UNIVERSIDAD AUTÓNOMA DE MADRID

ESCUELA POLITÉCNICA SUPERIOR

TRABAJO FIN DE MÁSTER

Desarrollo de un sistema de monitorización y diagnóstico en entornos de agricultura de precisión

Máster Universitario en Investigación e Innovación en Tecnologías de la Información y las Comunicaciones

Autor: REDONDO QUINTERO, Alejandro
Tutor: BELLOGÍN KOUKI, Alejandro

FECHA: Septiembre, 2017

DESARROLLO DE UN SISTEMA DE MONITORIZACIÓN Y DIAGNÓSTICO EN ENTORNOS DE AGRICULTURA DE PRECISIÓN

AUTOR: Alejandro Redondo Quintero
TUTOR: Alejandro Bellogín Kouki
PONENTE: Fernando Díez Rubio

Dpto. Ingeniería Informática
Escuela Politécnica Superior
Universidad Autónoma de Madrid

Septiembre de 2017

Resumen

Durante los últimos años, España ha sido el principal exportador mundial de fruta y verduras frescas. Esto ha sido posible gracias a la tradición hortícola del país y a un amplio conocimiento que se ha adquirido con el paso de los años. Este conocimiento apenas ha sabido adaptarse a las posibilidades que ofrecen las nuevas tecnologías, especialmente en temas de ahorro y optimización. Las decisiones de gestión de cultivos se toman basadas en la experiencia, de forma intuitiva y cuando un problema ya ha ocurrido. Además, las opciones tecnológicas de monitorización de cultivos existentes son muy heterogéneas, y no hay ninguna alternativa que haya sido capaz de unificarlas con éxito.

El objetivo de este proyecto es desarrollar un sistema que recoja datos tomados en diferentes cultivos y los simplifique para mostrárselos al usuario. Debido a la variedad de opciones, el sistema debe ser capaz de recuperar datos de los principales fabricantes de sensores y de predicciones climáticas. A partir de estos datos, el sistema detectará situaciones de riesgo para el crecimiento de las plantas que se hayan producido o se puedan producir en el futuro. Estas situaciones se mostrarán al usuario a través de una interfaz web que combinará las métricas calculadas junto con información espacial.

Para todo esto es necesario investigar los requisitos para la obtención de datos de cada uno de los fabricantes de sensores, estudiar las situaciones de riesgo que pueden ocurrir durante el crecimiento de la planta, diseñar un sistema de eventos que detecte dichas situaciones y planificar el desarrollo de la aplicación web que mostrará los resultados obtenidos. Hay que destacar que este proyecto ha sido desarrollado en un entorno real, utilizando datos procedentes de fincas reales que están siendo cultivadas.

Palabras clave

Monitorización, diagnóstico, cultivos, recuperación de datos, agricultura, aplicación web.

Abstract

In the last years, Spain has become the world's largest exporter of fresh fruits and vegetables. This has been possible because of the horticultural tradition of the country, together with a broad knowledge that has been acquired over the years. This knowledge has barely been able to adapt to the possibilities offered by new technologies, especially in terms of saving and optimization issues. Crop management decisions are made based on experience and intuition, and once a problem has already happened. Additionally, technological options for monitoring crops are very heterogeneous, and none of them has been able to unify successfully all the systems.

The goal of this project is to develop a system that collects data from different crops and simplify them before showing them to the final user. Because of the variety of options, such a system must be able to collect data from the main manufacturers of sensors and from weather predictions. Based on these data, the system will detect risk situations for the plants growth that have occurred or may occur in the future. Those situations will be presented to the user through a web interface that will combine the computed metrics with spatial information.

With these goals in mind, it is necessary to investigate the requirements to obtain the data from each manufacturers of sensors, to study the risk situations that can occur during the plant growth, to design an event system which detects these situations, and to plan the development of a web application that will show the obtained results. It should be noted that this project has been developed in a real environment, using data from real estates which are being cultivated.

Keywords

Monitoring, diagnosis, crops, data recovering, agriculture, web application.

Índice de contenidos

Índice de ilustraciones

Índice de tablas

1. Introducción

La agricultura contribuye positivamente al balance comercial de la economía española con un superávit de 8.017 M€ (Ministerio de Agricultura, Alimentación y Medio Ambiente, 2014). Esto se debe fundamentalmente a la producción de frutas y hortalizas que se localiza en la costa mediterránea. En estas zonas de cultivo, el principal reto al que se enfrenta la agricultura es el de mantener la calidad y seguridad de la producción haciendo frente a cada vez una menor disponibilidad de agua. Además, debido al cambio climático y a la competencia con otros sectores económicos, es esperable que en los próximos años los recursos hídricos disponibles para la agricultura puedan ser más escasos y caros (Grupo Intergubernamental de Expertos sobre el Cambio Climático, 2014).

Para hacer frente a esta situación, además de continuar con las modernizaciones de los regadíos, es cada vez más importante incrementar la eficiencia en el uso del agua (Ministerio de Medio Ambiende y Medio Rural y Marino, 2015). En particular, a nivel de parcela, en la actualidad el riego suele aplicarse de forma muy empírica dado que no existen herramientas sofisticadas y de fácil empleo que permitan tomar decisiones sobre la programación del riego. Las investigaciones realizadas en los últimos diez años han puesto a punto nuevas técnicas de riego para el manejo eficiente del agua en parcela (Carlos, y otros, 2014) (F., y otros, 2014) que, sin embargo, no se aplican a nivel comercial debido a:

1. Carencia de capacitación
2. Difícil acceso a la información
3. Complejidad que conlleva materializar las estrategias de riego eficiente en fincas comerciales.

El liderazgo mundial español en especies arbóreas y hortícolas está asentado en un *know-how* de gestión técnica de estos cultivos que incluye el control de los procesos de la planta (desarrollo vegetativo, estrés, cuajados, etc.) mediante el uso de *inputs* (agua, abono, energía, tratamientos, etc.). Este conocimiento tiene una base técnica que muchos profesionales españoles están exportando por todo el mundo, aunque tomando decisiones intuitivas, ya que no se emplean para ello datos objetivos.

Los cultivos arbóreos y hortícolas, que mueven todos los años miles de millones de euros, tienen un coste oculto debido a que la mayoría de decisiones de gestión sobre los mismos son realizadas de forma estimada, basadas en la experiencia, una vez que el problema ya es visible, sin tener datos objetivos del estado de la planta, del suelo, del clima ni utilizando referencias objetivas de las experiencias del pasado. Una complicación adicional es que, como se sabe, cada explotación tiene unas características propias que la hacen única (microclima, suelo, tipo de agua, etc.).

Otro problema añadido es la integración de todas las disciplinas que lo componen ya que la utilización del conocimiento se hace de forma personalizada y poco sistemática, lo que impide alcanzar todo el potencial del *know-how*. Es decir, cada profesional tiene su propio conocimiento y su modo de actuar, adquirido tras la experiencia de años realizando su trabajo, y la transmisión del mismo entre profesionales es complicada. Se constituye así un círculo vicioso que impide la expansión del conocimiento por la falta de datos con un lenguaje común para comunicar la experiencia.

El hecho de que la agricultura sea una actividad mayoritariamente artesanal y donde la gestión de los *inputs* es intuitiva y basada en la experiencia del propio agricultor, hace que sea un sector muy mejorable. Se riega por intuición, se abona y se trata por estandarización, y la mayor parte del proceso se calcula sin un estricto rigor objetivo. Esto supone una catástrofe en términos de eficiencia.

La mejora de estos procesos sólo se puede hacer mediante una reingeniería total de los mecanismos de toma de decisión en el campo gracias al uso de nuevas tecnologías. Lo que se propone en este proyecto es el desarrollo de un sistema que obtenga información de distintas fuentes de datos (información espacial, mediciones de sensores y predicciones climáticas) y los combine aplicando una serie de cálculos. Los resultados se mostrarán en una interfaz web que permita a los usuarios (agricultores y técnicos de finca) conocer el estado de sus tierras para poder hacer frente a posibles problemas en las mismas.

1.1. Oferta actual de soluciones

Actualmente hay una gran oferta de opciones para tratar de mejorar los procesos de control de cultivos, pero ninguna ofrece una solución definitiva. A continuación se detallan los componentes principales de dichas alternativas:

- Dataloggers y red de comunicación: se conectan a sensores y envían sus datos al receptor principal vía radio o internet. Se ha pasado de disponer sólo de sistemas tipo radio a sistemas GPRS, que facilitan el acceso y abaratan el servicio. Esta es la base de negocio de los fabricantes de plataformas de comunicación, que invierten en su desarrollo para intentar diferenciarse. Desde hace 5 años hay una explosión mundial de fabricantes a escala "local", que suelen venir del mundo industrial. Los sensores necesitan dataloggers para captar los datos, por lo que es una industria con gran potencial de crecimiento y hay mucha competencia, pero no tienen crecimiento agronómico. El mercado presenta dataloggers de alta calidad a un elevado precio, como ADCON (ADCON Telemetry), pero la tendencia es encontrar GPRS aceptables por precios más asequibles como, por ejemplo, ModPow (Modpow).
- Software: suele tratarse de un servicio online con sistemas SCADA (SCA17) que ofrecen gráficas multivariables. Excepto los líderes del mercado, como ADCON, la mayoría de proveedores tiene software SCADA básico. Hay algunos que lo venden y otros que cobran por el alojamiento (*hostage*) online.
- Sensores: en agricultura los sensores se pueden aplicar a distintos niveles, por ello, se diferencian según sus ámbitos de funcionalidad:
    - Planta: aunque hay varias empresas ofertando sensores, todavía existe mucha controversia científica sobre el uso de su información, por lo compleja que es, lo que hace que apenas haya agricultores usándolo en el mundo. Simboliza el gran reto para los próximos años: utilizar datos de planta de distintas fuentes (dendrometría, fotografía, sensores de nutrición y satélite) y someterlos a estudio.
    - Suelo: los sensores de humedad en el suelo se han asentado y son la gran novedad de los últimos 20 años. Su precio ha bajado y su interpretación es sencilla: sirven para saber si las raíces están trabajando y a qué

profundidad está llegando el riego. Su uso es necesario para una primera aproximación al mejor rendimiento de los cultivos. Todas las plataformas están incluyendo sensores de agua en suelo.

  - Clima: todas las plataformas actuales ofrecen sensores climáticos. Estos sensores miden magnitudes meteorológicas como la temperatura, humedad relativa, nivel de radiación, pluviometría o velocidad del viento. Aunque tradicionalmente se utilizaban datos climáticos recogidos a nivel nacional o regional, en los últimos años se está comprobando la importancia de tener estaciones meteorológicas dentro de la propia finca que sean capaces de monitorizar el microclima específico de la zona cultivada.

- Previsión de clima: la mayoría de servicios ofrecen algún tipo de previsión climática global, como enlaces a previsiones de AEMET (Agencia Estatal de Meteorología) o similar. No obstante, la verdadera evolución que aún no está teniendo lugar consiste en una previsión climática que aprenda con microclima, lo cual supone el primer escalón para promover la venta de equipos a nivel de finca, dado que conocer lo que va a ocurrir con precisión es la clave para optimizar tratamientos y riego en el entorno de cada finca. En este ámbito, las soluciones que ofrece el mercado son de empresas muy especializadas, que tienen costes muy elevados, o servicios gratis por internet, que no tienen valor real. Se debería tratar de un servicio que fuera muy barato, para que se pueda producir la implantación a gran escala. La tendencia es que los fabricantes de dataloggers negocien este servicio con las grandes empresas de servicios meteorológicos y lo empiecen a ofrecer en los próximos años.
- Teledetección y drones: es otro de los retos que apenas tiene aplicación en España. Algunas empresas ofrecen estos servicios, pero su utilización en cultivos arbóreos (vid, olivo, frutas, etc.) es, por el momento, muy limitada, y su uso está más implantado en cultivos herbáceos (maíz, soja, etc.).
- Sistemas de Información Geográfica (conocido como GIS, por sus siglas en inglés) (MappingGIS): existen muchas soluciones propietarias, que implican un alto coste. Su principal inconveniente es que las empresas que lo ofertan no suelen

permitir su integración con otras aplicaciones, lo cual es fundamental para aprovechar toda la información de forma unificada.

- Análisis de savia: técnica para el análisis de cantidad de nutrientes patentada en España por Carlos Cadahía (López, 2008), resuelve muchas de las limitaciones del análisis foliar y se adapta muy bien a la metodología desarrollada, al completar la información puntual nutricional de la planta junto con los datos de la sonda de nitratos y el dendrómetro.
- Maquinaria de tratamientos: se están empezando a registrar todos los datos de recorrido del tractor y las condiciones de tratamiento, y se busca integrar con las condiciones de clima y con GIS.

El principal problema que tiene una empresa que quiera entrar en la llamada "agricultura Smart" es que tiene que trabajar con, al menos, 6 proveedores diferentes de estas tecnologías (sensores, previsión de clima, teledetección, nutrición, GIS, maquinaria), pero todos éstos están descoordinados y poseen softwares diferentes. Al final, los datos que pertenecen al diagnóstico complejo de un mismo problema están en distintas fuentes y no pueden ser utilizados por el técnico para entender dicho problema, lo cual limita siempre el éxito en el uso de la tecnología. La experiencia indica que la venta de servicios vinculados a cada tecnología tiene un mercado limitado, porque solo resuelven problemas parciales y no completos.

1.2. Problemas a resolver en la agricultura Smart

Actualmente, hay una gran desconexión entre las necesidades de los agricultores y las soluciones que se están ofreciendo en el mercado de la agricultura de precisión. Los clientes necesitan que se les hable de sus objetivos, que suelen ser el resultado de integrar muchos factores (la calidad de un vino, el calibre y sabor de un tomate, etc.) y, en cambio, solo se les habla de la optimización de procesos parciales (el riego, el abonado, tratamiento de cultivos, etc.) pero no se les ofrece una solución integrada de datos con trazabilidad que ayude a entender y mejorar el resultado de los productos que ellos venden.

Los principales problemas a la hora de integrar soluciones tecnológicas al mundo agrario son:

- Riesgo de la inversión. En general, se percibe que se va a necesitar una alta inversión inicial con una confusa identificación de los posibles beneficios.
- Complejidad de las soluciones: existe un alto temor a que la incorporación de nuevas tecnologías acarree costes adicionales, pérdidas de tiempo y nuevas fuentes de problemas técnicos.
- Falta de compatibilidad y escalabilidad. Una de las principales restricciones para el intercambio de datos se debe a que las soluciones software que se ofrecen no utilizan el mismo formato de datos y es difícil almacenar los datos que se generan. La falta de estándares abiertos conduce a problemas de compatibilidad, y los anchos de banda de transferencia de datos son limitados para manejar los datos recogidos por varios sensores, los enfoques y las escalas temporales y espaciales (Kempenaar, 2014).
- Gestión temporal y espacial de datos. No existen técnicas que faciliten la gestión temporal y espacial de datos, que es crítica para ofrecer una solución válida para grandes fincas.
    - Hay que resolver la ubicación óptima de los puntos de muestreo de sensores.
    - No hay soluciones que integren en mapas datos de diferente origen tecnológico (teledetección, recorrido de tratamientos, etc.) con otras fuentes de datos.
- Pobre procesamiento de datos. Las soluciones existentes potencian sobre todo la visualización en gráficas, pero apenas hay aprendizaje y este se ve limitado si solo se utilizan sensores de clima y humedad en el suelo, pues estos no aportan la información total necesaria para entender los resultados.
- En el sector agrario no es habitual pagar por procesamiento de datos.
- Problemas de las grandes organizaciones. Los casos de éxito que han tenido lugar en pequeñas fincas se ha producido como consecuencia de una gran inversión económica. El uso de la agricultura de precisión en grandes organizaciones apenas existe debido a:
    - La inversión de equipos es alta y no está resuelta de forma eficiente la extrapolación a nivel espacial de las alertas que se pueden detectar en los puntos de monitorización.

    - La complejidad organizativa dificulta promover el trabajo en equipo entre todos los eslabones de la cadena de trabajo (agricultores, técnicos, etc.).
- Optimización de la gestión técnica.
    - Las decisiones de gestión de inputs en producción son tomadas por los agricultores básicamente por intuición, con el apoyo puntual de los técnicos, y no existe un lenguaje que permita analizar, extrapolar y entender los resultados y que, además, facilite la comunicación para enseñar las experiencias.
    - Las decisiones técnicas más críticas son los tratamientos, el riego, el abonado y la recolección, y afectan a la rentabilidad, calidad y sostenibilidad. Estas decisiones tienen un fuerte componente preventivo y se dan con un fuerte margen de seguridad. Se prefiere regar, abonar y tratar de más a tener un problema.
    - No se dispone de un registro de los datos técnicos de lo ocurrido en cada finca por lo que no hay un aprendizaje que pueda servir para identificar buenas y malas prácticas de cultivo analizando los resultados finales.

1.3. Oportunidades del proyecto

A continuación, se enumeran los elementos diferenciadores del proyecto propuesto frente a las soluciones existentes en el momento actual:

- España es líder mundial en exportación de frutas, hortalizas, aceite y vino, y tiene un *know-how* de gestión técnica de éxito donde se controlan las plantas de forma "intuitiva". Con la realización del proyecto, se pretende documentar, mejorar y sistematizar el conocimiento existente para que se convierta en un valor diferencial añadido para el país, nos asegure el liderazgo futuro y se fomente la exportación de conocimiento.
- Se ha avanzado mucho en términos de soluciones técnicas, pero, para que la agricultura de precisión sea ampliamente adoptada, se necesita la integración y el desarrollo de nuevas soluciones. Este proyecto busca involucrar al agricultor, asesor, investigador y miembros de la industria para que trabajen en colaboración.

- Se ofrece una solución para los problemas relacionados con el almacenamiento, intercambio y procesamiento automático de datos de diferentes tecnologías y fuentes, de forma que el técnico no pierda tiempo en estas tareas y se pueda centrar en el análisis de sus datos procesados como apoyo a la toma de decisiones.
- Con la aplicación del sistema, se pretende crear un nuevo lenguaje entre los agricultores y los técnicos que facilite que puedan trabajar en equipo para disminuir costes, reducir problemas de calidad y, a partir de este cambio, poder realizar una estrategia de marketing sobre los métodos utilizados.
- Existe la posibilidad de avanzar en nuevos modelos de negocio y del surgimiento de nuevos proyectos derivados del análisis de los datos y los posibles servicios vinculados a ellos.

1.4. Objetivos

El objetivo general del proyecto es construir un sistema que resuelva los principales problemas que han surgido al introducir las nuevas tecnologías en el mundo de la agricultura. Para ello, el sistema deberá:

- Unificar los principales sistemas de información existentes en uno, con el fin de ofrecer a los usuarios la posibilidad de estudiar todos los datos de su finca en un solo sitio.
- Adaptarse para poder recuperar datos de los principales fabricantes de sensores y ser compatible así con el mayor número de fincas posible.
- Obtener predicciones climáticas a corto plazo de las fincas incluidas en la base de datos.
- Transformar toda la información recuperada de las distintas fuentes de datos para simplificarla y poder ofrecer a los usuarios una visión más global del estado de sus tierras.
- Combinar los valores tomados por sensores con las predicciones climáticas recuperadas para informar de los eventos que se han producido en el pasado y de los que pueden ocurrir en el futuro próximo.

- Mostrar los resultados de la información simplificada a través de una interfaz web con el fin de que sea accesible desde cualquier dispositivo.

2. Definición del proyecto

Dados todos los antecedentes expuestos anteriormente, se entiende que las nuevas tecnologías no han sido capaces de explotar de manera realmente útil el sector de la agroalimentación, especialmente en términos de análisis de recursos, mejora de la sostenibilidad y monitorización de cultivos para fomentar un aumento de su productividad o incluso evitar riesgos que les afecten.

El presente proyecto pretende desarrollar un sistema de monitorización y diagnóstico para el control y la gestión de cultivos agrícolas. El sistema utilizará datos recogidos por sensores repartidos en los distintos cultivos, realizará cálculos sobre ellos y creará una serie de indicadores avanzados que, con su posterior análisis, permitirán llevar a cabo un control estricto del desarrollo de las plantas. Con esta definición, por tanto, se entiende que el sistema funcionará en 3 fases, que serán llevadas a cabo a su vez por 3 módulos funcionales:

1. Obtención de los datos de distintas fuentes de información y almacenamiento en una base de datos. Esta fase la llevará a cabo el Sistema adaptador de datos.
2. Transformación de los datos en métricas definidas para que puedan ser analizados, que será ejecutada por el Sistema de ejecución de cálculo.
3. Visualización y estudio de los datos mediante una aplicación web.

En los siguientes apartados se describe en detalle cada uno de los módulos que forman el sistema, tanto a nivel funcional como de implementación. No obstante, antes es necesario definir una serie de conceptos importantes en el ámbito del proyecto.

2.1. Definiciones

- **Finca**: terreno de cultivo delimitado que un usuario quiere monitorizar.
- **Parcela**: cada una de las secciones de una finca que el agricultor ha delimitado, ya sea a nivel de registro propio o en un registro catastral.
- **Punto de Seguimiento Detallado (PSD)**: punto geográfico en el que se encuentran los sensores de una determinada parcela. Las parcelas están

formadas por zonas geográficas cuyas características son similares, de forma que las medidas que se toman en un punto, son extrapolables al resto de la parcela.

- **Dendrómetro**: sensor utilizado para medir el ancho del tronco de una planta. Concretamente, se mide la variación del ancho del tronco con respecto a un diámetro inicial. La monitorización de los datos devueltos por este sensor sirve para conocer los ciclos de vida de la planta, su estado fenológico (etapa de crecimiento en la que se encuentra), nivel de estrés, etc.

2.2. Visión global

Para lograr los objetivos marcados por el proyecto, se han definido tres sistemas que funcionarán por separado pero que no serán del todo independientes, ya que deben compartir información a través de la base de datos. Estos sistemas son:

- Sistema adaptador de datos: encargado de la recuperación de datos de sensores y predicciones climáticas y su almacenamiento en la base de datos.
- Sistema de ejecución de cálculos: recuperará los datos obtenidos por el sistema adaptador y los transformará mediante cálculos matemáticos. Además, se encargará de evaluar si se producen una serie de eventos en los cultivos, los cuales se explicarán en detalle en el apartado 4. Los resultados se almacenarán también en la base de datos.
- Aplicación web de visualización de datos: mostrará tanto los resultados de los cálculos como los datos en crudo recuperados por el sistema adaptador.

La complejidad de cada uno de los sistemas anteriormente descritos implica que cada uno de ellos tenga sus propios requisitos funcionales. Dichos requisitos se describen en detalle en los apartados dedicados individualmente a cada uno de los sistemas.

En el siguiente esquema se puede observar el funcionamiento de los distintos módulos que componen el sistema:

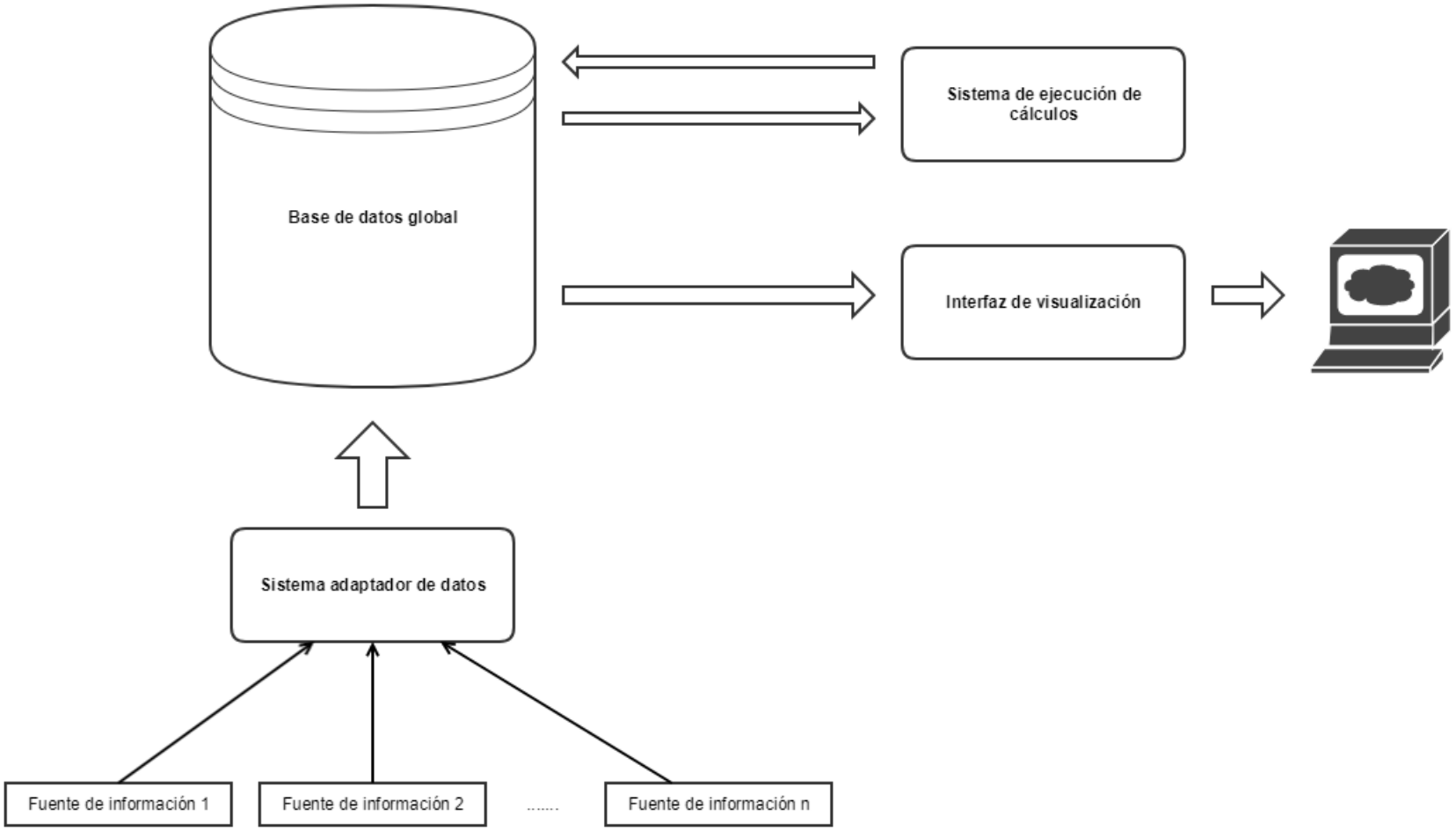


IMAGEN 1: VISIÓN GLOBAL DE LOS MÓDULOS DEL SISTEMA

3. Sistema adaptador de datos

El sistema adaptador de datos es el encargado de recuperar la información de los sensores que están repartidos por las distintas fincas y almacenarla en la base de datos del sistema.

3.1. Requisitos funcionales

Los requisitos funcionales que debe cumplir este sistema son los siguientes:

- Realizar peticiones contra los distintos servidores de datos y recibir la respuesta de los mismos.
- Interpretar los datos recuperados por cada uno de los servidores.
- Insertar los datos recuperados en la base de datos del sistema.
- Convertir los datos a las unidades necesarias para poder realizar operaciones sobre ellos posteriormente.
- Detectar recalibraciones en los sensores en los que se puedan producir y almacenarlas en la base de datos del sistema.
- Gestionar los errores que puedan producirse en la obtención de las medidas de los sensores.

3.1.1. Recuperación de los datos de los servidores.

En este módulo se engloba el conjunto de funciones y ficheros que se encargan de realizar peticiones a los servidores que contienen las medidas tomadas de los cultivos y esperar la respuesta de los mismos. Dado que el sistema permite trabajar con sensores de distintos fabricantes, y cada uno de ellos configura los servidores según su criterio, el sistema debe ser capaz de reconocer a cada fabricante y realizar las peticiones en el formato correcto para cada uno de ellos. Concretamente, se han desarrollado adaptadores para tres fabricantes de sensores y otro para el sistema que realiza las predicciones climáticas.

A continuación, pasamos a describir las especificaciones de cada uno de los fabricantes de los que el sistema es capaz de recuperar información.

3.1.1.1. ADCON TELEMETRY

Esta es la plataforma con más experiencia y equipos instalados por todo el mundo. ADCON ofrece una amplia gama de innovadoras RTU's (Unidades de Telemetría Remota o datalogger) que recogen y transmiten datos de sensores a través de redes con escalabilidad ilimitada, gestión totalmente automática mediante el Telemetry Gateway y de fácil manejo. Las estaciones se comunican unas con otras por radio o GPRS y transmiten los datos a la estación base. Automáticamente, los datos son transmitidos desde la memoria de la estación central al servidor donde está instalado el software SCADA addVANTAGE de ADCON (ADCON Telemetry) (SCA17), que los almacena y permite realizar después consultas sobre dichos datos. Este software implementa el protocolo addUPI bajo una pila TCP/IP. Para obtener los datos, un cliente puede realizar peticiones HTTP mediante el método GET y el servidor genera respuestas en forma de documentos XML. El proceso de comunicación con un servidor de ADCON debe seguir los siguientes pasos:

1. El cliente se autentifica frente al servidor enviando su usuario y contraseña.
2. El servidor devuelve un XML con un identificador de usuario.
3. El cliente realiza peticiones para cada uno de los sensores, utilizando el identificador de sesión que le ha devuelto el servidor para permanecer autentificado.
4. El servidor recibe las peticiones del cliente, comprueba en cada una que el ID de la sesión no ha caducado y devuelve los datos solicitados en formato XML.
5. Cuando el cliente termina de realizar peticiones de datos, debe cerrar la sesión con el servidor para que el identificador de sesión quede invalidado. Si el cliente está un tiempo limitado sin enviar peticiones al servidor, éste invalida la sesión automáticamente.

Las peticiones que se realizan a los servidores de ADCON deben seguir el siguiente formato genérico:

http://hostname:port/addUPI=function=fn&session-id=nnnn&param1=p1&...paramn=pn

Donde:

- hostname: dirección IP del servidor donde se ejecuta el software de addVANTAGE.
- port: número de puerto correspondiente al software dentro del servidor.
- function: nombre del método invocado.
- sesión-id: identificador de sesión recibido tras el proceso de login.
- param<i>=pn: parámetros específicos para el método que se está llamando.

Los métodos que se utilizan para la recuperación de los datos son:

- login: método de identificación de sesión del cliente en el servidor. Es necesario especificar los parámetros "user" y "passwd" con usuario y contraseña del cliente. Ejemplo:

  http://XXX.XXX.XXX.XXX:8080/addUPI?function=login&user=**&passwd=**

  La respuesta que devolverá el servidor, en el caso de que los datos de identificación del cliente sean correctos, tiene el siguiente formato:

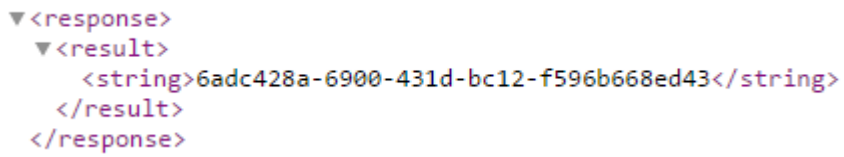


IMAGEN 2: RESPUESTA DE ADCON TRAS LOGIN CORRECTO

- getdata: método para obtener los datos registrados por un sensor, tomados cada 15 minutos. Los parámetros necesarios para este método son:
  - session-id: identificador de sesión obtenido tras realizar el login en el servidor.
  - id: identificador del sensor cuyos datos se quieren recuperar.
  - date: fecha inicial a partir de la cual se quieren recibir los datos del servidor. Debe estar en el formato "yyyymmddThh:mm:ss".
  - slots: número de medidas que se desean recuperar desde la fecha inicial.

  Un ejemplo de una llamada a este método para obtener 9 valores del sensor 291 el día 05/04/2016 sería:

http://XXX.XXX.XXX.XXX:8080/addUPI?function=getdata&session-id=6adc428a-6900-431d-bc12-f596b668ed43&id=291&date=20160405T00:00:00&slots=9

El formato de la respuesta devuelta por el servidor es:

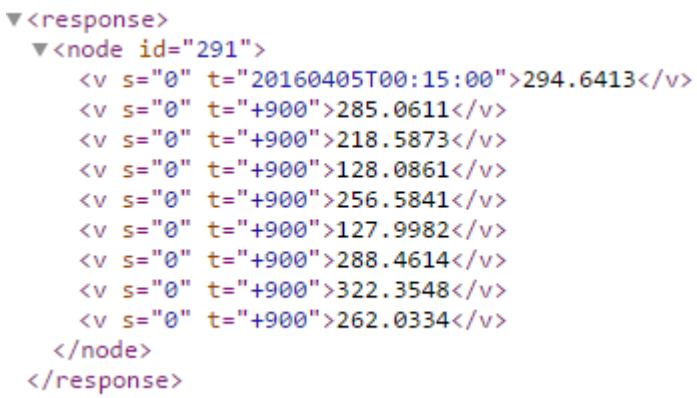


IMAGEN 3:RESPUESTA DE ADCON TRAS PETICIÓN DE DATOS

Se puede observar que cada etiqueta <v> del XML contiene una medida del sensor. Además, cada etiqueta cuenta con otros dos atributos:

- "s": representa el "status" de la medida. Tendrá valor 0 si la medida es correcta o distinto de 0 si el servidor ha detectado algún error a la hora de recuperar la medida concreta.
- "t": indica el tiempo en segundos que ha pasado entre esta medida y la actual.

- logout: cierra la sesión del cliente en el servidor. Solo necesita que se le pase como parámetro el ID de la sesión abierta. Por ejemplo:

  http://XXX.XXX.XXX.XXX:8080/addUPI?function=logout&session-id==6adc428a-6900-431d-bc12-f596b668ed43

La respuesta en este caso está vacía:

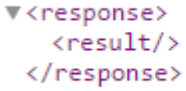


IMAGEN 4: RESPUESTA DE ADCON TRAS LOGOUT

3.1.1.2. NAZARIES

El sistema de NAZARIES (CERES) es también remoto y multiplataforma, lo cual permite que los datos puedan ser solicitados desde cualquier dispositivo a través de un

navegador web. El dispositivo electrónico que registra los datos en tiempo real recibe el nombre de “datalogger” y toma la ubicación por medio de instrumentos y sensores propios o externos.

La API implementada por este servicio devuelve los datos en un archivo XML, en el que cada entrada corresponde a una medida del sensor tomada cada 15 minutos. La URL utilizada para realizar peticiones al servicio será similar a:

http://XXX.XXX.XXX.XXX/api/add_uppi/node/NNNN/date/yyyy,mm,dd,hh,mm,ss.xml

Donde:

- XXX.XXX.XXX.XXX: corresponde a la IP del servidor donde se aloja el software que devuelve los datos.
- NNNN: es el ID del sensor que se solicita dentro de la base de datos del sistema.
- yyyy: año de la fecha en que se quieren pedir datos.
- mm: número del mes del que se quieren recuperar datos, de 1 a 12.
- dd: número del día del que se quieren recuperar datos, de 1 a 31.
- hh: hora a la que se quiere empezar a pedir datos, de 0 a 23.
- mm: minuto a partir del cual se quieren recuperar datos, de 0 a 59.
- ss: segundo a partir del cual se quieren recuperar datos, de 0 a 59.

Como se puede observar en este formato de URL no se especifica el número de medidas que se quieren obtener ni la hora de fin del período de tiempo que se quiere recuperar. Lo que hace este sistema es asumir como fecha de fin del período la misma hora que pone en la petición, pero del día siguiente. De esta forma, si en la petición se pone la fecha 15/09/2016 12:30:00, el sistema devolverá medidas desde esa hora (incluida) hasta el 16/09/2016 12:30:00 (no incluida).

La respuesta devuelta por el servidor si se hace una petición correcta es:

```
<VerdtechXML>
    <Nodo id="60015">
        <Atributos>
            <a nombre="name" tipo="string">Temperatura Ambiente</a>
            <a nombre="EUID" tipo="int">2710</a>
            <a nombre="maxValue" tipo="double">100.0</a>
            <a nombre="minValue" tipo="double">-20.0</a>
        </Atributos>
        <Datos>
            <valor fecha="20160915T00:00:00" status="2">0.0</valor>
            <valor fecha="20160915T00:15:00" status="2">0.0</valor>
            <valor fecha="20160915T00:30:00" status="0">28.4</valor>
            ...
            <valor fecha="20160915T23:45:00" status="2">0.0</valor>
        </Datos>
    </Nodo>
</VerdtechXML>
```

IMAGEN 5: RESPUESTA DE NAZARIES TRAS PETICIÓN DE DATOS

Donde se pueden observar dos grupos de datos:

- Atributos: se devuelve información del sensor del que se han solicitado los datos. Cada etiqueta “<a>” representa un atributo distinto, y en ellos se puede ver la magnitud medida por el sensor, el valor máximo y el mínimo histórico.
- Datos: contiene la lista de datos que hay almacenados para el sensor. Cada etiqueta “<valor>” contiene un dato, y dentro de la etiqueta se pueden ver dos atributos:
    - Fecha: indica el momento en que se tomó la medida. Viene en formato “YYYYMMDDThh:mm:ss”.
    - Status: indica el estado de la medida correspondiente. Si tiene valor 0, significa que la medida es considerada correcta, pero si tiene valor distinto de 0, significa que ha habido algún error y no se puede recuperar el valor.

3.1.1.3. RANCH SYSTEMS

Los sistemas de RANCH SYSTEMS utilizan un software basado en un servicio web que permite recuperar datos en tiempo real de su base de datos en formato JSON. El acceso a los datos se realiza de forma segura mediante SSL y autenticación mediante usuario y contraseña.

La implementación del servicio web es mediante una API RESTful, por lo que utiliza peticiones HTTP, tales como GET Y POST, y es independiente del estado (toda petición puede ser ejecutada en cualquier orden y devolverá los mismos resultados exceptuando la naturaleza dinámica de los datos de los sensores).

Las URLs necesarias para hacer las peticiones de datos a los sistemas de RANCH SYSTEMS deben seguir el formato:

https://XXX.XXX.XXX.XXX/rsapp15/jsp/rsjson.jsf?uname=**&pword=**&reqtype=**

Donde:

- uname: nombre de usuario con permisos para acceder a los datos.
- pword: contraseña con la que se identifica el usuario en el sistema.
- reqtype: tipo de petición que se hace al sistema.

Además, dependiendo de la petición que se realice, será necesario añadir más parámetros a la URL. El sistema de RANCH SYSTEMS permite hace peticiones que devuelvan tanto medidas tomadas por los sensores, como información geolocalizada sobre la finca correspondiente (como la lista de parcelas o las zonas agregadas del terreno). Dependiendo de la petición que se realice, será necesario añadir más parámetros a la URL. Ya que para el desarrollo del sistema solo se ha precisado recuperar las medidas tomadas por los sensores, la única petición que se realiza es de tipo “data”, y tiene los siguientes parámetros:

- rmsids: lista de identificadores de los sensores de los que se quieren recuperar los valores, separados por comas.
- from: fecha inicial del intervalo de tiempo en el que se quieren recuperar datos. Debe estar en formato UNIX en milisegundos.
- to: fecha final del intervalo de tiempo en el que se quieren recuperar datos. Debe estar en formato UNIX en milisegundos.

Con esto, la URL para realizar una petición de datos entre, por ejemplo, el 17/07/2017 a las 00:00:00 y el 18/07/2017 a las 00:00:00 será:

https://XXX.XXX.XXX.XXX/rsapp15/jsp/rsjson.jsf?uname=****&pword=*****&reqtype=data&rmsids=YYYY&from=1468713600000&to=1468800000000

La respuesta JSON devuelta por el servidor tiene los siguientes atributos:

- result: indica si la petición se ha podido resolver correctamente con “OK” o si no ha sido así, con “ERROR”.
- auth: indica si la petición ha sido aceptada (“TRUE”) o no (“FALSE”).
- data: array con los datos de los sensores que se han consultado. Cada elemento del array es un objeto JSON con los siguientes campos:
    - id: identificador del sensor en el sistema.
    - rmsdata: array con las medidas recuperadas para dicho sensor. Cada elemento del array tiene dos atributos:
        - x: fecha de la medida en formato UNIX en milisegundos
        - y: valor de la medida.

En la siguiente imagen se puede observar una respuesta para una petición de este tipo:

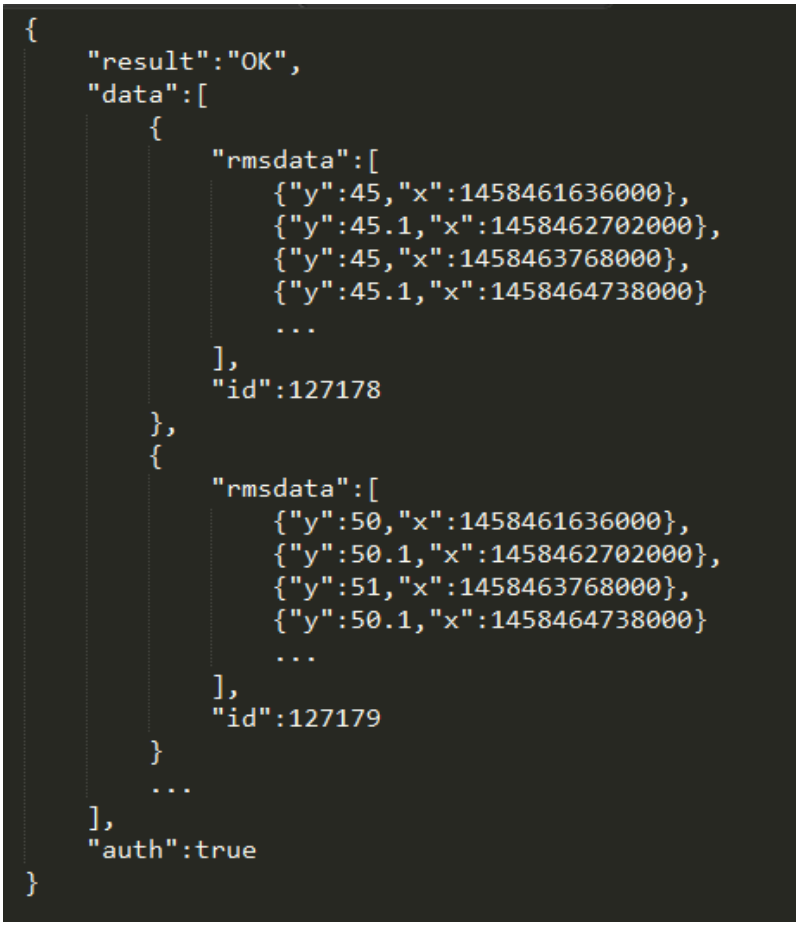


IMAGEN 6: RESPUESTA DE RANCH SYSTEMS TRAS PETICIÓN DE DATOS

3.1.1.4. METEOGRID

Es la plataforma encargada de realizar predicciones climáticas personalizadas basadas en los datos recogidos por las estaciones de clima repartidas por los distintos campos de cultivo. Este sistema, desarrollado por la Fundación para la Investigación del Clima (FIC) (Fundación para la Investigación del Clima), combina técnicas de modelización numérica con aproximaciones estadísticas que habitualmente se apoyan de registros climáticos.

Los parámetros climáticos que el sistema es capaz de predecir son:

- Temperatura
- Humedad relativa
- Precipitaciones
- Velocidad del viento
- Radiación

La URL a la que hay que acceder para realizar peticiones al servicio es:

https://sigym3.com/users/******/preds/predictions?

Habría que sustituir los asteriscos por un nombre de usuario válido para el servidor. Además, hay que incluir los siguientes parámetros:

- sgformat: formato en el que se quieren recuperar los datos. Admite los valores XML, JSON y HTML. Si no se especifica nada, devuelve los datos en una tabla HTML.
- time: rango de fechas para las que se quieren recuperar los datos. Deben ser dos fechas (inicio y fin) separadas por una coma o por "%2C" (equivalente en hexadecimal de la coma en codificación ASCII). El formato que deben tener las fechas es "yyyy-mm-ddThh:mmZ", donde lo que hay después de la "T" es la hora correspondiente y la "Z" del final indica que se trata de hora zulú.
- Type: lista de parámetros que se quieren recuperar del servidor separados por comas. Si no se especifica ningún valor, el servidor devuelve los datos de todos los parámetros que tenga disponibles. En la siguiente tabla se puede ver el nombre con el que hay que realizar la petición para cada uno de los parámetros que se pidan al servicio.

Temperatura	temperatura_combi_corr
Humedad Relativa	humedad_combi_corr
Precipitaciones	precipitacion
Velocidad del viento	viento_mod_combi_corr
Radiación	radiacion_horaria

TABLA 1: RELACIÓN DE VARIABLES CLIMÁTICAS CON NOMBRE PARA PEDIR PREDICCIÓN

- Name: nombres de las estaciones climáticas de las que se quieren obtener las predicciones, separados por comas. Si no se asigna ningún valor, el servicio devuelve los datos de todas las estaciones climáticas que tenga el usuario que realiza la petición.
- AUTH_TYPE: tipo de autentificación con la que se va a realizar la petición. Cuando se hace una petición desde el navegador sin estar autentificado previamente, el sistema redirige a una página de login en la que hay que identificarse mediante usuario y contraseña. Como el sistema adaptador no funciona mediante navegador sino mediante scripts de NodeJS, hay que asignar a este parámetro el valor “basic” y especificar las credenciales de autentificación con cada petición de la siguiente forma:

```
var url = "https://sigym3.com/users/******/preds/predictions?AUTH_TYPE=basic&" +
  "time=2014-12-01T00%3A00Z%2C2014-12-02T03%3A00Z&type=ET0&sgformat=json";
request({
  url: url,
  auth: {
    'user': username,
    'pass': password
  }
});
```

IMAGEN 7: EJEMPLO DE PETICIÓN DE DATOS DE PREDICCIONES METEOROLÓGICAS

3.1.2. Conversión a unidades estándar

Este módulo del sistema se encarga de pasar los datos de la tabla “raw_tag_data” a la tabla “raw_tag_standard_units_data”, pasándolos previamente a unidades estándar. Se ha dado el nombre de “unidades estándar” a las unidades en las que deben estar los datos de los sensores para poder realizar los cálculos que ejecutará posteriormente el sistema.

El requisito que debe cumplir la conversión de unidades que se realiza es que debe ser calculable mediante la aplicación de una función lineal sobre los datos en crudo, es decir, que se pueda calcular mediante el producto y la suma de dos factores al valor original de la medida.

3.1.3. Detección de recalibraciones.

Este módulo cumple la función de detectar saltos espurios entre dos valores consecutivos devueltos por los sensores encargados de medir el diámetro del tronco de las plantas (llamados dendrómetros). Se realiza esta comprobación porque no es habitual que en el intervalo de tiempo entre dos medidas (que suele ser de 15 minutos) la diferencia de valores sea muy elevada, por lo que suele deberse a una modificación manual del dendrómetro, lo que se conoce como salto indeseado. El objetivo de este módulo es detectar dichos saltos y hacer que se ignoren, para que no se confundan con crecimiento o decrecimiento reales de la planta. En la sección 3.2.5. se explica en detalle cómo se realiza este proceso.

3.1.4. Gestión de errores.

En los procesos que tienen lugar en servidores, especialmente si se comunican con otros servidores por red, es importante llevar un control de los errores que puedan producirse durante la ejecución. En el caso del sistema adaptador, se pueden dar dos situaciones que provoquen algún tipo de error: el sistema no puede conectar con alguno de los servidores de datos, o se produce algún fallo intentando recuperar los datos de un sensor determinado.

La consecuencia de cualquiera de estos errores es que habrá “huecos” en los datos del sistema de almacenamiento. Para corregir este problema, el sistema deberá llevar un registro de los errores que se produzcan y ejecutar periódicamente un proceso que intente recuperar y corregir los datos que falten en la base de datos o cuyo valor no sea correcto.

El diagrama de flujo simplificado de este proceso de corrección de errores se muestra en la siguiente figura:

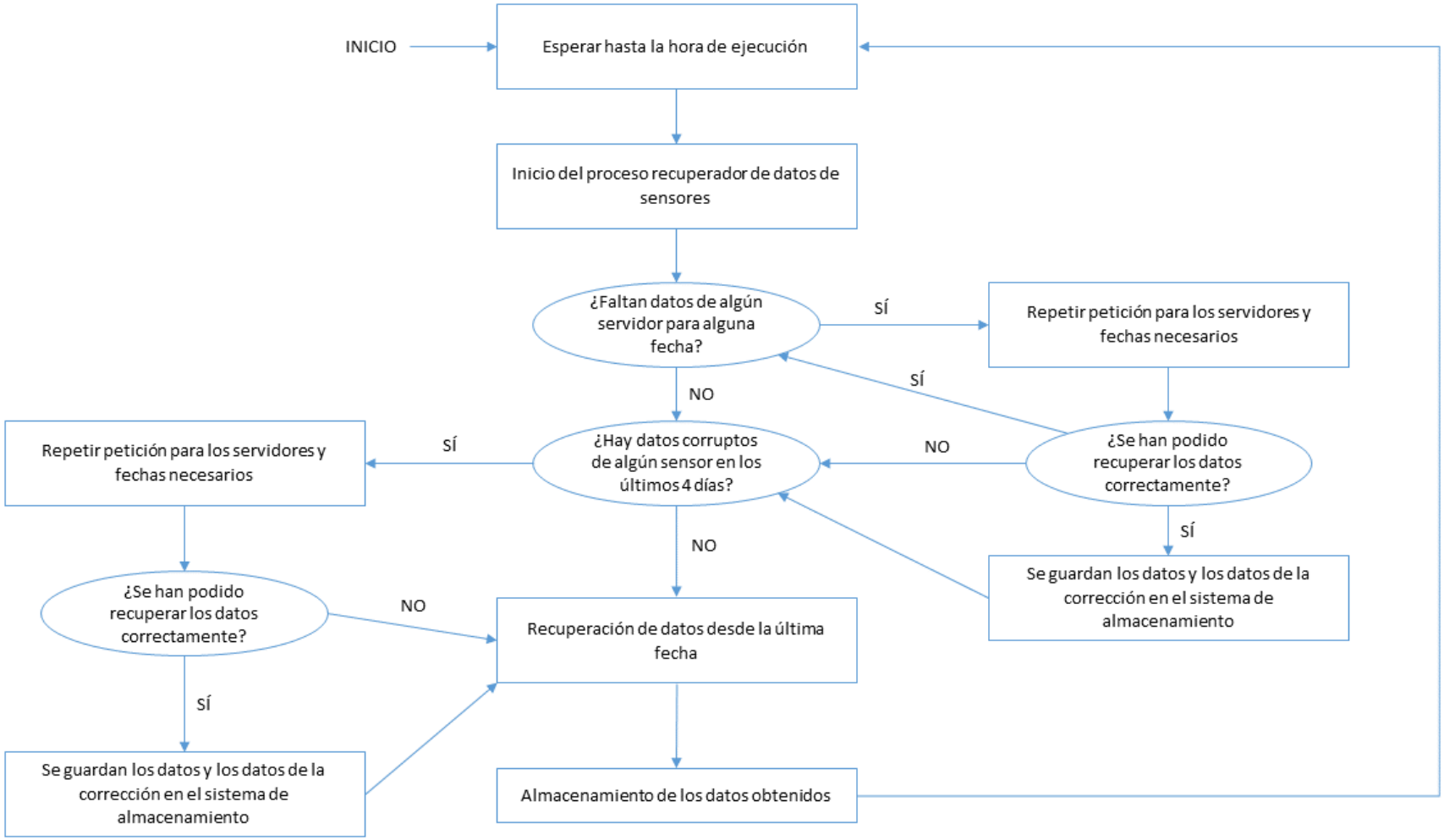


IMAGEN 8: DIAGRAMA DE FLUJO DE SISTEMA DE CORRECCIÓN DE ERRORES

3.2. Arquitectura del sistema

El sistema de recuperación de datos está implementado en un servidor de NodeJS (Stefanov, 2010) que se encarga de llamar periódicamente a los distintos módulos encargados de la obtención de medidas de los sensores y de la comprobación de errores producidos en las ejecuciones anteriores.

Teniendo en cuenta los requisitos que debía satisfacer el sistema adaptador, descritos a lo largo de la sección 3.1., se han definido distintos módulos funcionales, los cuales se explican en los siguientes apartados.

3.2.1. Sistema de recuperación de errores de servidor

Este módulo se encarga de comprobar si hay algún error de servidor pendiente de resolver. Se produce un error de servidor cuando el sistema recuperador de datos no se ejecuta correctamente o no puede conectar con alguno de los servidores de datos. En caso de error, se introduce una entrada en la tabla "server_error_logs" de la base de datos. Esta tabla tiene 4 campos de tipo booleano que indican hasta qué punto se ha podido solucionar un error de servidor. Estos campos son:

- error_solved_raw: corresponde a la fase de obtención de datos del servidor.
- error_solved_raw_standard_units: corresponde a la fase de conversión a unidades estándar.
- error_solved_calculations: corresponde a la fase de ejecución de cálculos.
- error_solved_events: corresponde a la fase de evaluación de eventos.

Cuando se detecta que el sistema adaptador no puede conectar con un servidor, se introduce una nueva entrada en la tabla con los 4 campos a FALSE. En la siguiente ejecución del sistema adaptador, el módulo de recuperación de errores de servidor buscará en la base de datos las entradas de la tabla con los 4 campos a FALSE, y para cada uno de los errores que haya:

1. Se mira la fecha del error y el servidor al que corresponde.
2. Se piden los datos al servidor de datos correspondiente para la fecha del error.
3. Si se consiguen recuperar los datos, se insertan en la base de datos y se actualiza el registro en la tabla "server_error_logs" con el campo "error_solved_raw" a TRUE.

4. Si no se consiguen recuperar los datos, se espera a recuperar el error en la siguiente ejecución del sistema adaptador.

3.2.2. Sistema de recuperación de errores de sensor

Se entiende por "error de sensor" aquel que viene indicado en uno de los datos devueltos por un servidor de datos. Como se ha explicado en la especificación funcional de cada uno de los fabricantes (sección 3.1.1.), es posible que un sensor indique que, por algún motivo, una medida que devuelve es incorrecta. En la siguiente imagen se puede ver un ejemplo de una medida incorrecta devuelta por un servidor de ADCON:

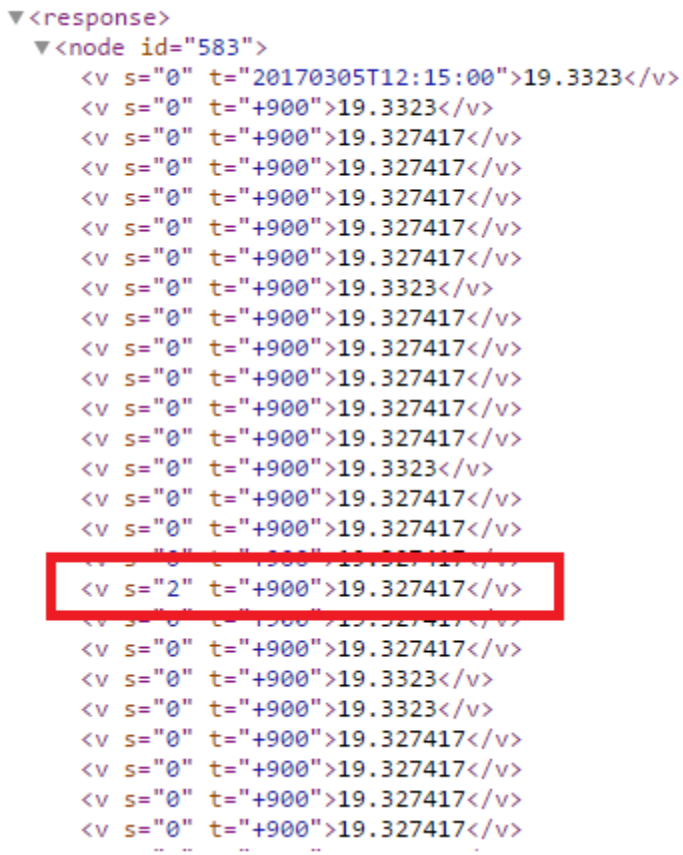


IMAGEN 9: EJEMPLO DE DATO INCORRECTO EN RESPUESTA DE ADCON

Cuando un sensor devuelve alguna medida incorrecta, no se guarda el valor que devuelve ya que no tiene ninguna utilidad y puede dar lugar a tomas de decisiones erróneas. En lugar de esto, se almacena una entrada con valor nulo ("NULL") en la tabla de medidas y se introduce un registro en la tabla "tag_error_logs", que tiene 4 campos iguales que los explicados para la tabla "server_error_logs".

Debido a la pequeña memoria de los sensores, solo tienen capacidad para almacenar los datos de 4 días. Esto quiere decir que, si en 4 días no se ha podido resolver un error de sensor, no tiene sentido seguir intentándolo porque el sensor correspondiente ya habrá borrado la medida correcta de su memoria y no será capaz de recuperarla.

En las pruebas realizadas, se ha comprobado que los errores de sensor no suelen ser aislados, sino que es bastante habitual encontrar varios errores de sensor consecutivos. Por esto, se ha tomado la decisión de, por cada error de sensor en la base de datos, pedir todas las medidas del día en que se produjo el error por si se diera el caso de que dicho sensor tuviera más errores en el mismo día.

Teniendo todo esto en cuenta, el proceso consta de los siguientes pasos:

1. Se busca en la base de datos los errores de sensor cuya fecha se encuentre entre los últimos 4 días y con los 4 campos de booleanos de la tabla "tag_error_logs" a "FALSE".
2. Para cada error, se mira el sensor al que corresponde y su fabricante.
3. Se piden todas las medidas disponibles del sensor para el día en que se produjo el error.
4. Se miran los errores que haya para ese sensor y esa fecha y se corrigen todos los que se pueda con los datos recuperados del servidor.
5. Si se ha podido corregir algún error, se actualiza el registro correspondiente en la tabla "tag_error_logs" poniendo el campo "error_solved_raw" a "TRUE".

3.2.3. Sistema de recuperación de datos

Se trata del módulo principal del sistema adaptador, encargado de obtener las medidas de todos los sensores almacenados en la base de datos del sistema. Cada vez que se ejecuta, realiza los siguientes pasos:

1. Obtiene la lista de servidores del sistema.
2. Para cada sensor, se comprueba la última fecha para la que se han obtenido medidas y se guarda esta fecha como inicio del período de tiempo en el que se quieren recuperar datos. La fecha final del período será la fecha en que se está ejecutando el sistema. En caso de que no hubiera ninguna medida para el

servidor correspondiente, se recuperarán los datos de los últimos N días, siendo N una constante que inicialmente tiene valor 70.

3. Se comprueba el fabricante del servidor y se llama a las funciones de recuperación de errores implementadas para dicho fabricante.
4. El sistema debe gestionar, además, los errores devueltos por cada servidor o por cada sensor del servidor, introduciendo los registros correspondientes en las tablas “server_error_logs” o “tag_error_logs”.

En el siguiente diagrama se puede observar el funcionamiento del sistema recuperador de datos:

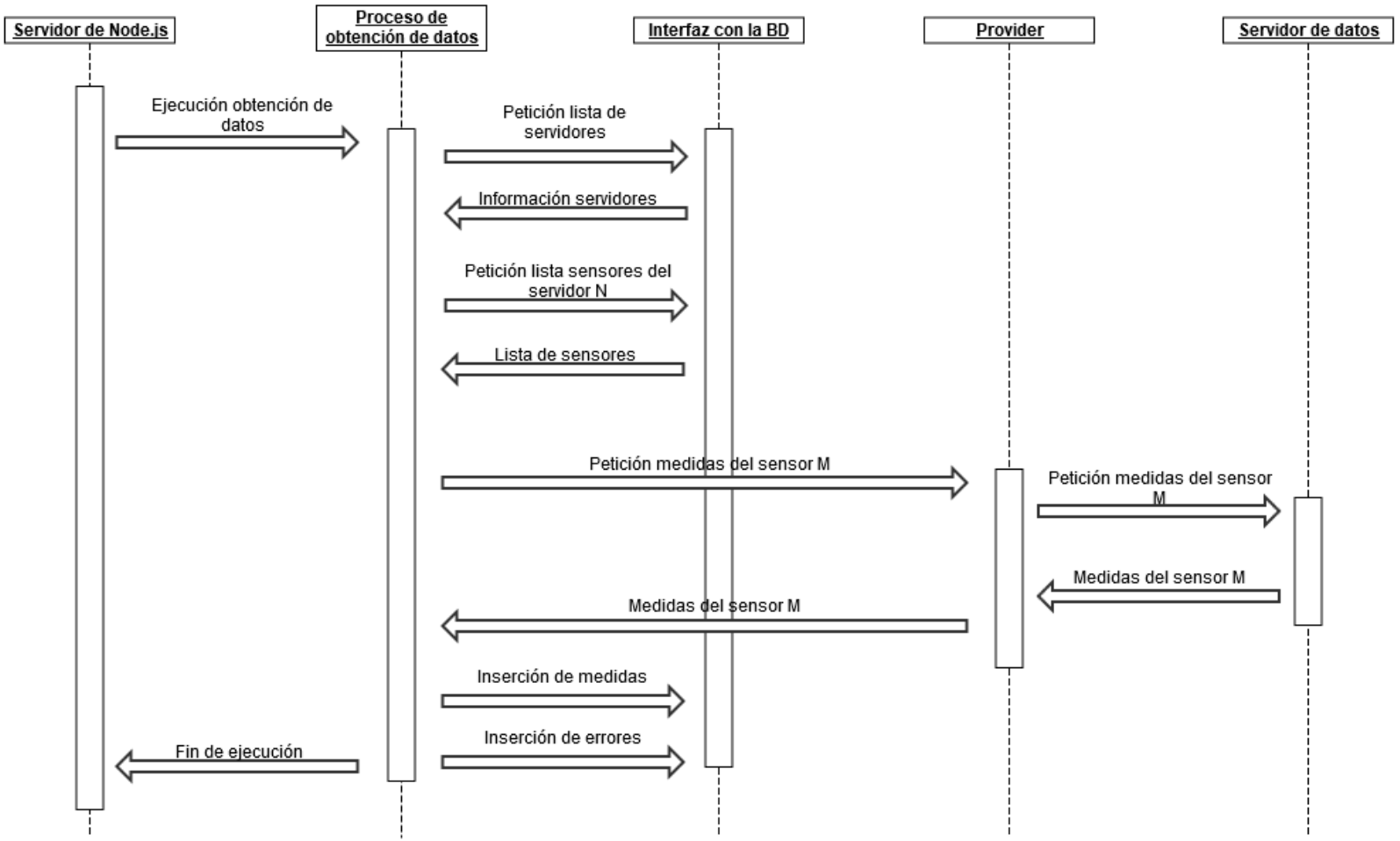


IMAGEN 10: DIAGRAMA DE FLUJO DEL SISTEMA RECUPERADOR DE DATOS

3.2.4. Sistema de conversión a unidades estándar

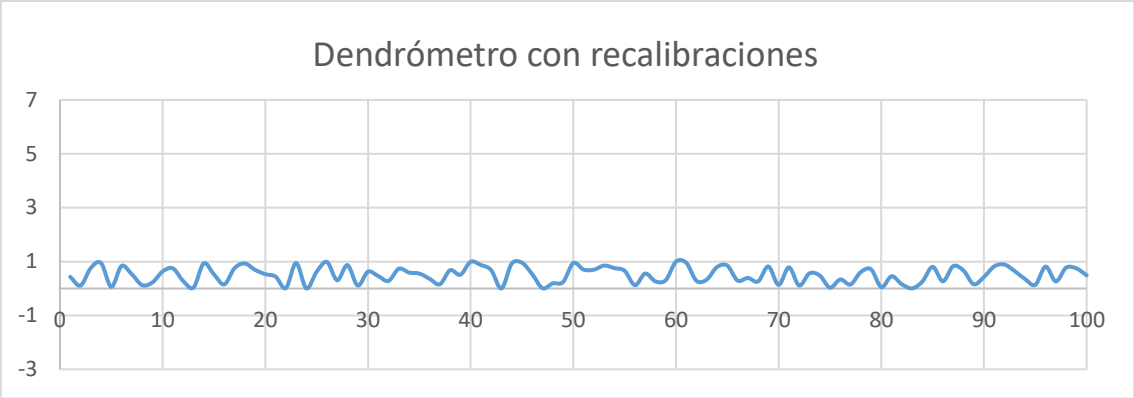
Este procedimiento se lleva a cabo utilizando la información almacenada en la tabla de unidades “units”. En cada entrada de esta tabla se indica la unidad en la que mide un sensor y las operaciones que hay que realizar para pasar de una medida en esa unidad a la unidad estándar que le corresponda. Estas operaciones se indican con los campos “factorA” y “factorB” aplicando la siguiente fórmula:

$$Medida\ en\ unidad\ estándar = medida\ en\ crudo * factorA + factorB$$

Lo que se hace en este proceso es coger las medidas que se han insertado en la tabla “raw_tag_data” durante la recuperación de datos, realizar la conversión a unidades estándar e insertarlas en la tabla “raw_tag_standard_units_data”. Esto se hace con todas las medidas salvo con las tomadas por dendrómetros, que requieren un tratamiento especial que se estudiará a continuación.

3.2.5. Detección de recalibraciones

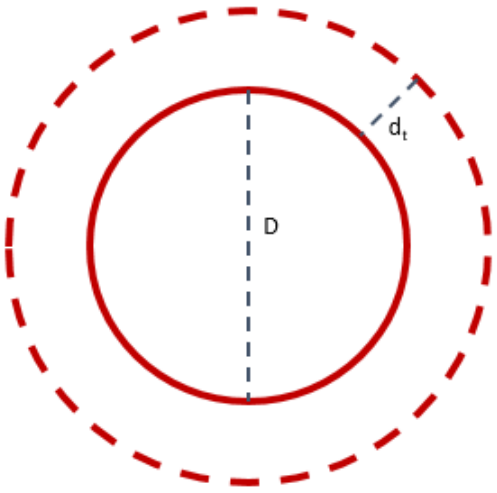
Este sistema se encarga de recorrer las medidas de dendrómetros ordenadas de forma temporal. Cuando se detecta una diferencia entre dos valores consecutivos superior a un umbral definido, se inserta una entrada en la tabla “recalibrations” con el valor del salto, la fecha y hora donde se ha detectado y el sensor al que corresponde. A las medidas posteriores al salto, se le restará el valor del salto detectado, con el fin de obviar dicho salto a la hora de estudiar los valores de las medidas de dendrómetro. Cuando hay más de un salto en la tabla “recalibrations” para un sensor, el valor que se le resta a los valores posteriores a las recalibraciones es la suma de todas las recalibraciones detectadas. En la siguiente imagen se puede observar la diferencia entre una señal de dendrómetro original y otra aplicando recalibraciones:



Se han definido dos tipos de recalibraciones que se pueden insertar en la tabla "recalibrations":

- Automática: en este grupo estarán las recalibraciones insertadas como consecuencia de la detección automática de un salto en los datos recuperados por el sistema adaptador.
- Manual: serán de este tipo las recalibraciones que se inserten de forma manual en la base de datos.

El objetivo de las recalibraciones manuales es insertar el valor del ancho que tiene el tronco en el momento de colocar cada sensor en su planta correspondiente. De esta forma, dado que el dendrómetro mide las variaciones en el tronco a partir de un ancho determinado, cuando el sistema de recalibraciones aplique la recalibración manual sobre las medidas del tronco, lo que se insertará en la tabla "raw_tag_standard_units_data" será el diámetro del tronco en cada momento.



- D = diámetro inicial del tronco, que se inserta como una recalibración manual.
- d_t = incremento del tronco en el momento t, medido por un dendrómetro.
- D + d_t = ancho del tronco en el momento t, valor que se inserta en raw_tag_standard_units_data.

IMAGEN 11: EJEMPLO DE MEDICIÓN DE DENDRÓMETRO

El sistema solo tiene en cuenta las recalibraciones almacenadas en la base de datos desde la última recalibración manual insertada (la cual también se tiene en cuenta). De esta forma, si se detectase algún error en las recalibraciones que se aplican, bastaría con insertar una recalibración manual con el valor del ancho del tronco en ese momento para dejar de tener en cuenta las recalibraciones detectadas hasta ese momento.

3.2.6. Interfaz con la base de datos

Este sistema engloba los módulos que permiten interactuar con la base de datos, tanto para obtener la información de los servidores que se necesitan para realizar peticiones como para insertar los datos que se recuperan. Para evitar tener que realizar esta interacción con la base de datos utilizando código SQL puro, se utiliza un ORM como capa de abstracción sobre la base de datos. Un ORM (llamado así por las siglas de *Object-Relational Mapping*) es un modelo de programación que permite interactuar con las tablas de una base de datos manejando entidades que simplifiquen las tareas de acceso básico. Dado que el lenguaje utilizado para toda la implementación del sistema adaptador ha sido JavaScript, el ORM elegido es BookshelfJS, una librería basada en promesas (estilo de programación asíncrona en JavaScript) que permite la configuración de distintos accesos a bases de datos. Los archivos que agrupan las distintas tareas de interacción con la base de datos están en los paquetes “DB” y “models” y se describen a continuación:

- ‘DB/knex_config.js’: archivo con las configuraciones de los accesos para las distintas bases de datos.
- ‘DB/db_interface.js’: contiene las funciones utilizadas para la consulta e inserción de información en la base de datos.
- ‘models’: en este paquete se incluyen el conjunto de archivos utilizados para definir modelos del ORM. Cada modelo se debe corresponder con una tabla de la base de datos.

3.2.7. Providers

Se ha dado el nombre de “provider” a un conjunto de funciones que se utilizan para realizar peticiones a un servidor de datos de sensores. Como se ha explicado

anteriormente, la forma de solicitar datos a cada servidor cambia en función de su fabricante, por lo que es necesario crear un proveedor distinto para cada fabricante que se dé de alta en el sistema. Dentro del conjunto de *providers* se pueden diferenciar dos grupos:

- Providers de datos de predicciones meteorológicas, que inicialmente solo estará compuesto por el archivo "meteogridProvider.js".
- Providers de medidas de sensores, que estará formado por "adconProvider.js", "nazariesProvider.js" y "ranchProvider.js".

En cada uno de los *providers* de medidas de sensores, debe haber dos funciones definidas, que serán llamadas desde el programa principal del sistema adaptador.

- executeDataLoadFromServer(): ejecuta el conjunto de instrucciones necesarias para realizar el proceso de obtención de datos del servidor y su almacenamiento en la base de datos.
- recoverTagErrorData(): conjunto de instrucciones para recuperar los datos de un error de sensor y almacenar el valor correcto en la base de datos si se pudiera.

Dado que los datos de los sensores se recolectan en formatos diferentes, es necesario que cada provider cuente con un proceso de interpretación de dichos datos y transformación a un formato común que pueda ser utilizado por las funciones de la interfaz con la base de datos para ser almacenados. Para este formato se utilizarán un objeto de tipo JSON que debe tener los siguientes campos:

- tag_code: identificador del sensor dentro de su servidor de datos.
- date: fecha a la que corresponde la medida tomada por el sensor.
- value: valor de la medida.
- status: indicador de error en el valor de la medida devuelta por el sensor.

En la siguiente imagen se puede observar la conversión de unos datos devueltos por un servidor de Nazaríes al formato estándar definido:

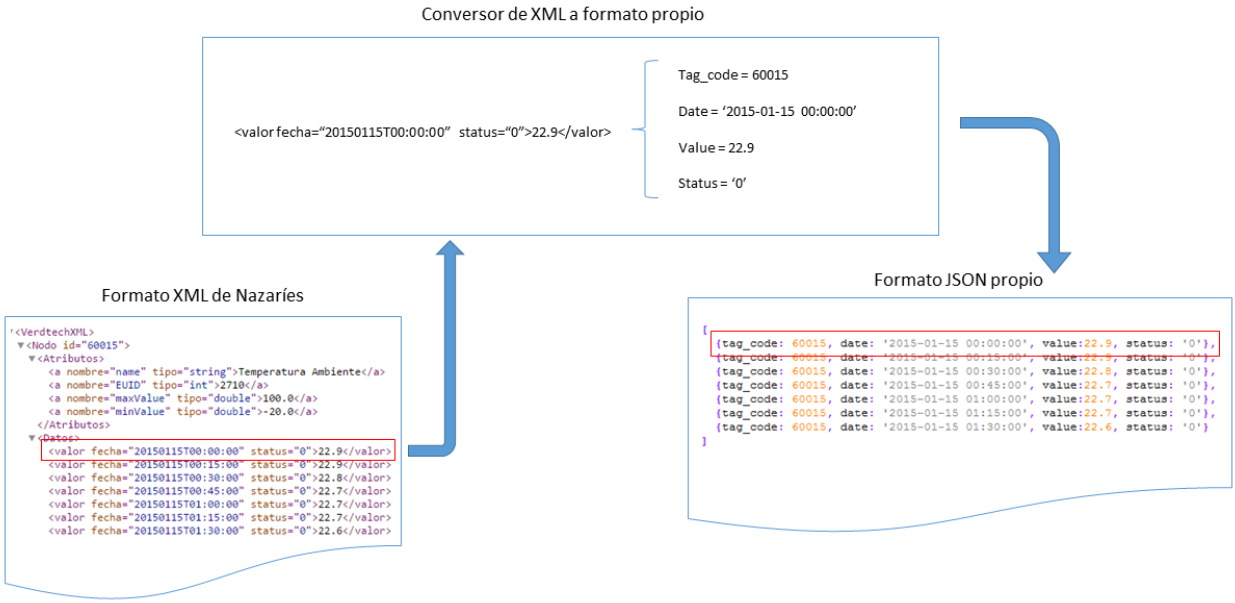


IMAGEN 12: EJEMPLO DE CONVERSIÓN DE DATOS DE NAZARIES A FORMATO JSON

4. Sistema de ejecución de cálculos

El sistema que lleva a cabo la ejecución de los cálculos es el encargado de interactuar con la información recuperada por el sistema adaptador. Su labor es realizar distintos tipos de transformaciones sobre los datos en crudo para convertirlos en métricas válidas que el usuario pueda interpretar y que le permitan sacar conclusiones sobre el estado de sus cultivos.

Esta parte del sistema es consecuencia de la complejidad y el volumen de los datos recogidos, que hacen que sea prácticamente imposible para un usuario poder analizar toda la información que recibe de forma óptima. Mediante el análisis de las necesidades de los cultivos, se han podido identificar una serie de patrones que ponen en peligro la estabilidad de la planta. A partir de estos patrones, se han definido una serie de situaciones de riesgo que se deben evitar, y también un conjunto de acciones que un usuario puede llevar a cabo para corregir dichas situaciones.

4.1. Requisitos funcionales

El método desarrollado consiste en combinar la información de distintos puntos de monitorización mediante sensores de planta, clima y suelo representativos en áreas definidas de cada finca. De esta forma, se dispone de factores que se pueden controlar (agua en el suelo y nutrición) y factores que no se puede controlar (clima), que se pueden combinar con el fin de controlar el comportamiento de la planta. El objetivo será diseñar y definir distintos eventos basados en la experiencia de cada usuario vinculados a los objetivos de eficiencia de cada finca.

4.1.1. Subsistema de modelos

Este módulo es el encargado de realizar distintas transformaciones sobre los datos en crudo obtenidos por el sistema adaptador y que luego serán analizados para la evaluación de eventos. Los cálculos llevados a cabo por los modelos pueden realizarse a dos niveles: a nivel de PSD o de árbol. Esta diferenciación modifica los períodos de tiempo que se tienen en cuenta a la hora de llevar a cabo el cálculo correspondiente.

- En los cálculos a nivel de PSD, el período de tiempo que se tiene en cuenta es un día natural, desde las 00:00h hasta las 23:59h. Este tipo de cálculo es el que se

utiliza principalmente en métricas de clima y suelo, al calcular valores como la temperatura media de un día, la lluvia total de un día o la humedad máxima del suelo.

- En los cálculos a nivel de árbol, el inicio y fin del período de cálculos viene marcado por los ciclos de vida de la planta. Estos ciclos de vida se calculan monitorizando el diámetro del tronco de dicha planta y observando en qué momento alcanza el máximo de cada día. El período de cálculo a nivel de árbol de un día N abarcará desde el momento de diámetro máximo del día N-1 hasta el momento de diámetro máximo del día N. Es importante tener en cuenta esta diferenciación cuando se comparan métricas que están directamente relacionadas con los diámetros de la planta con otras que no lo están. Por ejemplo, si para la activación de un evento hay que observar el crecimiento diario de una planta y la temperatura media, ambas métricas deberán ser calculadas a nivel de árbol para que la comparación sea válida, a pesar de que la temperatura no sea una medida que esté directamente relacionada con el valor del diámetro de la planta.

4.1.2. Subsistema de eventos

Este sistema busca que el usuario entienda que hay una serie de objetivos de control y que tiene indicadores sencillos que sirven para buscar y monitorizar cómo va su cumplimiento. En primera instancia, cada usuario tendrá una serie de eventos por defecto, basados en experiencias previas con otras fincas, pero los podrá ir personalizando a medida que se enriquezca su experiencia y se familiarice con el sistema. Los eventos por defecto se agrupan en:

- Eventos de planta, evaluados según los siguientes indicadores:
    - Crecimiento diario: indica si el diámetro de la planta aumenta o disminuye de un día para otro.
    - Actividad fotosintética.
    - Estrés bueno y malo.
    - Medidas de campo (área foliar, crecimiento de nudos, etc.).

- Eventos de suelo: se definen unos niveles para ciertas métricas entre los cuales deben oscilar las medidas tomadas a nivel de raíces y drenaje. Concretamente, se monitorizan las siguientes métricas:
    - Humedad
    - Conductividad
    - Nitrato
    - Potasio
- Eventos de clima: basados en sucesos climáticos que se sabe que afectan al comportamiento de la planta.

La evaluación de todos estos tipos de eventos dará lugar a la activación de una serie de alarmas que le indicarán al usuario que debe llevar a cabo ciertas acciones sobre sus cultivos.

4.2. Arquitectura del sistema

Para la implementación del sistema de cálculos se ha utilizado lenguaje Java, ya que, al ser orientado a objetos, facilita la organización del código y facilita la ampliación de funcionalidades en el futuro si se deseara. Otro motivo de la elección de Java para la implementación del sistema de cálculos es que permite realizar llamadas a programas desarrollados en Matlab, lo cual será útil en trabajos futuros en los que se desea añadir características de aprendizaje automático al sistema.

4.2.1. Subsistema de modelos

A continuación, se van a describir las distintas partes en las que se ha dividido este módulo:

- Core: paquete Java en el que se incluyen los modelos encargados de realizar cálculos básicos para el sistema. Se entiende como cálculo básico:
    - Cálculos de cuyo resultado dependa algún evento del sistema.
    - Cálculos cuyo resultado pueda ser utilizado por otros modelos.

  Dentro de este paquete, los modelos se han dividido en distintas clases en función de si llevan a cabo cálculos relacionados con métricas de suelo, clima o planta, y de si se realizan a nivel de árbol o PSD.

- PsdModel.java: clase abstracta de la que deben heredar las clases que implementen modelos a nivel de PSD. Define los métodos que deben tener estas clases para que su funcionamiento sea correcto.
- psdModels: paquete en el que se incluyen las clases que contengan modelos implementados a nivel de PSD y que deben heredar de la clase abstracta PsdModel.java. La ejecución de estas clases se realiza de forma automática, y no deben contener dependencias de cálculos realizados por otros modelos que no sean los que se incluyen en el paquete “core”.
- TreeModel.java: clase abstracta de la que deben heredar las clases que implementen modelos a nivel de árbol. Define los métodos que deben tener estas clases para que su funcionamiento sea correcto.
- treeModels: paquete en el que se incluyen las clases que contengan modelos implementados a nivel de árbol y que deben heredar de la clase abstracta TreeModel.java. La ejecución de estas clases se realiza de forma automática, y no deben contener dependencias de cálculos realizados que no sean lo que se incluyen en el paquete “core”.
- ModelsRunner.java: clase que lleva a cabo el proceso de ejecución periódica de los modelos. También ejecuta de forma automática las clases incluidas en los paquetes “psdModels” y “treeModels”.

En la siguiente imagen se muestra el diagrama de clases correspondiente al paquete de modelos:

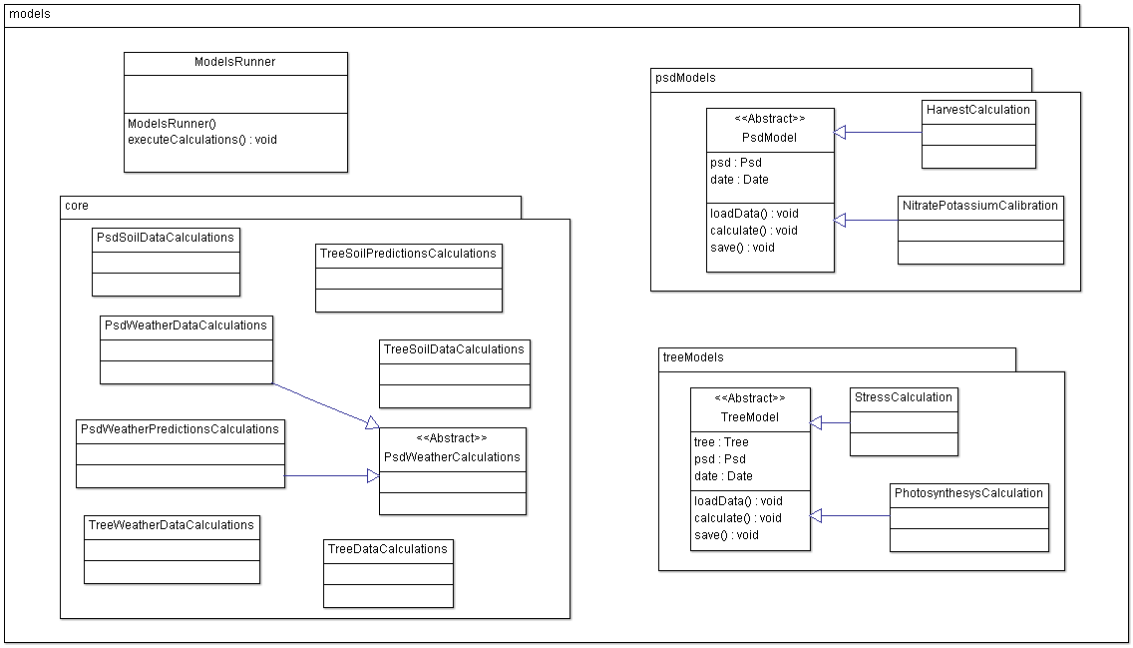


IMAGEN 13: DIAGRAMA DE CLASES DEL PAQUETE DE MODELOS

4.2.2. Subsistema de eventos

Como se ha indicado anteriormente, los eventos son avisos que reciben los usuarios y que les indican que el estado de la planta no es todo lo óptimo que debería. Para que estos avisos puedan conllevar más complejidad, se han dividido en reglas que pueden personalizar los propios usuarios y administradores del sistema que se han denominado patrones. Estas reglas pueden consistir en condiciones simples, basadas en la observación de que una variable se encuentre entre ciertos umbrales, o más complejas si implican comparaciones entre varias variables. El cálculo de un patrón consiste en la evaluación de la condición que tiene definida para saber si se activa o no.

Al igual que en el subsistema de modelos, es necesario hacer una separación entre los patrones que se evalúan a nivel de árbol y los que se evalúan a nivel de PSD. Esta separación se hace en función de las variables que necesita cada patrón para ser evaluado. El módulo que evalúa los eventos servirá como capa de abstracción a la hora de gestionar esta diferenciación, ya que los eventos serán solo a nivel de PSD.

Definición de patrones

En este apartado se va a estudiar la forma en que se pueden definir y almacenar patrones en el sistema. Los patrones almacenados son condiciones basadas en la comparación de una variable calculada con otro valor, que podrá ser una constante fija,

un valor establecido por el usuario y otro valor calculado por el sistema. La evaluación de un patrón implica, por tanto, asignarle un valor de verdadero o falso en función de si se cumple su condición o no. Para almacenar patrones en el sistema se ha diseñado una estructura de datos con los siguientes atributos:

- Nombre: nomenclatura que se le da al patrón y que es utilizado para la evaluación de eventos.
- Operador: comparador que se utiliza para calcular el valor del patrón.
- Parámetro: dato calculado por el sistema que se utiliza para evaluar el patrón.
- Umbral: valor fijo con el que se compara el atributo “parámetro” para saber si el patrón se activa o no.
- Constante: valor que establece el usuario que se compara con el atributo “parámetro” para saber si el patrón se activa o no.
- Parámetro 2: dato calculado por el sistema que se compara con el atributo “parámetro” para saber si el patrón se activa o no.
- Tipo: campo utilizado para separar patrones de PSD y de árbol.

Como se ha mencionado anteriormente, el parámetro se puede comparar con un umbral, una constante u otro parámetro, por lo que, para cada patrón, sólo uno de los atributos “umbral”, “constante” y “parámetro 2” (aquel con el que haya que comparar) tendrá un valor distinto de NULL. El único caso en que dos de estos atributos pueda ser distinto de NULL es cuando se quiera comparar un parámetro con otro multiplicado por un cierto valor. Para especificar ese valor se utilizaría el atributo “constante”, para que pueda ser personalizado por el usuario.

A continuación, se puede ver una tabla con ejemplos de patrones y una explicación del funcionamiento de cada uno de ellos:

ID	NOMBRE	PARÁMETRO	OPERADOR	PARÁMETRO 2	UMBRAL	CONSTANTE
1	HR_LT_CTE_EC	Humedad relativa media	<	-	-	Humedad relativa mínima para estrés climático
2	T_GT_T_PREV	Temperatura media	>	Temperatura media día previo	-	Multiplicador para temperatura de riesgo día previo
3	H_VV_GT_P	Horas de velocidad del viento con peligro de propagación	>	-	4	-

TABLA 2: EJEMPLO DE ALMACENAMIENTO DE PATRONES

1. HR_LT_CTE_EC: patrón que indica cuándo la humedad relativa mínima está por debajo de un valor. Se activa cuando el parámetro "Humedad relativa mínima" es menor que el valor que el usuario haya dado para la constante "Humedad relativa mínima para estrés climático".
2. T_GT_T_PREV: busca los días en que la temperatura media sube considerablemente respecto al día anterior. Para ello, compara el parámetro "Temperatura media" con el valor de "Temperatura media día previo" multiplicado por el valor que haya asignado el usuario a la constante "Multiplicador para temperatura de riesgo día previo" (por ejemplo, si el usuario quiere que el patrón busque los días en que la temperatura suba un 20% respecto el día anterior, deberá asignarle a la constante un valor de 1.2).
3. H_VV_GT_P: el patrón avisará de los días en que la velocidad del viento haya superado un límite que se considera de riesgo para propagación durante 4 horas.

La implementación de este sistema se ha dividido en las siguientes partes:

- PatternEvaluatorInterface.java: interfaz de Java que deben implementar las clases que se encarguen de evaluar patrones. Declara los siguientes métodos que luego deben implementar las clases:
    - buildExpression(): se encarga de construir la expresión lógica utilizada para evaluar el patrón correspondiente. Para ello une el valor del parámetro del patrón, el operador, y el segundo operando del patrón.
    - evaluate(): evalúa la expresión construida en la primera función y calcula el valor del patrón.
    - save(): almacena el valor calculado para el patrón en la base de datos.
    - getSecondParameterValue(): devuelve el valor que tiene el segundo operando para el patrón correspondiente. Su funcionamiento variará dependiendo de si se trata de un patrón de PSD o árbol y de si es un patrón que utiliza un umbral fijo, una constante asignada por el usuario u otro parámetro.
- PsdPatternEvaluator.java: clase encargada de evaluar los patrones de PSD a tiempo pasado. Debe implementar las funciones de la interfaz PatternEvaluatorInterface.
- PsdPatternPredictionEvaluator.java: clase encargada de evaluar los patrones de PSD a tiempo futuro, basados en predicciones. Debe implementar las funciones de la interfaz PatternEvaluatorInterface.
- TreePatternEvaluator.java: clase encargada de evaluar los patrones de árbol a tiempo pasado. Debe implementar las funciones de la interfaz PatternEvaluatorInterface
- TreePatternPredictionEvaluator.java: clase encargada de evaluar los patrones de árbol a tiempo futuro, basados en predicciones. Debe implementar las funciones de la interfaz PatternEvaluatorInterface.
- PatternExpression.java: clase auxiliar que utilizan las clases de evaluación de patrones para construir la expresión con la que se calcula el valor de cada patrón.

En la siguiente imagen se puede observar el diagrama de clases correspondiente al paquete de patrones:

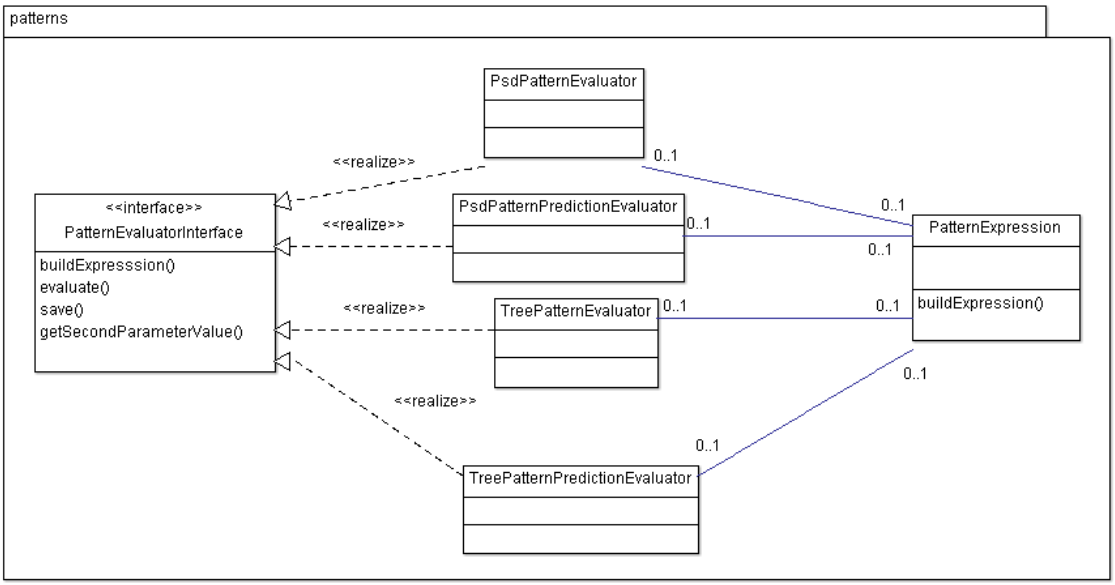


IMAGEN 14: DIAGRAMA DE CLASES DEL PAQUETE DE PATRONES

Definición de eventos

Los eventos del sistema son el resultado de la combinación de distintos patrones. Como ya se ha mencionado en el apartado anterior, un patrón puede estar activo o inactivo, lo que se traduce en que sea verdadero o falso, por lo que el resultado de la evaluación de un evento también será verdadero o falso, dependiendo de los valores que tengan los patrones que lo componen. A la hora de relacionar entre sí los patrones de un evento, se planteó la posibilidad de que se tuvieran que activar todos los patrones de un evento para activar el evento. Esto simplificaría considerablemente el desarrollo, pero, observando casos reales y con el fin de cubrir todas las necesidades que pudieran tener los usuarios, se decidió desarrollar un motor de interpretación de expresiones capaz de interpretar expresiones lógicas compuestas por los valores de distintos patrones. Con este objetivo, se ha asociado a cada evento una expresión compuesta por operadores lógicos y nombres de patrones (que irán entre corchetes). Un ejemplo basado en los patrones vistos anteriormente podría ser:

([HR_LT_CTE_EC] && [T_GT_T_PREV]) || [H_VV_GT_P]

El sistema implementado debe ser capaz de reconocer los nombres de los patrones, buscar sus valores calculados, sustituir cada nombre por su valor correspondiente y evaluar la expresión resultante en un valor de verdadero o falso. En la siguiente imagen se puede ver un esquema del funcionamiento del generador de eventos:

Se crea un evento llamado "Estrés climático", y se configura el sistema de almacenamiento de información para que dicho evento se active cuando se cumpla alguno de los siguientes patrones:

- Humedad relativa media de un dia inferior al 30%
- Temperatura media de un dia mayor que 33ºC.

Patrones

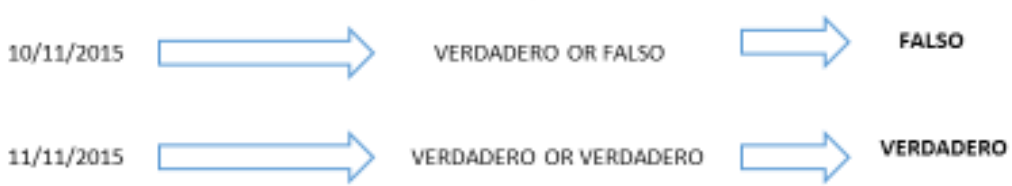
ID	NOMBRE	DESCRIPICIÓN
1	HR_LT_30	Condición de humedad relativa menor que 30%
2	T_GT_33	Condición de temperatura mayor que 33ºC

Eventos

ID	NOMBRE	EXPRESIÓN
1	Estrés climático	HR_LT_30 OR T_GT_33

Valores de eventos

ID DE CONDICIÓN	FECHA	VALOR
1	10/11/2015	VERDADERO
2	10/11/2015	FALSO
1	11/11/2015	VERDADERO
2	11/11/2015	VERDADERO



El dia 10/11/2015 el evento de "Estrés climático" no estará activado.
El dia 11/11/2015 el evento de "Estrés climático" estará activado.

IMAGEN 15: ESQUEMA DEL FUNCIONAMIENTO DEL GENERADOR DE EVENTOS

Anteriormente se ha explicado que tanto el subsistema de modelos como el subsistema de patrones funcionan de forma distinta si se evalúan datos a nivel de PSD o a nivel de árbol. Los eventos generados por el sistema, en cambio, solo existen a nivel de PSD, por lo que la generación de eventos debe servir como una capa de abstracción entre los cálculos realizados y lo que se muestra al usuario. Esto supone un problema cuando se da el caso de que algunos patrones de un mismo evento se calculen a nivel de árbol y otros a nivel de PSD, ya que eso implica tener que combinar el valor de los patrones de cada una de las plantas del PSD y el valor de los patrones globales del mismo PSD. Esto se entiende mejor con un ejemplo: si se define un nuevo evento que se activa cuando alguno de los árboles del PSD tenga crecimiento diario negativo y la temperatura sea mayor que 25°, la expresión de dicho evento será la siguiente:

[CD_LT_0] && [T_GT_25]

En el caso de que en un PSD haya 3 árboles, el motor de interpretación debe ser capaz de extraer los valores de crecimiento diario de cada uno de esos árboles (CD1, CD2 y CD3) y combinarlo con el valor de la temperatura de todo el PSD, por lo que la expresión lógica que debe evaluar el sistema para el evento es:

([CD1_LT_0] && [T_GT_25]) || ([CD2_LT_0] && [T_GT_25]) ||

([CD3_LT_0] && [T_GT_25])

A continuación, se describen las distintas partes en que se ha dividido la implementación de este módulo:

- EventsGenerator.java: es la clase principal del módulo. Se encarga de recorrer la lista de PSDs del sistema e ir llamando a las clases que se encargan de calcular el valor de los eventos correspondientes y almacenar dichos valores en la base de datos.
- EventEvaluator.java: clase que calcula el estado de un evento para un PSD y fecha determinados. Comprueba la lista de patrones que componen el evento y busca sus valores para el PSD y la fecha correspondiente en las tablas de cálculo de patrones de tiempo pasado. Además, debe comprobar si se combinan patrones a nivel de árbol y a nivel de PSD y gestionar la correcta interpretación del evento en este caso.

- EventPredictionEvaluator.java: realiza las mismas funciones que la clase EventEvaluator pero tomando como valor de cada patrón el más reciente en las tablas de cálculo de patrones a tiempo futuro.

5. Aplicación web de visualización de datos

La aplicación web presenta a los usuarios una interfaz mediante la cual éstos podrán hacer un seguimiento detallado de sus cultivos y visualizar toda la información disponible en la base de datos del sistema.

Las principales funcionalidades se enumeran a continuación:

- Dashboard o cuadro de mando en el que se resumirá la información más relevante que permitirá realizar con facilidad el seguimiento de los cultivos.
- Análisis detallado de las diferentes variables relacionadas con el comportamiento de los cultivos. Se podrá, también, mediante la visualización de información geográfica, analizar espacialmente la evolución de los cultivos.
- Visualización de predicciones meteorológicas a corto plazo.
- Visualización de alertas que avisen de comportamientos no deseados en los cultivos.

5.1. Especificaciones funcionales

Como se ha mencionado anteriormente, el sistema debe mostrar al usuario un resumen del estado general de su finca. Para ello, se definen tres focos de información que, combinada, ofrecen al usuario toda la información necesaria para que el usuario pueda decidir sobre las acciones que debe ejecutar sobre sus tierras:

- **Eventos**: el usuario puede navegar entre los distintos eventos definidos para su finca, que se activarán en función de si hay o no una situación de riesgo que debe ser atendida por el técnico agrario. Estos eventos se definen de forma jerárquica, agrupando aquellos que tengan similitudes, ya sea por el nivel sobre el que actúan (planta, clima o suelo) o por depender de condiciones similares.
- **Información geolocalizada**: se debe indicar al usuario la zona concreta de su finca en la que se produce un problema. Para ello, se mostrará un mapa de sus tierras que incorpore la separación territorial llevada a cabo por el técnico encargado de los cuidados de las mismas.

- **Información detallada**: además de resumir la información relativa al estado de la finca, el usuario podrá ver en todo momento los datos recogidos por los sensores que tiene repartidos por sus tierras.

La combinación de esta información, ha dado lugar a la interfaz web que se muestra en la siguiente imagen. Está compuesta principalmente por los cuatro elementos enumerados a continuación, los cuales son descritos en detalle en los siguientes apartados:

1. Panel lateral.
2. Carrusel temporal
3. Mapa
4. Sección de gráficas

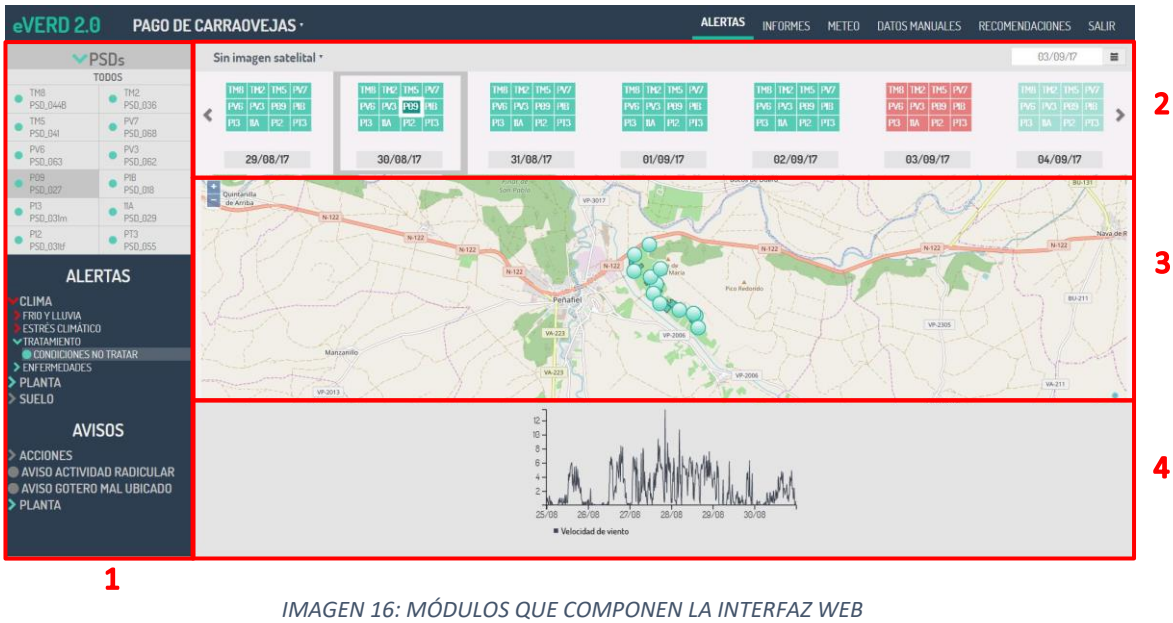


IMAGEN 16: MÓDULOS QUE COMPONEN LA INTERFAZ WEB

5.1.1. Panel lateral

El panel lateral está dividido en dos partes claramente diferenciadas: la sección de PSDs y la sección de eventos:

IMAGEN 17: PANEL LATERAL DE LA INTERFAZ WEB

Sección de PSDs

Contiene una tabla con los Puntos de Seguimiento Detallado que componen la finca seleccionada. Junto al nombre de cada uno de los PSDs hay un color que indica si el evento que se haya seleccionado está activo en el terreno que abarca dicho punto. Se pueden seleccionar los distintos PSDs de la tabla para ver el estado de todos los eventos

para cada uno de ellos. Además, se ha incluido un botón de “TODOS” que puede seleccionarse para ver el estado de los eventos para todos los PSDs.

Sección de eventos

Contiene una lista jerárquica y desplegable con los distintos eventos de tipo “Alerta” o tipo “Aviso” que hay declarados para la finca seleccionada. Esta diferenciación entre alerta o aviso se lleva a cabo para distinguir la gravedad de las consecuencias que pueden ocurrir si se ignora un evento:

- Alerta: evento indicativo de que está ocurriendo algo que afecta negativamente a los cultivos y que hay que atender de forma inmediata.
- Aviso: indica que se podría producir una situación de riesgo en los próximos días, que puede dar lugar a una alerta.

Junto al nombre de cada evento, hay un icono de una flecha si el evento correspondiente tiene más hijos en la jerarquía y se puede desplegar o un icono de un punto si dicho evento no tiene más hijos. Además, el color del icono varía en función del estado del evento de la siguiente manera:

- Verde: indica que ni ese evento ni ninguno de sus hijos se ha activado en el PSD que hay seleccionado.
- Rojo: indica que una alerta se ha activado o que alguno de los hijos de dicho evento se ha activado. En el caso de los avisos, el color elegido es naranja.
- Gris: indica que un evento no se ha podido calcular. Esto suele ocurrir cuando se produce algún error durante la ejecución de los cálculos.

Los eventos de la lista también son seleccionables, de forma que, al seleccionar un aviso o alerta, los colores de los PSDs cambiarán en función de si dicho evento está activo para ese punto.

5.1.2. Carrusel temporal

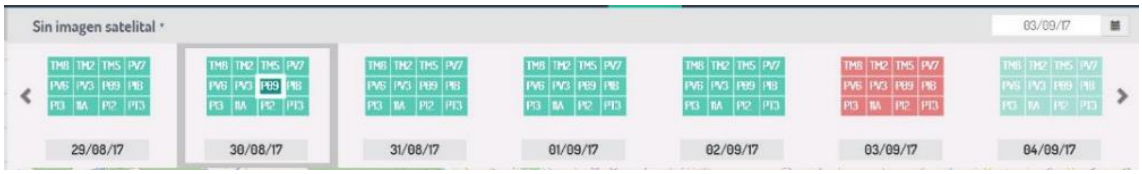


IMAGEN 18: CARRUSEL DE NAVEGACIÓN TEMPORAL

El carrusel temporal es la sección que muestra la evolución temporal de los eventos en la finca seleccionada. En él, se muestra una serie de tablas con los PSDs de la finca coloreados en función del estado del evento que haya seleccionado en la sección de eventos del panel lateral. Concretamente, en esta sección se podrán visualizar 7 días a la vez, aunque con las flechas laterales se podrá navegar por 21 días. Para ver el estado de los eventos en fechas que no estén incluidas en esos 21 días, se puede seleccionar una fecha concreta en el selector de fecha que hay encima del carrusel y la aplicación cargará los eventos de dicho día, rellenando el carrusel con los 10 días anteriores y los 10 posteriores.

La representación de los PSDs se hace en una malla que se adapta en función del número de puntos que tenga la finca. Dicha malla podrá representar desde un solo punto hasta un máximo de 25, de la forma que se ve en la figura:

IMAGEN 19: ADAPTACIÓN DE LA MALLA DE PSDS DEL CARRUSEL

Los PSDs que se muestran en el carrusel son también seleccionables, de manera análoga a la tabla de PSDs que hay en el panel lateral.

Además de modificarse el color del PSD en función del estado del evento que haya seleccionado en ese momento, la opacidad con la que se muestra el PSD también proporciona información. Concretamente, indica si el evento correspondiente ha sido calculado con datos medidos por sensores o si está basado en predicciones climáticas. Los eventos que sean de días futuros se mostrarán con un mayor nivel de transparencia, dando a entender que dicha información es menos fiable. El nivel de transparencia

seguirá aumentando a medida que se avance en los días futuros, ya que las predicciones utilizadas tendrán una menor precisión.

5.1.3. Mapa

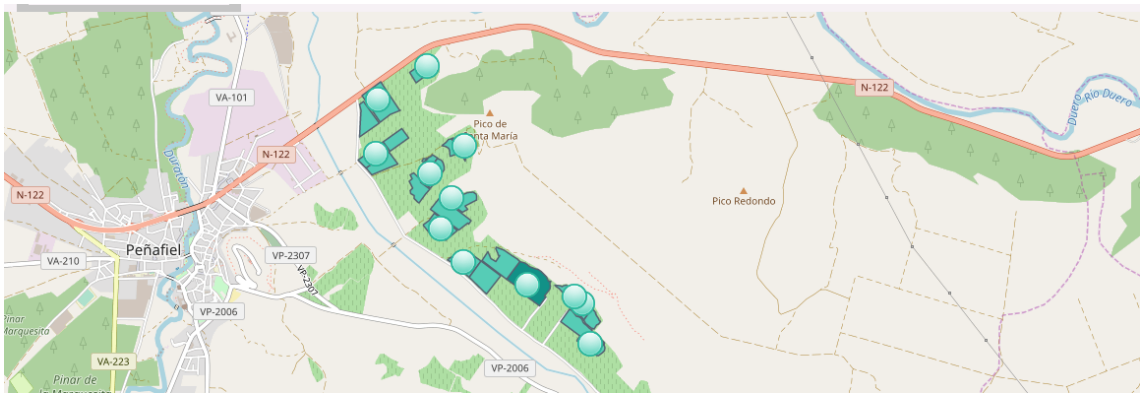


IMAGEN 20: MAPA DE LA INTERFAZ WEB

La sección del mapa permite la visualización geográfica de las zonas monitorizadas. De esta forma, se ofrece al usuario una visión general de la finca y su estado. Las entidades que se muestran en el mapa pueden ser de dos tipos:

- Unidades de manejo: se representan con un polígono sobre el mapa, y muestran el conjunto de tierra en el que puede extrapolarse la información medida en un solo punto de la finca.
- PSDs: se representa con puntos en el mapa que indican el lugar exacto en el que se encuentran los sensores que toman las medidas de dicho punto.

Los colores en los que se muestran las entidades en el mapa funcionan de forma similar a los PSDs de la sección lateral y del carrusel, es decir, el color cambia en función de si el evento seleccionado se encuentra activo en el día correspondiente.

5.1.4. Sección de gráficas

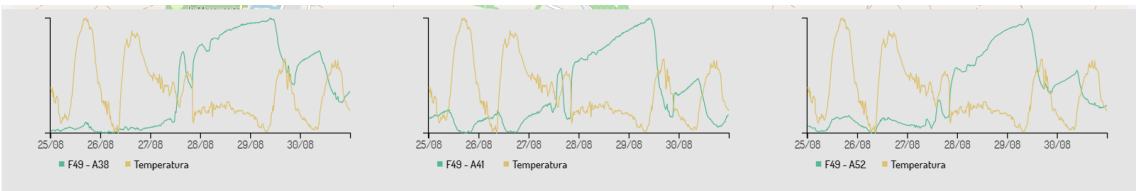


IMAGEN 21: SECCIÓN DE GRÁFICAS DE LA INTERFAZ WEB

En esta sección se muestra la información de lo que ocurre en la finca de la forma más detallada posible mediante gráficas que mostrarán las variables asociadas a cada evento, esto es, aquellas que sirvan para entender mejor el motivo de la activación o no

activación de un evento. Estas gráficas podrán ser multivariables y tendrán la escala temporal en el eje X y la escala de valores en el eje Y. Solo se mostrará el eje Y en aquellas gráficas que solo tengan una variable. En cualquier caso, se podrán ver valores concretos al posar el cursor sobre alguna de las gráficas en un punto determinado.

5.2. Arquitectura del sistema

5.2.1. Tecnologías utilizadas

Al tratarse de una aplicación web, el sistema se divide en dos partes: el código de cliente que se ejecuta en el navegador de cada usuario que accede a la aplicación y código de servidor.

Tecnologías utilizadas en el lado del cliente

HTML

El "HyperText Markup Language" (lenguaje de marcas de hipertexto) es el lenguaje de programación por excelencia para la elaboración de páginas web. Es el encargado de definir la estructura básica y el contenido de los sitios web. Su funcionamiento se basa en el envío de los ficheros con el código HTML del servidor a los clientes, cuyos navegadores serán los encargados de interpretar dicho código y representarlo. A día de hoy, prácticamente la totalidad de los navegadores estándar del mundo son compatibles con este lenguaje. Es un estándar a cargo de la W3C.

CSS

Corresponden en inglés a las siglas de "Cascading StyleSheet" (Hojas de Estilo en Cascada). Se trata de un estándar que define la manera en que se deben presentar los componentes de una web declarados en el código HTML. Está permitido definir la forma en que se deben representar los elementos dentro del propio fichero HTML en el que se declaran, o bien hacerlo en un fichero separado que solo contendría código CSS. Esta última forma de distribuir el código es el paradigma de la idea que había tras el surgimiento de este estándar: separar la presentación de los elementos (su estilo) de la declaración de los mismos.

JavaScript

Se trata de un lenguaje de programación orientado a objetos que se ejecuta en el lado del cliente, principalmente con el fin de llevar a cabo mejoras en la interfaz de usuario y la creación de páginas web dinámicas. Se ha desarrollado una gran cantidad de frameworks y librerías basadas en este lenguaje de programación, que permiten dar a los programadores una mayor facilidad a la hora de aprovechas las capacidades del lenguaje. Concretamente, en el desarrollo de este proyecto, se han utilizado las siguientes librerías basadas en JavaScript:

- jQuery: probablemente la librería más extendida de JavaScript. Está formada por una serie de clases y funciones que facilitan el manejo de eventos en los elementos HTML y que permiten ejecutar animaciones en los mismos. Además, ofrece una API para realizar peticiones HTTP asíncronas median AJAX que resulta muy útil para dar dinamismo a la interfaz de la aplicación, permitiendo recuperar datos del servidor sin necesidad de recargar la página en la que el usuario está navegando.
- D3js: se trata de una librería diseñada para la representación de información. Ofrece un conjunto de métodos que permiten enlazar una serie de datos con elementos DOM y llevar a cabo transformaciones en ellos. Las funciones de esta librería se encargan de generar y/o alterar código HTML y CSS mediante la creación de SVG (Scalable Vector Graphics). La principal característica de esta librería, lo que la diferencia en gran medida de muchas de las que han surgido a partir de ella, es que se centra en proporcionar a los desarrolladores un conjunto de funcionalidades que les permiten manipular elementos HTML basados en conjuntos de datos. Es decir, no ofrece un conjunto de opciones para representar los datos de distinta manera, sino que proporciona las herramientas para que sea el desarrollador quien construya las representaciones que mejor se adapten a sus necesidades.
- AngularJS: es un framework mediante el cual es posible crear aplicaciones web multiplataforma completas de página única. Esta tecnología facilita la implantación de los patrones de diseño Modelo-Vista-Controlador (MVC) y Modelo-Vista-VistaModelo (MVVM) en el código cliente, con el fin de separar la

manipulación de la vista de la lógica de la aplicación. Esto facilita la capación de prueba del código.

- OpenLayers: es una librería de JavaScript para visualización de mapas dinámicos e interactivos en páginas web. Permite mostrar polígonos, imágenes y vectores sobre mapas obtenidos de distintas fuentes de información (OpenStreetMaps, Bing, Google Maps, etc.). Además, permite una sincronización con AngularJS de forma sencilla gracias al uso de la directiva “angular-openlayers”.

Tecnologías utilizadas en el lado del servidor

Node.js

Es un entorno en tiempo de ejecución multiplataforma de código abierto, que puede usarse tanto en la capa del servidor como en el cliente. Está basado en el lenguaje de programación ECMAScript, asíncrono, con entrada y salida de datos en una arquitectura orientada a eventos y basado en el motor V8 de Google. La principal ventaja que ofrece frente a otros motores del lado del servidor es que permite, mediante el uso de patrones tales como MVC, la reutilización de código del mismo modelo de interfaz entre el lado del cliente y el lado del servidor. Además, tiene una amplia comunidad de desarrolladores detrás, lo que hace que existan multitud de librerías de código abierto que pueden ser utilizadas en una aplicación.

ExpressJS

Framework de código abierto para Node.js que proporciona un conjunto sólido de características para la construcción de aplicaciones web y móviles. Ofrece, además, una gran cantidad de métodos que facilitan la creación de APIs que se pueden emplear en las aplicaciones desarrolladas.

Mocha

Librería para tests unitarios de JavaScript que se ejecuta bajo el motor de Node.js y en el navegador, haciendo más simple la creación de tests tanto síncronos como asíncronos. También proporciona muchas utilidades para la ejecución y el reporte de los tests.

Otras tecnologías utilizadas

A continuación, se describe un conjunto de tecnologías que se han utilizado en el desarrollo de la aplicación de forma global.

npm

Se trata de un gestor de paquetes libre para JavaScript en el lado del cliente. Contiene el mayor archivo de librerías existente en el mundo. Permite instalar, compartir y distribuir librerías, gestionar dependencias entre frameworks en un proyecto y enviar y recibir opiniones y peticiones de otros desarrolladores.

Bower

Es un gestor de paquetes similar a npm pero para librerías de lado del cliente, que reciben el nombre de "componentes". Si bien npm solo gestiona librerías construidas con Node.js, los componentes de Bower pueden contener código HTML, CSS, JavaScript, fuentes o incluso imágenes. Bower no minimiza el código de los componentes de ninguna forma, solo instala las versiones correctas necesarias para el proyecto junto con las dependencias correspondientes.

Git

Git es el software de control de versiones más extendido y utilizado por desarrolladores de todo el mundo. El control de versiones es un sistema que registra los cambios realizados sobre un archivo o conjunto de archivos a lo largo del tiempo, de modo que puedas recuperar versiones específicas más adelante. Fue desarrollado pensando en la eficiencia y la confiabilidad del mantenimiento de versiones de aplicaciones cuando éstas tienen un gran número de archivos de código fuente.

6. Conclusiones y trabajo futuro

6.1. Conclusiones

Al hacer frente a este proyecto, se ha comprobado que, a día de hoy, todavía hay muchos sectores a los que no se han aplicado las nuevas tecnologías. Este fenómeno no se produce por dificultades técnicas de la propia aplicación tecnológica, sino por desconocimiento de las necesidades reales de los trabajadores de estos sectores.

En el sector agrario, concretamente, se han estudiado las distintas alternativas existentes en el mercado a día de hoy, y se ha comprobado que ninguna cumple con la totalidad de las demandas del sector. Esto obliga a los trabajadores a tener que utilizar múltiples herramientas, con el gasto económico y de tiempo que esto conlleva.

Gracias al sistema desarrollado, se ofrece una solución que combina los principales sistemas tecnológicos con los que se trabaja a día de hoy en el sector agrario: información geográfica, datos de sensores y predicciones climáticas. Además, el sistema de recuperación de datos de sensor está adaptado para los principales fabricantes de hardware del sector.

Con el sistema de ejecución de cálculos, se consigue reducir toda la información recogida por los sensores a una serie de eventos que avisan cuando haya situaciones desfavorables en los cultivos. Dichos eventos, además, son ajustables a las necesidades de cada agricultor, gracias al sistema de evaluación de condiciones que se ha desarrollado.

Por último, la aplicación web de visualización de datos implementada supone un importante ahorro de tiempo a la hora de diagnosticar los problemas que haya en una determinada finca, ya que permite a un técnico, con un simple vistazo, detectar inmediatamente las zonas en las que tendrá que hacer frente a algún contratiempo.

6.2. Trabajo futuro

Durante el proceso de diseño y análisis del sistema, fueron surgiendo ideas y mejoras aplicables al proyecto que hubo que ir descartando en favor de otras funcionalidades

más necesarias para una primera versión de la aplicación. Algunas de estas ampliaciones del sistema son:

- Módulo de administración mediante el cual un usuario pueda modificar los datos relativos a su finca. De esta forma, un cliente podría dar de alta nuevos sensores o PSDs en la finca, modificar los eventos que calcula el sistema o las variables asociadas que se muestran con cada evento.
- Sección específica para la visualización de predicciones climáticas. En ocasiones, a los usuarios les resulta útil poder observar la evaluación que tendrá el clima en su finca directamente, sin tener que navegar entre los eventos que tiene activos.
- Módulo para la visualización de todos los datos almacenados en el sistema. Actualmente, la única opción que tiene el usuario de ver los datos que la aplicación ha recogido sobre sus datos es la sección de gráficas. Sería de gran utilidad una sección en la que los usuarios puedan ver los datos en crudo de su finca en otro formato, como tablas, y poder exportarlos en ficheros descargables.
- Aplicación de algoritmos predictivos sobre los datos del sistema. Actualmente en el sistema solo hay predicciones relativas a datos de magnitudes climáticas, porque el sistema adaptador se encarga de recuperarla. Un posible trabajo futuro es añadir al sistema de cálculos la ejecución de algoritmos basados en aprendizaje automático para realizar la predicción de otras variables. Ya existen trabajos en los que se ha utilizado este tipo de algoritmos para predecir el rendimiento de cultivos (Xujun, y otros, 2007) o el momento de arado de tierras (R. B., y otros, 2016).

7. Referencias

ADCON Telemetry: smart wireless solutions [En línea] / aut. ADCON Telemetry. - 6 de 9 de 2017. - http://www.adcon.com/.

AEMET - Gobierno de España [En línea] / aut. Agencia Estatal de Meteorología. - 6 de 9 de 2017. - http://www.aemet.es/es/portada.

Automatic arable land detection with supervised machine learning [Publicación periódica] / aut. R. B. Arango [y otros] // Earth Science Informatics. - 11 de 2016. - 4 : Vol. 6. - págs. 535-545.

Cambio climático 2014: Impactos, adaptacion y vulnerabilidad. [Publicación periódica] / aut. Grupo Intergubernamental de Expertos sobre el Cambio Climático. - 2014.

Estrategia Nacional para la Modernización Sostenible de los Regadíos, Horizonte 2015 [Informe] / aut. Ministerio de Medio Ambiende y Medio Rural y Marino. - 2015.

FIC [En línea] / aut. Fundación para la Investigación del Clima. - 6 de 9 de 2017. - https://www.ficlima.org/.

JavaScript Patterns [Libro] / aut. Stefanov Stoyan. - [s.l.] : O'Reilly, 2010.

La savia como índice de fertilización: cultivos agroenergéticos, hortícolas, frutales y ornamentales [Libro] / aut. López Carlos Cadahía. - [s.l.] : Ediciones Mundi-Prensa, 2008.

Long-term response of 'Clementina de Nules' citrus trees to summer regulated deficit irrigation. [Publicación periódica] / aut. Carlos Ballester [y otros] // Agricultural Water Management.. - 2014. - Vol. 138. - págs. 78-84.

MappingGIS. Formación GIS. Difusión tecnológica [En línea] / aut. MappingGIS. - 6 de 9 de 2017. - https://mappinggis.com/.

Memoria de MAGRAMA 2014 [Informe] / aut. Ministerio de Agricultura, Alimentación y Medio Ambiente. - 2014.

Modpow. Wireless monitoring and engineering [En línea] / aut. Modpow. - 06 de 9 de 2017. - https://www.modpow.es/es/.

Overview of topics and questions to be addressed by the FG Mainstreaming Precision Farming. Starting paper for Focus Group on Mainstreaming precision Farming meeting, European Innovation Partnership. [Informe] / aut. Kempenaar Corné. - 2014.

Physiological and agronomic mandarin trees performance under saline reclaimed water combined with regulated deficit irrigation [Publicación periódica] / aut. F. Pedrero [y otros] // Agricultural Water Management. - 2014. - Vol. 146. - págs. 228-237.

Prediction of citrus yield from airborne hyperspectral imagery [Publicación periódica] / aut. Xujun Ye [y otros] // Precision Agriculture. - 6 de 2007. - 3 : Vol. 8.

SCADA [En línea]. - 6 de 9 de 2017. - https://en.wikipedia.org/wiki/SCADA.

Tecnología agrícola [En línea] / aut. CERES. - 6 de 9 de 2017. - http://www.tecnologia-agricola.com/.

8. Glosario

- **Datalogger**: dispositivo que registra datos en el tiempo por medio de instrumentos y sensores propios o conectados externamente.
- **Estado fenológico**: cada una de las etapas por las que pasa una planta durante su fase de crecimiento y maduración.
- **Eventos**: situaciones de riesgo que evalúa el sistema de ejecución de cálculos y que están compuestas por una serie de patrones.
- **GIS**: acrónimo de *Geographic Information System* (Servicio de Información Geográfica). Sistema que integra información de distintos componentes para la manipulación de grandes cantidades de datos vinculados a una referencia espacial.
- **Patrones**: condiciones relativas al estado de un cultivo que son evaluadas por el sistema de cálculos.
- **PSD**: Punto de Seguimiento Detallado. Cada uno de los puntos donde se toman las medidas de una finca y cuya información es extrapolable a un área de terreno delimitada.
- **Sensores**: conjunto de instrumentos utilizados para medir magnitudes físicas en el terreno cultivado.
- **Sistemas GPRS**: conjunto de sensores que utilizan el Servicio General de Paquetes vía Radio (*General Packet Radio Service*) para el envío de datos a los dataloggers.
- **Sistemas SCADA**: acrónimo de *Supervisory Control And Data Acquisition* (Supervisión, Control y Adquisición de Datos). Sistema que permite la supervisión y gestión de procesos industriales a distancia.