

Universidad Internacional de La Rioja (UNIR)

Escuela de Ingeniería

**Máster universitario en Dirección e Ingeniería
de Sitios Web**

RAMA INVESTIGACIÓN

Sistema de riego autónomo basado en la Internet de las Cosas

Trabajo Fin de Máster

Presentado por: Castro Silva, Juan Antonio

Director: García Díaz, Vicente

Ciudad: Neiva – Huila (Colombia)

Fecha: 15 de enero de 2016

Resumen

Las necesidades de agua para la producción de alimentos seguirán en aumento debido al crecimiento de la población mundial. En este trabajo de investigación se ha construido un sistema de riego autónomo basado en la Internet de las Cosas (IoT). Se emplean elementos de bajo costo y hardware - software libre (Raspberry Pi, Arduino, Linux, Java, Wildfly, Python, etc.) para implementar Redes de Sensores Inalámbricos (WSN) que permiten obtener la información de las variables agroclimáticas (Humedad del suelo, temperatura ambiente, precipitación, etc.). Se implementó un sistema de Aprendizaje Maquina (Machine Learning) para la predicción del calendario de riego empleando servicios de Computación en la Nube. Los Servicios Web tipo RESTful y los formatos Json y Xml se emplearon para permitir la interoperabilidad entre los diferentes subsistemas (predicción, riego y cliente), hardware (Raspberry Pi, Xbee, computación en la nube) y software (Python, Java). Para hacer un uso eficiente del agua este prototipo de Agricultura Inteligente (Smart Farming) se apoya en la Internet de las Cosas y el Aprendizaje Maquina para responder a las preguntas de cuándo y cuánto regar.

Palabras Clave: Internet de la Cosas (IoT), Agricultura de Precisión, Aprendizaje Maquina, Redes de Sensores Inalámbricos (WSN), Arduino, Raspberry PI, Servicios Web RESTful, Computación en la Nube.

Abstract

Water needs for food production will increase due to the growth of the world population. In this research it has built a self-watering system based on Internet of Things (IoT). Inexpensive items and free hardware - software (Raspberry Pi, Arduino, Linux, Java, Wildfly, Python, etc.) are used to implement Wireless Sensor Networks (WSN), allowing the information about agro-climatic variables (soil moisture, ambient temperature, precipitation, etc.). It was implemented a Machine learning system for the prediction of irrigation scheduling using Cloud Computing's service. RESTful Web Services, JSON and XML formats are used to enable interoperability between different subsystems (forecasting, irrigation and client),

hardware (Raspberry Pi, Xbee, cloud computing) and software (Python, Java). To improve efficient use of water, this prototype of Smart Farming is based on the Internet of Things and the Machine Learning to answer the questions: When and How much to water?

Keywords: Internet of Things (IoT), Precision Agriculture, Wireless Sensor Network (WSN), Machine Learning, Arduino, Raspberry PI, RESTful Web Services, Cloud Computing.

Tabla de contenido

1. INTRODUCCION	9
2. PROBLEMA	11
3. OBJETIVOS	12
3.1 General	12
3.2 Específicos	12
4. JUSTIFICACION	13
5. ESTADO DEL ARTE	14
5.1 Internet de las Cosas	14
5.2 Programación del Riego	21
5.3 Autonomía – Machine Learning	36
5.4 Hardware	40
5.5 Comunicación	41
5.6 Servicios Web	42
5.7 Seguridad	44
5.8 Computación en la nube	45
6. DISEÑO METODOLOGICO	47
6.1 Arquitectura del Sistema	47
6.2 Componentes de Hardware	49
6.3 Comunicación	52
6.4 Nodo Sensor	56
6.5 Nodo Coordinador – Gateway – Servidor Local	56
6.6 Nodo de Control de Riego	57
6.7 Servicios Web	58
6.7.1 Servicios Infraestructura	58
6.7.2 Servicios Hardware	59
6.7.3 Servicios Riego	59
6.8 Clientes	60
6.9 Machine Learning	60
6.9.1 Plot de los datos	60
6.9.1.1 Temperatura	61

6.9.1.2 Humedad.....	61
6.9.1.3 Radiación Solar.....	62
6.9.1.4 Precipitación.....	62
6.9.1.5 Velocidad del Viento.....	63
6.9.2 Selección del algoritmo de clasificación.....	64
7. RESULTADOS OBTENIDOS.....	66
7.1 Hardware.....	66
7.1.1 Nodo Sensor Simple (Sin respaldo de la información).....	66
7.1.2 Nodo Sensor Complejo (Con respaldo de la información).....	67
7.1.3 Sistema de Control de Riego.....	68
7.1.4 Nodo Coordinador, Gateway, Servidor Local.....	69
7.2 Machine Learning.....	71
7.2.1 Modelos de predicción.....	71
7.2.2 Sistema de Machine Learning.....	79
7.2.2.1 Catálogo de Servicios Web.....	79
7.2.2.2 Listados de los Servicios Web.....	80
7.3 Aplicación Web.....	82
7.4 Seguridad.....	97
8. CONCLUSIONES Y RECOMENDACIONES.....	100
9. REFERENCIAS.....	101
ANEXOS.....	103
ANEXO 1. CASOS DE USO.....	103
ANEXO 2 – DICCIONARIO DE DATOS.....	112
ANEXO 3 – DIAGRAMA DE CLASES.....	129

Lista de gráficas

Figura 2. La Internet de las Cosas nació entre 2008 y 2009	14
Figura 3. Tecnologías involucradas en la Agricultura Inteligente	20
Figura 4. Evapotranspiración de referencia (ET _o).....	23
Figura 5. Evapotranspiración del cultivo (ET _c)	23
Figura 6. Evapotranspiración bajo condiciones no estándar (ET _{c aj}).....	24
Figura 7. Procesos de evaporación, filtración profunda y escorrentía.....	26
Figura 8. Eficiencia de aplicación	27
Figura 9. Curvas real y teórica típicas de coeficiente de cultivo para especies anuales, según las diferentes fases de desarrollo.....	27
Figura 10. Duración de las etapas de crecimiento del cultivo para distintos períodos de siembra y regiones climáticas (días)	28
Figura 11. Valores del coeficiente único del cultivo, K _c y alturas medias máximas de las plantas para cultivos no estresados y bien manejados en climas sub-húmedos (HR _{min} ≈ 45%, u ₂ ≈ 2 m s ⁻¹) para usar en la fórmula de la FAO Penman-Monteith ET _o	29
Figura 12. Rangos de profundidad máxima efectiva de las raíces (Z _r) y fracción de agotamiento de la humedad en el suelo (p) para condiciones sin estrés hídrico, para cultivos comunes.....	29
Figura 13. Intervalo de Humedad Disponible (IHD) para diferentes tipos de suelo.	30
Figura 14. Representación esquemática del Nivel de Agotamiento Permisible.....	30
Figura 15. Representación esquemática del Déficit de Agua en el Suelo	31
Figura 16. Balance de agua en el sistema suelo-planta.....	31
Figura 17. Componentes del balance de agua.....	32
Figura 18. Estrategia de riego basada en aplicar las necesidades brutas cuando el DAS alcance el NAP	33
Figura 19. Estrategia de riego con la que se aplica una cantidad de agua fija.....	34
Figura 20. Riego establecido por turnos.....	35
Figura 21. Ciclo de bombo (despliegue publicitario) de Tecnologías Emergentes, 2015	37
Figura 22. Raspberry Pi 2.....	40
Figura 23. Arduino Uno	41
Figura 24. Arquitectura del Sistema de Riego Autónomo basado en la IoT	48
Figura 25. Tablas para dispositivos.....	54
Figura 26. Ejemplo de configuración de un nodo sensor	54
Figura 27. Consulta de los elementos de un dispositivo	54
Figura 28. Tablas para almacenar la información de las mediciones.....	55
Figura 29. Nodo Sensor con respaldo de la información.....	56
Figura 30. Nodo Coordinador – Gateway – Servidor Local	57
Figura 31. Componentes del Nodo de Control de Riego.....	58
Figura 35. Plot de los datos de Temperatura Ambiente	61

Figura 36. Plot de los datos de Humedad Ambiente	61
Figura 37. Plot de los datos de Radiación Solar.....	62
Figura 38. Plot de los datos de Precipitación	62
Figura 39. Plot de los datos de Velocidad del Viento	63
Figura 40. Plot de los datos de Evapotranspiración	63
Figura 41. Comparación de los Modelos Lineal y RBF	64
Figura 42. Comparación de los modelos SVR y KRR	65
Figura 43. Nodo Sensor Simple	66
Figura 44. Nodo Sensor Complejo	67
Figura 45. Sistema de Control de Riego	68
Figura 46. Control de las Electroválvulas	69
Figura 47. Nodo Coordinador, Gateway, Servidor Local.....	70
Figura 48. Modelo predicción de ETo (ejemplo estación Adra)	72
Figura 45. Visualización de datos de temperaturas diarias estación meteorología adra año 2001-2014	75
Figura 46. Visualización de predicción de temperaturas diarias del modelo para la estación meteorología adra año 2001-2014.....	75
Figura 47. Predicción de temperaturas diarias del modelo para la estación meteorología adra año 2001-2015	76
Figura 48. Residuos del modelo para la estación meteorología adra año 2001-2014	77
Figura 49. Estacionalidad ajustada del modelo para la estación meteorología adra año 2001-2014	77
Figura 50. Visualización de los componentes de la serie de tiempo para temperaturas diarias de la estación meteorología adra año 2001-2015.....	78
Figura 49. Modelo de predicción de caudal (ejemplo).....	78
Figura 50. Configuración del dominio de seguridad	97
Figura 51. Tablas para la seguridad.....	98
Figura 52. Consulta de usuarios y grupos.....	98
Figura 53. Ejemplo de restricción de seguridad	98
Figura 54. Configuración de la autenticación	99
Figura 55. Anotaciones para asegurar el acceso a los servicios web	99

Lista de tablas

Tabla 1. La Internet de las Cosas desde la perspectiva del desarrollo de Internet	15
Tabla 2. Eficiencia de aplicación (Ea) esperable con distintos métodos de riego	33
Tabla 3. Desarrollo por prototipos	47
Tabla 4. Catálogo de Servicios Web - Entidad	58
Tabla 5. Catálogo de Servicios Web – Bloque	58
Tabla 6. Catálogo de Servicios Web – Espacio.....	59
Tabla 7. Catálogo de Servicios Web – Dispositivo	59
Tabla 8. Catálogo de Servicios Web – Ítems por Dispositivo	59
Tabla 9. Catálogo de Servicios Web – Espacio para Cultivar	59
Tabla 10. Catálogo de Servicios Web – Calendario de Riego.....	60
Tabla 11. Catálogo de Servicios Web – Programación del Riego.....	60
Tabla 12. Catálogo de Servicios Web – Eventos del Riego	60
Tabla 13. Catálogo de Servicios Web del Sistema de Predicción.....	79

1. INTRODUCCION

La población mundial está en aumento, se requiere producir más alimentos y por lo tanto se necesitan mayores cantidades de agua para regar los cultivos. En muchos países existen problemas de escasez de agua, la cual debe ser tratada como un recurso valioso para la humanidad.

Para contribuir al uso eficiente del agua de manera sostenible, en este trabajo de investigación se ha construido un sistema de riego autónomo basado en la Internet de las Cosas (IoT).

Teniendo en cuenta que la mayor parte de los agricultores son de bajos recursos, en esta aproximación a la solución se emplean elementos de bajo costo y componentes de hardware - software libre (Raspberry Pi, Arduino, Linux, Java, Wildfly, Python, etc.).

Las características del sector agrícola hacen necesario implementar Redes de Sensores Inalámbricos (WSN) que permitan obtener la información de las variables agroclimáticas tales como humedad del suelo, temperatura ambiente, precipitación, etc. Para la construcción del prototipo inicial se emplearon los módulos de radio Xbee por su gran facilidad de configuración, pero se debe hacer un intento de emplear elementos de más bajo costo, teniendo en cuenta las características socio-económicas del sector.

Para abaratar los costos y brindar un soporte centralizado a la comunidad se emplearon los servicios de computación en la nube en la implementación del sistema de Aprendizaje Máquina (Machine Learning). Para la construcción de los modelos de predicción se emplearon datos de estaciones meteorológicas. Este sistema de predicción permite generar los calendarios de riego, ayudando de esta manera a responder las preguntas de cuándo y cuánto regar. También es posible implementar el sistema de riego en un solo Raspberry sin conexión a internet.

Para permitir la interoperabilidad entre los diferentes subsistemas (predicción, riego y cliente), hardware (Raspberry Pi, Xbee, computación en la nube) y software

(Python, Java), se implementaron Servicios Web tipo Restful. El formato Json se utilizó para enviar la información de los nodos sensores al nodo coordinador – recolector (Xbee).

Se implementó un modelo de seguridad basada en roles con el apoyo del Servicio de Autorización y Autenticación de Java (JAAS), y el uso de certificados digitales para la transmisión de la información por el protocolo https.

2. PROBLEMA

El problema que se pretende solucionar es:

“Como desarrollar un sistema de Internet de las Cosas autónomo (inteligente) que permita calcular las necesidades de agua de un cultivo y los horarios de riego para hacer un uso eficiente del agua.”

Las siguientes preguntas guiarán el proyecto de investigación:

- Cuáles son los parámetros agroclimáticos más significativos para calcular las necesidades de agua de un cultivo.
- Cuál es el método más apropiado para calcular la evapotranspiración.
- Como crear redes de sensores inalámbricos con elementos de hardware y software libre.
- Como comunicar de forma segura diferentes dispositivos y nodos utilizando servicios web.

3. OBJETIVOS

3.1 General

Desarrollar un sistema de riego autónomo basado en la Internet de las Cosas.

3.2 Específicos

- Determinar los parámetros agroclimáticos más significativos para hacer un uso eficiente del agua en el riego.
- Calcular la evapotranspiración utilizando los métodos de Penman-Monteith.
- Interconectar los diferentes tipos de dispositivos empleando diferentes protocolos (Serial, USB, Radio Frecuencia, ZigBee, WiFi, I2C, etc.).
- Crear un prototipo de Agricultura de Precisión con Hardware y Software Libre.
- Determinar los mecanismos de seguridad que permiten la comunicación segura en la Internet de las Cosas.

4. JUSTIFICACION

La necesidad de producir más alimentos para una población en aumento requiere que se automaticen las labores del campo y hacer un uso eficiente del agua en el proceso de riego de los cultivos. Para responder a estas necesidades, a los efectos adversos del cambio climático y la seguridad alimentaria, los sistemas de información se deben dotar de inteligencia y autonomía para que basados en datos provenientes de estaciones meteorológicas, modelos de predicción e información en tiempo real de las redes de sensores, se determinen las necesidades de agua de los cultivos y de esta manera se justifica el desarrollo de un sistema de riego autónomo basado en la Internet de las Cosas.

5. ESTADO DEL ARTE

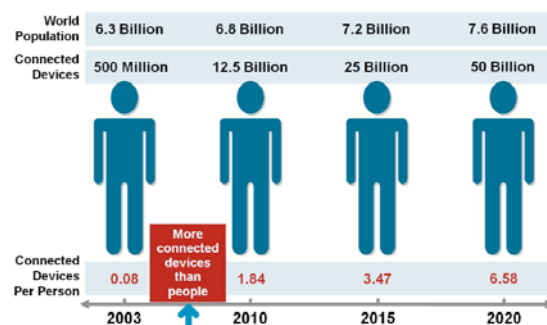
5.1 Internet de las Cosas

El término “Internet de las Cosas” fue oficialmente utilizado por primera vez en una publicación en 1999, donde Kevin Ashton publicó su visión en la revista RFID:

“Las ideas y la información son importantes, pero las cosas son mucho más importantes. Aun hoy la tecnología de la información es tan dependiente de los datos originados por la gente que nuestras computadoras conocen más acerca de ideas que cosas. Si tuviéramos computadoras que sabían todo lo que hay que saber acerca de las cosas - utilizando datos que ellas generan sin ninguna ayuda nuestra – seríamos capaces de rastrear y contar todo, y reducir grandemente los residuos, las pérdidas y los costos. Podríamos saber cuándo las cosas necesitan sustitución, reparación, o rellamado (recordatorio), y si estaban frescas o paso su mejor momento. **La Internet de las Cosas** tiene el potencial de cambiar el mundo, tal como lo hizo Internet. Tal vez aún mucho más.”(Sousa, 2015)

Se podría decir que la Internet de las Cosas (IoT) simplemente es el punto en el tiempo cuando más “cosas u objetos” que personas fueron conectados a la Internet. (Evans, 2011)

Figura 1. La Internet de las Cosas nació entre 2008 y 2009



Fuente: Cisco IBSG, Abril 2011

Desde la perspectiva del desarrollo de la Internet se puede considerar que la Internet de las Cosas como la tercera fase del desarrollo. En la primera fase, el uso

de la World Wide Web (WWW), que comenzó a mediados de la década de los 1990, proporciona a las personas un medio de recopilar y difundir información en todo el mundo. En la segunda fase, la Web 2.0 surgió a mediados de la década del 2000. Esta tecnología ha facilitado la creación de valor y comunidades a través de la colaboración entre las personas que utilizan Internet como un "lugar", así como las transacciones directas que involucran bienes y servicios. La tercera fase, el comienzo de la que estamos viendo ahora, comprende la "convergencia de Internet y el mundo real." Específicamente, se utilizan RFID y sensores para enviar datos sobre el estado y cualquier cambio del mundo real a Internet. El sistema reconoce automáticamente la totalidad del mundo real en tiempo real y crea la información que tiene valor a través del análisis de grandes cantidades de datos. Esta información se utiliza para controlar automáticamente los equipos y dispositivos. (Inoue, Hayakawa, & Kamei, 2011)

Tabla 1. La Internet de las Cosas desde la perspectiva del desarrollo de Internet

	Primera fase (A partir de mediados de los 1990) Propagación de WWW	Segunda fase (A partir de mediados de la década del 2000) Actualizar a Web 2.0	Tercera fase (A partir de finales de los años 2000) Uso práctico de IOT
Significado de la Internet	Medios de recolección y difusión de información	Lugar para la creación de valor y formación de la comunidad a través de la colaboración entre las personas, y las transacciones directas que involucran bienes y servicios	Medios para recolectar datos del mundo real mediante el uso de tecnologías de sensores; medios para controlar automáticamente equipos y dispositivos
Sujeto principal conectado a Internet	Gente	Gente	Cosas
Tecnología principal	WWW	Motor de búsqueda	Motor analítico para grandes cantidades de datos

Fuente: (Inoue et al., 2011)

No existe una definición universal de IoT, el concepto básico es que los objetos cotidianos puedan ser equipados con capacidades de identificación, sensado, red y procesamiento que les permitirá comunicarse con otros dispositivos y servicios a través de internet para alcanzar algún objetivo útil. (Whitmore, Agarwal, & Da Xu, 2015)

Basado en la revisión de la literatura se presentan algunas de las áreas de aplicación más comunes. (Miorandi, Sicari, De Pellegrini, & Chlamtac, 2012)

Casas Inteligentes / Edificios Inteligentes. La automatización del hogar – (Domótica) es cualquier cosa que su casa realiza de forma automática para hacer

que la vida allí sea más agradable o productiva. Una casa inteligente es aquella que parece aplicar la inteligencia para que esto suceda.(Goodwin, 2010)

En esta aplicación, los sensores son usados tanto para monitorear el consumo de los recursos (electricidad, agua) como para detectar proactivamente las necesidades de los usuarios actuales. Aquí se integra un gran número de subsistemas diferentes, y por lo tanto se requiere un nivel alto de estandarización para asegurar la interoperabilidad. También es necesaria la habilidad para razonar en un forma distribuida, cooperativa y actuar con el fin de asegurar que las decisiones tomadas sobre los recursos bajo control (ejemplo., prender/apagar luz, enfriar, calentar, etc.) se ajustan a las necesidades y expectativas de los usuarios, que a su vez están estrictamente entrelazadas con las actividades que realizan o planean realizar.

Ciudades Inteligentes. El término 'Ciudades Inteligentes' se utiliza para denotar el ecosistema cyberfísico emergente mediante el despliegue de infraestructura de comunicación avanzada y nuevos servicios a través de escenarios a nivel urbano. Por medio de servicios avanzados, es posible optimizar el uso de la infraestructura física de la ciudad (ejemplo, redes de carreteras, red eléctrica, etc.) y la calidad de vida para sus ciudadanos.

La IoT hace posible monitorear el flujo del tráfico vehicular en ciudades grandes o autopistas y desplegar servicios que ofrezcan consejo de enrutamiento del tráfico para evitar la congestión, retornar información agregada tal como promedio de velocidad y numero de carros.

Además, el sistema de dispositivos de parqueo inteligente, basado en tecnologías RFID y de sensores, puede permitir monitorear espacios de parqueo disponibles y proporcionar a los conductores consejos de parqueo automatizado, y así mejorar la movilidad en el área urbana.

Los sensores podrían detectar los niveles de polución del aire, retornar información de contaminación (humo) tal como el nivel de dióxido de carbono, PM10, etc., y suministrar tal información a las agencias de salud.

Asimismo, los sensores podrían ser usados en un entorno forense, al detectar violaciones y al transmitir los datos relevantes a los organismos policiales con el fin de identificar al violador, o para almacenar información que será proporcionada en caso de accidente para el análisis posterior de la escena del accidente.

Monitoreo ambiental. En las aplicaciones de monitoreo ambiental la tecnología IoT tiene la habilidad de sensor, de una forma distribuida y de auto-gestión, procesos y fenómenos naturales (ejemplo, temperatura, viento, lluvia, altura de los ríos), así como integrar perfectamente esos datos heterogéneos en aplicaciones globales.

Se puede construir una plataforma sólida para detectar y monitorear anomalías que puedan conducir a poner en peligro la vida humana y animal mediante el procesamiento de información en tiempo real, junto con la habilidad de un gran número de dispositivos para comunicarse entre ellos.

Para permitir el acceso a áreas críticas, donde la presencia de operadores humanos podría no representar una opción viable (ejemplo, áreas volcánicas, abismos oceánicos, áreas remotas), se puede hacer un gran despliegue de dispositivos miniaturizados desde donde la información sensada se pueda comunicar a un punto de decisión con el fin de detectar condiciones anómalas.

En la detección de fuego la respuesta rápida tiene la consecuencia de salvar vidas humanas, mitigar el daño a la propiedad o vegetación y en general reducir el nivel de desastre, aquí los sensores pueden enviar una alarma directamente al departamento de bomberos en un tiempo corto (explotando las características de comunicación avanzadas de la plataforma IoT), junto con otros parámetros que son útiles para la toma de decisiones y soporte, tales como la descripción de área objeto del fuego, la presencia posible de personas, de materiales inflamables, etc.

Cuidados de Salud. En el sector cuidado de la salud se pueden encontrar un gran número de aplicaciones. También, se pueden utilizar para mejorar las soluciones de vida asistida actuales. Los pacientes llevarán sensores médicos para monitorear parámetros tales como temperatura corporal, presión sanguínea, actividad

respiratoria. Otros sensores, ya sean vestibles (ejemplo, acelerómetros, giroscopios) o fijos (proximidad) serán usados para obtener datos empleados para monitorear actividades de los pacientes en sus ambientes de vida. La información será agregada localmente y transmitida a los centros médicos remotos, los cuales estarán en capacidad de ejecutar monitoreo remoto avanzado y serán capaces de acciones de respuesta rápida cuando sea necesario. La interconexión de tales sensores heterogéneos podría proporcionar una imagen completa de los parámetros de salud, iniciando así una intervención del personal médico al detectar condiciones que podrían conducir a deterioros de la salud, realizando de esta manera cuidados preventivos.

Las soluciones de cuidado de salud personalizado y de bienestar son otro sector de aplicación relevante. El uso de sensores vestibles, junto con aplicaciones apropiadas que se ejecutan sobre dispositivos de computación personal posibilitan a la gente rastrear sus actividades diarias (pasos caminados, calorías quemadas, ejercicios ejecutados, etc.), proporcionando sugerencias para mejorar su estilo de vida y prevenir la aparición de problemas de salud.

Negocios Inteligentes / Inventario y Gestión de Productos. La capacidad de las tecnologías RFID para identificar y proporcionar soporte para rastrear bienes, permiten que sean utilizadas en muchos sectores para el manejo de inventarios, a través de la cadena de suministros y distribución.

Las tecnologías IoT posibilitan interoperabilidad continua entre las aplicaciones basadas en RFID utilizadas por diferentes actores relacionados con el producto a lo largo de las distintas fases de su ciclo de vida y pueden proporcionar flexibilidad mejorada en términos de posiciones de lectores. Normalmente, las etiquetas RFID se pegan directamente a los ítems (o a los contenedores que las llevan), mientras los lectores se ubican en toda la instalación a monitorear.

En aplicaciones al detal (de tiendas), se pueden utilizar las tecnologías IoT para monitorear en tiempo real la disponibilidad de productos y mantener un inventario de existencias exacto.

En el soporte postventa pueden jugar un papel importante, los usuarios pueden retornar automáticamente todos los datos acerca de los productos que ellos compraron. Las tecnologías de identificación proporcionan a los productos un identificador único incluyendo una completa y confiable descripción del bien en sí y por lo tanto pueden ayudar en la limitación de los robos y en la lucha contra la falsificación.

En la industria de alimentos, por ejemplo, los dispositivos RFID se pueden utilizar para identificar y rastrear los productos, mientras los bio-sensores pueden monitorear parámetros tales como temperatura y composición bacterial con el fin de permitir controlar los procesos de producción, garantizar la calidad requerida del producto final y el posible deterioro de la vida del producto.

Seguridad y vigilancia. En los edificios empresariales, centros comerciales, fábricas, parqueaderos de carros, etc. la vigilancia de la seguridad se ha convertido en una necesidad. Se pueden utilizar sensores ambientales para monitorear la presencia de químicos peligrosos.

Los sistemas de advertencia temprana pueden evaluar la presencia de personas que actúan de manera sospechosa, mediante el uso de sensores que monitorean el comportamiento de la gente.

Los sistemas control de acceso basado en roles en conjunto con las tecnologías IoT pueden proporcionar un nivel alto de flexibilidad, siendo capaces de afrontar las políticas de acceso (ejemplo, a diferentes áreas de los edificios) las cuales pueden cambiar sobre el tiempo debido a cambios logísticos y / o a cambios en el rol del usuario y / o de acuerdo a información de contexto (ejemplo, algunas áreas no accesibles en un día determinado debido a trabajos de mantenimiento en ejecución).

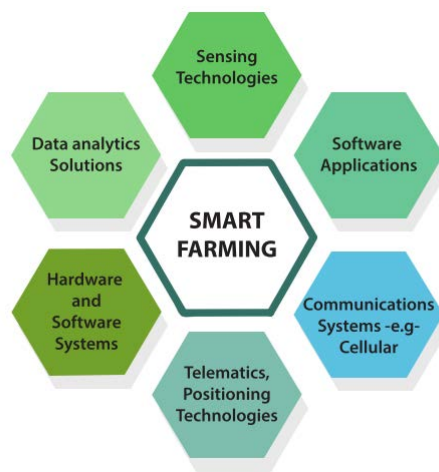
Agricultura Inteligente

La agricultura de precisión es a veces conocida como "agricultura inteligente" (Smart farming), un término general para facilitar la comparación con otras implementaciones basadas en IoT tales como la medición inteligente, ciudades

inteligentes y así sucesivamente. La agricultura de precisión tiene como objetivo optimizar el rendimiento por unidad de tierra de cultivo mediante el uso de los medios más modernos (servicios GPS, sensores y big data) de una manera continua sostenible, para lograr lo mejor en términos de calidad, cantidad y rentabilidad financiera.(Beecham Research, 2014)

Los sistemas de apoyo de decisiones basadas en TIC, respaldados por datos en tiempo real, pueden proporcionar, además, información sobre todos los aspectos de la agricultura a un nivel de granularidad no era posible anteriormente. Esto permite tomar mejores decisiones, lo que resulta en menos desperdicio y máxima eficiencia en las operaciones.

Figura 2. Tecnologías involucradas en la Agricultura Inteligente



Fuente: (Beecham Research, 2014)

Algunas de las aplicaciones de la Agricultura Inteligente son:

- Gestión - seguimiento de la flota de vehículos agrícolas
- Cultivo del campo, grandes y pequeños campos de cultivo
- Monitoreo de la ganadería
- Agricultura en Interiores - invernaderos y establos
- El cultivo de peces
- Silvicultura, Ciencias forestales
- Monitoreo de almacenamiento - tanques de agua, tanques de combustible

5.2 Programación del Riego

Entre ahora y 2050, la población del mundo incrementará por un tercio. Muchos de los adicionales 2 billones de personas vivirán en países en desarrollo. Al mismo tiempo, más personas vivirán en las ciudades (70 por ciento contra el 50 por ciento actual). La urbanización y el incremento de los ingresos en los países en desarrollo están conduciendo a aumentos en el consumo de productos animales (FAO, 2009a). Dadas estas tendencias la FAO estima que la producción tendrá que aumentar el 60 por ciento para el 2050 para satisfacer las demandas esperadas por alimentos y piensos (porción de alimento seco que se da al ganado.) (Conforti, 2011). (Food and Agriculture Organization, 2013)

Los aumentos de esta demanda se pueden satisfacer de tres formas:

- Elevando la productividad agrícola.
- Aumentando la superficie cultivable.
- Incrementando la intensidad de cultivo (número de cultivos por año).

En los últimos 30 años, la mayor parte del crecimiento — más de las tres cuartas partes — se debió al incremento de la productividad, principalmente como consecuencia de la Revolución Verde. En los países en desarrollo, durante los próximos 30 años, se espera que así sea: que el 69 por ciento del aumento de la producción proceda del incremento de la productividad; el 12 por ciento del aumento de la intensidad de cultivo y el resto de la extensión de la superficie cultivable. (FAO, 2002)

Evapotranspiración

La evapotranspiración (ET) está compuesta por los procesos de evaporación directa de agua desde el suelo y la transpiración de las plantas y se utiliza para determinar las necesidades de agua de los cultivos.

Evaporación

La evaporación es el proceso por el cual el agua líquida se convierte en vapor de agua (vaporización) y se retira de la superficie evaporante (remoción de vapor). El

agua se evapora de una variedad de superficies, tales como lagos, ríos, caminos, suelos y la vegetación mojada. (Allen G., Pereira, Raes, & Smith, 2006)

Transpiración

La transpiración consiste en la vaporización del agua líquida contenida en los tejidos de la planta y su posterior remoción hacia la atmósfera. El agua, junto con algunos nutrientes, es absorbida por las raíces y transportada a través de la planta. La vaporización ocurre dentro de la hoja, en los espacios intercelulares, y el intercambio del vapor con la atmósfera es controlado por la abertura estomática. Casi toda el agua absorbida del suelo se pierde por transpiración y solamente una pequeña fracción se convierte en parte de los tejidos vegetales. (Allen G. et al., 2006)

La evapotranspiración se expresa normalmente en milímetros (mm) por unidad de tiempo. Esta unidad expresa la cantidad de agua perdida de una superficie cultivada en unidades de altura de agua. La unidad de tiempo puede ser una hora, día, 10 días, mes o incluso un completo período de cultivo o un año. Como una hectárea tiene una superficie de 10 000 m² y 1 milímetro es igual a 0,001 m, una pérdida de 1 mm de agua corresponde a una pérdida de 10 m³ de agua por hectárea. Es decir 1 mm día⁻¹ es equivalente 10 m³ ha⁻¹ día⁻¹. (Allen G. et al., 2006)

Variables climáticas

Los principales parámetros climáticos que afectan la evapotranspiración son la radiación solar, la temperatura del aire, la humedad atmosférica y la velocidad del viento. Se han desarrollado varios procedimientos para determinar la evapotranspiración a partir de estos parámetros.

Conceptos de evapotranspiración

El concepto de evapotranspiración incluye tres diferentes definiciones: evapotranspiración del cultivo de referencia (ET_o), evapotranspiración del cultivo bajo condiciones estándar (ET_c), y evapotranspiración del cultivo bajo condiciones no estándar (ET_{c aj}). (Allen G. et al., 2006)

Evapotranspiración del cultivo de referencia (ET_o)

La tasa de evapotranspiración de una superficie de referencia, que ocurre sin restricciones de agua, se conoce como evapotranspiración del cultivo de referencia, y se denomina ET_o.

Figura 3. Evapotranspiración de referencia (ET_o)

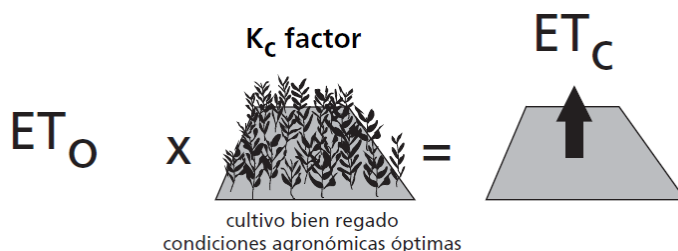


Fuente: (Allen G. et al., 2006)

Evapotranspiración del cultivo bajo condiciones estándar (ET_c)

La evapotranspiración del cultivo bajo condiciones estándar se denomina ET_c, y se refiere a la evapotranspiración de cualquier cultivo cuando se encuentra exento de enfermedades, con buena fertilización y que se desarrolla en parcelas amplias, bajo óptimas condiciones de suelo y agua, y que alcanza la máxima producción de acuerdo a las condiciones climáticas reinantes.

Figura 4. Evapotranspiración del cultivo (ET_c)



Fuente: (Allen G. et al., 2006)

La cantidad de agua requerida para compensar la pérdida por evapotranspiración del cultivo se define como necesidades de agua del cultivo. A pesar de que los valores de la evapotranspiración del cultivo y de las necesidades de agua del cultivo son idénticos, sus definiciones conceptuales son diferentes. Las necesidades de agua del cultivo se refieren a la cantidad de agua que necesita ser proporcionada al

cultivo como riego o precipitación, mientras que la evapotranspiración del cultivo se refiere a la cantidad de agua perdida a través de la evapotranspiración. La necesidad de riego básicamente representa la diferencia entre la necesidad de agua del cultivo y la precipitación efectiva.

Evapotranspiración del cultivo bajo condiciones no estándar (ET_{c aj})

La evapotranspiración del cultivo bajo condiciones no estándar (ET_{c aj}) se refiere a la evapotranspiración de cultivos que crecen bajo condiciones ambientales y de manejo diferentes de las condiciones estándar. Bajo condiciones de campo, la evapotranspiración real del cultivo puede desviarse de ET_c debido a condiciones no óptimas como son la presencia de plagas y enfermedades, salinidad del suelo, baja fertilidad del suelo y limitación o exceso de agua. Esto puede resultar en un reducido crecimiento de las plantas, menor densidad de plantas y así reducir la tasa de evapotranspiración por debajo de los valores de ET_c.

Figura 5. Evapotranspiración bajo condiciones no estándar (ET_{c aj})

$$ET_0 \times \begin{matrix} K_s \times K_c \text{ ajustado} \\ \text{estres hídrico y ambiental} \end{matrix} = ET_{c \text{ aj}}$$

Fuente: (Allen G. et al., 2006)

La evapotranspiración del cultivo bajo condiciones no estándar se calcula utilizando un coeficiente de estrés hídrico K_s o ajustando K_c a todos los otros tipos de condiciones de estrés y limitaciones ambientales en la evapotranspiración del cultivo.

ET calculada con datos meteorológicos

Como resultado de una Consulta de expertos llevada a cabo en mayo de 1990, el **método FAO Penman-Monteith** se recomienda actualmente como el único método estándar para la definición y el cálculo de la evapotranspiración de referencia en base a información meteorológica (radiación solar, temperatura del aire, humedad

atmosférica y velocidad del viento), ET_0 . Los datos diarios de temperatura máxima y mínima del aire son los requerimientos mínimos de datos necesarios para aplicar el método FAO Penman-Monteith de estimación de ET_0 . La ET del cultivo bajo condiciones estándar se determina utilizando los coeficientes de cultivo (K_c) que relacionan la ET_c con la ET_0 . (Allen G. et al., 2006)

$$ET_0 = \frac{0.408\Delta(R_n - G) + \gamma \frac{900}{T + 273} u_2 (e_s - e_a)}{\Delta + \gamma(1 + 0.34u_2)} \quad (1)$$

Donde:

ET_0	evapotranspiración de referencia (mm día-1)
R_n	radiación neta en la superficie del cultivo (MJ m-2 día-1)
R_a	radiación extraterrestre (mm día-1)
G	flujo del calor de suelo (MJ m-2 día-1)
T	temperatura media del aire a 2 m de altura (°C)
u_2	velocidad del viento a 2 m de altura (m s-1)
e_s	presión de vapor de saturación (kPa)
e_a	presión real de vapor (kPa)
$e_s - e_a$	déficit de presión de vapor (kPa)
Δ	pendiente de la curva de presión de vapor (kPa °C-1)
γ	constante psicométrica (kPa °C-1)

Pérdidas de agua en el suelo: escorrentía, filtración profunda y evaporación

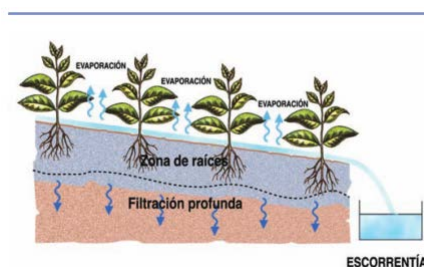
Un suelo es un almacén de agua. Sin embargo, la cantidad de agua almacenada cambia con el tiempo debido a que las demandas varían mucho dependiendo de las condiciones climáticas, el estado de desarrollo del cultivo y de las prácticas de riego. Los aportes de agua al suelo son la lluvia y el riego, sin embargo no toda el agua aportada es almacenada y puesta a disposición de las plantas, sino que se producen pérdidas debido a los siguientes fenómenos:

- **Escorrentía:** representa la cantidad de agua de lluvia o de riego que cae sobre la superficie del suelo pero que éste no puede infiltrar. Así, el agua sobrante escurre sobre él sin ser aprovechada por el cultivo. La escorrentía puede ser grande en algunos sistemas de riego por superficie

(principalmente riego por surcos), sin embargo no suele ser frecuente que se produzca en riegos por aspersión bien diseñados y manejados. Por lo general, en riego localizado no se produce escorrentía.

- **Filtración profunda o percolación:** cuando el agua aplicada sobre la superficie del suelo se infiltra, pasa poco a poco hacia capas más profundas. Si la cantidad de agua aplicada es mayor que la capacidad de retención, el agua infiltrará hacia zonas en las que las raíces del cultivo no pueden acceder, siendo por lo tanto agua perdida.
- **Evaporación:** es el proceso por el cual el agua pasa de la superficie del suelo a la atmósfera en forma de vapor. La evaporación es tanto más intensa cuanto más seco sea el ambiente y mayor la temperatura del aire, es decir, la demanda evaporativa sea mayor; también será mayor cuanto más húmedo esté el suelo en superficie ya que el agua estará más disponible para ser evaporada y cuanto mayor sea el viento reinante en la zona. (Fernandez Gomez, Avila Alabarces, Lopez Rodriguez, Gavilan Zafra, & Oyonarte Gutierrez, 2010)

Figura 6. Procesos de evaporación, filtración profunda y escorrentía

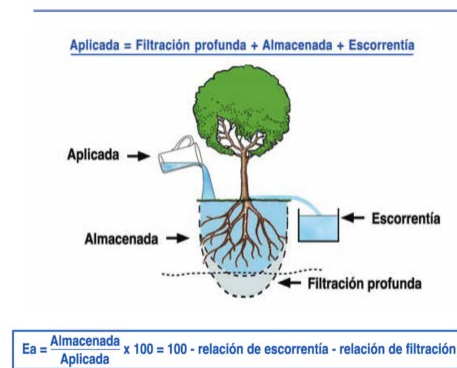


Fuente; Fundamentos del Riego (Fernandez Gomez et al., 2010)

La eficiencia de aplicación (Ea)

La eficiencia de aplicación es la relación entre el agua que realmente queda almacenada (Almacenada) en la zona de raíces del cultivo (y por lo tanto podrá ser aprovechada por ellas) y el agua total aplicada con el riego (Aplicada).

Figura 7. Eficiencia de aplicación

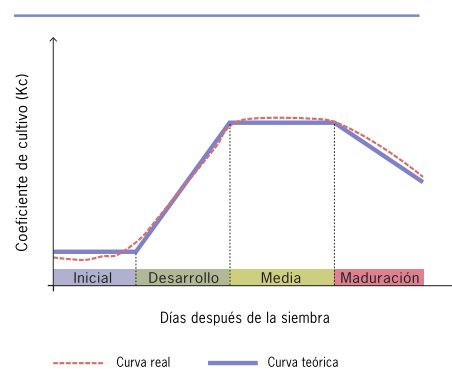


Fuente: Fundamentos del Riego (Fernandez Gomez et al., 2010)

Coeficiente de cultivo

El coeficiente de cultivo (K_c) describe las variaciones en la cantidad de agua que las plantas extraen del suelo a medida que se van desarrollando, desde la siembra hasta la recolección.

Figura 8. Curvas real y teórica típicas de coeficiente de cultivo para especies anuales, según las diferentes fases de desarrollo



Fuente: Fundamentos del Riego (Fernandez Gomez et al., 2010)

En los cultivos anuales normalmente se diferencian cuatro etapas o fases del cultivo:

1. **Inicial:** desde la siembra hasta un 10% de cobertura del suelo aproximadamente.
2. **Desarrollo:** desde el 10% de cobertura y durante el crecimiento activo de la planta.
3. **Media:** entre floración y fructificación, correspondiente en la mayoría de los casos al 70–80% de cobertura máxima de cada cultivo.
4. **Maduración:** desde madurez hasta recolección.(Fernandez Gomez et al., 2010)

Duración de las etapas de crecimiento

La publicación de la FAO – “Evapotranspiración del cultivo: Guías para la determinación de los requerimientos de agua de los cultivos” incluye duraciones generales para las cuatro etapas de crecimiento de distintos cultivos, así como la duración total de la temporada de crecimiento de cada cultivo, para distintos tipos de clima y diferentes localidades. (Allen G. et al., 2006)

Figura 9. Duración de las etapas de crecimiento del cultivo para distintos períodos de siembra y regiones climáticas (días)

Cultivo	Inic. (L _{in})	Des. (L _{des})	Med (L _{med})	Final (L _{fin})	Total	Fecha de Siembra	Región
Cebada/Avena/Trigo	15	25	50	30	120	Noviembre	India Central
	20	25	60	30	135	Marzo/Abril	35-45 °L
	15	30	65	40	150	Julio	Este de África
	40	30	40	20	130	Abril	
	40	60	60	40	200	Noviembre	
Maíz (grano)	20	50	60	30	160	Diciembre	
	30	50	60	40	180	Abril	Este de África (alt.)
	25	40	45	30	140	Dic./Ene.	Clima Árido
	20	35	40	30	125	Junio	Nigeria (húmedo)
	20	35	40	30	125	Octubre	India (seco, frío)
Maíz (dulce)	30	40	50	30	150	Abril	España (prim,ver); Calif. EU
	30	40	50	50	170	Abril	Idaho, EU
	20	20	30	10	80	Marzo	Filipinas
Maíz (dulce)	20	25	25	10	80	May./Junio	Mediterráneo
	20	30	50/30	10	90	Oct./Dic.	Clima Árido
	30	30	30	10 ²	110	Abril	Idaho, EU
	20	40	70	10	140	Ene.	Calif. Desierto, EU

Fuente: (Allen G. et al., 2006)

Determinación numérica del valor de Kc

Se puede determinar el valor del coeficiente Kc, para un período determinado de la temporada de crecimiento, considerando que el valor de Kc durante la etapa inicial y la etapa de mediados de temporada es constante e igual al valor de Kc durante la etapa considerada. Durante las etapas de desarrollo y final, el valor de Kc varía en forma lineal entre el valor de Kc correspondiente a la etapa previa (Kc prev) y el valor de Kc al comienzo de la próxima etapa (Kc prox), el cual corresponde al Kc fin en el caso de la etapa final: (Allen G. et al., 2006)

$$K_{ci} = K_{c\ prev} + \left[\frac{i - \sum(L_{prev})}{L_{etapa}} \right] (K_{c\ prox} - K_{c\ prev}) \quad (2)$$

Coeficientes del cultivo

En la publicación de la FAO – “Evapotranspiración del cultivo” se incluyen valores típicos de K_c ini, K_c med y K_c fin correspondientes a distintos cultivos. (Allen G. et al., 2006)

Figura 10. Valores del coeficiente único del cultivo, K_c y alturas medias máximas de las plantas para cultivos no estresados y bien manejados en climas sub-húmedos ($HR_{min} \approx 45\%$, $u_2 \approx 2 \text{ m s}^{-1}$) para usar en la fórmula de la FAO Penman-Monteith E_{To} .

Cultivo	$K_{c_{ini}}$ ¹	$K_{c_{med}}$	$K_{c_{fin}}$	Altura Máx. Cultivo (h) (m)
i. Cereales	0,3	1,15	0,4	
Cebada		1,15	0,25	1
Avena		1,15	0,25	1
Trigo de Primavera		1,15	0,25-0,4 ¹⁰	1
Trigo de Invierno		1,15	0,25-0,4 ¹⁰	1
– con suelos congelados	0,4	1,15	0,25-0,4 ¹⁰	1
– con suelos no-congelados	0,7	1,15	0,25-0,4 ¹⁰	1
Maíz, (grano)		1,20	0,60,0,35 ¹¹	2
Maíz, (dulce)		1,15	1,05 ¹²	1,5
Mijo		1,00	0,30	1,5
Sorgo		1,00-1,10	0,55	1-2
– grano		1,20	1,05	2-4
– dulce				
Arroz	1,05	1,20	0,90-0,60	1

Fuente: (Allen G. et al., 2006)

El agua del suelo en relación con el riego

Cuando se trata de calcular el agua que es preciso aportar con el riego, se debe conocer la profundidad de la capa de suelo que es realmente ocupada por las raíces. Algunos valores de profundidad de las raíces máxima para diferentes cultivos se exponen en la tabla siguiente.

Figura 11. Rangos de profundidad máxima efectiva de las raíces (Z_r) y fracción de agotamiento de la humedad en el suelo (p) para condiciones sin estrés hídrico, para cultivos comunes

Cultivo	Profundidad radicular máxima ¹ (m)	Fracción de agotamiento ² (para $ET \approx 5 \text{ mm día}^{-1}$) p
i. Cereales		
Cebada	1,0-1,5	0,55
Avena	1,0-1,5	0,55
Trigo de Primavera	1,0-1,5	0,55
Trigo de Invierno	1,5-1,8	0,55
Maíz, (grano)	1,0-1,7	0,55
Maíz, (dulce)	0,8-1,2	0,50
Mijo	1,0-2,0	0,55
Sorgo	1,0-2,0	0,55
– grano	1,0-2,0	0,55
– dulce	1,0-2,0	0,50

Fuente: (Allen G. et al., 2006)

El **Intervalo de Humedad Disponible (IHD)** es la cantidad de agua del suelo que teóricamente está a disposición para las plantas y viene determinado por la diferencia entre el límite superior e inferior de humedad, es diferente para cada textura del suelo.

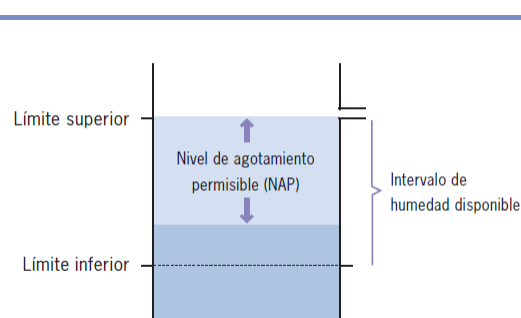
Figura 12. Intervalo de Humedad Disponible (IHD) para diferentes tipos de suelo.

Textura (mm de agua por m de profundidad del suelo)	IHD
Arenoso	70 – 100
Franco-arenoso	90 – 150
Franco	140 – 190
Franco-arcilloso	170 – 220
Arcilloso	200 – 250

Fuente: Fundamentos del Riego (Fernandez Gomez et al., 2010)

El **Nivel de Agotamiento Permisible (NAP)** representa un nivel de humedad entre el límite superior y el inferior a partir del cual las raíces encuentran **mayor dificultad para extraer el agua**, aunque las plantas pueden extraer agua del suelo hasta un nivel de humedad que corresponde con el límite inferior. Produce una disminución en la transpiración, lo que suele traer consigo pérdidas de producción, normalmente corresponde a una fracción del Intervalo de Humedad Disponible (IHD).

Figura 13. Representación esquemática del Nivel de Agotamiento Permisible



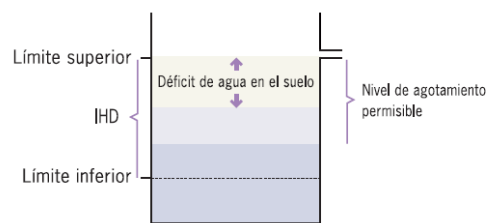
Fuente: Fundamentos del Riego (Fernandez Gomez et al., 2010)

La humedad correspondiente al Nivel de Agotamiento Permisible es la cantidad de agua que el suelo debería tener siempre, como mínimo, para que la producción fuera siempre la máxima posible. En programación de riegos suele emplearse muy frecuentemente un valor entre 0.6 y 0.8 (un valor de 0.65 se considera muy

adecuado y es utilizado con frecuencia), pero en cultivos de alto valor económico, como por ejemplo los hortícolas, no debe usarse un valor de NAP mayor de 0.5 para asegurar que el cultivo no sufrirá en ningún momento falta de agua y ello pueda disminuir la producción.

El **Déficit de Agua en el Suelo (DAS)** es la cantidad de agua que va faltando con respecto al límite superior y va aumentando a medida que pasa el tiempo debido a la salida de agua por la evapotranspiración (ET), con lo cual se va agotando el agua del suelo día a día.

Figura 14. Representación esquemática del Déficit de Agua en el Suelo

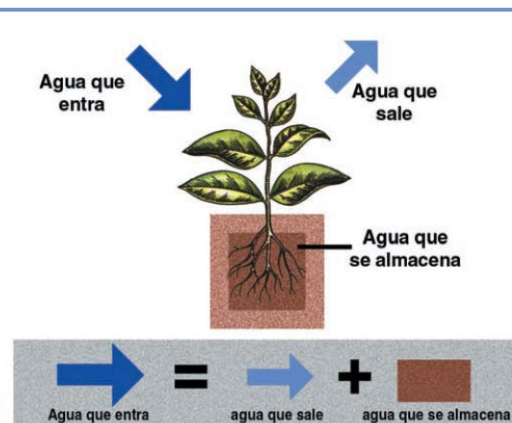


Fuente: Fundamentos del Riego (Fernandez Gomez et al., 2010)

Estimación de las necesidades de riego usando el método del balance de agua

El sistema formado por el suelo y el cultivo tiene unos aportes y unas salidas de agua. Por lo general esas cantidades no son iguales, por lo **que el contenido de humedad del suelo irá cambiando**, quedando de manifiesto el papel del suelo como almacén de agua.

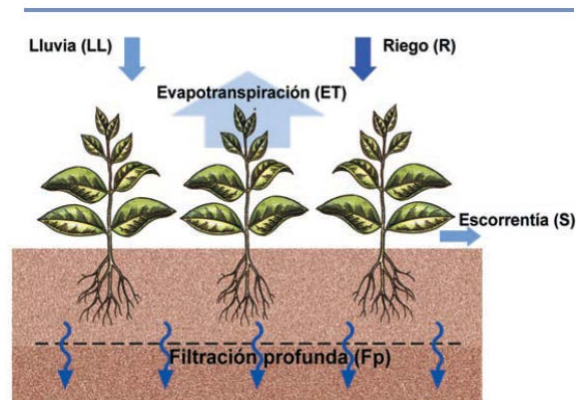
Figura 15. Balance de agua en el sistema suelo-planta



Fuente: Fundamentos del Riego (Fernandez Gomez et al., 2010)

Las entradas de agua pueden ser debidas a la lluvia (LL) o al riego (R). Por su parte, las salidas de agua se deberán a la *evapotranspiración* (ET), la *escorrentía* (S) o la *filtración profunda* (Fp).

Figura 16. Componentes del balance de agua



Fuente: Fundamentos del Riego (Fernandez Gomez et al., 2010)

Las **Necesidades netas de riego (Nn)** se definen como la cantidad de agua que necesita el cultivo y se ha de aportar con el riego, y corresponderán a la diferencia entre la cantidad de agua que el conjunto suelo-planta pierde, la evapotranspiración (ET), y el agua que se aporta de forma natural, la lluvia (LL). Todo esto si se considera un **sistema de riego bien diseñado** en el que no existe escorrentía, $S=0$. Y además, suponiendo que la filtración profunda sea nula, $Fp=0$.

$$\text{Necesidades netas de riego} = \text{Evapotranspiración} - \text{Lluvia} \quad (3)$$

Esta cantidad de agua, se denomina **lámina de agua requerida** y se expresa en altura de la lámina de agua por metro cuadrado de superficie de suelo.

La eficiencia *de aplicación del riego* (E_a), es diferente para cada método de riego, superficie, aspersión y localizado y dentro de cada uno de ellos, según cada sistema. Además no toda el agua que se aporta al suelo con un riego es aprovechada por las raíces del cultivo, sino que parte se pierde por escorrentía y/o filtración profunda.

Tabla 2. Eficiencia de aplicación (Ea) esperable con distintos métodos de riego

Método de riego	Eficiencia de aplicación (%)
Riego por superficie	55 – 90
Riego por aspersión	65 – 90
Riego localizado	75 – 90

Fuente: Fundamentos del Riego (Fernandez Gomez et al., 2010)

Las **necesidades brutas de riego (Nb)** se definen como la cantidad real de agua que ha de aplicarse durante el riego para satisfacer las necesidades netas de riego. Se pueden determinar si se conoce la eficiencia de aplicación, utilizando la fórmula:

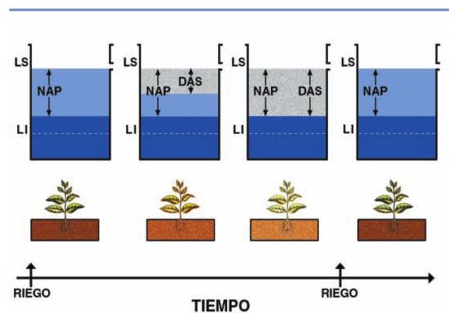
$$Necesidades\ brutas\ de\ riego = \frac{Necesidades\ netas\ de\ riego}{Eficiencia\ de\ aplicación\ del\ riego} \times 100 \quad (3)$$

Estrategias de riego

Las estrategias de riego se pueden entender como criterios para decidir el momento de efectuar un riego y la cantidad de agua a aplicar.

1. Un criterio general es aplicar el riego cuando el **Déficit de Agua en el Suelo (DAS)** sea igual al **Nivel de Agotamiento Permisible (NAP)**, aplicando las **necesidades brutas de riego (Nb)**. Teniendo en cuenta estrictamente el balance de agua (agua que se aporta al sistema suelo-planta menos agua que se extrae del sistema) es la estrategia más recomendable, ya que permite que no haya problemas de extracción de agua y que la producción final no se vea afectada, aplicando el menor número posible de riegos. (Fernandez Gomez et al., 2010)

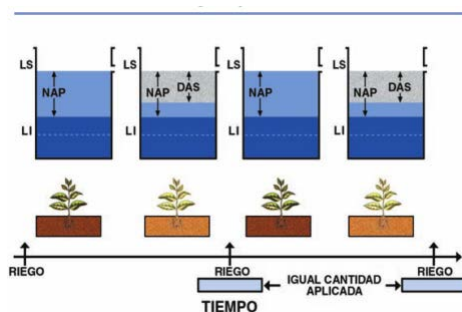
Figura 17. Estrategia de riego basada en aplicar las necesidades brutas cuando el DAS alcance el NAP



Fuente: Fundamentos del Riego (Fernandez Gomez et al., 2010)

2. Si el valor comercial del cultivo es muy alto, es conveniente asegurarse que las raíces de las plantas no tengan problema en extraer el agua en ningún momento, aplicando las necesidades brutas de riego antes de que el DAS alcance el NAP. De esta manera se aumenta el número de riegos y, dependiendo del método de riego empleado, su coste.
3. En ocasiones es conveniente aplicar una cantidad de agua fija con los riegos, de manera que se aproveche al máximo el sistema de riego. Los sistemas automecanizados de riego por aspersión (por ejemplo el pivotante, más conocido por "pívot") son un claro ejemplo de aplicación de una cantidad fija, que depende de la velocidad a la que se desplace la máquina. En estos casos, el momento de realizar el riego es aquel en que el Déficit de Agua en el Suelo iguale a las necesidades netas, pero teniendo en cuenta que se aplicarán las necesidades brutas.

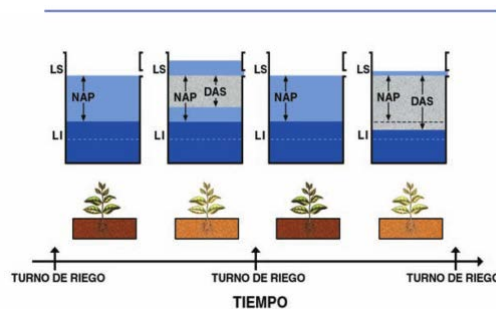
Figura 18. Estrategia de riego con la que se aplica una cantidad de agua fija



Fuente: Fundamentos del Riego (Fernandez Gomez et al., 2010)

4. En numerosos sistemas de riego (principalmente en riego por superficie) existen restricciones para elegir el momento de riego ya están organizados por turnos en los que cada agricultor riega cuando le está permitido. En este caso puede ser que el Déficit del Agua en el Suelo supere al Nivel de Agotamiento Permisible, lo que es indeseable. Lo más usual es que en estos casos el agricultor procure aplicar el agua correspondiente a las necesidades brutas, es decir cargar el suelo de agua en previsión de que el turno de riego se pueda retrasar.

Figura 19. Riego establecido por turnos



Fuente: Fundamentos del Riego (Fernandez Gomez et al., 2010)

Programación de los riegos

Para hacer un uso eficiente del agua debemos responder a las preguntas de **cuándo se ha de regar** y **cuánta agua aplicar**, para ello debemos conocer las características del cultivo, las características físicas del suelo y las condiciones climáticas de la zona.

La influencia del cultivo es importante puesto que las necesidades de agua serán mayores o menores en función del tipo de planta y de su estado de desarrollo. De la misma forma, las raíces de un cultivo ocupan diferente profundidad del suelo en distintas fases dentro del ciclo por lo que la cantidad de agua disponible en esa zona de suelo varía con el estado del cultivo. La capacidad de cada suelo para retener agua también es diferente lo que implica que tanto la cantidad de agua a aplicar con el riego como la que pueden extraer las plantas puede variar mucho. A ello hay que añadir que las necesidades de agua serán también dependientes del clima, radiación solar, viento, precipitación, etc., por lo que es preciso conocer las características climáticas de la zona y del cultivo para programar adecuadamente los riegos. (Fernandez Gomez et al., 2010)

Calendarios medios de riego. Programación en tiempo real

Las estrategias de riego son unos criterios generales, que se concretan elaborando un calendario medio de riegos en el que se precisa el momento del riego y la cantidad de agua que se aplica en cada uno de ellos.

Contando con datos del cultivo, suelo y clima, se puede establecer un calendario medio de riegos asumiendo el caso más simple, en el que se supone que la lluvia

es nula durante el ciclo del cultivo y que los valores de evapotranspiración de referencia son los de la media de los últimos años, lo que suele producirse en cultivos de primavera-verano en zonas semiáridas. Para ello es preciso contar con datos de:

- Evapotranspiración de referencia (ET_r) en la zona.
- Coeficiente de cultivo (K_c) del cultivo a regar en distintas fases de desarrollo de éste.
- Profundidad radicular media en distintas fases del cultivo.
- Intervalo de humedad disponible del suelo.
- Nivel de agotamiento permisible para el cultivo en cuestión.
- Datos diversos del sistema de riego como por ejemplo la eficiencia.

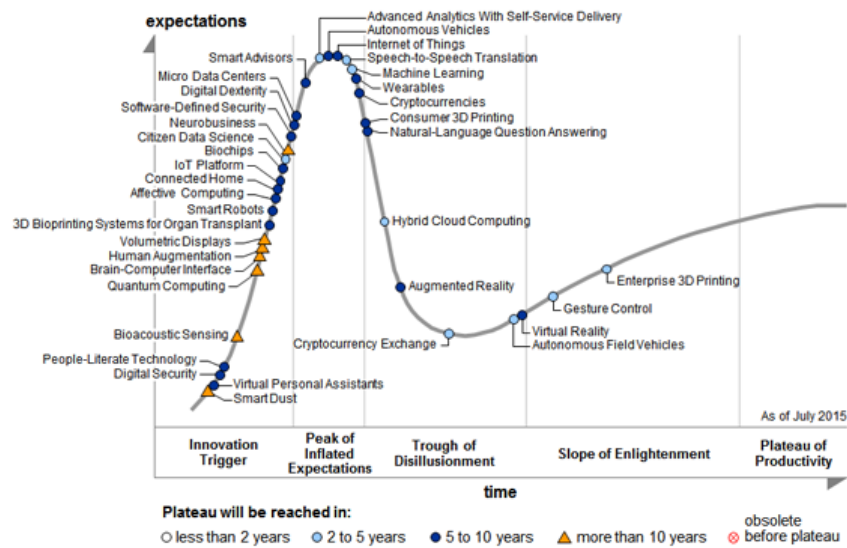
5.3 Autonomía – Machine Learning

De acuerdo con el “Ciclo de Bombo (despliegue publicitario) para las Tecnologías Emergentes” de Gartner (2015), para las empresas que emprenden el viaje para convertirse en negocios digitales, será crítico identificar y emplear las tecnologías correctas en el momento adecuado. La hoja de ruta de Gartner para los negocios digitales incluye seis modelos progresivos de la era de negocios con los que las empresas pueden identificarse hoy y aspirar a alcanzarlos en el futuro:

- Etapa 1: Análoga
- Etapa 2: Web
- Etapa 3: Negocios Electrónicos
- Etapa 4: Mercadeo Digital
- Etapa 5: Negocios Digitales
- Etapa 6: Autonomía

Es importante resaltar el lugar que ocupan las tecnologías empleadas en esta investigación como son la Internet de las Cosas y el Machine Learning según Gartner.

Figura 20. Ciclo de bombo (despliegue publicitario) de Tecnologías Emergentes, 2015



Fuente: <http://www.gartner.com/newsroom/id/3114217>

Autonomía (Etapa 6): La autonomía representa la etapa final post-nexus. Esta etapa está definida por la habilidad de una empresa para aprovechar tecnologías que proporcionen capacidades parecidas a los humanos o que reemplazan a los humanos.

Algunos ejemplos que marcan la etapa de autonomía son: usar vehículos autónomos para transportar personas o productos, emplear sistemas cognitivos para recomendar una estructura potencial para una respuesta a un email, escribir textos o responder preguntas de los clientes. Las empresas que estén buscando alcanzar esta etapa para ganar competitividad deberían considerar estas tecnologías en el Ciclo del Bombo: Vehículos Autónomos, Sensado Bioacústico, Biochips, Interface Cerebro-Computador, Destreza Digital, Aumentación Humana, Aprendizaje Maquina, Neuronegocios, Tecnología para personas analfabetas, Computación Cuántica, Robots Inteligentes, Asistentes Personales Virtuales, Realidad Virtual, Asesores Inteligentes, Polvo Inteligente, Pantallas Volumétricas y Holográficas. ("Gartner's Hype Cycle for Emerging Technologies," 2015)

En cuanto a la autonomía del sistema de riego, se podrá lograr empleando algoritmos de Machine Learning (Aprendizaje Maquina). En 1959, Arthur Samuel lo definió como:

"Campo de estudio que da a los computadores la habilidad para aprender sin ser explícitamente programados".

Otra definición más completa propuesta por Tom Mitchell (1997) es:

"Un programa de computador se dice que aprende de una experiencia E con respecto a alguna tarea T y algún rendimiento medido P, si su rendimiento en T, medido por P, mejora con la experiencia E.

Tomando esta última definición vamos a hacer una pequeña aproximación a nuestro problema.

Aprender de la Experiencia:

Los sistemas de Machine Learning se describen con frecuencia que aprenden de la experiencia con o sin supervisión de los humanos. En los problemas de **aprendizaje supervisado** un programa predice una salida para una entrada al aprender de pares de entradas y salidas etiquetadas; es decir, el programa aprende de los ejemplos de las respuestas correctas. En el **aprendizaje no supervisado**, un programa no aprende de datos etiquetados. En cambio, intenta descubrir patrones en los datos.

Tareas de Machine Learning:

Dos de las tareas de Machine Learning supervisado más comunes son clasificación y regresión. En los problemas de regresión el programa debe predecir el valor de una variable de respuesta continua. Ejemplos de problemas de regresión incluyen predecir las ventas para un nuevo producto, o el salario para un trabajo basado en su descripción. Similar a la clasificación, los problemas de regresión requieren aprendizaje supervisado.

Las medidas de rendimiento, el sesgo (bias) y varianza

Para medir si un programa está aprendiendo o no a ejecutar su tarea más efectivamente se pueden utilizar muchas métricas. Para problemas de aprendizaje supervisado, muchas métricas de rendimiento miden el número de errores de predicción. Hay dos causas fundamentales de errores de predicción: un sesgo del modelo y su varianza.

Un modelo con un sesgo alto producirá errores similares para una entrada sin importar del conjunto de datos de entrenamiento con el que haya sido entrenado; el modelo sesga sus propias asunciones acerca de las relaciones reales sobre las relaciones demostradas en los datos de entrenamiento.

Asuma que tiene muchos conjuntos de datos de entrenamiento que son todos únicos, pero igualmente representativos de la población. Un modelo con una alta varianza, al contrario, producirá errores diferentes para una entrada dependiendo del conjunto de entrenamiento con el que fue entrenado, Un modelo con un sesgo alto es inflexible, pero un modelo con una varianza alta puede ser tan flexible que este modela el ruido en el conjunto de datos de entrenamiento. Es decir, un modelo con alta varianza over-fits (ajusta por encima) los datos de entrenamiento, mientras que un modelo con sesgo alto under-fits (ajunta por debajo) los datos de entrenamiento. (Hackeling, 2014)

Para nuestro caso se desea que el sistema haga la predicción de las variables meteorológicas para la programación del riego (tarea T), para ello ejecutamos algoritmos de Machine Learning con datos de años pasados de las estaciones (experiencia E) y, si este ha “aprendido” exitosamente, entonces hará mejores predicciones para los futuros valores de las variables meteorológicas (medida de rendimiento P).

Dentro de los lenguajes de programación para Machine Learning más utilizados están Java, Python, R, Matlab, Scala, Clojure, Ruby, etc. Existen diferentes librerías para Machine Learning tales como Weka, Mahout, Spark, Orange, etc.

Para esta investigación se utilizará scikit-learn, una de las librerías de Machine Learning para Python más populares, que ha liberado versiones desde 2007, proporciona algoritmos para las diferentes tareas incluyendo clasificación, regresión, reducción de la dimensionalidad y agrupamiento. También provee módulos para extraer características (features), procesar datos, y evaluar modelos. Scikit-learn es popular para la investigación académica debido a que tiene una API bien documentada, fácil de usar y versátil.

5.4 Hardware

Existen múltiples plataformas hardware y software para el desarrollo de la Internet de las Cosas, entre ellas Intel® Galileo, Intel® Edison, BeagleBone, Raspberry PI, Arduino, etc.

La Raspberry Pi 2 es una computadora del tamaño de una tarjeta de crédito tiene una memoria de 1GB, 4 puertos USB, 40 pines GPIO, Conector HDMI, Tarjeta de Red 10/100, Procesador Broadcom BCM2836 ARMv7 Quad Core a 900MHz que corre 6 veces más rápido que el modelo anterior, etc.

Figura 21. Raspberry Pi 2



Fuente: <https://www.raspberrypi.org/>

El sistema operativo oficial soportado por la Raspberry PI es el Raspbian, basado en Debian. También hay versiones de terceras partes como Microsoft Windows 10 IoT Core, UBUNTU, OPENELEC, etc.

Arduino es una plataforma de prototipado rápido open-source basada en hardware y software fácil de usar. Nació en el Instituto de Diseño de la Interacción Ivrea como

una herramienta dirigida a estudiantes sin conocimientos previos en electrónica y programación. Todas las tarjetas Arduino son completamente open-source (open-hardware) lo cual permite a los usuarios construirlas independientemente y de ser necesario adaptarlas a sus necesidades particulares. El software, también, es open-source, usa el lenguaje de programación Arduino (basado en Wiring), y el Software Arduino (IDE), basado en Processing. (“Arduino - Sitio Oficial,” 2015)

Figura 22. Arduino Uno



Fuente: <https://www.arduino.cc/en/Guide/Windows>

El Arduino Uno es una tarjeta microcontroladora basada en el ATmega328P. Tiene 14 pines de entrada/salida digital (6 de los cuales se pueden usar como salidas PWM), 6 entradas análogas, un cristal de 16 MHz, una conexión USB, un conector de alimentación, un encabezado ICSP y un boton de reseteo.

En esta investigación utilizaremos la Raspberry PI 2 y el Arduino Uno como plataformas de IoT para gestionar los actuadores (válvulas) y sensores (humedad en suelo, temperatura, etc.).

5.5 Comunicación

Los módulos de comunicación Xbee son una marca de radio que soporta una variedad de protocolos de comunicación, incluyendo ZigBee, 802.15.4, y WiFi, entre otros. ZigBee es un protocolo de comunicación estándar para redes mesh de baja potencia e inalámbricas.

La capa de red detrás de ZigBee que soporta sus características avanzadas se conoce como IEE 802.15.4. Es un conjunto de estándares que definen el manejo de la energía, direccionamiento, corrección de errores, formatos de mensaje, y otros específicos punto a punto necesarios para que tenga lugar una comunicación apropiada entre un radio y otro.(Faludi, 2011)

5.6 Servicios Web

En 2000, Roy Fielding, uno de los contribuidores clave al HTTP y URI, codificó la arquitectura de la Web en su tesis doctoral titulada “Estilos de Arquitectura y el Diseño de Arquitecturas de Red Basadas en Red”. En su tesis, introdujo un estilo de arquitectura conocido como Transferencia del Estado Representacional (REST). Este estilo, en términos abstractos, describe las bases de la World Wide Web. Las tecnologías que componen estas bases incluyen el Protocolo de Transferencia de Hipertexto (HTTP), el Identificador de Recursos Uniforme (URI), lenguajes de marca tales como HTML y XML, y formatos amigables tal como JSON.(Allamaraju, 2010)

En REST cualquier recurso se direcciona por un identificador único de formato estándar vía Localizadores de Recursos Uniformes (URL) utilizando el Protocolo de Transferencia de Hipertexto (HTTP) y sus métodos (por ejemplo GET, POST, PUT, DELETE) para accederlos. El estado de un recurso, por ejemplo un nodo sensor, se puede obtener o asignar por los métodos HTTP GET o POST, respectivamente. (Bouguettaya, Sheng, & Daniel, 2014)

Creando un Recurso con POST

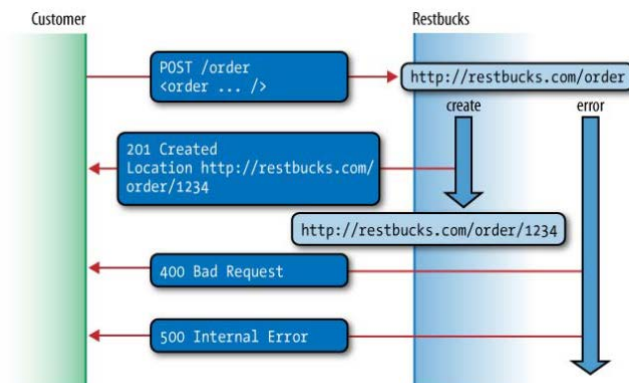


Figure 4-4. Creating an order via POST

Leyendo el Estado de un Recurso con GET

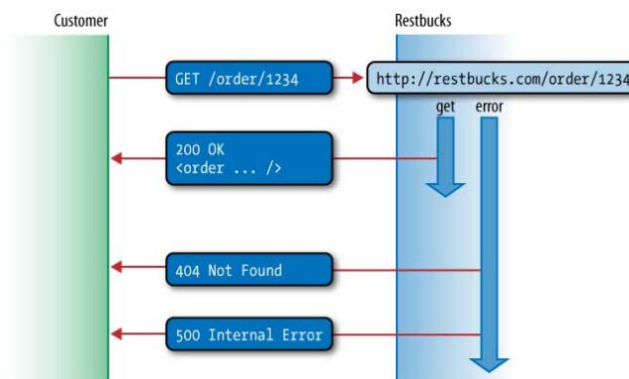


Figure 4-5. Reading a coffee order with GET

Actualizando un Recurso con PUT

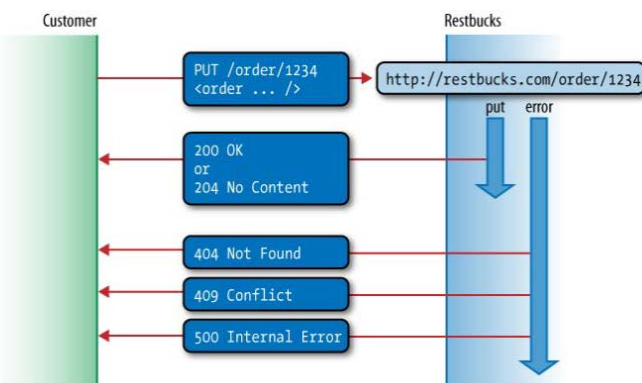


Figure 4-6. PUT request and responses

Eliminando un Recurso con DELETE

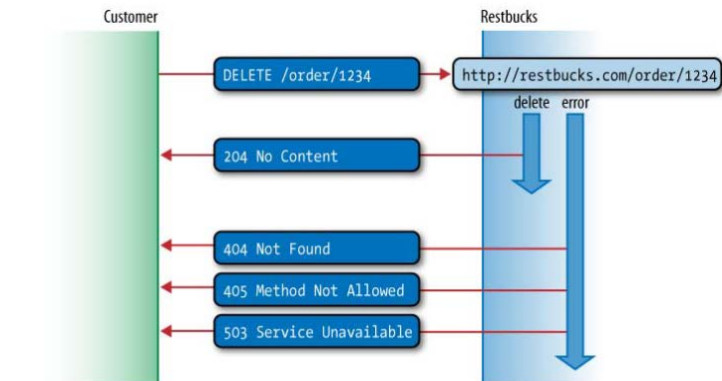


Figure 4-7. DELETE request and responses

Fuente: (Webber, Parastatidis, & Ian, 2010)

5.7 Seguridad

Para Webber et al. (2010) los siguientes cuatro pilares básicos de la computación segura son utilizados para la construcción de aplicaciones para la Web:

- Confidencialidad: La habilidad para mantener la información privada mientras se encuentra en tránsito o almacenada.
- Integridad: La capacidad para prevenir que la información sea cambiada sin detectarse.
- Identidad: La habilidad para autenticar las partes involucradas en una interacción.
- Confianza: Autorizar una parte para interactuar con un sistema de una forma prescrita (acordada).

El Servicio de Autorización y Autenticación de Java (JAAS) fue introducido como un paquete adicional (extensión) al Java 2 SDK, Edición Estándar (JSDK), v 1.3. JAAS fue integrado en el J2SDK 1.4. ("JAAS Reference Guide," 2015)

JAAS se puede utilizar para dos propósitos:

- Para autenticación de usuarios, para determinar de una manera confiable y segura quien está actualmente ejecutando código Java, sin importar que el código este corriendo como una aplicación, un applet, un bean o un Servlet;
- y
- Para autorización de usuarios para asegurar que ellos tengan los derechos de control de acceso (permisos) requeridos para hacer las acciones ejecutadas.

La autenticación JAAS se ejecuta de manera conectable (puggable). Esto permite a las aplicaciones permanecer independientes de tecnologías de autenticación subyacentes. Tecnologías nuevas o actualizadas de autenticación se pueden conectar bajo una aplicación sin requerir modificaciones a la propia aplicación.

El proyecto abierto de seguridad de aplicaciones web (OWASP) es una organización internacional y la Fundación OWASP soporta los esfuerzos alrededor del mundo. OWASP es una comunidad abierta dedicada a habilitar a las organizaciones para concebir, adquirir, operar y mantener aplicaciones que puedan ser confiables. Todas las herramientas, documentos, foros y capítulos de OWASP son libres y abiertos para todos aquellos interesados en mejorar la seguridad de las aplicaciones. (“OWASP,” 2015)

ESAPI (La API de Seguridad Empresarial de OWASP) es una librería open source y gratuita para el control de la seguridad de las aplicaciones Web que hace más fácil para el programador escribir aplicaciones de riesgo bajo. Se puede utilizar en distintos lenguajes de programación, tales como Java, .NET, ASP Clásico, PHP, ColdFusion & CFML, Python y JavaScript. (“OWASP Enterprise Security API,” 2015)

5.8 Computación en la nube

El Instituto Nacional de Estándares y Tecnología (NIST) define la computación en la nube como:

La computación en la nube es un modelo para permitir acceso conveniente a la red bajo demanda a un conjunto compartido de recursos informáticos configurables (por ejemplo, redes, servidores, almacenamiento, aplicaciones y servicios) que pueden ser rápidamente provisionados y liberados con un mínimo esfuerzo de gestión o interacción proveedor de servicios. (National Institute of Standards and Technology, 2009)

También se definen los tres modelos de servicio:

- **Software en la Nube como un Servicio (SaaS).** La capacidad proporcionada al consumidor para utilizar las aplicaciones del proveedor que se ejecutan sobre la infraestructura de la nube.
- **Plataforma en la Nube como un Servicio (PaaS).** La capacidad proporcionada al cliente para distribuir sobre la infraestructura de la nube aplicaciones creadas o adquiridas por el consumidor utilizando lenguajes de programación y herramientas soportadas por el proveedor.
- **Infraestructura en la Nube como un Servicio (IaaS).** La capacidad proporcionada al consumidor para aprovisionar procesamiento, almacenamiento, redes, y otros recursos computacionales fundamentales

donde el consumidor está en capacidad de distribuir y ejecutar software arbitrario, lo cual puede incluir sistemas operativos y aplicaciones.

Existe una gran cantidad de proveedores de Servicios de Computación en la Nube, tales como Amazon Web Services, Microsoft Azure, Oracle Cloud, Google Cloud Platform, Heroku, OpenShift, etc.

OpenShift Online es la plataforma pública en la nube de Red Hat (PaaS) para el desarrollo y hospedaje de aplicaciones que automatiza el aprovisionamiento, gestión, y escalado de aplicaciones de modo que usted se pueda enfocar en escribir código para su negocio, startup o gran idea.(OpenShift, 2015)

OpenShift Online soporta una variedad de tecnologías a través del uso de cartridges (cartuchos), incluyendo Java (Wildfly, JBossEAP, Tomcat), PHP, Node.js, Python, Perl, MySQL, PostgreSQL, MongoDB, Jenkins, Cron, and JBoss xPaaS Services (Fuse, BPM Suite, BRMS, Data Virtualization, Aerogear, and mas).

Para esta investigación se utilizará OpenShift por la facilidad de uso, integración con eclipse, uso de git, la posibilidad de usar java (wildfly), Python, Flask, scikit-learn, Postgres, facilidad para instalar otras librerías, uso de https, etc.

6. DISEÑO METODOLOGICO

Se va a construir un sistema de riego autónomo basado en la Internet de las Cosas de una forma iterativa e incremental, por ejemplo:

Tabla 3. Desarrollo por prototipos

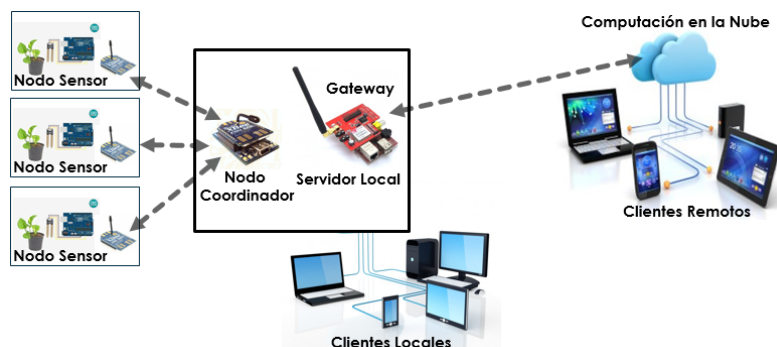
Prototipo	Énfasis
1	Protocolos y Computación en la Nube
2	Bases de Datos
3	Servicios Web
4	Seguridad
5	Autonomía - Inteligencia
6	Aplicación Web

En las tareas de Desarrollo y Pruebas de Prototipo se abarcan las fases de Modelado del Negocio, Análisis de Requerimientos, Análisis y Diseño, Implementación, Test – Pruebas y Distribución. Los prototipos comprenden los elementos de Hardware y Software.

6.1 Arquitectura del Sistema

El Sistema de Riego Autónomo basado en la Internet de las Cosas, en una primera fase, enfocada a la predicción de las necesidades de agua para los cultivos, comprende los nodos de sensores (principalmente humedad del suelo), el nodo coordinador o Gateway, y el nodo de computación en la nube.

Figura 23. Arquitectura del Sistema de Riego Autónomo basado en la IoT



El prototipo construido tiene tres nodos sensores, que envían la información obtenida a un nodo coordinador. El Servidor Local es una réplica del Servidor en la Nube y la funcionalidad del sistema está disponible a través de Servicios Web RESTful. El proveedor de los servicios de computación en la Nube es OpenShift, el cual nos permite utilizar WildFly como servidor de aplicaciones, Java como lenguaje de programación, RestEasy como framework para la creación de los Servicios Web, PostgreSQL como motor de bases de datos. Para la programación de las tareas se utilizara Quartz dentro del servidor de Aplicaciones (también se puede utilizar Cron). Todas estas opciones de software se pueden instalar en la Raspberry PI.

El lenguaje de programación escogido es Java con eclipse como IDE, el cual nos permite el desarrollo de la parte web (JSP, Servlets, JavaBeans, etc.), las aplicaciones móviles y el Java ME embebido en Raspberry PI.




En la construcción del sistema de Machine Learning para las predicciones se utilizará el lenguaje de programación Python y las librerías scikit-learn, pandas, numpy, etc. Se usará Flask y Flask-Restful para la creación de los Servicios Web y SQLAlchemy para la conexión a la base de datos.


El sistema tendrá una base de datos compartida por las dos aplicaciones (Java y Python) implementada en Postgres.


6.2 Componentes de Hardware


Para este sistema vamos a utilizar los siguientes componentes de hardware, el criterio principal de selección es la integración con Arduino – Raspberry PI, el precio y la facilidad de adquirirlos en el mercado local.


Si bien existen componentes de mayor calidad, el costo no los hace viables, al menos en esta primera fase de construcción del prototipo.





Sensor de Humedad en Suelo	
	<p>Este sensor nos permitirá conocer el nivel de humedad del suelo.</p>
Sensor de Temperatura en Suelo	
	<p>Este sensor nos permitirá conocer la temperatura del suelo.</p>
Modulo Detección de gotas de lluvia y humedad [C]	
	<p>Estos módulos nos permiten detectar las gotas de lluvia y la humedad.</p>





Sensor de Distancia por Ultrasonico	
	<p>Este sensor nos permitirá medir los niveles de agua.</p>

Sensor de Temperatura Ambiente	
	<p>Estos sensores nos permitirán conocer la temperatura y la humedad.</p>

Válvula Solenoide – Electroválvula	
	<p>Esta válvula solenoide nos permite controlar el flujo de líquido.</p>

Bomba de líquido	
	<p>Esta bomba para fluido nos permite mover el líquido para los sistemas de riego o manejo de los reservorios de agua, es completamente sumergible.</p>

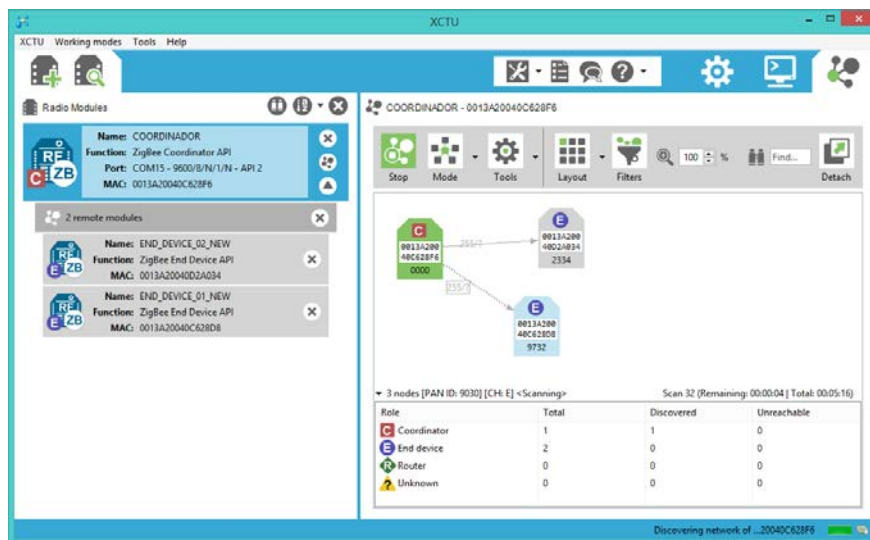
Sensor de Flujo de Liquido	
	<p>Este sensor nos permitirá conocer la cantidad de líquido que pasa por la válvula en un tiempo determinado.</p>
Arduino	
	<p>Arduino es una plataforma open-source para programación basada en una tarjeta con entradas/salidas y un ambiente de desarrollo. El IDE open-source puede descargarse gratuitamente (para Mac, Windows y Linux). Tiene 14 entradas/salidas digitales (incluye 6 salidas PWM), 6 entradas análogas, 32K de Memoria Flash, Alimentación: 7V hasta 12V.</p>
Raspberry – Pi	
	<p>Raspberry Pi es un computador del tamaño de una tarjeta de crédito. Características: Adaptador de red, USB; Salida de video compuesto y HDMI; Salida de audio conector 3.5mm; Puerto Ethernet; Puertos GPIO; Alimentación: 5V mediante conector micro USB; Almacenamiento mediante tarjeta SD;</p>
Xbee - Módulo de comunicación ZigBee serie 2B de 2.4GHz	
	<p>Este módulo de comunicación ZigBee-Pro de 2.4Ghz, 250 Kbps, 256000 BAUD, cumple con el estándar IEEE 802.15.4 con un alcance de 90m en interiores (edificios de oficinas) y 1600m en exteriores (con línea de vista).</p>

HC-06 Módulo de comunicación Bluetooth	
	<p>Este módulo de comunicación Bluetooth se conecta fácilmente al Arduino y permite la conexión con Android.</p>
Módulo de comunicación WIFI	
	<p>Este módulo de comunicación WIFI conectado a la Raspberry PI nos permite la comunicación Inalámbrica a la Red.</p>
Módulo RASPBERRY PI SIM900 GSM/GPRS	
	<p>Este módulo nos permite comunicarnos a través de la red celular, se conecta al Raspberry PI. Permite enviar y recibir mensajes SMS, hacer y recibir llamadas y conectarse a Internet.</p>
Kit de Energía Solar	
	<p>Estos kits incluyen un panel solar, un cargador y una batería. Serán utilizados para alimentar los nodos inalámbricos.</p>

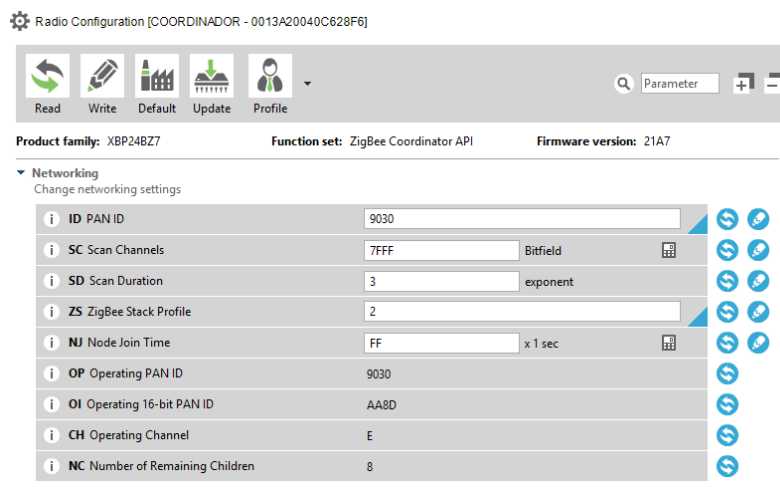
6.3 Comunicación

Para la comunicación inalámbrica entre los nodos se utilizarán los módulos de radio Xbee. Para nuestro caso se implementará una red con un nodo coordinador y dos nodos sensores. Un nodo sensor de las condiciones del suelo, principalmente la

humedad y temperatura. Y un nodo sensor de las condiciones meteorológicas, temperatura y humedad ambiente, precipitación, etc.



El software XCTU permite configurar los módulos de radio para crear la red.



La base de datos del sistema permite el registro de los nodos sensores (tabla dispositivo), el campo dirección nos permite identificar los nodos. Los diferentes sensores y elementos que corresponden a un dispositivo (sensor), se registran en la tabla dispositivo_item.

Figura 24. Tablas para dispositivos



Ejemplo de configuración en la base de datos del Nodo Sensor 01, el cual contiene tres (3) sensores de humedad en suelo, a diferente profundidad (30, 60 y 90 cm) y un sensor de temperatura.

Figura 25. Ejemplo de configuración de un nodo sensor

	dis_nombre character varying(100)	dii_codigo bigint	dii_nombre character varying(100)	dii_tipo_medicion integer	tim_nombre character varying(100)	dii_acronimo character varying(100)
1	Nodo Sensor 01	37	Humedad en Suelo 01 - 30 cm	27	Humedad en Suelo	humedad1
2	Nodo Sensor 01	43	Humedad en Suelo 02 - 60 cm	27	Humedad en Suelo	humedad2
3	Nodo Sensor 01	44	Humedad en Suelo 03 - 90 cm	27	Humedad en Suelo	humedad3
4	Nodo Sensor 01	45	Temperatura en Suelo 01	28	Temperatura en Suelo	temperatura

Los módulos Xbee se configuraron con el parámetro AP = 2 lo cual habilita el modo API, y permite la programación de los módulos de radio, tanto en Arduino como en Java. Desde Arduino con el uso de la librería arduino-xxee se envía una cadena de texto en formato json con los datos de los sensores y sus mediciones, la cual se recibe en una aplicación en java que se encarga de enviar los datos recibidos y la dirección del nodo emisor a un servicio web que adiciona la información en la base de datos.

Ejemplo de Json enviado por Arduino:

```
{'temperatura':31.23, 'humedad1':825, 'humedad2':600, 'humedad3':200}
```

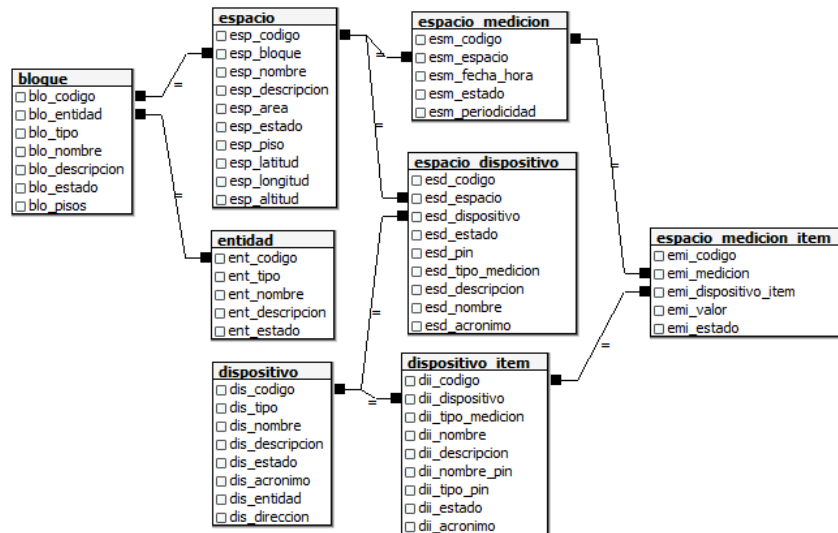
Al consultar la base de datos a partir de la dirección del dispositivo se puede conocer qué información se debe recuperar del objeto Json y a que sensor (dispositivo_item) corresponde.

Figura 26. Consulta de los elementos de un dispositivo

	dii_codigo bigint	dii_acronimo character varying(100)
1	37	humedad1
2	43	humedad2
3	44	humedad3
4	45	temperatura

Las tablas dispositivo y espacio_dispositivo permiten conocer el espacio (lugar) de donde proviene la información, a partir de la dirección del dispositivo. La información de las mediciones (sensado) se almacena en las tablas espacio_medicion y espacio_medicion_item.

Figura 27. Tablas para almacenar la información de las mediciones



Ejemplo de consulta de la información de las mediciones, datos de una medición de la Estación Meteorológica de Prueba y del Nodo Sensor 01.

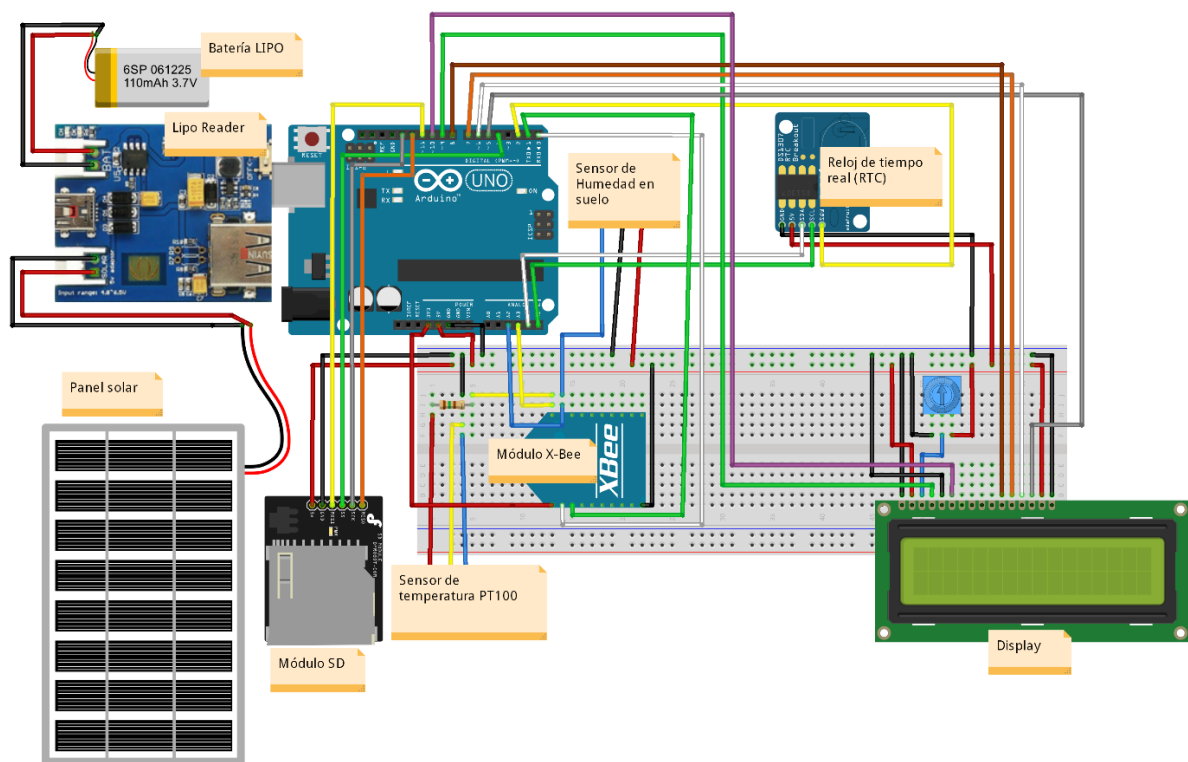
	esp_nombre character varying(100)	dis_nombre character varying(100)	esm_codigo bigint	esm_fecha_hora timestamp without time zone	dii_nombre character varying(100)	emi_valor character varying(100)
1	Lote para cultivar 01	Estación Meteorológica de Prueba	67	2016-01-04 02:19:49.092	Temperatura (°C)	31.67
2	Lote para cultivar 01	Estación Meteorológica de Prueba	67	2016-01-04 02:19:49.092	Precipitación	480.93
3	Lote para cultivar 01	Estación Meteorológica de Prueba	67	2016-01-04 02:19:49.092	Humedad (%)	45.78
4	Lote para cultivar 01	Nodo Sensor 01	68	2016-01-04 02:19:49.314	Humedad en Suelo 01 - 30 cm	825.00
5	Lote para cultivar 01	Nodo Sensor 01	68	2016-01-04 02:19:49.314	Humedad en Suelo 02 - 60 cm	500.00
6	Lote para cultivar 01	Nodo Sensor 01	68	2016-01-04 02:19:49.314	Humedad en Suelo 03 - 90 cm	200.00
7	Lote para cultivar 01	Nodo Sensor 01	68	2016-01-04 02:19:49.314	Temperatura en Suelo 01	30.00

La comunicación del nodo coordinador – Gateway con el proveedor de servicios de computación en la nube se hará a través de WiFi o a través de un módulo GPRS.

6.4 Nodo Sensor

El Arduino es el encargado de controlar el Nodo Sensor y tiene como función capturar los valores de las lecturas de los sensores, obtener la fecha y hora de la medición del reloj de tiempo real (RTC) y registrar la información en el micro SD para tener un respaldo. También debe mostrar al usuario final los valores convertidos a grados centígrados (temperatura) y porcentaje (humedad) en el Display. El módulo Xbee es el encargado de enviar la información al nodo coordinador, según el intervalo de tiempo configurado para ello. El panel solar y la batería LIPO alimentan en el gestor de energía LIPO Reader (cargador) el cual mantiene el nodo en funcionamiento.

Figura 28. Nodo Sensor con respaldo de la información



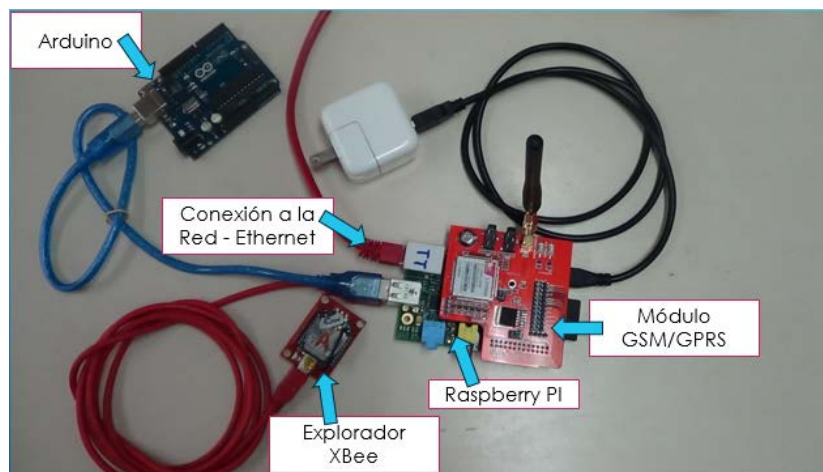
fritzing

6.5 Nodo Coordinador – Gateway – Servidor Local

El Servidor Local está instalado en la Raspberry Pi que tiene acceso a la red a través del adaptador Ethernet (modo cableado) o por medio de un módulo WIFI conectado a un puerto

USB. El Nodo Coordinador es gestionado por el módulo Xbee Explorer que a su vez se conecta a la Raspberry PI a través del puerto USB. La información que envían los nodos sensores es capturada por una aplicación hecha en java con la librería XBee-API y a través de JDBC permite su almacenamiento en una base de datos local en Postgres. El módulo GSM/GPRS permite el envío y recepción de mensajes SMS, hacer y recibir llamadas, conectar el sistema a internet (modem) y hacer uso de los protocolos HTTP y FTP, y así poder enviar a información al proveedor de servicios de computación en la nube (OpenShift).

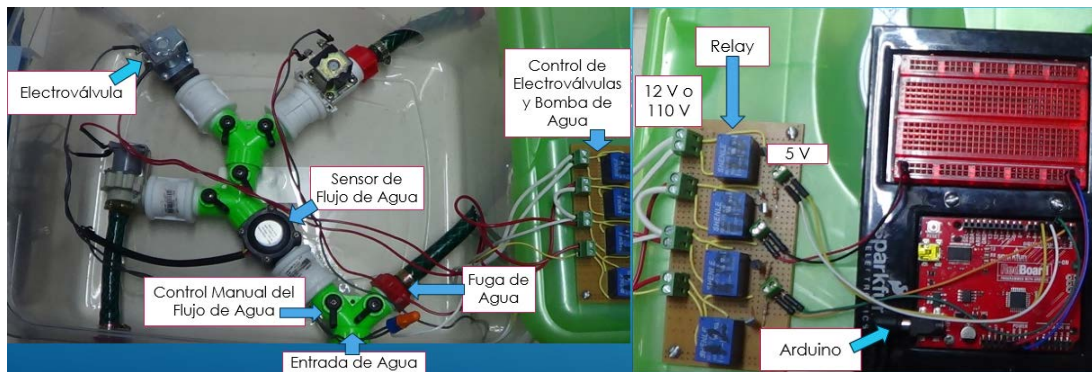
Figura 29. Nodo Coordinador – Gateway – Servidor Local



6.6 Nodo de Control de Riego

En este nodo encontramos las electroválvulas (actuadores) que controlan el flujo de líquido, también se puede hacer un control manual a través de las llaves de paso. Los actuadores son controlados por un Arduino a través de los Relays, que permiten o no el paso de la corriente (encendido – apagado). Un módulo de BlueTooth conectado al Arduino permite al usuario local con una aplicación Android controlar el Sistema de Riego. El control del Arduino desde El Servidor Local (Raspberry) se puede hacer a través de comunicación Serial o I2C principalmente.

Figura 30. Componentes del Nodo de Control de Riego



6.7 Servicios Web

Este sistema está implementado en Java utilizando para ello Servicios Web RESTful, con el apoyo de RestEasy de jboss, el cual nos permite crear clientes en Java y en AJAX y el uso de XML y JSON. Estos servicios están disponibles tanto para el Servidor Local (Raspberry Pi) y el Servidor Remoto (OpenShift).

A continuación se presentan los principales servicios web que pertenecen al sistema de riego.

6.7.1 Servicios Infraestructura

Tabla 4. Catálogo de Servicios Web - Entidad

Servicios	Descripción
../servicios/entidad/adicionar	Adiciona un registro. (create – POST)
../servicios/entidad/listar	Retorna el listado de registros. (read – GET)
../servicios/entidad/modificar	Modifica un registro. (update – PUT)
../servicios/entidad/eliminar	Elimina un registro. (destroy – DELETE)

Tabla 5. Catálogo de Servicios Web – Bloque

Servicios	Descripción
../servicios/bloque/adicionar	Adiciona un registro. (create – POST)
../servicios/bloque/listar	Retorna el listado de registros. (read – GET)
../servicios/bloque/modificar	Modifica un registro. (update – PUT)
../servicios/bloque/eliminar	Elimina un registro. (destroy – DELETE)

Tabla 6. Catálogo de Servicios Web – Espacio

Servicios	Descripción
../servicios/espacio/adicionar	Adiciona un registro. (create – POST)
../servicios/espacio/listar	Retorna el listado de registros. (read – GET)
../servicios/espacio/modificar	Modifica un registro. (update – PUT)
../servicios/espacio/eliminar	Elimina un registro. (destroy – DELETE)

6.7.2 Servicios Hardware

Tabla 7. Catálogo de Servicios Web – Dispositivo

Servicios	Descripción
../servicios/dispositivo/adicionar	Adiciona un registro. (create – POST)
../servicios/dispositivo/listar	Retorna el listado de registros. (read – GET)
../servicios/dispositivo/modificar	Modifica un registro. (update – PUT)
../servicios/dispositivo/eliminar	Elimina un registro. (destroy – DELETE)

Tabla 8. Catálogo de Servicios Web – Ítems por Dispositivo

Servicios	Descripción
../servicios/dispositivo_item/adicionar	Adiciona un registro. (create – POST)
../servicios/dispositivo_item/listar	Retorna el listado de registros. (read – GET)
../servicios/dispositivo_item/modificar	Modifica un registro. (update – PUT)
../servicios/dispositivo_item/eliminar	Elimina un registro. (destroy – DELETE)

6.7.3 Servicios Riego

Tabla 9. Catálogo de Servicios Web – Espacio para Cultivar

Servicios	Descripción
../servicios/espacio_cultivo/adicionar	Adiciona un registro. (create – POST)
../servicios/espacio_cultivo /listar	Retorna el listado de registros. (read – GET)
../servicios/espacio_cultivo /modificar	Modifica un registro. (update – PUT)
../servicios/espacio_cultivo /eliminar	Elimina un registro. (destroy – DELETE)

Tabla 10. Catálogo de Servicios Web – Calendario de Riego

Servicios	Descripción
../servicios/riego_calendario/adicionar	Adiciona un registro. (create – POST)
../servicios/riego_calendario/listar	Retorna el listado de registros. (read – GET)
../servicios/riego_calendario/modificar	Modifica un registro. (update – PUT)
../servicios/riego_calendario/eliminar	Elimina un registro. (destroy – DELETE)

Tabla 11. Catálogo de Servicios Web – Programación del Riego

Servicios	Descripción
../servicios/riego_programacion/adicionar	Adiciona un registro. (create – POST)
../servicios/riego_programacion/listar	Retorna el listado de registros. (read – GET)
../servicios/riego_programacion/modificar	Modifica un registro. (update – PUT)
../servicios/riego_programacion/eliminar	Elimina un registro. (destroy – DELETE)

Tabla 12. Catálogo de Servicios Web – Eventos del Riego

Servicios	Descripción
../servicios/riego_evento/adicionar	Adiciona un registro. (create – POST)
../servicios/riego_evento/listar	Retorna el listado de registros. (read – GET)
../servicios/riego_evento/modificar	Modifica un registro. (update – PUT)
../servicios/riego_evento/eliminar	Elimina un registro. (destroy – DELETE)

6.8 Clientes

La función principal del sistema es ejecutar de forma automática las tareas de riego programadas. Sin embargo también se debe permitir un control manual, el cual se puede hacer desde la Web.

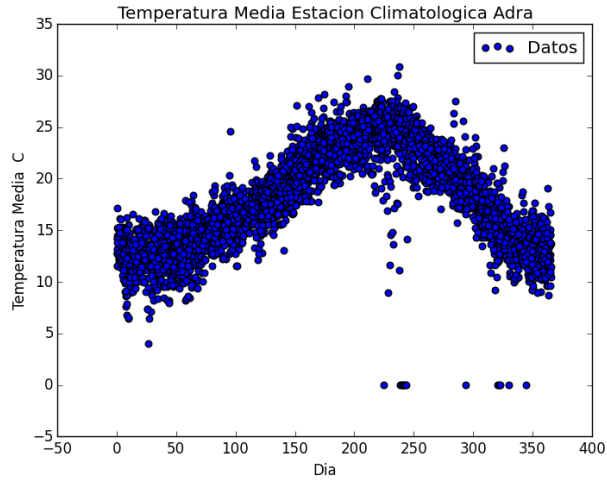
6.9 Machine Learning

6.9.1 Plot de los datos

Como primera fase en el desarrollo del sistema de predicciones se analiza el comportamiento de las variables meteorológicas. A continuación se muestra un ejemplo de los gráficos generados para la estación meteorológica Adra.

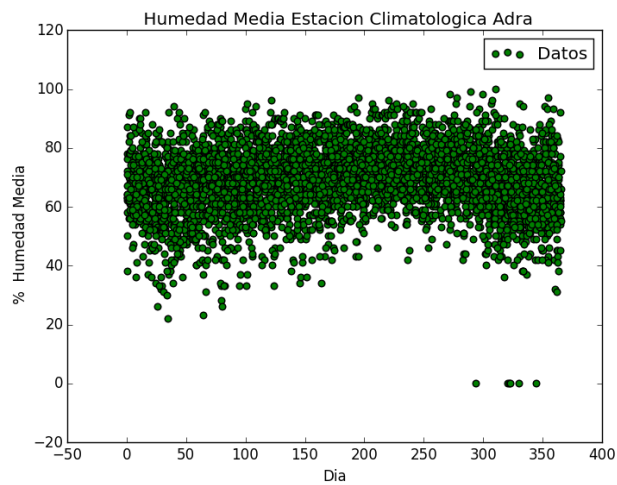
6.9.1.1 Temperatura

Figura 31. Plot de los datos de Temperatura Ambiente



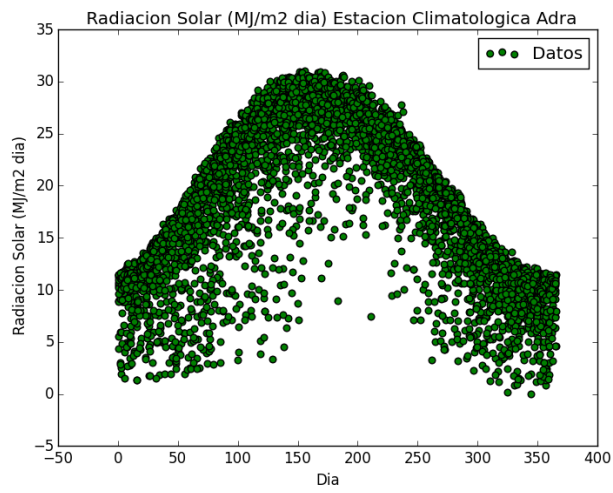
6.9.1.2 Humedad

Figura 32. Plot de los datos de Humedad Ambiente



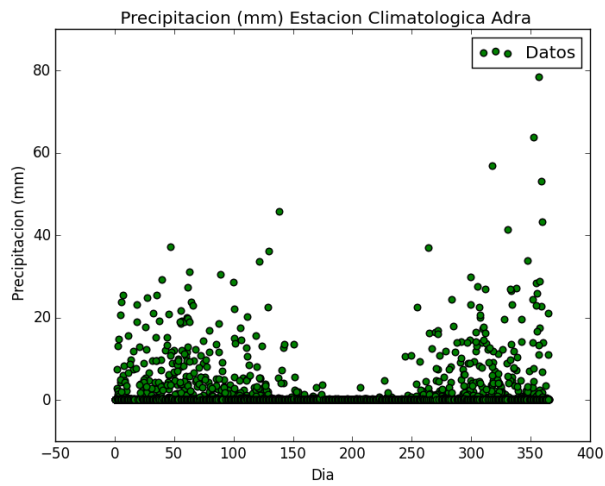
6.9.1.3 Radiación Solar

Figura 33. Plot de los datos de Radiación Solar



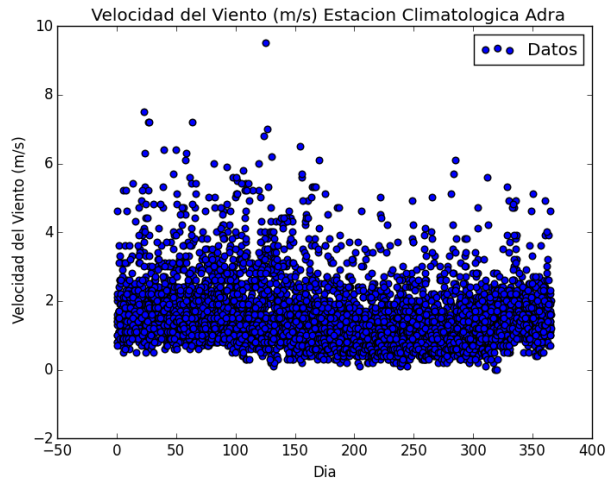
6.9.1.4 Precipitación

Figura 34. Plot de los datos de Precipitación



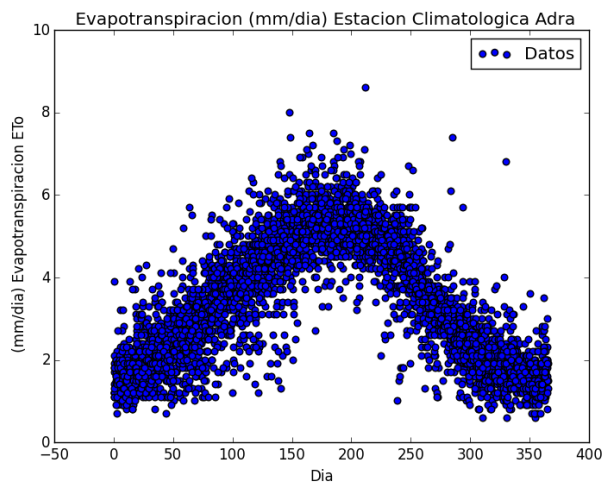
6.9.1.5 Velocidad del Viento

Figura 35. Plot de los datos de Velocidad del Viento



6.9.1.6 Evapotranspiración

Figura 36. Plot de los datos de Evapotranspiración

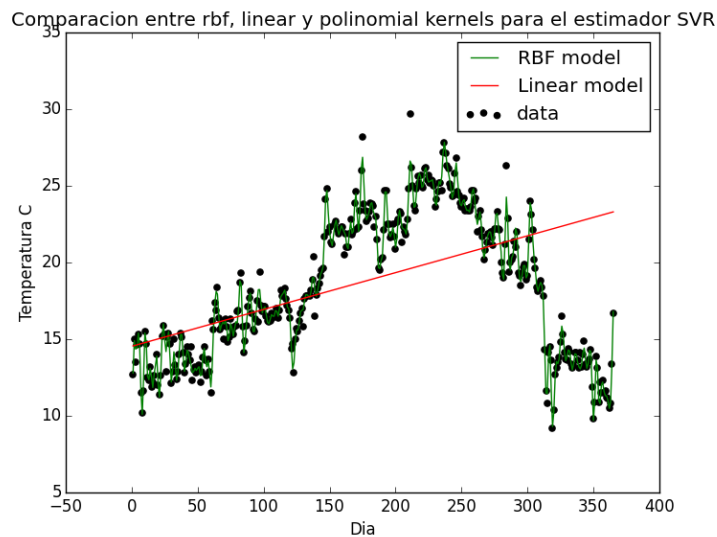


6.9.2 Selección del algoritmo de clasificación

Para seleccionar el algoritmo de regresión se hace una comparación de los resultados obtenidos, esto lo vamos a mostrar a través de gráficas.

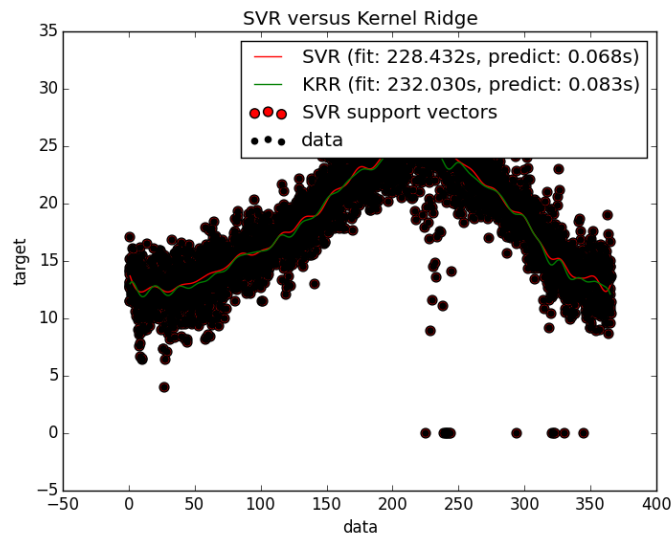
Observando la gráfica podemos ver que la regresión con el modelo lineal no es el más apropiado y precedimos a descartarlo.

Figura 37. Comparación de los Modelos Lineal y RBF



Seguido comparamos dos algoritmos de regresión como son el SVR y el KRR, aquí vemos que los dos se ajustan bien al comportamiento de los datos. Hemos escogido el SVR porque se requiere menos tiempo para llevar a cabo las predicciones.

Figura 38. Comparación de los modelos SVR y KRR



7. RESULTADOS OBTENIDOS

7.1 Hardware

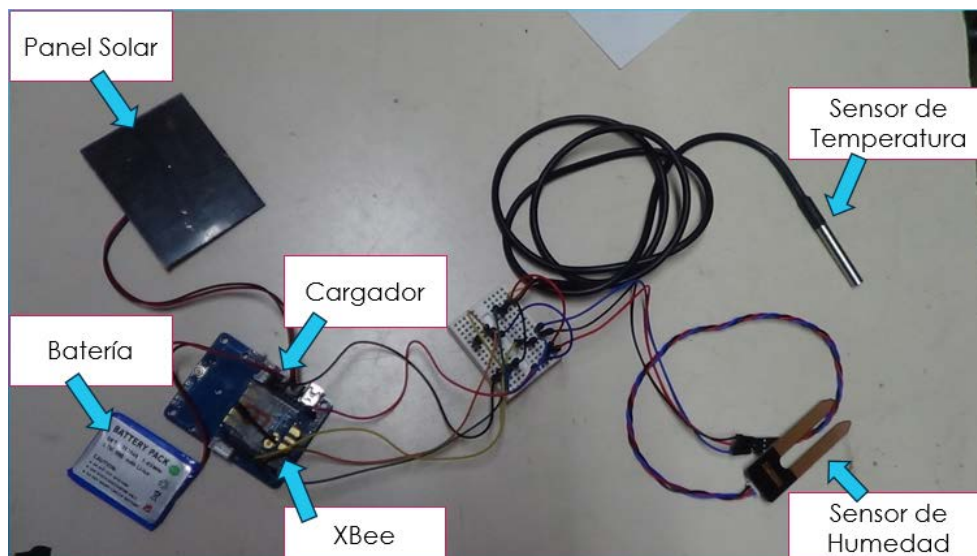
Para este sistema de IoT se construyeron los siguientes prototipos:

7.1.1 Nodo Sensor Simple (Sin respaldo de la información)

Este tipo de nodo sensor no tiene almacenamiento local de la información recolectada, se compone de tres partes principales:

- **Sensores:** Contiene un sensor de temperatura del ambiente y del suelo, y un sensor de humedad en suelo.
- **Transmisión:** Incluye un módulo Xbee PRO Serie 2 al cual se le pueden conectar directamente los sensores.
- **Alimentación Eléctrica:** Comprende un panel solar, una batería y un gestor de carga, lo cual permite una gran autonomía del sistema y el aprovechamiento de energías verdes.
- **Programación:** Este nodo no se programa pues no tiene un Arduino, simplemente se configura el Xbee para enviar los datos en un intervalo de tiempo, por ejemplo cada 15 minutos, a un nodo coordinador.

Figura 39. Nodo Sensor Simple

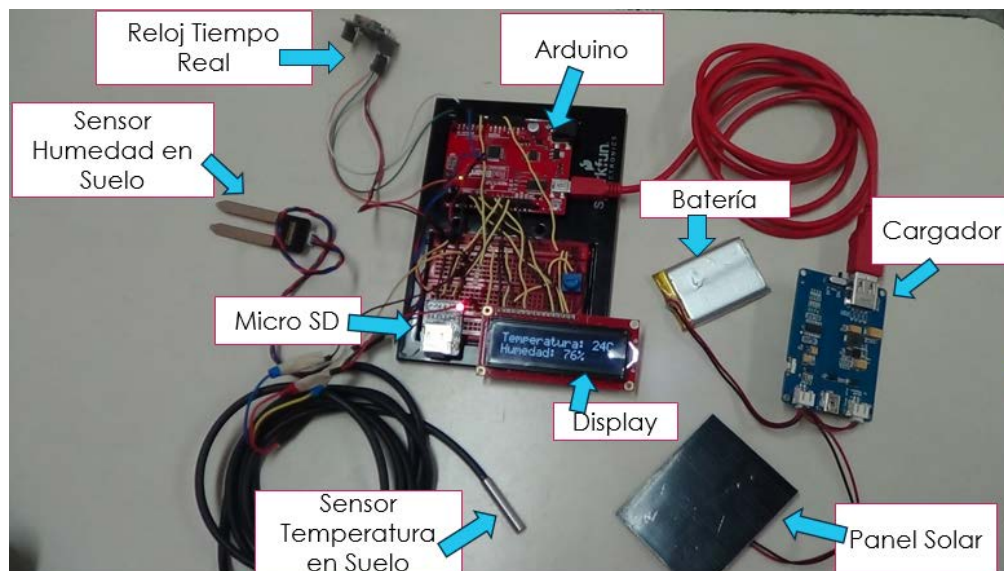


7.1.2 Nodo Sensor Complejo (Con respaldo de la información)

Este es el nodo sensor recomendado para proyectos en donde la información de los sensores sea de vital importancia y el factor económico lo permita. Se compone de las siguientes partes:

- **Sensores:** Incluye dos sensores, uno de temperatura y otro de humedad en suelo.
- **Control:** Contiene un Arduino, un reloj de tiempo real (RTC), un módulo de lectura y escritura en Micro SD y un Display. Permite almacenar los valores recolectados por los sensores, estampando la fecha y hora de la lectura y la identificación de nodo sensor. Además permite mostrar al usuario los valores de las lecturas del sensor en el Display. Se puede programar el Arduino para enviar los datos al nodo coordinador mediante la librería del Xbee.
- **Transmisión:** Incluye un módulo XBee PRO Serie 2 al cual se le pueden conectar directamente los sensores o hacer la transmisión con el apoyo del Arduino.
- **Alimentación Eléctrica:** Comprende un panel solar, una batería y un gestor de carga, lo cual permite una gran autonomía del sistema y el aprovechamiento to de energías verdes.

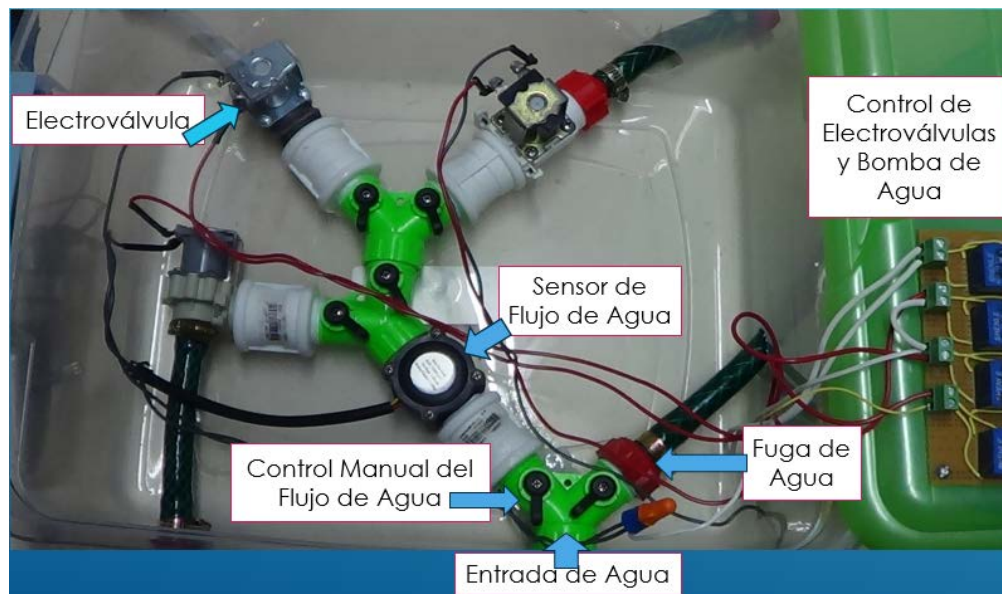
Figura 40. Nodo Sensor Complejo



7.1.3 Sistema de Control de Riego

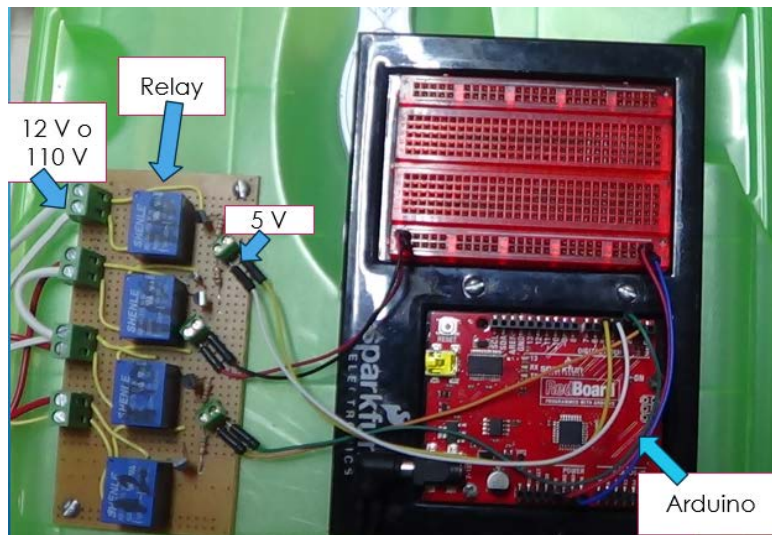
El sistema de control de riego regula el flujo de agua a través de las electroválvulas, las cuales permiten o no el paso de líquido. También poseen un mecanismo de apertura y cierre manual. El sistema también incluye un sensor de flujo de agua, el cual nos indica la cantidad de líquido empleada en un momento determinado.

Figura 41. Sistema de Control de Riego



El control de las electroválvulas se hace a través del Arduino y un Relay, el cual hace de switch permitiendo o no el paso de energía (prender / apagar).

Figura 42. Control de las Electroválvulas

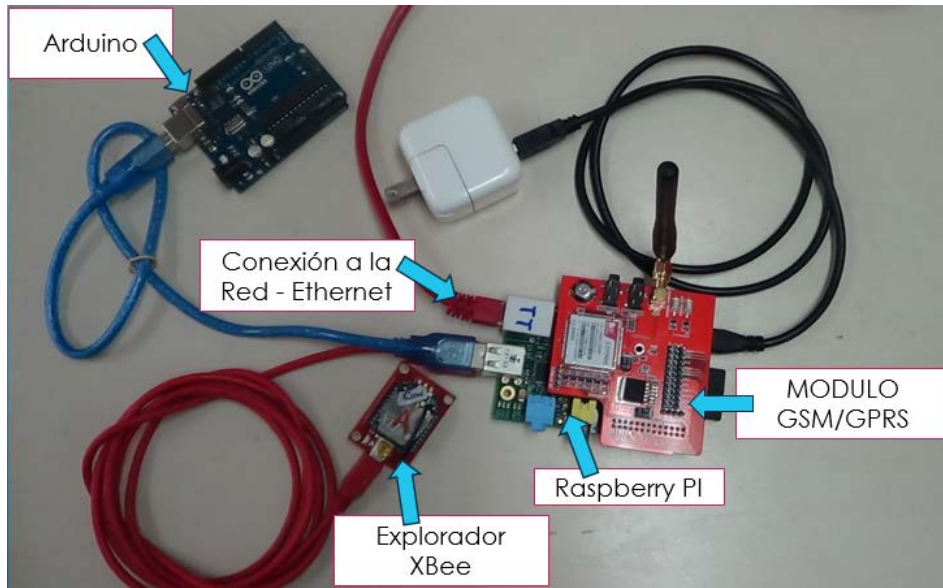


7.1.4 Nodo Coordinador, Gateway, Servidor Local

El nodo coordinador contiene los siguientes elementos:

- Comunicación: El explorador de Xbee se conecta al Raspberry PI y mediante una aplicación en java (Anexo X) con el uso de la librería xbee-api recolecta los datos enviados por los nodos sensores.
- Arduino: Para hacer el control del sistema de riego.
- Raspberry Pi: Hace de servidor local y permite hacer un respaldo de la información enviada por los sensores.
- Módulo GSM/GPRS: Permite al Arduino o al Raspberry PI enviar la información recolectada por los sensores al proveedor de servicios de Computación en la Nube (OpenShift).

Figura 43. Nodo Coordinador, Gateway, Servidor Local



7.2 Machine Learning

7.2.1 Modelos de predicción

7.2.1.1 Modelo de predicción de las variables meteorológicas

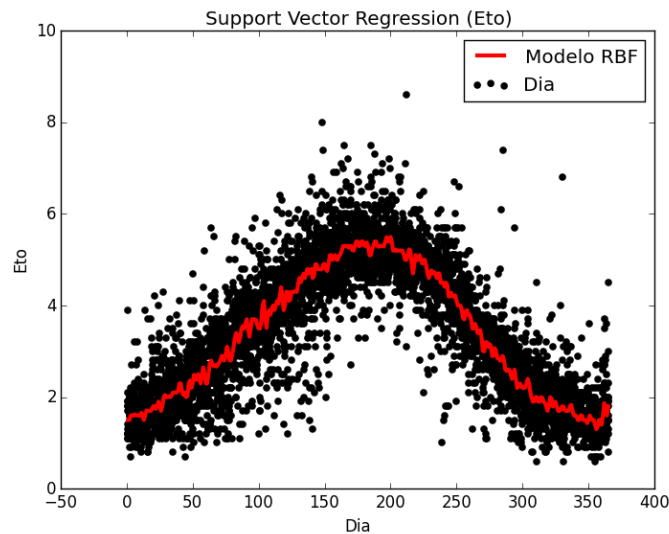
Para generar los modelos de predicción de las variables meteorológicas se tomaron datos de las estaciones meteorológicas de España, provenientes de los sitios web de la Agencia Estatal de Meteorología (AEmet) y la Junta de Andalucía.

Los datos de la AEmet corresponden a 30 años, con una periodicidad mensual y para cerca de 87 estaciones. Y los datos de la Junta de Andalucía comprenden 15 años, desde el año 2000 a la fecha, con una periodicidad diaria, para más de 30 estaciones.

Para migrar estos datos se creó una aplicación en java encargada leer la información proveniente de páginas html (AEmet) y de archivos planos (Junta de Andalucía) y grabarla en la base de datos del sistema.

El sistema tiene la opción para generar y grabar los modelos de predicción para cada estación y variable meteorológica, automáticamente estos se graban en una carpeta que se crea por estación. De esta forma los modelos generados pueden ser utilizados para las predicciones.

Ejemplo grafico del modelo generado para predecir la Evapotranspiración (ETo) para la estación Adra.

Figura 44. Modelo predicción de ETo (ejemplo estación Adra)

Sin embargo, este modelo no toma en cuenta las tendencias de los datos a través del tiempo, ni la periodicidad, lo cual nos obliga a recurrir a un modelo para series de tiempo.

Es importante entender que los datos meteorológicos forman series de tiempo, la cual se define como:

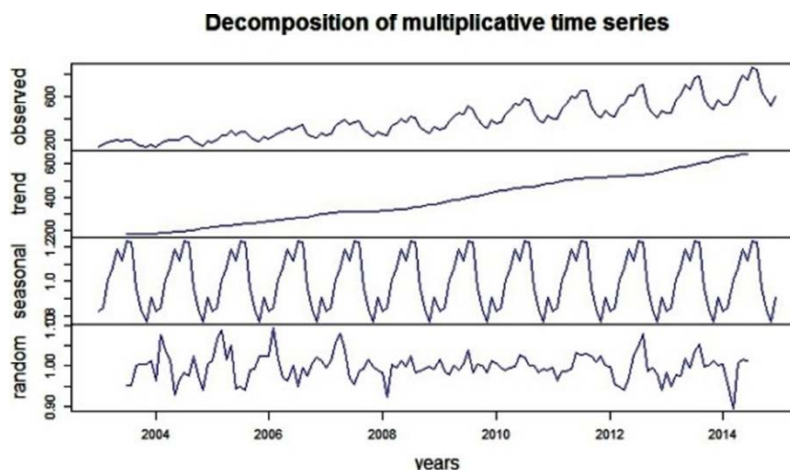
“El objetivo principal del análisis de series temporales es desarrollar modelos matemáticos que proporcionan descripciones plausibles para los datos de la muestra. A fin de proporcionar un ajuste estadístico para describir el carácter de datos que aparentemente fluctúan de forma aleatoria en el tiempo, suponemos una serie de tiempo puede definirse como una colección de variables aleatorias indexadas de acuerdo con el orden en que se obtienen en el tiempo. Por ejemplo, podemos considerar una serie de tiempo como una secuencia de variables aleatorias, x_1, x_2, x_3, \dots , Donde la variable aleatoria x_1 denota el valor tomado por la serie en el primer punto de tiempo, la variable x_2 denota el valor para el segundo período de tiempo, x_3 denota el valor para el tercer período de tiempo, y así sucesivamente.” (Shumway & Stoffer, 2011)

De igual manera las predicciones deben ser generadas a partir de un modelo propio de las series de tiempo como Arima, arma, Holt-winters, etc.

La idea es descomponer la serie de tiempo en varios componentes independientes para eso utilizamos Time Series Decomposition de la cual resulta lo siguiente:

- Tren = Dirección de la serie de tiempo.
- Seasonality = patrones mensuales, anuales, etc.
- Cycle = ciclos de negocio a largo plazo.
- Irregular remainder = ruido aleatorio que resulta después de la extracción de todos los componentes.

“La razón por la cual descomponemos la serie es porque es más fácil predecir valores regulares individuales producidos a través de la descomposición de series de tiempo que la misma serie de tiempo” (“Forecasting & Time Series Analysis,” 2016)



Después de seleccionar un modelo el siguiente paso es su especificación, la especificación de un modelo de predicción involucra seleccionar las variables a ser incluidas, seleccionando la forma de la ecuación de relación y estimando los valores de los parámetros en la ecuación.

Una vez que el modelo es especificado, sus características de rendimiento deberían ser verificadas y validadas comparando sus predicciones con datos históricos para los cuales fue entrenado.

Medidas de error tales como MAPE, RAE, MSE son usadas a menudo para evaluar la calidad de las predicciones, su elección es importante para definir cuál es mejor, La predicción de series de tiempo asume que una serie de tiempo es una combinación de patrones y algún error aleatorio.

La meta es separar los patrones del error entendiendo la tendencia, su crecimiento y decrecimiento a largo plazo y su estacionalidad, el cambio causado por factores de estacionalidad. Existen varios métodos para hacer predicción de series de tiempo como por ejemplo (Moving Averages method, Linear Regression with Time, Exponential Smoothing, etc.) (“Introduction to time series analysis.,” 2016)

Las predicciones para este sistema se realizaran con la técnica Holt-Winters Exponential Smoothing para ser aplicadas a series de tiempo que contiene estacionalidad como lo son los valores de variables meteorológicas (temperatura, precipitaciones, etc.).

Exponential smoothing

Es un procedimiento para revisar continuamente una predicción a la vista de experiencias más recientes. Este asigna medias decreciendo exponencialmente como la observación se torna más antigua. En otras palabras las observaciones reciente toman más relativamente más peso en las predicciones que las observaciones antiguas. (“Introduction to time series analysis.,” 2016)

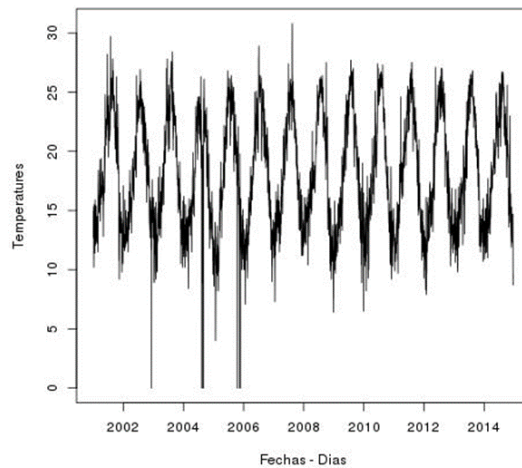
Modelo Holt-Winter

Holt (1957) y Winters (1960) extiende del método Holt para capturar la estacionalidad. El método estacional Holt-Winters comprende la ecuación de predicciones y 3 (smoothing equations) — uno para el nivel \hat{t} , uno para la tendencia b_t , y otro para el componente estacional denotado por s_t , con parámetro de (smoothing) α , β^* and γ . Usamos m para denotar el periodo de la estacionalidad, ejemplo, el número de estaciones en el año.

El modelo de predicción utiliza lo mejor de los lenguajes (R y Python) para hacer las predicciones a través del modelo Holt-winters.

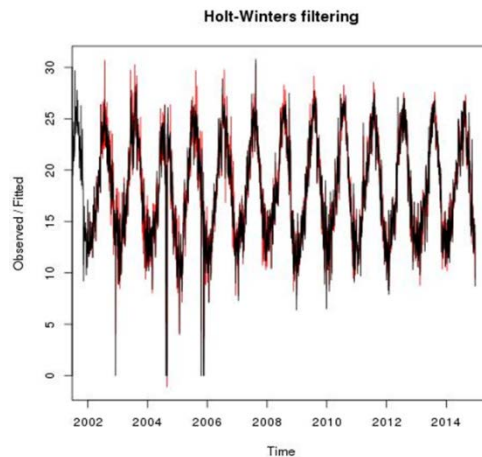
Aquí se presenta la visualización de la variable meteorológica (Temperatura Media), Se tomaron datos del 2001 al 2014 para la generación del modelo. La variable meteorológica analizada, descompuesta y sometida al modelo Holt-Winters fue la temperatura media, que presenta estacionalidad anual:

Figura 45. Visualización de datos de temperaturas diarias estación meteorología adra año 2001-2014



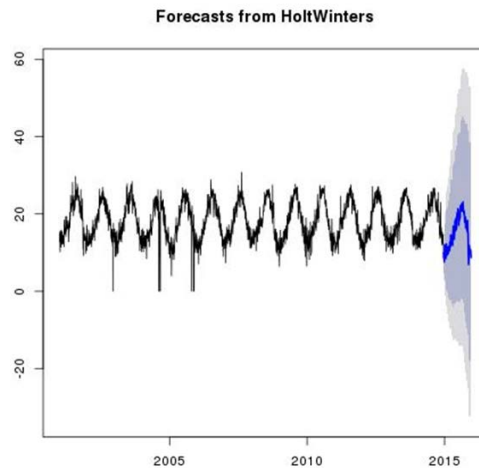
La predicción del modelo sobre sus datos de entrenamiento es óptima.

Figura 46. Visualización de predicción de temperaturas diarias del modelo para la estación meteorología adra año 2001-2014

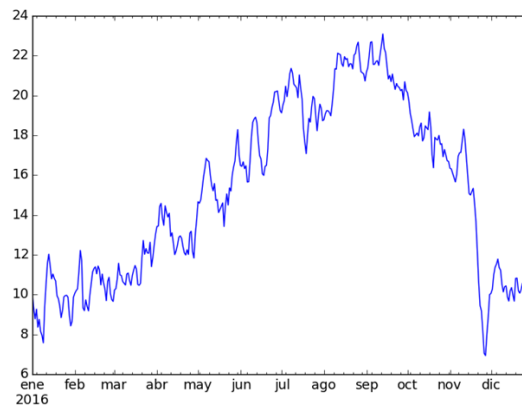


La predicción sobre el año 2015 (AÑO FUTURO PARA EL DATASET) fue el siguiente:

Figura 47. Predicción de temperaturas diarias del modelo para la estación meteorología adra año 2001-2015

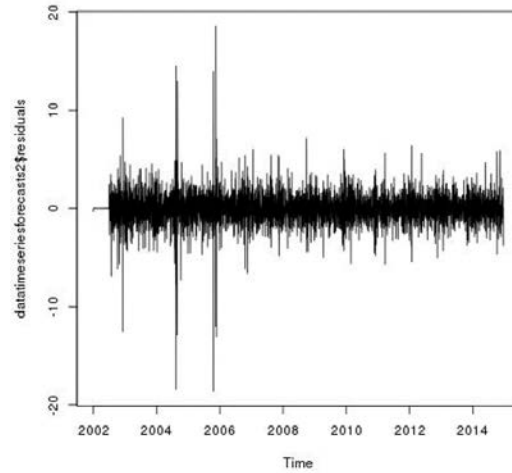


Haciendo plot de las mismas en matplotlib de Python se obtiene las predicciones de temperaturas medias para el año 2016.



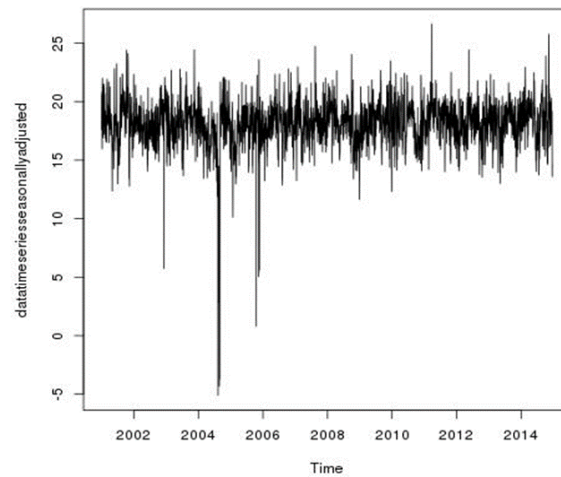
Estos son los residuos de los errores del modelo.

Figura 48. Residuos del modelo para la estación meteorología adra año 2001-2014



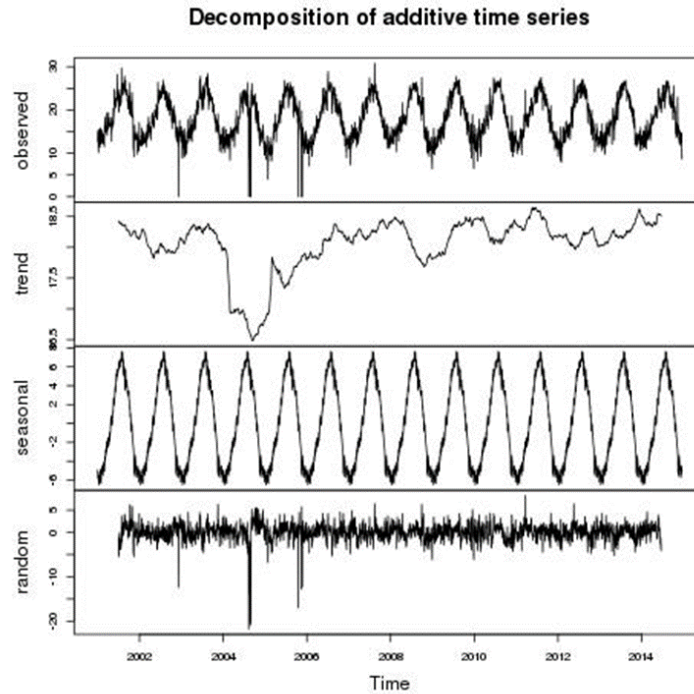
La estacionalidad ajustada es:

Figura 49. Estacionalidad ajustada del modelo para la estación meteorología adra año 2001-2014



En la descomposición se puede observar claramente cada uno de los componentes de la serie estacional:

Figura 50. Visualización de los componentes de la serie de tiempo para temperaturas diarias de la estación meteorológica adra año 2001-2015

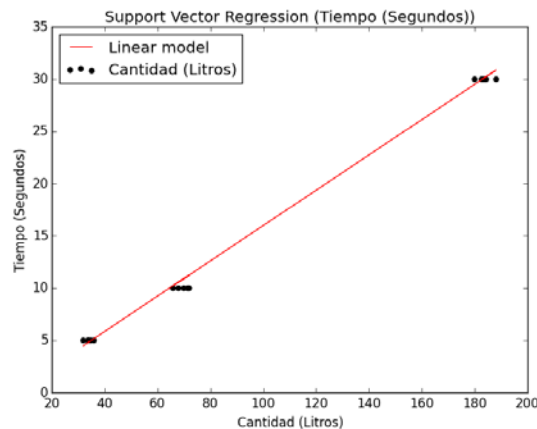


7.2.1.2 Modelo de predicción de caudal

Para cada dispositivo de riego, compuesto por la electroválvula, los aspersores, mangueras, tubería y demás, se debe determinar la cantidad de agua que el sistema puede regar por unidad de tiempo (para nuestro caso en segundos).

Ejemplo de grafica de los modelos que genera el sistema de predicción de caudal.

Figura 51. Modelo de predicción de caudal (ejemplo)



7.2.2 Sistema de Machine Learning

Para la implementación del Sistema de Machine Learning se utilizó lo siguiente:

- OpenShift como proveedor de servicios de computación en la nube para crear la aplicación web.
- Python como lenguaje de programación
- Las librerías para Machine Learning utilizadas fueron scikit-learn, matplotlib, numpy, pandas, etc.
- Flask y Flask-Restful para la implementación de los Servicios Web.
- SQLAlchemy para la conexión a la base de datos
- Postgresql como base de datos

La funcionalidad de este sistema es principalmente la de generar y almacenar los modelos de predicción de Machine Learning para las estaciones meteorológicas y las diferentes variables (temperatura, humedad, ETo, etc.) o dispositivo y la de predecir valores utilizando los modelos. Esta funcionalidad se implementó a través de Servicios Web de tipo REST.

7.2.2.1 Catálogo de Servicios Web

Tabla 13. Catálogo de Servicios Web del Sistema de Predicción

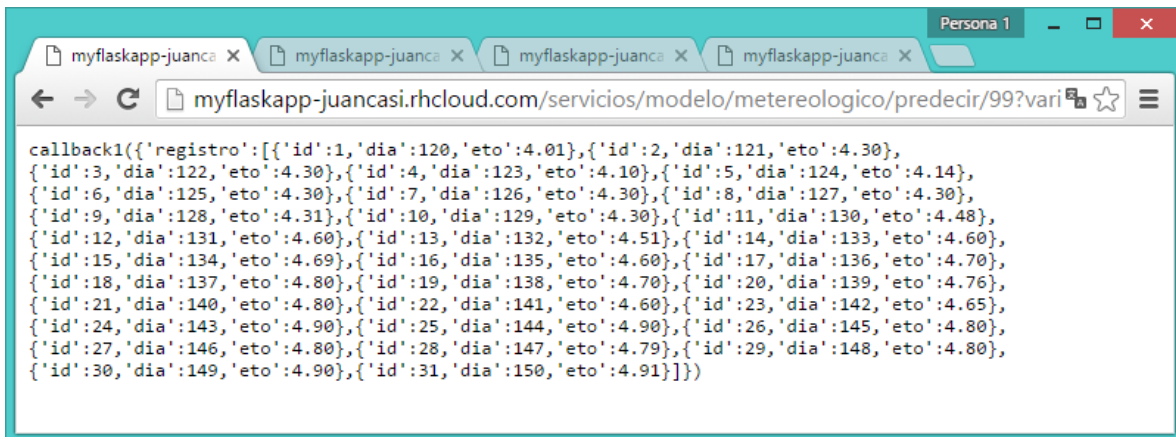
#	URL	METODO	ACCION
1	/servicios/modelo/metereologico/generar/<estacion> Parámetros: <ul style="list-style-type: none"> • estacion: (Path) el código de la estación meteorológica. • variable: (Query) el código de la variable. 	POST	Genera el modelo de predicción para una variable meteorológica de una estación.
2	/servicios/modelo/metereologico/predecir/<estacion> Parámetros: <ul style="list-style-type: none"> • estacion: (Path) el id de la estación meteorológica • variables: (Query) el conjunto de códigos de las variables que desea predecir. • nombres: (Query) el conjunto de nombres para las variables. • inicio: (Query) el día de inicio. • fin: (Query) el día final. 	GET	Retorna la predicción de un conjunto de variables de una estación meteorológica.
3	/servicios/modelo/caudal/generar/<modelo> Parámetros: <ul style="list-style-type: none"> • modelo: (Path) el código del modelo de caudal. 	POST	Genera el modelo de predicción del caudal para un dispositivo.

4	/servicios/modelo/caudal/predecir/<modelo> Parámetros: <ul style="list-style-type: none"> • modelo: (Path) el código del modelo de caudal. • cantidad: (Query) la cantidad de agua que se desea regar. 	GET	Retorna el tiempo requerido para regar una cantidad de agua.
---	--	-----	--

7.2.2.2 Listados de los Servicios Web

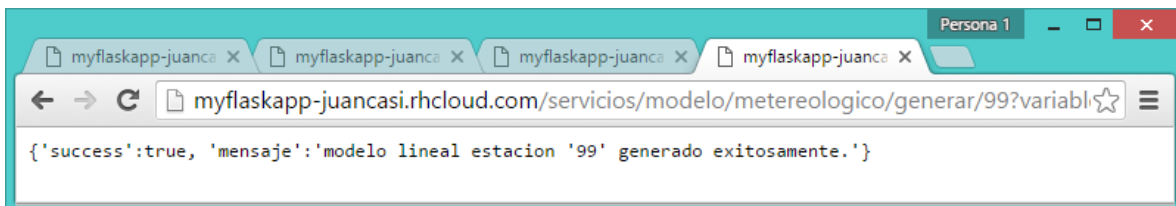
7.2.2.2.1 Servicio Web para predecir la evapotranspiración de referencia ETo

<http://myflaskapp-juancasi.rhcloud.com/servicios/modelo/metereologico/predecir/99?variables=33&nombres=eto&inicio=120&fin=150&callback=callback1>



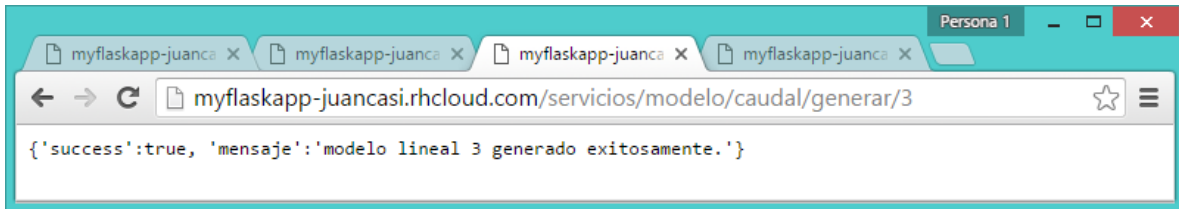
7.2.2.2.2 Servicio Web para generar el modelo de predicción de la evapotranspiración de referencia ETo

<http://myflaskapp-juancasi.rhcloud.com/servicios/modelo/metereologico/generar/99?variableX=34&variableY=28>



7.2.2.2.3 Servicio Web para generar el modelo de caudal

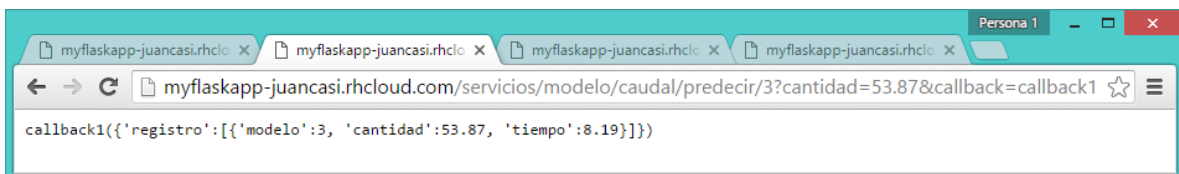
<http://myflaskapp-juancasi.rhcloud.com/servicios/modelo/caudal/generar/3>



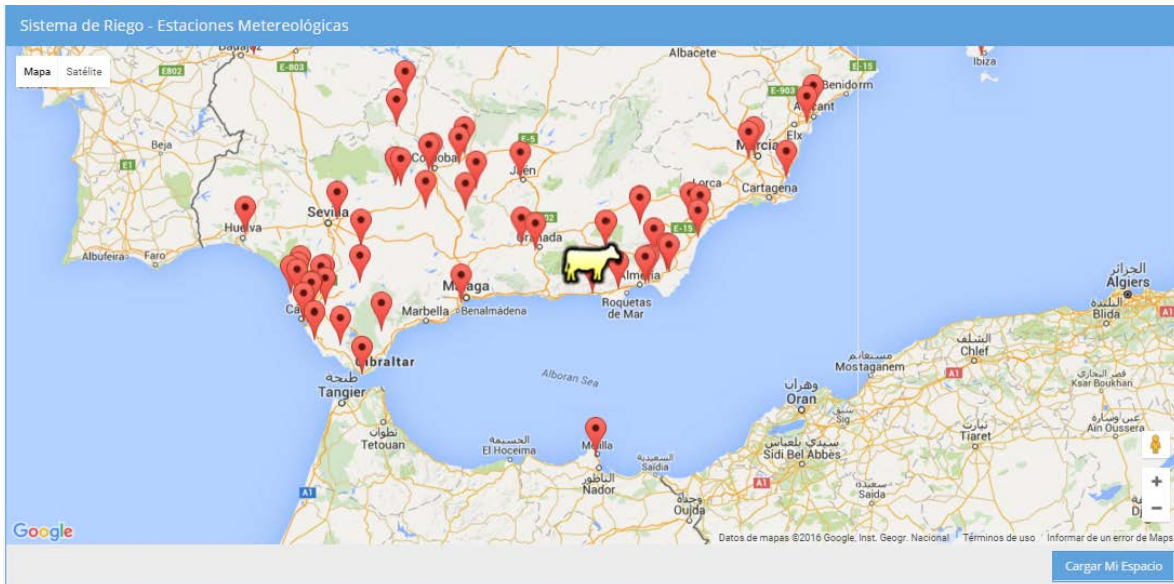
7.2.2.2.4 Servicio Web para predecir el tiempo de riego para un dispositivo (válvula)

<http://myflaskapp-juancasi.rhcloud.com/servicios/modelo/caudal/predecir/3>

`?cantidad=53.87&callback=callback1`



7.3 Aplicación Web



Gestión de Entidades.

Gestión de Entidades				
	Código	Tipo	Nombre	Estado
1	1	Agencia Estatal de Meteorología	Agencia Estatal de Meteorología - ESPAÑA	Activo
2	2	Agencia Estatal de Meteorología	Consejería de Agricultura, Pesca y Desarrollo Rural - Junta de Andalucía - ESPAÑA	Activo
3	3	Finca - Hacienda	Finca de Prueba (para cultivar) - ESPAÑA	Activo

Mostrando 1 a 3 de 3 registros

En el formulario para la gestión de las entidades se debe digitar la información requerida.

✕
Gestión de Entidades

Tipo:

Nombre:

Descripción:

Estado:

Gestión de Bloques.

Gestión de Bloques					
Código	Entidad	Tipo	Nombre	Estado	
1	Finca de Prueba (para cultivar) - ESPAÑA	Bloque - Terreno Total	Bloque Principal	Activo	

Adicionar Modificar Eliminar

<< < | Página 1 de 1 | > >> |

Mostrando 1 a 1 de 1 registros

Formulario para la gestión de los bloques.

✕
Gestión de Bloques

Entidad:

Tipo:

Nombre:

Descripción:

Pisos:

Estado:

Modificar
 Cancelar

Gestión de espacios .

Gestión de Espacios					
	Código	Bloque	Nombre	Descripción	Estado
1	1	Bloque Principal	Lote para cultivar 01	Espacio para cultivar	Activo
2	2	Bloque Principal	Lote para cultivar 02	Espacio dedicado al cultivo	Activo

Mostrando 1 a 2 de 2 registros

Formulario para la gestión de los espacios.

✕
Gestión de Espacios

Bloque:

Nombre:

Descripción:

Area: Piso:

Estado:

Gestión de los espacios para cultivar.

Gestión de Espacios para Cultivos									
	Código	Espacio	Cultivo	Nombre	Metodo Riego	Tipo Suelo	Fecha Siembra	Hora Riego	Estado
1	1	Lote para cultivar 01	Maíz (Grano)	MAIZ 2016	Riego por aspersión	Franco	2016-05-01	06:00:00	Activo
2	3	Lote para cultivar 02	Judía verde	FRIJOL 2016 1	Riego por aspersión	Franco	2016-05-01	07:00:00	Activo

Mostrando 1 a 2 de 2 registros

Formulario para la gestión de espacios para cultivar.

Gestión de Espacios para Cultivos

Espacio:

Nodo Predictor:

Cultivo:

Metodo de Riego: Tipo de Suelo:

Nombre:

Descripción:

Fecha de Siembra:

Hora de Riego:

Eficiencia del Metodo de Riego:

Nivel de Agotamiento Permisible:

Estado:

Gestión de dispositivos.

Gestión de Dispositivos						
➕ Adicionar ✎ Modificar 🗑 Eliminar						
	codigo	Tipo	nombre	descripcion	acronimo	Estado
1	27	Estación Metereológica	Estación Metereológica - ESPAÑA	Generica	A3	Activo
2	2	Sensor Temperatura	rr		rr	Activo
3	31	Sensor Humedad	Estación Metereológica de Prueba		EMA-JACS	Activo
4	33	Nodo Sensor	Nodo Sensor 01		sensor01	Activo
5	30	Estación Metereológica	Estacion Metereológica - Junta de Andalucia	Generica	EMA	Activo
6	32	Sistema de Riego	Sistema de Riego de Prueba		RIEGO-JACS	Activo
7	4	Sensor Temperatura	Sensor de temperatura y humedad - DHT22	Sensor de temperatura y humedad, d..		Activo
8	3	Sensor Temperatura	Sensor Presion 00		SP	Activo
9	1	Sensor Temperatura	Sensor Temperatura 00		ST	Activo

<< < | Page of 1 | > >> | 🔄

Mostrando 1 a 9 de 9 registros

Formulario para la gestión de los dispositivos.

✕
Gestión de Dispositivos

Tipo:

Nombre:

Descripción:

Acronimo:

Estado:

✎ Modificar
✖ Cancelar

Gestión de ítems por dispositivo.

Gestión de Items por Dispositivo							
➕ Adicionar ✎ Modificar 🗑 Eliminar							
	Codigo	Dispositivo	Tipo Medición	Nombre	Tipo Pin	Acrónimo	Estado
1	2	Estación Metereológica - ESPAÑA	Humedad relativa media mensual (°C)	Humedad relativa media mensual (°C)	ANALOGO		Activo
2	4	Estación Metereológica - ESPAÑA	Insolación total mensual, en horas	Insolación total mensual, en horas	ANALOGO		Activo
3	5	Estación Metereológica - ESPAÑA	Precipitación total mensual en mm	Precipitación total mensual en mm	ANALOGO		Activo
4	6	Estación Metereológica - ESPAÑA	Temperatura media de las máximas (°C)	Temperatura media de las máximas (°C)	ANALOGO		Activo
5	7	Estación Metereológica - ESPAÑA	Temperatura media de las mínimas (°C)	Temperatura media de las mínimas (°C)	ANALOGO		Activo
6	3	Estación Metereológica - ESPAÑA	Temperatura media mensual (°C)	Temperatura media mensual (°C)	ANALOGO		Activo
7	1	Estación Metereológica - ESPAÑA	Tensión de vapor en hPa	Tensión de vapor en hPa	ANALOGO		Activo
8	48	Estación Metereológica de Prueba	Humedad Media	Humedad (%)	ANALOGO	humedad	Activo
9	46	Estación Metereológica de Prueba	Precipitación	Precipitación	ANALOGO	precipitacion	Activo
10	47	Estación Metereológica de Prueba	Temperatura Media	Temperatura (°C)	ANALOGO	temperatura	Activo

Mostrando 26 a 31 de 31 registros

Formulario para la gestión de los ítems por dispositivo.

✕
Gestión de Items por Dispositivo

Dispositivo:

Tipo Medición:

Nombre:

Descripción:

Pin Nombre:

Tipo Pin:

Acronimo:

Id:

Estado:

✎ Modificar
✖ Cancelar

Gestión de dispositivos asignados a un espacio.

Gestión de Dispositivos por Espacio				
	Codigo	Espacio	Dispositivo	Estado
1	3	Lote para cultivar 01	Estación Metereológica de Prueba	Activo
2	2	Lote para cultivar 01	Nodo Sensor 01	Activo
3	1	Lote para cultivar 01	Sistema de Riego de Prueba	Activo
4	4	Lote para cultivar 02	Sistema de Riego de Prueba	Activo

Mostrando 1 a 4 de 4 registros

Formulario para gestionar los dispositivos por espacio.

✕
Gestión de Dispositivos por Espacio

Espacio:

Dispositivo:

Estado:

Gestión de Nodos.

Gestión de Nodos									
➕ Adicionar ✎ Modificar 🗑 Eliminar									
	Código	Entidad	Tipo	Municipio	Nombre	Latitud	Longitud	Altitud	Estado
1	3	Agencia Estatal de Met...	Estación	TARRAG...	0016A - REUS AEROPUERTO	41°08'5...	01°10'4...	71	Activo
2	4	Agencia Estatal de Met...	Estación	BARCEL...	0076 - BARCELONA AEROPUERTO	41°17'3...	02°04'1...	4	Activo
3	5	Agencia Estatal de Met...	Estación	BARCEL...	0200E - BARCELONA, FABRA	41°25'0...	02°07'2...	412	Activo
4	6	Agencia Estatal de Met...	Estación	GIRONA	0367 - GIRONA AEROPUERTO	41°54'4...	02°45'4...	143	Activo
5	7	Agencia Estatal de Met...	Estación	GIPUZK...	1014 - HONDARRIBIA, MALKARROA	43°21'2...	01°47'3...	4	Activo
6	8	Agencia Estatal de Met...	Estación	GIPUZK...	1024E - SAN SEBASTIÁN, IGUELDO	43°18'2...	02°02'2...	251	Activo
7	9	Agencia Estatal de Met...	Estación	BIZKAIA	1082 - BILBAO AEROPUERTO	43°17'5...	02°54'2...	42	Activo
8	10	Agencia Estatal de Met...	Estación	CANTAB...	1109 - SANTANDER AEROPUERTO	43°25'4...	03°49'5...	5	Activo
9	11	Agencia Estatal de Met...	Estación	ASTURIAS	1212E - ASTURIAS AEROPUERTO	43°34'0...	06°02'3...	127	Activo
10	12	Agencia Estatal de Met...	Estación	ASTURIAS	1249I - OVIEDO	43°21'1...	05°52'2...	336	Activo

⏪ ⏩ | Página de 5 | ⏪ ⏩ | 🔄
Mostrando 1 a 25 de 125 registros

Formulario para gestionar los nodos.

✕
Gestión de Nodos

Tipo:

Entidad:

Municipio:

Nombre:

Descripción:

Acrónimo:

Dirección:

Latitud: Longitud:

Altitud: Estado:

✎ Modificar
✕ Cancelar

Gestión de dispositivos por nodo.

Gestión de Dispositivos por Nodo				
+ Adicionar ✎ Modificar ✖ Eliminar				
	Codigo	Nodo	Dispositivo	Estado
851	598	C929I - HIERRO AEROPUERTO	Sensor Temperatura 00	Activo
852	861	Estación Meteorológica de Adamuz	Estación Metereológica - ESPAÑA	Activo
853	860	Estación Meteorológica de Adra	Estación Metereológica - ESPAÑA	Activo

Mostrando 851 a 853 de 853 registros

Formulario para asignar los dispositivos a un nodo.

✕
Gestión de Dispositivos por Nodo

Nodo:

Dispositivo:

Estado:

✎ Modificar
✖ Cancelar

Gestión del calendario de riego.

Cultivo a Programar

Espacio para Cultivar: Espacio: Lote para cultivar 02 Cultivo: Judía verde
 Método de Riego: Riego por aspersión Hora de Riego: 07:00:00 Fecha de Siembra: 2016-05-01
 Tipo de Suelo: Franco Nivel de Agotamiento Permissible: 0.65 Eficiencia del Método de Riego: 75
 Nodo Predictor: Estación Meteorológica de Adra

Calendario de Riego

| |

Fecha Riego	Hora Riego	Cantidad	Unidad	Estado
2016/02/10	07:00	28.60	Litros / Minuto	Activo
2016/03/26	07:00	114.40	Litros / Minuto	Activo
2016/02/24	07:00:00	28.6	Litros / Minuto	Activo

Formulario para predecir el calendario de riego.

Calendario de Riego									
	Fase	Fecha	ETr	Kc	ET	Pr	DAS	NAP	Nb
24	1	2016-02-06	2.10	0.35	0.73	0.20	15.72	21.45	0.00
25	1	2016-02-07	2.10	0.35	0.73	0.20	16.46	21.45	0.00
26	1	2016-02-08	2.15	0.35	0.75	0.20	17.21	21.45	0.00
27	2	2016-02-09	2.30	0.70	1.61	0.20	18.82	21.45	0.00
28	2	2016-02-10	2.30	0.70	1.61	0.20	20.43	21.45	0.00
29	2	2016-02-11	2.10	0.70	1.47	0.20	21.90	21.45	28.60
30	2	2016-02-12	2.05	0.70	1.43	0.20	1.43	21.45	0.00
31	2	2016-02-13	2.10	0.70	1.47	0.20	2.90	21.45	0.00
32	2	2016-02-14	2.00	0.70	1.40	0.20	4.30	21.45	0.00
33	2	2016-02-15	2.02	0.70	1.41	0.20	5.72	21.45	0.00
34	2	2016-02-16	2.30	0.70	1.61	0.20	7.33	21.45	0.00

Gestión de la programación del riego.

Cultivo a Programar

Espacio para Cultivar: FRIJOL 2016 1 Espacio: Lote para cultivar 02 Cultivo: Judía verde

Método de Riego: Riego por aspersion Hora de Riego: 07:00:00 Fecha de Siembra: 2016-05-01

Tipo de Suelo: Franco Nivel de Agotamiento Permisible: 0.65 Eficiencia del Método de Riego: 75

Nodo Predictor: Estación Meteorológica de Adra

Programación del Riego

Adicionar
 Modificar
 Eliminar

	Código	Nombre	Cantidad	Unidad	Fecha Inicio	Fecha Fin	Duración	Estado
1	5	FRIJOL 2016 1	34	Litros / Minuto	2016/01/13 01:11:50	2016/01/13 01:12:30	14	No Activo
2	4	FRIJOL 2016 1	12	Litros / Minuto	2016/01/13 01:14:10	2016/01/13 01:14:50	0	Activo
3	8	FRIJOL 2016 1	12	Litros / Minuto	2016/01/13 09:13:50	2016/01/13 09:14:39	0	Activo

« < | Página 1 de 1 | > »
Mostrando 1 a 3 de 3 registros

Formulario para la gestión de la programación del riego.

✕
Programación del Riego

Espacio para Cultivar: FRIJOL 2016 1

Fecha Inicio: 2016/01/13 Hora Inicio: 01:14:10 ▼

Fecha Fin: 2016/01/13 Hora Fin: 01:14:50 ▼

Cantidad: 12 Duración: 0

Unidad: Litros / Minuto ▼ Estado: Activo ▼

Modificar
 Cancelar

Gestión de eventos.

Evento a Reportar

Espacio para Cultivar:	FRIJOL 2016 1	Espacio:	Lote para cultivar 02	Cultivo:	Judía verde
Método de Riego:	Riego por aspersión	Hora de Riego:	07:00:00	Fecha de Siembra:	2016-05-01
Tipo de Suelo:	Franco	Nivel de Agotamiento Permisible:	0.65	Eficiencia del Método de Riego:	75
Nodo Predictor:	Estación Meteorológica de Adra				

Eventos del Riego

Adicionar
 Modificar
 Eliminar

	Código	Cultivo	Tipo Evento	Cantidad	Unidad	Fecha Hora	Comentario	Estado
1	2	FRIJOL 2016 1	RIEGO	12	Litros / Minuto	2016/01/13 01:57:10		Activo
2	3	FRIJOL 2016 1	LLUVIA	2	Litros / Minuto	2016/01/13 01:58:38	FUERTE LLUVIA	Activo
3	1	FRIJOL 2016 1	LLUVIA	6	Litros / Minuto	2016/01/13 01:47:52		Activo

« < | Página 1 de 1 | > » |
Mostrando 1 a 3 de 3 registros

Formulario para la gestión de eventos sobre el riego.

✕
Eventos del Riego

Espacio para Cultivar: FRIJOL 2016 1

Tipo Evento:

Fecha: Hora:

Cantidad: Unidad:

Comentario:

Estado:

Modificar
 Cancelar

Gestión de los dispositivos, para cada espacio se presenta la información con los dispositivos y elementos (sensores, actuadores) que lo componen.

Lote para cultivar 01

Estación Metereológica de Prueba

1	Humedad (%)	Información
2	Precipitación	Información
3	Temperatura (°C)	Información

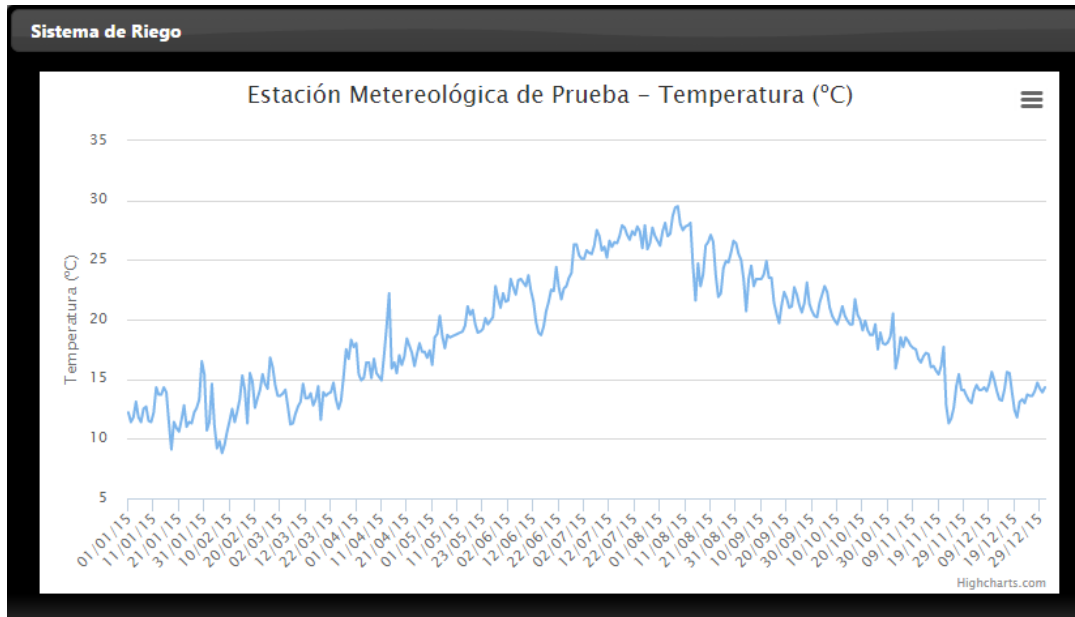
Nodo Sensor 01

		Nivel Humedad Suelo
1	Humedad en Suelo 01 - 30 cm	Información
2	Humedad en Suelo 02 - 60 cm	Información
3	Humedad en Suelo 03 - 90 cm	Información
4	Temperatura en Suelo 01	Información

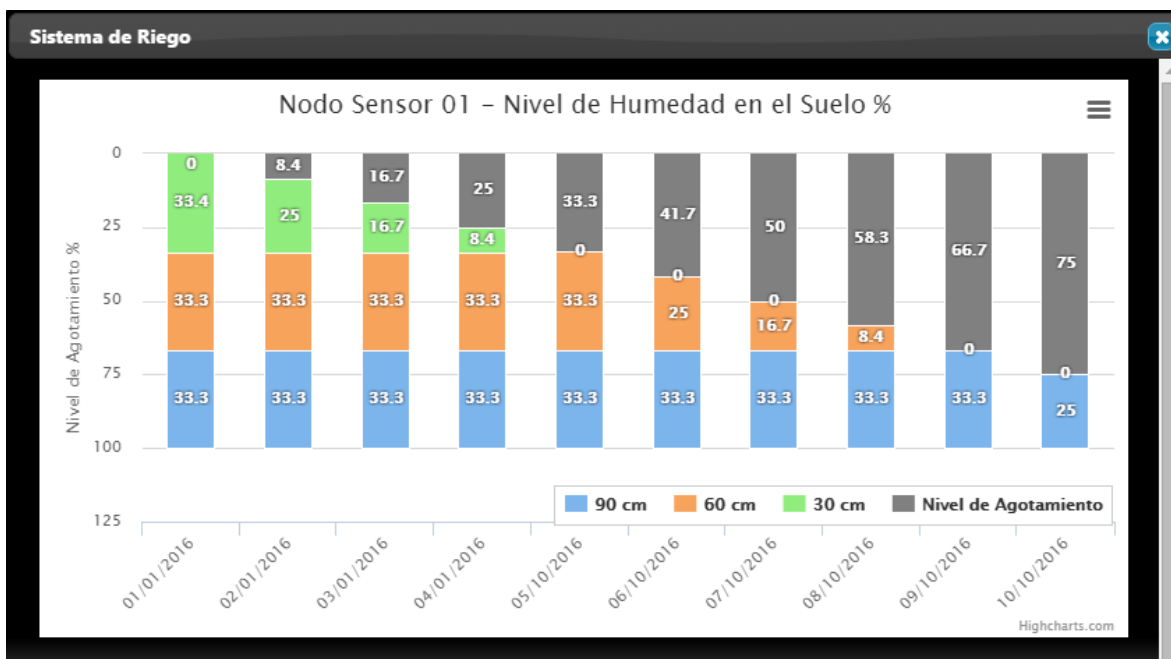
Sistema de Riego de Prueba

1	ElectroVálvula	Prender <input type="checkbox"/>
2	Moto Bomba	<input type="checkbox"/> Apagar

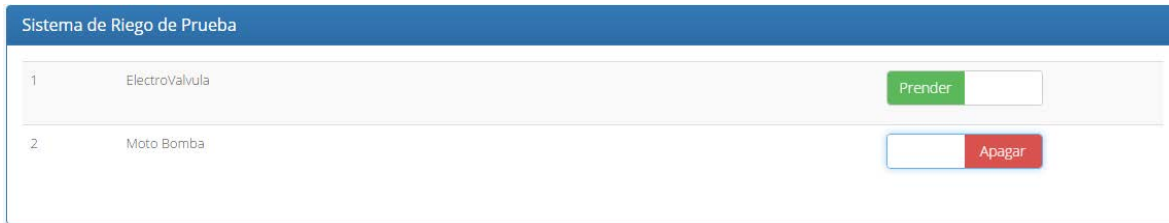
Al hacer click en el boton [Información] de cada una de los elementos que hacen parte de un dispositivo (sensores) se puede visualizar el comportamiento en el tiempo de la variable meteorológica.



Cuando se hace click en el boton [Nivel Humedad Suelo] se puede ver el comportamiento del conjunto de sensores ubicados a diferentes niveles de profundidad en el suelo (30 cm, 60 cm, 90 o 120 cm). Este reporte nos permite monitorear el nivel de agotamiento permisible y determinar cuándo hacer el riego.



Al hacer click en el boton [Prender / Apagar] se puede activar o no los dispositivos electrónicos (actuadores) tales como electroválvulas, motobombas, etc.



7.4 Seguridad

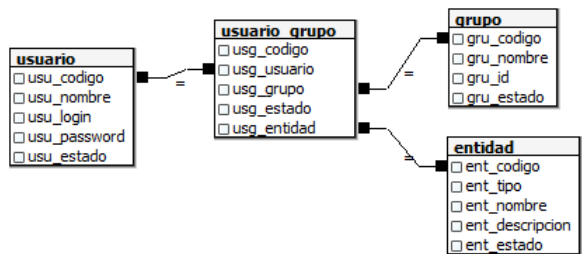
La seguridad en JAAS se puede implementar con archivos, ldap, etc. En este sistema se va a implementar con una base de datos en postgresql. En el archivo de configuración del servidor del wildfly (standalone.xml) se especifica el nombre del pool de conexiones que se va a emplear y las consultas SQL que retornan la clave y los roles de un usuario.

Figura 52. Configuración del dominio de seguridad

```
<security-domain name="seguridadRiego">
  <authentication>
    <login-module code="Database" flag="required">
      <module-option name="dsJndiName" value="java:jboss/datasources/PostgreSQLDS"/>
      <module-option name="principalsQuery" value="select usu_password
from usuario where usu_login = ?"/>
      <module-option name="rolesQuery" value="SELECT grupo.gru_id, 'Roles'
FROM public.usuario, public.usuario_grupo, public.grupo
WHERE usuario.usu_codigo = usuario_grupo.usg_usuario
AND grupo.gru_codigo = usuario_grupo.usg_grupo and usu_login=?"/>
      <module-option name="hashAlgorithm" value="MD5"/>
      <module-option name="hashEncoding" value="base64"/>
    </login-module>
  </authentication>
</security-domain>
```

Para implementar la seguridad en java utilizando JAAS, se crearon las tablas usuario, entidad, grupo y usuario_grupo.

Figura 53. Tablas para la seguridad



Al crear los usuarios se encriptaron las claves utilizando el algoritmo MD5 y un encoding base64.

Figura 54. Consulta de usuarios y grupos

	ent_nombre character varying(100)	usu_nombre character varying(50)	usu_login character varying(20)	usu_password character varying(64)	gru_id character varying(100)
1	Finca de Prueba (para cultivar) - ESPAÑA	Pedro Perez	pedro	dlunU7YJ2Es4E5kf4j+Bsw==	ADMINISTRADOR
2	Finca de Prueba (para cultivar) - ESPAÑA	Maria Cardenas	maria	GiYdglRzdpXWDZd1Egsc1Q==	AGRICULTOR
3	Finca de Prueba (para cultivar) - ESPAÑA	Jose Lopez	jose	xaGphkmnh03g3vCT6xNiYg==	ADMINISTRADOR
4	Finca de Prueba (para cultivar) - ESPAÑA	Jose Lopez	jose	xaGphkmnh03g3vCT6xNiYg==	AGRICULTOR

El mecanismo de seguridad nos permite proteger los recursos (url-pattern) por grupo (seguridad basada en roles), esto se hace en el archivo de configuración de la aplicación web, en el archivo web.xml.

Figura 55. Ejemplo de restricción de seguridad

```

<security-constraint>
  <display-name>Restricción de Seguridad - Agricultor</display-name>
  <web-resource-collection>
    <web-resource-name>Area de Agricultor</web-resource-name>
    <url-pattern>/programacion/*</url-pattern>
    <http-method>DELETE</http-method>
    <http-method>GET</http-method>
    <http-method>POST</http-method>
    <http-method>PUT</http-method>
  </web-resource-collection>
  <auth-constraint>
    <role-name>AGRICULTOR</role-name>
  </auth-constraint>
  <user-data-constraint>
    <transport-guarantee>NONE</transport-guarantee>
  </user-data-constraint>
</security-constraint>
    
```

En la sección login-config se define el método de autenticación, para nuestro caso basada en formularios.

Figura 56. Configuración de la autenticación

```

<login-config>
  <auth-method>FORM</auth-method>
  <realm-name>seguridadRiego</realm-name>
  <form-login-config>
    <form-login-page>/login.jsp</form-login-page>
    <form-error-page>/error.jsp</form-error-page>
  </form-login-config>
</login-config>

```

También es posible asegurar el acceso a los servicios web con JAAS a través de anotaciones en los métodos (@RolesAllowed), en este caso solo el grupo ADMINISTRADOR tendrá acceso al recurso.

Figura 57. Anotaciones para asegurar el acceso a los servicios web

```

@RolesAllowed("ADMINISTRADOR")
@POST
@Path("/adicionar")
@Consumes({"application/json"})
@Produces("application/json")
public Response adicionar(Item item){
    String direccion = item.getDireccion();
    String datos = item.getDatos();
}

```

Los módulos Xbee también permiten encriptar la información que se transmite mediante los parámetros de seguridad.

▼ Security
Change security parameters

EE Encryption Enable	Enabled [1]
EO Encryption Options	0 Bitfield
KY Encryption Key	
NK Network Encryption Key	

8. CONCLUSIONES Y RECOMENDACIONES

Es importante facilitarle al usuario final la interacción con el sistema debido a que está pensado para el sector agrícola en donde la alfabetización digital podría ser muy baja. El uso del Servidor Local ayuda a abaratar los costos de operación así como la posibilidad de controlar el Sistema de Riego desde una aplicación móvil conectada a través de Bluetooth o WiFi. Es importante resaltar las posibilidades de Java como lenguaje pues se ha utilizado en todos los escenarios posibles tales como la aplicación Web, la programación de la Raspberry PI, la conexión con los nodos coordinadores (Xbee-api), la creación de los servicios web (RestEasy) y la creación de la aplicación en la nube (OpenShift).

También se debe resaltar la facilidad con que se pudo implementar el sistema en la nube con OpenShift, la flexibilidad para tener dos aplicaciones con lenguajes de programación distintos Java y Python y Postgres como la base de datos compartida. La facilidad para implementar los algoritmos de Machine Learning en Python.

Como recomendaciones para un trabajo futuro está la utilización de Apache Spark para la parte de Big Data y Machine Learning, hacer la prueba con datos en tiempo real de una estación agroclimatológica desde la fecha de siembra hasta la cosecha.

9. REFERENCIAS

- Allamaraju, S. (2010). *RESTful Web Services Cookbook*. O'Reilly Media, Inc.
- Allen G., R., Pereira, L. S., Raes, D., & Smith, M. (2006). Evapotranspiración del cultivo: Guías para la determinación de los requerimientos de agua de los cultivos. *FAO :Estudios FAO Riego Y Drenaje 56*, 297.
- Arduino - Sitio Oficial. (2015). Retrieved from <https://www.arduino.cc/en/Guide/Introduction>
- Beecham Research, L. (2014). *Towards Smart Farming: Agriculture Embracing The IoT vision* (Vol. 44).
- Bouguettaya, A., Sheng, Q. Z., & Daniel, F. (2014). *Advanced Web Services. Pro PHP: Patterns, Frameworks, Testing and More*. Springer Science & Business Media. Retrieved from http://link.springer.com/chapter/10.1007/978-1-4302-0279-0_20
- Evans, D. (2011). *The Internet of Things: How the Next Evolution of the Internet Is Changing Everything*.
- Faludi, R. (2011). *Building Wireless Sensor Networks*. O'Reilly Media, Inc.
- FAO. (2002). Agua y Cultivos-Logrando el uso optimo del agua en el agricultura, 28. Retrieved from ftp://ftp.fao.org/agl/aglw/docs/cropsdrops_s.pdf
- Fernandez Gomez, R., Avila Alabarces, R., Lopez Rodriguez, M., Gavilan Zafra, P., & Oyonarte Gutierrez, N. A. (2010). *Fundamentos del Riego*.
- Food and Agriculture Organization. (2013). *Climate-Smart Agriculture Sourcebook*. Retrieved from <http://www.fao.org/docrep/018/i3325e/i3325e00.htm>
- Forecasting & Time Series Analysis. (2016). Retrieved from ucAnalytis.com
- Gartner's Hype Cycle for Emerging Technologies. (2015). Retrieved December 1, 2015, from <http://www.gartner.com/newsroom/id/3114217>
- Goodwin, S. (2010). *Smart Home Automation with Linux*. <http://doi.org/10.1007/978-1-4302-2779-3>
- Hackeling, G. (2014). *Mastering Machine Learning with scikit-learn*. Retrieved from <http://books.google.com/books?id=fZQeBQAAQBAJ&pgis=1>
- Inoue, T., Hayakawa, A., & Kamei, T. (2011). China ' s Initiative for the Internet of Things and Opportunities for China ' s Initiative for the Internet of, (165).
- Introduction to time series analysis. (2016). Retrieved from <http://www.itl.nist.gov/div898/handbook/pmc/section4/pmc4.htm>
- JAAS Reference Guide. (2015). Retrieved from <http://docs.oracle.com/javase/7/docs/technotes/guides/security/jaas/JAASRefGuide.html>

- Miorandi, D., Sicari, S., De Pellegrini, F., & Chlamtac, I. (2012). Internet of things: Vision, applications and research challenges. *Ad Hoc Networks*, 10(7), 1497–1516. <http://doi.org/10.1016/j.adhoc.2012.02.016>
- Mitchel, T. (1997). *Machine Learning*. McGraw Hill.
- National Institute of Standards and Technology. (2009). The NIST Definition of Cloud Computing. Retrieved December 1, 2015, from <http://www.nist.gov/itl/cloud/upload/cloud-def-v15.pdf>
- OpenShift. (2015). What is OpenShift? Retrieved May 20, 2012, from <https://developers.openshift.com/en/overview-what-is-openshift.html>
- OWASP. (2015). Retrieved from https://www.owasp.org/index.php/About_OWASP#The_OWASP_Foundation
- OWASP Enterprise Security API. (2015). Retrieved from <https://www.owasp.org/index.php/ESAPI>
- Shumway, R. H., & Stoffer, D. S. (2011). *Time Series Analysis and Its Applications With R Examples*. Book. <http://doi.org/10.1007/978-1-4419-7865-3>
- Sousa, M. de. (2015). *Internet of Things With Intel Galileo*. Packt Publishing Ltd. Retrieved from <https://developer.ibm.com/iot/recipes/intel-galileo/>
- Webber, J., Parastatidis, S., & Ian, R. (2010). *REST in Practice: Hypermedia and Systems Architecture*. O'Reilly Media, Inc.
- Whitmore, A., Agarwal, A., & Da Xu, L. (2015). The Internet of Things—A survey of topics and trends. *Information Systems Frontiers*, 17(2), 261–274. <http://doi.org/10.1007/s10796-014-9489-2>

ANEXOS

ANEXO 1. CASOS DE USO

CASO DE USO	
Código:	001
Nombre:	Gestionar Entidades
Actor:	Usuario

Descripción:

El sistema permitirá al usuario adicionar, modificar y eliminar las entidades que se encuentren registradas en el sistema.

Pre-Condiciones:

- El usuario debe estar registrado en el sistema.
- El usuario debe estar activo.
- El usuario debe ingresar al sistema.

Post-Condiciones:

- La cantidad de entidades cambiará dependiendo de lo hecho por el usuario; o la información de algunos ya existentes.

Prioridad:

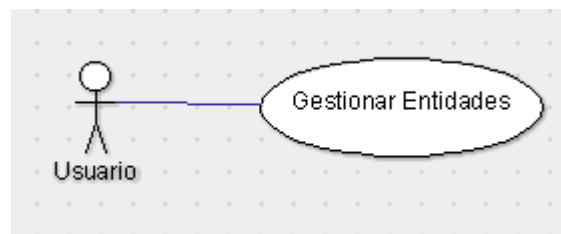
MEDIA

Actores:

Los usuarios registrados en el sistema.

Observaciones:

En caso de presentar problema al momento de adicionar, modificar o eliminar una entidad, verificar si la sesión esta aún vigente.



CASO DE USO

Código:	002
Nombre:	Gestionar Bloques
Actor:	Usuario

Descripción:

El sistema permitirá al usuario adicionar, modificar y eliminar los bloques que se encuentren registrados en el sistema.

Pre-Condiciones:

- El usuario debe estar registrado en el sistema.
- El usuario debe estar activo.
- El usuario debe ingresar al sistema.

Post-Condiciones:

- La cantidad de bloques cambiará dependiendo de lo hecho por el usuario; o la información de algunos ya existentes.

Prioridad:

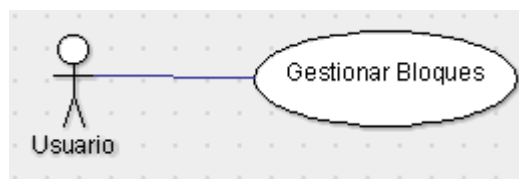
MEDIA

Actores:

Los usuarios registrados en el sistema.

Observaciones:

En caso de presentar problema al momento de adicionar, modificar o eliminar un bloque, verificar si la sesión esta aún vigente.



CASO DE USO

Código:	003
Nombre:	Gestionar Espacios
Actor:	Usuario

Descripción:

El sistema permitirá al usuario adicionar, modificar y eliminar los espacios que se encuentren registrados en el sistema.

Pre-Condiciones:

- El usuario debe estar registrado en el sistema.
- El usuario debe estar activo.
- El usuario debe ingresar al sistema.

Post-Condiciones:

- La cantidad de espacios cambiará dependiendo de lo hecho por el usuario; o la información de algunos ya existentes.

Prioridad:

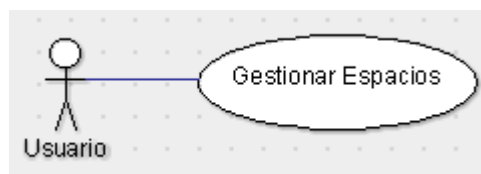
MEDIA

Actores:

Los usuarios registrados en el sistema.

Observaciones:

En caso de presentar problema al momento de adicionar, modificar o eliminar un espacio, verificar si la sesión esta aún vigente.



CASO DE USO

Código:	004
Nombre:	Gestionar Espacios para Cultivar
Actor:	Usuario

Descripción:

El sistema permitirá al usuario asignar, modificar y eliminar los espacios programados para ser cultivados que se encuentren registrados en el sistema.

Pre-Condiciones:

- El usuario debe estar registrado en el sistema.
- El usuario debe estar activo.
- El usuario debe ingresar al sistema.

Post-Condiciones:

- La cantidad de espacios dispuestos para ser cultivados cambiará dependiendo de lo hecho por el usuario; o la información de algunos ya existentes.

Prioridad:

MEDIA

Actores:

Los usuarios registrados en el sistema.

Observaciones:

En caso de presentar problema al momento de asignar, modificar o eliminar un espacio para ser cultivado, verificar si la sesión esta aún vigente.



CASO DE USO

Código:	005
Nombre:	Gestionar Dispositivos
Actor:	Usuario

Descripción:

El sistema permitirá al usuario adicionar, modificar y eliminar los dispositivos que se encuentren registrados en el sistema.

Pre-Condiciones:

- El usuario debe estar registrado en el sistema.
- El usuario debe estar activo.
- El usuario debe ingresar al sistema.

Post-Condiciones:

- La cantidad de dispositivos cambiará dependiendo de lo hecho por el usuario; o la información de algunos ya existentes.

Prioridad:

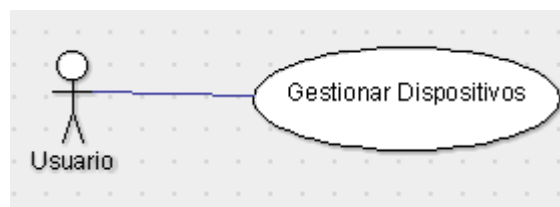
MEDIA

Actores:

Los usuarios registrados en el sistema.

Observaciones:

En caso de presentar problema al momento de adicionar, modificar o eliminar un dispositivo, verificar si la sesión esta aún vigente.



CASO DE USO

Código:	006
Nombre:	Gestionar Ítems por Dispositivo
Actor:	Usuario

Descripción:

El sistema permitirá al usuario adicionar, modificar y eliminar ítems para un dispositivo específico que se encuentren registrados en el sistema.

Pre-Condiciones:

- El usuario debe estar registrado en el sistema.
- El usuario debe estar activo.
- El usuario debe ingresar al sistema.

Post-Condiciones:

- La cantidad de ítems que tengan los dispositivos cambiará dependiendo de lo hecho por el usuario; o la información de algunos ya existentes.

Prioridad:

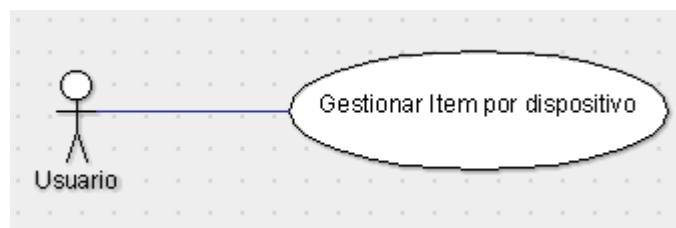
MEDIA

Actores:

Los usuarios registrados en el sistema.

Observaciones:

En caso de presentar problema al momento de adicionar, modificar o eliminar un ítem por dispositivo, verificar si la sesión esta aún vigente.



CASO DE USO

Código:	007
Nombre:	Gestionar Dispositivos por Espacios
Actor:	Usuario

Descripción:

El sistema permitirá al usuario adicionar, modificar y eliminar dispositivos para un determinado espacio que se encuentre registrado en el sistema.

Pre-Condiciones:

- El usuario debe estar registrado en el sistema.
- El usuario debe estar activo.
- El usuario debe ingresar al sistema.

Post-Condiciones:

- La cantidad de dispositivos que estén asignado a un espacio cambiará dependiendo de lo hecho por el usuario; o la información de algunos ya existentes.

Prioridad:

MEDIA

Actores:

Los usuarios registrados en el sistema.

Observaciones:

En caso de presentar problema al momento de adicionar, modificar o eliminar un dispositivo para un determinado espacio, verificar si la sesión esta aún vigente.



CASO DE USO

Código:	008
Nombre:	Gestionar Nodo
Actor:	Usuario

Descripción:

El sistema permitirá al usuario adicionar, modificar y eliminar nodos que se encuentre registrado en el sistema.

Pre-Condiciones:

- El usuario debe estar registrado en el sistema.
- El usuario debe estar activo.
- El usuario debe ingresar al sistema.

Post-Condiciones:

- La cantidad de nodos registrados en el sistema cambiará dependiendo de lo hecho por el usuario; o la información de algunos ya existentes.

Prioridad:

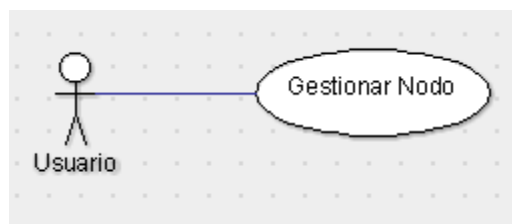
MEDIA

Actores:

Los usuarios registrados en el sistema.

Observaciones:

En caso de presentar problema al momento de adicionar, modificar o eliminar un nodo, verificar si la sesión esta aún vigente.



CASO DE USO

Código:	009
Nombre:	Gestionar Dispositivos por Nodo
Actor:	Usuario

Descripción:

El sistema permitirá al usuario adicionar, modificar y eliminar dispositivos a un específico nodo que se encuentre registrado en el sistema.

Pre-Condiciones:

- El usuario debe estar registrado en el sistema.
- El usuario debe estar activo.
- El usuario debe ingresar al sistema.

Post-Condiciones:

- La cantidad de dispositivos que tiene un nodo registrados en el sistema, cambiará dependiendo de lo hecho por el usuario; o la información de algunos ya existentes.

Prioridad:

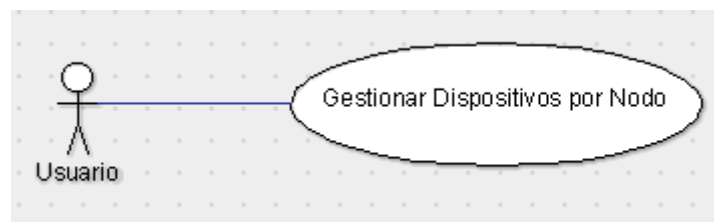
MEDIA

Actores:

Los usuarios registrados en el sistema.

Observaciones:

En caso de presentar problema al momento de adicionar, modificar o eliminar un dispositivo a un nodo del sistema, verificar si la sesión esta aún vigente.



ANEXO 2 – DICCIONARIO DE DATOS

Tablas de la base de datos (POSTGRESQL)

TABLA: BLOQUE		
Descripción: Almacenamiento de un conjunto de espacios que para el caso de la agricultura podrían ser lotes, graneros, etc.		
Nombre	Tipo de Dato	Descripción
blo_codigo	Bigserial	Llave primaria de identificación única para cada bloque.
blo_entidad	Bigint	Llave Foránea a la tabla entidad a la cual pertenece.
blo_tipo	Integer	Llave Foránea a la tabla tipo_bloque para asignar el tipo de la bloque.
blo_nombre	Character varying	EL nombre del bloque
blo_descripcion	Text	Descripción del bloque
blo_estado	Integer	Estado actual del bloque
blo_pisos	Integer	Número de pisos del bloque.

TABLA: CULTIVO		
Descripción: Datos propios del cultivo, etc.		
Nombre	Tipo de Dato	Descripción
cul_codigo	integer	Llave primaria de identificación única para cada cultivo.
cul_nombre	Character Varying	Nombre del cultivo
cul_descripcion	Text	Descripción del cultivo
cul_estado	integer	Estado actual del cultivo
cul_tipo	Integer	Llave foránea a la tabla tipo_cultivo

TABLA: CULTIVO_COEFICIENTE		
Descripción: Datos fundamentales para determinar el coeficiente del cultivo, según su fase.		
Nombre	Tipo de Dato	Descripción
cuc_codigo	serial	Llave primaria de identificación única para cada coeficiente del cultivo.
cuc_cultivo	integer	Llave Foránea a la tabla cultivo , que específica a que cultivo pertenece.
cuc_fase	Integer	Llave foránea a la tabla cultivo fase
cuc_coeficiente	Double precision	Coeficiente para determinar la evapotranspiración.
cuc_estado	Integer	Estado del coeficiente del cultivo

TABLA: CULTIVO_COEFICIENTE_BASAL		
Descripción: Datos fundamentales para determinar el coeficiente basal del cultivo, según su fase.		
Nombre	Tipo de Dato	Descripción
ccb_codigo	serial	Llave primaria de identificación única para cada coeficiente basal del cultivo.
ccb_cultivo	integer	Llave Foránea a la tabla cultivo , que específica a que cultivo pertenece.
ccb_coeficiente_ini	Double precision	Coeficiente basal del cultivo en la etapa inicial
ccb_coeficiente_med	Double precision	Coeficiente basal del cultivo en la etapa med
ccb_coeficiente_fin	Double precision	Coeficiente basal del cultivo en la etapa fin
ccb_estado	integer	Estado del Coeficiente basal del cultivo
ccb_descripcion	text	Descripción del coeficiente basal del cultivo

TABLA: CULTIVO_DURACION		
Descripción: Información correspondiente a las fases de desarrollo de un cultivo en una determinada región, etc.		
Nombre	Tipo de Dato	Descripción
cud_codigo	serial	Llave primaria de identificación única para la duración de cada cultivo.
cud_cultivo	integer	Llave Foránea a la tabla cultivo para el cultivo al cual pertenece.
cud_fase1	Integer	Numero de días de la Fase 1 de cada cultivo
cud_fase2	Integer	Numero de días de la Fase 2 de cada cultivo
cud_fase3	Integer	Numero de días de la Fase 3 de cada cultivo
cud_fase4	Integer	Numero de días de la Fase 4 de cada cultivo
cud_fecha_siembra	character varying	Fecha de la siembra
cud_region	character varying	Región en la cual se encuentra el cultivo.
cud_estado	integer	Estado

TABLA: CULTIVO_FASE		
Descripción: Información acerca de las fases del cultivo, etc.		
Nombre	Tipo de Dato	Descripción
cuf_codigo	serial	Llave primaria de identificación única para cada fase.
cuf_nombre	character varying	Nombre de cada fase.
cuf_descripcion	Text	Descripción de la fase.
cuf_acronimo	character varying	Acrónimo para identificar a cada fase

TABLA: CULTIVO_FRACCION_AGOTAMIENTO		
Descripción: Información acerca de la fracción de agotamiento del cultivo, etc.		
Nombre	Tipo de Dato	Descripción
cfa_codigo	serial	Llave primaria de identificación única para cada cultivo y su fracción de agotamiento.
cfa_cultivo	integer	Llave Foránea a la tabla cultivo para el cultivo al cual pertenece.
cfa_profundidad_min	double precision	Profundidad mínima en metros
cfa_profundidad_max	double precision	Profundidad máxima en metros
cfa_fraccion_agotamiento	double precision	Fracción de agotamiento

TABLA: CULTIVO_PROFUNDIDAD_RAICES		
Descripción: Información acerca de la profundidad de las raíces del cultivo etc.		
Nombre	Tipo de Dato	Descripción
cpr_codigo	serial	Llave primaria de identificación única para cada cultivo y su profundidad.
cpr_cultivo	integer	Llave Foránea a la tabla cultivo para el cultivo al cual pertenece.
cpr_profundidad_min	double precision	Profundidad mínima (raíz) en metros
cpr_profundidad_max	double precision	Profundidad máxima (raíz) en metros

TABLA: DEPARTAMENTO		
Descripción: Información acerca de cada departamento		
Nombre	Tipo de Dato	Descripción
dep_codigo	Bigserial	Llave primaria de identificación única para cada departamento.
dep_pais	integer	Llave Foránea a la tabla país para el cultivo al cual pertenece.
dep_nombre	character varying	Nombre del departamento
dep_acronimo	character varying	Acrónimo

TABLA: DISPOSITIVO		
Descripción: Información acerca de cada Dispositivo		
Nombre	Tipo de Dato	Descripción
dis_codigo	Bigserial	Llave primaria de identificación única para cada dispositivo.
dis_tipo	integer	Llave Foránea a la tabla tipo_dispositivo para el tipo al cual pertenece.
dis_nombre	character varying	Nombre del dispositivo
dis_descripcion	Text	Descripción del dispositivo
dis_estado	integer	Estado del dispositivo
dis_acronimo	character varying	Acrónimo del dispositivo
dis_entidad	bigint	Entidad a la que pertenece el dispositivo
dis_direccion	character varying	Dirección única del dispositivo

TABLA: DISPOSITIVO_ITEM		
Descripción: Información acerca de cada Dispositivo e ítems		
Nombre	Tipo de Dato	Descripción
dii_codigo	Bigserial	Llave primaria de identificación única para cada dispositivo e ítem.
dii_dispositivo	Bigint	Llave Foránea a la tabla dispositivo para saber a cual pertenece.
dii_tipo_medicion	integer	Llave Foránea a la tabla tipo_medicion para saber a qué tipo de medicion pertenece.
dii_nombre	character varying	Descripción del dispositivo ítem (diferentes medidas)
dii_descripcion	text	Descripción del dispositivo ítem
dii_pin_nombre	character varying	Nombre del pin
dii_tipo_pin	integer	Tipo de pin
dii_estado	integer	Estado del dispositivo ítem
dii_acronimo	character varying	Acrónimo del dispositivo ítem
dii_id	character varying	Id del dispositivo ítem

TABLA: ENTIDAD		
Descripción: Información acerca de cada entidad		
Nombre	Tipo de Dato	Descripción
ent_codigo	Bigserial	Llave primaria de identificación única para cada entidad.
ent_tipo	integer	Llave Foránea a la tabla tipo_entidad para saber a qué entidad pertenece.
ent_nombre	character varying	Nombre de la entidad
ent_descripcion	text	Descripción de la entidad
ent_estado	integer	estado de la entidad

TABLA: ENTIDAD_COORDENADA		
Descripción: Información acerca de cada entidad en su respectivas coordenadas		
Nombre	Tipo de Dato	Descripción
enc_codigo	Bigserial	Llave primaria de identificación única para cada entidad en su coordenada
enc_entidad	bigint	Llave Foránea a la tabla entidad para saber a que entidad pertenece.
enc_altitud	character varying	Altitud geográfica de la entidad
enc_latitud	character varying	Latitud geográfica de la entidad
enc_longitud	character varying	Longitud geográfica de la entidad

TABLA: ESPACIO		
Descripción: Información propia del espacio		
Nombre	Tipo de Dato	Descripción
esp_codigo	Bigserial	Llave primaria de identificación única para cada espacio
esp_bloque	bigint	Llave Foránea a la tabla bloque para saber a qué bloque pertenece.
esp_nombre	character varying	Nombre del espacio
esp_descripcion	text	Descripción del espacio
esp_area	double precision	Área del espacio
esp_estado	integer	Estado del espacio
esp_piso	integer	Piso en el cual está el espacio
esp_latitud	character varying	Latitud geográfica de la espacio
esp_longitud	character varying	Longitud geográfica de la espacio
esp_altitud	character varying	Altitud geográfica de la espacio

TABLA: ESPACIO_COORDENADA		
Descripción: Información propia del espacio		
Nombre	Tipo de Dato	Descripción
esc_codigo	Bigserial	Llave primaria de identificación única para cada espacio en sus coordenadas
esc_espacio	bigint	Llave Foránea a la tabla espacio para saber a qué espacio pertenece.
esc_altitud	character varying	Altitud geográfica de la espacio
esc_latitud	text	Latitud geográfica de la espacio
esc_longitud	double precision	Longitud geográfica de la espacio

TABLA: ESPACIO_CULTIVO		
Descripción: Información propia del espacio y su cultivo		
Nombre	Tipo de Dato	Descripción
esc_codigo	Bigserial	Llave primaria de identificación única para cada espacio en el cultivo
esc_espacio	bigint	Llave Foránea a la tabla espacio para saber a qué espacio pertenece.
esc_cultivo	bigint	Llave Foránea a la tabla cultivo para saber a qué espacio pertenece.
esc_nombre	Character varying	Nombre del cultivo
esc_descripcion	text	Descripción del cultivo
esc_riego_metodo	integer	Método de riego
esc_riego_hora	time without time zone	Hora del riego
esc_siembra_fecha	date	Fecha de siembra
esc_estado	integer	Estado del cultivo
esc_tipo_suelo	integer	Tipo de suelo
esc_nivel_agotamiento_permisible	double precision	Nivel agotamiento permisible
esc_riego_metodo_eficiencia	integer	Método eficiencia de riego
esc_nodo_predictor	bigint	Nodo predictor

TABLA: ESPACIO_CULTIVO		
Descripción: Información propia del espacio y su cultivo		
Nombre	Tipo de Dato	Descripción
esc_codigo	Bigserial	Llave primaria de identificación única para cada espacio en el cultivo
esc_espacio	bigint	Llave Foránea a la tabla espacio para saber a qué espacio pertenece.

esc_cultivo	bigint	Llave Foránea a la tabla cultivo para saber a qué cultivo pertenece.
esc_nombre	Character varying	Nombre del cultivo
esc_descripcion	text	Descripción del cultivo
esc_riego_metodo	integer	Método de riego
esc_riego_hora	time without time zone	Hora del riego
esc_siembra_fecha	date	Fecha de siembra
esc_estado	integer	Estado del cultivo
esc_tipo_suelo	integer	Tipo de suelo
esc_nivel_agotamiento_permisible	double precision	Nivel agotamiento permisible
esc_riego_metodo_eficiencia	integer	Método eficiencia de riego
esc_nodo_predictor	bigint	Nodo predictor

TABLA: ESPACIO_CULTIVO_EVENTO		
Descripción: Información propia del espacio, cultivo y los eventos programados en él.		
Nombre	Tipo de Dato	Descripción
ece_codigo	Bigserial	Llave primaria de indentificacion unica para cada espacio en el cultivo y su respectivo evento
ece_espacio_cultivo	bigint	Llave Foránea a la tabla espacio para saber a qué espacio pertenece.
ece_fecha_hora	timestamp without time zone	Fecha para el evento
ece_tipo_evento	integer	Tipo de evento
ece_cantidad	double precision	cantidad
ece_unidad	integer	Llave Foránea a la tabla unidad para saber a qué cultivo pertenece.
ece_comentario	text	comentario
ece_estado	integer	estado

TABLA: ESPACIO_DISPOSITIVO		
Descripción: Información propia del espacio y el dispositivo		
Nombre	Tipo de Dato	Descripción
esd_codigo	Bigserial	Llave primaria de indentificacion unica para cada espacio en el cultivo y sus dispositivos
esd_espacio	bigint	Llave Foránea a la tabla espacio para saber a qué espacio pertenece.
esd_dispositivo	bigint	Llave Foránea a la tabla dispositivo para saber a que espacio pertenece.

esd_estado	integer	Estado
esd_pin	integer	Pin
esd_tipo_medicion	integer	Tipo de medicion
esd_descripcion	text	Descripción
esd_nombre	character varying	nombre
esd_acronimo	character varying	Acrónimo

TABLA: ESPACIO_MEDICION		
Descripción: Información propia del espacio y sus mediciones		
Nombre	Tipo de Dato	Descripción
esm_codigo	Bigserial	Llave primaria de indentificacion unica para cada espacio y las mediciones de sus dispositivos.
esm_espacio	bigint	Llave Foránea a la tabla espacio para saber a qué espacio pertenece.
esm_fecha_hora	timestamp without time zone	Fecha y hora
esm_estado	integer	Estado
esm_periodicidad	integer	Llave Foránea a la tabla tipo_periodicidad para saber a que espacio pertenece.

TABLA: ESPACIO_MEDICION_ITEM		
Descripción: Información propia del espacio y sus mediciones e ítems.		
Nombre	Tipo de Dato	Descripción
emi_codigo	Bigserial	Llave primaria de indentificacion unica para cada espacio y las mediciones de sus dispositivos.
emi_medicion	bigint	Llave Foránea a la tabla espacio_medicion para saber a qué espacio pertenece.
emi_dispositivo_item	bigint	Llave Foránea a la tabla dispositivo_item .
emi_valor	character varying	valor
emi_estado	integer	estado

TABLA: GRUPO		
Descripción: Información propia de los grupos		
Nombre	Tipo de Dato	Descripción
gru_codigo	Bigserial	Llave primaria de indentificacion unica para cada grupo
gru_nombre	bigint	Nombre.
gru_id	bigint	id

gru_estado	character varying	estado
------------	-------------------	--------

TABLA: INTERVALO_HUMEDAD_DISPONIBLE

Descripción: Información propia de la humedad disponible.

Nombre	Tipo de Dato	Descripción
ihd_codigo	serial	Llave primaria de indentificacion unica para cada grupo
ihd_tipo_suelo	integer	Nombre.
ihd_humedad_min	double precision	id
ihd_humedad_max	double precision	estado

TABLA: MEDICION

Descripción: Información propia de las mediciones.

Nombre	Tipo de Dato	Descripción
med_codigo	bigserial	Llave primaria de indentificacion unica para cada medicion.
med_nodo	bigint	Llave Foránea a la tabla nodo para saber a qué espacio pertenece.
med_fecha_hora	timestamp without time zone	fecha
med_estado	integer	estado
med_escala	integer	Llave Foránea a la tabla tipo_escala para saber a qué espacio pertenece.

TABLA: MEDICION_ITEM

Descripción: Información propia de las mediciones y los ítems

Nombre	Tipo de Dato	Descripción
mei_codigo	bigserial	Llave primaria de indentificacion unica para cada medicion.
mei_medicion	bigint	Llave Foránea a la tabla medicion para saber a que medicion pertenece.
mei_nodo_dispositivo	bigint	Llave Foránea a la tabla nodo_dispositivo para saber a qué dispositivo pertenece.
mei_estado	integer	estado
mei_valor	Character varying	Valor

TABLA: MODELO		
Descripción: Información propia de las modelos		
Nombre	Tipo de Dato	Descripción
mod_codigo	bigserial	Llave primaria de indentificacion unica para cada medicion.
mod_tipo	integer	Tipo de modelo
mod_nodo	bigint	Llave Foránea a la tabla nodo para saber a que dispositivo pertenece.
mod_dispositivo_item	bigint	Llave Foránea a la tabla dispositivo_item para saber a que medicion pertenece.
mod_nombre	Character varying	Nombre del modelo
mod_descripcion	text	descripción
mod_archivo	Character varying	ubicación
mod_algoritmo	Character varying	algoritmo
mod_numero_datos	integer	Numero datos
mod_comentario	text	comentario
mod_fecha_hora	timestamp without time zone	Fecha y hora
mod_estado	Integer	estado

TABLA: MODELO_MEDICION		
Descripción: Información propia de Los modelos y sus mediciones		
Nombre	Tipo de Dato	Descripción
mom_codigo	bigserial	Llave primaria de indentificacion unica para cada medicion de los modelo.
mom_modelo	bigint	Llave Foránea a la tabla modelo para saber a qué medicion pertenece.
mom_cantidad	double precision	cantidad
mom_tiempo	Double precision	tiempo
mom_comentario	text	comentario
mom_fecha_hora	timestamp without time zone	Fecha y hora
mom_estado	integer	estado

TABLA: MODELO_MIGRACION		
Descripción: Información propia de Los modelos de migración		
Nombre	Tipo de Dato	Descripción
mom_codigo	bigserial	Llave primaria de indentificacion unica para cada modelo.
mom_nombre	Character varying	nombre
mom_estado	integer	estado

TABLA: MODELO_MIGRACION_ITEM		
Descripción: Información propia de Los modelos de migración y los ítems		
Nombre	Tipo de Dato	Descripción
mmi_codigo	bigserial	Llave primaria de indentificacion unica para cada modelo.
mmi_modelo	bigint	Llave Foránea a la tabla modelo_migracion para saber a qué medicion pertenece.
mmi_columna	Integer	Columna
mmi_dispositivo_item	bigint	Llave Foránea a la tabla dispositivo_item para saber a qué medicion pertenece.
mmi_estado	integer	estado

TABLA: MUNICIPIO		
Descripción: Información propia de Los municipios		
Nombre	Tipo de Dato	Descripción
mun_codigo	bigserial	Llave primaria de indentificacion unica para cada municipio.
mun_departamento	Integer	Llave Foránea a la tabla departamento para saber a qué medicion pertenece.
mun_nombre	character varying	nombre
mun_acronimo	character varying	acrónimo

TABLA: NODO		
Descripción: Información propia de Los nodo		
Nombre	Tipo de Dato	Descripción
nod_codigo	bigserial	Llave primaria de indentificacion unica para cada nodo.
nod_tipo	Integer	Llave Foránea a la tabla tipo_nodo para saber a qué medicion pertenece.
nod_nombre	character varying	nombre
nod_descripcion	text	descripción

nod_estado	integer	Estado nodo
nod_entidad	integer	entidad
nod_direccion	character varying	dirección
nod_altitud	character varying	altitud
nod_latitud	character varying	latitud
nod_longitud	character varying	longitud
nod_municipio	bigint	municipio
nod_acronimo	character varying	acrónimo

TABLA: NODO_DISPOSITIVO		
Descripción: Información propia de Los nodos y dispositivos		
Nombre	Tipo de Dato	Descripción
nod_codigo	bigserial	Llave primaria de indentificacion unica para cada nodo con sus dispositivos.
nod_nodo	bigint	Llave Foránea a la tabla nodo para saber a que medicion pertenece.
nod_dispositivo	bigint	Llave Foránea a la tabla dispositivo para saber a que medicion pertenece.
nod_estado	integer	estado
nod_pin	integer	pin
nod_tipo_medicion	integer	Tipo medicion
nod_descripcion	text	descripción
nod_nombre	character varying	nombre
nod_acronimo	character varying	acrónimo

TABLA: NODO_PIN		
Descripción: Información propia de Los nodos y sus pines		
Nombre	Tipo de Dato	Descripción
nop_codigo	bigserial	Llave primaria de indentificacion unica para cada nodo con sus pines
nop_tipo_pin	integer	Tipo pin
nop_pin	character varying	pin
nop_descripcion	text	descripción
nop_estado	integer	estado
nop_nodo	integer	nodo

TABLA: PAIS		
Descripción: Información propia de Los Países		

Nombre	Tipo de Dato	Descripción
pai_codigo	bigserial	Llave primaria de indentificacion unica para cada país.
pai_nombre	integer	nombre
pai_acronimo	character varying	acrónimo

TABLA: RIEGO_CALENDARIO		
Descripción: Información propia de Los calendarios de riego		
Nombre	Tipo de Dato	Descripción
ric_codigo	bigserial	Llave primaria de indentificacion unica para cada calendario de riego
ric_espacio_cultivo	bigint	Llave Foránea a la tabla espacio_cultivo para saber a qué espacio del cultivo pertenece.
ric_fecha_hora	timestamp without time zone	Fecha y hora
ric_cantidad	double precision	cantidad
ric_unidad	integer	Llave Foránea a la tabla unidad para saber a qué unidad pertenece.
ric_estado	integer	estado

TABLA: RIEGO_EVENTO		
Descripción: Información propia de Los eventos de riego		
Nombre	Tipo de Dato	Descripción
rie_codigo	bigserial	Llave primaria de indentificacion unica para cada eventos de riego
rie_espacio_cultivo	bigint	Llave Foránea a la tabla espacio_cultivo para saber a qué espacio del cultivo pertenece.
rie_tipo_evento	integer	Llave Foránea a la tabla tipo_evento para saber a qué espacio del cultivo pertenece.
rie_fecha_hora	timestamp without time zone	Fecha y hora
rie_cantidad	double precision	cantidad
rie_unidad	integer	Llave Foránea a la tabla unidad para saber a qué unidad pertenece.
rie_estado	integer	estado
rie_comentario	text	comentario

TABLA: RIEGO_METODO		
---------------------	--	--

Descripción: Información propia de los métodos de riego

Nombre	Tipo de Dato	Descripción
rim_codigo	serial	Llave primaria de indentificacion unica para cada métodos de riego
rim_nombre	character varying	nombre
rim_descripcion	text	descripción
rim_estado	integer	estado

TABLA: RIEGO_METODO_EFICIENCIA

Descripción: Información propia de La eficiencia de los métodos de riego

Nombre	Tipo de Dato	Descripción
rim_codigo	serial	Llave primaria de indentificacion unica para cada evento de riego
rim_nombre	character varying	nombre
rim_descripcion	text	descripción
rim_estado	integer	estado

TABLA: RIEGO_PROGRAMACION

Descripción: Información propia de La programación de los riegos

Nombre	Tipo de Dato	Descripción
rip_codigo	bigserial	Llave primaria de indentificacion unica para cada evento de riego
rip_espacio_cultivo	bigint	Llave Foránea a la tabla espacio_cultivo para saber a que unidad pertenece.
rip_fecha_hora_inicio	timestamp without time zone	Fecha y hora fin
rip_fecha_hora_fin	timestamp without time zone	Fecha y hora inicio
rip_cantidad	double precision	cantidad
rip_duracion	integer	duración
rip_unidad	integer	Llave Foránea a la tabla unidad para saber a qué unidad pertenece.
rip_estado	integer	estado

TABLA: SUELO_CARACTERISTICAS_HUMEDAD

Descripción: Información propia de las características del suelo. (Cuadro 19 Pág. 144 FAO56)

Nombre	Tipo de Dato	Descripción
sch_codigo	serial	Llave primaria de indentificacion unica para cada evento

sch_tipo_suelo	integer	Tipo suelo
sch_fc_min	double precision	Características propias del suelo
sch_fc_max	double precision	Características propias del suelo
sch_wp_min	double precision	Características propias del suelo
sch_wp_max	double precision	Características propias del suelo
sch_fc_wp_min	double precision	Características propias del suelo
sch_fc_wp_max	double precision	Características propias del suelo
sch_afe_min	integer	Características propias del suelo
sch_afe_max	integer	Características propias del suelo
sch_aet_min	integer	Características propias del suelo
sch_aet_max	integer	Características propias del suelo
sch_estado	integer	estado
sch_descripcion	text	Descripción

Las tablas tipo_pin, tipo_nodo, tipo_periodicidad, tipo_bloque, tipo_bloque, tipo_cultivo, tipo_dispositivo, tipo_entidad, tipo_escalas, tipo_espacio, tipo_medicion, tipo_evento, tipo_estado tienen la misma estructura y sirven para la parametrización del sistema.

TABLA: TIPO_SUELO		
Descripción: Información propia de tipos de suelo		
Nombre	Tipo de Dato	Descripción
tis_codigo	bigserial	Llave primaria de identificación única para cada tipos de suelo
tis_nombre	character varying	Nombre de tipos de suelo
tis_descripcion	text	Descripción
tis_estado	integer	estado
tis_acronimo	character varying	Acrónimo

TABLA: TIPO_VALOR		
Descripción: Información propia de tipos de valor		
Nombre	Tipo de Dato	Descripción
tiv_codigo	bigserial	Llave primaria de identificación única para cada tipos de valor
tiv_nombre	character varying	Nombre de tipos de valor
tiv_rango_inicio	double precision	Rango inicio
tiv_rango_fin	double precision	Rango fin

tiv_unidad	integer	Llave Foránea a la tabla medida para saber a qué unidad pertenece.
------------	---------	---

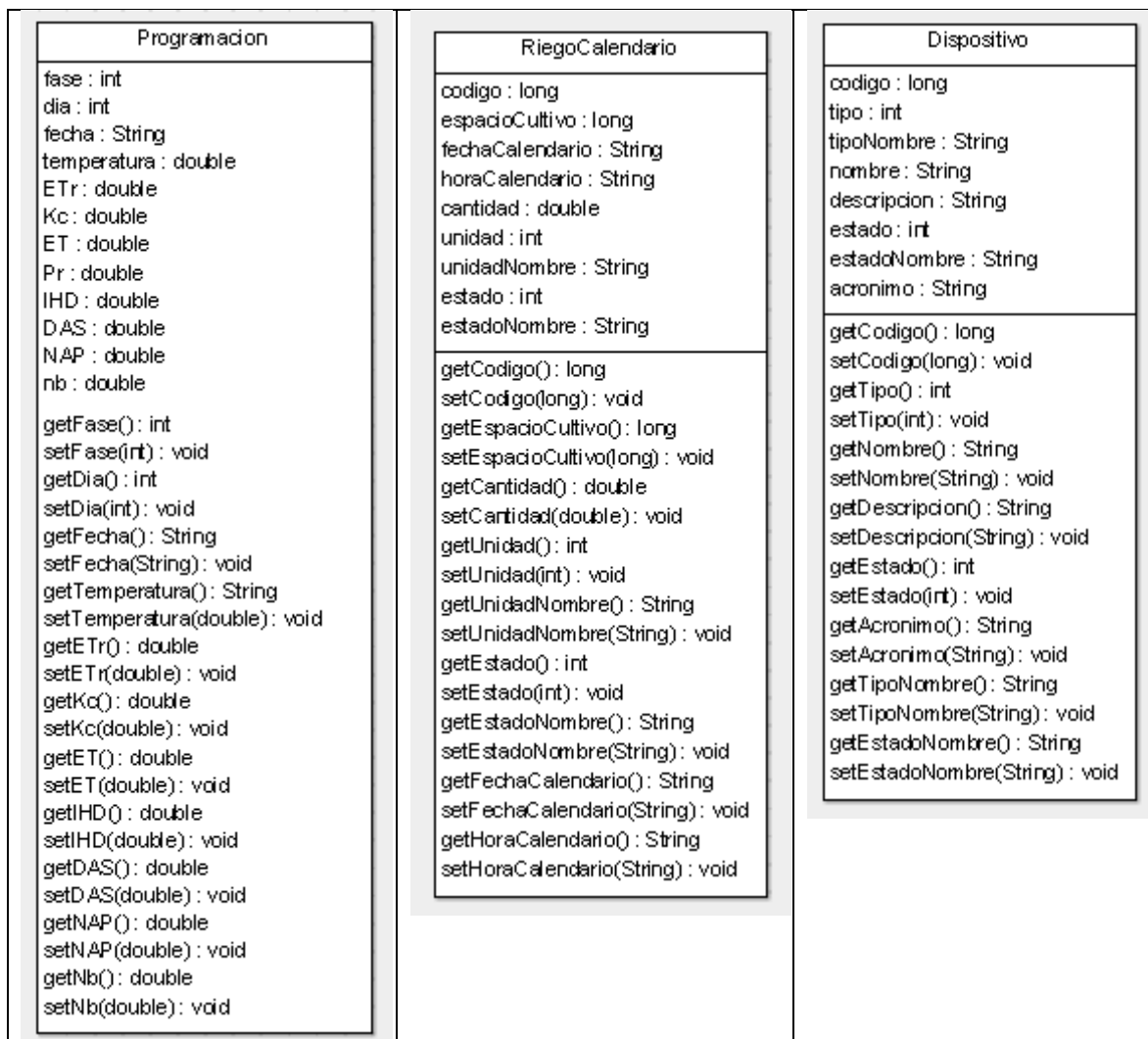
TABLA: UNIDAD		
Descripción: Información propia de las unidades		
Nombre	Tipo de Dato	Descripción
uni_codigo	serial	Llave primaria de identificación única para cada tipo de unidad
uni_nombre	character varying	Nombre de tipos de unidad
uni_acronimo	double precision	acrónimo
uni_descripcion	text	descripción

TABLA: USUARIO		
Descripción: Información propia de usuarios		
Nombre	Tipo de Dato	Descripción
usu_codigo	bigserial	Llave primaria de identificación única para cada usuario
usu_nombre	character varying	Nombre de usuario
usu_login	character varying	login
usu_password	character varying	password
usu_estado	integer	estado

TABLA: USUARIO_GRUPO		
Descripción: Información propia de usuarios y los grupos a los que pertenece		
Nombre	Tipo de Dato	Descripción
usg_codigo	bigserial	Llave primaria de identificación única para cada usuario y los grupos a los que pertenece
usg_usuario	bigint	Llave Foránea a la tabla usuario para saber a qué unidad pertenece.
usg_grupo	bigint	Llave Foránea a la tabla grupo para saber a qué unidad pertenece.
usg_estado	integer	Estado del grupo
usg_entidad	bigint	A quien pertenece

TABLA: WEB_PARAMETROS		
Descripción: Información propia de los parámetro web		
Nombre	Tipo de Dato	Descripción
wep_codigo	serial	Llave primaria de identificación única para cada uno de los parámetro web
wep_id	character varying	id
wep_nombre	character varying	Nombre del parámetro web
wep_descripcion	text	Descripción del parámetro web
wep_valor	character varying	Valor del parámetro web
wep_estado	integer	estado

ANEXO 3 – DIAGRAMA DE CLASES



Bloque	Entidad	Espacio
codigo : long entidad : long entidadNombre : String tipo : int tipoNombre : String nombre : String descripcion : String estado : int estadoNombre : String pisos : int	codigo : long tipo : int tipoNombre : String nombre : String descripcion : String estado : int estadoNombre : String	codigo : long bloque : long bloqueNombre : String nombre : String descripcion : String estado : int estadoNombre : String piso : int altitud : String latitud : String longitud : String area : double
getCodigo() : long setCodigo(long) : void getTipo() : int setTipo(int) : void getNombre() : String setNombre(String) : void getDescripcion() : String setDescripcion(String) : void getEstado() : int setEstado(int) : void getEntidad() : long setEntidad(long) : void getPisos() : int setPisos(int) : void getTipoNombre() : String setTipoNombre(String) : void getEstadoNombre() : String setEstadoNombre(String) : void getEntidadNombre() : String setEntidadNombre(String) : void	getCodigo() : long setCodigo(long) : void getTipo() : int setTipo(int) : void getNombre() : String setNombre(String) : void getDescripcion() : String setDescripcion(String) : void getEstado() : int setEstado(int) : void getTipoNombre() : String setTipoNombre(String) : void getEstadoNombre() : String setEstadoNombre(String) : void	getCodigo() : long setCodigo(long) : void getBloque() : long setBloque(long) : void getNombre() : String setNombre(String) : void getDescripcion() : String setDescripcion(String) : void getEstado() : int setEstado(int) : void getPiso() : int setPiso(int) : void getAltitud() : String setAltitud(String) : void getLatitud() : String setLatitud(String) : void getLongitud() : String setLongitud(String) : void getArea() : double setArea(double) : void getBloqueNombre() : String setBloqueNombre(String) : void getEstadoNombre() : String setEstadoNombre(String) : void

EspacioCultivo	EspacioDispositivo	RiegoEvento
<p>codigo : long espacio : long espacioNombre : String cultivo : long nombre : String descripcion : String riegoMetodo : int riegoMetodoNombre : String riegoHora : time siembraFecha : date estado : int estadoNombre : String tipoSuelo : int tipoSueloNombre : String nivelAgotamientoPermisible : double riegoMetodoEficiencia : int nodoPredictor : long nodoPredictorNombre : String</p> <hr/> <p>getCodigo() : long setCodigo(long) : void getEspacio() : long setEspacio(long) : void getCultivo() : long setCultivo(long) : void getNombre() : String setNombre(String) : void getDescripcion() : String setDescripcion(String) : void getRiegoMetodo() : int setRiegoMetodo(int) : void getRiegoHora() : time setRiegoHora(time) : void getSiembraFecha() : date setSiembraFecha(date) : void getEstado() : int setEstado(int) : void getTipoSuelo() : int setTipoSuelo(int) : void getNivelAgotamientoPermisible() : double setNivelAgotamientoPermisible(double) : void getRiegoMetodoEficiencia() : int setRiegoMetodoEficiencia(int) : void getEspacioNombre() : String setEspacioNombre(String) : void getCultivoNombre() : String setCultivoNombre(String) : void getRiegoMetodoNombre() : String setRiegoMetodoNombre(String) : void getTipoSueloNombre() : String setTipoSueloNombre(String) : void getEstadoNombre() : String setEstadoNombre(String) : void getNodoPredictor() : long setNodoPredictor(long) : void getNodoPredictorNombre() : String setNodoPredictorNombre(String) : void</p>	<p>codigo : long espacio : long espacioNombre : String dispositivo : long dispositivoNombre : String estado : int estadoNombre : String nombre : String descripcion : String acronimo : String</p> <hr/> <p>getCodigo() : long setCodigo(long) : void getEspacio() : long setEspacio(Long) : void getDispositivo() : long setDispositivo(long) : void getEstado() : int setEstado(int) : void getNombre() : String setNombre(String) : void getDescripcion() : String setDescripcion(String) : void getEspacioNombre() : String setEspacioNombre(String) : void getDispositivoNombre() : String setDispositivoNombre(String) : void getEstadoNombre() : String setEstadoNombre(String) : void getAcronimo() : String setAcronimo(String) : void</p>	<p>codigo : long espacioCultivo : long espacioCultivoNombre : String fechaHora : String tipoEvento : int tipoEventoNombre : String fecha : String hora : String cantidad : double unidad : double unidadNombre : String estado : int estadoNombre : String comentario : String</p> <hr/> <p>getCodigo() : long setCodigo(long) : void getEspacioCultivo() : long setEspacioCultivo(long) : void getCantidad() : double setCantidad(double) : void getUnidad() : int setUnidad(int) : void getUnidadNombre() : String setUnidadNombre(String) : void getEstado() : int setEstado(int) : void getEstadoNombre() : String setEstadoNombre(String) : void getFecha() : String setFecha(String) : void getHora() : String setHora(String) : void getTipoEvento() : int setTipoEvento(int) : void getTipoEventoNombre() : String setTipoEventoNombre(String) : void getComentario() : String setComentario(String) : void getEspacioCultivoNombre() : String setEspacioCultivoNombre(String) : void getFechaHora() : String setFechaHora(String) : void</p>

Programacion	ProgramacionRiego
fase : int dia : int fecha : String temperatura : double ETr : double Kc : double ET : double Pr : double IHD : double DAS : double NAP : double nb : double	municipio : int cultivo : int fechalnicio : Calendar riegoMetodo : int riegoMetodoEficiencia : int tipoSuelo : int nivelAgotamientoPermisible : double nodoPredictor : long
getFase() : int setFase(int) : void getDia() : int setDia(int) : void getFecha() : String setFecha(String) : void getTemperatura() : String setTemperatura(double) : void getETr() : double setETr(double) : void getKc() : double setKc(double) : void getET() : double setET(double) : void getIHD() : double setIHD(double) : void getDAS() : double setDAS(double) : void getNAP() : double setNAP(double) : void getNb() : double setNb(double) : void	getCalendarFromString(String) : Calendar getMunicipio() : int setMunicipio(int) : void getCultivo() : int setCultivo(int) : void getFechalnicio() : Calendar setFechalnicio(Calendar) : void setFechalnicio(String) : void getRiegoMetodoEficiencia() : int setRiegoMetodoEficiencia(int) : void getTipoSuelo() : int setTipoSuelo(int) : void getNivelAgotamientoPermisible() : double setNivelAgotamientoPermisible(double) : void getRiegoMetodo() : int setRiegoMetodo(int) : void getNodoPredictor() : long setNodoPredictor(long) : void main(String[]) : void getCalendarioRiego() : ArrayList getPrediccionREST(long,String,String,int,int) : ArrayList