



Diseño de un sistema de fertirrigación automático con control telemático y sensores

Oinatz Aspiazu Ituarte

Máster Universitario en Ingeniería de Telecomunicación UOC-URL
Telemática

Dr. José López Vicario

Dr. Javier Vilajosana Guillén

Junio 2019



Esta obra está sujeta a una licencia de Reconocimiento-NoComercial-SinObraDerivada [3.0 España de Creative Commons](https://creativecommons.org/licenses/by-nc-nd/3.0/es/)

FICHA DEL TRABAJO FINAL

| | |
|---|--|
| Título del trabajo: | <i>Diseño de un sistema de fertirrigación automático con control telemático y sensores</i> |
| Nombre del autor: | <i>Oinatz Aspiazu Ituarte</i> |
| Nombre del consultor/a: | <i>Dr. José López Vicario</i> |
| Nombre del PRA: | <i>Dr. Javier Vilajosana Guillén</i> |
| Fecha de entrega (mm/aaaa): | 06/2019 |
| Titulación:: | <i>Máster Universitario en Ingeniería de Telecomunicación UOC-URL</i> |
| Área del Trabajo Final: | <i>Telemática</i> |
| Idioma del trabajo: | <i>Español</i> |
| Palabras clave | <i>Zigbee, LTE, sensores</i> |
| <p>Resumen del Trabajo (máximo 250 palabras): <i>Con la finalidad, contexto de aplicación, metodología, resultados i conclusiones del trabajo.</i></p> | |
| <p>Este trabajo pretende demostrar la viabilidad económica del diseño de un sistema de fertirrigación automático en un entorno rural que implica utilizar una extensión importante de terreno. Para poder mantener la calidad de los cultivos, se realiza un estudio teórico donde se introducen varios sensores para medir los valores de pH de las tierras, así como la humedad relativa y temperatura del ambiente. Por otro lado se pretende recoger desde un ordenador central y de manera telemática todas estas métricas capturados por los sensores. Con esta información recogida, desde el ordenador central podemos posteriormente enviar y recibir así las señales adecuadas a través del circuito de riego.</p> <p>Para ello, dada la dificultad de un cableado en el medio rural y con el objetivo de evitar en lo posible costes, se introduce una estación base propia utilizando LTE en banda no licenciada. También se utilizan redes de sensores en malla que se comunican mediante Zigbee. Ello implica el uso de dispositivos que sepan entender ambos protocolos de red, diseñar las correspondientes adaptaciones de <i>hardware</i> y de código fuente y tener que hacer frente a las limitaciones propias de las redes inalámbricas como la pérdida de señal debido a las atenuaciones por la distancia y los obstáculos.</p> <p>Este diseño implica poder realizar una validación de los elementos utilizados, siendo interesante la monitorización constante del sistema. Se introduce finalmente una valoración económica para validar el proyecto realizado.</p> | |

Abstract (in English, 250 words or less):

This work aims to demonstrate the economic feasibility of designing an autonomous fertigation system in a rural environment that involves the use of an important extension of land. In order to maintain the quality of the crops, a theoretical study is carried out where several sensors are introduced to measure the pH quality of the land, as well as the relative humidity and temperature of the environment. On the other hand it is intended to collect from a central computer and telematically, all these values captured by the sensors. With this information collected, from the central computer we can subsequently send and receive the appropriate signals through the irrigation circuit.

To do this, given the difficulty of wiring in rural areas and with the aim of avoiding costs as much as possible, an own base station is introduced using LTE in unlicensed band. Mesh sensor networks that communicate through Zigbee are also used. This implies the use of devices that can understand both network protocols, design the corresponding hardware and source code adaptations and have to face the limitations of wireless networks such as signal loss due to distance attenuations and the obstacles.

This design implies being able to perform a validation of the elements used, being interesting the constant monitoring of the system. Finally, an economic valuation is introduced to validate the project carried out.

Índice

| | |
|---|----|
| 1. Introducción..... | 1 |
| 1.1. Contexto y justificación del Trabajo..... | 1 |
| 1.2 Objetivos del Trabajo..... | 1 |
| 1.2.1 Objetivos Generales..... | 1 |
| 1.2.2 Objetivos Específicos..... | 2 |
| 1.3 Enfoque y método seguido..... | 2 |
| 1.4 Planificación del Trabajo..... | 2 |
| 1.5 Breve resumen de productos obtenidos..... | 4 |
| 1.6 Breve descripción de los otros capítulos de la memoria..... | 4 |
| 2. Estado del Arte..... | 6 |
| 2.1 Sistemas de lazo cerrado..... | 6 |
| 2.1.1 Riego automático..... | 7 |
| 2.2 pH y nitratos..... | 7 |
| 2.3 Redes inalámbricas..... | 10 |
| 2.3.1 Zigbee..... | 10 |
| 2.3.1.1 Tipos de dispositivos en una red Zigbee..... | 10 |
| 2.3.1.2 Consumo de energía..... | 11 |
| 2.3.1.3 Estructura interna en capas..... | 11 |
| 2.3.1.4 Campos de los tipos de paquetes básicos..... | 12 |
| 2.3.1.5 Configuraciones de red..... | 13 |
| 2.3.1.6 Seguridad..... | 14 |
| 2.4.2 4G..... | 15 |
| 2.4.2.1 LTE vs UMTS..... | 16 |
| 2.4.2.2 Bandas de la red LTE..... | 16 |
| 2.4.2.3 E-UTRAN y eNodeB..... | 17 |
| 2.4.2.4 Componentes de la red LTE..... | 18 |
| 2.4.2.5 Enlace descendente..... | 19 |
| 2.4.2.6 Enlace ascendente..... | 20 |
| 2.4.2.7 Transmisión de datos..... | 21 |
| 2.4.2.8 Canales..... | 22 |
| 2.4.2.9 MIMO..... | 24 |
| 2.4.2.10 LTE Advanced..... | 26 |
| 2.5 Por qué LTE y Zigbee..... | 27 |
| 2.6 Atenuaciones y obstáculos..... | 28 |

| | | |
|---------|--|----|
| 2.6.1 | Cálculo de la atenuación en medio libre | 28 |
| 2.6.2 | Propagación sobre espacio libre | 28 |
| 2.6.3 | Modelo Erceg | 29 |
| 2.7 | Interconexión de sensores a través de topologías de red de malla | 30 |
| 2.8 | Software utilizado | 31 |
| 2.8.1 | Software del ordenador central | 31 |
| 2.8.2 | Software dispositivos Raspberry Pi | 31 |
| 2.8.3 | Software genérico | 32 |
| 3. | Diseño de arquitectura | 33 |
| 3.1 | Aspectos del problema a resolver | 33 |
| 3.2 | Solución deseada | 34 |
| 3.3 | Componentes del sistema | 36 |
| 3.3.1 | Sensores y actuadores | 37 |
| 3.3.1.1 | Sensor de temperatura y humedad relativa | 37 |
| 3.3.1.2 | Sensor de pH | 38 |
| 3.3.2 | Resistencias varias | 40 |
| 3.3.3 | Raspberry Pi | 41 |
| 3.3.3.1 | Módulo XBee Zigbee Pro | 42 |
| 3.3.3.2 | Módulo LTE | 44 |
| 3.3.4 | Estación base LTE-U | 46 |
| 3.3.5 | Ordenador central | 48 |
| 3.4 | Conexiones de sensores y dispositivos a Raspberry Pi | 48 |
| 3.4.1. | Sensor de pH y Raspberry Pi | 48 |
| 3.4.2. | Sensor de temperatura y humedad Raspberry Pi | 49 |
| 3.4.3 | Módulo XBee y Raspberry Pi | 51 |
| 3.4.4 | Módulo LTE y Raspberry Pi | 51 |
| 3.4.5 | Resolviendo conflictos con sensores | 51 |
| 3.5. | Construyendo una red Zigbee | 51 |
| 3.6. | Distribución de los diversos elementos | 53 |
| 3.7. | Resumen elementos implementados | 54 |
| 4 | Validación del sistema | 55 |
| 4.1 | Requisitos del sistema | 55 |
| 4.2 | Pruebas a realizar | 55 |
| 4.3 | Calibraciones | 56 |
| 4.3.1 | Calibración de sensores y comparación de métricas recogidas | 56 |
| 4.3.1.1 | Calibración del sensor de temperatura | 56 |
| 4.3.1.2 | Calibración del sensor de humedad relativa | 57 |

| | |
|--|----|
| 4.3.1.3 Calibración del sensor de pH..... | 57 |
| 4.4 Métricas para medir prestaciones..... | 57 |
| 4.4.1 Medición de corriente eléctrica | 57 |
| 4.4.2 Información recogida de los dispositivos intermedios | 58 |
| 4.4.3. Atenuaciones y Zigbee..... | 60 |
| 4.4.4 LTE y atenuaciones en espacio libre | 62 |
| 4.4.5 LTE y Erceg | 63 |
| 4.5 Elementos de riego..... | 65 |
| 4.6 Seguridad de los sistemas | 66 |
| 4.7 Resumen de elementos utilizados en la verificación de requisitos | 66 |
| 5. Valoración económica y viabilidad del proyecto | 67 |
| 6. Conclusiones..... | 69 |
| 7. Glosario | 70 |
| 8. Bibliografía | 71 |
| 9. Anexos | 75 |
| 9.1 Código apartado 3.3.1..... | 75 |
| 9.2 Código apartado 3.3.2 | 79 |
| 9.3 Código apartado 3.5. Dispositivo final | 80 |
| 9.4 Código apartado 3.5. Coordinador | 80 |
| 9.5. Código apartado 3.5. Sensores..... | 80 |
| 9.6. Código apartado 4.3.4 | 81 |

Lista de figuras

| | |
|---|----|
| Tabla 1 - Planificación del trabajo | 3 |
| Tabla 2 - pH fertilizantes sólidos y líquidos | 8 |
| Tabla 3 - Requerimientos de pH por cultivo | 9 |
| Tabla 4 - Estructura interna en capas | 12 |
| Tabla 5 - Tramas de red..... | 12 |
| Tabla 6 - Especificaciones LTE | 15 |
| Tabla 7 - Bandas LTE..... | 17 |
| Tabla 8 - Modos de uso de MIMO | 26 |
| Tabla 9 - Porqué LTE y Zigbee | 27 |
| Tabla 10 - Parámetros a, b, c y s , según el medio | 29 |
| Tabla 11 - Especificaciones DHT22 | 38 |
| Tabla 12 - Elementos kit pH Atlas Scientific..... | 39 |
| Tabla 13 - Especificaciones circuito de pH..... | 40 |
| Tabla 14 - Características XBee Zigbee Pro | 44 |
| Tabla 15 - Características AirPrime® EM7565 LTE-Advanced Pro..... | 45 |
| Tabla 16 - Especificaciones U4G-AP1000 | 47 |
| Tabla 17 - Conexiones DHT22 y Raspberry Pi..... | 50 |
| Tabla 18 - Resumen elementos implementados | 54 |
| Tabla 19 - Resumen pruebas a realizar | 56 |
| Tabla 20 - Relación elemento y corriente eléctrica utilizada..... | 58 |
| Tabla 21 - Valores método Erceg..... | 64 |
| Tabla 22 - Resumen verificación requisitos..... | 66 |
| Tabla 23 - Valoración económica | 67 |
| Tabla 24 - Retorno de inversión | 68 |
| | |
| Gráfico 1 - Diagrama de Gantt | 4 |
| Gráfico 2 - Sistema de lazo cerrado | 7 |
| Gráfico 3 - Configuraciones de red | 14 |
| Gráfico 4 - E-UTRAN | 17 |
| Gráfico 5 - Componentes LTE | 18 |
| Gráfico 6 - Relación ancho de banda y subportadoras | 19 |
| Gráfico 7 - Enlace descendente | 20 |
| Gráfico 8 - Enlace ascendente | 21 |
| Gráfico 9 - Símbolos, slots, bloques radio y tramas | 22 |
| Gráfico 10 - Canales enlace descendente | 23 |

| | |
|--|----|
| Gráfico 11 - Canales enlace ascendente | 24 |
| Gráfico 12 - MIMO | 25 |
| Gráfico 13 - Red de malla | 30 |
| Gráfico 14 - Red de malla parcial | 31 |
| Gráfico 15 - Diagrama de bloques | 35 |
| Gráfico 16 - Grupo sensado | 36 |
| Gráfico 17 - Sensor DHT22 | 38 |
| Gráfico 18 - Sonda pH | 39 |
| Gráfico 19 - Soluciones salinas | 39 |
| Gráfico 20 - PCB sonda pH | 40 |
| Gráfico 21 - Resistencias | 41 |
| Gráfico 22 - Raspberry Pi | 42 |
| Gráfico 23 - Conector GPIO Raspberry Pi | 42 |
| Gráfico 24 - XBee Zigbee Pro | 43 |
| Gráfico 25 - Adaptador USB para XBee | 44 |
| Gráfico 26 - AirPrime® EM7565 LTE-Advanced Pro | 45 |
| Gráfico 27 - Adaptador USB 2.0 a M.2 mini PCI-E | 46 |
| Gráfico 28 - Antenas dispositivo LTE | 46 |
| Gráfico 29 - U4G-AP1000 | 47 |
| Gráfico 30 - Conexiones kit sensor pH a Raspberry Pi | 49 |
| Gráfico 31 - Conexiones DHT22 | 50 |
| Gráfico 32 - Red Zigbee | 52 |
| Gráfico 33 - Subred Zigbee | 52 |
| Gráfico 34 - Distribución elementos | 53 |

1. Introducción

1.1. Contexto y justificación del Trabajo

La medición de la cantidad de los niveles de nitratos cobra especial relevancia cuando se pretende realizar un análisis de nutrientes del suelo, la savia de las plantas para evaluar la eficiencia de la fertilización y la nutrición de la cosecha o medición de soluciones de sustrato en invernaderos. También se utilizan para analizar alimentos, evaluar la calidad de los ríos o del agua potable.

Estos motivos hacen que se antoje de interés acercarse al mundo rural mediante un estudio de diversos tipos de sensores para comprender cómo evaluar la calidad de los cultivos y recoger toda la información de manera telemática. Para ello, además de pretender conocer el nivel de nitratos en el suelo a través del estudio de pH del mismo, tenemos que pensar en que habitualmente los cultivos en el medio rural suelen ser de difícil alcance a nivel de comunicaciones. Se estudiará qué tipo de sensores utilizar en cada caso y cuáles son los medios inalámbricos más óptimos, desde el uso de redes de bajo consumo energético en las proximidades de los mismos, a redes de mayor alcance y ancho de banda, de cara a poder recopilar de manera inalámbrica toda la información pudiendo realizar un análisis y procesamiento de los datos que ayude a obtener un cultivo eficiente.

La información proporcionada por los diversos sensores, nos sirve para diseñar un sistema de fertirrigación [1], donde se añadirán al agua fertilizantes apropiados para regar los cultivos. Este sistema de riego, aunque implica un desembolso inicial, supone un ahorro importante de costes de agua y fertilizante (además del ahorro de mano de obra o desplazar maquinaria con la consiguiente erosión adicional de las tierras).

1.2 Objetivos del Trabajo

1.2.1 Objetivos Generales

1. Diseñar un sistema de riego y fertilización automáticas
2. Análisis del entorno rural
3. Envío y recepción de señales en el medio rural para controlar el sistema de riego y fertilización
4. Obtener un sistema viable y eficiente en costes

1.2.2 Objetivos Específicos

1. Análisis de pH de suelo (evaluación de la concentración de nutrientes en el suelo)
2. Análisis de temperatura ambiente y humedad relativa ambiente.
3. Análisis de la calidad de las aguas de riego
4. Diseño de redes en malla eficientes
5. Transmisión de datos por LTE eficiente y con costes mínimos

1.3 Enfoque y método seguido

Aunque existen investigaciones al respecto, muchos de los proyectos que podemos encontrar de riego automático, se refieren especialmente a ámbitos pequeños donde se reducen a un pequeño huerto controlado directamente por alguna pequeña placa y sensores sin un control telemático.

Dado que por un lado no se dispone del terreno de estudio, tratándose de una extensión elevada y por otro lado, el coste de todos los dispositivos es muy elevado para adquirirlos para el estudio académico, la estrategia a seguir consiste en construir un diseño teórico apoyándonos en bibliografía diversa. Se han revisado estos proyectos referenciándose la bibliografía cuando corresponde y se diseña una solución experimental propia. Para ello se adoptan dispositivos con procesadores embebidos que sirven para proyectos de ámbitos pequeños y se llevan a un entorno que implica controlarlos en una distancia elevada, teniendo que tener en cuenta cómo enviar y recoger la información a través de las señales propagadas.

Debido a que los dispositivos remotos pueden estar instalados en lugares de difícil acceso, el enfoque implica tener en cuenta que hay que buscar ser eficientes con un coste óptimo. Además se utilizan diferentes tipos de sensores, adaptados al caso de estudio, implicando el desarrollo de diferentes métodos, rutinas y envío y recepción de señales adecuadas.

1.4 Planificación del Trabajo

En la primera entrega, se realiza una primera planificación donde se estima tener el proyecto terminado para la fecha de entrega de la PEC2 y poder utilizar el tiempo de la PEC3 para correcciones.

Conforme se va avanzando durante la segunda entrega, surgen nuevos problemas y causísticas que denotan que es necesaria una replanificación.

A través de la herramienta ProjectLibre [39], se realiza la siguiente planificación del trabajo en la tercera PEC (modificada sobre la planificación entregada con la PEC2):

| Nombre | Duración | Inicio | Terminado | Predecesores |
|--|------------------|----------------------|-----------------------|--------------|
| TFM | 77 days? | 28/02/19 8:00 | 14/06/19 17:00 | |
| PEC1 | 5 days? | 28/02/19 8:00 | 6/03/19 17:00 | |
| Planificación - Gantt | 3 days? | 28/02/19 8:00 | 4/03/19 17:00 | |
| Definición de Objetivos | 2 days? | 5/03/19 8:00 | 6/03/19 17:00 | 3 |
| PEC2 | 35 days? | 7/03/19 8:00 | 24/04/19 17:00 | 4 |
| Búsqueda y recopilación de documentación | 2 days? | 7/03/19 8:00 | 8/03/19 17:00 | |
| Formación sobre sensores y tecnologías | 6 days? | 11/03/19 8:00 | 18/03/19 17:00 | 6 |
| Diseño trabajo investigación | 21,125 days? | 19/03/19 8:00 | 17/04/19 9:00 | 7 |
| Formato Memoria parcial | 5,875 days? | 17/04/19 9:00 | 24/04/19 17:00 | 8 |
| PEC3 | 18,5 days | 25/04/19 8:00 | 21/05/19 13:00 | 9 |
| Correcciones y mejoras | 10 days | 25/04/19 8:00 | 8/05/19 17:00 | |
| Validación del sistema | 3 days | 9/05/19 8:00 | 13/05/19 17:00 | 11 |
| Valoración económica y viabilidad del proyecto | 1 day | 14/05/19 8:00 | 14/05/19 17:00 | 12 |
| Código fuente | 2 days | 15/05/19 8:00 | 16/05/19 17:00 | 13 |
| Formato Memoria PEC3 | 2 days | 17/05/19 13:00 | 21/05/19 13:00 | 14 |
| Cierre TFM | 12 days? | 23/05/19 8:00 | 7/06/19 17:00 | 15 |
| Memoria | 6 days? | 23/05/19 8:00 | 30/05/19 17:00 | |
| Bibliografía | 2 days? | 31/05/19 8:00 | 3/06/19 17:00 | 17 |
| Conclusiones | 1,875 days? | 4/06/19 8:00 | 5/06/19 16:00 | 18 |
| Aceptación | 1,75 days? | 5/06/19 16:00 | 7/06/19 14:00 | 19 |
| Revisión y entrega memoria final | 0,375 days? | 7/06/19 14:00 | 7/06/19 17:00 | 20 |
| Presentación | 5 days? | 10/06/19 8:00 | 14/06/19 17:00 | 21 |
| Realizar presentación | 4 days? | 10/06/19 8:00 | 13/06/19 17:00 | |
| Revisar presentación | 1 day? | 14/06/19 8:00 | 14/06/19 17:00 | 23 |

Tabla 1 - Planificación del trabajo

De cara a reducir la extensión, en la tercera entrega de seguimiento, se elimina el sensor de humedad del suelo dado que es redundante al tener ya el sensor de pH (se recogen además ya 3 métricas diferentes y añadir una métrica adicional no aporta calidad al estudio).

También se elimina la medición de nitratos en la calidad de agua de riego, control de bombas de riego y control de válvulas y actuadores. Si bien su uso es importante en el sistema de riego y el envío de las diversas señales eléctricas es interesante a nivel eléctrico, no corresponde realmente al área de Telemática, objeto final de este proyecto. Sobre este último punto ocurre también que ya se realiza el envío y recepción de señales a otros elementos como los sensores a través del sistema telemático.

Durante la PEC3, surgen nuevas dificultades teniéndose que realizar diversas correcciones que implican una nueva replanificación y reorganización del trabajo realizado en la PEC2. Se incluyen también dos nuevos apartados al proyecto: Un apartado de validación del sistema (apartado 4) y un apartado de valoración económica (apartado 5)

Se reorganiza y reescribe la documentación (especialmente los apartados 2 y 3), de cara a adaptarla a los objetivos solicitados después del seguimiento de la segunda PEC.

de desarrollar una red inalámbrica correcta para cubrir toda la extensión del terreno y se implementa un sistema de monitorización con las validaciones oportunas sobre el sistema de riego.

El capítulo 5 incluye la valoración económica y la viabilidad del proyecto.

El capítulo 6, son las conclusiones obtenidas con el desarrollo del trabajo.

Se incluye un glosario de términos y bibliografía con todas las fuentes utilizadas y / o consultadas para realizar esta memoria.

2. Estado del Arte

Nos encontramos en una época en pleno auge del Internet de las cosas. Las nuevas redes de bajo consumo y las posibilidades proporcionadas por el alcance y el ancho de banda de las redes basadas en 4G (y en un futuro en 5G), además de los avances tecnológicos en el campo de los microprocesadores, ayudan a plantear el uso de dispositivos autónomos, pequeños, con un bajo consumo eléctrico y que pueden ser colocados en lugares de difícil acceso.

El medio rural donde prevalecen los cultivos en áreas extensas, siendo habitualmente de difícil acceso y ninguna posibilidad de cableado eléctrico o de comunicaciones, es ideal para realizar un estudio de comunicaciones inalámbricas y de sensores eficientes que permitan diseñar un sistema de riego y fertilización que ayude a mejorar la calidad de los cultivos. Además, aunque un estudio sobre el medio denote una inversión inicial elevada, la amortización puede ser relativamente rápida dado que una vez que el sistema está en marcha, sólo tenemos que preocuparnos del mantenimiento del mismo con el consiguiente ahorro de costes que supone.

Respecto a los medios inalámbricos, se tendrá en cuenta, el diseño de un sistema de envío de señales de largo alcance, de cara a conseguir el envío y recepción de las señales a lo largo de toda la extensión a controlar y también que las señales sean eficientes energéticamente en el lado de los sensores, de cara a controlar el consumo eléctrico. Es por ello por lo que se pretenderá combinar señales 4G a través de varios repetidores y el uso de señales Zigbee de bajo consumo en el lado de los sensores. Se ideará también un mecanismo que sea capaz de recoger de manera intermedia las señales 4G y transmitir información por Zigbee y viceversa.

2.1 Sistemas de lazo cerrado

El diseño del sistema de riego que se está definiendo, implica recoger información desde y hacia los sensores, correspondiéndose a un sistema de control de lazo cerrado (realimentado), donde se permite conseguir que la salida mantenga el valor deseado a pesar de las perturbaciones externas no previstas o las propias variaciones internas de los elementos que conforman el sistema [8].

El operador fija las variables de referencia responsables del comportamiento deseado y el sistema de control genera las señales de mando para conseguir que la salida del proceso, acabe teniendo el valor que queremos sin verse modificada por las perturbaciones externas.

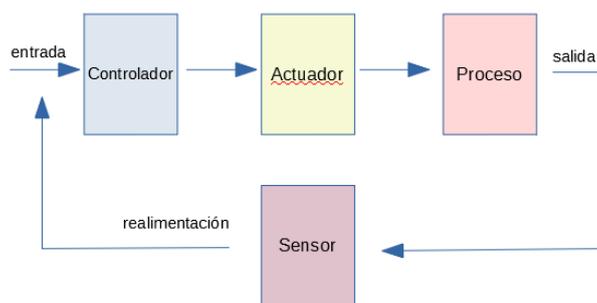


Gráfico 2 - Sistema de lazo cerrado

2.1.1 Riego automático

Aunque se entrará en detalle posteriormente a lo largo del apartado 3, se realiza una breve introducción de los componentes principales, de cara a introducir posteriormente las diversas tecnologías de estudio.

El sistema de lazo cerrado correspondiente al riego automático, comprende los siguientes elementos:

- El controlador: un programador de riego. Es el elemento del sistema que toma las decisiones de cuándo regar, fertilizar, abrir y cerrar las válvulas o enviar y recoger la información necesaria y durante cuánto tiempo.
- Uno o más actuadores: Son válvulas que se activan eléctricamente de forma que al recibir una señal eléctrica del programador se abren y permiten el paso del agua o fertilizante por una tubería, llegando a las plantas.
- Sensores: Permiten saber el estado de la tierra para saber si hay que regar o fertilizar. El sistema de control al tener sensores, hay realimentación (el controlador recibe información sobre el proceso, enviando las señales necesarias sobre los actuadores, abriéndose o cerrándose las válvulas correspondientes).
- Canal y señales: Se define además un canal, en este caso el medio aéreo, por el cual se transmitirán las señales con la información necesaria.

2.2 pH y nitratos

En referencia a los sensores introducidos, se utilizan varios tipos de ellos. Sirven para medir la humedad relativa, la temperatura o el valor de pH del suelo. Este último cobra especial interés para tener un control preciso del riego. Es por ello,

que se introduce en este apartado una descripción que indique cómo se relaciona el pH con la cantidad de nitratos del suelo.

El control del pH en la disolución nutritiva utilizada en fertirrigación es imprescindible para obtener los valores óptimos para la absorción de nutrientes. Este pH de la “solución nutritiva ideal” se encuentra en el rango 5,5 – 6. [22]

A continuación se indican los pH de ciertos fertilizantes utilizados en fertirrigación para un concentración del 1%:

| Fertilizantes sólidos | Riqueza | pH | CE (mmho /cm) 1% | Solubilidad (g/l) |
|--------------------------|---------------------------------------|------|------------------|-------------------|
| Nitrato amónico | 34,5% N | 5,17 | 11,58 | 1,970 |
| Nitrato cálcico | 15,5%N 27%CaO | 5,87 | 1,18 | 1,260 |
| Nitrato magnésico | 11%N 15,7%MgO | 5,43 | 0,88 | 420 |
| Sulfato cálcico | 15%MgO | 5,43 | 0,75 | 360 |
| Fosfato monoamónico | 12%N 61%P ₂ O ₅ | 4,9 | 0,89 | 380 |
| Nitrato potásico | 13%N 46%K ₂ O | 7,02 | 1,36 | 320 |
| Cloruro potásico | 60%K ₂ O | 7,01 | 1,88 | 340 |
| NPK cristalino | varios | - | - | - |
| Fertilizantes líquidos | Riqueza | pH | CE (mmho /cm) 1% | Solubilidad (g/l) |
| Solución nitrogenada 32% | 32%N | 5,54 | 0,69 | - |
| Solución nitrogenada 20% | 20%N | 6,37 | 0,87 | - |
| Ácido nítrico | 12,5%N | <1 | <0,5 | - |
| Solución N. cal | 8%N 16%CaO | 6,4 | 0,63 | - |
| Solución N. magnésico | 7%N 9,5%CaO | 5,4 | 0,5 | - |
| Ácido fosfórico | 52% P ₂ O ₅ | <1 | 1,5 | - |
| Solución potásica | varios | - | - | - |
| Solución NPK | varios | - | - | - |

Tabla 2 - pH fertilizantes sólidos y líquidos

Dado que la intención es sólo utilizar fertilizantes líquidos, se contempla sólo la segunda parte de la tabla.

Por otro lado, se conoce además que dependiendo del tipo de cultivo, las necesidades de pH son diferentes [23]. La siguiente tabla indica los pH óptimos dependiendo del tipo de cultivo:

| Hortícolas | pH óptimo | Frutales | pH óptimo | Extensivos | pH óptimo |
|-----------------|-----------|--------------|-----------|----------------|-----------|
| Acelga | 6.0-7.5 | Albaricoque | 6.0-6.8 | Alfalfa | 6.5-7.8 |
| Apio | 6.1-7.4 | Almendro | 6.0-6.8 | Algodón | 5.0-6.2 |
| Berenjena | 5.4-6.0 | Avellano | 6.0-7.0 | Alpiste | 6.0-7.0 |
| Boniato | 5.1-6.0 | Café | 5.0-7.0 | Altramuz | 5.0-7.0 |
| Brócoli | 6.0-7.2 | Castaño | 5.0-6.5 | Arroz | 5.0-6.5 |
| Calabaza | 5.6-6.8 | Encina | 4.8-6.0 | Avena | 5.2-7.1 |
| Cebolla | 6.0-7.2 | Grosellero | 6.0-7.0 | Batatas | 5.3-6.5 |
| Col | 6.0-7.5 | Limonero | 6.0-7.5 | Cacahuete | 5.3-6.5 |
| Col de Bruselas | 5.7-7.2 | Manzano | 5.3-6.7 | Caña de azúcar | 6.0-7.8 |
| Coliflor | 6.0-7.2 | Melocotonero | 5.3-6.8 | Cáñamo | 6.2-7.2 |
| Escarola | 5.6-6.8 | Membrillero | 5.5-7.2 | Cebada | 6.4-7.8 |
| Espárrago | 6.3-7.5 | Naranja | 6.0-7.5 | Centeno | 5.3-6.8 |
| Espinaca | 6.3-7.1 | Nogal | 6.2-7.8 | Colza | 5.8-7.1 |
| Fresa | 5.0-6.2 | Olivo | 6.0-7.8 | Dáctilo | 5.6-7.2 |
| Guisantes | 5.9-7.3 | Peral | 5.6-7.2 | Girasol | 6.0-7.2 |
| Judías | 5.8-6.8 | Pino | 5.0-6.0 | Habas | 7.4-8.1 |
| Lechugas | 5.8-7.2 | Platanera | 6.0-7.5 | Lenteja | 5.0-7.0 |
| Maíz dulce | 5.6-6.8 | Pomelo | 6.0-7.5 | Lino | 5.5-7.5 |
| Melón | 5.7-7.2 | Vid | 5.3-6.7 | Maíz | 5.5-7.5 |
| Acelga | 6.0-7.5 | Albaricoque | 6.0-6.8 | Alfalfa | 6.5-7.8 |
| Apio | 6.1-7.4 | Almendro | 6.0-6.8 | Algodón | 5.0-6.2 |
| Berenjena | 5.4-6.0 | Avellano | 6.0-7.0 | Alpiste | 6.0-7.0 |
| Boniato | 5.1-6.0 | Café | 5.0-7.0 | Altramuz | 5.0-7.0 |

Tabla 3 - Requerimientos de pH por cultivo

Se deben evitar valores de pH inferiores a 5 (a pH = 4 se dañaría la raíz de la mayoría de los cultivos). También se deben evitar valores superiores a 6,5 donde bajarían drásticamente la disponibilidad de algunos micronutrientes. Hay que te-

ner en cuenta que en los suelos o en los sustratos orgánicos, debido a la capacidad tampón que éstos poseen, las variaciones de pH se produce a largo plazo, sin embargo en la mayoría de los sustratos minerales tiene capacidad tamponadora muy baja y las variaciones de pH se producen rápidamente.

Dado que se tiene por tanto una medida óptima de pH y una relación directa entre pH requerido por los cultivos y pH que proporcionan los abonos líquidos, se puede regular el pH del suelo conforme se vaya acidificando o se estime que hay que introducir nitratos como abono.

2.3 Redes inalámbricas

El medio aéreo definido como canal anteriormente, se utiliza para las diversas tecnologías inalámbricas detalladas a continuación.

2.3.1 Zigbee

ZigBee [2], es una tecnología inalámbrica con velocidades comprendidas entre 20 kB/s y 250 kB/s y rangos de 10 m a 75 m. Puede usar las bandas libres ISM de 2,4 GHz, 868 MHz (Europa) y 915 MHz (EEUU).

Los productos ZigBee trabajan en una banda de frecuencias que incluye la 2.4 Ghz (mundial), de 902 a 928 Mhz (en Estados Unidos) y 866Mhz (en Europa). La transferencia de datos de hasta 250 Kbps puede ser realizada en la banda de 2.4Ghz (16 canales), hasta 40kps en 915Mhz (10 canales) y a 20kbps en la de 868Mhz (un solo canal). Al igual que WiFi, ZigBee usa la DSSS (secuencia directa de espectro ensanchado) en la banda 2.4 Ghz. En las bandas de 868 y 900Mhz también se utiliza la secuencia directa de espectro ensanchado pero con modulación de fase binaria.

ZigBee es un sistema ideal para redes domóticas, específicamente diseñado para reemplazar la proliferación de sensores/actuadores individuales. ZigBee fue también creado para cubrir la necesidad del mercado de un sistema a bajo coste, un estándar para redes Wireless de pequeños paquetes de información, bajo consumo, seguro y fiable

2.3.1.1 Tipos de dispositivos en una red Zigbee

Los dispositivos involucrados pueden adoptar diferentes tipos de roles:

- El coordinador de red, que mantiene en todo momento el control del sistema. Es el más sofisticado de los tipos de dispositivos, requiere memoria y capacidad de computación.

- El dispositivo de función completa (FFD) capaz de recibir mensajes del estándar 802.15.4. Este puede funcionar como un coordinador de red. La memoria adicional y la capacidad de computar, lo hacen ideal para hacer las funciones de *Router* o para ser usado en dispositivos de red que actúen de *interface* con los usuarios.
- El dispositivo de función reducida (RFD) de capacidad y funcionalidad limitadas (especificada en el estándar) para el bajo coste y simplicidad. Son los sensores/actuadores de la red.

Las redes ZigBee pueden usar el entorno 'con balizas' o 'sin balizas'. Las balizas son usadas para sincronizar los dispositivos de la red, identificando la red doméstica, y describiendo la estructura de la 'supertrama'. Los intervalos de las balizas son determinados por el coordinador de red y pueden variar desde los 15msg hasta los 4 minutos.

2.3.1.2 Consumo de energía

Una red ZigBee puede estar formada por hasta 255 nodos los cuales tienen la mayor parte del tiempo el transceptor ZigBee dormido con objeto de consumir menos que otras tecnologías inalámbricas. El objetivo, es que un sensor equipado con un transceptor ZigBee pueda ser alimentado con dos pilas AA durante al menos 6 meses y hasta 2 años (siendo este consumo bajo especialmente interesante para el estudio en el medio rural)

Las redes ZigBee son diseñadas para conservar la potencia en los nodos 'esclavos'. Durante mucho tiempo, un dispositivo 'esclavo' está en modo 'dormido' y sólo de 'despierta' por una fracción de segundo para confirmar que está 'vivo' en la red de dispositivos. Por ejemplo, la transición del modo 'dormido' al modo 'despierto' (cuando transmite) dura unos 15ms y la enumeración de 'esclavos' dura unos 30ms.

2.3.1.3 Estructura interna en capas

Siguiendo el estándar del modelo de referencia OSI, en el gráfico posterior, aparece la estructura interna de Zigbee en capas:

| |
|-----------------------------------|
| Objetos – Dispositivos Zigbee |
| GOF - Trama operaciones generales |
| Capa de Red |
| MAC (IEEE 802.15.4) |
| Física (IEEE 802.15.4) |

Tabla 4 - Estructura interna en capas

Las primeras dos capas, la física (PHY) y la de acceso al medio (MAC), son definidas por el estándar IEEE 802.15.4. Las capas superiores son definidas por la Alianza ZigBee. El grupo de trabajo de IEEE pasó el primer borrador de la capa física y la de acceso al medio en 2003. Se define una versión final de la capa de red (NWK) y en Junio de 2005 se obtiene ya un Zigbee 1.0 publico.

La trama general de operaciones (GOF) es una capa que existe entre la de aplicaciones y el resto de capas. La GOF suele cubrir varios elementos que son comunes a todos los dispositivos, como el subdireccionamiento y los modos de direccionamientos y la descripción de dispositivos, como el tipo de dispositivo, potencia, modos de 'dormir' y coordinadores de cada uno. Utilizando un modelo, la GOF especifica métodos, eventos, y formatos de datos que son utilizados para constituir comandos y las respuestas a los mismos

La capa de red (NWK) une o separa dispositivos a través del controlador de red, implementa seguridad, y encamina tramas a sus respectivos destinos. Además, la capa de red del controlador de red es responsable de crear una nueva red y asignar direcciones a los dispositivos de la misma.

2.3.1.4 Campos de los tipos de paquetes básicos

La siguiente figura muestra los campos de los cuatro tipos de paquetes básicos: datos, ACK, MAC y baliza:

| |
|-------------------|
| Trama DATOS |
| Trama ACK |
| Trama MAC Comando |
| Trama baliza |
| Trama MAC |
| Capa física |

Tabla 5 - Tramas de red

El paquete de datos tiene una carga de datos de hasta 104 bytes. La trama está numerada para asegurar que todos los paquetes llega y se incluye un campo que asegura que el paquete se ha recibido sin errores. Esta estructura aumenta la fiabilidad en condiciones complicadas de transmisión.

Otra estructura importante es la de ACK, o reconocimiento. Esta trama es una realimentación desde el receptor al emisor, para confirmar que el paquete se ha recibido sin errores. Se puede incluir un 'tiempo de silencio' entre tramas, para enviar un pequeño paquete después de la transmisión de cada paquete.

El paquete MAC, se utiliza para el control remoto y la configuración de dispositivos/nodos. Una red centralizada utiliza este tipo de paquetes para configurar la red a distancia.

Para acabar, el paquete baliza 'despierta' los dispositivos, que escuchan y luego vuelven a 'dormirse' si no reciben nada más. Estos paquetes son importantes para mantener todos los dispositivos y los nodos sincronizados, sin tener que gastar una gran cantidad de batería estando todo el tiempo encendidos.

Dos mecanismos de acceso al canal se implementan en 802.15.4. Para una red 'sin balizas', un estándar ALOHA CSMA-CA envía reconocimientos positivos para paquetes recibidos correctamente. En una red 'con balizas', una estructura de 'supertrama' se usa para controlar el acceso al canal. La supertrama es estudiada por el coordinador de red para transmitir 'tramas baliza' cada ciertos intervalos (múltiples cada de 15.38 msg, hasta cada 252 sg). Esta estructura garantiza el ancho de banda dedicado y bajo consumo.

Los dispositivos se direccionan empleando 64-bits y un direccionamiento corto opcional de 16 bits. El campo de dirección incluido en MAC puede contener información de direccionamiento de ambos orígenes y destinos (necesarios para operar punto a punto). Este doble direccionamiento es usado para prevenir un fallo dentro de la red.

2.3.1.5 Configuraciones de red

La capa de red soporta múltiples configuraciones de red incluyendo estrella, árbol, y rejilla, como se muestra en el gráfico 3.

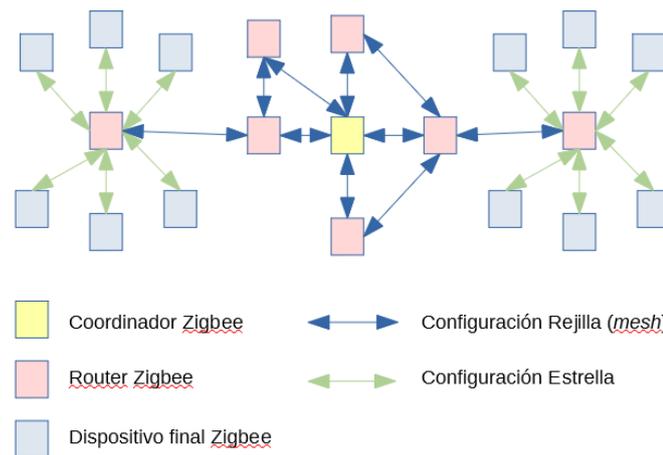


Gráfico 3 - Configuraciones de red

En la configuración en estrella, uno de los dispositivos tipo FFD asume el rol de coordinador de red y es responsable de inicializar y mantener los dispositivos en la red. Todos los demás dispositivos ZigBee, conocidos con el nombre de dispositivos finales, ‘hablan’ directamente con el coordinador. En la configuración de rejilla, el coordinador ZigBee es responsable de inicializar la red y de elegir los parámetros de la red, pero la red puede ser ampliada a través del uso de *routers* ZigBee. El algoritmo de encaminamiento utiliza un protocolo de pregunta-respuesta (*request-response*) para eliminar las rutas que no sean óptimas, La red final puede tener hasta 254 nodos, siendo difícil tener que necesitar tantos en una única red.

2.3.1.6 Seguridad

ZigBee utiliza el modelo de seguridad de la subcapa MAC IEEE 802.15.4, la cual especifica 4 servicios de seguridad.

- Control de accesos. El dispositivo mantiene una lista de los dispositivos ‘comprobados’ en la red.
- Datos Encriptados, los cuales usan una encriptación con un código de 128 bits.
- Integración de tramas para proteger los datos de ser modificados por otros.
- Secuencias de refresco, para comprobar que las tramas no han sido reemplazadas por otras. El controlador de red comprueba estas tramas de refresco y su valor, para ver si son las esperadas.

Dependiendo del dispositivo final creado se realizará la decisión de dotarlo de más o menos seguridad.

2.4.2 4G

La cuarta generación de tecnologías móviles se basa totalmente en IP [3], definiéndose dos estándares: LTE y IEEE 802.16, que recoge el estándar de facto WiMAX.

Sobre este último, en el momento de la redacción de esta memoria, se considera una tecnología obsoleta. WiMAX es una opción muy interesante para cubrir extensiones de terreno en bandas no licenciadas pero ha ido en detrimento en favor de LTE.

LTE, *Long Term Evolution*, es un estándar de la norma 3GPP como fruto de la evolución de UMTS debido a sus limitaciones. En lugar de una única transmisión, los datos se dividen en muchos flujos más lentos y son transmitidos simultáneamente empleándose para ello OFDM (*Orthogonal Frequency Division Multiplexing*). Para aumentar la velocidad de transmisión se aumenta el canal incrementando el número de portadoras de banda estrecha. LTE se puede adaptar fácilmente a un ancho de banda menor de 5 MHz. Qué ancho de banda se emplee dependerá de la banda de frecuencias disponible y de la cantidad de espectro posible.

Las especificaciones se detallan en la siguiente tabla:

| | LTE (3GPP R8) | LTE-Advanced (3GPP R10) |
|--------------------------|---|---|
| Capa física | DL: OFDMA UL: SC-FDMA | DL: OFDMA UL: SC-FDMA |
| Modo dúplex | FDD Y TDD | FDD Y TDD |
| Movilidad de usuario | 350 km/h | 350 km/h |
| Ancho de Banda del canal | 1.4, 3, 5, 10, 15, 20 MHz | 1.4, 3, 5, 10, 15, 20 MHz |
| Bitrate máximo | DL: 302 Mbps (MIMO 4x4) UL: 75 Mbps (MIMO 2x4) con 20 MHz FDD | DL: 1 Gbps UL: 300 Mbps |
| Eficiencia espectral | DL: 1.91 bps/Hz UL: 0.72 bps/Hz | DL: 30 bps/Hz UL: 15 bps/Hz |
| Latencia | Capa de enlace < 5 ms Traspaso < 50 ms | Capa de enlace < 5 ms Traspaso < 50 ms |
| Capacidad VoIP | 80 usuarios por sector/MHz (FDD) | > 80 usuarios por sector/MHz (FDD) |

Tabla 6 - Especificaciones LTE

2.4.2.1 LTE vs UMTS

Además del ancho de banda flexible, LTE soporta transmisiones MIMO, que permite a la estación base transmitir varios flujos de datos sobre la misma portadora.

Para separar el canal ascendente del descendente se usa principalmente FDD, aunque también se encuentra disponible en TDD aunque no es tan popular.

El otro gran cambio es la aproximación todo IP. LTE se basa en un núcleo de red IP a excepción del SMS, que se transporta mediante los canales de señalización. Adicionalmente todas las interfaces de red entre los nodos están basadas en IP, lo cual es una gran simplificación comparada con las anteriores tecnologías y más barata.

Para lograr ser universal, los dispositivos LTE deben de ser compatibles con anteriores generaciones, lo que significa compatibilidad con GSM, GPRS, EDGE y UMTS. La arquitectura de red LTE es similar a GSM y UMTS. En principio, la red se encuentra separada entre la red radio y la parte del núcleo de red. No obstante, el número de nodos lógicos ha sido reducido.

2.4.2.2 Bandas de la red LTE

La mayoría de dispositivos LTE también soportan GSM y UMTS, por lo que también pueden emplear las otras bandas de frecuencia. Por ejemplo, un móvil vendido en Europa soporta las bandas de 900 MHz y 1800 MHz para GSM, 2100 MHz para UMTS y las bandas de 850 MHz y 1900 MHz para la itinerancia o roaming internacional GSM y UMTS, respectivamente.

Las bandas de la red LTE actualmente, son las siguientes [17]:

| Banda | Rango de frecuencias | Ancho de banda | Agregación de portadoras |
|-------|----------------------|----------------|---------------------------|
| 33 | 1900 MHz - 1920 MHz | 20 MHz | |
| 34 | 2010 MHz - 2025 MHz | 15 MHz | |
| 35 | 1850 MHz - 1910 MHz | 60 MHz | |
| 36 | 1930 MHz - 1990 MHz | 60 MHz | |
| 37 | 1910 MHz - 1930 MHz | 20 MHz | |
| 38 | 2570 MHz - 2620 MHz | 50 MHz | Intrabanda contigua CA_38 |
| 39 | 1880 MHz - 1920 MHz | 40 MHz | |
| 40 | 2300 MHz - 2400 MHz | 100 MHz | Intrabanda contigua CA_40 |
| 41 | 2496 MHz - 2690 MHz | 194 MHz | Intrabanda contigua CA_41 |
| 42 | 3400 MHz - 3600 MHz | 200 MHz | |

| | | | |
|----|---------------------|---------|--------------------------------|
| 43 | 3600 MHz - 3800 MHz | 200 MHz | |
| 44 | 703 MHz - 803 MHz | 100 MHz | |
| 46 | 5150 MHz - 5925 MHz | 775 MHz | Banda sin licencia restringida |

Tabla 7 - Bandas LTE

El proyecto presentado, se centra en la transmisión y recepción mediante la banda libre 46 (LTE-U). Su uso implica prescindir de los servicios de una operadora de telecomunicaciones, con el ahorro de costes consecuente.

2.4.2.3 E-UTRAN y eNodeB

E-UTRAN [4] es la red de acceso especificada para LTE, que utiliza la tecnología OFDMA en la interfaz radio para la comunicación con los equipos de usuario.

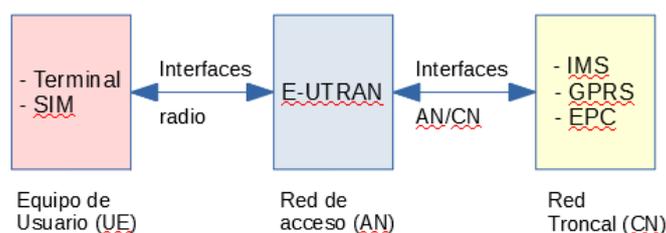


Gráfico 4 - E-UTRAN

La red de acceso E-UTRAN y la red troncal EPC proporcionan de forma conjunta servicios de transferencia de paquetes IP entre los equipos de usuario y redes de paquetes externas tales como plataformas IMS y/o otras redes de telecomunicaciones como Internet. Las prestaciones de calidad de servicio (e.g., tasa de datos en bits/s, comportamientos en términos de retardos y pérdidas) de un servicio de transferencia de paquetes IP puede configurarse en base a las necesidades de los servicios finales que lo utilicen, cuyo establecimiento (señalización) se lleva a cabo a través de plataformas de servicios externas (e.g., IMS) y de forma transparente a la red troncal EPC.

Formalmente, el servicio de transferencia de paquetes IP ofrecido por la red LTE entre el equipo de usuario y una red externa se denomina servicio portador EPS (EPS *Bearer Service*). Asimismo, la parte del servicio de transferencia de paquetes que proporciona la red de acceso E-UTRAN se denomina E-UTRAN *Radio Access Bearer* (ERAB).

La infraestructura de una red LTE, además de los equipos propios que implementan las funciones del estándar 3GPP, también integra otros elementos de red propios de las redes IP tales como *routers*, servidores DHCP para la configuración automática de las direcciones IP de los equipos de la red LTE y servidores DNS para asociar los nombres de los equipos con sus direcciones IP.

La arquitectura de la red de acceso se compone de una única entidad de red denominada *evolved NodeB* (eNodeB) que constituye la estación base de E-UTRAN. Así pues, la estación base E-UTRAN integra toda la funcionalidad de la red de acceso, a diferencia de las redes de acceso de GSM y UMTS compuestas por estaciones base (BTS, NodoB) y equipos controladores (BSC y RNC). La descripción de la arquitectura de E-UTRAN se detalla en las especificaciones del 3GPP TS 36.300 y TS 36.401.

2.4.2.4 Componentes de la red LTE

La siguiente figura muestra un resumen de la red LTE y sus componentes:

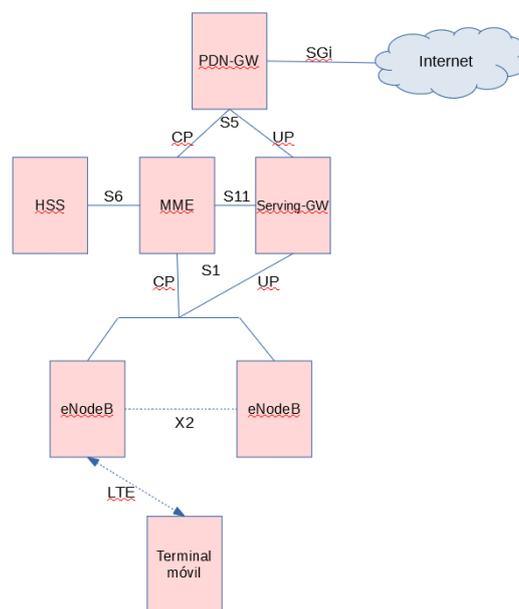


Gráfico 5 - Componentes LTE

El nodo-B o eNode-B, es el dispositivo más complejo en la red LTE. Consiste en tres grandes elementos:

- Las antenas
- Los módulos de radio
- Los módulos digitales, que procesan las señales radio y actúan como interfaz hacia el núcleo de la red.

Además, las estaciones base en LTE son unidades autónomas. Es decir, integran la mayoría de las funcionalidades que anteriormente formaban parte del controlador radio de la red. Por lo tanto el eNode-B es responsable de:

- La interfaz aérea
- Organización de usuarios y programación de los recursos radio
- Asegurar un QoS y requisitos mínimos dependiendo del perfil del usuario.

- Balancear la carga entre los diferentes usuarios que emiten simultáneamente.
- Lograr la movilidad.
- Reducir las interferencias con las células vecinas.

En cuanto al resto de elementos en la arquitectura de red, se tiene:

La interfaz S1, dividida en dos partes lógicas, el plano de usuario UP y plano de control CP. UP se encarga de los datos de usuario y CP se encarga de la interacción con el núcleo de la red y de la señalización.

Mobility Management Entity (MME), que es el nodo responsable de toda la señalización entre las estaciones base y el núcleo de la red y entre los usuarios y núcleo.

Serving Gateway (S-GW): Es el encargado del túnel de los datos de usuarios entre el eNodo-B y el PDN-GW.

El Packed Data Network Gateway (PDN-GW): Es la puerta de enlace entre Internet y el operador de red.

Home Subscriber Server (HSS): Se refiere al Home Location Register (HLR) y al Mobile Application Part (MAP) en los sistemas GSM y UMTS. Cada cliente se encuentra registrado en esta base de datos.

2.4.2.5 Enlace descendente

Para el enlace descendente LTE emplea Orthogonal Frequency Division Multiple Access (OFDMA). La información a transmitir es dividida entre varios flujos más lentos y se envían mediante portadoras simultáneas. La ventaja del envío simultáneo en paralelo puede ser mayor que la desventaja del efecto multipath.

La siguiente tabla muestra el número de subportadoras utilizadas dependiendo del ancho de banda. A mayor ancho de banda, mayor número de subportadoras es necesaria.

| Ancho de banda (MHz) | Número de subportadoras | Tamaño de la FFT |
|----------------------|-------------------------|------------------|
| 1.25 | 76 | 128 |
| 2.5 | 150 | 256 |
| 5 | 300 | 512 |
| 10 | 600 | 1024 |
| 15 | 900 | 1536 |
| 20 | 1200 | 2048 |

Gráfico 6 - Relación ancho de banda y subportadoras

Para ahorrar ancho de banda, las subportadoras están espaciadas de tal forma que el lóbulo lateral del espectro de cada una de ellas es exactamente cero en la frecuencia central de la subportadora adyacente (ortogonales). El siguiente gráfico lo muestra conceptualmente:

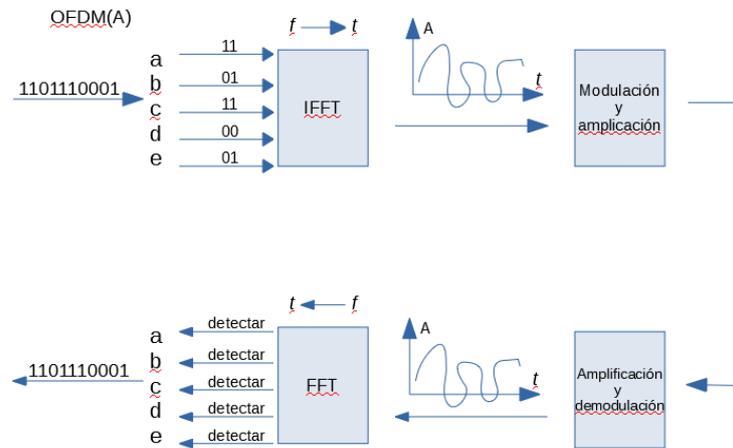


Gráfico 7 - Enlace descendente

Arriba a la izquierda los datos se ponen en flujos paralelos, cada uno de los cuales se mapea en su correspondiente subportadora. A continuación se emplea la transformada inversa de Fourier para trasladar la señal al dominio del tiempo antes de ser modulada y enviada a través del aire al receptor. La recepción es mostrada en la parte baja del gráfico. Después de la demodulación de la señal se le aplica la transformada de Fourier FFT para convertirla al dominio de la frecuencia de nuevo para poder separar cada una de las subportadoras. Finalmente todos los flujos son unidos de nuevo en uno solo y es entregado a la capa superior de la pila de protocolos.

Los parámetros físicos para las subportadoras son:

- Espaciado entre subportadoras: 15 KHz
- Duración de símbolo (OFDM): 66.667 μ s
- Prefijo estándar cíclico: 4.7 μ s. Este prefijo es transmitido antes de cada símbolo para prevenir interferencia intersimbólica debido a las diferentes longitudes de cada camino.

2.4.2.6 Enlace ascendente

Para el enlace ascendente OFDMA no es apropiado debido a su alto pico de consumo medio o *Peak to Average Power Ratio* (PAPR) cuando los diferentes flujos son combinados. Se utiliza *Single-Carrier Frequency Division Multiple Access* (SC-FDMA). Es similar a OFDMA pero tiene pasos adicionales en el proce-

samiento. En vez de dividir los datos en varios flujos y mapearlos en subportadoras, la señal en el dominio del tiempo es convertida a una en el dominio de la frecuencia mediante una función Transformada de Fourier. Esto distribuye la información de cada bit en todas las subportadoras, reduciendo así el consumo entre las diferentes subportadoras. El número de subportadoras depende de las condiciones de la señal, la potencia de transmisión del dispositivo y el número de usuarios simultáneos en el enlace ascendente.

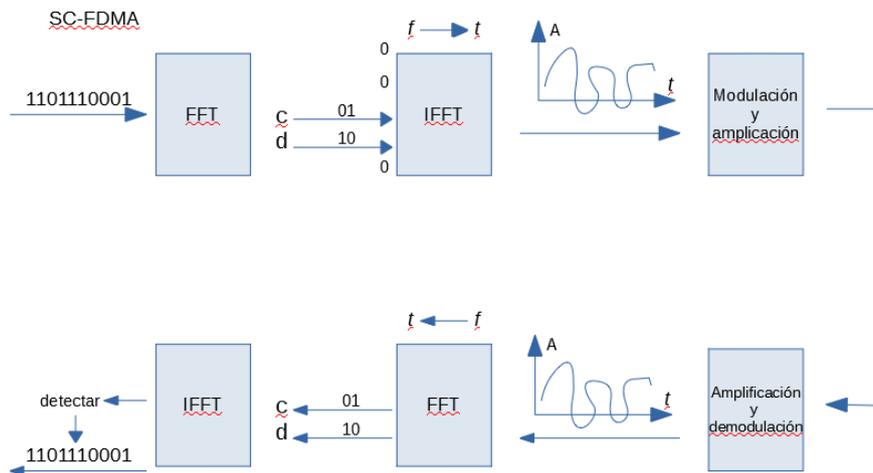


Gráfico 8 - Enlace ascendente

En el gráfico se puede observar la diferencia de pasos con el enlace descendente, que empleaba OFDMA. En lugar de dividir primero el flujo y ponerlo en diferentes flujos la señal se convierte al dominio de la frecuencia mediante una transformada de Fourier. De esta forma la información se distribuye a cada una de las subportadoras.

2.4.2.7 Transmisión de datos

La transmisión de datos se organiza de la siguiente manera: la unidad de transmisión más pequeña en cada subportadora es una transmisión con una duración de 66.667 μ s correspondiente a un símbolo. El número de bits por símbolo dependerá de la codificación escogida y esta a su vez de las condiciones del entorno. Si las condiciones de radio son ideales se empleará 64-QAM y se enviarán 6 bits por símbolo. En peores condiciones de entorno emplearemos 16-QAM (4 bits por símbolo) o QPSK (2 bits por símbolo). A continuación se le añade una cabecera para asignar cada símbolo individual a cada usuario o propósito. Siete símbolos consecutivos en 12 subportadoras son agrupados en un *Resource Block* (RB). Un RB ocupa exactamente un slot con una duración de 0.5 ms.

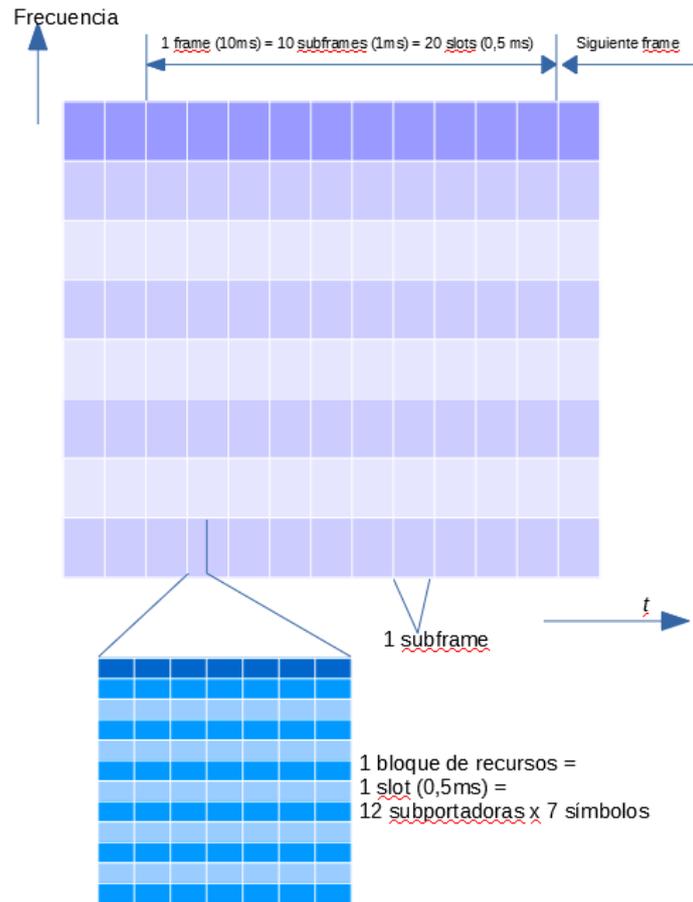


Gráfico 9 - Símbolos, slots, bloques radio y tramas

Una subtrama representa el esquema de tiempo de LTE y está formada por 2 slots con una duración de 1 ms. Esto quiere decir que en cada milisegundo el eNodo-B decide a cuántos usuarios están permitidos y cuántos recursos se les asigna a cada uno de ellos.

La red tiene dos opciones para transmitir una subtrama. La primera opción es mediante *Localized Virtual Resource Blocks* (LVRBs), transmitidos en grupos como muestra la anterior figura. En este caso el eNodo-B necesita un canal de realimentación de banda estrecha para organizar los bloques en subportadoras y evitar que sufran desvanecimiento en banda estrecha. La segunda opción es transmitir los datos mediante *Distributed Virtual Resource Blocks* (DVRBs), donde los símbolos que forma un bloque se encuentran dispersos por todo el ancho de banda. En este caso el móvil no tiene ningún canal de realimentación. Y finalmente 10 subtramas se combinan en una trama de radio LTE, la cual tiene una duración de 10 ms.

2.4.2.8 Canales

Todo el tráfico de señalización y los datos de usuario están organizados en canales.

En la capa lógica los datos de cada usuario se transmiten mediante el *Dedicated*

Traffic Channel (DTCH). Cada usuario tiene un DTCH. No obstante, en la interfaz aérea todos los canales son mapeados en un único canal compartido que ocupa todos los bloques. Algunos símbolos en cada bloque de recursos se asignan a otros propósitos y no puede ser usado para datos. El mapeo de los DTCH hacia el canal único compartido se realiza en dos pasos. Primero, los DTCH se mapean en el *Downlink Shared Channel* (DL-SCH). En el segundo paso este flujo de datos se mapea al canal físico o *Physical Downlink Shared Channel* (PDSCH).

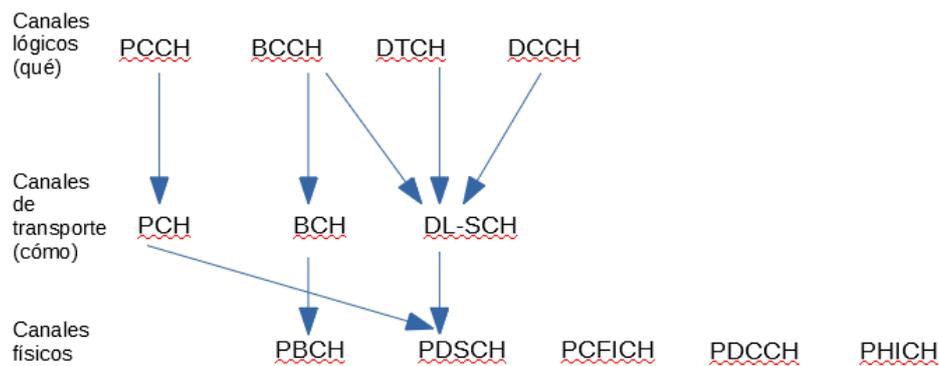


Gráfico 10 - Canales enlace descendente

Además de los canales de datos se le añaden los canales lógicos antes de ser mapeados al canal físico. Todos los flujos de canales superiores se mapean en el canal físico compartido, incluyendo el canal de control de paginación o *Paging Control Channel* (PCCH). El único canal que no se mapea en el compartido junto a los otros es el que se encarga de la sincronización con la celda. Este canal es el *Physical Broadcast Channel* (PBCH). Es transmitido cada 40 ms y contiene las señales de sincronización primaria y secundaria.

En el canal ascendente se emplea un modelo de canal similar al de la dirección descendente. También se utilizan canales lógicos, de transporte y físicos para separar los flujos de datos lógicos y multiplexarlos en un único canal. El canal más importante es el *Physical Uplink Shared Channel* (PUSCH), cuya tarea es transportar datos de usuario, señalización e información sobre la calidad de la señal.

Los datos del canal PUSCH están divididos en tres canales lógicos: el canal de los datos de usuario *Dedicated Traffic Channel* (DTCH), el de señalización *Dedicated Control Channel* DCCH y el de señalización en el establecimiento que es el *Common Control Channel* CCCH.

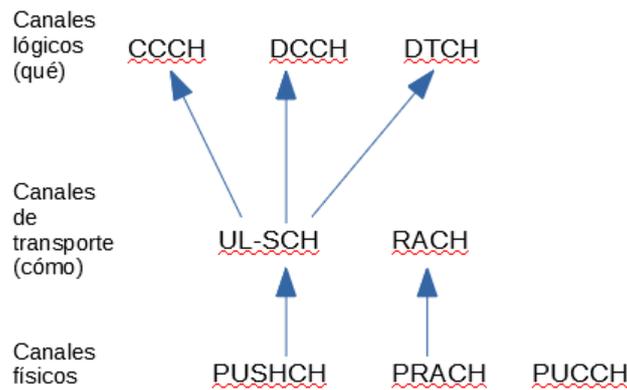


Gráfico 11 - Canales enlace ascendente

2.4.2.9 MIMO

Aparte de los esquemas de modulación mejorados, a partir del 3GPP *Release 8* se especifica y requiere el uso de las técnicas multiantena, mejor conocidas como MIMO (*Multiple Input Multiple Output*) en la dirección descendente. A pesar de que se especificó en HSPA no fue ampliamente utilizada debido a la escasa compatibilidad con los dispositivos del momento. El concepto principal en MIMO es el de enviar varios flujos de datos independientes sobre la misma interfaz aérea simultáneamente.

En su primera versión se han especificado el uso de 2 ó 4 flujos simultáneos. Solamente se emplea en el canal compartido y únicamente para transmitir en aquellos usuarios que posean una muy buena calidad de señal. La transmisión de flujos simultánea sobre el mismo canal es posible solo si los distintos flujos se mantienen independientes durante el recorrido a través del canal.

Esto se logra cumpliendo dos requisitos básicos:

- En el lado del transmisor, si se quieren transmitir dos flujos distintos, se necesitan dos antenas. En la práctica se puede hacer empleando una sola antena y distinta polarización para cada uno de los flujos (horizontal y vertical). El receptor también requiere dos o cuatro antenas.

- El segundo requisito para la transmisión MIMO es que la señal tiene que permanecer independiente durante el recorrido entre transmisor y receptor. Esto se puede conseguir por ejemplo si las transmisiones simultáneas alcanzan el dispositivo móvil mediante distintos caminos. De todas formas y como es normal, las transmisiones simultáneas interfieren entre sí lo que reduce la máxima velocidad.

Si la interferencia es muy fuerte se tendrá que bajar el esquema de modulación y se empleará 16-QAM en vez de 64-QAM, y entonces puede que interese más

emplear un único canal con 64-QAM que dos que se interfieran con 16-QAM cada uno. Por lo tanto sólo en condiciones muy ideales y distancias muy cortas entre transmisión y receptor se podrá utilizar un esquema 64-QAM y MIMO que garantiza el máximo ancho de banda.

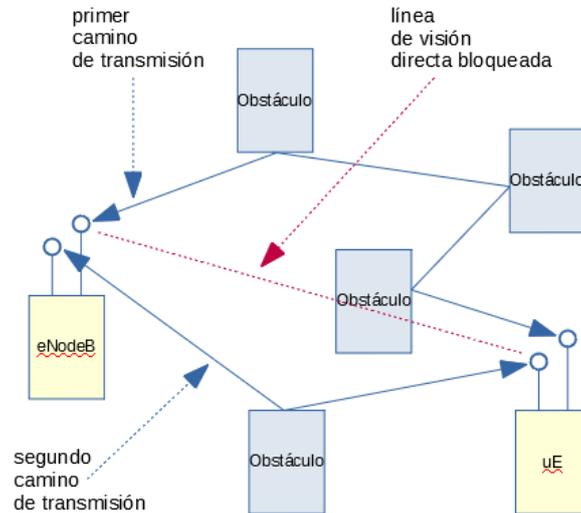


Gráfico 12 - MIMO

En las especificaciones de LTE empleamos el término de rangos para describir el uso de MIMO. El rango 1 se especifica para una única transmisión y el rango 2 para una transmisión MIMO de dos flujos para un bloque de datos. En la estación base o eNodo-B podemos encontrar dos modos de funcionamiento para MIMO:

- Modo de operación en bucle cerrado, donde una matriz de precodificación es aplicada en el flujo de datos antes de ser enviada para cambiar la modulación de las dos señales con el fin de aumentar el rendimiento en el receptor.
- Modo de operación en bucle o lazo abierto, para usuarios en movimiento, donde adaptar su transmisión a una matriz de precodificación no es lo suficientemente rápida. En este caso únicamente el móvil decide las características de la transmisión.

Por último también cabe destacar que MIMO también aporta fiabilidad. Puede que la transmisión MIMO no aporte más velocidad pero al enviar varios flujos ayuda al receptor en la decodificación.

En total hay 12 modos de uso de MIMO, de forma que el eNodo-B tiene un amplio rango de opciones para adaptarse a unas condiciones de señal cambiantes.

| Modo de transmisión | Esquema de transmisión del PDSCH |
|---------------------|---|
| Modo 1 | Una antena |
| Modo 2 | Diversidad en la transmisión |
| Modo 3 | Diversidad en la transmisión o gran retraso en el CDD |
| Modo 4 | Diversidad en la transmisión o MIMO en bucle cerrado |
| Modo 5 | Diversidad en la transmisión o multiusuario MIMO |
| Modo 6 | Diversidad en la transmisión o MIMO en bucle cerrado |
| Modo 7 | Una antena |

Tabla 8 - Modos de uso de MIMO

2.4.2.10 LTE Advanced

Finalmente, se hace una referencia a *LTE Advanced*. Implica una serie de funcionalidades y características que mejoran la red LTE y se encuentran incluidas en *3GPP Release 10*. El principal objetivo de estas mejoras es reducir aún más el coste e incrementar las capacidades máximas por usuario. Las principales mejoras relacionadas con la capa física son las siguientes:

- Se incrementa el ancho de banda agregando una nueva portadora al actual ancho de banda de 20 MHz. Esta agregada se emplea para combinar la capacidad de varias portadoras individuales. Las portadoras pueden estar adyacentes o no, en una única banda y también en diferentes bandas. Una configuración puede ser combinar las bandas LTE 7 (2600 MHz) y 3 (1800 MHz) para conseguir un ancho de banda total de 20 MHz.
- Mejoras en la transmisión MIMO. Se añade el modo de transmisión 8x8 (tasas de hasta 1Gbps) y en el enlace ascendente se añade hasta MIMO 4x4, obteniendo un rendimiento de hasta 300Mbit/s
- Para aumentar el rendimiento y ampliar la cobertura se pueden emplear femtocélulas, que son pequeñas y baratas. También se pueden utilizar nodos de retransmisión. Estos nodos funcionan como células estándares pero a diferencia de las celdas normales que emplean el cobre, fibra óptica o microondas para la conexión con el núcleo de la red emplean la propia interfaz LTE para transportar esos datos hacia otras células y así ampliar la cobertura disponible.

2.5 Porqué LTE y Zigbee

Ateniéndose al proyecto, dado el medio rural donde un cableado es inviable, es necesario recurrir a tecnologías inalámbricas para la recepción y envío de información.

Presentadas ambas tecnologías inalámbricas en los apartados anteriores, se detalla un breve resumen que indica los principales motivos por los que se han elegido:

| LTE |
|--|
| Permite tasas de bits altas con bajas latencias. |
| Alta eficiencia espectral |
| Multi-antena de aplicación |
| Arquitectura de simple protocolo |
| Es posible transmitir a largas distancias con una estación base |
| Permite utilizar bandas libres sobre 5GHz, prescindiendo de costes adicionales |
| Es relativamente barato considerando la cantidad de terreno a cubrir. |
| Permite tasas de bits altas con bajas latencias. |
| Zigbee |
| Óptimo para redes de baja tasa de transferencia de datos. Útil para trabajar con sensores |
| Reduce tiempos de espera en el envío y recepción de paquetes. |
| Soporte para múltiples topologías de red: Estática, dinámica, estrella y malla. |
| Hasta 65.000 nodos en una red. |
| Opera en la banda libre de 2.4 GHz para conexiones inalámbricas. |
| Es muy eficiente a nivel energético ayudando a ser eficientes en el manejo de los sensores |
| Ideal para conexiones punto a punto y punto a multipunto |
| Es muy barato, contribuyendo a mantener los costes reducidos. |
| No se depende de terceros que incrementan los costes de la infraestructura. |

Tabla 9 - Porqué LTE y Zigbee

2.6 Atenuaciones y obstáculos

Las tecnologías inalámbricas sufren de problemas relacionados con la atenuación conforme incrementa la distancia y aparecen múltiples obstáculos en el camino que une al transmisor y al receptor.

En el caso de estudio, además de la atenuación producida por la distancia, los propios cultivos o los diversos obstáculos derivados de un terreno irregular, afectarán a la calidad de la señal recibida. Es por ello que cuando se transmite inalámbricamente, tanto en el caso de Zigbee como en el de LTE, hay que tener en cuenta las atenuaciones producidas en la señal.

2.6.1 Cálculo de la atenuación en medio libre

Dado que se van a considerar enlaces punto a punto, se puede calcular la atenuación en espacio libre [5] entre antenas isótropas (antenas que distribuyen la potencia de manera uniforme en todas las direcciones), de la siguiente manera:

$$L_{bf} = 20 \log\left(\frac{4\pi d}{\lambda}\right) \text{ dB}$$

Siendo:

L_{bf} : pérdida básica de transmisión en espacio libre (dB)

d : distancia

λ : longitud de onda

2.6.2 Propagación sobre espacio libre

Con este método [6] se considera que la potencia recibida P_R depende de la potencia transmitida P_T , de la ganancia del transmisor G_T y del receptor G_R , de la distancia d y de la longitud de onda de la señal ($\lambda = c/f$).

$$P_R = P_T \cdot G_T \cdot G_R \cdot \left(\frac{\lambda}{4\pi d}\right)^2$$

En dB, siendo $G_R[\text{dB}] = 10 \log G_R[\text{lin}]$:

$$P_R [dBm] = P_T [dBm] + G_T [dB] + G_R [dB] - 20 \log\left(\frac{4\pi d}{\lambda}\right)$$

2.6.3 Modelo Erceg

Dado que se pretende hacer un estudio genérico donde se desconocen los obstáculos en la extensión de terreno del proyecto, se utiliza este método ya que permite realizar un estudio simulado de una manera que se considera relativamente precisa.

Se trata de un método de la *Stanford University* que trabaja con varios tipos de canales: SUI-1 y SUI-2 para terrenos planos, SUI-3 y SUI-4 para terrenos con algunos obstáculos y SUI-5 y SUI-6 para lugares con muchos obstáculos y vegetación frondosa.

Está pensando para frecuencias inferiores a 11 Ghz y por lo tanto es útil para el sistema diseñado. Es válido para medios rurales, suburbanos y urbanos para alturas de móvil h_m entre 2 y 10 m, alturas de base h_b entre 10 y 80 m y distancias entre 100m y 8km, siendo muy interesante para el proyecto.

$$L [dB] = 20 \log\left(\frac{4 \cdot \pi \cdot 100}{\lambda}\right) + 10 \cdot \left(a - b \cdot h_b + \frac{c}{h_b}\right) \cdot \log\left(\frac{d}{100}\right) + 6 \cdot \log\left(\frac{f}{2.000}\right) + X_h + s$$

Siendo: X_h :

→ $-10.8 \cdot \log\left(\frac{h_m}{2}\right)$ para entornos A y B

→ $-20 \cdot \log\left(\frac{h_m}{2}\right)$ para entornos C

Donde d está en metros, f en MHz y h_b y h_m en metros.

a , b , c y s dependen del tipo de entorno según la siguiente tabla:

| | a | b | c | s |
|---------------------------------|-----|--------|------|------|
| Medio A (canales SUI-5 y SUI-6) | 4,6 | 0,0075 | 12,6 | 10,6 |
| Medio B (canales SUI-3 y SUI-4) | 4 | 0,0065 | 17,1 | 9,6 |
| Medio C (canales SUI-1 y SUI-2) | 3,6 | 0,005 | 20 | 8,2 |

Tabla 10 - Parámetros a , b , c y s , según el medio

2.7 Interconexión de sensores a través de topologías de red de malla

En este apartado se realiza una descripción teórica de las topologías de red de malla. Dado que se diseñan redes de sensores donde se recoge la información a través de Zigbee, siendo Zigbee compatible con diversas topologías de red, es necesario dar una pequeña descripción que indique porqué se utiliza esta topología de red para el envío y recepción de información desde y hacia los sensores utilizados.

Debido al medio rural, es muy posible que los nodos vayan fallando a lo largo del tiempo, teniendo que reemplazarlos. Ocurre además que el fallo de uno de los nodos, no puede comprometer al resto de los nodos del sistema pudiendo estos nodos estar en lugares de difícil acceso. Por todo ello se antoja necesario utilizar una topología de red que permita enviar y recoger la información introduciendo la redundancia de caminos adecuada hacia los dispositivos finales.

Una topología de red de malla (*mesh*), es un diseño descentralizado en el cual cada nodo de la red se conecta al menos a otros dos nodos [7].

Las redes de malla pueden utilizar una topología de malla completa o una topología de malla parcial. En una topología de malla completa, cada nodo de red está conectado a todos los otros nodos en la red. En una topología de malla parcial, al menos un nodo se conecta directamente a todos los demás nodos, mientras que otros pueden solo conectarse a esos nodos con los que intercambian datos de manera frecuente.

A diferencia de los nodos en una topología en estrella, que requieren un *router* para entregar servicios de internet, los nodos de red pueden "hablar" directamente entre sí sin necesidad de la ayuda de una conexión a Internet. Una gran ventaja de esta topología descentralizada es que no puede haber un único punto de fallo (SPOF). Si un nodo ya no puede funcionar, todo el resto todavía puede comunicarse entre sí, directamente o a través de uno o más nodos intermedios.

La imagen posterior muestra una red de malla completa con seis nodos. Cada nodo se muestra como un rectángulo y las conexiones se muestran como líneas rectas.

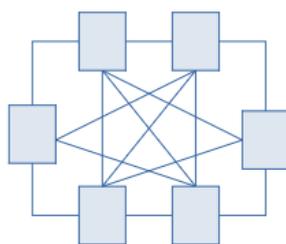


Gráfico 13 - Red de malla

La siguiente imagen muestra, por otro lado, una red de malla parcial con seis nodos. Cada nodo se muestra como un rectángulo y las conexiones se muestran como líneas rectas. Las conexiones, ya sea en una red completa o parcial, pueden ser alámbricas o inalámbricas.

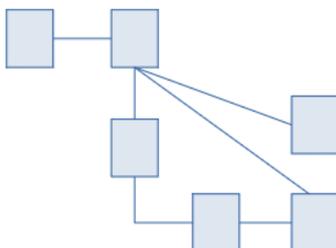


Gráfico 14 - Red de malla parcial

2.8 Software utilizado

2.8.1 Software del ordenador central

Este ordenador tendrá un sistema operativo basado en GNU/Linux. Se opta por una distribución basada en Debian [30], aunque cualquiera de las distribuciones es válida.

Se instalará un servidor de VPN de cara a montar una red privada sobre la red LTE inalámbrica y disponer de un rango de IPs privadas en los diversos dispositivos Raspberry Pi.

El servidor no estará enganchado a Internet y dado que los dispositivos Raspberry utilizarán la banda libre 4G, sólo hay que preocuparse de la seguridad a nivel local. Se utiliza inicialmente un firewall basado en *software* como puede ser *iptables*. En un futuro, implantado el proyecto, se deberá reforzar la seguridad mediante algún dispositivo *hardware*

2.8.2 Software dispositivos Raspberry Pi

Se utilizará la distribución Raspbian GNU/Linux [31] específica para Raspberry Pi, basada en Debian.

Se instalará y se arrancará un servidor *ssh* en una de las Raspberry Pi para poder administrar el dispositivo desde el ordenador central. También se introduce un cliente VPN, de cara a la comunicación por la red privada virtual. Confi-

gurada finalmente una de las Raspberry Pi, se crea una imagen de todo el *software* para llevarlo a todas las Raspberry Pi (teniendo en cuenta sólo los cambios de direccionamiento de red)

2.8.3 Software genérico

Mediante *Fritzing* [35] se pueden realizar diversos diseños relacionados con las conexiones de Raspberry Pi. Esta herramienta es interesante ya que provee a modo de representación diversas placas como Raspberry Pi y todo tipo de PCBs de dispositivos embebidos comerciales, pudiendo realizar representaciones gráficas entre los pines de las mismas y otros componentes utilizados. Se incluyen diversos esquemáticos en esta memoria (Gráficos 31 y 32)

Se utiliza *Libreoffice* como herramienta de ofimática para los diseños [40] y *Ktechlab* [41] para los diagramas electrónicos cuando es requerido. Como herramienta de desarrollo, dado que se programa directamente en una de las Raspberry Pi a través de la conexión por *ssh*, sirve cualquier editor de texto que traiga la distribución.

3. Diseño de arquitectura

En este apartado se realizarán todos los diseños y configuraciones de la plataforma. Para ello se identificarán los obstáculos principales y se detallarán las soluciones planteadas ante ellos.

3.1 Aspectos del problema a resolver

Los hitos principales vienen dados por el envío y recepción de datos en un entorno rural:

- En los cultivos a gran escala, las distancias pueden ser considerables.
- Es posible que los receptores estén en lugares inaccesibles y que haya diversos obstáculos hasta los mismos influyendo en la calidad de la señal de los datos de envío y recepción.
- El uso de los sensores que se detallan, implica adaptar el diseño hacia dispositivos que puedan utilizarlos mediante protocolos eficientes energéticamente en envío y recepción de señales.
- Entendida la necesidad del uso de señales de largo alcance y también de señales con alcances inferiores pero eficientes energéticamente, es necesario realizar las adaptaciones necesarias para que los datos fluyan por todo el camino desde el emisor hacia los receptores finales y viceversa.
- Se deben obtener las métricas adecuadas a través de los sensores para garantizar el correcto funcionamiento del sistema de riego.
- El uso de diversos tipos de sensores implica adicionalmente una solución *hardware* capaz de entenderse con todos ellos.
- Dado que el entorno rural implica que los sensores se encuentren en lugares remotos, pudiendo ser de difícil acceso, es necesario que los dispositivos cercanos a ellos, sean eficientes energéticamente para mantener una autonomía energética durante el mayor tiempo posible, evitando tener que acercarse a cambiar las baterías cercanas.
- El control del riego depende completamente de la calidad de la información recogida, siendo completamente crítica la transmisión y recepción de información correcta.
- Es necesaria una redundancia adecuada, tanto en el diseño de las redes de los sensores como en la transmisión de información en la distancia a cubrir.
- El proyecto es viable sólo si es eficiente a nivel de costes, siendo importante utilizar dispositivos económicos y transmitir la información de una manera adecuada.

3.2 Solución deseada

La solución planteada trata de cubrir todos estos aspectos. Para ello se diseña un caso de uso tal que:

- Es contemplada una distancia considerable de terreno. Ello implica realizar un estudio sobre las atenuaciones debidas a la distancia
- Se suponen múltiples obstáculos que influyen en la calidad de la señal recibida
- Se utilizan protocolos eficientes en envío y recepción de señales. LTE y Zigbee (detallados los motivos anteriormente)
- Se diseña el camino de envío y recepción desde el ordenador central, hasta los sensores finales y viceversa, considerándose robusto.
- Se obtienen las métricas adecuadas a través de los diversos sensores que analizan la calidad de pH de la tierra y la temperatura ambiente y la humedad relativa.
- Todos los dispositivos utilizados son muy eficientes energéticamente. Incluso los dispositivos intermedios basados en Raspberry Pi, que detallaremos como *Gateway* posteriormente, tienen un consumo óptimo y contenido.
- Los grupos de múltiples sensores presentados, así como la cantidad de dispositivos intermedios introducidos, implica que se introduce redundancia importante, de cara a evitar en lo posible puntos de fallo.
- Todos los dispositivos, son muy económicos. Las herramientas utilizadas se basan también en *software* libre. La excepción es la antena base que implica un desembolso importante pero también permite utilizar LTE en banda libre produciendo un retorno de inversión que se espera en un tiempo relativamente corto.

Se diseña un caso de uso donde se considera una extensión ejemplo de un terreno de 1500m de longitud a cubrir con LTE. Durante la extensión de este terreno, se colocan dispositivos intermedios (*Gateway*) con grupos de sensores que se comunican por Zigbee cada 90m. Se estima poner sensores en las 4 direcciones cardinales. Ello implica que el terreno de estudio, por simplicar tiene una forma alargada (se considera adecuado ya que este diseño sirve para calcular las atenuaciones en distancias máximas). Como se verá posteriormente, el uso de Zigbee y LTE será correcto mediante el diseño realizado.

Se presenta el siguiente diagrama de bloques:

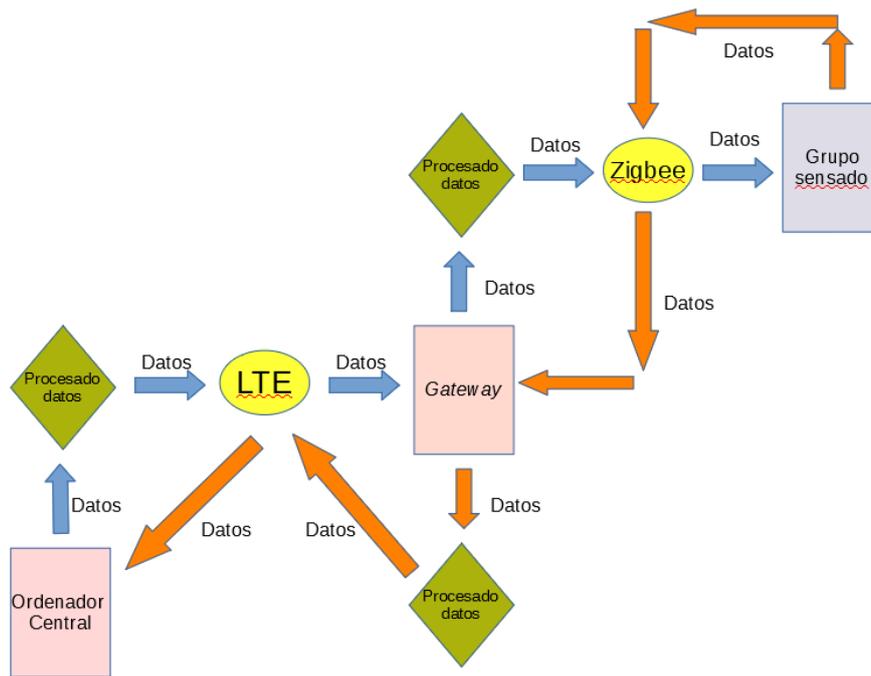


Gráfico 15 - Diagrama de bloques

Observamos dos flujos de datos.

El flujo de datos azul, indica el camino desde el ordenador central hacia los sensores. Este flujo corresponde a las actuaciones que realiza el controlador en base a la información que recibe de los flujos naranjas. Ello implica que el ordenador central, en base a la información recibida, procesa los datos y envía la información necesaria a través de la red LTE hacia los dispositivos intermedios (introducidos como *gateway*).

Estos dispositivos intermedios, basados en Raspberry Pi, procesan los datos recibidos desde la red LTE, enviándolos a través de Zigbee hacia el bloque representado como *Grupo sensado*.

El flujo de datos naranja indica el camino desde el bloque *Grupo sensado* hasta el ordenador central. Los dispositivos intermedios, que hacen de *Gateway* o puerta de enlace, recogen los datos de este *Grupo sensado* a través de la red Zigbee. Los datos recogidos por los Gateway, son procesados en los mismos dispositivos y se envían a través de LTE en banda libre hacia el ordenador central, donde se procesarán los datos recibidos para ser tratados, realimentándose así el circuito de lazo cerrado.

Por otro lado, se detalla el bloque *Grupo sensado*. Consiste en un conjunto de nodos coordinados a través de Zigbee. Como se presentaba, se mantendrán al menos cuatro dispositivos de los cuales haremos que dos de ellos sean coordinadores, para mantener la redundancia de coordinación y evitar puntos de fallo. Los otros dos dispositivos se configurarán como *endpoint*

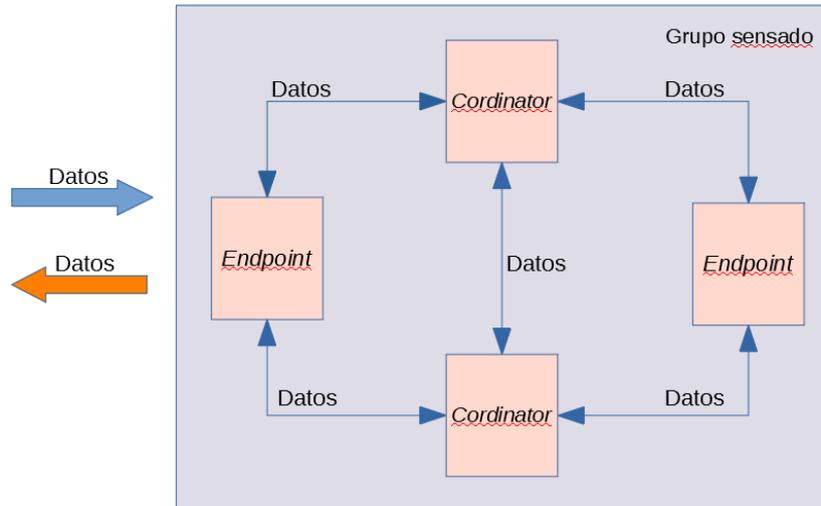


Gráfico 16 - Grupo sensado

Tanto los datos de entrada que proceden desde el *Gateway*, como los de salida que se envían hacia el *Gateway*, se envían y reciben por uno de los nodos coordinadores. Estos nodos coordinadores recogen toda la información necesaria del grupo de sensado. El segundo nodo coordinador es para mantener la redundancia, ya que en caso de fallo del coordinador principal perderíamos toda la información del grupo de sensado. Este nodo secundario actuará como un *endpoint* normal y en caso de fallo del coordinador principal, adquirirá el rol de coordinador, siendo el encargado de enviar y recibir los datos desde y hacia el *gateway* oportuno.

3.3 Componentes del sistema

Como se introducía brevemente en el apartado anterior, el sistema de lazo cerrado, tendrá los siguientes componentes:

Una central, con un ordenador que recibe y envía toda la información del sistema. Desde aquí, se podrá interpretar toda la información recibida por los sensores y enviar la información necesaria a los actuadores.

Una estación base LTE-U, con la que se crea una red LTE en espectro no licenciado.

Sensores y actuadores, formados por un conjunto de sensores próximos a las plantas, capaces de detectar la humedad relativa en la tierra, el nivel de pH de la misma y la temperatura y humedad relativa del ambiente. Los actuadores serán las válvulas que se abrirán o cerrarán en función de la necesidad de inyectar agua con fertilizante. Se sitúan además los sensores por grupos formando redes

de malla. El motivo es aprovechar este tipo de arquitectura para enviar información entre ellos y evitar fallas de sistema.

Se utilizan unos dispositivos intermedios, próximos a los sensores y actuadores, que son capaces de recoger y enviar datos a través de la red 4G (entre estos dispositivos y la central) y que son también capaces de enviar y recibir datos a través de Zigbee (hacia y desde los sensores / actuadores). Estos dispositivos se conectarán a los sensores y actuadores para recoger y enviar la información cuando se precise.

Señales. Las señales aéreas serán a través de LTE y *Zigbee*

Hay que tener en cuenta además que el sistema necesita otros componentes como bombas de agua y componentes mecánicos para poder enviar el agua y los fertilizantes hacia las plantas. Su descripción y funcionamiento es ajeno a este proyecto, ya que escapa del ámbito de la Ingeniería de Telecomunicaciones.

En los próximos apartados se presentan los diversos componentes utilizados. A la hora de escogerlos, se ha tenido en cuenta su función, la capacidad para presentar los datos necesarios, la compatibilidad con los diversos estándares utilizados, la eficiencia energética y especialmente el coste de cara a conseguir un proyecto viable económicamente.

3.3.1 Sensores y actuadores

3.3.1.1 Sensor de temperatura y humedad relativa

Se utiliza un sensor de temperatura y humedad relativa en un SoC que además tiene la ventaja de ser digital [11]. Con ello se evita introducir puntos de fallo, como la necesidad de un convertidor analógico a digital (ADC), diversas resistencias, soldaduras, ruido adicional o tener que crear un puente de Wheatstone [32]. Si bien el estudio mediante un sensor analógico es realmente interesante, hay que buscar la coherencia en utilizar un sistema preciso, simple y económico.

Para ello, se escoge el sensor DHT22 [33].



Gráfico 17 - Sensor DHT22

El fabricante proporciona las siguientes especificaciones:

| | |
|---------------------------|--|
| Model | DHT22 |
| Power supply | 3.3-6V DC |
| Output signal | digital signal via single-bus |
| Sensing element | Polymer capacitor |
| Operating range | humidity 0-100%RH; temperature -40~80Celsius |
| Accuracy | humidity +-2%RH(Max +-5%RH); temperature <+-0.5Celsius |
| Resolution or sensitivity | humidity 0.1%RH; temperature 0.1Celsius |
| Repeatability | humidity +-1%RH; temperature +-0.2Celsius |
| Humidity hysteresis | +/-0.3%RH |
| Long-term Stability | +/-0.5%RH/year |
| Sensing period | Average: 2s |
| Interchangeability | fully interchangeable |
| Dimensions | small size 14*18*5.5mm; big size 22*28*5mm |

Tabla 11 - Especificaciones DHT22

Su precio total son 3,82€ con gastos de envío incluidos.

3.3.1.2 Sensor de pH

En este caso se va a utilizar un kit de sensor de pH de Atlas Scientific [25], que proporciona:

| |
|---|
| Elementos |
| 1x Circuito de pH EZO |
| 1x Sonda de pH de plata / cloruro de plata de calidad científica. |
| 3 Soluciones de calibración de 125 ml (pH 4, pH 7 y pH 10) |

| |
|--|
| 1x 125ml de solución de almacenamiento de pH |
| 1x Conector hembra BNC premontado |
| 1x pH deca |

Tabla 12 - Elementos kit pH Atlas Scientific

La sonda es ideal para introducirla en la tierra y realizar las mediciones de pH:



Gráfico 18 - Sonda pH

Las soluciones salinas servirán para realizar las calibraciones oportunas:



Gráfico 19 - Soluciones salinas

La PCB servirá para conectar la sonda a los dispositivos Raspberry Pi, del cual se hablará posteriormente:



Gráfico 20 - PCB sonda pH

Se recogen las siguientes especificaciones aportadas por el fabricante:

| |
|---|
| Circuito de pH |
| Compatible con cualquier microprocesador que admita UART o el protocolo I2C |
| Voltaje de funcionamiento: 3,3V a 5V |
| Modo de suspensión del consumo de energía: 0,995mA a 3,3V |
| Sonda de pH |
| Rango de pH: 0-14 (error de Na + a > 12,3 pH) |
| Temperatura de funcionamiento: 1°C - 99°C |
| Presión máxima: 690 kPa |
| Profundidad máxima 60m |
| Punto isotencial: pH 7,00 |
| Longitud del cable: 1 metro |
| Velocidad de respuesta: 95% en 1 segundo |
| Peso: 49g |

Tabla 13 - Especificaciones circuito de pH

De los dos protocolos de datos, se utiliza I2C como se verá posteriormente. El precio de este kit son 210\$ con gastos de envío a España incluidos.

3.3.2 Resistencias varias

Se necesitan algunas resistencias para poder adaptar los circuitos de cara al funcionamiento en Raspberry Pi. Se hacen los cálculos oportunos de sus valores posteriormente.



Gráfico 21 - Resistencias

3.3.3 Raspberry Pi

Se utilizan dispositivos intermedios basados en Raspberry Pi [12] con una distribución GNU/Linux. El motivo es que se necesita cierta inteligencia para mantener la red en malla para hacer que los sensores se comuniquen (realmente son las Raspberry Pi las que se comunican), utilizando Zigbee, siendo Zigbee una red de corto alcance como se comentaba anteriormente.

Por otro lado, también se utiliza LTE para transmitir información desde la plataforma de control hacia los sensores / actuadores. Para ello se necesita también este dispositivo intermedio que sepa recoger las señales LTE y transmitir por Zigbee y viceversa. Cada bloque de sensores en malla que se comunican localmente por Zigbee, habrá al menos una Raspberry Pi que tenga un módulo LTE

Raspberry Pi, es un pequeño ordenador en una placa simple (SBC) de bajo coste diseñado con el objeto de estimular la enseñanza de informática en las escuelas. Aunque se trata de un *hardware* con derechos de marca manteniéndose el control de la plataforma por los fundadores, se permite su uso libre tanto a nivel educativo como particular. El *software* es de código abierto, existiendo múltiples distribuciones GNU/Linux o FreeBSD, siendo Debian la distribución oficial (en su versión adaptada denominada Raspbian). Hay también una versión de Windows 10. La placa cuenta con un procesador Broadcom basado en arquitectura ARM, memoria RAM, una GPU, puertos USB, salida HDMI, conector Ethernet, 40 pines GPIO y un conector para cámara.

Para cargar el sistema operativo y todas las utilidades y código necesario, se utiliza una tarjeta microSD como dispositivo de almacenamiento. Dado su bajo consumo, se puede alimentar con baterías sin depender de la red eléctrica (5V)

El proyecto utiliza el último modelo (Raspberry Pi 3B+) que salió al mercado en Marzo de 2018. El siguiente diagrama muestra la placa, así como sus especificaciones:

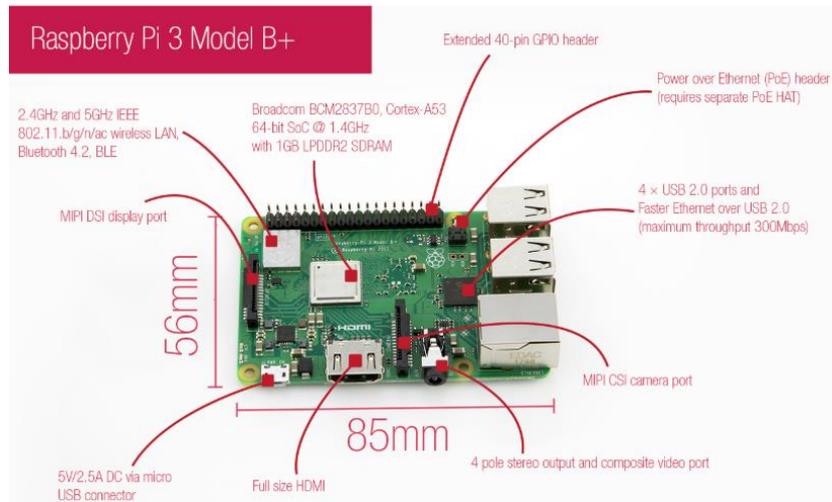


Gráfico 22 - Raspberry Pi

Es especialmente interesante detallar la entrada GPIO, donde se observa que hay múltiples pines GPIO (amarillos), donde se puede trabajar con sensores y ADC :

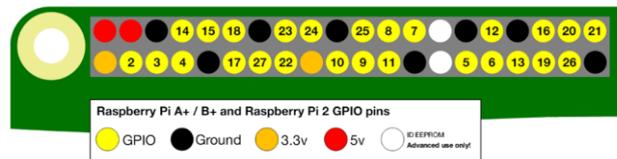


Gráfico 23 - Conector GPIO Raspberry Pi

Su precio son 37,49€

3.3.3.1 Módulo XBee Zigbee Pro

Se utiliza el módulo Digi XBee Zigbee Pro [13], que se conectará directamente a los dispositivos Raspberry Pi. Existen otras dos versiones del SoC pero dada la escasa diferencia de precio entre ellos, se elige esta versión por sus mejores características.



Gráfico 24 - XBee Zigbee Pro

Este módulo proporciona conectividad inalámbrica Zigbee de bajo consumo y es ideal para construir redes en malla. El módulo se basa en el Soc Ember EM35x de SiliconLabs e incorpora un procesador ARM Cortex™ M3 de 32 bit, eliminando la necesidad de necesitar un procesador adicional para el cómputo necesario para construir las redes en malla. El interfaz serie proporciona una conectividad ideal de alta velocidad con Raspberry Pi. Dado que el *software* inalámbrico queda aislado, se pueden desarrollar las aplicaciones sin riesgos de seguridad o de influir en el rendimiento de las señales.

De la hoja de especificaciones del fabricante, se detallan las siguientes:

| | |
|----------------------------------|---|
| Device | Digi XBee-PRO S2C Zigbee |
| Transceiver chipset | Silicon Labs EM357 SoC |
| Data rate | RF 250 Kbps, Serial up to 1 Mbps |
| Indoor / urban range | Up to 90m |
| Outdoor / RF line-of-sight range | Up to 3200m |
| Transmit power | 18dBm |
| Receiver sensitivity | -101dBm |
| Frequency band | ISM 2.4GHz |
| Operating temperatura | -40° C to +85° C |
| Encryption | 128-bit AES |
| Channels | 15 channels |
| Transmit current | Standard: 120 mA @ 3.3 VDC Programmable: 120 mA @ 3.3 VDC |
| Receive current | Standard: 31 mA @ 3.3 VDC Programmable: 45 mA @ 3.3 VDC |
| Protocol | Zigbee PRO 2007, HA-Ready with support for binding/multicasting |

| | |
|-----------------|--|
| Antenna options | Through-Hole: PCB Antenna, U.FL Connector, RPSMA Connector, or Integrated Wire SMT: RF Pad, PCB Antenna, or U.FL Connector |
|-----------------|--|

Tabla 14 - Características XBee Zigbee Pro

De estas especificaciones, se observan unas buenas sensibilidades y áreas de visión en espacio libre. El precio del módulo son 18,19\$

Se adquiere también un módulo USB [36] para poder programar los módulos XBee desde un ordenador personal, pudiendo ser el PC de la central de control.



Gráfico 25 - Adaptador USB para XBee

El precio de este dispositivo son 13,99€

3.3.3.2 Módulo LTE

Se utiliza un módulo LTE compatible con todo tipo de bandas, incluida la banda libre LTE-U (B46) . Dado que se pretende construir una red inalámbrica de largo alcance y que WiMAX es una tecnología que está desapareciendo, parece adecuado aprovechar las bandas libres para el diseño presentado. Si bien, actualmente en España, no hay mucha información sobre la posibilidad de utilizar LTE en bandas sin licencia, se entiende que es perfectamente viable y que el estudio es de interés. Son además pocos los receptores LTE que son compatibles con LTE-U y además tienen driver para GNU/Linux, siendo posible su instalación en Raspberry Pi.

El módulo AirPrime® EM7565 LTE-Advanced Pro [18], además de disponer el driver necesario, se sabe que funciona en Raspberry Pi



Gráfico 26 - AirPrime® EM7565 LTE-Advanced Pro

Se adoptan las siguientes características que proporciona el fabricante:

| | AirPrime® EM7565 LTE-Advanced Pro |
|---------------------------------------|--|
| Frequency bands | B1, B2, B3, B4, B5, B7, B8, B9, B12, B13, B18, B19, B20, B26, B28, B29, B30, B32, B41, B42, B43, B46, B48, B66 |
| Peak Download rate | 600Mbps |
| Peak Upload rate | 150Mbps |
| System drivers | Windows® 8, Windows® 8.1, Windows® 10, Linux Driver: Standard ECM, Android RIL |
| Antenna | Diversity, MIMO, GNSS Bias, Connectors: Main, Aux/GNSS, GNSS |
| Temperature range | -30°C / +70°C, -40°C / +85°C |
| Conducted TX power (worst case) | 21.5dBm |
| Conducted RX sensitivity (worst case) | -93.3dBm |

Tabla 15 - Características AirPrime® EM7565 LTE-Advanced Pro

En el caso de la sensibilidad de recepción, no hay datos en la banda 46 (en otras bandas varía entre -96,3 dBm y -93,3 dBm) . Se adoptan las peores tolerancias de transmisión y peores sensibilidades de recepción en los datos indicados, de cara a realizar un estudio sobre el dispositivo.

El precio de este dispositivo son 132,85€.

La tarjeta viene en formato M.2 mini PCI-E, siendo necesario un adaptador a USB como el siguiente [19], que además de soportar la tarjeta, proporciona en la parte trasera el *slot* necesario que la tarjeta no posee para introducir la tarjeta SIM cuando es necesaria. Este adaptador, vale 4,87€ :



Gráfico 27 - Adaptador USB 2.0 a M.2 mini PCI-E

Dada la configuración MIMO, se utilizan unas antenas que proporcionan 6dBi [20] según el fabricante y que se conectarán a las entradas Main y Aux de la tarjeta:



Gráfico 28 - Antenas dispositivo LTE

El precio de estas antenas es 5,50€ .

3.3.4 Estación base LTE-U

Como se comentaba la red WiMAX ya no es posible utilizarla. La tecnología se considera obsoleta y se ha ido sustituyendo por LTE. Dentro de las estaciones base que proporcionan LTE, existe la opción de engancharse a un eNodeB de un proveedor como Telefónica o buscar construir una red propia sobre LTE. Se opta por esta segunda opción. Aunque la inversión inicial es elevada dado que ello implica comprar una estación base compatible, se estima que la posibilidad de utilizar bandas libres, dada la cantidad de dispositivos que se pueden llegar a disponer, hará que la cantidad a desembolsar sea rentabilizada en un tiempo relativamente corto.

Se utiliza un dispositivo capaz de trabajar en bandas libres, U4G-AP1000 - Base Station de Baltic Networks [21]



Gráfico 29 - U4G-AP1000

Se recogen las siguientes características que indica el fabricante:

| | U4G-AP1000 |
|-----------------------|---|
| LTE Mode | FDD |
| Frequency bands | DL:5725MHz-5825MHz ,UL: 5150MHz-5250MHz |
| Channel Bandwidth | 5MHz/10MHz/15MHz/20 MHz |
| Max Output Power | 27 dBm /Ant(Max) |
| Receive Sensitivity | -102 dBm/Ant |
| Power Consumption | < 65W |
| Peak Rate | DL: 150Mbps, UL: 50Mbps @BW=20MHz |
| User Capacity | 32 concurrent users |
| Modulation | UL: QPSK, 16QAM, DL: QPSK, 16QAM, 64QAM |
| Operating Temperature | -40°C to 55°C |

Tabla 16 - Especificaciones U4G-AP1000

El precio de este dispositivo son 3,699.00\$

3.3.5 Ordenador central

Es necesario un dispositivo central encargado de recoger y enviar todos los datos a través de la red LTE hacia y desde los dispositivos Raspberry Pi. Será el lugar de proceso de toda la información y quien se encargará de enviar todas las señales oportunas de cara al control del sistema de riego.

3.4 Conexiones de sensores y dispositivos a Raspberry Pi

Utilizando una de las Raspberry Pi como modelo, se pueden realizar las diversas conexiones de sensores.

Raspberry Pi permite utilizar los pines del conector GPIO habilitando los módulos I2C del kernel Linux. I2C necesita dos líneas de señal (reloj, *serial clock line* – SCL) y la línea de datos (*serial data line*, SDA). Ambas líneas precisan de resistencias *pull-up* hacia VDD. Cualquier sensor que se conecte a estas líneas, es de drenador o colector abierto, lo cual en combinación con las resistencias crea un circuito *Wired-AND*. El bus I2C trabajará con lógica positiva, haciendo que un nivel alto en la línea de datos corresponda a un 1 lógico y un nivel bajo a un 0.

Con los módulos de kernel habilitados, quedan los pines GPIO 2 y 3 como SDA y SCL

3.4.1. Sensor de pH y Raspberry Pi

Dada la configuración I2C realizada, la conexión TX del circuito del sensor, se conectará a SDA (pin 2) y la conexión RX al SCL (pin 3) de la Raspberry Pi respectivamente.

El circuito proporcionado por el kit del sensor de pH tiene dos modos de funcionamiento (UART e I2C). El fabricante da las instrucciones para setearlo en el modo I2C comentado, pudiendo trabajar de esta manera con Raspberry Pi.

Recordando el conector GPIO y el apartado anterior se hacen las siguientes conexiones [26]:

- GND del SoC del sensor a GND del conector GPIO de Raspberry Pi
- Vcc del SoC del sensor a 3,3V del conector GPIO de Raspberry Pi
- TX (SDA) al pin 2 del conector GPIO de Raspberry Pi
- RX (SCL) al pin 3 del conector GPIO de Raspberry Pi
- PRB y PGND se conectan al adaptador incluido en el kit, que llevará la sonda de pH en el otro extremo

Utilizando *Fritzing*, gráficamente sería:

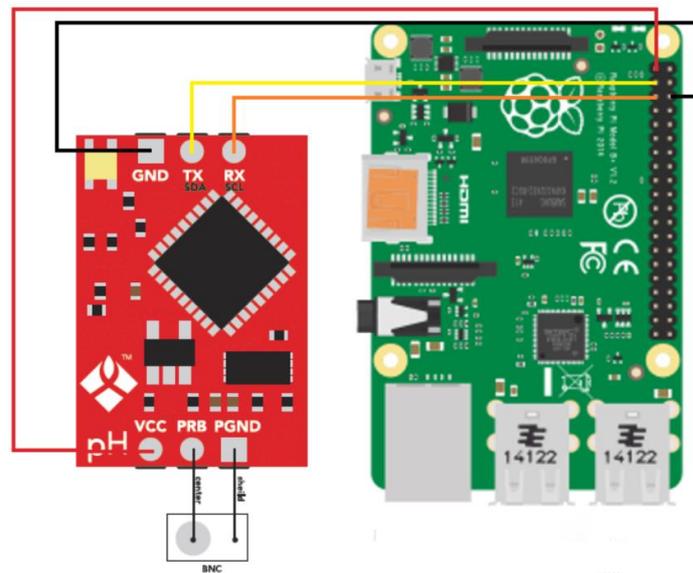


Gráfico 30 - Conexiones kit sensor pH a Raspberry Pi

Con los módulos de kernel cargados y realizadas las conexiones oportunas, el sensor será detectado mediante I2C.

Finalmente, para obtener las lecturas del sensor, se realiza la rutina necesaria, presentada en el Anexo [9.1 – Código apartado 3.3.1] .

Este script se puede lanzar en *background*, de manera que quede libre la *shell* y dejando un archivo de *log* donde se guardan los valores recogidos.

3.4.2. Sensor de temperatura y humedad Raspberry Pi

Para trabajar con el sensor DHT22, se realizan las siguientes conexiones:

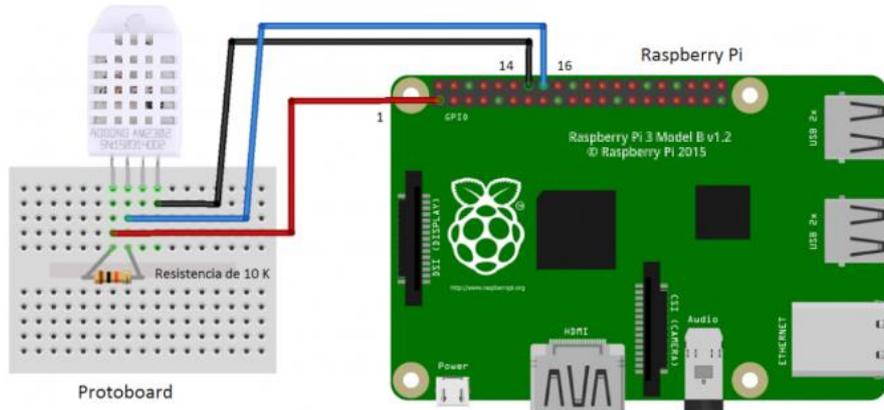


Gráfico 31 - Conexiones DHT22

Siendo la tabla de conexiones:

| Sensor DHT22 | Raspberry Pi |
|-----------------------|------------------|
| Pin 1 | Pin 1 (3,3V) |
| Pin 2 | Pin 16 (GND) |
| Pin 3 (No se utiliza) | - |
| Pin 4 | Pin 14 (GPIO 23) |

Tabla 17 - Conexiones DHT22 y Raspberry Pi

Se conecta también una resistencia de 10K conectada entre los pines 1 y 2 del sensor.

A diferencia del apartado anterior, en este caso se dispone del repositorio de Adafruit [34] que se puede clonar en cada dispositivo Raspberry Pi. Este repositorio proporciona las librerías necesarias para poder trabajar con el sensor DHT22.

Revisando la API, se observa que proporciona el método `Adafruit_DHT.read_retry()`, que al invocarlo da la temperatura y humedad relativa. Para ello hay que tener en cuenta que hay que indicar el sensor adecuado y el pin al que está conectado en el puerto GPIO de la Raspberry Pi.

Con ello, se diseña la rutina que se incluye en el Anexo [9.2 – Código apartado 3.3.2]

Una vez más, el script del diseño, se puede lanzar en *background*, de manera que quede libre la *shell* y dejando un archivo de *log* donde se guardan los valores.

3.4.3 Módulo XBee y Raspberry Pi

Se conecta el módulo XBee con el adaptador USB directamente a la Raspberry Pi. Se utiliza este módulo para evitar el conflicto que se tendría en el pin 14 con el sensor DHT22, teniendo que recurrir a un multiplexor, aunque una opción podría ser que el sensor funcione mediante el bus I2C como se hacía con el sensor de pH.

3.4.4 Módulo LTE y Raspberry Pi

El módulo LTE se conecta directamente con el adaptador USB. Hay que instalar además el driver que proporciona el fabricante, que se sabe que funciona en Raspberry Pi con el kernel proporcionado en las distribuciones actuales de Raspbian GNU/Linux

3.4.5 Resolviendo conflictos con sensores

Es interesante detallar este apartado porque es inevitable tener conflictos conforme se vayan añadiendo sensores y Raspberry Pi proporciona una manera interesante de solventarlos.

Raspberry Pi [37], permite utilizar múltiples dispositivos I2C utilizando los pines GPIO. Esta solución es muy rápida en comparación a la del multiplexor comentada anteriormente, además de ser más óptima y evitar posibles puntos de fallo. Dado que un bus I2C puede trabajar con múltiples dispositivos que tienen direcciones I2C diferentes, lo que se hará es crear buses adicionales por cada uno de los sensores que se pretenden conectar. Esto se realiza añadiendo parámetros a los módulos I2C que se habilitan en el kernel, de manera que cuando se inicia el dispositivo, se le indican qué pines GPIO se quieren habilitar para utilizar dispositivos I2C.

3.5. Construyendo una red Zigbee

La idea principal de la red Zigbee es tener un dispositivo coordinador Zigbee y varios dispositivos finales Zigbee que se comunican con este coordinador. Este dispositivo coordinador es además el que tiene también la tarjeta LTE, siendo capaz de comunicarse con el nodo central.

Si se parte de una estructura teórica de Zigbee, en la que se contemplan los 3 posibles roles [38]:

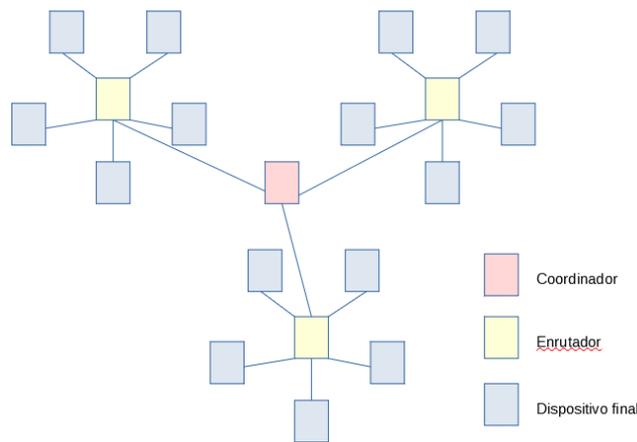


Gráfico 32 - Red Zigbee

Se eliminará el rol *coordinador* entre las diversas subredes Zigbee, quedando sólo el rol *enrutador* y el rol *dispositivo final*. Se hará también que cada rol *enrutador* sea ahora un rol *coordinador*.

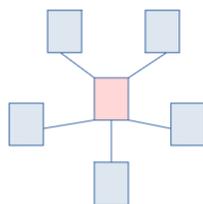


Gráfico 33 - Subred Zigbee

No obstante, al hacer este diseño, se está utilizando una topología en estrella, con un punto de fallo importante. Si se cae el nodo *coordinador*, se pierde la información de todos los nodos conectados. Por ello, se hará también que alguno de los nodos *dispositivo final*, sea también un nodo *coordinador* y que esté conectado a todos los nodos (y no sólo al otro *coordinador*).

De esta manera se construyen N redes en una malla parcial (en lugar de una red en la que los coordinadores van uniendo los diversos grupos)

Este segundo *coordinador*, sólo actuará mientras no haya otro *coordinador*. Es decir, si se le envían datos al *coordinador* y el destino no escucha, se puede hacer que se levante el servidor utilizando el script posterior (hay que hacer también que el módulo se configure en modo *coordinador*).

Además cada *coordinador* dispondrá del módulo LTE y los N *dispositivos finales* sólo dispondrán del módulo Zigbee para comunicarse con el *coordinador*. De esta manera se ahorran costes (ya que no es necesario que todos los dispositivos tengan módulo LTE). Una malla completa sería ideal con todos los nodos viéndose entre ellos entre sí, pero se busca el ahorro de costes del módulo LTE.

Se configuran los módulos XBee como *dispositivo final*, cuando se trate de un *dispositivo final*. Para ello se crea la subrutina del Anexo [9.3 – Código apartado 3.5 . Dispositivo final], haciendo que escriba el mensaje cada 5 segundos a través del puerto USB.

Cada coordinador, hay que configurarlo como *coordinador AT* . Por otro lado, es relativamente sencillo hacer que funcione este coordinador, haciendo sólo que lea el puerto USB. Para ello, se crea la subrutina del Anexo [9.4 – Código apartado 3.5 . Coordinador]

Con estas dos subrutinas, se obtiene por tanto una manera sencilla de construir la malla de nodos Zigbee.

Por otro lado, para recoger los datos de los sensores, simplemente se hará que en cada código Python detallado en los apartados de los sensores (tanto del sensor de pH como del DHT22), en lugar de escribir por pantalla, se escriba en los nodos *dispositivos finales* a través del puerto USB donde está enganchado el módulo Zigbee. Así, el servidor será capaz de recibir los mensajes con la información necesitada.

Se detallan estos cambios en el Anexo [9.5 – Código apartado 3.5 . Sensores]

3.6. Distribución de los diversos elementos

Se presenta la siguiente figura a modo de resumen del diseño detallado, que presenta la distribución de los elementos sobre el terreno de estudio:

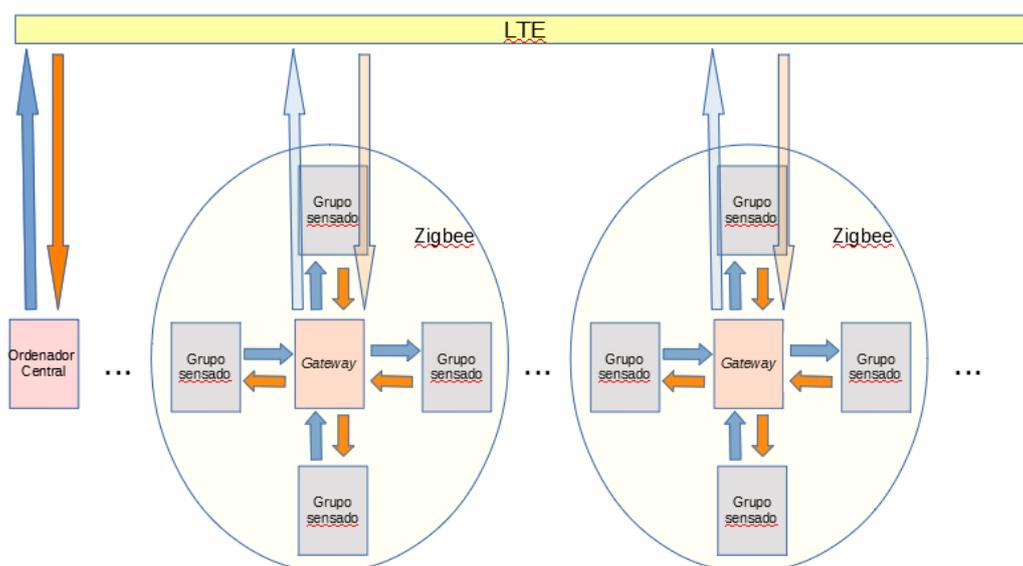


Gráfico 34 - Distribución elementos

Se observa:

- Ordenador central: Se comunica a través de la red LTE para enviar y recibir datos
- Dispositivos *Gateway*: Se comunican con cada grupo sensado (detallado anteriormente) a través de las diversas redes Zigbee. Se comunican con el ordenador central a través de la red LTE.

3.7. Resumen elementos implementados

Se presenta a modo de resumen una tabla que indica qué elementos se han codificado y qué se ha diseñado a nivel teórico en este apartado

| | |
|----------------|--|
| Sensores | Código fuente: Obtención de métricas de sensor de temperatura |
| | Código fuente: Obtención de métricas de sensor de humedad relativa |
| | Código fuente: Obtención de métricas de sensor de pH |
| <i>Gateway</i> | Conexiones con módulos Zigbee y LTE |
| | Diseño conexiones electrónicas con los diversos sensores |
| Redes en malla | Diseño de la arquitectura relacionada con los sensores |
| | Código fuente: Envío y recepción de datos nodo coordinador |
| | Código fuente: Envío y recepción de datos nodo <i>endpoint</i> |
| Red LTE | Diseño teórico sobre los dispositivos utilizados y conectados |

Tabla 18 - Resumen elementos implementados

4 Validación del sistema

4.1 Requisitos del sistema

A la hora de crear el sistema de riego, hay que valorar principalmente los siguientes puntos:

- Extensión del terreno.
- La topografía del terreno. Presencia de obstáculos y la manera de evitarlos
- Llevar el agua hasta el punto de riego
- Viabilidad económica
- Seguridad inalámbrica
- Redundancia de nodos
- Autosuficiencia energética
- Validez de la información recogida

4.2 Pruebas a realizar

El análisis que se realiza es teórico ya que no se dispone de la extensión de terreno de estudio y tampoco se han adquirido los sensores ni los módulos Zigbee ni LTE para realizar esta memoria. Si bien el coste es pequeño pensando el ahorro económico que supone automatizar un sistema de riego de una extensión elevada de terreno, la inversión que supone todo ello escapa del ámbito de esta memoria.

Entre las pruebas a realizar sobre un terreno real, cabe señalar:

- Validar experimentalmente los cálculos teóricos de las atenuaciones que se estiman posteriormente
- Validar la calidad de las métricas proporcionadas por los sensores
- Validar que el sistema es robusto ante fallos de nodos
- Validar que las redes inalámbricas son seguras
- Análisis y validación del consumo energético de los dispositivos

Para ello, se realizan las siguientes medidas:

| | |
|----------------|--|
| Calibraciones: | Sensor de temperatura |
| | Sensor de humedad relativa |
| | Sensor de pH |
| Métricas: | Medición de corriente eléctrica |
| | Monitorizaciones Nagios (cpu, memoria, carga, disco y scripts) |
| | Atenuaciones: Zigbee y LTE |
| | Estimaciones modelo Erceg y LTE |

Tabla 19 - Resumen pruebas a realizar

Se procede posteriormente a detallar cada una de ellas.

4.3 Calibraciones

4.3.1 Calibración de sensores y comparación de métricas recogidas

Para validar que la información de los sensores es correcta, hay que recurrir a las comparaciones mediante calibrado de los sensores utilizados. Se puede calibrar a pequeña escala y con ello realizar las comparaciones oportunas con las métricas recogidas por el resto de sensores distribuidos a lo largo del terreno de estudio.

Se tiene que ir realizando este procedimiento habitualmente, para mediante la comparación de los grupos de sensores calibrados con los grupos no calibrados, proceder a la calibración adecuada del resto de sensores cuando sea necesario.

4.3.1.1 Calibración del sensor de temperatura

Para el sensor de temperatura utilizaremos la caracterización como método de calibración [42]

La caracterización es el tipo de calibración en el que la resistencia de la unidad bajo testeo (UBT), está determinada por diferentes puntos de temperatura ajustando los datos a una expresión matemática (proporcionando generalmente nuevos coeficientes de calibración y una tabla de calibración provenientes del resultado de la calibración). Implica 5 pasos:

- Colocar la sonda de referencia y las UBT en la fuente de temperatura muy cerca el uno del otro

- Conectar los cables a las lecturas asegurando una conexión adecuada de 2,3 ó 4 hilos
- Medir la sonda de referencia y determinar la temperatura
- Medir y registrar la resistencia de los UBT
- Ajustar los datos

4.3.1.2 Calibración del sensor de humedad relativa

Se utilizan soluciones salinas saturadas como patrón. Se sabe que algunas sales al disolverlas en una solución de agua destilada o desionizada, producen una micro-atmósfera de humedad relativa conocida [43]

Se introduce el sensor de humedad dentro de un recipiente pequeño y cerrado herméticamente, con una micro-atmósfera determinada por la solución salina saturada (y sin que el sensor esté en contacto físico con la solución). Como soluciones se utiliza cloruro de sodio, cloruro de litio o cloruro de magnesio a una temperatura conocida y constante para proporcionar un %HR (dos soluciones salinas serían suficientes para comparar resultados y tener una calibración precisa).

Con ello, se hace la lectura del sensor cuando se llegue al equilibrio, pudiendo ser algunos minutos y comparamos los resultados obtenidos con ambas soluciones con los valores que proporcionan los sensores distribuidos a lo largo de la red.

4.3.1.3 Calibración del sensor de pH

El sensor de pH de Atlas Scientific [25], proporciona ya un conjunto de soluciones salinas para realizar las calibraciones oportunas. Se procede al uso de las mismas para realizar las calibraciones y se comparan las mediciones con las obtenidas a lo largo de la red. En caso de ser necesario se recalibrará en aquellos puntos que sea oportuno.

4.4 Métricas para medir prestaciones

4.4.1 Medición de corriente eléctrica

Se antoja interesante inventariar la corriente eléctrica necesaria, siendo el flujo de carga eléctrica que recorre un material. De esta manera se puede estimar si las baterías que alimentan los dispositivos son correctas

Dado que los dispositivos pueden ser alimentados entre 3,3V y 5V, se proporciona el siguiente inventario a 5V:

| Elemento | Corriente eléctrica |
|--|--|
| Sensor temperatura y humedad relativa: | 2.5mA |
| Sensor pH: | 1.5mA |
| Raspberry Pi 3B+ | 520mA con consumo de CPU elevado |
| XBee Zigbee Pro: | 120mA máximos |
| Módulo LTE: | No proporcionado (Indica TBD en la hoja de especificaciones) |

Tabla 20 - Relación elemento y corriente eléctrica utilizada

Se pretende alimentar todo con 5V y se ha inventariado una corriente eléctrica total de 679mA. Hay que añadir la corriente del módulo LTE (no proporcionada en las especificaciones) y multiplicar los valores de sensores de pH, temperatura y humedad relativa por el número de sensores implicado.

Una batería de 5V y 2500mAh tiene un precio aproximado de 8€. Se pueden buscar baterías de más capacidad a un precio relativamente barato o pensar en conectarlas en paralelo (siempre entendiendo que el sistema tiene que ser autónomo y que las baterías tendrán que estar a su vez alimentadas mediante los transformadores correspondientes)

Además, hay que tener en cuenta las mediciones experimentales dado que puede haber posibles pérdidas o resistencias propias debido al cableado. Aplicando la ley de Ohm ($P=V \cdot I$, $V=I \cdot R$), dado el valor de 5V y los valores de corriente estimados, se obtiene la resistencia y la potencia teórica requerida sobre los elementos inventariados. Si bien, hay que ser minuciosos realizando las medidas adecuadas con multímetro, de cara a la validación que implica escoger las baterías adecuadas.

4.4.2 Información recogida de los dispositivos intermedios

Los dispositivos intermedios juegan un papel clave para la recogida de información. Para poder validar la disponibilidad de los mismos y la calidad de las métricas recogidas es coherente, se estima necesario montar un sistema de monitorización que tenga el control de revisar constantemente los dispositivos enganchados a la red. Para ello, se puede utilizar un entorno Nagios [44], basado en *software* libre.

Nagios es un sistema de monitorización muy versátil y conocido frecuentemente en el ámbito de administración de sistemas. Para poder utilizarlo, se instalará en el ordenador central el servidor de Nagios y en cada Raspberry Pi, los *scripts* necesarios que contienen los chequeos sobre el sistema. Dado que cada Raspberry Pi tiene el puerto SSH abierto y accesible desde el ordenador central, se

pueden invocar estos scripts en cada Raspberry Pi en local desde Nagios con las credenciales adecuadas.

Se recogen los *scripts* para Nagios que monitorizan los siguientes recursos [44]:

- Ocupaciones en disco: Mediante el *script check_disk* , se pueden obtener las ocupaciones de los diferentes *filesystems* en disco y generar eventos en Nagios cuando se supera un determinado umbral. Para evitar problemas, se deben mantener rotados de los archivos de log generados.
- Consumo de CPU: Mediante el *script check_cpu* , se obtiene el consumo de CPU. Es interesante saber cuándo la CPU está trabajando demasiado para generar los eventos correspondientes.
- Carga (*load average*): Este *script* es especialmente importante de cara a validar la capacidad de proceso. Mediante el *script check_load* se conoce la carga que está soportando el sistema. Esta métrica indica cuándo la CPU no está trabajando de manera holgada y se traduce en que se observan *threads* en espera. En caso de contemplarse un valor de carga superior al número de CPUs del dispositivo dispositivo Raspberry Pi, se está indicando que el procesador no está soportando toda la capacidad de proceso que se le está pidiendo y por lo tanto hay un problema en el diseño.
- Memoria RAM: El *script check_ram* , indica el consumo de RAM del dispositivo.
- Acceso a Raspberry Pi por SSH. El *script check_ssh*, indica si se está llegando al dispositivo Raspberry Pi. También se prestará especial atención al mismo. Un evento en cualquiera de los dispositivos, indica que no se llega al mismo desde el ordenador central y por lo tanto habría un problema de red llegando o una caída del dispositivo.
- Ejecución de *scripts* de recogida de métricas: Se validan los *scripts* que recogen las métricas de los sensores invocándolos a través del servidor Nagios. De esta manera, cuando el *script* de medida de pH o humedad relativa y temperatura ambiente no funcionan o proporcionan valores incorrectos, se sabrá que falla y en qué nodo falla. Se desarrolla un caso de uso el Anexo [9.6 – Código apartado 4.3.4]

El funcionamiento a través de estos *scripts* es el habitual en entornos Nagios:

- En cada Raspberry Pi se dejan los *scripts* recogidos y los *scripts* propios en un *path* con los permisos correctos para poder ser invocados en remoto.
- El ordenador central tiene el servidor Nagios. El servidor Nagios, invoca los *scripts* remotos por *ssh* en cada Raspberry Pi. Nagios provee esta funcionalidad y se pueden lanzar los *scripts* en remoto de manera nativa y recoger los resultados procesados en las máquinas remotas y que Nagios necesita.

- Los *scripts* se lanzan por tanto en cada dispositivo. El consumo de CPU derivado de la obtención de métricas, se realiza únicamente en cada dispositivo. No obstante, en el caso de los *scripts* recogidos, hay que tener en cuenta que las métricas obtenidas son muy simples y los valores se devuelven de comandos directos de sistema de GNU/Linux (*sar*, *top*, *du* y *df*). Este consumo de lanzar los comandos una vez cada varios minutos, es completamente despreciable. En el caso de los *scripts* propios, ocurre un poco algo similar ya que las métricas se obtienen directamente de la información proporcionada por los sensores.
- Nagios provee la funcionalidad de configurar umbrales de alertas. Los *scripts*, están ya preparados para devolver los resultados en valores de porcentajes al servidor central de Nagios. En caso de que una ocupación en disco, memoria, cpu o carga de un dispositivo, supere los valores de 95%, se procede al escalado desde el ordenador central, al teléfono correspondiente. Para ello, dado el plugin [46] de Telegram e instalado en el servidor de Nagios, se procederá a las notificaciones oportunas cuando corresponda a los dispositivos móviles indicados. En el caso de los *scripts* propios, se ha diseñado el *script* de *shell* indicado en el Anexo [9.6 – Código apartado 4.3.4], para replicar este comportamiento devolviendo un 0 ó 1 conforme se estime un valor correcto o incorrecto (valor obtenido a su vez del código programado en Python para obtener la información de las sondas).

También se pretende cuidar el ancho de banda dados los dispositivos enganchados a la red. Para ello, aunque el uso de datos es bajo, es importante tener en cuenta este apartado de cara a pensar en posibles escalados o diseños con cientos de nodos. En el proyecto propuesto, el tráfico se reduce a texto puntual, no obstante, se decide utilizar ventanas diferentes de tiempo en las invocaciones de los *scripts* de todos los dispositivos, de cara a reducir el tráfico de red.

Dado que en el presupuesto posterior, se estima un caso de uso con 50 dispositivos Raspberry Pi y teniendo en cuenta que no son críticos unos minutos en un sistema de riego como el propuesto, se puede extraer la información de 5 dispositivos cada minuto. Ello implica que se tardan 10 minutos en tener la información de todos los dispositivos, de donde se deriva que la tasa de refresco de nueva información en Nagios de cada dispositivo es cada 10 minutos. Es decir, cada 10 minutos se refrescará la información mostrada en Nagios con los valores obtenidos de cada dispositivo.

4.4.3. Atenuaciones y Zigbee

Los próximos estudios, tanto en Zigbee como posteriormente en LTE, tendrían que aplicarse con los valores de un caso de uso real. Los cálculos teóricos se realizan utilizando las especificaciones de los fabricantes de los dispositivos adquiridos y a modo de ejemplo considerando múltiples obstáculos en una distancia considerable.

Los valores obtenidos sirven para realizar una estimación que a priori indique si el diseño es viable para los supuestos realizados pero tienen que validarse experimentalmente midiendo la calidad de recepción de las señales.

En Zigbee se utiliza la banda libre de 2,4GHz. Las especificaciones anteriores también indicaban un alcance de 90m en edificios en zonas urbanas y 3200m en línea recta en zonas rurales sin obstáculos.

Se sitúa por ejemplo cada nodo coordinador al que se le colocan 4 dispositivos Zigbee a 90m cada uno (se estima la distancia del peor caso posible). Uno de los motivos de utilizar la medida de 90m, es que se pretende tener una cierta precisión en las medidas de pH del suelo. Dada la posible variedad de los cultivos, se estima necesario tener dispositivos Zigbee cada cierta distancia.

Cada nodo coordinador, tendrá 4 dispositivos Zigbee, uno en cada una de de las 4 direcciones cardinales. Dada la pequeña distancia, se conoce que las atenuaciones en espacio libre entre el nodo coordinador y los nodos finales, vienen dadas por:

$$L_{bf} = 20\log\left(\frac{4\pi d}{\lambda}\right) \text{dB} = 20\log\left(\frac{4\pi 90m}{\frac{3 \cdot 10^8 \frac{m}{s}}{2,4 \cdot 10^9 s^{-1}}}\right) = 79,1308 \text{dB}$$

Dado que la potencia de transmisión es 18dBm, la potencia recibida es la potencia de transmisión menos la atenuación más la ganancia de la antena del receptor (el fabricante proporciona 7dBi de ganancia respecto a una antena isotrópica ideal).

La potencia recibida es entonces:

$$18\text{dBm} - 79,1308\text{dB} + 7\text{dBi} = 5,869\text{dBm}$$

Como se intuía, es muy superior a la sensibilidad del receptor (-101dBm). Es deducible ya que el fabricante indica que en espacio libre hay un alcance de 3200m frente a los 90m utilizados.

Continuando con el cálculo de las atenuaciones y dado que los dispositivos Raspberry Pi están a nivel del suelo, se supone un modelo donde se restan las atenuaciones de los objetos intermedios (COST-231). Si en estos 90m, se estima que hay 30 plantas o árboles (una planta o un árbol cada 3m) y cada árbol se cree que puede atenuar 3dBm, se obtiene una atenuación total de 90dBm.

$$5,869\text{dBm} - 90 \text{dBm} = -84,131\text{dBm}$$

El terreno tampoco es llano y además de las atenuaciones producidas por los propios cultivos, habrá otros obstáculos. No obstante, dada la sensibilidad del receptor de -101dBm y realizando el ejercicio donde se han considerado unas atenuaciones muy importantes de 90dBm, no parece posible que la recepción se vea menguada en espacio libre como si se estudiara un medio urbano. De hecho los cultivos habitualmente se plantan en hileras para favorecer la recogida de los frutos y es sencillo introducir las Raspberry Pi entre las hileras para evitar en lo posible estas atenuaciones máximas.

Por lo tanto, se puede deducir a priori que dadas las especificaciones del fabricante en cuanto a los módulos Zigbee y la ganancia de las antenas, es posible utilizar bloques de grupos de Zigbee donde cada nodo coordinador podría tener 4 nodos finales a 90m cada uno. Además, cabe recordar que uno de los nodos finales, tendrá visibilidad con todos los nodos (y un módulo LTE instalado). En caso de detectar que no se puede comunicar con el coordinador, se configuraría en modo coordinador.

4.4.4 LTE y atenuaciones en espacio libre

Por cada grupo de redes en malla en Zigbee, se utilizan dos dispositivos que tienen un módulo LTE para poder enviar y recibir datos desde la central.

Como se detallaba anteriormente, se usará la banda libre de 5,4GHz.

Recurriendo a las especificaciones, en el caso de la estación base, se obtienen 27 dBm de potencia de emisión y -102dBm como sensibilidad de recepción.

Para el módulo LTE instalado en los dispositivos Raspberry Pi, hay además 23dBm de potencia de emisión y 6dBi por antena. En las especificaciones del fabricante, no se especifica la sensibilidad de recepción de la banda 46, así que se hace un supuesto recurriendo a la peor sensibilidad encontrada en la tabla entre todas las bandas, siendo esta -93,3dBm.

Se parte de un caso de estudio en el que en la distancia más larga, hay un terreno de 1500m a cubrir con LTE, calculándose la atenuación en espacio libre en este peor caso.

Si la antena LTE está a 40m de altura y hay 1500m hasta el dispositivo Raspberry Pi más lejano, la distancia es la hipotenusa del triángulo formado por ambas distancias, es decir:

$$d = \sqrt{40^2 + 1500^2} = 1500,533 \text{ m}$$

Siendo la atenuación en espacio libre:

$$L_{bf} = 20 \log \left(\frac{4\pi d}{\lambda} \right) \text{dB} = 20 \log \left(\frac{4\pi 1500,533 \text{m}}{\frac{3 \cdot 10^8 \frac{\text{m}}{\text{s}}}{5,4 \cdot 10^9 \text{s}^{-1}}} \right) = 110,614 \text{dB}$$

Cada dispositivo Raspberry Pi tiene 2 antenas y tenemos 23dBm de potencia de emisión. La potencia recibida considerando sólo las atenuaciones en espacio libre, es:

$$23 \text{dBm} + 2 \cdot 6 \text{dBi} - 110,614 \text{dB} = -75,614 \text{dBm}$$

Dada la mínima sensibilidad de recepción de -93,3dBm, en espacio libre se pueden cubrir teóricamente distancias superiores a los 1500m de terreno utilizados como ejemplo.

4.4.5 LTE y Erceg

Se supone ahora una estimación más realista mediante el modelo Erceg:

$$L[\text{dB}] = 20 \log \left(\frac{4 \cdot \pi \cdot 100}{\lambda} \right) + 10 \cdot \left(a - b \cdot h_b + \frac{c}{h_b} \right) \cdot \log \left(\frac{d}{100} \right) + 6 \cdot \log \left(\frac{f}{2.000} \right) + X_h + s$$

Se considera que se está estudiando un medio rural con terrenos planos, donde además la antena de la estación base se va a poner a 40m del suelo y las Raspberry Pi a 4m de altura. Esto se hace porque realizando diversas pruebas con varios tipos de distancias y alturas, se comprueba que el método implica elevar tanto el transmisor como el receptor para hacer que las atenuaciones sean adecuadas.

Por un lado, esto implica reducir las atenuaciones producidas por los obstáculos intermedios y por otro lado, si no se hace, se resuelve que el método proporciona escenarios donde se necesitan decenas de estaciones base, siendo el proyecto inviable.

Hay que tener en cuenta que también hay un sensor de pH que hay que introducir en el suelo. Si en algún momento al alargar la distancia del cable que conecta el sensor con la PCB hace que las lecturas sean imprecisas, hay que jugar con las alturas o buscar la manera de introducir nuevos dispositivos intermedios que se comuniquen específicamente con estos sensores.

Se utilizan por tanto los siguientes valores:

| | | |
|-------|---------------------|--------------------|
| a=3,6 | b=0,005 | c=20 |
| s=8,2 | h _b =40m | h _m =4m |

Tabla 21 - Valores método Erceg

También:

$$X_h = -10,8 \log \frac{h_m}{2} = -10,8 \log \frac{4m}{2} = -3,251 \text{ dB}$$

Además:

$$20 \log \frac{4 \cdot \pi \cdot 100}{3 \cdot 10^8 \frac{m}{s}} + 10 \cdot (3,6 - 0,005 \cdot 20m + \frac{20}{40m}) \cdot \log \frac{1500,533m}{100} + 6 \cdot \log \frac{5,4 \cdot 10^9 s^{-1}}{2000} + 8,2 = 135,148 \text{ dB}$$

Dadas las pérdidas calculadas, la potencia que llega al receptor lejano será por tanto:

$$23\text{dBm} + 2 \cdot 6\text{dBi} - 135,148\text{dB} + 3,251\text{dB} = -96,897\text{dBm}$$

Como la sensibilidad de recepción es de -93,3dBm, en este peor caso posible considerado, se resuelve que la estación base LTE es insuficiente para cubrir los 1500m de extensión.

Jugando un poco con la altura a la que situamos la Raspberry Pi más lejana, por ejemplo 8m:

$$X_h = -10,8 \log \frac{h_m}{2} = -10,8 \log \frac{8m}{2} = -6,502 \text{ dB}$$

Siendo la potencia recibida:

$$23\text{dBm} + 2 \cdot 6\text{dBi} - 135,148\text{dB} + 6,502\text{dB} = -96,646\text{dBm}$$

Dada la sensibilidad de recepción de -93,3dBm estimada, se obtendrían ya valores adecuados para esta última Raspberry Pi en un terreno de 1500m de longitud, utilizando una única estación base LTE. Se puede elevar un poco más la Raspberry Pi a los 10m máximos que permite el método, siendo:

$$X_h = -10,8 \log \frac{h_m}{2} = -10,8 \log \frac{10m}{2} = -7,54 \text{ dB}$$

El margen mejora ligeramente, aunque habría 10m desde el dispositivo Raspberry Pi hasta el suelo donde está la sonda de pH. Lo que se puede hacer es introducir una Raspberry Pi adicional en este punto, más cercana al suelo y que los sensores se comuniquen por Zigbee con los anteriores. Es decir, dada la poca distancia entre ambas Raspberry Pi, es posible montar una red de sensores utilizando Zigbee y que esta última red, cubra los sensores conectados a ambas Raspberry Pi.

Recordamos para ello que el alcance en espacio libre de las redes Zigbee era de 3200m, siendo los 90m el peor caso estimado con múltiples obstáculos. Es por tanto viable hacer que los últimos nodos Zigbee coordinador, formen una única malla con todos los nodos finales indicados.

De esta manera se pueden cubrir los 1500m de distancia sin introducir estaciones base adicionales costosas.

El ejemplo utilizado indica que la distancia y las atenuaciones utilizadas, hacen que el diseño esté cerca del límite posible de viabilidad. Para cubrir distancias superiores, habría que pensar en introducir una estación base adicional o buscar algún tipo de repetidor que ayude sin tener que llegar a desembolsar el precio de una estación base nueva.

4.5 Elementos de riego

Si bien, el funcionamiento de los dispositivos mecánicos o el uso de los mismos mediante señales eléctricas queda fuera del alcance del proyecto, de cara a validar el sistema, hay que tener en cuenta el mantenimiento de los mismos.

Las válvulas, actuadores o los cabezales de riego pueden mantenerse con ácido nítrico añadido al agua cada cierto tiempo para evitar la obstrucción de los mismos. Este compuesto además de evitar la obturación de los goteros, permitirá también mejorar la calidad del agua, ya que descompone los bicarbonatos que contiene de manera natural. Añadir estos compuestos, afectará al pH del subsuelo y habrá que monitorizar los niveles del mismo después de proceder a ello. En el caso de las bombas de riego, también son un elemento crítico sobre los cuales habrá que realizar los mantenimientos adecuados y por lo tanto, cabe realizar validaciones sobre el funcionamiento de las mismas.

Dado que se disponen sondas de pH repartidas a lo largo de todo el terreno a monitorizar, una manera de poder saber si todos estos elementos están funcionando correctamente, es mediante el control de las métricas recogidas por las sondas Nagios. Cuando regamos, los valores de pH recogidos por las sondas tienen que variar respecto a los valores recogidos inmediatamente antes del riego. En caso de detectarse que no varían, se sabe que hay un problema en la zona de riego monitorizada.

Se puede jugar con todos los valores recogidos en todas las zonas de riego para saber dónde está este problema. Otra opción interesante podría ser introducir caudalímetros por el circuito de riego o cualquier elemento que ayude a detectar posibles fugas o problemas adicionales.

4.6 Seguridad de los sistemas

Para evitar vulnerabilidades que comprometan el sistema, se mantienen actualizados los equipos con las últimas versiones disponibles de *kernel* y sistema operativo tanto en las Raspberry Pi como el ordenador central.

4.7 Resumen de elementos utilizados en la verificación de requisitos

Con las pruebas realizadas, se introduce la siguiente tabla que indica un resumen de la verificación de los requisitos introducidos:

| Elemento | Verificación requisitos |
|---|---|
| Calibraciones sensores | Se valida que la calidad de las mediciones que van a realizar los sensores, va a ser correcta. |
| Monitorizaciones Nagios (valores de sensores) | Validan los datos obtenidos de los sensores, de cara a estimar si hay que regar la parte de terreno medida por cada elemento en cuestión. |
| Medición corriente eléctrica | Permite validar los consumos eléctricos de los elementos de estudio utilizados. |
| Cálculo atenuaciones Zigbee | Para los elementos que se comunican a través de Zigbee, se valida que la transmisión de datos es válida en la extensión de estudio. |
| Cálculo atenuaciones LTE | Para los elementos que se comunican a través de LTE, se valida que la transmisión de datos es válida en la extensión de estudio. |
| Modelo Erceg - LTE | Se realiza una validación teórica de la calidad del diseño inalámbrico propuesto. |
| Monitorizaciones nagios (<i>gateway</i>) | Permiten validar la disponibilidad de los elementos <i>gateway</i> y el estado de funcionamiento de cada uno de ellos. |
| Notificaciones nagios | Aseguran que las personas que tienen que conocer los eventos que se producen sobre el sistema, los reciben |
| Actualizaciones seguridad sistemas | Permiten validar que los sistemas son seguros |

Tabla 22 - Resumen verificación requisitos

5. Valoración económica y viabilidad del proyecto

A continuación se indica el coste económico de los elementos utilizados. Hay que tener en cuenta que para todo el sistema de riego hay muchos costes adicionales propios de las bombas de riego, la canalización de todas las tuberías, cableado eléctrico y baterías o elementos como las válvulas o actuadores. Adicionalmente habrá costes derivados del propio mantenimiento de cada elemento.

No obstante se pueden inventariar los costes del sistema de automatización propuesto, detallados a continuación:

| Elemento | Coste de un elemento | Coste de varios elementos |
|---|----------------------|---------------------------|
| Sensor DHT22: | 3,82€ | 191€ (50 unidades) |
| Sensor y PCB pH Atlas Scientific: | 188,20€ | 9410€ (50 unidades) |
| Resistencias: | 10 cent /€ | 5€ (50 unidades) |
| Raspberry Pi 3B+: | 37,49€ | 1874,5€ (50 unidades) |
| Módulo XBee Zigbee Pro: | 16,30€ | 815€ (50 unidades) |
| Adaptador USB para XBee: | 13,99€ | 699,5€ (50 unidades) |
| Módulo AirPrime® EM7565 LTE-Advanced Pro: | 132,85€ | 6642,5€ (50 unidades) |
| Adaptador USB 2.0 a M.2 mini PCI-E: | 4,87€ | 243,5€ (50 unidades) |
| Antenas Dispositivo LTE: | 5,50€ | 275€ (50 unidades) |
| Estación Base U4G-AP1000: | 3314,44€ | 3314,44€ (1 unidad) |
| Ordenador central: | 500€ | 500€ (1 unidad) |
| Horas ingeniería e instalación: 200 horas | 10€ | 2000€ |
| TOTAL | 4227,56€ | 25970,44€ |

Tabla 23 - Valoración económica

El cálculo de unidades es orientativo para entender cómo pueden crecer los costes según sea relativamente extensa la cantidad de tierras a regar. Concretamente, en el apartado de los módulos Zigbee, se estimaba un caso de uso de una distancia de 1500m de longitud con nodos Zigbee cada 90m.

Las 50 unidades estimadas, implican también que en cada nodo, además de los dispositivos en línea, se utilizan dos dispositivos en dirección perpendicular adicionales: $(1500 / 90) \times 3$). Es también es un valor que da un múltiplo sencillo de cara a tener una estimación genérica en el presupuesto.

Se estiman también 200 horas de ingeniería e instalación del sistema con un precio cerrado de 2000€ por la implantación del proyecto.

El terreno de estudio tiene: 1500m x 180m = 270000m²

De la referencia [47], se estiman 0,08€ por m² de terreno de riego (incluyendo costes de mano de obra) por cada 1 m³ de agua utilizado. Se disponen 21.600€ por cada m³ de agua utilizado, englobando los costes de mano de obra. Dado que la medición de los m³ de riego necesarios son fuertemente dependientes del cultivo, se utiliza este valor de 1m³ de agua teórico.

Se estima que al menos se ahorra la mitad de agua cada año por cada m³ de riego y considera el retorno de la inversión (ROI), como:

$$ROI = \frac{\text{Ingresos} - \text{Inversión}}{\text{Inversión}}$$

Si se quiere desear un ROI de un 133%, se obtiene la cantidad de ingreso necesaria:

$$1,33 = \frac{x - 25970,44 \text{ €}}{25,970,44 \text{ €}} \rightarrow x = 60.511,1252 \text{ €}$$

Se preveen cada año unos costes de mantenimiento del sistema de 2600€ (una décima parte del coste inicial). Se pretende tener en cuenta sólo el ahorro de mano de obra relacionada con el riego y el agua regada (sin tener en cuenta los beneficios derivados de la venta), para comprobar que el retorno de la inversión sin los beneficios, podría obtenerse en un tiempo relativamente corto de tiempo:

| Año | Coste inicial | Coste de agua con el proyecto implantado | Coste de mantenimiento | Coste total acumulado | Coste de agua sin el proyecto implantado |
|--------------|---------------|--|------------------------|-----------------------|--|
| 1 | 25970,44€ | 10800€ | 2600€ | 39370,44€ | 21600€ |
| 2 | - | 10800€ | 2600€ | 52,770,44€ | 21600€ |
| 3 | - | 10800€ | 2600€ | 66170,44€ | 21600€ |
| 4 | - | 10800€ | 2600€ | 79570,44€ | 21600€ |
| TOTAL | | | | 79570,44€ | 86400€ |

Tabla 24 - Retorno de inversión

Por otro lado, es muy posible tener mayor eficiencia en los frutos obtenidos dadas la eficiencia de riego gracias a las mediciones de los sensores, lo cual contribuiría a mejorar el beneficio y por tanto el ROI en menos tiempo.

6. Conclusiones

Se implementa un diseño teórico de un sistema de riego sobre una extensión elevada de terreno a partir de los requisitos implicados en el diseño. Para ello se han detallado los desarrollos teóricos oportunos, se han presentado los dispositivos y elementos *hardware* utilizados, se detallan las conexiones pertinentes de todos los dispositivos y se desarrollan diversas rutinas necesarias implicadas en el funcionamiento del sistema.

Se introduce un componente de validación del sistema apoyándose especialmente en la calibración de los sensores y en la monitorización de los elementos del sistema, de cara a obtener métricas adecuadas y evitar puntos de fallo sobre el mismo. También se realizan los cálculos oportunos sobre la viabilidad del diseño en las redes inalámbricas de estudio y el caso de uso presentado.

Finalmente la valoración económica presentada pretende concluir la viabilidad del proyecto y el tiempo estimado para tener el retorno de la inversión.

Todo ello muestra que se cumplen los hitos principales de la propuesta inicial y que el estudio es además viable.

Hay que realizar algunas observaciones de interés. Por un lado, el desarrollo ha sido teórico. El coste del conjunto, incluyendo la estación base LTE-U, es elevado para adquirirse para realizar el trabajo académico.

Además, el desarrollo de un sistema completo, implica que hay que introducir componentes mecánicos, motores para las bombas de agua y las tuberías con las válvulas correspondientes. Todo ello implicaría alargar demasiado el proyecto con detalles sin duda interesantes, pero ajenos a la Ingeniería de Telecomunicación y por tanto, no se contemplan en esta memoria.

7. Glosario

OSI: Open Systems Interconnection

Router: Enrutador

DHCP: Dynamic Host Configuration Protocol

DNS: Domain Name Server

SPOF: Single point of failure

SoC: System on chip

FDD: Frequency Division Duplexing

TDD: Time Division Duplexing

UART: Universal Asynchronous Receiver-Transmitter

GPIO: General Purpose Input/Output

SSH: Secure Shell

Shell: Terminal o consola

Script: Código, programa

I2C: Inter-integrated circuit

SDA: Serial data line

SCL: Serial clock line

String: Cadena de caracteres

Timestamp: Marca temporal

ADC: Convertidor analógico - digital

Root: Usuario administrador en sistemas Unix

Python: Lenguaje de programación

Hostname: Nombre de la máquina

ROI: Retorno de inversión

8. Bibliografía

- [1] AGROMATICA. Guía para realizar una fertirrigación adecuada. Publicado en Marzo de 2014. Disponible en: <https://www.agromatica.es/guia-de-fertilizacion/>
- [2] DOMODESK. A fondo: Zigbee. Disponible en: <https://www.domodesk.com/216-a-fondo-zigbee.html>
- [3] ALMAGRO Martos, Leonardo. 4G: CUARTA GENERACIÓN MÓVIL. Disponible en: <http://ceres.ugr.es/~alumnos/tutorialcfc/cuatro.html>
- [4] ALMANSA López, Leticia. Arquitectura de red de LTE. Publicado en Julio de 2013. Disponible en: <http://intotally.com/tot4blog/2013/07/11/lte-for-beginners-day-2-brief-description-about-lte-network-architecture/?lang=es>
- [5] ITU. Cálculo de la atenuación en espacio libre. Publicado en 1994. Disponible en: https://www.itu.int/dms_pubrec/itu-r/rec/p/R-REC-P.525-2-199408-!!!PDF-S.pdf
- [6] SATUÉ Villar, Antonio. Comunicaciones sin hilos. UOC PID_00158041. Propagación sobre espacio libre, páginas 27 a 38. Revisado en 2017.
- [7] ROUSE, Margaret. Topología de red de malla. Disponible en: <https://searchdatacenter.techtarget.com/es/definicion/Topologia-de-red-de-malla-red-de-malla>
- [8] DOMÍNGUEZ Gros, Ferran. SOLÉ Casals, Jordi. Descripción de un sistema de instrumentación UOC- PID_00214784. Páginas 8 -10. Revisado en 2017.
- [9] DIGIKEY. Grove Moisture Sensor. Actualizado en 2019. Disponible en: https://www.digikey.es/product-detail/es/seeed-technology-co.,-ltd/101020008/1597-1141-ND/5482619?utm_adgroup=Evaluation+Boards+-+Expansion+Boards%2C+Daughter+Cards&mkwid=sU2ZTLO4G&pcrid=278710408307&pkw=&pmt=&pdv=c&productid=5482619&slid=&gclid=CjwKCAjwqLbIBRBYEiwAV3pCJfuu1I8GBSrXnJcwQbayd4KIsX6y3L7-RxtOK6hVqLdF766swi5JxoCEpsQAvD_BwE
- [10] SEED STUDIO. Grove – Moisture Sensor. Actualizado en 2019. Disponible en: https://seeedoc.github.io/Grove-Moisture_Sensor/
- [11] FARNELL. HIH-5030/5031 Series. Low Voltage Humidity Sensors. Publicado en Marzo de 2010. Disponible en: http://www.farnell.com/datasheets/1676925.pdf?_ga=2.141332978.1516939014.1526490154-1350854175.1526490154

[12] RASPBERRY PI FOUNDATION. Raspberry Pi. Actualizado en 2019. Disponible en: <https://www.raspberrypi.org/>

[13] DIGI. Digi Xbee Zigbee. Actualizado en 2019. Disponible en: <https://www.digi.com/products/embedded-systems/rf-modules/2-4-ghz-modules/xbee-zigbee>

[14] WAVESHARE. 4G/3G/2G/GSM/GPRS/GNSS HAT for Raspberry Pi, LTE CAT4, for Southeast Asia, West Asia, Europe, Africa. Actualizado en 2019. Disponible en: <https://www.waveshare.com/SIM7600E-H-4G-HAT.htm>

[15] DIGIKEY. Heraeus Sensor Technology USA 32208550. Actualizado en 2019. Disponible en: <https://www.digikey.es/product-detail/es/heraeus-sensor-technology-usa/32208550/1759-1021-ND/6677771>

[16] HERAEUS. Pt RTD thin film element type M222. Publicado en Junio de 2016. Disponible en: https://www.heraeus.com/media/media/group/doc_group/products_1/hst/m_sensors/us_3/M222_HST-USA.pdf

[17] XGPFORUM. Frequency Band of 3GPP LTE TDD mode . Actualizado en 2019. Disponible en: https://www.xgpforum.com/new_XGP/en/001/TDD_band.html

[18] SIERRA WIRELESS. AirPrime® EM7565 LTE-Advanced Pro Module. Actualizado en 2019. Disponible en: <https://www.sierrawireless.com/products-and-solutions/embedded-solutions/products/em7565/>

[19] ALIEXPRESS. Mini PCI-E al adaptador USB con SIM 8 Pin ranura para tarjeta WWAN/LTE módulo para soporte de Escritorio PC SIM 6pin/8pin conecte la tarjeta. Actualizado en Abril de 2019. Disponible en: https://es.aliexpress.com/store/product/Mini-PCI-E-to-USB-Adapter-with-SIM-8-Pin-Card-Slot-for-WWAN-LTE-Module/627075_32869302601.html?spm=a219c.search0104.3.209.6c746e23Zv4KYj&ws_ab_test=searchweb0_0,searchweb201602_1_10065_10068_10547_319_10891_317_10548_10696_10084_453_454_10083_10618_10307_10820_10821_10301_10303_537_536_10059_10884_10887_321_322_10103,searchweb201603_56,ppcSwitch_0&algo_expid=1943027c-f0b6-46d2-b5fd-42216d81b223-29&algo_pvid=1943027c-f0b6-46d2-b5fd-42216d81b223&transAbTest=ae803_5

[20] ALIEXPRESS. 2 x 6dBi de banda Dual M.2 IPEX MHF4 U fl Cable a RP-SMA antena Wifi DE Intel AC 9260, 9560, 8265, 8260, 7265, 7260 NGFF M.2 tarjeta. Actualizado en Abril de 2019. Disponible en: https://es.aliexpress.com/store/product/2-x-6dBi-Dual-Band-M-2-IPEX-MHF4-U-fl-Cable-to-RP-SMA-Wifi/3353004_32891736728.html?spm=a219c.search0104.3.34.58076fc75BDB6c&ws_ab_test=searchweb0_0,searchweb201602_1_10065_10068_10547_319_10891_317_10548_10696_10084_453_454_10083_10618_10307_10820_10821_10301_10303_537_536_10059_10884_10887_321_322_10103,searchw

eb201603_56,ppcSwitch_0&algo_expId=f8544ca4-36e9-4de2-b190-8e7129625d08-5&algo_pvid=f8544ca4-36e9-4de2-b190-8e7129625d08&transAbTest=ae803_5

[21] BALTIC NETWORKS. U4G-AP1000 - Baicells LTE-U Spectra FDD/TDD Unlicensed 5.8GHz 15dBi Base Station. Actualizado en 2019. Disponible en: <https://www.balticnetworks.com/baicells-lte-u-spectra-fdd-tdd-unlicensed-5-8ghz-15dbi-base-station.html>

[22] INFOAGRO. Concepto de pH e importancia en fertirrigación. Disponible en: http://www.infoagro.com/abonos/pH_suelo.htm

[23] EUMEDIA. Valores de pH para los diferentes fertilizantes utilizados en fertirrigación <http://www.eumedia.es/user/articulo.php?id=1072>

[24] ADAFRUIT. ADS1115 16-Bit ADC - 4 Channel with Programmable Gain Amplifier. Publicado en 2019. Disponible en: <https://www.adafruit.com/product/1085>

[25] ATLAS SCIENTIFIC. pH Kit. Actualizado en 2019. Disponible en: https://www.atlas-scientific.com/product_pages/kits/ph-kit.html

[26] MIHIDROPI. Raspberry Pi Ph Sensor. Publicado en Mayo de 2016. Disponible en: <https://myhydropi.com/connecting-a-ph-sensor-to-a-raspberry-pi>

[27] TEXAS INSTRUMENTS. ADS1115. Actualizado en 2019. Disponible en: <http://www.ti.com/product/ADS1115>

[28] TEXAS INSTRUMENTS. TLC2262C. Actualizado en 2019. Disponible en: <http://www.ti.com/product/TLC2262>

[29] MOUSER. Analog Devices Inc. AD620 Instrumentation Amplifiers. Revisado en 2019. Disponible en: https://www.mouser.es/new/Analog-Devices/adi-ad620-amplifiers/?gclid=Cj0KCQjw19DIBRCSARIsAOnfRej6xXGbBBN3O_xK2pgeVYQZHTfrPSBcLSZ8zwQ6LQg0wmSJrj3wggMaAInfEALw_wcB

[30] DEBIAN. Debian GNU/Linux. Actualizado en Abril de 2019. Disponible en: <https://www.debian.org/>

[31] RASPBIAN. Raspbian. Actualizado en Abril de 2019. Disponible en: <http://raspbian.org>

[32] DOMÍNGUEZ Gros, Ferran. SOLÉ Casals, Jordi. Introducción a los sensores. UOC- PID_00214781. Página 17. Revisado en 2017.

[33] SPARKFUN. DHT22. Disponible en: <https://www.sparkfun.com/datasheets/Sensors/Temperature/DHT22.pdf>

- [34] ADAFRUIT. Adafruit Python DHT Sensor Library. Actualizado en Noviembre de 2018. Disponible en: https://github.com/adafruit/Adafruit_Python_DHT
- [35] FRITZING. Fritzing. Actualizado en 2019. Disponible en: <http://fritzing.org>
- [36] WaveShare. XBee USB Adapter. Actualizado en 2019. Disponible en: https://www.waveshare.com/wiki/XBee_USB_Adapter
- [37] INSTRUCTABLES. Raspberry PI Multiple I2C Devices. Publicado en Enero de 2018. Disponible en: <https://www.instructables.com/id/Raspberry-PI-Multiple-I2c-Devices/>
- [38] LIBELIUM COMUNICACIONES DISTRIBUIDAS. XBee for Arduino and Raspberry Pi. Disponible en: <https://www.cooking-hacks.com/documentation/tutorials/xbee-arduino-raspberry-pi-tutorial>
- [39] PROJECTLIBRE. ProjectLibre. Actualizado en 2019. Disponible en: <http://www.projectlibre.com>
- [40] LIBREOFFICE. Libreoffice. Actualizado en 2019. Disponible en: <https://es.libreoffice.org/>
- [41] KTECHLAB. Ktechlab. Actualizado en 2019. Disponible en: <https://github.com/ktechlab/ktechlab>
- [42] DAVIS INSTRUMENTS. How to Calibrate an RTD or Platinum Resistance Thermometer (PRT). Publicado en Septiembre de 2010. Disponible en: <http://www.davis.com/TechLibraryArticle/1225>
- [43] DÁVILA Pacheco, Jesús A. y MARTÍNES López, Enrique. Validación del método de calibración de higrómetros capacitivos con soluciones sal-agua como generadoras de humedad. Publicado en Julio de 2007. Disponible en: <https://www.cenam.mx/dme/pdf/TM06.pdf>
- [44] NAGIOS. Nagios Open Source. Actualizado en 2019. Disponible en: <https://www.nagios.org/>
- [45] MONITORING PLUGINS. Monitoring plugins. Actualizado en 2019. Disponible en: <https://www.monitoring-plugins.org/>
- [46] NAGIOS EXCHANGE. Telegram Notifications. Actualizado en 2019. Disponible en: <https://exchange.nagios.org/directory/Addons/Notifications/Telegram-Notifications/details>
- [47] CYPE INGENIEROS. Generador de precios – Riego. Disponible en: http://www.generadordeprecios.info/obra_nueva/Urbanizacion_interior_de_la_parcela/Jardineria/Acondicionamiento_del_terreno/Riego.html

9. Anexos

9.1 Código apartado 3.3.1.

Módulos a importar

```
import io # lecturas y escrituras
import fcntl # acceso a parámetros I2C como direcciones
import time # esperas por sleep y timestamps
import string # parseo de strings
```

Funciones con la lógica necesaria para el control del sensor de pH:

```
class atlas_i2c:
    long_timeout = 5 # timeout necesario para consultar lecturas y
    calibraciones
    short_timeout = 5 # timeout para comandos regulares
    default_bus = 1 # bus por defecto para I2C en Raspberry Pi 3B+
    default_address = 99 # dirección por defecto para el sensor pH

    def __init__(self, address=default_address, bus=default_bus):
        # Abrimos dos flujos de archivo, uno para leer y otro
        # para escribir
        # El canal específico I2C se selecciona con el bus
        # Es usualmente 1, except para versiones antiguas siendo 0
        # wb and rb indican lectura binaria (rb)
        # y escritura binaria (wb)
        self.file_read = io.open("/dev/i2c-" + str(bus), "rb", buffering=0)
        self.file_write = io.open("/dev/i2c-" + str(bus), "wb", buffering=0)

        # Inicializa I2C a una dirección especificada por el
        # usuario o por defecto
        self.set_i2c_address(address)
        def set_i2c_address(self, addr):
            # Configuramos las comunicaciones I2C al esclavo
            # especificado por la dirección
```

```

# Los comandos para desarrollar sobre I2C que usan las
# funciones ioctl se especifican en el archivo
# i2c-dev.h file del paquete i2c-tools instalado
I2C_SLAVE = 0x703
fcntl.ioctl(self.file_read, I2C_SLAVE, addr)
fcntl.ioctl(self.file_write, I2C_SLAVE, addr)

def write(self, string):
    # añade un carácter vacío al final
    # y envía el string por I2C
    string += "\00"
    self.file_write.write(string)

def read(self, num_of_bytes=31):
    # Leemos un número determinado de bytes desde I2C,
    # lo parseamos y mostramos el resultado
    res = self.file_read.read(num_of_bytes)
    # leemos desde la placa
    response = filter(lambda x: x != '\x00', res)
    # eliminamos el carácter vacío para conseguir la respuesta
    if(ord(response[0]) == 1):
        # si la respuesta no es un error
        char_list = map(lambda x: chr(ord(x) & ~0x80),
list(response[1:]))
        # cambiamos MSB a 0 para todos los caracteres
        #recibidos menos el primero
        #y sacamos una lista de los caracteres
        # NOTA: cambiar el MSB a 0 es un bug
        # en la Raspberry Pi y no debería tener que hacerse
        return "Command succeeded " + ''.join(char_list)
        # convertir la lista de caracteres a string
        # y devolverla
    else:
        return "Error " + str(ord(response[0]))

def query(self, string):

```

```

    # escribir un comando a la placa,
    #esperar el timeout correcto y leer la respuesta
    self.write(string)

    # los comandos de lectura y calibración
    # necesitan más tiempo
    if((string.upper().startswith("R")) or
        (string.upper().startswith("CAL"))):
        time.sleep(self.long_timeout)
    elif((string.upper().startswith("SLEEP"))):
        return "sleep mode"
    else:
        time.sleep(self.short_timeout)

    return self.read()

def close(self):
    self.file_read.close()
    self.file_write.close()

```

Rutina principal:

```

def main():
    device = atlas_i2c() # crea el objeto de puerto I2C,
                        # especifica la dirección o bus
                        # si es necesario

    print ">> Atlas Scientific sample code"
    print ">> Any commands entered are passed to the board via I2C
except:"
    print (">> Address,xx changes the I2C address the Raspberry Pi
"
"communicates with.")
    print (">> Poll,xx.x command continuously polls the board every
"
"xx.x seconds")
    print (" where xx.x is longer than the %0.2f second "
"timeout." % atlas_i2c.long_timeout)

```

```

print " Pressing ctrl-c will stop the polling"

# bucle principal
while True:
    myinput = raw_input("Enter command: ")

    # El comando de dirección permite cambiar la
    # dirección que la Raspberry Pi encuestará
    if(myinput.upper().startswith("ADDRESS")):
        addr = int(string.split(myinput, ',')[1])
        device.set_i2c_address(addr)
        print ("I2C address set to " + str(addr))

    # el comando de sondeo continuo, sondea
    # automáticamente la placa
    elif(myinput.upper().startswith("POLL")):
        delaytime = float(string.split(myinput, ',')[1])

        # comprobamos si el tiempo de sondeo es muy pequeño,
        # lo cambiamos al timeout mínimo si es muy pequeño
        if(delaytime < atlas_i2c.long_timeout):
            print ("Polling time is shorter than timeout, set-
ting "
                "polling time to %0.2f" % atlas_i2c.long_timeout)
            delaytime = atlas_i2c.long_timeout

        # conseguir la información de la placa que sondeamos
        info = string.split(device.query("I"), ",")[1]
        print ("Polling %s sensor every %0.2f seconds, press
ctrl-c "
            "to stop polling" % (info, delaytime))

    try:
        while True:
            print device.query("R")
            time.sleep(delaytime - atlas_i2c.long_timeout)
    except KeyboardInterrupt:

```

```

        # recoge el comando ctrl-c,
        # que rompe el bucle
        print "Continuous polling stopped"

# si no es una palabra clave especial, pasar el
# comando directamente a la placa
else:
    try:
        print device.query(myinput)
    except IOError:
        print "Query failed"

if __name__ == '__main__':
    main()

```

9.2 Código apartado 3.3.2

```

#!/usr/bin/python
# Importamos las librerías necesarias, incluida la que
# acabamos de instalar
import sys
import time
import Adafruit_DHT
# Configuración del tipo de sensor DHT
sensor = Adafruit_DHT.DHT22
# Configuración del puerto GPIO al cual esta conectado (GPIO 23)
pin = 23
# Ejecutamos
try:
# Bucle principal infinito hasta que se captura excepción
while True:
# Obtiene la humedad y la temperatura desde el sensor
humedad, temperatura = Adafruit_DHT.read_retry(sensor, pin)
# Imprime en la consola las variables temperatura y humedad con un
decimal
print('Temperatura={0:0.1f}* Humedad={1:0.1f}%'.format(temperatura,
humedad))

```

```
# Duerme 10 segundos
time.sleep(10)
# Se ejecuta en caso de que falle alguna instrucción dentro del try
except Exception,e:
# Imprime en pantalla el error e
print str(e)
```

9.3 Código apartado 3.5. Dispositivo final

```
#!/usr/bin/python
import time
import serial

serie = serial.Serial('/dev/ttyUSB0', 9600)
print 'Conectando puerto serial'

while True:
    salida = serie.write('Hola Raspberry Pi!\n')
    time.sleep (5)
```

9.4 Código apartado 3.5. Coordinador

```
#!/usr/bin/python
import serial

serie = serial.Serial('/dev/ttyUSB0', 9600)
print 'Conectando puerto serial'

while True:
    entrada = serie.readline().strip()
    print 'Recibo: %s' % entrada
```

9.5. Código apartado 3.5. Sensores

Teniendo definido el puerto USB como:

```
serie = serial.Serial('/dev/ttyUSB0', 9600)
```

Para el sensor de pH, en el código del sensor de pH, se hará la siguiente modificación para que el *endpoint* escriba los datos del sensor por el puerto USB en lugar de enviarlos por pantalla :

```
import socket
...
salida = socket.gethostname()+ '    '+serie.write(device.query("R"))
```

En el caso del dispositivo DHT22, se hace:

```
import socket
...
salida = socket.gethostname()+ '    '+serie.write('Tempera-
tura={0:0.1f}* Humedad={1:0.1f}%'.format(temperatura, humedad))
```

Teniendo en el nodo coordinador, los datos de los sensores. También se dará un identificador a cada nodo.

Se ha optado por introducir el *hostname* de cada máquina. Dado que el sistema de riego comprende de múltiples sensores y dispositivos intermedios, de esta manera se puede identificar en la red de dónde procede cada línea que entra al nodo coordinador.

9.6. Código apartado 4.3.4

```
#!/bin/bash
fecha=`date +%d%m%Y`
umbral= "valor"
# Recoger el último valor del archivo de log que deja el script
# del sensor correspondiente.
# archivo_log es el archivo de log
# $col es el número de la columna que tiene el valor del sensor
valor = `cat "archivo_log" | tail -1 | awk `{print $col}`

if [ $? ] && [ "$valor" -gt "$umbral" ]
then
    # Si el script se lanza correctamente y si el valor
    # que proporciona el sensor es mayor que el umbral que
    # buscamos, se sale con flag correcto para Nagios
    exit 0
else
    # Si falla o tenemos una métrica incorrecta,
    # salimos devolviendo error
    exit 1
fi
```