

Grado en Ingeniería de Tecnologías de Telecomunicación  
2018-2019

*Trabajo Fin de Grado*

Diseño de implementación de un sistema para monitorización de cultivos y equipos agrícolas

Clemente Rodríguez Arráez

Tutora

Florina Almenares Mendoza

(Lugar y fecha de presentación prevista)

Esta obra se encuentra sujeta a la licencia Creative Commons



**Reconocimiento – No Comercial – Sin Obra Derivada**

<Página en blanco>

<Página del tribunal>

<Página en blanco>

<Agradecimientos>

<Página en blanco>

<Resumen>

<Página en blanco>

<Abstract>

<Indice general>

<Índice de figuras>

<Índice de tablas>

<Capítulo 1. Introducción y Objetivos>

Resumen

El fin de este proyecto es el diseño de un sistema utilizando la tecnología actual que haga uso del concepto del internet de las cosas y la red de sensores. Se ha decidido enfocar este desarrollo en la extracción de datos medioambientales. Particularmente, en el diseño de implementación de un sistema para monitorización de cultivos (en especial de la vid) y equipos agrícolas.

Los objetivos principales de este proyecto son la mejora del riego de las plantas, automatizando esta tarea para optimizar el gasto de agua que se extrae de los acuíferos, y la seguridad de las máquinas agrícolas valiosas tales como los motores de gasoil que generalmente se encuentran desprotegidos en el exterior.

Las tareas principales del sistema son la recopilación de los datos de los sensores, el análisis de estos, el registro de todas las actividades que realice el sistema y el envío de alertas al usuario. Los datos se podrán acceder de tres maneras: en una página web, recibiendo las alertas vía SMS y correo electrónico y utilizando una conexión directa con el sistema utilizando la terminal.

El sistema se compone de varios elementos principales: un coordinador que realice todas las tareas instalado en una Raspberry Pi, dos aplicaciones destinadas al riego y a la seguridad, un sensor de humedad de suelo, un sensor de movimiento, un giroscopio y, además, se ha utilizado un módem que envía los datos en cualquier lugar del mundo utilizando la tecnología GSM.

**Palabras clave**: internet de las cosas, redes de sensores, GSM, monitorización, análisis de datos, seguridad agrícola, riego automático, Hologram.

Abstract

The goal of this project is to design a system by using the current technology in order to apply the concept of Internet of things and Wireless Sensor Networks. The extraction of environmental data is the main point of this development, especially, a design of a system implementation for farming (vineyards in particular) and agricultural equipment monitoring.

The main objectives in this capstone project are the plants’ irrigation improvement, by making this task automatic to optimize the use of water from aquifers, and the agricultural valuable machines security improvement such as diesel engines that are unprotected in farmers lands.

The main tasks of the system are the collection of data from the sensors, their analysis, the recording of all the activities carried out by the system and the sending of alerts to the user. The data can be accessed in three ways: on a web page, receiving alerts via SMS and email and using a direct connection to the system by using the terminal.

The system consists of several elements: a coordinator who performs all the tasks installed in a Raspberry Pi, two applications for irrigation and safety, a soil moisture sensor, a motion sensor, a gyroscope and, in addition, a modem that sends data from anywhere in the world by using GSM technology.

**Keywords**: Internet of Things, Wireless Sensor Networks, GSM, monitoring, data analysis, agricultural security, automatic irrigation, Hologram.

Índice general

1. [**Introducción y Objetivos**](#Introducción_y_Objetivos)
   1. [Contexto](#Contexto)
   2. [Problema](#Problema)
   3. [Objetivos](#Objetivos)
      1. [Objetivos específicos](#Objetivos_Específicos)
   4. [Medios utilizados](#Medios_Utilizados)
      1. [Software](#Medios_Software)
      2. [Hardware](#Medios_Hardware)
   5. [Entorno socioeconómico](#Entorno_Socioeconomico)
   6. [Marco regulador](#Marco_Regulador)
   7. [Estructura de la memoria](#Estructura_Memoria)
2. **Estado del arte**
   1. Internet de las Cosas
      1. Evolución de las tecnologías con el Internet de las Cosas
      2. Aplicaciones del Internet de las Cosas
   2. Red de sensores inalámbricas
      1. Usabilidad de las Redes de sensores inalámbricas
      2. Arquitectura de la red de sensores en la agricultura
      3. Arquitectura de los nodos sensores
   3. Tecnología y estándares usados en la agricultura
      1. Zigbee
      2. WiFi
      3. GSM y su evolución
   4. Herramientas utilizadas
      1. Python
      2. RStudio
      3. Raspbian OS
      4. Raspberry
      5. Sensores
      6. Comunicación por Sockets? Debería estar aquí creo
3. **Descripción del sistema**
   1. Requisitos del sistema
      1. Requisitos funcionales
      2. Requisitos no funcionales
      3. Casos de uso
         1. Actores
         2. Diagramas de los casos de uso
         3. Descripción de los casos de uso
   2. Arquitectura del sistema
4. **Implementación**
   1. Contexto
   2. Programa de riego – riego.py
   3. Programa de seguridad – seguridad.py
      1. Sistema de activación y desactivación del programa
      2. Detección de cuerpos térmicos en movimiento
      3. Análisis del movimiento angular y lineal del motor
      4. Localización del dispositivo
      5. Envío de alertas al usuario
      6. Inicio de parámetros y librerías
      7. Creación de archivos “log.txt” y “alertas.txt”
   4. Terminal
      1. Comandos
   5. Análisis del consumo de datos en la nube
      1. Informes de uso
      2. Uso en directo
   6. La unión de todas las piezas
   7. Simulación (WiFi)
   8. Caso real
5. **Planificación y presupuesto**
   1. Fases del desarrollo
6. **Conclusiones y líneas futuras**

**Referencias**

**glosario\* (Punto 1)**

Índice de figuras

1.1. Ejemplo de motor de gasoil – DEUTZ 33 kVA

1.2. Ejemplo de panel solar destinado al riego

1.3. Raspberry Pi 3 Model B

1.4. Higrómetro FC-28

1.5. Vista con detalle de la placa de medición estándar

1.6. Ejemplo del ADC MCP3008

1.7. Esquema de Raspberry Pi, el MCP3008 y el FC-28

1.8. Sensor PIR HC-SR501

1.9. Esquema de detección del sensor HC-SR501

1.10. Esquema de detección de movimiento

1.11. Vista general del giroscopio y acelerómetro MPU6050

1.12. Ejemplo tarjeta SIM Hologram

1.13. Módem - Hologram Nova

1.14. Comando de instalación

2.1. Internet de las Cosas

2.2. Número de dispositivos conectados a Internet en el mundo

2.3. Número de dispositivos conectados usados por persona en España

2.4. Aplicaciones del IoT en la agricultura

2.5. Uso del dron para monitorización de una viña en Rosario (Argentina)

2.6. Mejora de la domótica con el IoT

2.7. Ejemplo del uso de sensores para conducción autónoma

2.8. Internet de las Cosas Médicas

2.9. Internet Industrial de las Cosas

2.10. Ejemplo de red de sensores inalámbrica

2.11. Ejemplo de módulo de red Zigbee

2.12. Red de Sensores Zigbee

2.13. Ejemplo de triangulación por GSM para localizar un dispositivo

3.1. Esquema del escenario 1 – Interfaz Web

3.2. Esquema del escenario 2 - Terminal

3.3. Esquema del escenario 3 – Servidor

3.4. Arquitectura ideal del sistema

4.1. Diagrama de flujo – riego.py

Índice de tablas

2.1. Diferencias entre SiP y SoC . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .

3.1. Ejemplo de requisito del sistema

3.2. Requisito de sistema – Requisito funcional 01

3.3. Requisito de sistema – Requisito funcional 02

3.4. Requisito de sistema – Requisito funcional 03

3.5. Requisito de sistema – Requisito funcional 04

3.6. Requisito de sistema – Requisito funcional 05

3.7. Requisito de sistema – Requisito funcional 06

3.8. Requisito de sistema – Requisito funcional 07

3.9. Requisito de sistema – Requisito funcional 08

3.10. Requisito de sistema – Requisito funcional 09

3.11. Requisito de sistema – Requisito funcional 10

3.12. Requisito de sistema – Requisito no funcional 01

3.13. Requisito de sistema – Requisito no funcional 02

3.14. Requisito de sistema – Requisito no funcional 03

3.15. Requisito de sistema – Requisito no funcional 04

3.16. Requisito de sistema – Requisito no funcional 05

3.17. Requisito de sistema – Requisito no funcional 06

3.18. Requisito de sistema – Requisito no funcional 07

3.19. Requisito de sistema – Requisito no funcional 08

3.20. Requisito de sistema – Requisito no funcional 09

3.21. Requisito de sistema – Requisito no funcional 10

3.22. Requisito de sistema – Requisito no funcional 11

3.23. Requisito de sistema – Requisito no funcional 12

3.24. Requisito de sistema – Requisito no funcional 13

3.25. Requisito de sistema – Requisito no funcional 14

3.26: Ejemplo de caso de uso

3.27. Escenario 1 – Caso de uso 1

3.28. Escenario 1 – Caso de uso 2

3.29. Escenario 1 – Caso de uso 3

3.30. Escenario 1 – Caso de uso 4

3.31. Escenario 2 – Caso de uso 1

3.32. Escenario 2 – Caso de uso 2

3.33. Escenario 2 – Caso de uso 3

3.34. Escenario 2 – Caso de uso 4

3.35. Escenario 2 – Caso de uso 5

3.36. Escenario 2 – Caso de uso 6

3.37. Escenario 2 – Caso de uso 7

3.38. Escenario 2 – Caso de uso 8

3.39. Escenario 3 – Caso de uso 1

3.40. Escenario 3 – Caso de uso 2

3.41. Escenario 3 – Caso de uso 3

**Capítulo 1**

Introducción y Objetivos

El primer capítulo del presente documento tiene como objetivo introducir al lector en el entorno en el que se mueve este proyecto. Además, se presentarán los objetivos a realizar. Así pues, la primera sección explica el contexto donde se enmarca el trabajo. Acto seguido, se expondrán los problemas existentes en este ámbito para fijar, en la siguiente sección, los objetivos que se pretenden alcanzar con la realización de este proyecto. Finalmente, se presentará una estructura del documento indicando los capítulos que lo formarán junto con una descripción detallada del contenido en cada uno de ellos.

* 1. Contexto

Antes, hoy y mañana, el ser humano ha tenido la necesidad de construir y desarrollar nuevos objetos que faciliten su trabajo y mejoren su calidad de vida [<https://www.areatecnologia.com/historia-evolucion-tecnologia.htm>]. Entre los aspectos más notables en el área de las comunicaciones y la computación, destaca la aparición de los ordenadores personales y el acceso a internet. Estos dos elementos permiten que gran parte de empresas y organizaciones puedan intercambiar una gran cantidad de información en un tiempo muy corto. Además, la aparición de los dispositivos móviles inteligentes supuso un cambio muy drástico en la sociedad actual ya que la comunicación entre personas se ha facilitado radicalmente pasando de las cartas o las cabinas de teléfono a hacer llamadas desde un terminal de bolsillo. De hecho, otro paso para mejorar la efectividad de los trabajos se consigue gracias a la aparición del Internet de las Cosas, no solo teléfonos, sino miles de millones de dispositivos electrónicos (como televisiones, electrodomésticos, sensores…) están conectados a Internet [<https://www.juniperresearch.com/press/press-releases/iot-connections-to-grow-140-to-hit-50-billion>]. La posibilidad de compartir información de manera instantánea en cualquier parte del mundo elimina por completo las barreras que impedían el acceso a la comunicación y la información.

Desde la comunicación y la información, aparecen las Tecnologías de la Información y la Comunicación (o TIC) que son todos aquellos recursos, herramientas y programas que se utilizan para procesar, administrar y compartir la información mediante diversos soportes tecnológicos [<http://tutorial.cch.unam.mx/bloque4/lasTIC>]. Forman parte de la mayoría de sectores como la educación, robótica, el empleo, la salud o la alimentación [<http://noticias.iberestudios.com/%C2%BFque-son-las-tic-y-para-que-sirven/>]. Por ejemplo, en el punto de la alimentación, la agricultura y la ganadería están abiertas a este mundo de nuevas posibilidades ya que es necesario satisfacer la demanda de alimento de la población mundial que está en constante crecimiento. Muchas actividades como la siembra, la gestión del rebaño, la fertilización de las tierras o el mantenimiento de la máquina agrícola se pueden automatizar y aumentar la producción sin tener que malgastar grandes cantidades de recursos que son esenciales diariamente [S. Adamala, N.S. Raghuwanshi, A. Mishra [Development of surface irrigation systems design and evaluation software (SIDES)](https://www.sciencedirect.com/science/article/pii/S0168169913002743) Comput. Electron. Agric., 100 (2014), pp. 100-109].

* 1. Problema

Para continuar con el mundo de la agricultura y la ganadería, el siguiente proyecto se centrará en un municipio de Toledo, llamado Corral de Almaguer, a 100 kilómetros de Madrid. Su economía se basa principalmente en la agricultura y la ganadería. En cuanto a la agricultura, se fomenta mucho la recolección de las uvas para su procesamiento en vino y de las aceitunas en aceite de oliva.

No obstante, no todo es recoger el fruto de la planta. Se necesita trabajar la tierra y la planta todo el año para que al final de este, se tenga mucho fruto y calidad. Una de las cosas más importantes que necesitan las plantas y las personas es el agua. Es el principio básico para vivir y sin ella, el tiempo de supervivencia es menor. En el caso de las plantas, morirían y su fruto se perdería. Para ello, una solución que se está utilizando hoy en día, es la extracción de aguas subterráneas que se encuentran en acuíferos a diferentes niveles bajo tierra. Gracias a los motores de gasoil o las placas solares, se obtiene energía suficiente para que las bombas que se encuentran debajo de la superficie extraigan el agua necesaria para uso urbano y/o agrícola.

El agua subterránea de los acuíferos se extrae mediante unas bombas de extracción de diferentes potencias (en función de los metros en los que esté el acuífero) y la energía que se utiliza para hacer funcionar las bombas es o bien solar (gracias a una placa solar fija en el suelo), o bien por motores de gasoil que no están fijos. Alejándose del tema medioambiental, muchos agricultores todavía optan por el motor de gasoil ya que con un motor pueden ir a diferentes pozos y regar todo el tiempo que consideren ya que la placa solar está fija en el suelo y depende de la luz del sol para funcionar.

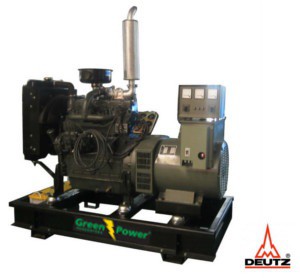


Figura 1.1: Ejemplo de motor de gasoil – DEUTZ 33 kVA [<https://www.greenpower.lk/katalog/artikel/greenpower-deutz-diesel-power-generator-30kva-24kw-open-frame-manual-starting/116586-0-50/>]



Figura 1.2: Ejemplo de panel solar destinado al riego [tandil.olx.com.ar]

Así pues, un aspecto que preocupa a los agricultores es la seguridad de su maquinaria agrícola, en especial de los motores de gasoil, debido a que en los últimos años se han producido robos totales de estas mismas [<https://www.eldigitaldealbacete.com/2017/05/18/desmantelado-grupo-criminal-castilla-la-mancha-especializado-robo-motores-riego/>, justificación 1]. Muchas de las veces las máquinas son irrecuperables ya que se han vuelto a vender de manera ilegal o se han desguazado para conseguir los metales de estas mismas [justificación 1]. Este es un problema grave ya que estos motores tienen un valor sustancial [justificación 1]y requiere de una inversión inicial por parte del agricultor para poder utilizarlos. ##Quizá se debería extender más. Poner algún dato de porcentaje de robos##·····

Para resolver estos problemas, que son comunes en la actualidad, es vital el uso de las Tecnologías de la Información y la Comunicación. El empleo del Internet de las Cosas y las redes de sensores inalámbricas ayudaría a automatizar estas tareas y a impulsar el desarrollo de las zonas rurales conectándolas con el futuro. Un futuro que se llama sociedad digital.

* 1. Objetivos

El objetivo general de este trabajo fin de grado consiste en diseñar e implementar un sistema de monitorización del riego de cultivos (especializado en la vid) y equipos agrícolas (motores de gasoil), utilizando sensores inalámbricos capaces de enviar información para que el sistema sea capaz de procesar y tomar decisiones a partir de la misma.

La monitorización y la toma de decisiones van orientadas a cumplir con dos metas fundamentales:

* Evitar gastar más agua de la que se necesita a partir del control del agua para un riego óptimo.
* Evitar el robo del motor de gasoil con un control de seguridad.
  + 1. Objetivos específicos

Para conseguir el objetivo general de este trabajo, se deben cumplir con los siguientes objetivos específicos:

* Desarrollar nodos que incorporen los sensores necesarios, denominados sensores, para cada situación particular, tanto de control de riego como de la seguridad.
* Incorporar, a dichos nodos, capacidad de comunicación tanto local como remota.
* Cifrar la información transmitida entre los sensores y la estación central usando un estándar que proporcione un nivel de seguridad alto.
* Diseñar el sistema para diferentes sectores de cultivo, determinando qué área (en metros cuadrados) abarca un sistema de sensores y cuántos de ellos se necesitarán para optimizar todo el cultivo.
  1. Medios utilizados
     1. Software
* **Python**: Es el lenguaje principal de programación utilizado en el proyecto gracias a su facilidad para conseguir información de los sensores y tratarla.
* **Raspbian**: Sistema operativo de la Raspberry Pi.
* **Microsoft Office 365 – Microsoft Word**: Utilizado para la documentación del proyecto.
* **Microsoft Office 365 – Microsoft Excel**: Utilizado para hacer el diagrama de Gantt.
* **MobaXterm**: Es la terminal utilizada con la que se pueden realizar conexiones *SSH* del ordenador a la Raspberry Pi.
* **RStudio**: El entorno de desarrollo en R que sirvió para el procesamiento de datos en el anexo 2.
* **R**: Lenguaje de programación útil para la manipulación de datos. Se utilizó en el anexo 2.
* **Hologram Cloud**: Lugar donde se gestionan los datos que se han utilizado en el módem conectado a la Raspberry Pi. Además, desde ahí se envían *SMS* y correos electrónicos de alerta al usuario.
* **Thingspeak**: Un sitio web donde se enviarán los datos de la humedad en tiempo real. Se podrá ver esta información gráficamente.
* **Draw.io**: Página web útil para realizar los diagramas de flujo y otros esquemas del docuemento.
  + 1. Hardware

En cuanto a *Hardware*, a continuación, se mostrarán aquellos dispositivos que han hecho posible el proyecto:

* **Raspberry Pi 3 Model B**: Como cualquier otro ordenador, la Raspberry Pi dispone de una serie de puertos: USB, HDMI, Ethernet, Audio Jack, tarjeta de memoria microSD, módulo para conectar una pantalla táctil, una cámara, Bluetooth y WiFi en su última versión. La placa monta un procesador ARM 64-bits de cuatro núcleos a 1.2 GHz y con un 1 GB de memoria RAM. Además, cuenta con 40 pines de GPIO destinados al uso de los sensores.

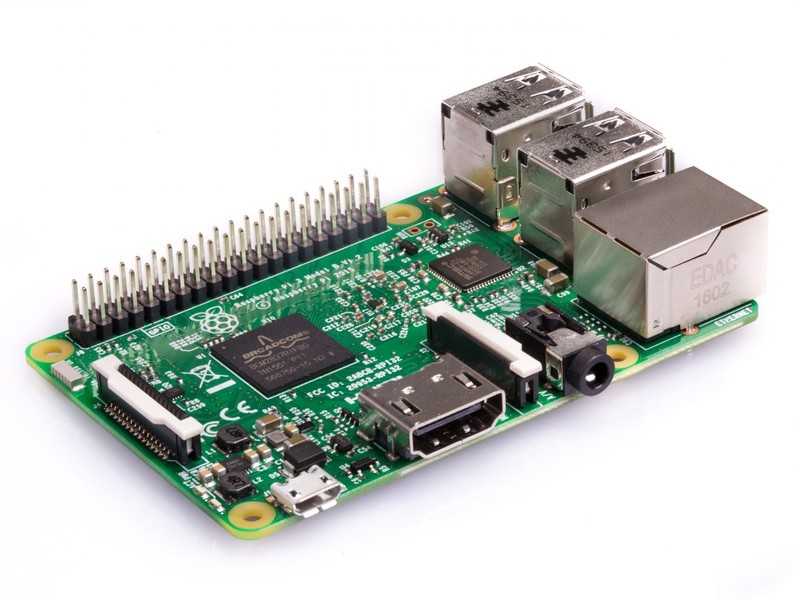


Figura 1.3: Raspberry Pi 3 Model B

* **Higrómetro FC-28**: Es un sensor que mide la humedad del suelo por la variación de su conductividad. Son ampliamente empleados en sistemas automáticos de riego para detectar cuando es necesario activar el sistema de bombeo.

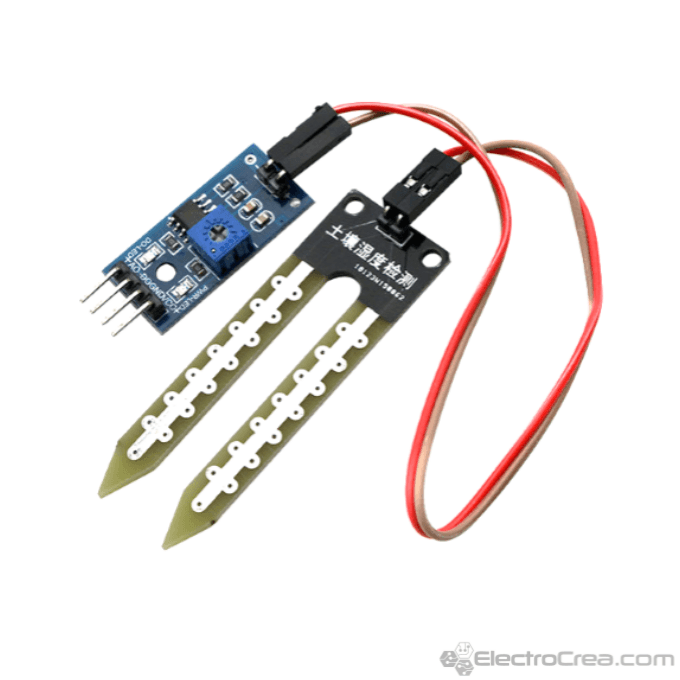


Figura 1.4: Higrómetro FC-28 [electrocrea.com]

El FC-28 viene acompañado con una placa de medición estándar que permite obtener la medición como valor analógico o como una salida digital, activada cuando la humedad supera un umbral. En el caso analógico, los valores obtenidos van desde 0 (sumergido en agua), a 1023 (en el aire o suelo muy seco). Sin embargo, en la salida digital, cuando la humedad supera un umbral que se ajusta con el potenciómetro de la figura X2, la salida se dispara. Se obtendrá una señal LOW si el suelo no está húmedo, y HIGH si el FC-28 supera el umbral [<https://www.luisllamas.es/arduino-humedad-suelo-fc-28/>].

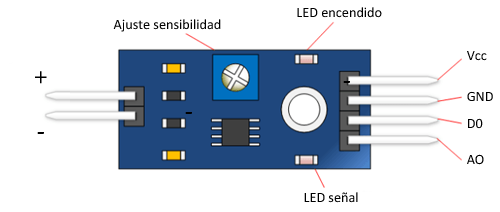


Figura 1.5: Vista con detalle de la placa de medición estándar [<https://www.luisllamas.es/arduino-humedad-suelo-fc-28/>]

* **MCP3008**: Es un convertidor analógico a digital de 10 bits diseñado por la empresa Microchip. Es muy útil ya que la Raspberry Pi no tiene pines de entrada analógicos. El ADC se colocaría entre la Raspberry PI y el FC-28 y permitiría que la Raspberry Pi pudiese leer los valores exactos del sensor de humedad FC-28.

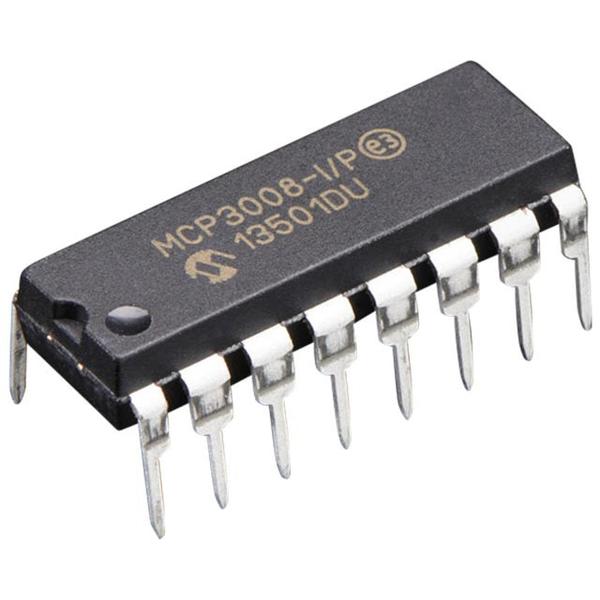


Figura 1.6: Ejemplo del ADC MCP3008 [<https://thepihut.com/products/adafruit-mcp3008-8-channel-10-bit-adc-with-spi-interface>]

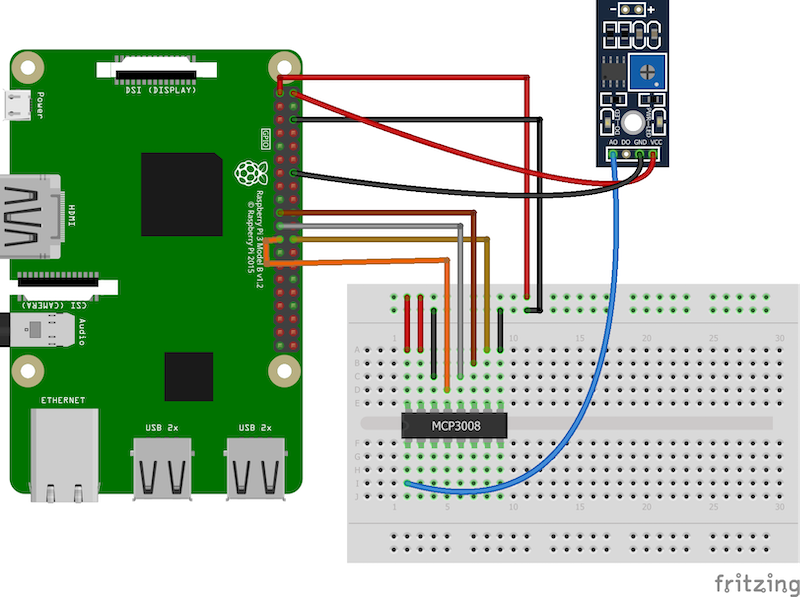


Figura 1.7: Esquema de Raspberry Pi, el MCP3008 y el FC-28 [<https://maker.pro/raspberry-pi/tutorial/interfacing-soil-moisture-sensor-with-raspberry-pi>]

* **HC-SR501**: Es un sensor infrarrojo pasivo o *PIR* que detecta variaciones de las radiaciones infrarrojas del medio ambiente que cubre como el calor del cuerpo humano o animales. Su sensor tiene una apertura de 110º y con un potenciómetro, se puede calibrar una distancia de detección de 3 a 7 metros.



Figura 1.8: Sensor *PIR* HC-SR501 [<https://www.rhydolabz.com/sensors-ir-pir-sensors-c-137_150/pir-motion-detection-sensor-hcsr501-p-1512.html?zenid=3ccvrqqaigdusb8qf17bgn6v93>]

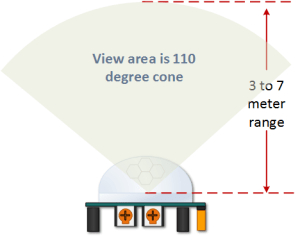


Figura 1.9: Esquema de detección del sensor HC-SR501

Como se puede ver en la figura anterior, el *PIR* tiene un segundo potenciómetro que sirve para calibrar cada cuándo se recogen datos del sensor.

Además, el sensor del *PIR* tiene dos ranuras con un material sensible a las radiaciones infrarrojas. Cuando ambas ranuras detectan la misma radiación, el sensor está en reposo, mientras que cuando las ranuras detectan diferente radicación, se detecta una variación de movimiento térmico.

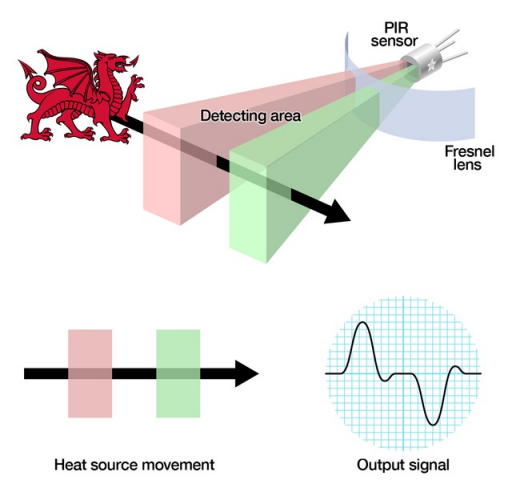


Figura 1.10: Esquema de detección de movimiento [<https://cdn-learn.adafruit.com/downloads/pdf/pir-passive-infrared-proximity-motion-sensor.pdf>]

* **MPU6050**: Es un dispositivo integrado de seguimiento de movimiento de 6 ejes que combina un giroscopio de 3 ejes y un acelerómetro de 3 ejes. Utiliza comunicación I2C, es decir, comunicación síncrona con cable para el reloj (SCL) y otro para los datos (SDA). El maestro sería la Raspberry Pi el cual crea la señal de reloj. Los ejes son X, Y se pueden ver en la figura “X” mientras que el eje Z es el ortogonal a los dos ejes anteriores.

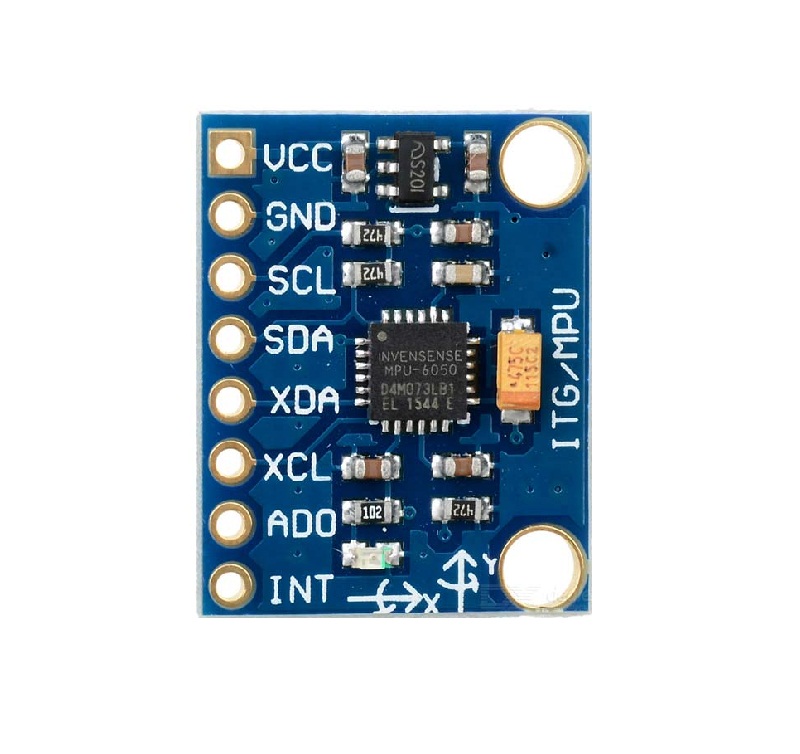


Figura 1.11: Vista general del giroscopio y acelerómetro MPU6050 [<https://www.makerlab-electronics.com/product/triple-axis-accelerometer-gyro-breakout-mpu6050/>]

* **Hologram SIM**: Es una tarjeta SIM *IoT* segura para una red celular global. Tiene fácil activación ya que no es necesario pedir permisos a las compañías de telefonía, sino que todo el proceso ya está gestionado por Hologram. De hecho, la tarjeta utiliza la cobertura del operador que más potencia suministre en el área donde se encuentre.

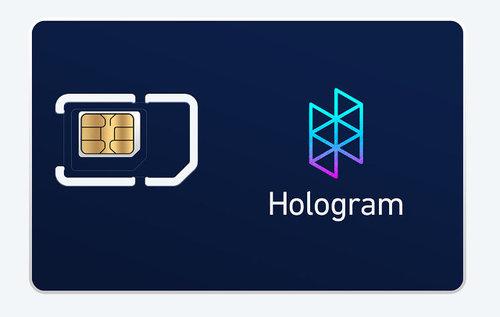


Figura 1.12: Ejemplo tarjeta SIM Hologram

* **Hologram Nova**: Es el módem recomendado por Hologram para poder tener conexión a la red en cualquier dispositivo. Ya que Raspberry Pi tiene varios USB, Hologram Nova es una opción sencilla de usar.



Figura 1.13: Módem - Hologram Nova

Para su funcionamiento, es necesario insertar la tarjeta SIM (en el formato nano SIM) en el módem y conectarlo directamente en la Raspberry Pi. Después, es necesario instalar el paquete Hologram en la Raspberry Pi con el siguiente comando:



Figura 1.14: Comando de instalación [<https://hologram.io/docs/reference/nova/user-manual/>]

Una vez hecho, Hologram Nova ya estará disponible para cualquier actividad en la Raspberry Pi.

* **Batería portátil**: Para poder mantener la Raspberry Pi encendida a larga distancia es necesario tener una batería portátil. En este proyecto, se ha utilizado una batería de 10000 mAh que da lugar a 12 horas de funcionamiento aproximadamente.
  1. Entorno socioeconómico

El objetivo principal del Internet de las Cosas es hacer del mundo un lugar más eficiente y, por consiguiente, conseguir un ahorro energético y económico. Potenciando esta tecnología, cada vez se incidirá más en su uso en el ámbito empresarial y en la modernización de otros sectores como el agrario ya que mediante el uso de sensores, se optimizan muchas tareas como el regado y cuidado de las plantas.

Además, es necesario tener un control entre los recursos del planeta junto con las condiciones materiales de vida de las personas. Por ejemplo, en la actualidad, el agua y su mantenimiento están siendo cada vez más los objetos principales de los debates en torno al desarrollo sostenible.

Sin embargo, muchos trabajadores cuestionan que esta evolución a la sensorización y la automatización suponga un problema en sus propios trabajos. Todo cambio tecnológico parece difícil de asumir, pero la población de hoy en día ha demostrado que es muy fácil adaptarse a las nuevas tecnologías. Algunos ejemplos se pueden encontrar en los *Smartphones* o en el sector automovilístico en los que se encuentran cada vez más funciones que facilitan una conducción más segura y cómoda.

* 1. Marco regulador

Con la expansión de las Tecnologías de Información y la economía digital, los ciudadanos han perdido capacidad para controlar sus datos personales y mantener su privacidad. En enero de 2012 la Comisión Europea comenzó un proceso de reforma de las normas comunitarias para la protección de los datos personales dentro de la Unión Europea [20].

El desarrollo del Internet de las Cosas podría verse frenado y entorpecido en la UE debido a la existente y futura regulación en materia de privacidad de datos. Por tanto, es importante tener en cuenta el marco legal tanto comunitario como específico de cada país a la hora de diseñar y desarrollar nuevos entornos inteligentes, respetando la protección de datos y la privacidad de los ciudadanos en todo momento.

* + 1. Normas y legislación aplicable al proyecto
    2. Licencias y tecnologías

A continuación, se citan las licencias de uso referentes a los programas y tecnologías empleados en el desarrollo del proyecto:

* **Microsoft Windows 10**: La licencia es de tipo Home, cedida gratuitamente por Microsoft en la transición de Windows 8 a 10.
* **Microsoft Office 365**: El paquete de ofimática empleado para el desarrollo del proyecto cuya licencia es de tipo estudiante, está cedida por Microsoft gracias a la colaboración de la Universidad Carlos III de Madrid.
* **MobaXTerm**: terminal compatible en Windows con herramientas de red para computación remota con licencia gratuita.
* **Raspbian OS**: El sistema operativo integrado en la Raspberry Pi con licencia *GNU*.
* **Python**: El lenguaje es administrado por Python Software Foundation, que tiene la licencia de código abierto llamado Python Software Foundation License.
* **RStudio**: El *IDE* para el lenguaje de programación R mantiene una licencia libre para uso no comercial.
* **Hologram Cloud**: La inspección del flujo de datos por el módem de la marca Hologram es gratuita ya que se asumen los costes de las tarifas de la tarjeta SIM y del módem.
* **Draw.io**: La aplicación web de creación de diagramas es de código abierto.
  + 1. Propiedad Intelectual

El proyecto ha sido una exploración de diversas tecnologías para la creación de un sistema completo. Así pues, su distribución es libre siempre y cuando se referencie al presente trabajo y al autor de este.

* 1. Estructura de la memoria

Sin tener en cuenta el presente capítulo, a continuación, se explicará cuáles son los siguientes capítulos del documento:

* **Capítulo 2. Estado del arte**: en este apartado se explicarán los conceptos actuales en los que se basará el proyecto, como el de las redes de sensores (*Wireless Sensor Networks*) y el Internet de las Cosas (*Internet of Things*), la tecnología relacionada con la agricultura y finalmente, se presentarán con detalle los dispositivos y sensores que van a ser útiles en este proyecto.
* **Capítulo 3. Diseño Lógico**: En un aspecto comercial, se explicará cuál deberá ser el diseño del proyecto para cumplir todos los objetivos que se proponen. Será un modelo lógico para entender cómo debería funcionar el sistema completo.
* **Capítulo 4. Implementación**: En el aspecto práctico, se explicará cómo se realizará un prototipo del proyecto y qué adaptaciones se harán para que siga funcionando de acuerdo con las especificaciones mínimas requeridas. #Se explicará el funcionamiento de los programas
* **Capítulo 5. Presupuesto**: En esta sección, se verá cual va a ser el coste de cada una de las partes del prototipo y como se distribuirían los gastos a lo largo del desarrollo del proyecto.
* **Capítulo 6. Marco Regulador**: De mostrarán aquellas normas o leyes que tengan relación con este proyecto y se explicará por qué se cumplen estas mismas.
* **Capítulo 7. Conclusiones**: En este apartado, se mostrarán los resultados obtenidos de la simulación del prototipo, así como propuestas de mejora para un mejor funcionamiento en el futuro. #No sé si incluir un capítulo que se llame anexos o referencias…

**Capítulo 2**

Estado del arte

Antes de la presentación del diseño y la implementación del prototipo, se van a explicar una serie de conceptos que son esenciales para el trabajo. En primer lugar, se abordará el concepto de Internet de las Cosas para explicar a continuación en qué consiste la Red de sensores inalámbricas. Finalmente, se presentarán los dispositivos que fueron utilizados para cumplir con las expectativas del proyecto.

* 1. Internet de las Cosas

En el momento en el que apareció Internet, éste empezó a evolucionar de una manera muy veloz, desde los antiguos módems hasta las eficientes líneas de fibra óptica actuales. Hoy en día, múltiples dispositivos electrónicos como los móviles, los Smart TV, los Smartwatch… hasta los frigoríficos están conectados a Internet [<http://cba.mit.edu/docs/papers/04.10.i0.pdf>]. ¿A qué se debe esto? El tema que engloba esta idea se llama Internet de las Cosas.

El Internet de las Cosas (del inglés, *Internet of Things* o *IoT*) se refiere a un sistema de dispositivos que están interconectados de manera digital con Internet. Son objetos que tienen identificadores únicos y son capaces de transmitir datos a través de una red [Internet of Things (IoT): A vision, architectural elements, and future directions. J.Gubbietal. Future Generation Computer Systems 29 (2013)1645–1660], es decir, en el IoT todas las comunicaciones son Máquina a Máquina (M2M).

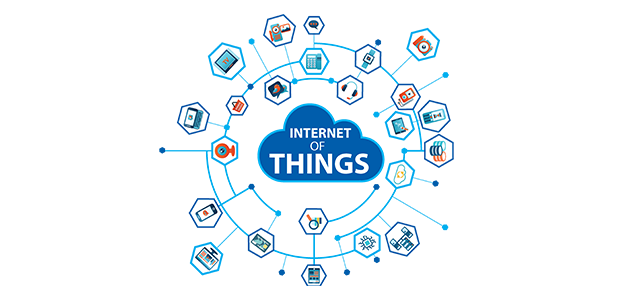


Figura 2.1: Internet de las Cosas [Hughes Europe - <https://europe.hughes.com/sectors/utilities/internet-things>]

El objetivo del *IoT* es facilitar la calidad de vida y a su vez, el modo de vivir de la sociedad. Es clave para administrar eficientemente los recursos que se tienen actualmente. Actualmente, se están realizando proyectos de *IoT* en los que se comprueba la mejora de todo tipo de análisis, recopilación y distribución de datos debido a la enorme capacidad para ello [<https://ticnegocios.camaravalencia.com/servicios/tendencias/internet-las-cosas-mejorar-la-productividad-empresa/>, <https://www.infoq.com/articles/iot-impact-productivity/>].

El concepto de internet de las cosas fue propuesto en 1999 por Kevin Ashton quien usó por primera vez esta expresión en el grupo de investigación Auto-ID Labs. Apareció gracias al estudio de los sistemas por radiofrecuencia, también conocidos como *RFID*, y en las nuevas tecnologías de detección de sensores [<https://en.wikipedia.org/wiki/Auto-ID_Labs>, <https://www.elespectador.com/tecnologia/la-historia-detras-de-la-internet-de-las-cosas-articulo-716678>]. Desde entonces, se utiliza de forma normal para referirse a este sistema de conexión de dispositivos a Internet [<https://www.master-internet-of-things.com/historia-iot/>].

Con el crecimiento del *IoT*, los objetos o *cosas* inteligentes ofrecen millones de datos provenientes del aumento del número de dispositivos conectados entre sí, ya sea a través de Internet o microchips. Es aquí donde aparece el concepto de *Big Data* y todos esos datos son gestionados por las empresas a través de una Analítica de Datos Masivos [<https://www.campusbigdata.com/big-data-blog/item/101-relacion-iot-con-big-data>].

De hecho, en 2018 se llegó a alcanzar alrededor de 17.000 millones de dispositivos conectados a Internet y se prevé que, en 2020, el número se incremente a 20.000 millones de “cosas” conectadas [<https://www.gartner.com/imagesrv/books/iot/iotEbook_digital.pdf>]. Además, se calcula que todo ser humano está rodeado, al menos, por un total de aproximadamente 1000 a 5000 objetos (conectados o no a Internet) [Waldner, Jean-Baptiste (2007). Inventer l'Ordinateur du XXIeme Siècle. London: Hermes Science. p. 254. ISBN 2746215160]. En España, como se puede ver en la figura X2, se puede observar que 1 de cada 4 españoles tienen 5 o más dispositivos conectados a Internet.

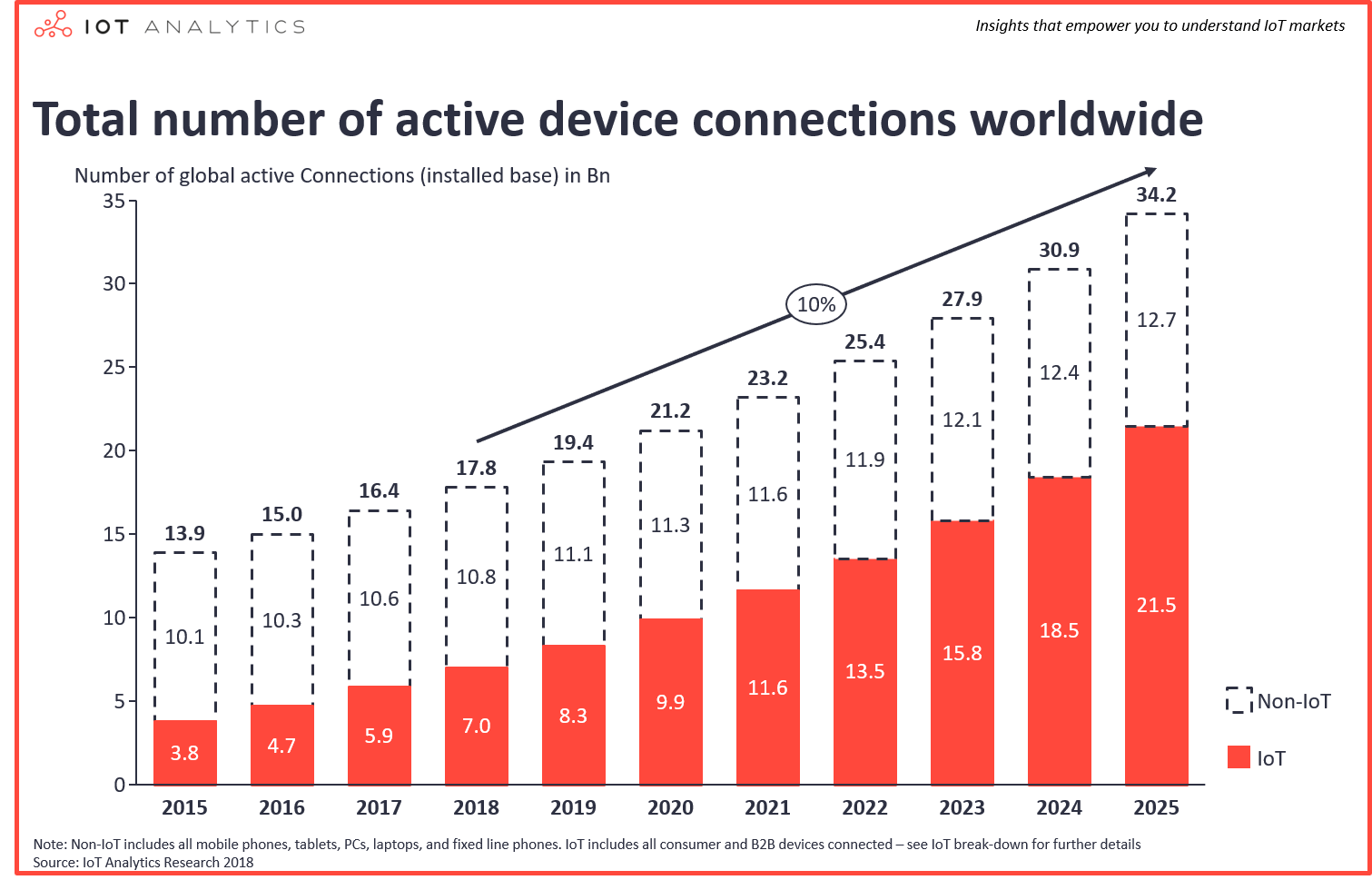


Figura 2.2: Número de dispositivos conectados a Internet en el mundo [<https://iot-analytics.com/state-of-the-iot-update-q1-q2-2018-number-of-iot-devices-now-7b/>]

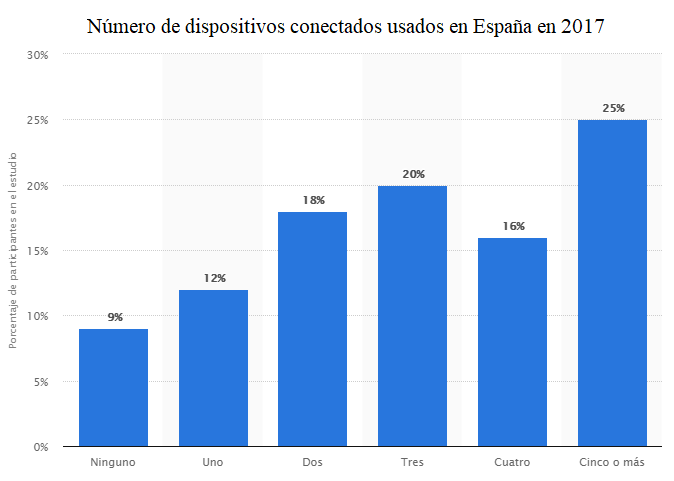


Figura 2.3: Número de dispositivos conectados usados por persona en España [<https://es.statista.com/estadisticas/481066/numero-de-dispositivos-conectados-por-persona-en-espana/>]

* + 1. Evolución de las tecnologías con el Internet de las Cosas

Las tecnologías relacionadas con *IoT* se han incrementado en los últimos años. Gracias a la constante innovación, no se puede tener una visión exacta de cómo evolucionará en el tiempo. Sin embargo, hay muchas tendencias tecnológicas que ayudan a ver esa evolución:

* **Mejora de la Inteligencia Artificial**: Cuando el IoT esté funcionando al máximo rendimiento, la Inteligencia Artificial debe estar preparada para recibir una gran cantidad de datos para analizar. Sobre todo, debe mejorar en el Aprendizaje Automático o *Machine Learning* porque para analizar este volumen de datos, es necesario que se pueda entrenar y aprender por sí sola [<https://blogthinkbig.com/tendencias-iot-futuro>].
* **Mejora de las plataformas móviles**: Volviendo al concepto de *IoT* y aplicando esta idea, las plataformas móviles se convertirán en un importante sistema de gestión para los dispositivos *IoT*.
* **Impacto en las relaciones con los clientes**: La experiencia del cliente se verá influenciada positivamente por la mayor personalización de los productos.
* **Crecimiento del pequeño negocio en los próximos años**: De acuerdo con Forrester Research [<https://www.forrester.com/Internet-Of-Things-%28IoT%29>], el *IoT* se convertirá en una herramienta fundamental para permitir la conexión entre los negocios pequeños y los clientes.
* **Mayor fragmentación**: A medida que vaya creciendo el *IoT*, también se volverá más fragmentado. Es necesario, que haya una mejora e investigación constante a pesar de que varias compañías lideren el impulso de los estándares y certificaciones de dispositivos *IoT*.
* **Seguridad**: Al igual que en el punto anterior, a medida que el *IoT* crezca, la cantidad de datos y dispositivos será mayor y esto indicará un mayor número de brechas susceptibles de ataques. Cuanto más complejo sea la red, más protocolos de seguridad se deben aplicar [Álvaro Manjarrés, 2018, <https://blogs.deusto.es/master-informatica/el-futuro-del-iot-y-los-riesgos-que-conlleva/>].
* **Mayor inversión y gasto**: La Corporación de Datos Internacional (*IDC*) predice que en 2023, el gasto mundial en *IoT* alcanzará 1,1 billones de dólares [<https://www.idc.com/getdoc.jsp?containerId=prUS44596319>]. Esto subraya la importancia de mejorar la experiencia del cliente en casi todas las industrias.
  + 1. Aplicaciones del Internet de las Cosas

Como se ha introducido anteriormente, el *IoT* está cambiando la relación de las personas con el mundo físico. A continuación, se verán las aplicaciones más significativas en las que el *IoT* tiene un rol importante:

* **Agricultura y ganadería inteligente**: El *IoT* tiene capacidad para transformar el mundo. Como se ve en los demás puntos, el *IoT* afecta a industrias más eficientes, la automoción, las *Smart Cities*… Sin embargo, la aplicación del IoT en la agricultura podría tener el mayor impacto. Es la aplicación fundamental para las ciudades más rurales donde se encuentra principalmente el sector primario.

De acuerdo con Beecham Research y la Organización de las Naciones Unidas para la Alimentación y la Agricultura, para mantener el ritmo con el crecimiento de la población mundial se deberá incrementar la producción agrícola en un 70% para 2050 [<http://www.beechamresearch.com/download.aspx?id=1051>] ya que la población para entonces será de 9600 millones de personas.

La agricultura inteligente basada en tecnologías *IoT* permitirá a los agricultores mejorar la productividad sin desperdiciar los recursos, desde la cantidad de fertilizante utilizado hasta el número de viajes realizados por vehículos agrícolas [<http://agriculturers.com/aplicaciones-de-iot-en-agricultura/>]. En cuanto a sus aplicaciones usando *IoT*, se puede destacar la automatización del riego, la erosión del suelo, la evaluación de salud de los cultivos y la fumigación automática de cultivos. En los dos últimos, se aprecia la existencia de drones agrícolas que realizan estas dos tareas de árbol en árbol [<https://www.xataka.com/especiales/penafiel-valladolid-hay-quienes-quieren-revolucionar-cultivo-vino-drones>].



Figura 2.4: Aplicaciones del *IoT* en la agricultura



Figura 2.5: Uso del dron para monitorización de una viña en Rosario (Argentina) [<https://borderperiodismo.com/2018/10/31/de-silicon-valley-a-rosario-con-la-mirada-puesta-en-el-campo/>]

* ***Smart Cities***: En el año 2018, el 23% de todos los proyectos en el mundo del *IoT* estaban dedicados a la mejora de las ciudades inteligentes [<https://iot-analytics.com/top-10-iot-segments-2018-real-iot-projects/>]. Ya que el número de personas en el mundo está creciendo continuamente, se necesita que se mejore la infraestructura en todas las ciudades como en la distribución del agua, el transporte automático, en los sistemas de gestión de la energía, la vigilancia inteligente, la seguridad urbana, la monitorización del medio ambiente o el control del tráfico. Este último punto es muy importante para muchas ciudades ya que hay muchos proyectos destinados al sistema del aparcamiento, control de tráfico, aplicaciones de bicicletas o patines eléctricos [<https://hipertextual.com/2019/02/patienetes-barrios-madrid>] o la mejora del transporte público. Todos estos proyectos ayudarían a reducir un problema muy presente en las ciudades, la contaminación del aire [<https://interestingengineering.com/reducing-air-pollution-in-smart-and-sustainable-future-cities>].
* **Domótica o *Smart Home***: Automatizar las tareas cotidianas aporta un plus de confort que proporciona más tiempo libre para dedicarlo a otras cosas. Uno de los beneficios más señalados es el aumento de seguridad tras la instalación de alarmas y cámaras en el interior y exterior de la vivienda. Otra de las razones importantes, es el ahorro energético que se obtiene cuando se acondiciona las distintas estancias de la casa. Además de repercutir en la economía del usuario, también lo hace en el medio ambiente ya que muchas marcas de electrodomésticos dedican un gran presupuesto en la investigación para crear dispositivos más eficientes y respetuosos con el entorno. Además, todo pasaría a controlarse con el teléfono inteligente ya que actualmente, se debe activar cada electrodoméstico de manera presencial [<https://www.pccomponentes.com/que-es-smarthome>]. Gracias al *IoT*, se pueden ver las alertas recibidas por cada electrodoméstico y actuar en función de la situación.



Figura 2.6: Mejora de la domótica con el *IoT* [<http://www.onptelecoms.com/domotica-iot/>]

* **Conducción Autónoma**: La tecnología digital automotriz se ha centrado en optimizar las funciones internas de los vehículos. Sin embargo, ahora, la atención está creciendo hacia la mejora de la experiencia del automóvil. El objetivo fundamental es la mejora de la percepción del medio ambiente con los sensores *IoT* para permitir una conducción más segura [<https://autopilot-project.eu/about-autopilot/>].

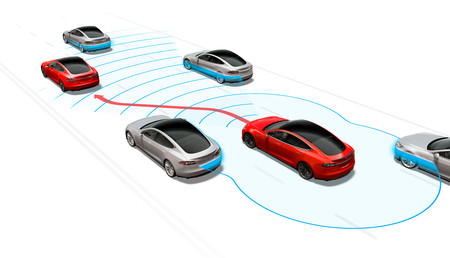


Figura 2.7: Ejemplo del uso de sensores para conducción autónoma [<https://www.motorpasion.com/tesla/precio-autopilot-tesla-subira-otros-1-000-dolares-nuevas-funciones-siguen-llegar>]

* **Sector Sanitario**: También llamado el Internet de las Cosas Médicas (*Internet of Medical Things*, *IoMT*). Son dispositivos que se usan en las áreas de la atención médica y la salud [<https://hitconsultant.net/2019/01/30/is-iomt-tech-iot-market-as-a-whole/#.XTL_fOgzbIU>]. Los dispositivos médicos capturan los datos y de forma inalámbrica los envían a otros dispositivos o a la nube. Estos datos son analizados y desencadenan la acción pautada en los protocolos. Finalmente, los médicos utilizan la información para la toma de decisiones. Sus beneficios son la disminución de los costos en asistencia en salud, reduce los errores de diagnóstico al recibir más datos, la experiencia del paciente mejora considerablemente y ayuda a los pacientes a adherirse a los medicamentos [<https://reportedigital.com/iot/internet-de-las-cosas-medicas-iomt/>].



Figura 2.8: Internet de las Cosas Médicas [<https://hitconsultant.net/2019/01/30/is-iomt-tech-iot-market-as-a-whole/#.XTL_fOgzbIU>]

* **Internet Industrial**: Al igual que en el sector sanitario, existe el Internet Industrial de las Cosas (*IIOT*). Son los dispositivos que están destinados al uso de la Industria. Los beneficios el *IIoT* son las mejoras en la gestión de instalaciones ya que los dispositivos pueden usarse para anticipar problemas originados por humedad o vibraciones, la logística porque la industria está conectada en tiempo real con el cliente y proveedores, y la medición inteligente porque los dispositivos pueden controlar el consumo de recursos y así reducir el gasto de manera significativa [<https://reportedigital.com/iot/internet-industrial-de-las-cosas-iiot/>].



Figura 2.9: Internet Industrial de las Cosas [<https://blog.prodware.es/el-internet-industrial-de-las-cosas-transforma-los-servicios-de-campo/#.XTM5Y-gzbIU>]

2.2. Red de sensores inalámbricas

Las redes de sensores inalámbricas (del inglés, WSN, *Wireless Sensor Networks*) están formadas por nodos sensores o motas, que son dispositivos autónomos distribuidos a lo largo de un área de interés y cuyo objetivo es monitorizar parámetros físicos o ambientales tales como temperatura, sonido, vibraciones, presión, movimiento o agentes contaminantes [<https://www.tekniker.es/es/redes-de-sensores>]. Estos dispositivos trabajan de manera colaborativa para recoger los datos y enviarlos a una pasarela o *Gateway* para que este los transmita a la red exterior (sea un servidor o la nube).

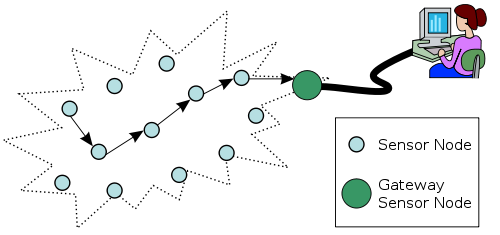


Figura 2.10. Ejemplo de red de sensores inalámbrica [Wikipedia.com]

* + 1. Usabilidad de las Redes de Sensores Inalámbricas

A continuación, se presentarán las características más importantes de las redes de sensores que se han habilitado como una herramienta potencial en el dominio agrícola.

1. **Capacidad de toma de decisiones inteligente**: Esta característica mejora la eficiencia de la energía de toda la red de sensores y, por lo tanto, el tiempo de usabilidad de la red aumenta. De hecho, muchos nodos o motas colaboran entre ellos y colectivamente toman una decisión definitiva [Computational intelligence in wireless sensor networks: A survey <https://www.scopus.com/record/display.uri?eid=2-s2.0-79951581342&origin=inward&txGid=4e3d35e5d823b7ac5de11a258be6fb9e>].
2. **Configuración de topología dinámica**: Para maximizar la vida útil de la red, la topología de la red se configura de tal manera que el número de nodos activos sea mínimo [A survey on topology control in wireless sensor networks: Taxonomy, comparative study, and open issues. <https://www.scopus.com/record/display.uri?eid=2-s2.0-84889567969&origin=inward&txGid=4f4d9973e93041cc9025b4993bfb1ee4>]. Los nodos sensores se mantendrán en el “modo de suspensión” para conservar la energía de la batería.
3. **Tolerancia al fallo**: Una de las dificultades más comunes en la implementación de las redes de sensores es que los nodos sean propensos a fallar [Younis et al., 2014. M. Younis, I.F. Senturk, K. Akkaya, S. Lee, F. Senel. Topology management techniques for tolerating node failures in wireless sensor networks: a survey Comput. Netw., 58 (2014), pp. 254-283]. Esto implica que la red se particiona y el rendimiento se ve afectado. Para ello, los nodos se organizan reconfigurando la topología dinámica.
4. **Conocimiento del contexto**: En base a la información que reciben los sensores, estos adquieren el conocimiento suficiente sobre el ambiente que les rodean. Las soluciones que los sensores toman posteriormente dependen del contexto [Vijay et al., 2011. G. Vijay, E.B.A. Bdira, M. Ibnkahla. Cognition in wireless sensor networks: a perspective IEEE Sens. J., 11 (3) (2011), pp. 582-592].
5. **Escalabilidad**: Los protocolos WSN están diseñados para implementarse en cualquier red, independientemente de su tamaño y número de nodos. Esta característica amplía el potencial del WSN para numerosas aplicaciones [<https://ieeexplore.ieee.org/abstract/document/4482903>].
6. **Heterogeneidad de los nodos**: Las redes de sensores son heterogéneas cuando tienen diferentes tipos de sensores en términos de energía, ancho de banda y memoria [<https://www.researchgate.net/post/What_is_the_exact_definition_of_heterogeneity_in_sensor_node>].
7. **Tolerancia contra los fallos de comunicación en condiciones ambientales adversas**: Debido a la amplia gama de aplicaciones en ambientes agrícolas abiertos, las redes de sensores sufren los efectos de las condiciones ambientales adversas. Parar ello, existe una pila de protocolos de WSN que incluye técnicas para resistir este tipo de fallos debido a los efectos medioambientales [Misra et al., 2014. S. Misra, P. Kar, A. Roy, M.S. Obaidat. Existence of dumb nodes in stationary wireless sensor networks. J. Syst. Softw., 91 (2014), pp. 135-146].
8. **Modo de funcionamiento autónomo**: Una característica importante de las WSN es su modo de funcionamiento autónomo [<https://reader.elsevier.com/reader/sd/pii/S1084804510001244?token=7906A272C9DA0A460964993B0F17A4542FFC8F460272093EC0FFE531A16147E8F14F6BC4172BAA6FBB903B6DDBAA178C>, J. Netw. Comput. Appl., 34 (5) (2011), pp. 1530-1544] y de adaptación [P. Nicopolitidis, G.I. Papadimitriou, A.S. Pomportsis, P.G. Sarigiannidis, M.S. Obaidat. Adaptive wireless networks using learning automata. IEEE Wirel. Commun., 18 (2) (2011), pp. 75-81]
9. **Seguridad de la información**: Las redes de sensores llevan información sin formato sobre los parámetros del campo. Para garantizar la seguridad de la información detectada y restringir a los usuarios no autenticados, las WSN proporcionan mecanismos de control de acceso y detección de anomalías [Karapistoli, E., Sarigiannidis, P., Economides, A.A., 2013. SRNET: a real-time, cross-based anomaly detection and visualization system for wireless sensor networks. In: Proceedings of the Tenth Workshop on Visualization for Cyber Security, pp. 49–56.].
   * 1. Arquitectura de la red de sensores en la agricultura

Dependiendo del tipo de aplicación que se necesita, se debe buscar una arquitectura u otra. Las arquitecturas de red se clasifican en varias categorías como el movimiento de los nodos sensores, en sus tipos de sensores en términos de heterogeneidad, o en la jerarquía [Wireless sensor networks for agriculture: The state-of-the-art in practiceand future challenges Tamoghna Ojhaa,b, Sudip Misraa, Narendra Singh Raghuwanshi, pp. 66–84]. En el primer punto, el movimiento de los nodos sensores, se pueden identificar las diferentes arquitecturas:

* **Arquitectura estacionaria**: Los nodos sensores están desplegados en una posición fija en el periodo en el que dure la aplicación. Por ejemplo, es ideal para sistemas de monitorización de riego o control del uso de fertilizantes.
* **Arquitectura móvil**: Los nodos sensores cambian su posición dependiendo del tiempo. Por ejemplo, una aplicación basada en esta arquitectura sería el control de tractores autónomos.
* **Arquitectura hibrida**: Sería la combinación entre nodos estacionarios y móviles. Por ejemplo, sería aplicable en sensores del campo, equipos agrícolas móviles, usuarios de teléfono móvil y transporte de ganado.

En cuanto a los tipos de sensores, se pueden encontrar las siguientes categorías:

* **Arquitectura homogénea**: Se compone de dispositivos equipados con sensores de potencial similar. Este tipo de marco se usa normalmente en aplicaciones basadas en implementaciones no planificadas. Sin embargo, este tipo de arquitectura carece de variedad en términos de hardware de comunicación. En consecuencia, los esquemas y protocolos de comunicación están diseñados teniendo en cuenta esta limitación. Un ejemplo de aplicación es la recolección de datos sobre la cantidad de nutrientes en el suelo.
* **Arquitectura heterogénea**: En este tipo de arquitectura, existen varios tipos de nodos sensores y dispositivos. Estos dispositivos varían en términos de potencia de cálculo, memoria, capacidad de detección y unidades de transceptor. Por ejemplo, en cualquier aplicación de gestión de riego, los nodos de sensores en campo comunican su información detectada a la pasarela, que de nuevo transfiere la información a un usuario remoto (comunicación por radio frecuencia o GSM).

Finalmente, en términos de jerarquía, las arquitecturas se clasificarían en las siguientes categorías:

* **Arquitectura de un solo nivel**: es común en aplicaciones de pequeña escala. En este tipo de arquitectura, los dispositivos en el campo y los nodos sensores comunican directamente sus datos a la pasarela, que está colocada cerca del área de aplicación.
* **Arquitectura multinivel**: Los nodos de sensores en el campo permanecen en el nivel inferior de la jerarquía y forman los grupos básicos o clústeres. Entonces, los siguientes niveles de jerarquía incluyen varios clústeres para llegar a los nodos de la pasarela. Normalmente, las arquitecturas multinivel consisten en nodos heterogéneos.
  + 1. Arquitectura de los nodos sensores

En este apartado, en la arquitectura de las motas o los nodos sensores se pueden distinguir dos tipos:

* **Sistema en paquete** (*System-in-Package,* *SiP*): Se define como la combinación de dispositivos múltiples que incluyen elementos pasivos (como resistencias y condensadores), montados juntos y manteniendo la disposición de conectar componentes externos posteriormente. Los factores para la selección de los componentes de una mota de este tipo son el procesador, el transmisor-receptor, la memoria, la potencia y el coste.
* **Sistema en chip** (*System on Chip*, *SoC*): Sigue una reducción de diseño más específica para cada aplicación. Así pues, minimiza los requisitos de energía y coste del diseño. Proporciona una integración de múltiples núcleos de procesadores programables, unidades de memoria, unidades de entrada/salida y bloques personalizados.

En la tabla siguiente, se pueden observar las diferencias entre ambos sistemas:

|  |  |  |
| --- | --- | --- |
| **Factor** | **SiP** | **SoC** |
| **Procesador** | Pocos y heterogéneos | Múltiple y heterogéneo |
| **Consumo** | Alto | Bajo |
| **Coste** | Alto | Bajo |
| **Tamaño** | Grande | Pequeño |
| **Memoria** | En un módulo separado | Integrado |

Tabla 2.1: Diferencias entre *SiP* y *SoC* [Elaboración propia, información de T. Ojha et al./Computers and Electronics in Agriculture 118 (2015) 66–88]

* 1. Tecnologías y estándares usados en la agricultura

En esta sección, se explicará con detalle las diferentes tecnologías de comunicación inalámbrica y los estándares presentes en las aplicaciones en la agricultura.

* + 1. Zigbee

Esta tecnología define la red y los protocolos de la capa de aplicación basados en la norma IEEE 802.15.4 y las definiciones de la capa MAC necesarias para diseñar una red inalámbrica de área personal (*Wireless Personal Area Network*, *WPAN*) utilizando dispositivos habilitados para radio de baja potencia. Admite la comunicación a corta distancia (10-20 metros) a través de redes multinivel y de malla. Su velocidad de datos es relativamente baja entre 20-40 kbps en la banda de frecuencia ISM (Bandas de radio industriales, científicas y médicas [<https://www.itu.int/net/ITU-R/terrestrial/faq/index.html#g013>]) de 868-915 MHz y 250 kbps a 2.4 GHz [IEEE 802.15 WPAN™ Task Group 4 (TG4) <http://www.ieee802.org/15/pub/TG4.html>]. No requiere de un *hardware* de alta especificación ya que se anuncian dispositivos con hasta 128 kB de almacenamiento [[http://www.ti.com/product/CC2430#](http://www.ti.com/product/CC2430)]. Es eficiente en energía, coste bajo y confiable por ello es muy utilizada en la agricultura. Se definen tres tipos distintos de dispositivo Zigbee [ZigBee Wireless Sensor and Control Network. By Ata Elahi, Adam Gschwender - Punto 2.2 ZigBee Device Types, <http://www.informit.com/articles/article.aspx?p=1409785&seqNum=3>]:

* **Coordinador Zigbee**: Uno por red ya que se encarga de controlar esta y los caminos que deben seguir los dispositivos para conectarse entre ellos.
* **Router Zigbee**: Interconecta dispositivos separados en la red.
* **Dispositivo final**: Tiene la funcionalidad necesaria de comunicarse su nodo padre (sea coordinador o router) pero no puede transmitir su información a otros dispositivos. De esta forma, el nodo suele estar en suspensión la mayor parte del tiempo, conservando la energía de las baterías.



Figura 2.11: Ejemplo de módulo de red Zigbee [<https://makerfabs.com/index.php?route=product/product&product_id=411>]

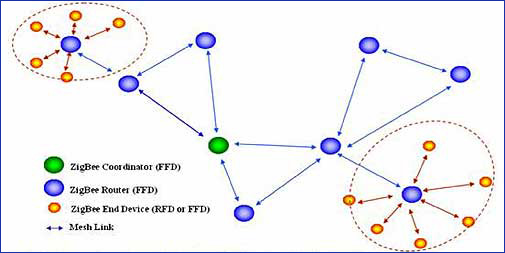


Figura 2.12: Red de Sensores Zigbee [<http://redesdesensoreswsn.blogspot.com/2012/07/redes-de-sensores-inalambricas-wsn-upt.html>]

* + 1. WiFi

WiFi es un estándar de red de área local inalámbrica (WLAN) para el intercambio de información o la conexión a Internet de forma inalámbrica basada en la familia de estándares IEEE 802.11 [IEEE Standard for Information technology, 2005-IEEE Standard for Information technology, 2012a]. WiFi ofrece un rango de comunicación adecuado del orden de 20 m (interior) a 100 m (exterior) con una velocidad de transmisión de datos del orden de 2–54 Mbps a una frecuencia de 2,4 GHz de la banda ISM [<https://www.lifewire.com/range-of-typical-wifi-network-816564>, <https://www.quora.com/Whats-the-maximum-distance-a-Wi-Fi-signal-can-be-reached>].

* + 1. GSM y su evolución

La red GSM (Sistema global de comunicaciones móviles o *Global System for Mobile Communications*) es un estándar de segunda generación (2G) ya que las comunicaciones empiezan a producirse de un modo completamente digital en comparación con la primera generación de teléfonos. [<https://es.ccm.net/contents/681-estandar-gsm-sistema-global-de-comunicaciones-moviles>, <https://books.google.es/books?id=go2fDwAAQBAJ&printsec=frontcover&dq=gsm&hl=es&sa=X&ved=0ahUKEwjju9mryMDjAhVOi1wKHayzCAMQ6AEIKzAA#v=onepage&q=gsm&f=false>]. En comparación con las otras tecnologías, su rango de transmisión de datos se extiende por todo el mundo. Su banda de frecuencia depende del territorio, aunque en Europa se utilizan las bandas de 900 y 1800 MHz y se logra una velocidad de datos de 50-100 kbps [Victor L. Granatstein, Physical Principles of Wireless Communications, Segunda Edición, página 29]. Los teléfonos o estaciones móviles necesitan de una tarjeta SIM para poder utilizar la tecnología GSM, que contiene información sobre el terminal y su usuario [<https://www.universidadviu.es/que-es-gsm-y-como-funciona/>].

Más adelante, nuevas generaciones aparecieron para mejorar la segunda generación como GPRS, EDGE, UMTS (3G), 4G y 5G. Actualmente, es fácil conectarse a la red vía 3G o 4G ya que el 5G todavía no se ha implementado en todos los lugares. En España, las velocidades de descarga rondan los 6.5 Mbps en 3G, mientras que en 4G rondan los 25 Mbps [<https://www.testdevelocidad.es/velocidad-internet-movil-3g-4g/>].

Por otro lado, a parte de su uso en comunicación, GSM también puede utilizarse para localizar un dispositivo sin tener que recurrir al uso del GPS. Esto es muy útil para aquellos dispositivos que no tengan GPS. Por lo general, un móvil o dispositivo con GSM está conectado a una o varias torres de telefonía. Mediante un cálculo de triangulación, es posible saber la posición del objeto con un pequeño margen de error que dependerá del número de torres en la zona. El dispositivo localizará las torres más cercanas y calculará el tiempo que tarda la señal de torre a torre y la fuerza de la misma. [<https://www.xatakamovil.com/conectividad/como-localizar-mi-posicion-gps-a-gps-wifi-y-redes-gsm>, <https://www.open-electronics.org/gsm-localizer-without-gps-part-1-introduction/>, <http://www.opencellid.org/#zoom=18&lat=37.802137&lon=-122.419357>, <https://www.instructables.com/id/Raspberry-Pi-Location-Tracker/>, <https://antenasgsm.com/39.8311807928473/-3.1802600639999845/12/true,true,true,true,true>].



Figura 2.13: Ejemplo de triangulación por GSM para localizar un dispositivo [<https://movilfacil.wordpress.com/2011/03/25/cap-8-posicionamiento-localizacion/>]

* 1. Tecnologías utilizadas

En este apartado se explicará qué recursos se han utilizado para hacer que el proyecto funcione. Más adelante, en los capítulos del diseño y la implementación se explicará cual será el papel fundamental de cada uno.

* + 1. Raspberry Pi

Raspberry Pi es un ordenador de bajo coste desarrollado por la Fundación Raspberry Pi en Reino Unido, del tamaño de una tarjeta de crédito, que se conecta a un monitor, y usa un teclado y ratón estándar. Es capaz de hacer todo lo que puede hacer un ordenador normal, desde navegar por Internet y reproducir videos hasta hacer hojas de cálculo, procesamiento de textos y otros programas [<https://www.raspberrypi.org/help/what-%20is-a-raspberry-pi/>].

Esta placa cuenta con una arquitectura ARM y el sistema operativo oficial sobre el que se monta se llama Raspbian OS, una versión de Debian. Además, se pueden instalar otros sistemas como Windows 10 IoT, que tiene menos recursos que el Windows 10 de escritorio, u otros sistemas basados en Linux.

Una de las ventajas más importantes de este ordenador, aparte de que su tamaño es comparable al de una tarjeta de crédito, es su precio bajo. Este modelo hoy en día tiene un coste aproximado de 35 Euros.

Las aplicaciones de este dispositivo son muy numerosas. Básicamente depende de la imaginación del usuario ya que la funcionalidad de la Raspberry Pi es muy flexible. Por ejemplo: seguimiento de objetos en movimiento, reconocimiento facial, monitorización, transmisión de datos, desactivador de drones en movimiento [<http://soloelectronicos.com/2017/04/24/top-100-de-proyectos-con-raspberry-pi/>].

* + 1. Sensores

Los sensores son dispositivos diseñados para recibir información de una magnitud del exterior y transformarla en otra magnitud, generalmente eléctrica, que se pueda cuantificar y manipular [<http://www.profesormolina.com.ar/tecnologia/sens_transduct/que_es.htm>]. Existen diferentes tipos de sensores: de posición lineal y angular, de desplazamiento y deformación, de velocidad lineal y angular, de aceleración, de presión, de caudal, de temperatura, de presencia, táctiles, de visión artificial, de proximidad, de acústico, de acidez, de luz y de captura de movimiento [<https://es.wikipedia.org/wiki/Sensor>].

* + 1. Raspbian OS

Raspbian es una distribución del sistema operativo *GNU*/Linux basado en Debian optimizado para el hardware de Raspberry Pi [<https://es.wikipedia.org/wiki/Raspbian>, <https://www.raspbian.org/>]. Raspbian no está afiliado a la Fundación Raspberry Pi, de hecho, fue creado por Mike Thompson y Peter Green como un proyecto independiente.

Al ser una distribución de *GNU*/Linux las posibilidades son infinitas ya que el usuario puede descargar multitud de programas contenidos en repositorios como si se tratase de una distribución *GNU*/Linux para equipos de escritorio.

* + 1. Python

Python es un lenguaje de programación interpretado cuyo objetivo principal es hacer que su sintaxis favorezca un código legible. Soporta orientación a objetos, programación imperativa y rasgos de programación funcional. Es considerado un lenguaje de alto nivel aunque su biblioteca estándar está combinada lenguajes de bajo nivel como C y C++ [<https://es.wikipedia.org/wiki/Python>, <https://wiki.python.org/moin/BeginnersGuide/Programmers>].

En el proyecto, se escribirán todos los programas en este lenguaje debido a su facilidad para recoger información de los sensores y su análisis.

* + 1. Hologram

Hologram se fundó en 2013 por Ben Forgan y Pat Wilbur en Chicago (Estados Unidos de América) con la visión de desarrollar una nueva forma de telecomunicaciones para proporcionar conectividad segura a los dispositivos IoT.

Hologram es la solución a aquellos millones de dispositivos conectados que necesitan más allá de una red WiFi para funcionar. Se basa en usar las redes celulares que pertenecen a los grandes proveedores de telefonía. Hologram negocia con ellos tener permiso a su uso verificando el hardware y estableciendo una arquitectura de datos compatible con las regulaciones. De esta manera, el modelo celular tradicional está cambiando con una solución integrada en software diseñada para expandir el alcance de la próxima generación de dispositivos conectados.

En la actualidad, el sistema de Hologram permite que cualquier cliente se conecte a más del 85% del área del planeta Tierra. El cliente solamente necesita una tarjeta SIM perteneciente a Hologram y un módulo compatible para conectar la tarjeta con el dispositivo [<https://hologram.io/products/cellular/>].

Además, existe una nube creada por Hologram el cual tiene las siguientes funciones: ver el tráfico de datos, enviar SMS, crear alertas o rutas para que una vez la nube reciba un mensaje determinado, actúe realizando una tarea específica (como enviar un email al usuario con una alerta, o publicar un resultado en una página web alternativa), entre otras.

* + 1. ThingSpeak

ThingSpeak es un servicio de plataforma de análisis de *IoT* que permite añadir, visualizar y analizar flujos de datos en vivo en la nube. ThingSpeak proporciona visualizaciones instantáneas de los datos publicados por los dispositivos en ThingSpeak. Además, es posible ejecutar código MATLAB para realizar análisis y procesamiento en línea de los datos que ingresan. ThingSpeak a menudo se usa para prototipos y sistemas de prueba de concepto de *IoT* que requieren análisis [<https://thingspeak.com/pages/learn_more>]. Para mostrar la información, se crea un canal en el cual se pueden ver hasta ocho gráficas con diferentes medidas.

* + 1. Google Maps

Google Maps es un servidor de aplicaciones de mapas que fue anunciado por primera vez en 2005. Hoy en día este sofware gratuito pertenece a Alphabet Inc (cuya principal subsidiaria es Google).

Este software ofrece imágenes y fotografías por satélite de todo el mundo. Además, incluye imágenes de la calle con Google Street View, el tráfico en tiempo real y un guía de rutas a pie, transporte público, coche y otros modos de transporte.

* + 1. MobaXterm

MobaXterm es un terminal mejorado para Windows con un servidor X11, un cliente *SSH* con pestañas y otras herramientas de comunicación remota como *VNC*, *telnet*, *rlogin*. En este proyecto, se utilizará MobaXterm principalmente para realizar conexiones *SSH*.

* + 1. R y RStudio

R es un lenguaje para computación estadística y gráficos. Fue desarrollado en los laboratorios Bell y se suele usar junto con RStudio (Principal entorno de programación de R). Es un conjunto de instalaciones de software para la manipulación de datos, el cálculo y la visualización gráfica [<https://www.r-project.org/about.html>].

* + 1. Draw.io

Draw.io es una tecnología perteneciente a JGraph Ltd de código abierto para crear aplicaciones de creación de diagramas, y la aplicación de diagramación gratuita de un usuario final basada en navegador más utilizada del mundo [<https://about.draw.io/about-us/>].

* + 1. Microsoft Office

Microsoft Office es un conjunto de programas ofimáticos que permiten automatizar y perfeccionar las actividades habituales de una oficina. Fue desarrollado por Microsoft Corporation en 1990 y actualmente entre sus programas destacan el procesador de texto Word; el programa de la hoja de cálculo, Excel; el programa para el desarrollo de presentaciones visuales, Power Point; y el administrador de información personal y cliente de correo electrónico, Outlook.

* + 1. Comunicación

En este apartado se mostrarán los diferentes tipos de comunicación que se han usado en el proyecto.

* ***Socket TCP/IP***: Proporciona un flujo de datos, generalmente de manera ordenada y fiable entre dos programas. La conexión punto a punto descrita por un socket se define de manera exclusiva por su dirección de Internet (IP), su protocolo de comunicación (Sea UDP o TCP) y el puerto, que identifica una aplicación (puede ser un puerto conocido como el SSH o 22, o uno definido por el usuario) [<https://www.ibm.com/support/knowledgecenter/en/SSB27H_6.2.0/fa2ti_what_is_socket_connection.html>].
* ***SMS***: También llamado servicio de mensajes cortos o simples. Es un servicio creado junto al sistema global para las comunicaciones móviles (GSM) que está permite el envío de mensajes cortos entre teléfonos móviles [<https://es.wikipedia.org/wiki/Servicio_de_mensajes_cortos>]. El SMS sirve para teléfonos fijos y otros dispositivos de mano.
* **Correo electrónico**: también llamado *email* es un servicio de red que permite a los usuarios recibir y enviar mensajes (cartas digitales) mediante redes de comunicación electrónica [<https://es.wikipedia.org/wiki/Correo_electr%C3%B3nico>]. Para usar este tipo de comunicación, el usuario debe disponer de una cuenta de correo electrónico del formato “usuario\_ejemplo@ejemplo.com”. Cada dirección de correo electrónico se compone de una parte local (usuario\_ejemplo), el separador (@), y la parte del dominio (ejemplo.com).
* ***SSH* (*Secure Shell*)**: Es un protocolo de comunicación que fue desarrollado originalmente por Tatu Ylonen en 1995. *SSH* usa encriptación para asegurar la conexión entre un cliente y un servidor. Todas las autenticaciones del usuario, comandos, salidas y transferencias de archivos están encriptadas para evitar ataques en la red [<https://www.ssh.com/ssh/>].

**Capítulo 3**

Análisis y diseño

En este capítulo, se explicará en un caso ideal cómo debería ser el sistema completo. Para ello, se abordarán los diferentes requisitos del sistema para su posterior reflexión sobre los casos de uso. De esta manera, con los objetivos y los requisitos definidos, se podrá diseñar de una manera lógica la arquitectura del sistema.

* 1. Requisitos del sistema

En este apartado, se mostrarán los **requisitos funcionales,** que son todas las capacidades que describen el sistema, y los **requisitos no funcionales**, los cuales son restricciones o propiedades impuestas por sistema a desarrollar. Sin embargo, es necesario tener presente las especificaciones importantes y las que sean incompatibles con otras. Finalmente, se presentarán los casos de uso en los diferentes objetivos del proyecto.

Los requisitos del sistema se van a presentar en una tabla con la siguiente estructura [<http://www.juntadeandalucia.es/servicios/madeja/contenido/recurso/456>]:

|  |  |
| --- | --- |
| RS-XX | |
| Nombre |  |
| Descripción |  |
| Dependencias |  |
| Importancia |  |
| Prioridad |  |
| Comentarios |  |

Tabla 3.1: Ejemplo de requisito del sistema

A continuación, se detallan los campos contenidos de la tabla X:

* **Identificador**: Es el valor que identifica de forma única cada uno de los requisitos desarrollados. La estructura del identificador mantiene la forma “RS-XX” ya que está compuesto por dos partes: “RS” indica que es un requisito del sistema y la segunda parte señala el número identificador del requisito. En la primera parte del identificador, se añade “F” si se trata de un requisito funcional y “NF” si es un requisito no funcional.
* **Nombre**: es el nombre que describe al identificador.
* **Descripción**: es la explicación detallada del requisito.
* **Dependencias**: en esta celda se indicarán qué otros requisitos son necesarios para cumplir con el requisito.
* **Importancia**: representa gradualmente la necesidad para que sea llevado a cabo el requisito. Los posibles valores son: opcional, deseable y esencial.
* **Prioridad**: representa el nivel de prioridad para que se cumpla el requisito en el sistema. Los posibles valores son: baja, media y alta.
* **Comentarios**: posibles valoraciones adicionales acerca del requisito.
  + 1. Requisitos funcionales

|  |  |
| --- | --- |
| RSF-01 | |
| Nombre | Valor de la humedad en el suelo en tiempo real. |
| Descripción | El sistema será capaz de devolver el valor de la humedad en el suelo en tiempo real. |
| Dependencias | \*\*\*\* |
| Importancia | Esencial. |
| Prioridad | Alta. |
| Comentarios | El valor de la humedad del suelo se recogerá automática y manualmente (si el usuario requiere una inspección manual). |

Tabla 3.2: Requisito de sistema – Requisito funcional 01

|  |  |
| --- | --- |
| RSF-02 | |
| Nombre | Valor de la aceleración en tiempo real. |
| Descripción | El sistema podrá devolver los valores de la aceleración en los ejes X, Y, Z recogidos por el giroscopio MPU6050. |
| Dependencias | \*\*\*\* |
| Importancia | Esencial. |
| Prioridad | Alta. |
| Comentarios |  |

Tabla 3.3: Requisito de sistema – Requisito funcional 02

|  |  |
| --- | --- |
| RSF-03 | |
| Nombre | Valor de la rotación en tiempo real. |
| Descripción | El sistema podrá devolver los valores de la rotación (en grados) en los ejes X e Y recogidos por el giroscopio MPU6050. |
| Dependencias | \*\*\*\* |
| Importancia | Esencial. |
| Prioridad | Alta. |
| Comentarios |  |

Tabla 3.4: Requisito de sistema – Requisito funcional 03

|  |  |
| --- | --- |
| RSF-04 | |
| Nombre | Detección de cuerpo térmico. |
| Descripción | El sistema podrá indicar si detecta un cuerpo térmico moviéndose en la zona de desarrollo. |
| Dependencias | \*\*\*\* |
| Importancia | Esencial. |
| Prioridad | Alta. |
| Comentarios | Valor binario (Sí o No). |

Tabla 3.5: Requisito de sistema – Requisito funcional 04

|  |  |
| --- | --- |
| RSF-05 | |
| Nombre | Registro de los datos a lo largo del tiempo. |
| Descripción | El sistema podrá devolver un registro de los datos de los diferentes sensores. |
| Dependencias | \*\*\*\* |
| Importancia | Esencial. |
| Prioridad | Alta. |
| Comentarios |  |

Tabla 3.6: Requisito de sistema – Requisito funcional 05

|  |  |
| --- | --- |
| RSF-06 | |
| Nombre | Interfaz web. |
| Descripción | Los datos del sistema podrán ser accedidos a través de una interfaz web. |
| Dependencias | \*\*\*\* |
| Importancia | Esencial. |
| Prioridad | Alta. |
| Comentarios |  |

Tabla 3.7: Requisito de sistema – Requisito funcional 06

|  |  |
| --- | --- |
| RSF-07 | |
| Nombre | Terminal. |
| Descripción | Los datos del sistema podrán ser accedidos a través de una terminal. |
| Dependencias | \*\*\*\* |
| Importancia | Esencial. |
| Prioridad | Alta. |
| Comentarios |  |

Tabla 3.8: Requisito de sistema – Requisito funcional 07

|  |  |
| --- | --- |
| RSF-08 | |
| Nombre | Comunicación del sistema. |
| Descripción | Los dispositivos del sistema serán capaces de comunicarse con el usuario. |
| Dependencias | \*\*\*\* |
| Importancia | Esencial. |
| Prioridad | Alta. |
| Comentarios |  |

Tabla 3.9: Requisito de sistema – Requisito funcional 08

|  |  |
| --- | --- |
| RSF-09 | |
| Nombre | Activación o desactivación del sistema de seguridad. |
| Descripción | El usuario podrá activar o desactivar el sistema cuando lo requiera. |
| Dependencias | \*\*\*\* |
| Importancia | Esencial. |
| Prioridad | Alta. |
| Comentarios |  |

Tabla 3.10: Requisito de sistema – Requisito funcional 09

|  |  |
| --- | --- |
| RSF-10 | |
| Nombre | Activación o desactivación del motor de gasoil. |
| Descripción | El sistema podrá activar o desactivar automáticamente el motor de gasoil en función de los parámetros obtenidos por los sensores. |
| Dependencias | \*\*\*\* |
| Importancia | Esencial. |
| Prioridad | Alta. |
| Comentarios | El motor de gasoil es el encargado de dar energía a la bomba de riego subterránea para sacar agua a la superficie. |

Tabla 3.11: Requisito de sistema – Requisito funcional 10

* + 1. Requisitos no funcionales

|  |  |
| --- | --- |
| RSNF-01 | |
| Nombre | Lenguaje de la aplicación del *Gateway*. |
| Descripción | El Gateway alojado en la Raspberry Pi estará desarrollado en Python. |
| Dependencias | \*\*\*\* |
| Importancia | Deseable. |
| Prioridad | Alta. |
| Comentarios | La aplicación del servidor en Python se desarrolla de una manera más interactiva. Sin embargo, otros lenguajes están permitidos. |

Tabla 3.12: Requisito de sistema – Requisito no funcional 01

|  |  |
| --- | --- |
| RSNF-02 | |
| Nombre | Lenguaje de la aplicación del cliente. |
| Descripción | El cliente estará desarrollado en Python. |
| Dependencias | \*\*\*\* |
| Importancia | Deseable. |
| Prioridad | Alta. |
| Comentarios | La aplicación del servidor en Python se desarrolla de una manera más interactiva. Sin embargo, otros lenguajes están permitidos. |

Tabla 3.13: Requisito de sistema – Requisito no funcional 02

|  |  |
| --- | --- |
| RSNF-03 | |
| Nombre | Sistema operativo del *Gateway*. |
| Descripción | El Gateway alojado en la Raspberry Pi tendrá como sistema operativo Raspbian. |
| Dependencias | \*\*\*\* |
| Importancia | Esencial. |
| Prioridad | Alta |
| Comentarios |  |

Tabla 3.14: Requisito de sistema – Requisito no funcional 03

|  |  |
| --- | --- |
| RSNF-04 | |
| Nombre | Datos de la humedad |
| Descripción | Los datos de la humedad se alojarán en la interfaz web. |
| Dependencias | \*\*\*\* |
| Importancia | Esencial. |
| Prioridad | Alta. |
| Comentarios |  |

Tabla 3.15: Requisito de sistema – Requisito no funcional 04

|  |  |
| --- | --- |
| RSNF-05 | |
| Nombre | Comunicación con la base de datos (interfaz web). |
| Descripción | La comunicación entre la Raspberry Pi y la interfaz web será posible gracias al uso de la red GSM. |
| Dependencias | \*\*\*\* |
| Importancia | Esencial. |
| Prioridad | Alta. |
| Comentarios |  |

Tabla 3.16: Requisito de sistema – Requisito no funcional 05

|  |  |
| --- | --- |
| RSNF-06 | |
| Nombre | Comunicación automática vía correo electrónico. |
| Descripción | Desde la Raspberry Pi, se enviarán alertas al usuario a su correo electrónico. |
| Dependencias | \*\*\*\* |
| Importancia | Esencial. |
| Prioridad | Alta. |
| Comentarios |  |

Tabla 3.17: Requisito de sistema – Requisito no funcional 06

|  |  |
| --- | --- |
| RSNF-07 | |
| Nombre | Comunicación automática vía SMS. |
| Descripción | Desde la Raspberry Pi, se enviarán alertas al usuario a su teléfono móvil. |
| Dependencias | \*\*\*\* |
| Importancia | Esencial. |
| Prioridad | Alta. |
| Comentarios | Ojo, pueden ser alertas de sobre-explotación del agua o de robo. |

Tabla 3.18: Requisito de sistema – Requisito no funcional 07

|  |  |
| --- | --- |
| RSNF-08 | |
| Nombre | Paso de datos entre la Raspberry Pi y la interfaz web |
| Descripción | La información enviada a la web se hará por un POST request. |
| Dependencias | \*\*\*\* |
| Importancia | Esencial. |
| Prioridad | Alta. |
| Comentarios |  |

Tabla 3.19: Requisito de sistema – Requisito no funcional 08

|  |  |
| --- | --- |
| RSNF-09 | |
| Nombre | Creación de túnel entre el usuario y la Raspberry Pi |
| Descripción | Previa a la conexión entre el usuario y la Raspberry Pi, se creará un túnel de conexión entre ambos para facilitar el sistema. |
| Dependencias | \*\*\*\* |
| Importancia | Esencial. |
| Prioridad | Alta. |
| Comentarios | Así podrá funcionar la comunicación vía terminal. |

Tabla 3.20: Requisito de sistema – Requisito no funcional 09

|  |  |
| --- | --- |
| RSNF-10 | |
| Nombre | Conexión SSH entre la Raspberry Pi y el usuario |
| Descripción | Además de conocer los datos en la interfaz web, también el usuario podrá monitorear el sistema pidiendo instrucciones a la Raspberry Pi vía terminal a través del puerto 12345. |
| Dependencias | RSNF-09 |
| Importancia | Esencial. |
| Prioridad | Alta |
| Comentarios | De esta manera, podrá ejecutar más funciones que ver los resultados en la interfaz web. |

Tabla 3.21: Requisito de sistema – Requisito no funcional 10

|  |  |
| --- | --- |
| RSNF-11 | |
| Nombre | Descarga de la base de datos. |
| Descripción | El usuario podrá descargarse los datos almacenados en la interfaz web en los formatos JSON, XML o CSV. |
| Dependencias | \*\*\*\* |
| Importancia | Esencial. |
| Prioridad | Alta. |
| Comentarios |  |

Tabla 3.23: Requisito de sistema – Requisito no funcional 11

|  |  |
| --- | --- |
| RSNF-12 | |
| Nombre | Uso fácil del sistema. |
| Descripción | El usuario podrá obtener la información de cualquiera de las aplicaciones realizando menos de 5 pasos. |
| Dependencias | \*\*\*\* |
| Importancia | Opcional. |
| Prioridad | Media. |
| Comentarios |  |

Tabla 3.24: Requisito de sistema – Requisito no funcional 12

|  |  |
| --- | --- |
| RSNF-13 | |
| Nombre | Durabilidad del sistema. |
| Descripción | El sistema tendrá una batería que alargará la vida del sistema y esta se cargará al mismo tiempo cuando el motor de gasoil está activo. |
| Dependencias | \*\*\*\* |
| Importancia | Esencial. |
| Prioridad | Alta. |
| Comentarios |  |

Tabla 3.25: Requisito de sistema – Requisito no funcional 13

* + 1. Casos de uso

Un caso de uso describe una acción o actividad en el cual indica cómo se utilizará el sistema a desarrollar por sus futuros usuarios para conseguir un objetivo específico [<http://www.juntadeandalucia.es/servicios/madeja/contenido/recurso/416>]. En este apartado, se muestran la especificación de los actores, los diagramas de los propios casos de uso y sus correspondientes descripciones.

* + - 1. Actores

Los actores o usuarios, aquellos que interactúan en el sistema, que se van a identificar en el sistema son los siguientes:

* **Usuario administrador**: Usuario que está autenticado en el sistema y tiene capacidades de administrador.
* **Usuario cliente**: Usuario normal que recibe los datos.
  + - 1. Diagramas de los casos de uso

A continuación, se presentarán las imágenes que representan las posibles operaciones que puede realizar el usuario en el sistema en cada uno de los tres escenarios posibles.

En el primer escenario, se representa como se accede a la información a través de la interfaz web. Cualquier usuario necesita iniciar sesión, pero solo el administrador puede conceder permisos para que los clientes puedan ver la información. Como se puede ver en la figura 3.1, las flechas verdes corresponden a las tareas disponibles del administrador, las azules corresponden a las del cliente y las naranjas a ambos.

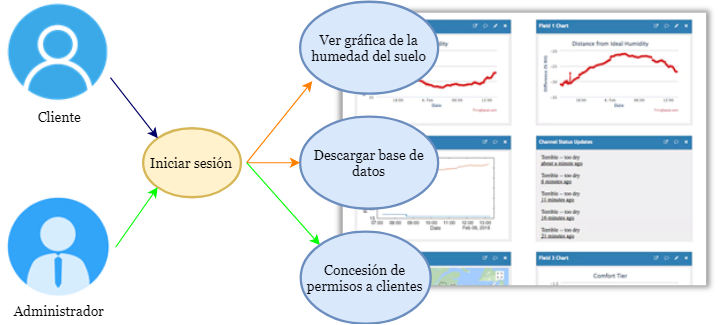


Figura 3.1: Esquema del escenario 1 – Interfaz Web

En el segundo escenario, el administrador puede acceder a los datos a través de la terminal. Solo él puede acceder mediante un inicio de sesión. Además, puede apagar o encender el dispositivo y parar o iniciar procesos.

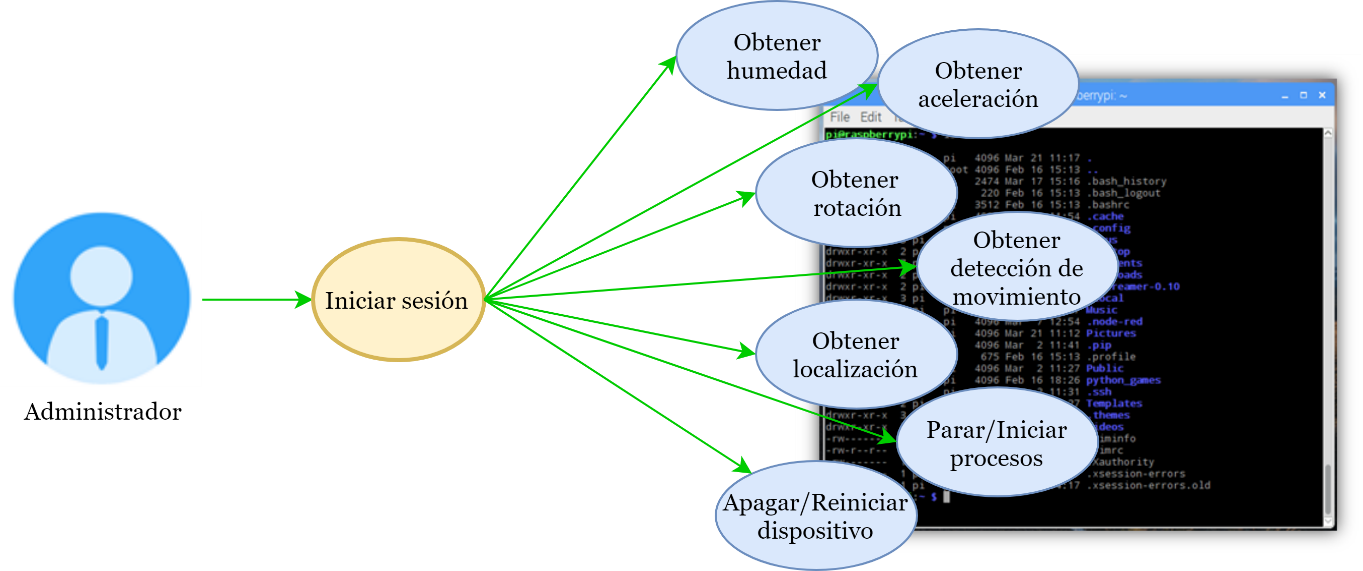


Figura 3.2: Esquema del escenario 2 - Terminal

El tercer escenario representa como interactúan el cliente y el administrador con el servidor. El cliente deberá tener red móvil para recibir SMS y acceso a Internet para recibir correos electrónicos, mientras que el administrador necesitará acceso a Internet para acceder al servidor.

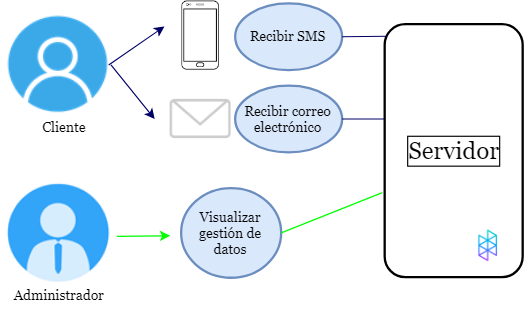


Figura 3.3: Esquema del escenario 3 – Servidor

* + - 1. Descripción de los casos de uso

Los casos de uso se representarán en una tabla con la siguiente estructura:

|  |  |
| --- | --- |
| CdU-EX-XX | |
| Nombre |  |
| Actores |  |
| Descripción |  |
| Precondiciones |  |
| Operación |  |
| Postcondiciones |  |
| Comentarios |  |

Tabla 3.26: Ejemplo de caso de uso

A continuación, se detallan los contenidos de la tabla 3.26:

* **Identificador**: Es el valor que identifica de forma única cada uno de los casos de uso desarrollados. La estructura del identificador mantiene la forma “CdU-EX-XX” ya que se compone de tres partes: “CdU” indica que es un caso de uso, la segunda parte señala el número de escenario descrito y la tercera parte indica el número de caso de uso.
* **Nombre**: es el nombre que describe al identificador.
* **Actores**: indica los usuarios que participan.
* **Descripción**: es la explicación detallada del caso de uso.
* **Precondiciones**: indica los requisitos que se tienen que dar para cumplir con el caso de uso.
* **Operación**: indica los pasos que debe realizar el usuario para cumplir con el caso de uso.
* **Postcondiciones**: consecuencias del sistema en el caso de uso.
* **Comentarios**: posibles valoraciones adicionales acerca del requisito.

1. Caso de uso para el escenario 1 (Interfaz Web)

|  |  |
| --- | --- |
| CdU-E1-01 | |
| Nombre | Inicio de sesión en la página web |
| Actores | Cliente y Administrador |
| Descripción | Para visualizar la información del sensor de humedad en la forma gráfica o descargar la información, es necesario introducir las credenciales. |
| Precondiciones | El usuario debe tener conexión a Internet. |
| Operación | 1. El usuario accede a la página web 2. Inserción de sus credenciales |
| Postcondiciones | El usuario se encontrará en la página principal de canales y podrá seleccionar el canal de la humedad del suelo. |
| Comentarios | Para acceder al canal, es necesario tener permisos del Administrador. |

Tabla 3.27: Escenario 1 – Caso de uso 1

|  |  |
| --- | --- |
| CdU-E1-02 | |
| Nombre | Visualización de valores de la humedad del suelo |
| Actores | Cliente y Administrador. |
| Descripción | El usuario visualiza el valor actual de la humedad del suelo y en un registro a lo largo del tiempo en una gráfica. |
| Precondiciones | El nodo que recoge los valores de la humedad debe estar encendido, junto con la Raspberry Pi como Gateway. El servidor debe funcionar para mandar la información a la interfaz web. El usuario debe tener conexión a Internet y tener permisos por parte del Administrador para acceder al canal. |
| Operación | 1. Se inicia sesión en la página web 2. Se pulsa el botón de canales para seleccionar el canal del sensor de humedad. 3. Visualización de la gráfica |
| Postcondiciones | La página web cambia su estado para mostrar una gráfica con el valor actual de la temperatura. Además, se muestran el resto de los valores almacenados en el registro a lo largo del tiempo. |
| Comentarios |  |

Tabla 3.28: Escenario 1 – Caso de uso 2

|  |  |
| --- | --- |
| CdU-E1-03 | |
| Nombre | Descarga de todos los valores de la humedad |
| Actores | Cliente y Administrador |
| Descripción | El usuario puede descargar todos los valores obtenidos en los formatos JSON, CSV o XML. |
| Precondiciones | El nodo que recoge los valores de la humedad debe estar encendido, junto con la Raspberry Pi como Gateway. El servidor debe funcionar para mandar la información a la interfaz web. El usuario debe tener conexión a Internet y tener permisos por parte del Administrador para acceder al canal. |
| Operación | 1. Se inicia sesión en la página web. 2. Se pulsa el botón de canales para seleccionar el canal del sensor de humedad. 3. Se pulsa el botón de exportar información para seleccionar el formato deseado y proceder a la descarga. |
| Postcondiciones | El usuario obtendrá un fichero en que podrá acceder a los datos del sensor de humedad desde el principio hasta ese momento. |
| Comentarios | Como característica de la página web, se almacenan los datos en el canal, aunque la Raspberry Pi se reinicie o mande nueva información, se mantendrá toda la información previa. |

Tabla 3.29: Escenario 1 – Caso de uso 3

|  |  |
| --- | --- |
| CdU-E1-04 | |
| Nombre | Concesión de permisos a clientes. |
| Actores | Administrador. |
| Descripción | El administrador puede conceder permisos a los clientes para ver o descargar la información de la humedad del suelo ya que esta información no es pública. |
| Precondiciones | El administrador debe mantener conexión a Internet. |
| Operación | 1. Iniciar sesión en la página web. 2. Pulsar “sharing” o compartir. 3. Seleccionar “Share channel view only with the following users” o compartir solo con los siguientes usuarios. 4. Introducir cuenta del cliente. |
| Postcondiciones | El administrador volverá a la página principal. El cliente tendrá los permisos para acceder. |
| Comentarios |  |

Tabla 3.30: Escenario 1 – Caso de uso 4

1. Casos de uso para el escenario 2 (Terminal)

|  |  |
| --- | --- |
| CdU-E2-01 | |
| Nombre | Inicio de sesión en la terminal. |
| Actores | Administrador. |
| Descripción | Es necesario introducir las credenciales del usuario para acceder al sistema completo. |
| Precondiciones | Tanto el usuario como la Raspberry Pi necesitan conexión a Internet. |
| Operación | 1. Introducir usuario y contraseña |
| Postcondiciones | El administrador tendrá acceso total al sistema. |
| Comentarios |  |

Tabla 3.31: Escenario 2 – Caso de uso 1

|  |  |
| --- | --- |
| CdU-E2-02 | |
| Nombre | Obtener información sobre la humedad. |
| Actores | Administrador. |
| Descripción | El usuario puede seleccionar ver la humedad actual en ese momento. |
| Precondiciones | El nodo sensor debe estar funcionando y tanto la Raspberry Pi como el usuario deben mantener la conexión a Internet. Se necesita autenticar al usuario para acceder al sistema. |
| Operación | 1. Iniciar sesión en el programa 2. Pedir información de la humedad |
| Postcondiciones | El usuario obtendrá la información de la humedad en la terminal. |
| Comentarios |  |

Tabla 3.32: Escenario 2 – Caso de uso 2

|  |  |
| --- | --- |
| CdU-E2-03 | |
| Nombre | Obtener información sobre la aceleración. |
| Actores | Administrador. |
| Descripción | El usuario puede seleccionar ver la aceleración actual en los tres ejes del nodo sensor en ese momento. |
| Precondiciones | El nodo sensor debe estar funcionando y tanto la Raspberry Pi como el usuario deben mantener la conexión a Internet. Se necesita autenticar al usuario para acceder al sistema. |
| Operación | 1. Iniciar sesión en el programa 2. Pedir información de la aceleración |
| Postcondiciones | El usuario obtendrá la información de la aceleración en la terminal. |
| Comentarios |  |

Tabla 3.33: Escenario 2 – Caso de uso 3

|  |  |
| --- | --- |
| CdU-E2-04 | |
| Nombre | Obtener información sobre la rotación. |
| Actores | Administrador. |
| Descripción | El usuario puede seleccionar ver la rotación actual en los ejes X e Y del nodo sensor en ese momento. |
| Precondiciones | El nodo sensor debe estar funcionando y tanto la Raspberry Pi como el usuario deben mantener la conexión a Internet. Se necesita autenticar al usuario para acceder al sistema. |
| Operación | 1. Iniciar sesión en el programa 2. Pedir información de la rotación |
| Postcondiciones | El usuario obtendrá la información de la rotación en la terminal. |
| Comentarios |  |

Tabla 3.34: Escenario 2 – Caso de uso 4

|  |  |
| --- | --- |
| CdU-E2-05 | |
| Nombre | Obtener información sobre cuerpos térmicos en movimiento. |
| Actores | Administrador. |
| Descripción | El usuario puede ver si se han detectado cuerpos térmicos en movimiento por la zona y saber cuántas veces. |
| Precondiciones | El nodo sensor debe estar funcionando y tanto la Raspberry Pi como el usuario deben mantener la conexión a Internet. Se necesita autenticar al usuario para acceder al sistema. |
| Operación | 1. Iniciar sesión en el programa 2. Pedir información sobre los cuerpos en movimiento |
| Postcondiciones | El usuario obtendrá la información del número de cuerpos en la terminal. |
| Comentarios |  |

Tabla 3.35: Escenario 2 – Caso de uso 5

|  |  |
| --- | --- |
| CdU-E2-06 | |
| Nombre | Obtener información de la localización del dispositivo. |
| Actores | Administrador. |
| Descripción | El administrador podrá ver donde se encuentra el dispositivo en todo momento recibiendo las coordenadas de éste. |
| Precondiciones | Tanto la Raspberry Pi como el usuario deben mantener la conexión a Internet. Se necesita autenticar al usuario para acceder al sistema. |
| Operación | 1. Iniciar sesión en el programa 2. Pedir información de la localización |
| Postcondiciones | El usuario podrá ver en la terminal los valores de la latitud y la longitud. |
| Comentarios |  |

Tabla 3.36: Escenario 2 – Caso de uso 6

|  |  |
| --- | --- |
| CdU-E2-07 | |
| Nombre | Apagar o reiniciar el dispositivo. |
| Actores | Administrador. |
| Descripción | El administrador podrá apagar o reiniciar el dispositivo para labores de mantenimiento o mejora del sistema. |
| Precondiciones | Tanto la Raspberry Pi como el usuario deben mantener la conexión a Internet. Se necesita autenticar al usuario para acceder al sistema. |
| Operación | 1. Iniciar sesión en el programa 2. Escribir “sudo shutdown -h now” en caso de apagar o “sudo reboot” en caso de reiniciar. |
| Postcondiciones | El usuario podrá ver si ha reiniciado o se ha apagado. |
| Comentarios |  |

Tabla 3.37: Escenario 2 – Caso de uso 7

|  |  |
| --- | --- |
| CdU-E2-08 | |
| Nombre | Parar y/o iniciar procesos. |
| Actores | Administrador. |
| Descripción | El administrador podrá parar y/o iniciar procesos por temas de mantenimiento o mejora en el sistema. |
| Precondiciones | Tanto la Raspberry Pi como el usuario deben mantener la conexión a Internet. Se necesita autenticar al usuario para acceder al sistema. |
| Operación | 1. Iniciar sesión en el programa 2. Buscar el identificador del proceso concreto 3. Escribir “sudo kill <Identificador>” |
| Postcondiciones | El proceso se habrá parado o iniciado. |
| Comentarios |  |

Tabla 3.38: Escenario 2 – Caso de uso 8

1. Casos de uso para el escenario 3 (Servidor)

|  |  |
| --- | --- |
| CdU-E3-01 | |
| Nombre | Recibir alertas SMS. |
| Actores | Cliente. |
| Descripción | El cliente recibirá una alerta del servidor en forma de SMS si se cumplen los requisitos. Además, recibirá en el mismo mensaje las coordenadas del nodo de localización. |
| Precondiciones | Los nodos sensores deben estar funcionando y la Raspberry Pi debe mantener la conexión a Internet y el servidor debe tener saldo para poder enviar un SMS al cliente. |
| Operación | El usuario debe esperar al mensaje. |
| Postcondiciones | El usuario podrá ver un nuevo mensaje en su teléfono móvil y acceder a la localización. |
| Comentarios | Las alertas pueden deberse por detección de cuerpo térmico o movimiento indebido del motor de gasolina. |

Tabla 3.39: Escenario 3 – Caso de uso 1

|  |  |
| --- | --- |
| CdU-E3-02 | |
| Nombre | Recibir alertas vía correo electrónico. |
| Actores | Cliente. |
| Descripción | El cliente recibirá una alerta del servidor en forma de correo electrónico si se cumplen los requisitos. Además, recibirá en el mismo mensaje las coordenadas del nodo de localización. |
| Precondiciones | Los nodos sensores deben estar funcionando y tanto la Raspberry Pi como el usuario deben mantener la conexión a Internet. |
| Operación | 1. Iniciar sesión en la cuenta de correo electrónico 2. Abrir “bandeja de entrada” |
| Postcondiciones | El usuario podrá ver un nuevo mensaje en su bandeja de entrada. |
| Comentarios | Las alertas pueden deberse por detección de cuerpo térmico o movimiento indebido del motor de gasolina. |

Tabla 3.40: Escenario 3 – Caso de uso 2

|  |  |
| --- | --- |
| CdU-E3-03 | |
| Nombre | Gestión de datos. |
| Actores | Administrador. |
| Descripción | El administrador podrá ver el consumo de datos móviles de la Raspberry Pi en una gráfica y en un formato numérico. |
| Precondiciones | El usuario debe tener conexión a Internet para acceder al consumo de datos móviles. |
| Operación | 1. Iniciar sesión en la página del servidor 2. Pulsar “Dashboard” o panel de control. 3. Clicar en la tarjeta SIM que se está usando en la Raspberry Pi. 4. Clicar en “Usage” o Uso |
| Postcondiciones | El usuario podrá visualizar el consumo de datos totales de manera numérica y en una gráfica. |
| Comentarios |  |

Tabla 3.41: Escenario 3 – Caso de uso 3

* 1. Arquitectura del sistema

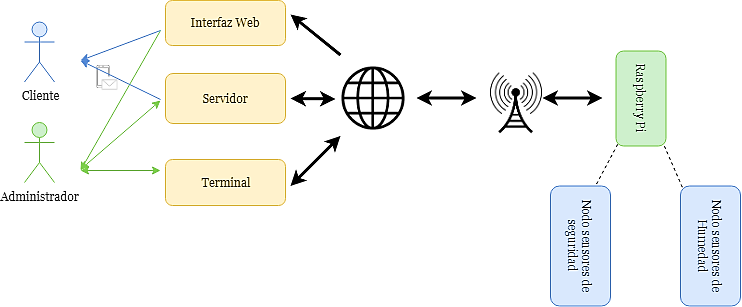
A partir del análisis realizado en el apartado anterior, se saben qué objetivos y requisitos se deben cumplir. A continuación, se describe en un modo lógico cómo debería ser la arquitectura del sistema:

Figura 3.4: Arquitectura ideal del sistema

\*Nota importante: hacer la flecha bidireccional al cliente con el servidor ya que el cliente debe tener poder (de alguna manera) para poder encender o apagar el dispositivo de seguridad\* Sea mandando un sms a la raspberry o un mensaje whatever.

Como se puede ver en la figura **3.X**, se pueden distinguir varias partes. En primer lugar, en la parte de la derecha se encuentra la Rasberry Pi junto con los nodos que recogen la información del entorno. En segundo lugar, en la parte central, la Rasberry Pi envía la información a Internet a través de las torres GSM. En último lugar, una vez llega a Internet, la información se deposita en los diferentes escenarios en los cuales el administrador y el cliente tienen diferentes roles.

Como ya se explicó en el punto anterior, se listan tres escenarios: la Interfaz Web, en los cuales el cliente y el administrador pueden visualizar a la información de la humedad en un formato gráfico y descargar la misma en diferentes formatos; el servidor, en que los usuarios tienen diferentes operaciones y la terminal, en el que solo el administrador puede entrar para comprobar los diferentes parámetros del sistema a larga distancia.

Además, se puede ver en el gráfico flechas unidireccionales y bidireccionales. Esto significa que la información podrá viajar en un solo sentido (unidireccional) como la flecha de la interfaz web (recepción de información y no envío) o en ambos (bidireccional) como el administrador y la terminal ya que cuando el administrador pide información, la recibe. Las flechas también están coloreadas en función del usuario: verde para el administrador y azul para el cliente.

**Capítulo 4**

Implementación

En este capítulo se explicará paso a paso cómo se ha desarrollado todo el proyecto de acuerdo con los objetivos presentados en el capítulo 1 y los requisitos del sistema, además de la arquitectura del sistema. Se decidió crear dos programas: uno para controlar automáticamente el riego y mandar información a la interfaz web y otro para monitorear el sistema de seguridad. Por otro lado, se mostrará cómo el administrador se comunicará con el sistema en un tercer modo de comunicación. Una vez todo junto, se simulará y se hará una prueba en la realidad.

* 1. Contexto

La vid necesita más o menos agua dependiendo de la estación del año en la que se encuentre. Como podrá observar en el ***Anexo 2***, además, se necesita recoger la cantidad óptima de agua subterránea para tener una recolección eficiente de la uva y para evitar agotar el acuífero. Para ello, también hay que tener en cuenta la lluvia útil en las diferentes estaciones.

Por otro lado, también es necesario observar el estado del motor de gasoil y alertar al usuario cuándo se está produciendo un robo del motor ya que supone una inversión económica considerable para el agricultor.

* 1. Programa de riego – riego.py

En este programa, se tienen que respetar los siguientes objetivos: obtener el valor de humedad en el momento, enviarlo a la plataforma web para su visualización en una gráfica, encender y apagar el motor de gasoil y hacer una copia de los parámetros utilizados en el programa en un archivo log. Esta información se resume en el siguiente diagrama de flujo:

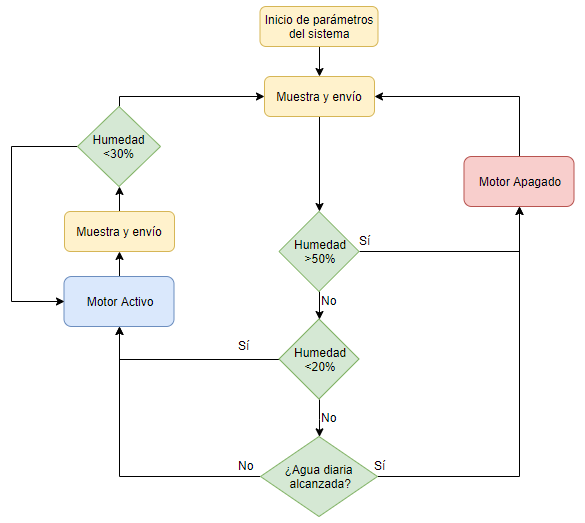


Figura 4.1: Diagrama de flujo – riego.py

Junto con riego.py existe otro archivo auxiliar llamado mcp3008.py que se encarga de recoger la información del sensor de humedad gracias al convertidor analógico – digital.

* + 1. Análisis de la humedad

Habrá dos requerimientos fundamentales a la hora de activar el motor o no: La humedad del suelo y la cantidad de agua mínima necesaria. Ciertamente, que haya agua implica que habrá humedad, pero en las épocas de floración, envero y maduración se necesita más agua para que vaya directamente al fruto y este crezca, además de un nivel de humedad mínimo asegurará que la planta no se seque (Más información en el anexo 2).

En primer lugar, si los niveles de humedad crecen considerablemente (más de un 50%), entonces no es necesario aplicar el riego, ya que tampoco es recomendable y con ese nivel la planta puede desarrollar el fruto correctamente. Ahora bien, en segundo lugar, si el nivel de humedad es inferior al 20%, entonces no se tiene en cuenta el nivel de agua alcanzada hasta que se alcance el 30% de humedad (De esta manera el nivel no está justo en el límite inferior).

Hay una tercera condición: el nivel de humedad que se encuentre entre el 20% y el 50%. En este caso, no se regará más si se ha alcanzado el agua disponible diaria y de lo contrario, regará hasta que llegue al límite del 50% o se haya agotado el agua destinada para ese día.

Al ser una implementación, se programará un led azul que actúa de motor de gasoil. Se encenderá para simular que el motor se enciende y se apagará en caso contrario.

* + 1. Cálculo del agua alcanzada

De una manera orientativa, para calcular la cantidad de agua que se está regando por metro cuadrado o agua alcanzada, se utilizará la información que tiene el motor de gasoil que se presentó en la figura 1.1 [<https://www.greenpower.lk/katalog/artikel/greenpower-deutz-diesel-power-generator-30kva-24kw-open-frame-manual-starting/116586-0-50/>]. Este motor de gasoil es capaz de alimentar a una bomba que extrae 8300 litros por hectárea y hora. Eso quiere decir:

Si el motor funciona una hora exactamente, se tendrá 0.83 litros en un metro cuadrado. Entonces para calcular el agua alcanzada, se aplicaría el siguiente cálculo:

En riego.py, el tiempo activo crece cuando se cumplen las condiciones de extracción. Como se leen datos cada 15 minutos, cada ciclo del “while” sumará 15 minutos al tiempo parado o al tiempo activo. En el programa se utilizará segundos como medida de tiempo, por lo tanto, ciclo\_15 será igual a 900 y el ritmo a 2.3055·10-4.

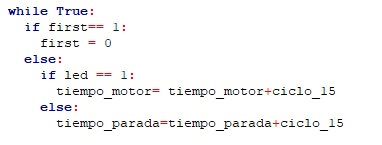


Figura 4.2: Tiempo motor (activo) y tiempo parada cada 15 minutos

* + 1. Cálculo de agua diaria disponible

De acuerdo con los cálculos realizados en el anexo 2, es necesario tener en cuenta las precipitaciones del año ya que cuanto más llueva en la zona, menos agua será necesaria para el riego. De un modo orientativo, se calculó la lluvia útil que caía en Corral de Almaguer en el año 2017 y se restó al agua necesaria en la vid por metro cuadrado. De esta manera se tenía el agua que era necesario en un mes, por lo tanto, para conseguir el agua necesaria diaria, se debía dividir el agua mensual por el número de días en cada mes (Resultado en la ***tabla X***).

* + 1. Envío de datos a ThingSpeak

A medida que el sensor recoja los datos de la humedad, el programa los irá enviando automáticamente a la página de ThingSpeak. Para ello, en riego.py, primero, se conectará a la plataforma para que se permita el envío de datos. Se escribirá la clave de escritura API y se enlazará con el URL de la página web:



Figura 4.2: Configuración de los parámetros para acceder a ThingSpeak

Para enviar la información, se necesitará realizar un POST request de la siguiente manera:

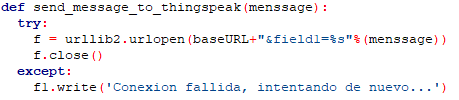


Figura 4.3: Función para enviar mensaje a la plataforma

Como se puede ver en la función “urlopen”, se necesita la dirección URL de la figura 4.2 junto con la clave de escritura. Además, para mandar la información es necesario que se elija qué gráfica mostrará los datos. En ThingSpeak, se pueden crear hasta ocho gráficas, pero como se está mostrando solo la humedad, se envía a la primera gráfica llamada “field1”.

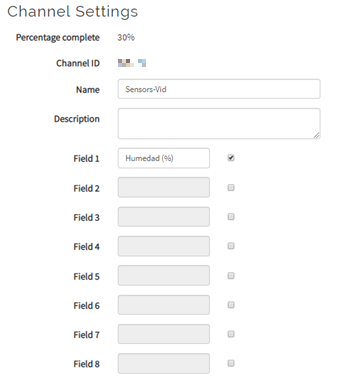


Figura 4.4: Configuración del canal en la página web. La humedad se puede visualizar en el primer gráfico (Field 1)

Una vez el programa está funcionando y los datos vayan llegando a la plataforma, se podrá ver un resultado como en la siguiente figura:

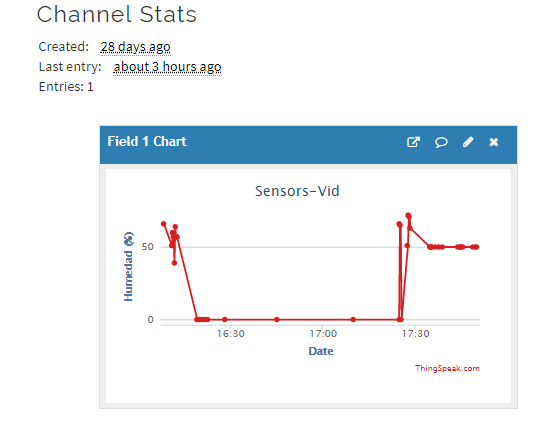


Figura 4.5: Gráfica de la humedad en tiempo real

A su vez, si el usuario estuviese interesado en conseguir la información “en crudo”, puede hacerlo descargando el fichero en los formatos JSON, XML o CSV. Para ello, debe clicar en exportar los datos recientes (*export recent data*) en la página principal del canal seleccionado.

* + 1. Creación del fichero “log\_riego.txt”

En el caso de que ocurriese algún problema con la transmisión de los datos o se desease ver qué movimientos ocurrieron en el pasado, existe un archivo llamado “log\_riego.txt” que se almacenará en la Raspberry Pi.

En este archivo, la información se insertará en el formato: <fecha> <agua utilizada> <Humedad> <tiempo motor> <tiempo parado>. El administrador podrá acceder al fichero utilizando la terminal (punto 4.4).

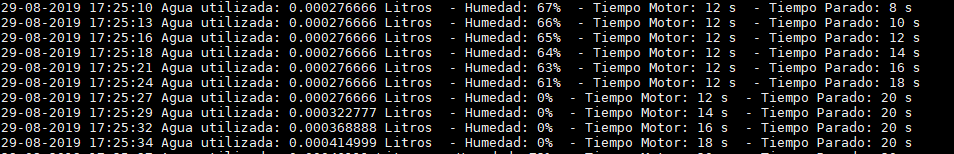


Figura 4.X: Ejemplo de “log\_riego.txt”

* 1. Programa de seguridad – seguridad.py

En este programa, se valorará cuándo se está produciendo un robo o no. Para ello se comprobarán los siguientes parámetros:

* Existencia de un cuerpo térmico (persona) en movimiento en la zona.
* Movimiento del motor (velocidad y rotación)

En el caso de que se detectase una actividad sospechosa, se realizará un envío de una alerta al cliente explicando cómo se ha detectado y acto seguido, se insertará un enlace a Google Maps con la localización en latitud y longitud obtenido por el módem *GSM*. Esta información se puede visualizar en el siguiente diagrama de flujo:

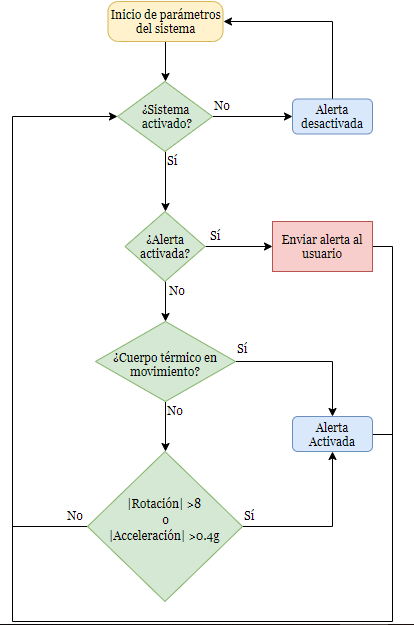


Figura 4.X: Esquema de funcionamiento – seguridad.py

* + 1. Sistema de activación y desactivación del programa

Una de las cosas más importantes que debe tener el sistema es la activación y la desactivación de la seguridad ya que el cliente debe ir a recargar el depósito de gasoil para que la máquina siga funcionando. Si no existiese una forma de paralizar la seguridad mientras el agricultor está llenando el depósito, recibiría muchos mensajes de alerta.

Para ello, se crea un programa llamado dueno.py (Se puede cambiar esto!!) que contiene una única función auxiliar para su uso en seguridad.py. Esta función mira si hay un SMS nuevo en la memoria y proporciona una respuesta que depende del SMS:

* Si no existe SMS, la respuesta será ‘U’ de *unknown* o desconocido.
* Si existe SMS y su mensaje es ‘D’ o ‘d’, la respuesta será ‘D’ o desactivado.
* Si existe SMS y su mensaje es ‘A’, la respuesta será ‘A’ o activado.
* Si existe SMS y su mensaje no es ni ‘D’ ni ‘A’ (o en minúscula), la respuesta será ‘U’ de *unknown* o desconocido.

En el programa principal, seguridad.py, habrá un bucle infinito que siempre llamará a esta función en primer lugar. Por defecto cuando empieza a funcionar el programa, el sistema está activo y está constantemente vigilando todos los parámetros cada 2 segundos. Sin embargo, cuando se desactiva, el sistema vuelve a llamar a la función de activación cada 15 minutos por si hay un mensaje de activación.

Hologram ofrece dos opciones para enviar SMS al dispositivo: que los envíe el administrador o que los envíe el cliente. En el primer caso, el cliente tendría que especificar al administrador cuándo debería enviar un SMS para parar y otro para activar el programa de seguridad. Esto es un poco laborioso, pero el administrador en el servicio de la nube de Hologram puede enviar mensajes gratuitos al dispositivo. En el segundo caso, para que el cliente los pueda enviar sin tener que llamar al administrador, tendría que adquirir un número de teléfono para la tarjeta SIM Hologram. En este caso, el administrador contrataría un número de teléfono por 1 dólar al mes y el cliente podría enviar todos los SMS a este número.

* + 1. Detección de cuerpos térmicos en movimiento

Para poder variación de movimiento en los cuerpos térmicos de alrededor es necesario utilizar el sensor *PIR*. Como ya se habló en el capítulo 1, el *PIR* es un sensor que detecta variaciones en las radiaciones infrarrojas.

Sin embargo, existe un problema con este sensor: puede detectar cualquier variación de radiación infrarroja, es decir, puede detectar una persona o un animal. Esto supone otro problema porque pueden ocurrir falsas alarmas.

Para ello, se tiene que aprovechar la altura en la que se coloca el sensor ya que el *PIR* establece una línea de detección con las lentes del propio sensor. Si un animal está debajo de esa línea, no será detectado. Por lo tanto, sabiendo que los animales de la zona son pequeños, la línea de detección puede estar a un metro y medio de altura.

Así pues, en seguridad.py, para comprobar el funcionamiento del *PIR*, se comprueba si se recibe alguna señal en el pin elegido para el sensor. En la figura 4.X, se incluye la fecha y la hora cuando se recibe una señal positiva.

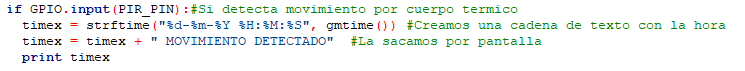


Figura 4.X: Sentencia condicional con el sensor *PIR*.

* + 1. Análisis del movimiento angular y lineal del motor

Esta herramienta es un recurso adicional para la seguridad del motor. Puede ser como una segunda medida de seguridad si el *PIR* no llegase a detectar un cuerpo térmico por variación de radiaciones infrarrojas en la zona.

Para ello, el sensor que se utiliza es el MPU6050. Este sensor se colocará en una superficie plana en el motor y analizará la rotación y la aceleración de este. En condiciones normales, el motor se encuentra en una posición estática, es decir, su movimiento angular (rotación) y la velocidad lineal no varían en ningún momento. En el momento en el que el dueño u otra persona quisiera mover el motor, habría variaciones en estos parámetros.

Para ello, se han hecho pruebas para ver el funcionamiento del sensor sin ningún movimiento. En la figura ***4.X*** se encuentran los resultados numéricos mientras que en la figura ***4.(X+1)*** se puede ver que el sensor está sobre una superficie plana (una mesa):

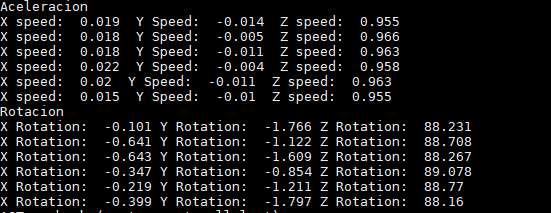


Figura 4.0X: Pruebas del sensor MPU6050 (Aceleración y Rotación)

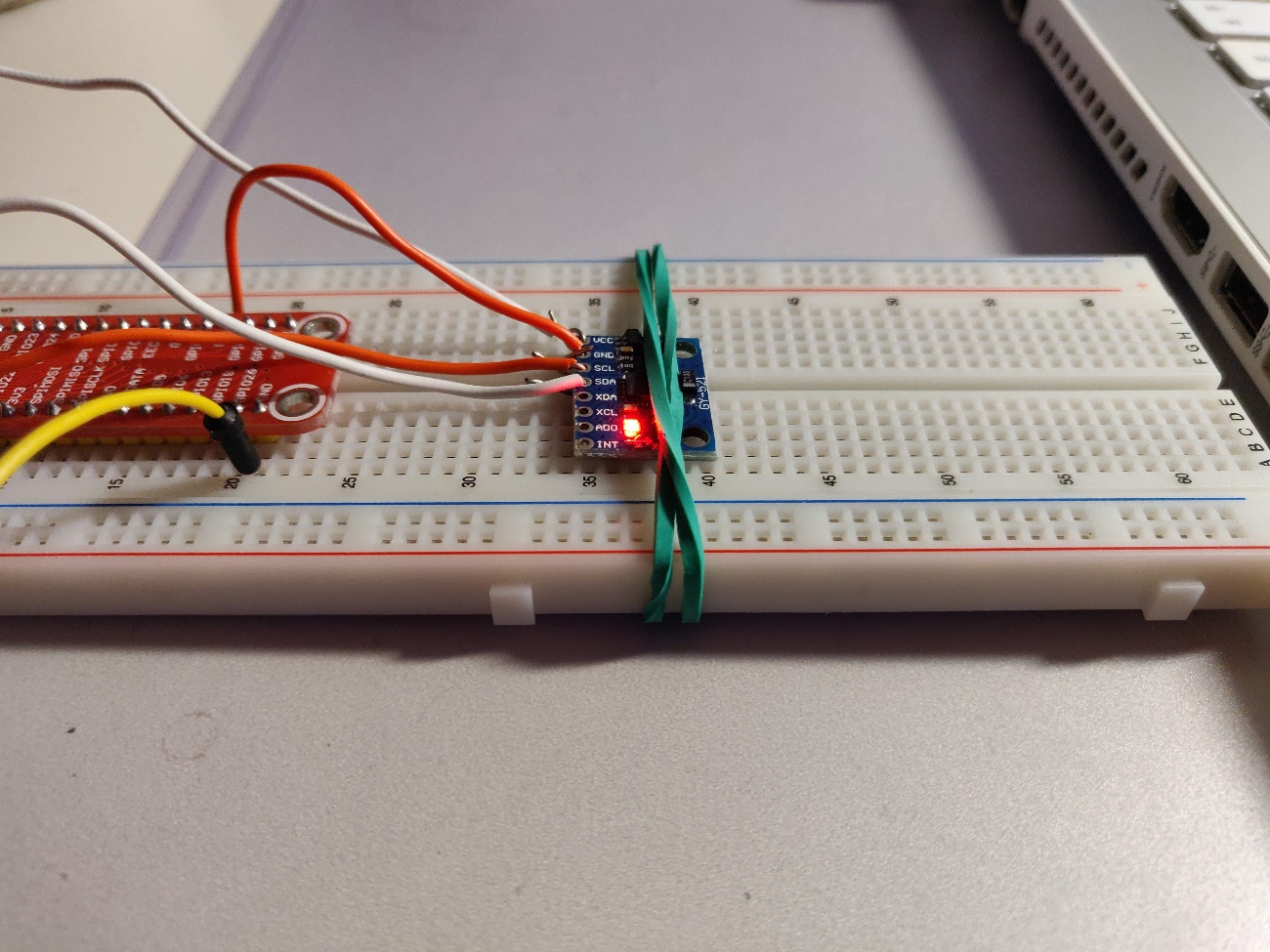


Figura 4.(X+1): Posición del sensor MPU6050

El valor de la aceleración se mide en unidades “g”, es decir, 9.8 m/s2. Esta unidad corresponde con la gravedad terrestre. Sobre la superficie plana, el eje z tendrá toda la aceleración de la gravedad. Entonces, los ejes X e Y determinarán la aceleración lineal:

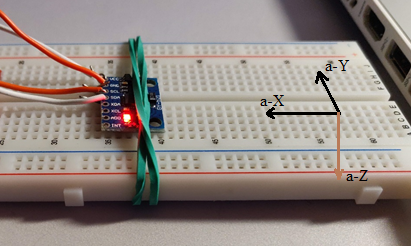
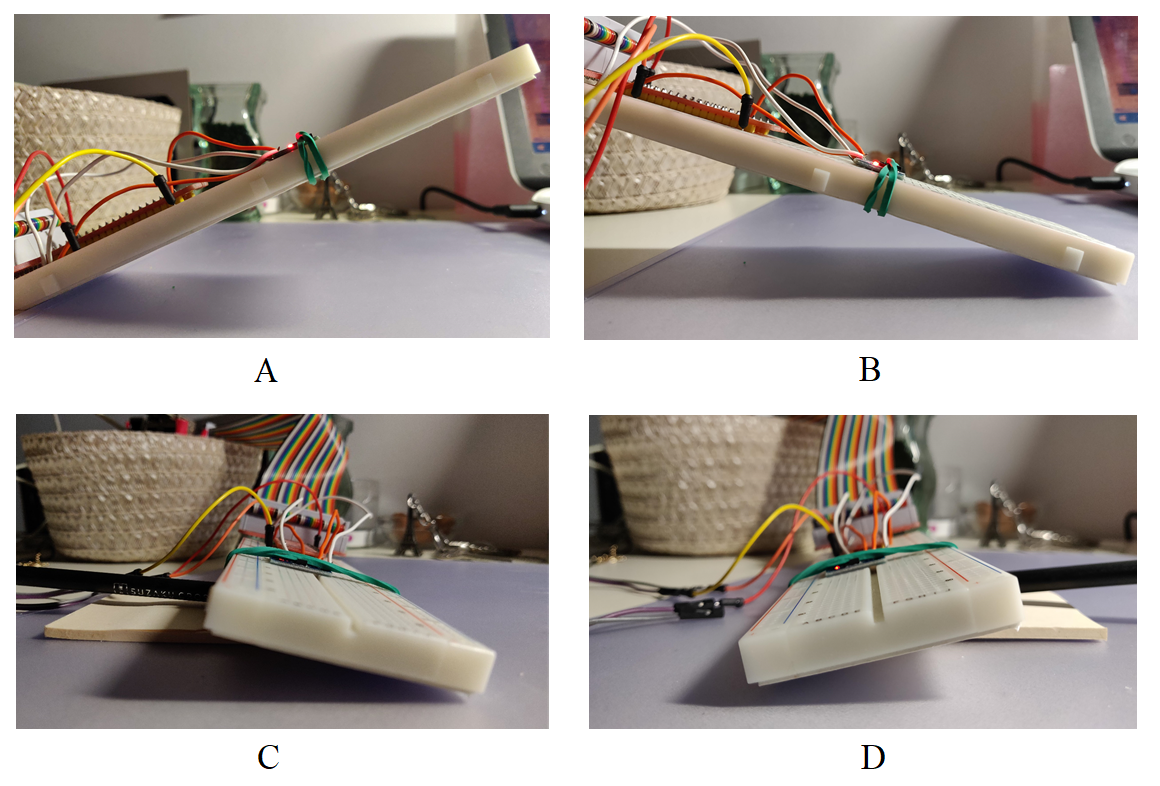


Figura 4.X: Vista de la aceleración en los diferentes ejes y el MPU6050

Como los resultados en los ejes X e Y dan valores en las centésimas en el estado inmóvil (virtualmente 0 g), el valor que determine movimiento sospechoso será mayor a 0.4g. Asumiendo que un vehículo que arrastre el motor logre 20 km/h en 3 segundos, equivaldría a acelerar a 6.66 m/s2 o 0.68g. Por lo tanto, para agudizar más el proceso, se optará por 0.4g en los ejes X e Y como valor máximo para activar la alarma.

Por otro lado, la rotación se mide en grados. Los valores importantes son de nuevo los ejes X e Y, ya que el eje Z depende de otro giroscopio para establecer la referencia. Sin embargo, no es necesario usar el eje Z ya que con los dos primeros sería suficiente. A continuación, en la siguiente figura se puede ver cómo aumenta la rotación en los diferentes ejes:

Figura 4.X: Diferentes posiciones del giroscopio

Donde:

* Imagen A: Se avanza en el eje Y negativo.
* Imagen B: Se avanza en el eje Y positivo.
* Imagen C: Se avanza en el eje X negativo.
* Imagen D: Se avanza en el eje X positivo.

Esto será útil para comprobar que está en movimiento ya que la superficie en la que se encuentra el motor es irregular y pedregoso. Por lo tanto, es fácil que el motor sobre un remolque rebote lo suficiente para que salte el sensor. Como se puede ver en los resultados de la figura ***4.0X***, la rotación llega a variar hasta 0.6 grados en un estado inmóvil. Por lo tanto, se establecerá que el límite máximo para no alertar al usuario será de 8 grados.

Figura 4.X: Sentencia condicional con sensor MPU6050

* + 1. Localización del dispositivo

En el caso de que salten las alarmas (sea por el *PIR* o el movimiento angular y lineal) se procedería a enviar la localización del motor cada 3 minutos.

Para ello, es necesario usar el módem *Hologram Nova*. Este dispositivo está conectado a uno de los USB de la Raspberry Pi:

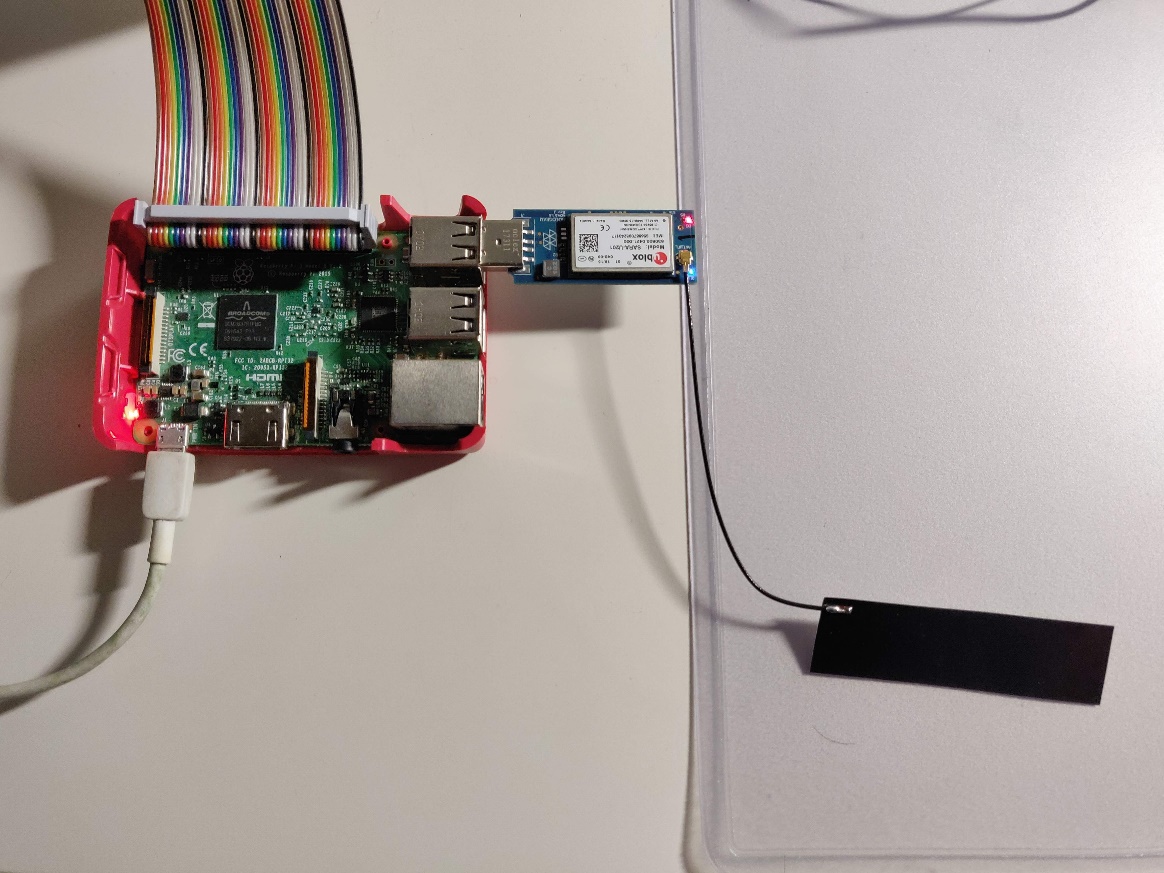


Figura 4.X: Raspberry Pi conectado al módem Hologram Nova

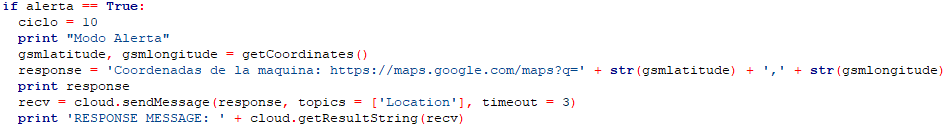
De esta forma, en el código se llamaría a la función get\_coordinates() que conseguiría los parámetros del módem: “network.location.latitude” y “network.location.longitude”. Considerando que se usa GSM para la localización, las coordenadas no son exactas, pero son aproximadas. Se usa Google Maps para insertar automáticamente las coordenadas para su posterior envío al usuario.

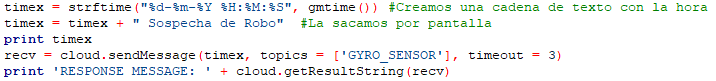
Figura 4.X: Sentencia condicional para enviar la localización

* + 1. Envío de alertas al usuario

El usuario recibirá las alertas de dos maneras diferentes: vía *SMS* y correo electrónico. Para ello, se utilizará el módem *Hologram Nova* para tener conexión a Internet y enviar las alertas a la plataforma de *Hologram Cloud* (Servidor).

Para ello, en el código se escribe el mensaje que se quiere enviar a la nube y se envía junto con una etiqueta o tema (*topic*). Esto ayuda a distinguir en la nube si la alerta es de tipo *PIR*, giroscopio o localización ya que usa el mecanismo de rutas para rebotar el mensaje de la nube al usuario mediante *SMS* o correo electrónico. Por ejemplo, en la siguiente figura se puede ver cómo se configuran las alertas del *PIR*, del giroscopio y de la localización.





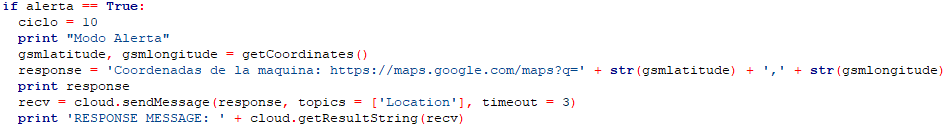


Figura 4.X: Envío de mensaje con la etiqueta PIR\_SENSOR, GYRO\_SENSOR y Location

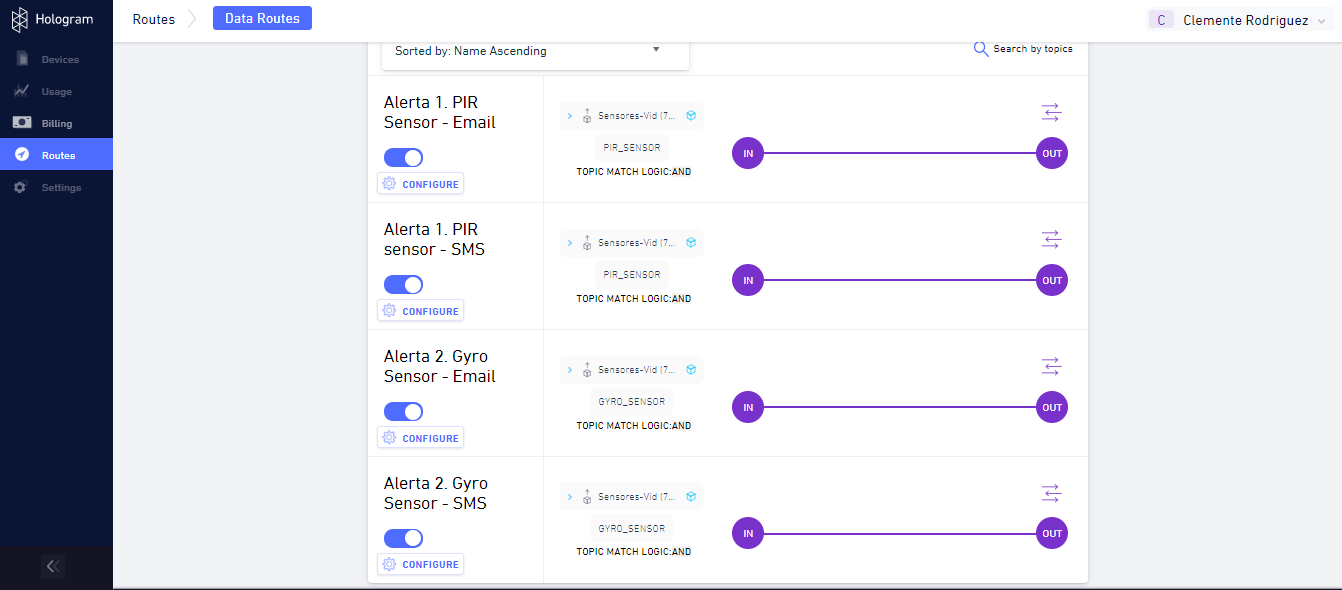
Cuando el mensaje llega a la nube, necesita ser enrutado a otro lugar ya que es necesario que el usuario sea consciente de la alerta. En la siguiente figura se puede observar que si llega un mensaje del módem conectado a la Raspberry Pi con el tema o etiqueta “PIR\_SENSOR” o “GYRO\_SENSOR”, entonces realiza la instrucción correspondiente:

Figura 4.X: Página de rutas en Hologram Cloud

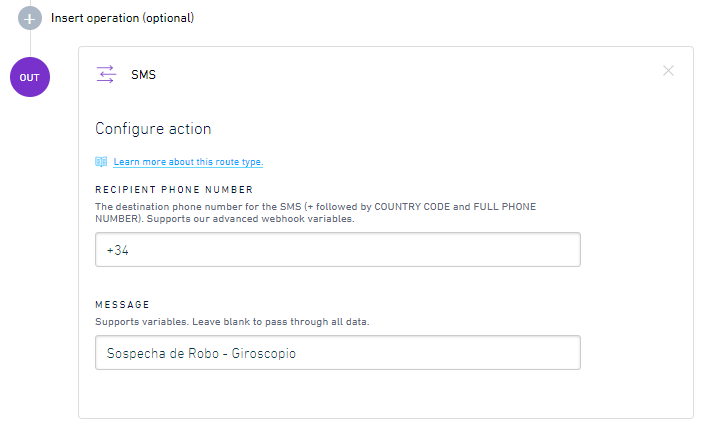


Figura 4.X: Ejemplo de configuración de ruta SMS

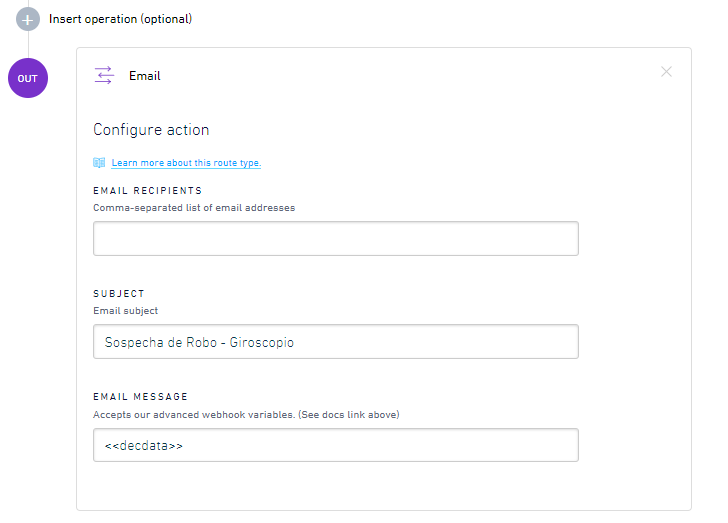


Figura 4.X: Ejemplo de configuración de ruta de correo electrónico

En este proyecto, existen 5 rutas en total: dos para las alertas del *PIR* y dos para las alertas del giroscopio (*SMS* y correo electrónico) y una para la localización (*SMS*). Las rutas se configuran de la misma manera, aunque las etiquetas son diferentes.

* + 1. Inicio de parámetros y librerías

En el programa de seguridad, se llaman a diferentes librerías y programas auxiliares. Como se ha comentado anteriormente, los programas auxiliares son dueno.py y gyro.py. En cuanto a las librerías, se usan las siguientes:

* **Time**: sirve para parar el programa con la función sleep() y conseguir la fecha y hora del momento y en la zona horaria correspondiente con localtime() y strftime().
* **Hologram.HologramCloud**: útil en el uso del módem para enviar y recibir las notificaciones a la nube.
* **RPi.GPIO**: para el uso de los pines de la Raspberry Pi.
* **Smbus**: para vincular el giroscopio en el programa.
* **Math**: para realizar cálculos y operaciones trigonométricas con los datos obtenidos en el giroscopio.
  + 1. Creación de archivos “log\_seguridad.txt” y “alertas.txt”

En el caso de que ocurriese algún problema con la transmisión de los datos o se desease ver qué movimientos ocurrieron en el pasado, existen dos archivos que contienen la información en la Raspberry Pi:

* **Log\_seguridad.txt**: contiene toda la información que aporta el programa de seguridad. Todos los datos están acompañados de la fecha en la que se captaron. En el caso del giroscopio, mantiene el formato <fecha> <rotación> <acelereración>, mientras que el *PIR* tiene <fecha> <Movimiento detectado>. Por otro lado, también tendrá la información cuando se activa y desactiva el sistema de seguridad y cuándo está en alerta.
* **Alertas.txt**: contiene solo las alertas ocasionadas por el *PIR* y el giroscopio. En el caso del *PIR*, el formato es el mismo que en log.txt, mientras que el giroscopio tiene <fecha> <Sospecha de robo – giroscopio>.

Figura 4.3.X: Ejemplo log\_seguridad.txt

Figura 4.3.(X+1): Ejemplo alertas.txt

* 1. Terminal

En este apartado, el administrador podrá visualizar todas las opciones de monitorización del sistema de sensores a través de la terminal. Para ello, se realizará una conexión *SSH* para obtener toda la información que aporta la Raspberry Pi ya que está conectada en todo momento a Internet.

Al ser una conexión *SSH*, se tiene acceso total a la Raspberry Pi, desde parar o reiniciar los programas de riego y seguridad, a gestionar los recursos de la Raspberry Pi. Esto a la vez tiene un riesgo, si un usuario que no es el administrador logra introducirse en la Raspberry Pi puede controlarlo todo, hasta eliminar archivos y perder información privilegiada.

Sin embargo, realizar la conexión *SSH* no es una tarea trivial ya que el sistema se encuentra conectado a un módem USB y no es posible obtener una IP pública para acceder directamente. No obstante, gracias al servicio de Hologram, *Spacebridge*, es posible realizar dicha conexión. Consiste en crear un túnel seguro para enviar datos entre el módem *USB* y el ordenador que esté autentificado.

Para ello, el administrador descargarse la aplicación de *Spacebridge Client* y activar el servicio de *Spacebridge* seleccionando la SIM de la Raspberry Pi en la nube de Hologram:

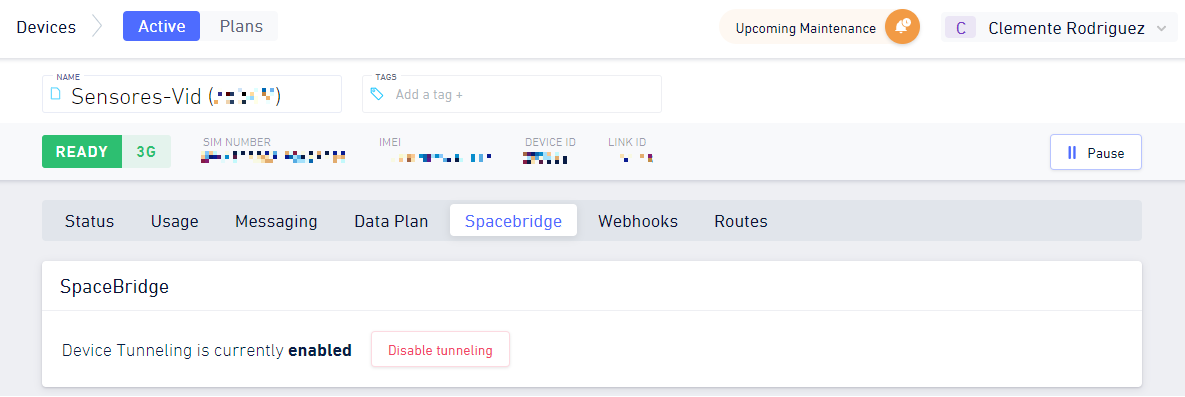


Figura 4.4.X: Activación de *Spacebridge* en la SIM seleccionada

En el ordenador, es necesario iniciar sesión con la clave de acceso de la cuenta de Hologram:

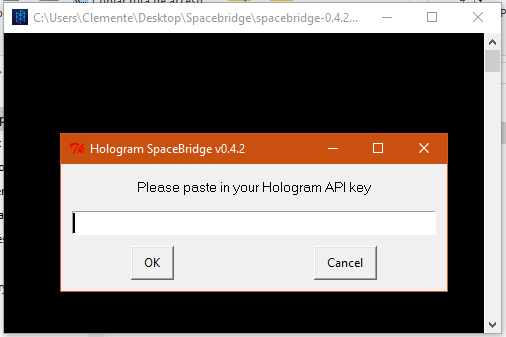


Figura 4.4.X: Incio del programa *Spacebridge*

Una vez dentro, el usuario deberá seleccionar la SIM que tenga el servicio de *Spacebridge* activado e indicar a qué puertos estará conectado el túnel seguro. En este caso, al ser una futura conexión *SSH*, el puerto de la Raspberry Pi será el 22 y el puerto local puede ser uno cualquiera (más de 1023). En este caso, se ha elegido el puerto 12345:

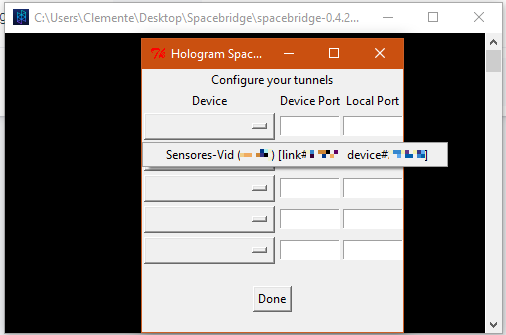


Figura 4.4.X: Selección de SIM y puertos

Si todo el proceso se ha realizado correctamente, debe aparecer una imagen como la siguiente figura. De esta manera, el túnel se ha creado:

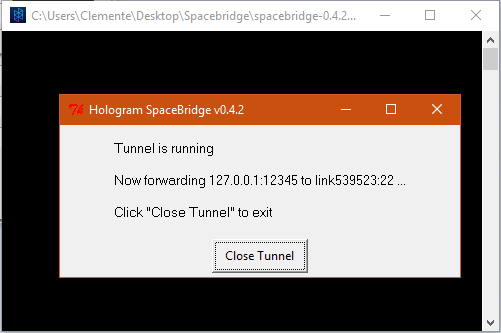


Figura 4.4.X: Túnel creado

Ahora, el usuario puede acceder por *SSH* a la Raspberry Pi. En este caso, se ha usado MobaXterm para acceder. Los parámetros son la IP del túnel, 127.0.0.1, y el puerto 12345. Una vez dentro, el administrador debe introducir usuario y contraseña de la Raspberry Pi. Nótese que es necesario cambiar el usuario y contraseña por defecto que tiene la Raspberry Pi para iniciar sesión.

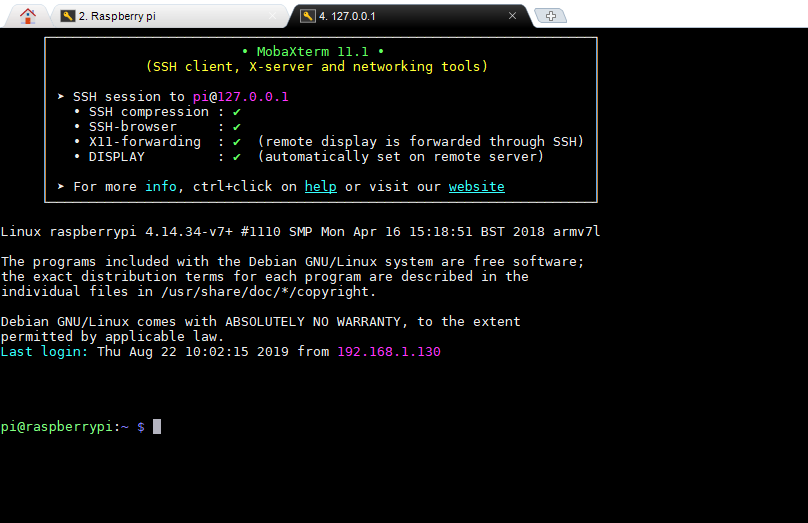


Figura 4.4.X: Acceso por *SSH*

Sin embargo, existe un problema con esta conexión. Es un poco más lenta ya que tiene conexión 3G, más lenta que si se hiciese con WiFi. Una ventaja que se consigue realizando el túnel y haciendo la conexión por SSH es que se consigue una doble seguridad. En primer lugar, para crear el túnel, primero se debe conocer la clave y que se encuentra en la página de Hologram (necesario iniciar sesión en la página). En segundo lugar, una vez creado el túnel, es necesario iniciar sesión en la Raspberry Pi y todas las comunicaciones están protegidas por SSH.

* + 1. Comandos

Ya que se dispone de una conexión *SSH*, se puede controlar la Raspberry Pi como si estuviese en casa. En cuanto a los comandos, son infinitos. El administrador puede desde iniciar o parar procesos hasta apagar o reiniciar la Raspberry Pi.

En este proyecto, el administrador tiene el deber de comprobar los archivos log\_seguridad.txt, log\_riego.txt y alertas.txt en el caso de que ocurriese algún problema o quisiese comprobar valores recogidos en el pasado. Por otro lado, también podrá acceder a la localización (aplicando el comando: sudo hologram modem location) y a los valores de la humedad. Además, deberá reiniciar o parar los programas de seguridad y riego si se encuentran problemas con los datos. Finalmente, apagará la Raspberry Pi si es necesario un mantenimiento o un arreglo del sistema.

* 1. Análisis del consumo de datos en la nube

El consumo de datos de una tarjeta SIM Hologram es gratuito siempre y cuando sea menos de 1 MB al mes. Esto es importante ya que es necesario inspeccionar cuándo y cómo se están gastando los datos contratados. Para visualizar el consumo, el administrador debe iniciar sesión en la nube de Hologram y clicar en uso (*Usage*). En esta sección existen dos apartados: Informes de uso y uso en directo (*Usage reports* y *live* respectivamente).

* + 1. Informes de uso

En este apartado, se puede ver de manera general el consumo de datos en un gráfico en el que se puede modificar las fechas de inicio y fin del informe.

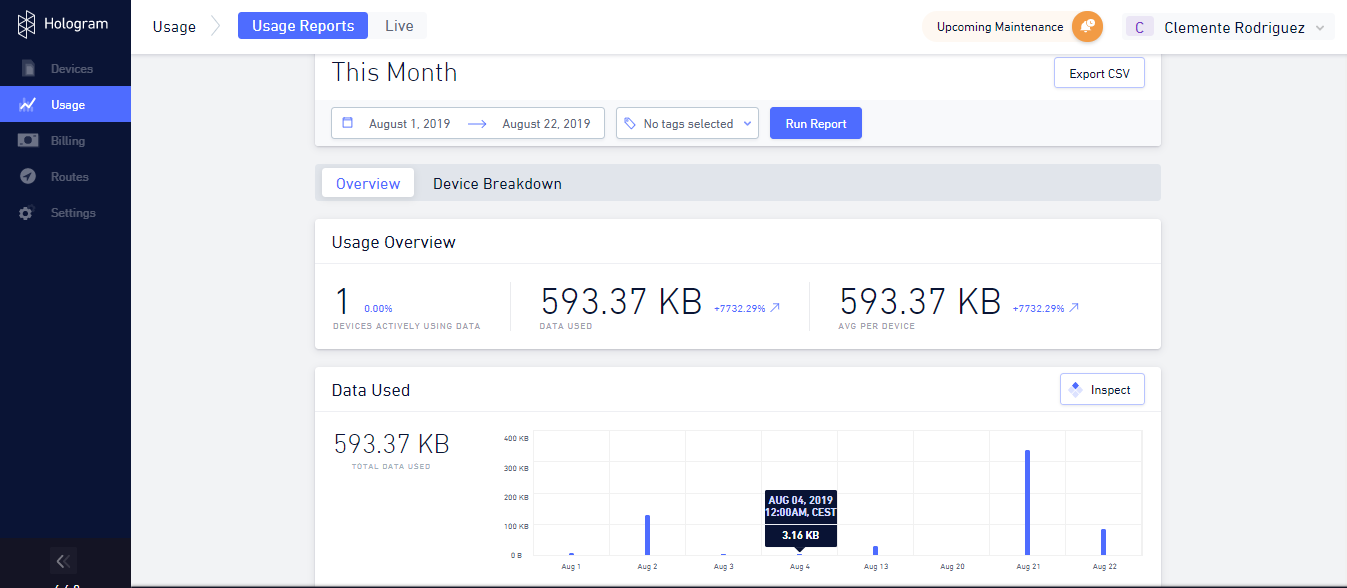


Figura 4.5.X: Informes de uso

Además, para más detalle, se puede clicar en exportar CSV (*export CSV*) para descargarse el archivo *CSV* con toda la información en ese periodo de tiempo. Por otro lado, si se quiere ver en la web, se clica en inspeccionar (*Inspect*). Se puede ver qué consumo hubo en cada día activo y en cuántas sesiones:

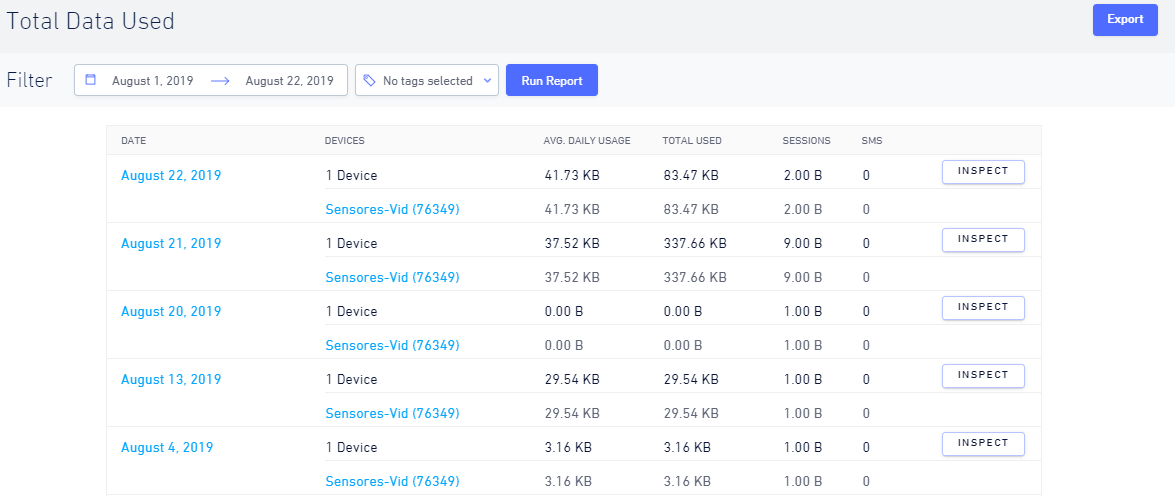


Figura 4.5.X: Inspeccionar por días

* + 1. Uso en directo

En este apartado, se puede ver en una gráfica el último mes de consumo y el usuario puede ampliar esa gráfica para visualizarlo en detalle tanto como desee. Además, debajo de la gráfica se encuentra el desglose de las sesiones.

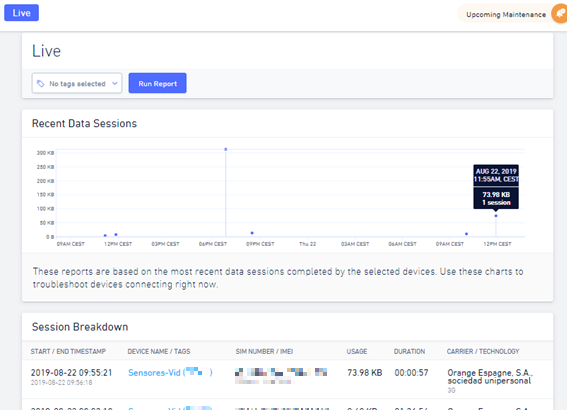


Figura 4.5.X: Consumo en vivo

* 1. La unión de todas las piezas

Para iniciar todo el sistema en la Raspberry Pi, es necesario iniciar todos los programas al mismo tiempo, es decir, riego.py y seguridad.py. Además, hay que escribir el comando para iniciar la conexión a Internet por el módem (Estado del arte).

En Linux (en este caso Raspbian), existe una herramienta para cumplir con este objetivo: Cron. Es un administrador de procesos en segundo plano que ejecuta procesos en diferentes intervalos de tiempo [<https://es.wikipedia.org/wiki/Cron_(Unix)>]. Los procesos que deben ejecutarse están guardados en un fichero llamado “crontab”.

Sin embargo, antes de guardar los comandos en este fichero, es necesario otorgar permisos al sistema de lectura, escritura y ejecución de los programas creados anteriormente. Para ello, se escribe *chmod 777 <archivo.py>*, donde archivo.py son riego.py y seguridad.py.

Una vez hecho esto, se accede al fichero crontab con el comando “crontab -e” y se escribe lo siguiente:

@reboot sudo hologram network connect && sleep 10 && sudo python2 /home/pi/Desktop/Hologram/seguridad.py && sleep 10 && sudo python2 /home/pi/Desktop/Hologram/riego.py

En orden, esto significa que en cada arranque de la Raspberry Pi se activa la conexión del módem a Internet. Una vez hecho, se espera diez segundos y se inicia el programa de seguridad. Otros diez segundos después, se inicia el programa de riego.

* 1. Simulación

En este apartado, se muestra un ejemplo de uso del sistema, cubriendo las principales acciones que se pueden realizar en el mismo. Para ello, se describirán paso a paso los diferentes eventos.

En primer lugar, la Raspberry Pi y la batería estarán colocadas en el lugar de trabajo, es decir, encima del motor de gasoil. El sensor de humedad tendrá un cable largo para que llegue al suelo y el sensor *PIR* se encontrará junto con el giroscopio en el motor. Este último estará mirando en el lado donde se encontrará el enganche del remolque, que es donde habría más presencia de personas en un caso sospechoso de robo.

\*Foto\*

Por otro lado, antes de iniciar el sistema, el administrador pide la información de correo electrónico y teléfono del cliente. Con ello, el administrador otorga los permisos para poder ver la información de la humedad en ThingSpeak y configura las rutas de alertas para lleguen al teléfono y al correo electrónico del cliente. En último lugar, le preguntaría al cliente si quiere adquirir un número de teléfono en la tarjeta SIM de la Raspberry Pi para enviar los SMS (1 dólar al mes) o preferiría que el administrador los enviase (el administrador cobraría cada vez que tuviese que enviar un SMS). En función del uso que le diese el cliente, elegiría entre estas dos opciones. En este caso, se elige la primera opción.

Para iniciar el sistema, el cliente debe conectar la batería a la Raspberry Pi. Mientras se inician los programas, el cliente ya se habría ido para evitar que el sensor *PIR* detecte su presencia.

Una vez en casa, si el cliente quiere ver cómo va la humedad del suelo, enciende el ordenador y se dirige a thingspeak.com. Después de iniciar sesión, con los permisos concedidos del administrador clica en el canal donde se encuentra la información de la humedad.

Después de esto, el cliente solamente tendría que esperar hasta volver a echar gasoil al motor. Al cabo de un tiempo, quiere ver cuál es el nivel de gasoil que hay en el depósito. Para ello, cuando llega al lugar de pruebas, envía el mensaje “D” y cuando reciba la respuesta de la Raspberry puede acceder al motor. Una vez haya terminado y quiera volver a casa, activa de nuevo el sistema enviando el mensaje “A”. Cuando reciba el mensaje de confirmación, el sistema de seguridad se ha vuelto a activar.

\*Foto de activación y desactivación\*

Si saltasen las alarmas, el cliente recibiría a la vez un SMS y un correo electrónico diciendo qué sensor ha saltado junto con la localización del dispositivo. Inmediatamente, si ve que la localización no es la del lugar de pruebas, avisaría a las autoridades e informaría de la situación hasta encontrar la máquina. Si todo se ha resuelto, el cliente debe desactivar y activar el sistema de seguridad utilizando los SMS.

\*Foto de email y SMS\*

Por otro lado, si hubiese un malfuncionamiento del sistema de seguridad o de riego, el cliente avisaría al administrador y este revisaría el sistema utilizando la terminal. En primer lugar, miraría los archivos log\_seguridad.txt, log\_riego.txt y alertas.txt para ver si hay una actividad anómala. Si es un problema de software, pararía, revisaría y volvería a activar los programas y si fuese un problema de hardware, apagaría la Raspberry Pi a distancia y examinaría el sistema in situ.

**Capítulo 5**

Evaluación

En este capítulo, se van a realizar las pruebas de evaluación de los requisitos funcionales del sistema. A continuación, se va a determinar si los requisitos se han cumplido en la implementación del producto:

* **RSF-01. Valor de la humedad en el suelo en tiempo real**: como se puede ver en riego.py, se recogen valores de la humedad cada 15 minutos. Por lo tanto, se cumple con este requisito correctamente.
* **RSF-02. Valor de la aceleración en tiempo real**: En seguridad.py, se recogen valores de la aceleración en los ejes X e Y cada 2 segundos a menos que esté desactivado o en modo alerta. Por lo tanto, se cumple con este requisito correctamente.
* **RSF-03. Valor de la rotación en tiempo real:** En seguridad.py, se recogen valores de la rotación en los ejes X e Y cada 2 segundos a menos que esté desactivado o en modo alerta. Por lo tanto, se cumple con este requisito correctamente.
* **RSF-04. Detección de cuerpo térmico:** En seguridad.py, se comprueba si hay variación de las radiaciones infrarrojas cada 2 segundos a menos que esté desactivado o en modo alerta. Por lo tanto, se cumple con este requisito correctamente.
* **RSF-05. Registro de los datos a lo largo del tiempo**: En el caso de la humedad, existe un archivo “log\_riego.txt” donde se guardaba la actividad del programa junto con los valores de humedad y además, se almacena la información de la humedad en la interfaz web de ThingSpeak en forma de gráfico o de archivo CSV, JSON o XML. En cuanto a los valores de la seguridad, se encuentran guardados en los archivos “log\_seguridad.txt” donde se guarda la actividad del programa junto con los valores de los sensores y “alertas.txt” donde se guardan solamente las alertas. Por lo tanto, se cumple con este requisito correctamente.
* **RSF-06. Interfaz web**: debido a las limitaciones en cuanto al uso de datos de Hologram (1 MB mensual) solamente se enviarían los datos de la humedad a ThingSpeak ya que es el dato más importante para ver en un gráfico. Se cumple parcialmente con el requisito.
* **RSF-07. Terminal**: gracias a una conexión SSH, se puede acceder totalmente al sistema y, por lo tanto, a los archivos log que tienen la información de los sensores. Adicionalmente, se puede parar o iniciar procesos y apagar o reiniciar el dispositivo a larga distancia. Por lo tanto, se cumple muy bien este requisito.
* **RSF-08. Comunicación del sistema**: Los dispositivos se pueden comunicar con el usuario de tres maneras diferentes: con ThingSpeak, con SMS y correos electrónicos de alerta y utilizando la terminal. Sin embargo, debido a las limitaciones del servicio de Hologram, no es posible enviar SMS a teléfonos españoles. En base a las pruebas, se consiguió enviar SMS a teléfonos de Reino Unido y Estados Unidos. De todas maneras, gracias al módem Hologram Nova, es posible cumplir correctamente este requisito en el resto de los aspectos.
* **RSF-09. Activación o desactivación del sistema de seguridad**: Con un SMS, el usuario tiene el poder de activar o desactivar el sistema de seguridad. Siguiendo la misma razón que en el requisito funcional 8, no es posible adquirir un número de teléfono español para la SIM en la Raspberry Pi y enviar SMS ahí. Como alternativa, se puede adquirir un número que funcione en otro país (como en Reino Unido o Estados Unidos) y enviar SMS allí o enviarlos gratuitamente desde la nube de Hologram. Se cumple parcialmente con este requisito.
* **RSF-10. Activación o desactivación del motor de gasoil**: En riego.py, el motor se enciende o se apaga en función de las condiciones de humedad en la zona. Por lo tanto, el requisito se cumple correctamente.

**Capítulo 6**

Planificación y presupuesto

#Introducción a la planificación y al desarrollo

5.1. Planificación

El desarrollo de este proyecto ha seguido diferentes fases: planificación previa, la definición del problema, el estudio de conceptos necesarios, el diseño, la implementación y la veracidad de su funcionamiento. Dada la longitud del proyecto, era necesario tener en cuenta el tiempo para su realización.

Una ventaja que se tenía en este proyecto es que se había elegido el problema con mucho tiempo. Esto permitía que se reflexionase sobre todos los puntos posibles a realizar ya que a pesar de que el problema principal estuviese muy claro, eran muchas las ideas para abordarlo. De hecho, a lo largo de su desarrollo, se cambiaron algunas herramientas para resolver el problema. A continuación, se incluirá una tabla que mostrará las horas totales de dedicación en este proyecto. En la misma, una columna extra indicará el número de horas estimadas diarias ya que no se invertía el mismo tiempo en diferentes épocas del año debido a otros trabajos o evaulaciones de carácter universitario. Además, después de la tabla, se incluirá un diagrama de Gantt para visualizar la información de la tabla:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Actividad | Inicio estimado (Día/Mes/Año) | Fin estimado (Día/Mes/Año) | Número días estimado | Estimación horas/día | Horas totales  actividad |
| Estudio Raspberry Pi – Instalaciones – Raspbian OS | 07/08/2018 | 18/08/2018 | 11 | 1 | 11 |
| Planificación – Estudio del entorno | 25/09/2018 | 02/10/2018 | 7 | 2 | 14 |
| Definición del problema | 01/10/2018 | 05/10/2018 | 4 | 1 | 4 |
| Estudio sensor humedad | 17/03/2019 | 19/03/2019 | 2 | 2 | 4 |
| Estudio giroscopio | 02/04/2019 | 04/04/2019 | 2 | 2 | 4 |
| Estudio *PIR* | 16/04/2019 | 18/04/2019 | 2 | 2 | 4 |
| Planificación – Investigación *IoT* | 15/05/2019 | 23/05/2019 | 8 | 2,5 | 20 |
| Definición del problema – Modificación de objetivos | 01/06/2019 | 07/06/2019 | 6 | 2,5 | 15 |
| Creación de programas – riego.py y seguridad.py | 20/06/2019 | 29/06/2019 | 9 | 2,5 | 22,5 |
| Estudio de Hologram Nova | 28/06/2019 | 10/07/2019 | 12 | 2,5 | 30 |
| Diseño del sistema completo | 20/07/2019 | 25/07/2019 | 5 | 2 | 10 |
| Modificación de los programas – Unión con Hologram | 28/07/2019 | 05/08/2019 | 8 | 2 | 16 |
| Simulaciones | 01/08/2019 | 12/08/2019 | 11 | 1 | 11 |
| Verificación y correcciones | 15/08/2019 | 24/08/2019 | 9 | 1 | 9 |
| Propuestas de mejora y conclusiones | 28/08/2019 | 30/08/2019 | 2 | 1 | 2 |
| Documentación | 01/06/2019 | 31/08/2019 | 91 | 1,5 | 136,5 |
|  | | Días totales | 127 | Horas totales | 313 |

Tabla 5.X: Evolución del proyecto

Figura 5.X: Diagrama de Gantt

5.2. Presupuesto

En este apartado, se expondrá el coste total del proyecto. Este consiste en costes relativos de personal contratado, el coste de los materiales utilizados y costes indirectos (aduanas y saldo hologram).

9770

916249635

916248984

5.2.1. Coste de personal

Al ser un Trabajo Fin de Grado, el proyecto se ha desarollado por una sola persona. Sin embargo, se ha realizado un estudio del presupuesto como si de un proyecto real en un entorno corporativo se tratase, dividiendo el proyecto en diferentes puestos en función del trabajo desempeñado.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Puesto | Salario estimado (anual bruto) | Salario por hora (1755 horas anuales [<https://www.boe.es/buscar/doc.php?id=BOE-A-2017-2536>]) | Horas totales | Salario Bruto total |
| Jefe de Proyecto |  |  |  |  |
| Analista |  |  |  |  |

5.2.2. Coste de materiales

En cuanto a los costes materiales, se tendrá en cuenta una tasa de amortización que determina el coste final de un material en función de su tiempo utilizado en el proyecto.

|  |  |  |  |
| --- | --- | --- | --- |
| Producto | Precio |  |  |
| Ordenador HP Pavilion … |  |  |  |
| One Plus 6T |  |  |  |
| Raspberry Pi 3 Model B | 36.30 € |  |  |
| RPi GPIO Breakout Board | 7.99 € |  |  |
| Sensor PIR HC-SR501 | 1.80 € |  |  |
| Giroscopio MPU6050 | 2.70 € |  |  |
| Sensor humedad del suelo | 2.40 € |  |  |
| Convertidor analógico a digital MCP3008 | 5.05 € |  |  |
| Tarjeta SIM - Hologram | $ 8.77 |  |  |
| Módem Hologram Nova |  |  |  |

**Capítulo 7**

Marco Regulador

**Capítulo 8**

Conclusiones y líneas futuras

7.1. Conclusión

El nivel de exigencia y dedicación que requería este proyecto era alto, pero me ha dejado un buen sabor de boca. Lo bueno de este trabajo es que la idea principal estaba muy clara desde el principio, ya que soy testigo de muchos agricultores que han invertido mucho tiempo para sacar su economía adelante y eso me motivó mucho, aunque al principio uno no sabía qué hacer para conseguirlo. Gracias a la labor de mi tutora y de resolver muchos problemas junto con buenos amigos, este proyecto llegó a ser posible. Además, me siento muy orgulloso de haber propuesto una nueva forma de ver el trabajo agrario en mi localidad.

Este proyecto se enfoca en aplicar los conceptos de internet de las cosas y de redes de sensores para mejorar el acceso de los usuarios a la información de un sistema de monitorización. El trabajo se adapta muy bien a la época actual ya que existe una tendencia a automatizar y digitalizar las cosas de nuestro entorno, es decir, a acceder a la información más rápido y sencillo.

Como ya se ha expuesto en el resto del documento, para optimizar el riego automático y mejorar la seguridad de las máquinas agrícolas, se han utilizado tres escenarios: El acceso a la información de la humedad en tiempo real gracias a la plataforma web, el envío de alertas al usuario vía SMS y correo electrónico gracias al servidor de Hologram y el uso de la terminal para tener a toda la información de los procesos en el lugar de pruebas.

El desarrollo del proyecto ha sido bastante interesante ya que se han utilizado diferentes dispositivos y tecnologías que nunca se había visto antes. Esto implica que se necesite más tiempo para pensar como utilizar bien todas las herramientas ya que hubo casos en los que se tuvo que cambiar de idea. Por ejemplo, antes se pensaba utilizar una conexión TCP en la terminal y se pasó haciendo una conexión SSH, mejorando la experiencia del usuario. Continuando con esto, uno de los retos que se tenía desde el principio era la comunicación a larga distancia entre el sistema de sensores y el ordenador de casa. Gracias al uso del módem de Hologram y de la numerosa cantidad de servicios disponibles en la nube, el traspaso de información se hizo de manera muy interactiva.

7.2. Líneas futuras

Hay muchas mejoras que se podrían realizar en el futuro, como el uso de datos de meteorología como la agencia estatal de meteorología (AEMET) para saber en el momento cuál sería la necesidad de agua diaria ya que el nivel de la lluvia no es el mismo todos los años. De esa manera, analizando la necesidad mensual y las probabilidades de lluvia diarias, se optimizaría aún más el riego automático.

Por otro lado, solo se ha utilizado un sistema de sensores para cubrir los dos objetivos principales. Sin embargo, se podría ampliar esto aumentando el número de nodos sensores de humedad (ya que el de seguridad estaría en el motor) y utilizar sistemas de comunicación como ZigBee en función del tamaño del terreno a analizar. Además, se podría cubrir otras tierras que tengan otros motores y controlar todas al mismo tiempo con el mismo sistema.

Por otro lado, mucha gente está empezando a utilizar placas solares para regar las plantas, aunque por la noche no se tenga la energía del sol para regar. También se podría enfocar el sistema en el uso la energía solar para favorecer un desarrollo más ecológico.

Finalmente, en cuanto a la batería, sería conveniente encontrar la manera de alimentarla gracias a un motor de gasoil o la placa solar para que el sistema pueda funcionar a todas horas.

Anexos

1. Glosario de acrónimos y definiciones
   1. Acrónimos

* API:
* TCP: Protocolo de Control de Transporte
* UDP: Protocolo de Datagramas de Usuario
* IP: Protocolo de Internet
* CSV:
* ADC: Conversor Analógica - Digital
* PIR: Sensor Infrarrojo Pasivo
* IDE: Entorno de Desarrollo Integrado
* GPS: Sistema de Posicionamiento Global
* *Modem*: Modulador – Demodulador
* IoT: Internet de las Cosas
* IIoT: Internet Industrial de las Cosas
* WSN: Redes de Sensores Inalámbricas
* TIC: Tecnologías de la Información y la Comunicación
* GNU: *GNU’s not Unix*. El diseño de GNU es de tipo *Unix* pero difiere de éste ya que es software libre y no contiene código *Unix*.
* IDC: Corporación de Datos Internacionales
* AI: Inteligencia Artificial
* RFID:
* SSH:
* WiFi
* GSM
* *VNC*, *telnet*, *rlogin*
  1. Definiciones
* Módem: es un dispositivo que convierte las señales digitales en analógicas (modulación) y viceversa (desmodulación) que sirve para enviar la señal moduladora mediante otra señal llamada portadora. [https://es.wikipedia.org/wiki/Modem]
* *Big Data*: En español, datos masivos, son archivos de información de gran volumen, alta velocidad y/o gran variedad que exigen formas rentables e innovadoras de procesamiento de información que permitan una mejor comprensión, toma de decisiones y automatización de procesos [<https://www.gartner.com/it-glossary/big-data/>].
* *Unix*: es un sistema operativo portable, multitarea y multiusuario; desarrollado en 1969 por un grupo de empleados de los laboratorios Bell de AT&T, entre los que figuran Dennis Ritchie, Ken Thompson y Douglas McIlroy [<https://es.wikipedia.org/wiki/Unix>]

1. Ampliación de conceptos sobre la vid
2. Mapa del Acuífero “¿Número?”

139.74 $ + 45 $ + 4.99$= 184.74$ +4.99$ = 189.73$

900 – 189.73 = 710.27$ 🡪 **634.53€**



Adamo

Llamadas móviles + 1000 MB + fijas ilimitadas

Ilimitadas + 3GB

84.99 🡪 3X

50MB 🡪78.99