, las características del sistema de riego, los datos climáticos y el tipo de estrategia de riego. Los datos climáticos los obtiene conectándose a la API de la AEMET y a la página web elTiempo.es y los demás valores los obtiene conectándose al bróker Orion-LD. El algoritmo da como resultado una predicción semanal del tiempo que el sistema de riego debe estar funcionando.

La tercera y última capa es la interfaz gráfica de usuario. Esta capa interactúa con la segunda mediante la API NGSI-LD a través del protocolo HTTP (Hypertext Transfer Protocol). Para su desarrollo se ha utilizado la librería de código libre ReactJS (<https://reactjs.org>).



Figura 2. Nodo de comunicación desarrollado por la Universidad de Córdoba.

Resultados del trabajo

Se instalaron dos nodos de comunicación, uno por cada sector. A cada nodo se le conectaron tres sensores de humedad a 15, 30 y 45 cm de profundidad y un sensor de potencial hídrico a 30 cm de profundidad. Debido a la localización remota de la explotación y al reducido número de dispositivos, se ha optado por la conectividad SigFox.

Tanto los datos medidos por los sensores como los resultados del algoritmo de riego se almacenan en tiempo real en una base de datos para posteriormente mostrarse en la aplicación de forma gráfica (**Figura 2**).



Sensor instalado en el olivar.

Ventajas de la plataforma creada

Esta plataforma se caracteriza principalmente por tres cosas:

- La primera es que no solo se limita a representar los valores de los sensores, sino que además realiza un balance de agua en el suelo para los próximos 7 días, basado en las predicciones meteorológicas, que permite programar el riego de una forma más eficiente.
- La segunda ventaja es que es polivalente. La arquitectura se ha pensado para que cualquier sensor se pueda conectar. Esto permite que la aplicación se adapte a cualquier uso, ya sea agrícola o de cualquier otro tipo. Un ejemplo de esta ventaja lo encontramos en las comunidades de regantes, en donde con sistemas de este tipo se podría integrar toda la información en una sola plataforma. Esta información, por un lado, corresponde a la red de distribución en la que se encuentran las tuberías principales, el sistema de bombeo, los embalses y otros elementos de la comunidad. Por otro lado, están las parcelas agrícolas, donde cada agricultor tiene su cultivo y su sistema de riego.

- La última gran ventaja es la estandarización de su API. La ventaja de estandarizar una API es que se consigue desarrollar aplicaciones en base a las mismas reglas. De este modo, podemos conectarlas entre si sin apenas configuración y sin necesidad de tener que conocer su funcionamiento. Esto sería de gran utilidad en las comunidades de regantes dado que esa información podría usarse para el desarrollo de sistemas de ayuda a la toma de decisiones o de modelos predictivos que mejoren su funcionamiento. Usando la arquitectura propuesta en este trabajo, se podrían conectar distintas plataformas e integrar la información de las mismas para poder tomar decisiones en tiempo real.

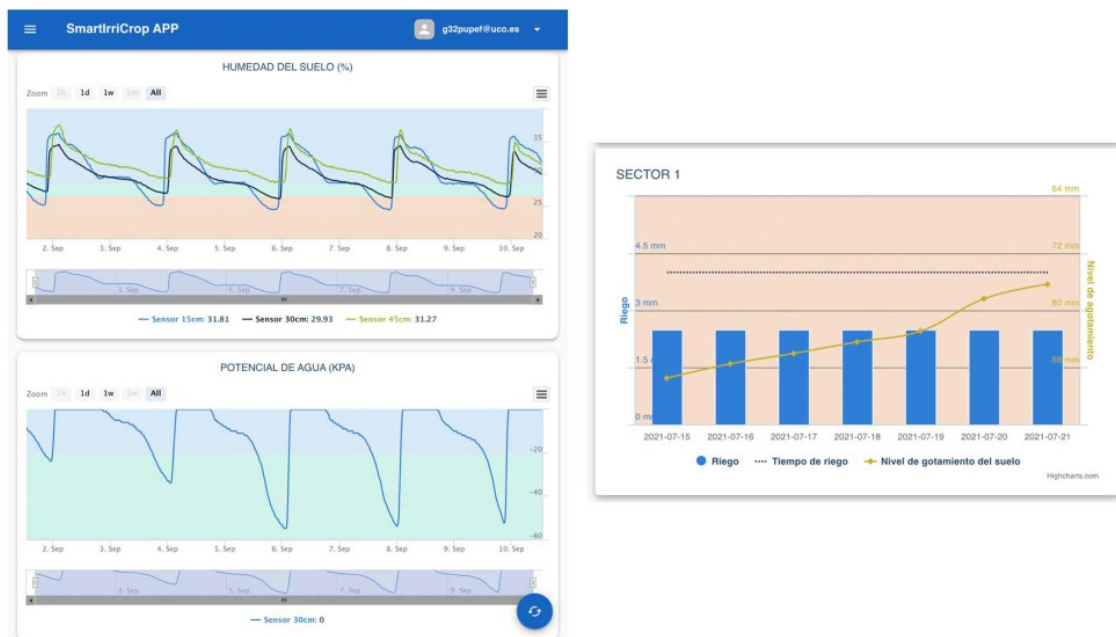


Figura 3. Gráficas que pueden verse en la aplicación web.

Conclusiones

En este trabajo se ha desarrollado un sistema IoT que permite la interconexión entre diferentes plataformas, es compatible con una gran cantidad de sensores y calcula diferentes estrategias de riego de precisión. El sistema predice para los próximos 7 días la variación de humedad en el terreno, basándose en la metodología propuesta por la FAO. Para ello se apoya en datos obtenidos por sensores de humedad y en predicciones meteorológicas.

La plataforma utiliza el broken Orion-LD de FIWARE que hace uso de la API NGSI-LD. Al estar estandariza y usar datos enlazados (linked-data), permite que plataformas desarrolladas con la misma API puedan conectarse entre sí sin apenas configuración. De esto modo se consigue tener una plataforma escalable y abierta que puede compartir y utilizar datos de otras plataformas IoT.

Agradecimientos

Este trabajo se ha desarrollado dentro de la Unidad de Excelencia María de Maeztu (DAUCO).

Bibliografía

Carlos Kamienski, Juha-Pekka Soininen, Markus Taumberger, Ramide Dantas, Attilio Toscano, Tullio Salmon Cinotti, Rodrigo Filev Maia and André Torre Neto, (2019), *Smart Water Management Platform: IoT-Based Precision Irrigation for Agriculture*, *Sensors* 2019, 19, 276.

Allen, R. G., Pereira, L. S., Raes, D., & Smith, M. (2006). *Evapotranspiración del cultivo Guías para la determinación de los requerimientos de agua de los cultivos*. In *Estudio FAO Riego y Drenaje*. (Vol. 56).

José Manuel Cantera, Martin Bauer, Guilles Privat, Wenbin Li, David Fernandez, Mike Fisher. S. G. C. Information, "ETSI GS CIM 009 V1.1.1 (2019-01), Context Information Management (CIM); NGSI-LD API," *tech. rep.*, ETSI Management (ISG-CIM), 2019.

F. Viola, F. Antoniazzi, C. Aguzzi, C. Kamienski, and L. Rof?a (2019), Mapping the NGSI-LD context model on top of a SPARQL event pro-cessing architecture: Implementation guidelines, in Proc. 24th Conf. Open Innov. Assoc. (FRUCT), pp. 493–501.