



UNIVERSIDADE DA CORUÑA



Escola Politécnica Superior

Trabajo Fin de Grado
CURSO 2021/2022

***SISTEMA IOT PARA LA MONITORIZACIÓN DE
TERRENOS DE CULTIVO USANDO UNA RED DE
ÁREA AMPLIA Y BAJA POTENCIA***

Grado en Ingeniería en Tecnologías Industriales

ALUMNA/O

Aldara Díaz García

TUTORAS/ES

José Antonio Becerra Permuy
Alejandro Paz López

FECHA

SEPTIEMBRE 2022

Título y Resumen

Sistema IoT para la monitorización de terrenos de cultivo usando una red de área amplia y baja potencia.

El objetivo del presente proyecto consiste en diseñar e implementar un sistema prototípico de monitorización de terrenos de cultivo utilizando tecnologías IoT, aportando al usuario información clave para tomar decisiones de mejora. Para ello se han definido unos parámetros medibles relacionados con el control de la calidad y productividad de los terrenos.

Una de las principales dificultades es el despliegue de sensores en áreas de cultivo amplias y alejadas. Para abordar este problema se seleccionan sensores con soporte para la tecnología de comunicación LoRa. Por otro lado, se utilizan dispositivos basados en Raspberry Pi con conectividad WiFi para la implementación de los nodos ubicados en instalaciones con cobertura mediante redes convencionales. Esto último sería la implementación de un dispositivo para el control del nivel almacenado de fertilizante.

El sistema se completa con una plataforma de computación en la nube, Amazon Web Services (AWS). En dicha plataforma se resuelve la integración de datos provenientes de los sensores, su almacenamiento persistente y el análisis de los mismos para envío de alertas vía correo electrónico en caso de detectarse valores fuera de los rangos configurados.

Finalmente, se integran y prueban todos los componentes sobre distintos escenarios para validar el funcionamiento del sistema.

Título e Resumo

Sistema IoT para a monitorización de terreos de cultivo usando unha rede de área ampla e baixa potencia.

O obxectivo do presente proxecto consiste en deseñar e implementar un sistema prototípico de monitorización de terreos de cultivo utilizando tecnoloxías IoT, achegando ao usuario información clave para tomar decisións de mellora. Para iso definíronse uns parámetros medibles relacionados co control da calidade e produtividade dos terreos.

Unha das principais dificultades é o despregamento de sensores en áreas de cultivo amplas e afastadas. Para abordar este problema selecciónanse sensores con soporte para a tecnoloxía de comunicación LoRa. Doutra banda, utilízanse dispositivos baseados en Raspberry Pi con conectividade WiFi para a implementación dos nodos situados en instalacións con cobertura mediante redes convencionais. Isto último sería a implementación dun dispositivo para o control do nivel almacenado de fertilizante.

O sistema complétase cunha plataforma de computación na nube, Amazon Web Services (AWS). Na devandita plataforma resólvese a integración de datos provenientes dos sensores, o seu almacenamiento persistente e a análise dos mesmos para envío de alertas vía correo electrónico en caso de detectarse valores fóra dos rangos configurados.

Finalmente, intégranse e proban todos os compoñentes sobre distintos escenarios para validar o funcionamento do sistema.

Title and Summary

IoT soil monitoring system using a low power and wide area network.

The objective of this project is the design and implementation of a monitoring soil prototype using IoT technology, providing the user important information for making improvement decisions. For this, measurable parameters have been defined associated to quality and productivity soil control.

One of the main difficulties is the deployment of sensors in wide and remote soil areas. To address this problem, a set of sensors with support for LoRa technology is chosen. Furthermore, we use devices based on Raspberry Pi with WiFi connectivity for the implementation of sensor nodes located in places with access to a more common communication network. The last is the case of the device that monitors the fill level of the fertilizer tanks.

The system is also supported by a cloud computing and IoT platform, specifically Amazon Web Services (AWS). This platform is used for the integration of data from the sensors, for the persistent storage and analysis of sensor data and for the notification of events and alerts via email in case of detecting values outside the configured ranges.

Finally, all the components are integrated and tested across different scenarios to validate the whole system.



UNIVERSIDADE DA CORUÑA



Escola Politécnica Superior

**TRABAJO FIN DE GRADO
CURSO 2021/2022**

*SISTEMA IOT PARA LA MONITORIZACIÓN DE
TERRENOS DE CULTIVO USANDO UNA RED DE
ÁREA AMPLIA Y BAJA POTENCIA*

Grado en Ingeniería en Tecnologías Industriales

Documento

MEMORIA

Índice de contenido

1 Introducción.....	11
1.1 Motivación	11
1.2 Objetivos	12
2 Metodología.....	13
3 Planificación	14
4 Especificación	16
4.1 Estudios y análisis previos	16
4.1.1 Internet de las Cosas (IoT)	16
4.1.2 Internet Industrial de las Cosas (IIoT).....	17
4.1.3 Parámetros relevantes	18
4.1.4 Monitorización inteligente de suelos.....	19
4.1.5 Tecnologías de comunicación relevantes.....	20
4.1.6 Plataformas en la nube	23
4.1.7 Lenguajes de programación	26
4.1.8 Protocolos de comunicación.....	27
4.1.9 Sensorización distribuida	28
4.2 Requisitos de usuario.....	29
5 Diseño e implementación	30
5.1 Vista global del sistema.....	30
5.2 Sensorización del entorno	31
5.2.1 Hardware	31
5.2.2 Software.....	36
5.2.3 Montaje	37
5.3 Plataforma IoT global	45
5.3.1 Envío y recepción de mensajes de sensorización con MQTT.....	46
5.4 Registro de datos en la nube.....	49
5.4.1 Implementación y pruebas	50
5.5 Notificación de alertas	52
5.5.1 Implementación y pruebas	53
5.6 Pruebas del sistema global	56
5.7 Código fuente	59
6 Referencias	60
Anexos	66
1. Presupuestos	66
2. Manuales	67

Índice de figuras

Figura 1. Crecimiento de la población mundial. Fuente: [3].	11
Figura 2. Diagrama de Gantt. Fuente: Elaboración propia.....	14
Figura 3. Gráfico comparativo de las tecnologías de comunicación inalámbricas más relevantes. Fuente: [17].....	21
Figura 4. Arquitectura LoRaWAN. Fuente: [20].....	22
Figura 5. Arquitectura Amazon Web Services. Fuente: [22].....	23
Figura 6. Arquitectura Microsoft Azure. Fuente: [24].....	24
Figura 7. Arquitectura Google Cloud Platform. Fuente: [26].....	25
Figura 8. Diferentes modos del gateway en una red LoRa. Fuente: [28].....	26
Figura 9. Estadística sobre la utilización de plataformas Cloud. Fuente: [31].....	27
Figura 10. Comunicación publicación/suscripción. Fuente: Elaboración propia.	28
Figura 11. Vista global del sistema. Fuente: Elaboración propia.....	31
Figura 12. Raspberry Pi 4 Model B. Fuente: [41].	32
Figura 13. Esquema sobre el funcionamiento de un Gateway. Fuente: Elaboración propia.	33
Figura 14. WisGate Edge Lite 2. Fuente: [42].	34
Figura 15. Dragino LSPH01. Fuente: [43].....	34
Figura 16. Milesight EM-500 CO2. Fuente: [45].....	35
Figura 17. Kit WisBlock GPS. Fuente: [46].	35
Figura 18. RAK1903. Fuente: [47].	35
Figura 19. Software Milesight ToolBox. Fuente: Elaboración propia.	37
Figura 20. Configuración de la banda de frecuencia del Gateway. Fuente: Elaboración propia.....	37
Figura 21. Configuración LoRa Basic Station del Gateway. Fuente: Elaboración propia.....	38
Figura 22. Botón de encendido de Milesight EM500-CO2. Fuente: Elaboración propia.	38
Figura 23. Interfaz gráfica de Milesight EM500-CO2. Fuente: Elaboración propia.39	39
Figura 24. Configuración de la banda de frecuencia de Milesight EM500-CO2. Fuente: Elaboración propia.....	39
Figura 25. Conexión del conversor USB-TTL al dispositivo LSPH01. Fuente: Elaboración propia.....	40
Figura 26. Comando AT para cambiar el intervalo de transmisión en LSPH01. Fuente: Elaboración propia.....	40
Figura 27. Encendido de Dragino LSPH01. Fuente: Elaboración propia.....	41
Figura 28. Colocación de la sonda de Dragino LSPH01. Fuente: Elaboración propia.	41
Figura 29. Montaje de la placa en Box-B2. Fuente: Elaboración propia.....	42

Figura 30. Conectores A de la placa base RAK5005-O. Fuente: Elaboración propia.	42
Figura 31. Conexión de las antenas LoRa y BLE al módulo RAK4631. Fuente: Elaboración propia.....	43
Figura 32. Conexión de la batería y placa solar a RAK5005-O. Fuente: Elaboración propia.....	43
Figura 33. Colocación final del sistema RAK1903. Fuente: Elaboración propia....	44
Figura 34. Gestor de librerías en Arduino IDE. Fuente: Elaboración propia.	44
Figura 35. Comprobación del código de arranque de RAK4631. Fuente: Elaboración propia.	45
Figura 36. Fragmento de código conexión broker MQTT. Fuente: Elaboración propia.....	46
Figura 37. Fragmento de código envío de mensajes. Fuente: Elaboración propia.	47
Figura 38. Visualización de mensajes en tema "Datos_Raspberry". Fuente: Elaboración propia.....	47
Figura 39. Arquitectura IoT Core for LoRaWAN. Fuente: Elaboración propia.	48
Figura 40. Acciones de la regla DecoderEM500. Fuente: Elaboración propia.....	48
Figura 41. Visualización de mensajes en tema "Parámetros_Ambientales_Decoder". Fuente: Elaboración propia.....	49
Figura 42. Visualización de mensajes en tema "Parámetros_Terreno_Decoder". Fuente: Elaboración propia.....	49
Figura 43. Visualización de mensajes en tema "Parámetros_Ambientales_Decoder". Fuente: Elaboración propia.....	49
Figura 44. Creación de una tabla DynamoDB. Fuente: Elaboración propia.	50
Figura 45. Ejemplo de instrucción SQL para la regla Dynamo_Rule_Ambiente. Fuente: Elaboración propia.....	51
Figura 46. Acciones de la regla Dynamo_Rule_Ambiente. Fuente: Elaboración propia.	51
Figura 47. Tabla de datos DynamoDB. Fuente: Elaboración propia.	52
Figura 48. Arquitectura Amazon SES. Fuente [70].	52
Figura 49. Creación de un usuario IAM para Amazon SES. Fuente: Elaboración propia.	53
Figura 50. Política asociada al IAM Rol SESLambdaRole. Fuente: Elaboración propia.	54
Figura 51. Fragmento de código empleado para la notificación de alertas. Fuente: Elaboración propia.....	55
Figura 52. Ejemplo de instrucción SQL para Notification_Terreno_Rule. Fuente: Elaboración propia.....	56
Figura 53. Notificación de alerta vía correo electrónico. Fuente: Elaboración propia.	56
Figura 54. Implementación real de los dispositivos EM500-CO2 y LSPH01. Fuente: Elaboración propia.....	57

Figura 55. Implementación real del dispositivo RAK1903. Fuente: Elaboración propia.....	58
Figura 56. Notificación de alerta en los parámetros de temperatura y humedad. Fuente: Elaboración propia.....	58
Figura 57. Notificación de alerta en los parámetros de pH y temperatura. Fuente: Elaboración propia.....	58
Figura 58. Ejemplo de mensajes almacenados en la tabla DynamoDB. Fuente: Elaboración propia.....	59
Figura 59. Notificación de alerta en el nivel de fertilizante disponible. Fuente: Elaboración propia.....	59
Figura 60. Notificación de alerta en el parámetro de temperatura. Fuente: Elaboración propia.....	59
Figura 61. Notificación de alerta en el parámetro de iluminancia. Fuente: Elaboración propia.....	59
Figura 62. Interfaz de configuración y supervisión del Gateway. Fuente: Elaboración propia.....	67
Figura 63. Configuración del modo AP del Gateway. Fuente: Elaboración propia.	68
Figura 64. Configuración del modo cliente del Gateway. Fuente: Elaboración propia.....	68
Figura 65. Actualización del firmware del Gateway. Fuente: Elaboración propia. .	69
Figura 66. Arduino IDE 1.8.19. Fuente: Elaboración propia.....	69
Figura 67. Comando para la actualización del gestor de arranque de RAK4631. Fuente: Elaboración propia.....	69
Figura 68. Comando para la actualización del firmware de RAK4631. Fuente: Elaboración propia.....	70
Figura 69. Actualización de firmware RAK4631-R realizada con éxito. Fuente: Elaboración propia.....	70
Figura 70. Gestor de tarjetas en Arduino IDE. Fuente: Elaboración propia.....	71
Figura 71. Instalación manual de la biblioteca de RAK1903. Fuente: Elaboración propia.....	71
Figura 72. Registro en AWS. Fuente: Elaboración propia.....	72
Figura 73. Definir la región de AWS. Fuente: Elaboración propia.....	72
Figura 74. Creación de un usuario IAM. Fuente: Elaboración propia.	73
Figura 75. Credenciales de seguridad del usuario. Fuente: Elaboración propia....	73
Figura 76. Política asociada a Raspberry Pi. Fuente: Elaboración propia.....	74
Figura 77. Propiedades del objeto Raspberry. Fuente: Elaboración propia.	75
Figura 78. Certificados y claves del objeto Raspberry. Fuente: Elaboración propia.	76
Figura 79. Definición del Gateway en AWS. Fuente: Elaboración propia.....	77
Figura 80. Certificados del Gateway. Fuente: Elaboración propia.....	77
Figura 81. Configuración LoRa Basic Station del Gateway. Fuente: Elaboración propia.	78

Figura 82. Política <i>IoTWirelessDestinationPolicy</i> . Fuente: Elaboración propia.....	79
Figura 83. Relaciones de confianza de <i>IoTWirelessDestinationRole</i> . Fuente: Elaboración propia.....	79
Figura 84. Rol de ejecución de la función Lambda. Fuente: Elaboración propia ..	80
Figura 85. Instrucción SQL para la regla. Fuente: Elaboración propia.	80
Figura 86. Creación de un destino. Fuente: Elaboración propia.....	81
Figura 87. Creación de un perfil de dispositivo. Fuente: Elaboración propia.	82
Figura 88. Creación de un perfil de servicio. Fuente: Elaboración propia.	83
Figura 89. Agregar el dispositivo EM500-CO2 a AWS. Fuente: Elaboración propia.	
	83

1 INTRODUCCIÓN

1.1 Motivación

La población mundial se encuentra en desarrollo constante, con la necesidad de afrontar todos los cambios y desafíos que surgen. De todas las actividades económicas existentes unas de las más afectadas son las pertenecientes al sector primario, en especial la agrícola, en la que se centrará el siguiente trabajo. En este primer capítulo se expondrán los retos a los que tiene que hacer frente y se valorará la importancia de las nuevas tecnologías para poder solucionarlos.

El primer gran reto al que se enfrenta es el firme crecimiento de la población mundial. Según un estudio de la ONU se prevé que para 2050 se llegue a los 9.700 millones de habitantes, y en 2100 se alcance un pico máximo de 11.000 millones, afectando en gran medida a los países más desarrollados [1]. Esto se debe principalmente al aumento de la longevidad, llegando de manera global a alcanzar los 77,1 años, provocando que las nuevas generaciones convivan mayor tiempo con las pasadas. Esta tendencia significará que la demanda de alimentos, tanto para el consumo humano como animal, crecerá de manera paralela [2].

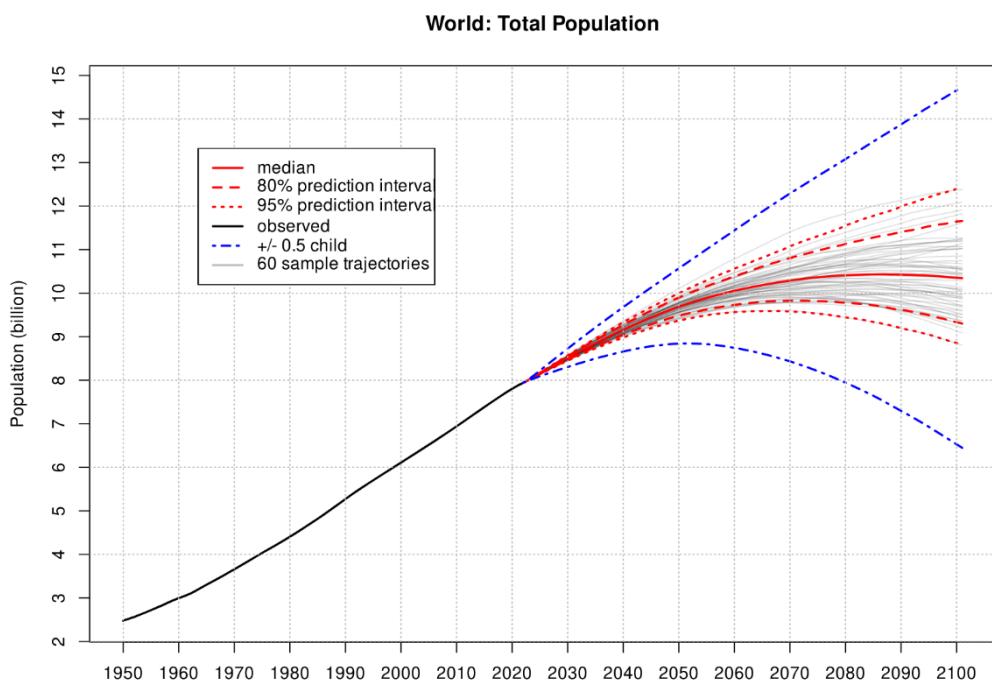


Figura 1. Crecimiento de la población mundial. Fuente: [3].

Sumado a esto aparece el éxodo rural [4]. Se estima que la población urbana alcance un 70% del total mundial, reduciéndose así los espacios destinados a los cultivos para utilizarlos con otros fines. Esto supone un problema de base, ya que si se reducen los espacios dedicados al sector primario, pero el reclamo de estos productos aumenta, habrá que buscar maneras de conseguir una mayor producción y rendimiento para abastecer la demanda.

El cambio climático también tiene una gran influencia en el sector primario [5] [6]. Los efectos que influyen directamente en la agricultura incluyen el aumento de las temperaturas, las prolongadas sequías, aumento del nivel del mar e inundaciones y tormentas. Todo esto hace que las variables ambientales cambien de manera aleatoria e incontrolada, no dándose las condiciones óptimas para el cultivo deseado. Derivado de esto surge la obligatoriedad de utilizar los recursos naturales de una manera mucho más controlada y eficiente. Así, el sector agrícola se ve obligado a adaptarse y

fusionarse con las nuevas tecnologías, ya que los métodos tradicionales de cultivo no son suficientes para hacer frente a todos los impedimentos derivados de estas alteraciones fortuitas en el clima. De esta unión surge la llamada “Smart Agriculture”, consistente en el empleo de tecnologías como IoT, sensores, actuadores o inteligencia artificial con el fin de maximizar tanto la rentabilidad como la producción de los terrenos, de una manera sostenible y eficiente a pesar de los obstáculos mencionados.

La aplicación de métodos y tecnologías de IoT en este sector ha sufrido una importante expansión en los últimos años [7] [8]. Desde conseguir una agricultura de precisión hasta el propio monitoreo del ganado, aportando un conjunto de ventajas que permiten dar solución a problemas a los que no es capaz de hacer frente la metodología tradicional utilizada hasta el momento [9]:

- Se generan modelos de predicción para que el usuario esté informado sobre las variables ambientales, plagas y enfermedades. Esto permitirá hacer frente y prevenir de forma anticipada los posibles riesgos para la plantación.
- Los agricultores podrán dedicar más tiempo al estudio y mejora de los cultivos, gracias a la automatización del trabajo.
- Monitorizar el estado de los cultivos, pudiendo variar ciertos valores gracias a actuadores.
- Recogida y análisis de datos, mejorando la eficiencia a la hora de tomar decisiones.
- Un mayor aprovechamiento de los recursos y reducción de la contaminación, creando una agricultura más sostenible.
- Mejora de la logística.
- Aumento en la calidad y cantidad de los productos.

1.2 Objetivos

El presente trabajo tiene como objetivo principal el análisis, diseño e implementación de un sistema IoT prototípico enfocado a la monitorización automatizada de un terreno de cultivo. Para ello se han planteado una serie de subobjetivos:

- Desarrollo de un sistema de sensorización automática del cultivo que facilite la obtención de aquellos parámetros que sean relevantes para la toma de decisiones.
- Análisis general de las tecnologías del ámbito IoT más adecuadas y de las ventajas derivadas de su aplicación en este escenario.
- Análisis de los diferentes enfoques y servicios proporcionados por las principales plataformas de computación en la nube y selección de los que mejor se adaptan al problema.
- Análisis de las tecnologías concretas de comunicación más adecuadas para el problema y selección del hardware más adecuado, poniendo especial atención en las últimas soluciones en redes de área amplia y baja potencia (LPWAN).
- Programación y configuración de los servicios necesarios en la nube para la integración y almacenamiento de los datos provenientes de todos los nodos de sensorización activos.

2 METODOLOGÍA

Para la elaboración de este trabajo se ha seleccionado una metodología basada en los principios de las metodologías iterativas e incrementales, definiendo un conjunto de iteraciones de desarrollo acorde con los objetivos del proyecto y los resultados esperados del mismo. Además, se han tenido en cuenta los siguientes aspectos:

- Coordinación constante con los tutores del TFG, que ejercerán el papel de clientes, participando en la definición de los requisitos y validando su cumplimiento. En esta línea ha sido útil el pensar en escenarios más complejos, ya que han mostrado sus necesidades, intercambiando ideas y realizándose mejoras y cambios hasta concluir unos objetivos finales.
- En cada iteración deberá completarse una o varias tareas de análisis o añadir alguna funcionalidad al sistema.
- En cada tarea de desarrollo, como norma general, se seguirá un ciclo clásico de análisis, diseño, implementación y pruebas.
- Las iteraciones se planificarán siguiendo ciclos cortos que posibiliten una pronta detección de posibles desviaciones.

Para llegar a cumplir con los objetivos establecidos se han definido, asimismo, los siguientes pasos clave a seguir:

1. Analizar en mayor detalle el ámbito del proyecto, en este caso el sector agrícola y las tecnologías aplicables al mismo en el contexto de la denominada *agricultura inteligente*.
2. Analizar las tecnologías existentes más actuales y valorar su adecuación al problema planteado. En concreto, se prestará especial atención a las que tengan una mayor difusión en el ámbito de IoT e IIoT, en particular las soluciones basadas en plataformas de computación en la nube.
3. Especificar de forma completa y detallada los requisitos de usuario del sistema, tanto a nivel de requisitos funcionales como no funcionales.
4. Definir los casos de uso, concretando como interactuará el usuario con el sistema para obtener la funcionalidad deseada.
5. Definir los nodos de sensorización necesarios para cumplir todos los requisitos del proyecto. En este punto también se documentarán las diferentes tecnologías software requeridas por cada uno de los dispositivos, y las opciones de compatibilidad con el resto del sistema.
6. Configuración de la plataforma en la nube para gestionar los datos provenientes de los sensores. La plataforma *cloud* ayudará a conseguir un sistema global mucho más flexible, independientemente del número de nodos, y facilitará la interacción del usuario cliente con el sistema. En este momento también surgirá la necesidad de determinar los diferentes escenarios que se pueden dar, para elegir la respuesta del sistema más adecuada.
7. Utilización de una base de datos para almacenar los valores medidos por los nodos sensorizados y poder realizar un historial, lo que permitirá prevenir problemas futuros con las cosechas.
8. Definición y ejecución de unas pruebas globales para confirmar que el prototipo cumple con los objetivos definidos. Esto también permitirá sacar conclusiones sobre el comportamiento del sistema cuando todos los dispositivos se encuentran integrados y evaluar el beneficio de emplear una plataforma en la nube de cara a un posible escalado del sistema a escenarios reales.

3 PLANIFICACIÓN

Para la planificación del proyecto se ha optado por dividir todo el proceso en una serie de actividades concretas estipulando las duraciones de estas. Esta planificación se ha realizado teniendo en cuenta los plazos de entrega y la búsqueda de márgenes de tiempo para poder hacer frente a los problemas que surjan, sin que alteren el objetivo. A mayores, como se muestra en el siguiente diagrama, también se contemplan los huecos de tiempo que se tendrá que invertir para la realización de exámenes del Grado e imposibilitarán el trabajo en este TFG.



Figura 2. Diagrama de Gantt. Fuente: Elaboración propia.

Las actividades planificadas son las siguientes:

- **A1:** Elección y planteamiento general del problema a resolver.
 - **A2:** Definición de objetivos.
 - **A3:** Búsqueda de información sobre las carencias en el sector agrícola y la aplicabilidad de las tecnologías actuales.
 - **A4:** Análisis y toma de decisiones sobre la tecnología de comunicación y plataforma *cloud* utilizada.
 - **A5:** Definición detallada de los requisitos de usuario.
 - **A6:** Corrección de la memoria redactada hasta el momento.
 - **A7:** Estudio y elección del hardware necesario para cumplir con los objetivos planteados.
 - **A8:** Adquisición y montaje del hardware.
 - **A9:** Configuración del software de los nodos de sensorización.
 - **A10:** Diseño global del sistema de manera detallada.
 - **A11:** Corrección de la memoria redactada hasta el momento.
 - **A12:** Integración de los dispositivos en la plataforma en la nube.
 - **A13:** Desarrollo y prueba de comunicaciones de tipo publicación/suscripción sobre protocolo MQTT.
 - **A14:** Desarrollo y prueba del almacenamiento persistente en una base de datos en la nube.
 - **A15:** Desarrollo y prueba de la solución para notificación de eventos al usuario.
 - **A16:** Desarrollo y prueba del prototipo de sistema global simulando una situación real.
 - **A17:** Corrección de la memoria final.
 - **A18:** Redacción de la memoria de manera paralela al resto de las actividades.

A medida que se ha desarrollado el trabajo han aparecido algunos problemas que obligaron a realizar ajustes en la planificación y reprogramar algunas actividades. En concreto surgió un retraso en los comienzos de las actividades de la 9 en adelante, debido a un problema de desabastecimiento de algunos de los recursos hardware que inicialmente se había planificado adquirir. Debido a esto surgió la necesidad de replantearse en parte el tipo de sensorización a utilizar, así como buscar proveedores alternativos. En concreto, la recepción del hardware de sensorización con soporte para comunicaciones LoRa se retrasó cerca de tres meses, por lo que algunas de las tareas dependientes del mismo tuvieron que retrasarse considerablemente.

4 ESPECIFICACIÓN

4.1 Estudios y análisis previos

El objetivo principal del presente Trabajo Fin de Grado es la elaboración de un prototipo experimental a pequeña escala mediante el cual se obtengan distintas cualidades del suelo y del ambiente, permitiendo así la monitorización del terreno de cultivo donde sea implementado.

Antes de diseñar e implementar el sistema es necesario profundizar en el conocimiento del dominio del problema, así como realizar diversos estudios y análisis que nos permitan tomar decisiones fundamentadas en lo referente a las tecnologías a utilizar, la arquitectura software de la solución o el hardware más adecuado.

Se parte de un concepto de sistema basado en los principios de los sistemas IoT más actuales, buscando una solución prototípica que en base a su arquitectura podría ser fácilmente escalable y extrapolable a un escenario realista con cientos o miles de dispositivos y grandes volúmenes de datos. Así, los pilares del sistema serán una solución de sensorización distribuida adecuada para el entorno del problema y una plataforma de computación en la nube con soporte para los servicios necesarios y adecuadas herramientas en los ámbitos de la privacidad y la seguridad.

De acuerdo con este planteamiento, se comenzará con una breve introducción al Internet de las Cosas e Internet Industrial de las Cosas, así como sobre los parámetros generales más relevantes en el ámbito de la monitorización de terrenos de cultivo. Inmediatamente a continuación, en este apartado se incluyen diferentes análisis sobre plataformas de computación en la nube e IoT existentes, hardware-software para la creación de los distintos nodos de sensorización remota, sensores existentes en el mercado para la monitorización del terreno o las tecnologías de comunicación adecuadas para las condiciones del entorno del problema.

4.1.1 Internet de las Cosas (IoT)

IoT o, dicho de otro modo, “Internet de las Cosas”, se puede definir como el conjunto de dispositivos conectados a Internet mediante una capa hardware y software de comunicaciones, la cual sirve de nexo entre el mundo real y el mundo digital [10]. En este mundo digital residen modelos de los objetos del mundo físico con el software necesario para dotarles de inteligencia (*smart objects*).

Dentro de los componentes que forman un *smart object*, se tiene:

- El objeto en sí mismo.
- Sensores, microprocesadores, almacenamiento de datos y software.
- Interfaces y protocolos de comunicación.

A su vez, estos objetos se pueden clasificar según el nivel de inteligencia que posean, de menor a mayor:

- Monitorización. El *smart object* puede hacer una monitorización, tanto del propio objeto como del entorno en el que se encuentra, pudiendo emitir notificaciones y alertas programadas.
- Control. Para ello existe un software, el cual puede ser o no embedido, que controla el funcionamiento del objeto.
- Optimización. En este caso no solo se controla el funcionamiento, sino que también se realiza una operación empleando algoritmos que optimizan algún parámetro de esta.
- Autonomía. Son objetos capaces de tomar decisiones ante circunstancias no preprogramadas.

Para que los sistemas IoT se empezasen a desarrollar y extender fue necesario alcanzar un nivel de madurez en todas las tecnologías involucradas. Estas tecnologías se clasifican en tres grupos principales:

- Hardware.
 - Sensores más capaces, a un precio y tamaño suficientemente reducidos.
 - Microprocesadores y microcontroladores cada vez más potentes, baratos y pequeños.
 - Redes inalámbricas con unas características de velocidad, latencia, alcance y consumo eléctrico adecuadas.
- Software.
 - Frameworks de desarrollo software completos y abiertos.
 - Sistemas capaces de operar en tiempo real.
 - Computación en la nube, incluyendo *big data*, *data analytics* y métodos de aprendizaje máquina.
- Índole sociológica.
 - Familiaridad con la informatización de las plantas industriales.
 - Generalización en el uso de dispositivos móviles conectados a Internet.

4.1.2 Internet Industrial de las Cosas (IIoT)

Cuando hablamos de Internet Industrial de las Cosas (IIoT), nos referimos, principalmente a la aplicación de soluciones y tecnologías IoT en el ámbito industrial. Esto hace referencia al uso de dispositivos, como sensores e instrumentos, conectados a través de Internet para lograr un mayor grado de automatización y autosupervisión en las máquinas y en los procesos industriales involucrados.

Aunque IIoT e Industria 4.0 no se refieren al mismo concepto, tienen una destacable cantidad de tecnologías comunes. IIoT es uno de los medios a través de los que se pueden conseguir los objetivos propuestos por la Industria 4.0. Entre estos objetivos destacan los siguientes:

- Llevar las tecnologías TIC a las líneas de producción.
- Unificar información generada por el cliente con la generada por las máquinas.
- Aprovechar la capacidad de comunicación entre máquinas.
- Tratar de que las líneas de producción sean lo más autónomas posible en su funcionamiento.

Si nos enfocamos en las particularidades de IIoT frente a IoT nos encontramos con las siguientes características:

- Los sistemas de producción suelen tener un ciclo de vida mucho mayor que los productos de consumo, con lo que la seguridad informática requiere de una mayor atención, en especial si se está implantando I-IoT en plantas pre-Industria 4.0.
- La propiedad intelectual, *know how*, es en muchos casos el bien máspreciado de una empresa.
- El software utilizado debe de estar muy probado por ser a menudo crítico.
- Con frecuencia, las soluciones IIoT deben coexistir con redes anteriores y con sistemas de Enterprise Resource Planning (ERP) ajenos al concepto IoT, con lo que una correcta interoperabilidad es importante y compleja de conseguir.
- Las soluciones IIoT deben poseer buenas cualidades en cuanto a escalabilidad, flexibilidad y facilidad de configuración.

- La importancia del *big data analytics*, es decir, de procesar de forma eficiente ingentes cantidades de datos para extraer información útil, es clave.
- Los *smart objects* pueden construirse en base a modelos matemáticos, pero también puede ser construido o mejorado a partir de los datos provenientes de sensores usando técnicas de aprendizaje máquina.
- Dada la cantidad de datos que en ocasiones se generan en una planta de producción, un sistema IIoT puede beneficiarse enormemente de este tipo de técnicas, en particular de las últimas técnicas basadas en métodos de aprendizaje profundo.

4.1.3 Parámetros relevantes

El éxito del desarrollo de los cultivos viene condicionado, en mayor o menor medida, por factores climatológicos [11]. Dentro de estos parámetros, los más importantes y a los que se les va a dar importancia en el presente proyecto, se encuentra la humedad y temperatura ambiente, temperatura y pH del suelo y la iluminancia en la zona.

Para ello, se medirá su valor, un número de veces razonable al día, gracias a la utilización de unos sensores específicos en cada caso. Posteriormente, dependiendo del cultivo elegido, se contrastará con unos valores óptimos y se dará notificación al usuario final si es necesario.

A continuación, se hará una breve explicación del motivo por el que se han escogido dichos parámetros en concreto y su influencia en el cultivo.

4.1.3.1 Temperatura y humedad del terreno

Ambas son condiciones fundamentales en el cultivo de las plantas y están en cierta medida relacionadas entre sí.

La temperatura del suelo afecta a la correcta germinación y desarrollo de la planta, afectando también biológica, química y físicamente a las características del suelo [12]. Esto es importante ya que influye en los nutrientes y materia orgánica que contiene el mismo lo que, a su vez, afecta a la calidad del cultivo.

De igual manera que la temperatura, el control de la humedad ayuda al correcto desarrollo de la planta [13]. La presencia de agua en el suelo sirve como autorregulador de temperatura y produce que se conduzca mejor el calor que en los suelos secos.

4.1.3.2 Temperatura y humedad del ambiente

De igual manera que en el caso de la temperatura y humedad del suelo, ambos parámetros están vinculados entre sí. En este caso, la humedad del ambiente varía en función de los cambios de temperatura que se produzcan.

Si el aire contiene unos niveles bajos de humedad, las plantas cerrarán sus estomas para que la pérdida de agua sea menor. Esto conlleva que la fotosíntesis sea más lenta, y que los cultivos tarden más tiempo en alcanzar el tamaño y la calidad adecuada para su venta. De manera contraria, si los niveles son demasiado altos, se producirán enfermedades en las hojas y raíces de los cultivos, afectando de manera negativa a su crecimiento [14].

En el caso de la temperatura ambiente, esta interviene en la apertura y cierre de los estomas de las plantas, influyendo negativamente en su correcto funcionamiento. En caso de que la temperatura sea excesivamente alta, la planta cerrará sus estomas para perder la menor cantidad posible de agua, y como ocurre cuando la humedad es baja, la fotosíntesis se producirá de manera más lenta.

4.1.3.3 Iluminancia

El término iluminancia hace referencia al flujo luminoso que incide en una superficie. Aplicado a la agricultura ayuda a sacar conclusiones importantes, ya que la realización

de la fotosíntesis depende en gran medida de la cantidad de luz que llega al cultivo [15]. Es decir, un bajo crecimiento de los cultivos puede deberse a la falta de luz, lo cual deriva en falta de clorofila en las hojas.

4.1.3.4 pH del terreno

El valor del pH de la tierra influye de manera muy directa en el desarrollo, y su medición es interesante ya que tiene una relación directa con la disponibilidad de nutrientes, permitiendo así sacar ciertas conclusiones sobre posibles carencias que tengan las cosechas. El nivel de acidez afectará a:

- Disponibilidad de nutrientes: se sabe que en un suelo que tiene un pH alto se encuentran en mayor medida nutrientes como el potasio y el azufre, mientras que, en el caso contrario, destacaría el hierro. Esto es importante ya que, aunque la tierra esté perfectamente abastecida de la mayoría de los nutrientes, el que se encuentre en menor medida será el que condicionará el crecimiento de la planta. Por otro lado, los valores extremos del pH, tanto al alza como a la baja, pueden provocar el precipitado de ciertos nutrientes. Esto quiere decir que dichos elementos se encontrarían en estado sólido, lo cual no conviene, ya que las plantas no serían capaces de digerirlos.
- Proceso fisiológico de absorción de las nutrientes: todos los cultivos tienen unos rangos de pH en los que su absorción de nutrientes es adecuada. De esta manera, si dichos valores se incumplen, la absorción por las raíces se ve perjudicada y, en algunos casos, incluso se deterioran y presentan toxicidades debido a la desmesurada ingesta de nutrientes.

4.1.4 Monitorización inteligente de suelos

Dentro de la agricultura inteligente este proyecto se centra en la monitorización de los terrenos agrícolas a gran escala, permitiendo medir determinadas propiedades útiles cara a la toma de decisiones. Estas propiedades abarcan desde parámetros comunes en el suelo y ambiente, como temperatura y humedad, hasta otros más específicos como el nivel de pH y nutrientes en el terreno, o la iluminancia y calidad del aire que inciden en la zona.

Como ya se ha mencionado, la recopilación de todos estos parámetros permite la toma de decisiones en las diferentes fases del desarrollo de la planta, pudiendo evaluar cómo se desenvuelve el crecimiento de la misma y actuar en consecuencia, previniendo problemas a futuro. Un claro ejemplo de esto es la vinculación con un sistema de riego automatizado que se acciona cuando los valores de humedad y temperatura son demasiado elevados como para garantizar la calidad del cultivo. A pesar de ello, hay momentos en los que los valores de los parámetros, sobre todo de los ambientales, pueden variar de manera repentina y no por ello es necesario intervenir. Por lo que, de manera paralela a la implementación remota con actuadores, cumple un papel fundamental la notificación de alertas a los usuarios en todo momento, ya que en gran medida serán los encargados de tomar decisiones e incluso poder llegar a variar la planificación de la explotación.

Para poder implementar un sistema IIoT que aporte todos estos beneficios, hay que enfrentarse a una serie de retos [16]:

- En un primer lugar se encuentra la conectividad. Tanto la conexión a Internet, en el caso de utilizar servicios en la nube, como entre los diferentes dispositivos del sistema, una adecuada conectividad es, obviamente, fundamental. Por esta razón surge la búsqueda de una tecnología de comunicación adecuada, con bajo coste y alta eficiencia energética.
- El entorno agrícola es mayoritariamente exterior y variable, siendo inapropiado para instalaciones complejas e incrementándose considerablemente el coste para poder llevar dispositivos de medición a diferentes puntos.

- La alimentación eléctrica de los dispositivos juega un papel importante debido a las grandes dimensiones de las zonas agrícolas, optándose habitualmente por la utilización de baterías o paneles solares. Por esta razón los dispositivos empleados deben tener un consumo especialmente bajo, ya que lo contrario podría provocar interrupciones y costes indeseables debido a los cambios o recargas.
- La seguridad y privacidad también son conceptos a valorar, ya que el riesgo de robo de información y ciberataques preocupa en gran medida a los responsables de las explotaciones. Como consecuencia, la solución deberá contar con sistemas de cifrado y autenticación.
- Se busca también que todo el sistema sea lo menos agresivo que sea posible con el entorno, evitando grandes instalaciones que afecten a la fauna y a las propias plantas que se desean cultivar. Dentro de estas limitaciones se encuentran las antenas, utilizadas para conseguir una comunicación estable, la colocación de postes para ubicar en ellos el sistema de cableado, o mismo la ocupación del suelo por dispositivos, disminuyendo así el espacio disponible y perjudicando el desarrollo de los cultivos.

4.1.5 Tecnologías de comunicación relevantes

El término tecnologías de comunicación hace referencia a la tecnología necesaria para transmitir información entre puntos distantes, en este caso entre los elementos de sensorización/actuación y la plataforma de IoT, o entre distintos procesos y servicios de la plataforma de IoT sobre la que se ejecutará el núcleo del sistema. Por lo tanto, será necesario realizar un estudio y análisis sobre las tecnologías más relevantes en este ámbito y finalmente concluir cuáles se adaptan mejor a la resolución del problema. En el sistema global que se propone hay cuatro escenarios principales en lo referente a tecnologías y protocolos de comunicación:

1. Los nodos de sensorización próximos a las instalaciones de control, que pueden implementarse sobre un dispositivo tipo Raspberry Pi, se comunicarán directamente con la plataforma en la nube mediante una red convencional, como puede ser una red WiFi existente en las instalaciones o desplegada a tal fin, y que tenga salida a Internet. Igualmente podría utilizarse una conexión cableada tipo ethernet si fuese factible en un espacio físico determinado.
2. La conexión entre los dispositivos que realizarán la sensorización del terreno y del ambiente y el resto del sistema implicará una tecnología de comunicación diferente a las más convencionales que cumpla con los requisitos particulares del sistema, ya que las zonas a cubrir pueden ser amplias y remotas, y a esto se añade la dificultad de la alimentación eléctrica de los dispositivos. Esta problemática se discutirá más ampliamente a lo largo de este apartado.
3. Para la conexión con los servicios de la plataforma alojados en la nube se necesitará una conexión a Internet segura. En última instancia, desde los nodos de sensorización, sean del tipo que sean, será necesario redirigir el tráfico a través de un elemento con salida a Internet.
4. La comunicación entre distintos servicios/procesos que se ejecutarán en la propia plataforma en la nube se resuelve de forma transparente para el desarrollador simplemente utilizando las APIs de programación proporcionadas por la plataforma en la nube seleccionada, con independencia de en dónde se esté ejecutando físicamente cada servicio/proceso.

Por tanto, el punto más delicado en cuanto a las comunicaciones es el correspondiente al envío de datos desde los campos de cultivo que se desea monitorizar. La primera decisión en este sentido es la de utilizar una tecnología inalámbrica. La gran ventaja de las redes cableadas reside en su velocidad de

transmisión/recepción y su calidad de servicio, pero debido a que las necesidades en ese sentido son bajas, ya que no se realizará la transmisión de grandes volúmenes de datos, esta característica no es condicionante frente a las ventajas que proporciona una red inalámbrica:

- Posibilidad de mover los dispositivos a cualquier posición dentro del área de cobertura.
- Se evita desplegar cableado de largas dimensiones entre el centro de control y los dispositivos.
- Facilita conectar un mayor número de dispositivos y aumentar su número de forma rápida y sencilla.

4.1.5.1 Tecnologías de comunicación inalámbrica

Haciendo hincapié en estas tecnologías, existen dos tipos diferenciados. Por un lado, se encuentran las comunicaciones de corto alcance, y por el otro las de largo alcance, variando entre centímetros y kilómetros, respectivamente.

A continuación, se muestra un gráfico donde se comparan la velocidad de la transmisión de datos y energía consumida con el rango de cobertura. Recordando los requisitos del sistema, se necesita una red que proporcione cobertura a zonas de cultivo que pueden llegar a ser de muchas hectáreas y con un consumo de energía bajo, acorde a las necesidades de transmisión de datos.

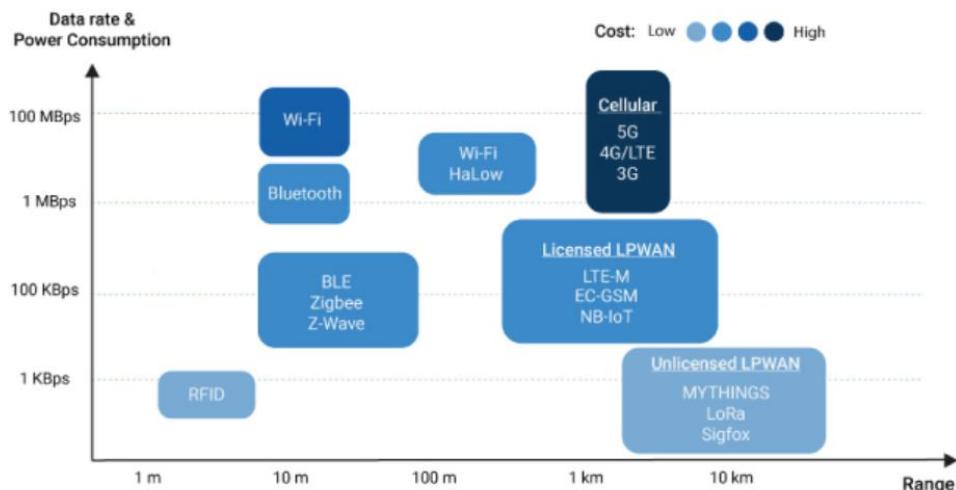


Figura 3. Gráfico comparativo de las tecnologías de comunicación inalámbricas más relevantes. Fuente: [17].

De esta forma las tecnologías como Wi-Fi, Bluetooth o Zigbee no serían suficientes para cumplir con las necesidades expuestas, ya que tienen un consumo muy elevado en comparación a las distancias que abarcan, por debajo de 1km.

En cuanto a las redes de telefonía móvil, estas tienen un rango de cobertura mucho mayor, pero su consumo sigue siendo elevado. A mayores, estas redes de comunicación son contratadas a las compañías telefónicas, lo que incrementa el coste de servicio además de la dependencia de un tercero.

Finalmente, las tecnologías de comunicación más apropiadas para el proyecto son LPWAN. Estas tecnologías se crearon específicamente para IoT e IIoT con el objetivo de solventar los problemas que aparecen con la utilización de las tecnologías tradicionales. Las características que las hacen adecuadas en estos escenarios son:

- Son redes de baja potencia y área amplia, lo que permite monitorizar zonas extensas con un bajo consumo de energía, utilizando dispositivos dotados de baterías.
- Debido a la transferencia de pocos datos y con baja frecuencia, las baterías son capaces de alimentar a los dispositivos durante años, lo que disminuye notablemente el coste por mantenimiento en las instalaciones.

Pertenecientes a estas tecnologías se encuentran LoRa y Sigfox, ambas con un alcance entorno a 10 km. Respecto a Sigfox, su transferencia de datos es muy pequeña, 140 mensajes diarios, por lo que limitaría bastante la cantidad de nodos de sensorización y el intervalo de transmisión [18].

Teniendo en cuenta este estudio previo, y prestando especial atención a las problemáticas con las que se encuentra este proyecto, la elección final es utilizar la tecnología LoRaWAN. Esta resuelve la mayoría de las problemáticas con las que se encuentra la monitorización de los cultivos y proporciona una solución asequible en cuanto a costes y a facilidades a la hora de crear un sistema a gran escala. En referencia a la seguridad, LoRaWAN implementa el encriptado de extremo a extremo y los dispositivos cuentan con una clave AES de 128 bits y un identificador EUI-64, por lo que resulta una opción atractiva a utilizar para proyectos en los que juegue una parte importante la privacidad de la información.

4.1.5.2 LoRaWAN

LoRaWAN define una arquitectura de comunicación inalámbrica de extremo a extremo que proporciona una solución de conectividad de bajo consumo, largo alcance, bajo coste, segura y escalable [19]. Se suele utilizar en dispositivos con muy bajas necesidades de transmisión de datos y baja potencia computacional, lo que hace posible que en muchos casos las baterías de los dispositivos duren años.

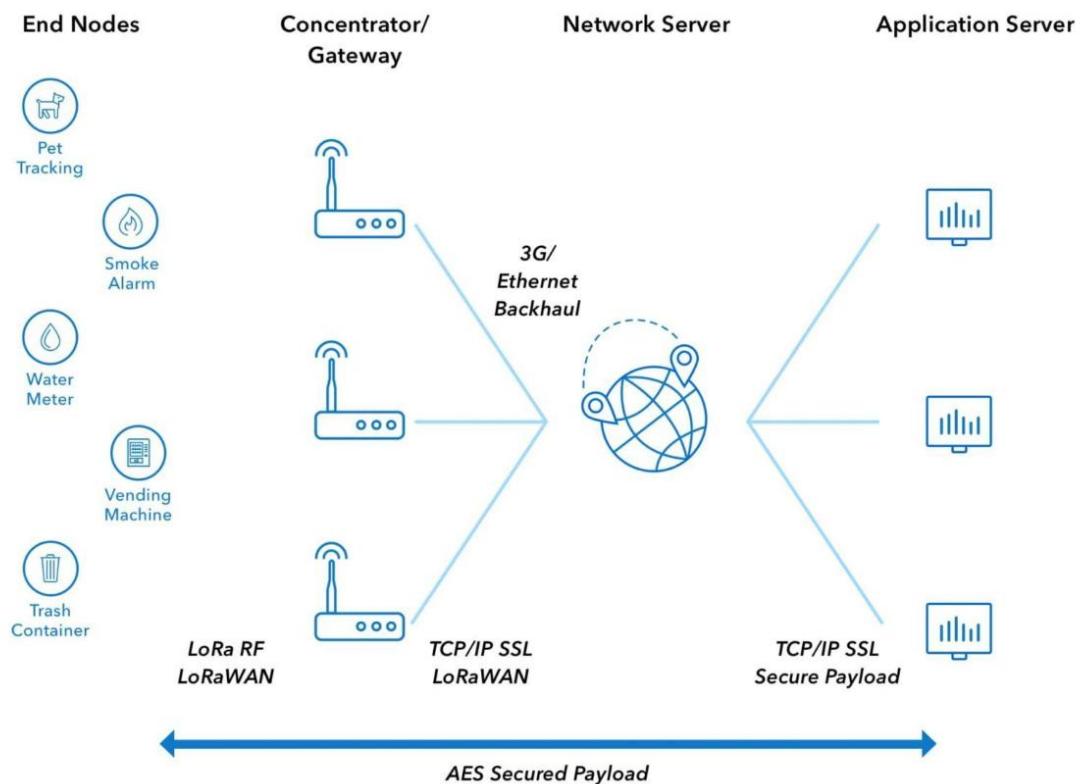


Figura 4. Arquitectura LoRaWAN. Fuente: [20].

Esta arquitectura utiliza la capa física LoRa y se basa en un estándar de cifrado avanzado de 128 bits para garantizar la seguridad de la red y, gracias a su espectro ensanchado, permite aplicaciones de ubicación sin utilizar GPS.

Toda la arquitectura y los protocolos del sistema están siendo desarrollados por LoRa Alliance, que es una asociación abierta que se encuentra en constante crecimiento y que abarca todo tipo de ámbitos relacionados con IoT, desde la fabricación de chips hasta proveedores de la nube.

4.1.6 Plataformas en la nube

Para la gestión e integración de los diferentes datos que nos proporcionarán los sensores remotos, lo más adecuado es utilizar una plataforma software de IoT que integre los servicios adecuados y sea fácilmente escalable en cuanto a los recursos computacionales utilizados según las necesidades presentes y futuras. Igualmente, en este ámbito, es importante que dicha plataforma proporcione soluciones avanzadas para garantizar la privacidad y seguridad en todos los puntos del sistema.

Los tres grandes líderes a nivel mundial en el ámbito de las plataformas de computación en la nube, que incluyen, entre otros muchos, servicios para integración de datos en soluciones IoT, son Amazon Web Services, Microsoft Azure y Google Cloud. Estos tres actores abarcan actualmente entorno al 80% del mercado de los servicios en la nube.

4.1.6.1 Amazon Web Services

Amazon Web Services, o AWS, es una plataforma software que integra el conjunto de herramientas y servicios de computación en la nube que ofrece Amazon. Esta plataforma es la líder mundial en el mercado de los servicios de computación en la nube, y fue el primero en proporcionar soluciones concretas e individualizadas al ámbito de IoT. Esto provoca que sea tan conocido y utilizado en este ámbito, sumándole que hoy en día sigue siendo la compañía que más ofertas enfocadas al Internet de las Cosas proporciona. Entre todos los servicios que ofertan se encuentra más de 200 con los que poder crear un entorno completo de computación en la nube dónde desplegar aplicaciones de muy diferente tipo.

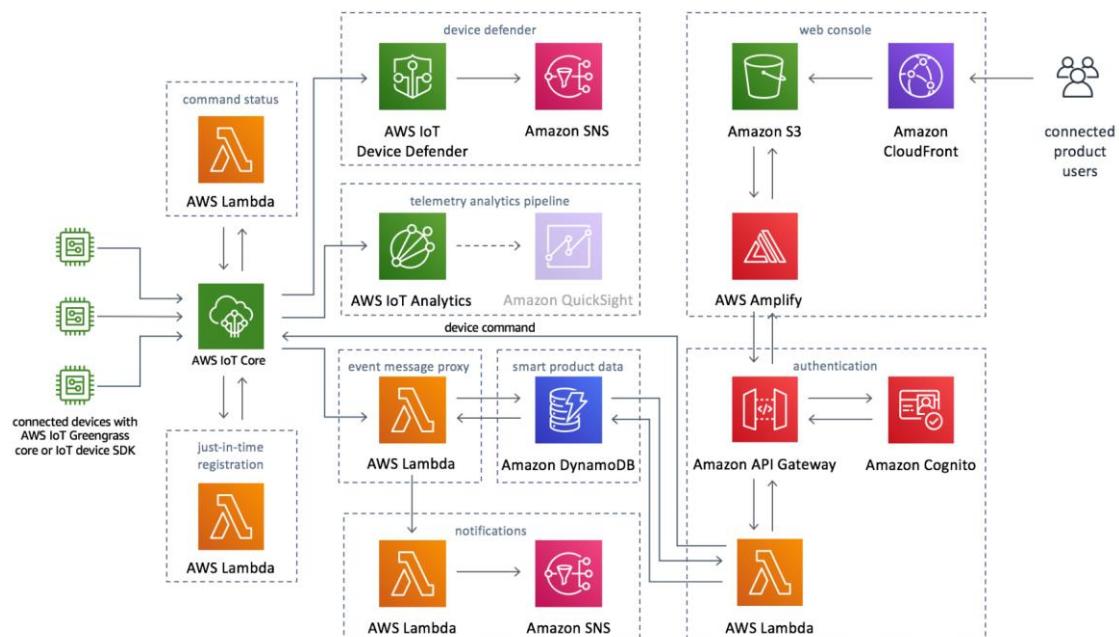


Figura 5. Arquitectura Amazon Web Services. Fuente: [22].

Algunos de los componentes de AWS más relevantes para IoT en general y para este trabajo en particular son [21]:

- AWS Greengrass: permite crear, implementar y administrar software de dispositivos.
- AWS IoT Core: es el servicio que permite conectar los dispositivos de IoT y dirigir los mensajes a otros servicios de AWS.
- AWS IoT Analytics: ejecución e instrumentación de análisis avanzados de grandes cantidades de datos de IoT.

En la figura previa se muestra la relación entre estos y otros componentes especialmente relevantes de la plataforma AWS:

4.1.6.2 Microsoft Azure

Microsoft Azure es una plataforma de servicios de computación en la nube que ofrece la empresa Microsoft. Esta plataforma está diseñada con el fin de poder ofrecer soluciones a los problemas actuales, siendo la más utilizada en ciertos ámbitos empresariales [23].

Los componentes más relevantes de Azure relacionados con IoT:

- IoT Hub: es el servicio que actúa de conexión entre los dispositivos y una aplicación de IoT, permitiendo así administrarlos.
- Stream Analytics: está diseñado para realizar un análisis en tiempo real de cargas de trabajo críticas.
- Azure Data Lake: repositorio para grandes cantidades de datos.
- Data Lake Analytics: conjunto de herramientas para análisis avanzados de datos.
- Time Series Insights: configurador y visualizador de flujos de datos de IoT.

El siguiente diagrama muestra la arquitectura de componentes que sigue Azure:

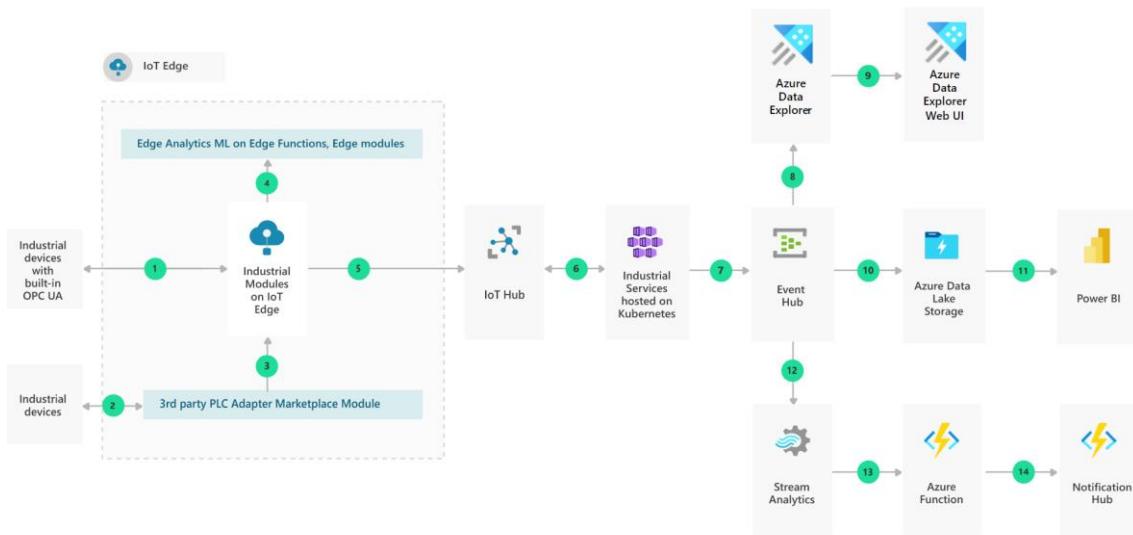


Figura 6. Arquitectura Microsoft Azure. Fuente: [24].

4.1.6.3 Google Cloud Platform

Google Cloud Platform es la plataforma en la nube proporcionada por Google, y reúne todos los servicios y aplicaciones para desarrollo en la nube que Google ofrecía previamente por separado [25].

De todos los componentes que Google ofrece relacionados con IoT, los más importantes son [26]:

- Cloud IoT Core: recibir datos y gestionar dispositivos a través de un concentrador de datos IoT Hub.
- Cloud Pub/Sub: servicios de comunicación de datos según el patrón publicación/subscripción.
- Cloud Dataflow/Cloud Functions: soluciones para implementar procesos en la nube para tratar datos.
- Cloud Bigtable/Cloud Storage: interfaz común para soluciones de almacenamiento de datos.
- Cloud Machine Learning (ML): métodos y herramientas de aprendizaje automático.
- Cloud BigQuery: almacenamiento y análisis de petabytes de datos.
- Google Data Studio: visualización de datos.

La arquitectura interna de Google Cloud Platform es la siguiente:

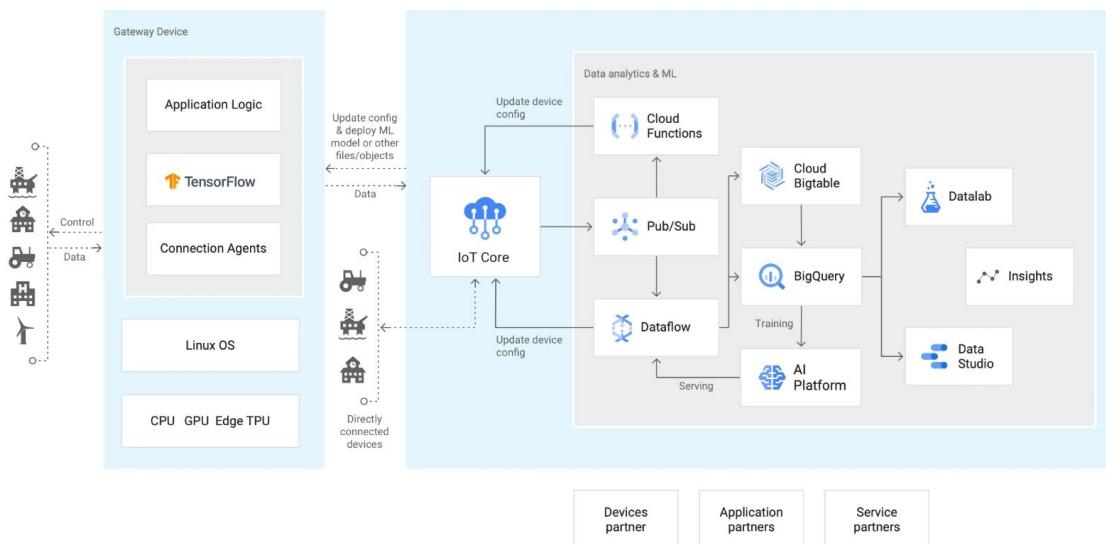


Figura 7. Arquitectura Google Cloud Platform. Fuente: [26].

4.1.6.4 Selección de la plataforma de computación en la nube

Tras analizar y probar las diferentes características de las plataformas en la nube previamente mencionadas, se puede concluir que las tres proporcionan una cobertura de servicios y herramientas muy amplia y similar en muchos aspectos, incluyendo opciones de acceso gratuito, bajo ciertas condiciones de uso, a un gran número de sus servicios, de modo que los usuarios puedan desarrollar y probar prototipos de sus sistemas sin coste.

La plataforma que finalmente se ha seleccionado para la realización de este trabajo es Amazon Web Services. Esta elección se hizo basándose en que es la que posee una cantidad mayor de servicios enfocados a IoT con soporte para un uso gratuito de los mismos. Entre estos recursos, es particularmente interesante para este trabajo el servicio AWS IoT Core for LoRaWAN, que permite conectar de forma sencilla todos los dispositivos inalámbricos y las puertas de enlace privadas en la nube, mediante el servidor de red LoRa LNS [27]. Esto es una diferenciación clave entre dicha plataforma y el resto, ya que, en el presente trabajo, se va a utilizar la tecnología de comunicación LoRaWAN, entre otras.

Para comprender mejor la ventaja de utilizar este recurso es necesario atender a los componentes obligatorios en una red LoRaWAN, en concreto a los Gateways o puertas de enlace [28]. Este elemento será el encargado de recibir los paquetes de datos enviados por los nodos de sensorización y redirigirlos a un destino y será imprescindible emplear dos, uno en modo Packet Forwarder (o Basic Station [29]) y otro como Network Server.

- Modo Packet Forwarder. Se ejecuta un programa en la puerta de enlace que funciona como un reenviador de paquetes a través de un enlace IP, es decir, posibilitando la transmisión de datos provenientes de sensores en la red LoRAWAN hacia una red IP convencional.
- Modo Network Server. Permite conectar los nodos de la red LoRa con las aplicaciones superiores, garantizando la seguridad de la red [30].

Además, este servicio que proporciona AWS para integrar datos provenientes de una red LoRaWAN, el Gateway correspondiente al Network Server residirá en la propia plataforma, por lo que no habrá que desplegar un servicio en un servidor físicamente independiente para el desarrollo del proyecto. Esto hace que sea más sencilla la integración de todos los nodos en el sistema gracias al resto de servicios ofrecidos por la plataforma para dirigir, procesar y almacenar los datos recibidos desde los nodos de sensorización.

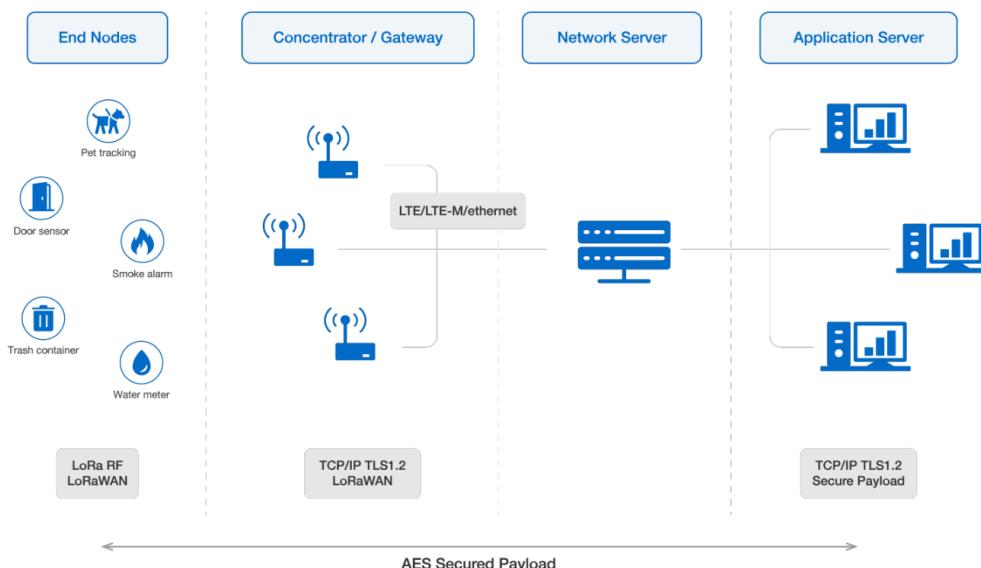


Figura 8. Diferentes modos del gateway en una red LoRa. Fuente: [28].

Otro aspecto que se ha analizado, y ha motivado en gran medida la elección, es que AWS es la plataforma más utilizada a nivel mundial, con un 54,22% de usuarios.

4.1.7 Lenguajes de programación

Como ya se ha concluido, se utilizará Amazon Web Services para procesar, organizar y analizar los datos. Para que todo esto sea posible es necesario realizar diferentes desarrollos de software, tanto en los nodos remotos de sensorización como en servicios en la nube de la propia plataforma AWS.

La plataforma AWS proporciona soporte para el desarrollo en varios lenguajes de programación ampliamente extendidos. En el presente proyecto se empleará Python como lenguaje principal para el desarrollo de los componentes software que se ejecutan en la nube y para la programación de los nodos de sensorización.



Figura 9. Estadística sobre la utilización de plataformas Cloud. Fuente: [31].

No obstante, en el caso concreto de algunos de los dispositivos LoRa empleados, será necesario utilizar el lenguaje C/C++ para la programación de las tareas de lectura y envío de datos. Asimismo, los procesos en la nube que reciben datos remotos provenientes de dispositivos LoRa serán programados en JavaScript por la facilidad de uso de las librerías JavaScript ya disponibles en AWS.

A mayores, SQL será utilizado como lenguaje de consultas sobre la base de datos que se implementará en AWS [32].

4.1.8 Protocolos de comunicación

En este proyecto se desea construir un sistema escalable, permitiéndose así tanto añadir como eliminar dispositivos y nodos sin afectar al resto del funcionamiento, consiguiendo un gran nivel de independencia entre elementos [33][34]. A mayores, también hay que prestar especial atención a la seguridad y privacidad de los datos, al tratarse de un sistema altamente distribuido y deslocalizado.

Al tratarse de una solución distribuida y heterogénea, se utilizarán diferentes protocolos de comunicación, según las necesidades, desde protocolos de envío de mensajes típicos de IoT como MQTT, al protocolo HTTPS común en Internet para comunicaciones con plataformas en la nube [35].

4.1.8.1 Protocolo de comunicación HTTPS

HTTP y HTTPS (versión segura) son los protocolos más habituales de Internet, utilizados típicamente en comunicaciones con servidores web [36]. Siguen el patrón cliente/servidor, dónde la máquina cliente hace peticiones a un servidor y éste le devuelve los resultados, existiendo una comunicación directa entre ambos.

HTTP no está enfocado específicamente a escenario de IoT, pero tiene gran difusión también en este ámbito, gracias a que, en conjunto con APIs REST, permite acceder de forma estándar a recursos disponibles en los servidores de procesos de IoT [37].

4.1.8.2 Protocolo de comunicación MQTT

Es un protocolo de comunicación basado en envío/recepción de mensajes ligero que utiliza el patrón publicador/suscripción, y fue creado principalmente para comunicación M2M, Machine to Machine, por lo que es muy utilizado en comunicaciones de dispositivos de IoT [38][39].

Dicho patrón funciona permitiendo que un dispositivo, llamado cliente, se conecte a un servidor central, llamado bróker, el cual es el encargado de enviar, recibir y distribuir

los mensajes a los distintos emisores o temas (*topics*). Para poder recibir los mensajes publicados en los temas, un nodo debe realizar una suscripción. A un mismo *topic* pueden suscribirse dinámicamente múltiples dispositivos sin necesidad de que se conozcan previamente entre ellos. Este modelo favorece enormemente la escalabilidad a la hora de añadir o quitar dispositivos en un sistema IoT complejo.

Todas las conexiones establecidas pueden ser autenticadas y cifradas siguiendo los modelos estándar más habituales.

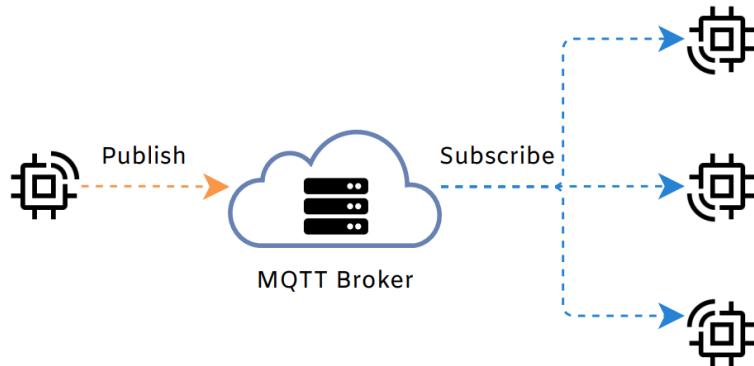


Figura 10. Comunicación publicación/suscripción. Fuente: Elaboración propia.

4.1.9 Sensorización distribuida

En lo referente a qué dispositivos se van a utilizar para medir características del entorno y del terreno, hay dos aproximaciones claramente diferenciadas: la utilización de sensores que ya tengan implementados la tecnología LoRaWAN, o sensores que carezcan de dicha tecnología.

Como elección final se ha escogido la utilización de unos sensores que tengan dicha tecnología de comunicación ya integrada al existir opciones en el mercado. La aproximación contraria, aumentaría la complejidad del proyecto, al tener que implementar y conectar de forma separada dispositivos capaces de actuar de concentradores de las señales de los sensores para su transmisión a través de una red LoRaWAN. Para tomar esta decisión, se ha hecho un análisis de viabilidad y costes para ambos casos.

En una solución con sensores convencionales que no incluyan capacidades de comunicación LoRaWAN, estos dispositivos deberían ir conectados a otro dispositivo (ej. Raspberry Pi o microcontrolador), donde este sería el encargado de enviar los datos obtenidos al Gateway mediante una antena LoRa. Esto implicaría la colocación de cierta cantidad de cableado, además del necesario empaquetado del conjunto en una caja apta para su instalación en exteriores.

Por el contrario, el coste de sensores convencionales es notablemente menor que el de aquellos que se pueden comunicar por sí mismos con un Gateway LoRa. Además, con una única Raspberry Pi o un microcontrolador es posible controlar varios sensores básicos.

Utilizar sensores con soporte para LoRa también implica que hay que limitarse a un conjunto mucho más reducido de dispositivos y a proveedores concretos, dado que se trata de una tecnología de uso incipiente. Por el contrario, al tratarse de una solución empaquetada y probada por el fabricante se corren menos riesgos en referencia al deterioro del sistema y también a posibles fallos en la comunicación.

En este proyecto, además de la sensorización de los terrenos de cultivo, también se contempla la problemática de introducir nodos de sensorización/actuación en instalaciones en las que sí se dispone de acceso a redes de comunicación

convencionales y a alimentación eléctrica. Para estos casos existen opciones baratas y potentes para crear un nodo capaz de controlar varios sensores/actuadores y con amplias capacidades de comunicación, como es el caso de Raspberry Pi. Esto tiene especial sentido si se desea monitorizar algún elemento que esté directamente relacionado con una explotación agrícola, pero a su vez pertenezca a una instalación local, como puede ser el control automático del nivel de fertilizantes almacenados.

4.2 Requisitos de usuario

Para poder realizar el proyecto correctamente y cumplir con los objetivos planteados es necesario definir claramente, para tener en cuenta a lo largo de todo el trabajo, los requisitos del sistema desde el punto de vista del usuario que lo va a utilizar. Esto hará que el sistema diseñado finalmente se adapte completamente a las necesidades del cliente.

Estos requisitos se pueden clasificar en funcionales o no funcionales, según si nos referimos a lo que se espera que haga el sistema o a las propiedades de este, respectivamente.

Requisitos funcionales:

- RF01. Consulta de datos actuales e históricos para los valores de los parámetros que permiten evaluar el estado del terreno y del ambiente. Deberá disponerse de una frecuencia de medición adecuada para cada parámetro.
- RF02. Capacidad de gestión de los distintos dispositivos conectados a la plataforma de forma remota y centralizada, con independencia del número actual de dispositivos.
- RF03. Disponer de un método centralizado de envío de alertas a los usuarios interesados a través de email, en caso de que los valores medidos estén fuera de los rangos configurados para el terreno.
- RF04. Posibilidad de monitorizar de forma remota y automática el estado de determinados elementos en las instalaciones del cliente, como el nivel actual de fertilizante almacenado.

Requisitos no funcionales:

- RNF01. Privacidad en el envío y almacenamiento de los datos. Debe utilizarse algún mecanismo de cifrado en todas las transmisiones de datos.
- RNF02. Seguridad en el acceso al sistema con el fin de evitar suplantaciones de identidad o accesos a los datos por parte de usuarios no autorizados.
- RNF03. Facilidad en el despliegue y bajo mantenimiento de los elementos de sensorización de los cultivos. Se debe priorizar el bajo consumo energético, minimizando la transferencia de datos en estos casos.
- RNF04. Posibilidad de detección remota de fallos, a través de la plataforma, como pérdidas de conectividad de dispositivos tras períodos excesivos sin transmisión.
- RNF05. Alcance de las comunicaciones que permita cubrir terrenos amplios y alejados de las instalaciones.
- RNF06. Bajo coste atribuible a la transmisión remota de datos, aun cuando se despliegue un gran número de sensores.

5 DISEÑO E IMPLEMENTACIÓN

En este capítulo se describe el diseño e implementación del sistema, así como las pruebas parciales y globales del mismo que se han realizado. Siguiendo la metodología de desarrollo descrita previamente, se realizan una serie de iteraciones a través de las que se van añadiendo y probando nuevas funcionalidades y componentes del sistema. En el último apartado del capítulo se describe el escenario implementado para las pruebas globales.

En cada uno de los puntos se incluye, además, la justificación de las decisiones de diseño adoptadas.

5.1 Vista global del sistema

Como primer paso para llevar a cabo el diseño final del sistema de monitorización de cultivos, es necesario concretar la arquitectura global del mismo, incluyendo los distintos componentes y la comunicación o relaciones existentes entre ellos, pero sin entrar en el diseño interno de detalle de cada uno.

En la siguiente ilustración se muestra una representación gráfica de la arquitectura diseñada para el sistema. Los principales componentes de esta arquitectura son los siguientes:

- Nodos de sensorización programables.
 - Nodos de tipo D1, unidos físicamente con un ordenador de placa reducida, Raspberry Pi, que les proporciona capacidad de programación y de comunicación mediante WiFi. Solución para sensorización/actuación en las instalaciones del cliente.
 - Nodos de tipo D2: corresponden con sensores dotados de capacidad de comunicación mediante LoRaWAN, por lo que se comunicarán de manera inalámbrica, utilizando dicha tecnología, con la puerta de enlace correspondiente. Solución para sensorización/actuación en los terrenos de cultivo.
- Solución de envío de mensajes a la nube con autenticación y cifrado utilizando el protocolo MQTT.
- Se empleará la plataforma Cloud de Amazon Web Services para gestionar los diferentes datos que proporcionan los nodos de sensorización. Asimismo, en la plataforma se ejecutarán los procesos automáticos de almacenamiento, consulta y detección de eventos relevantes y envío de alertas.
 - Para poder conectar los dispositivos a la plataforma y posteriormente dirigir los mensajes a diferentes *topics* se hará uso de los componentes IoT-Core y AWS-Lambda. Este último permitirá ejecutar procesos con código propio en la propia plataforma de forma que se asignen los recursos computacionales necesarios automáticamente y se facture al cliente únicamente por los recursos consumidos.
 - Otro componente indispensable es el de almacenamiento persistente, para lo cual se utilizará el recurso que mejor se adapta al problema concreto (Dynamo DB).
 - Se utiliza el componente Simple Email Service (SES), junto con los procesos AWS-Lambda, para programar el procesamiento de los datos y el envío de notificaciones de alertas vía correo electrónico a los usuarios registrados.

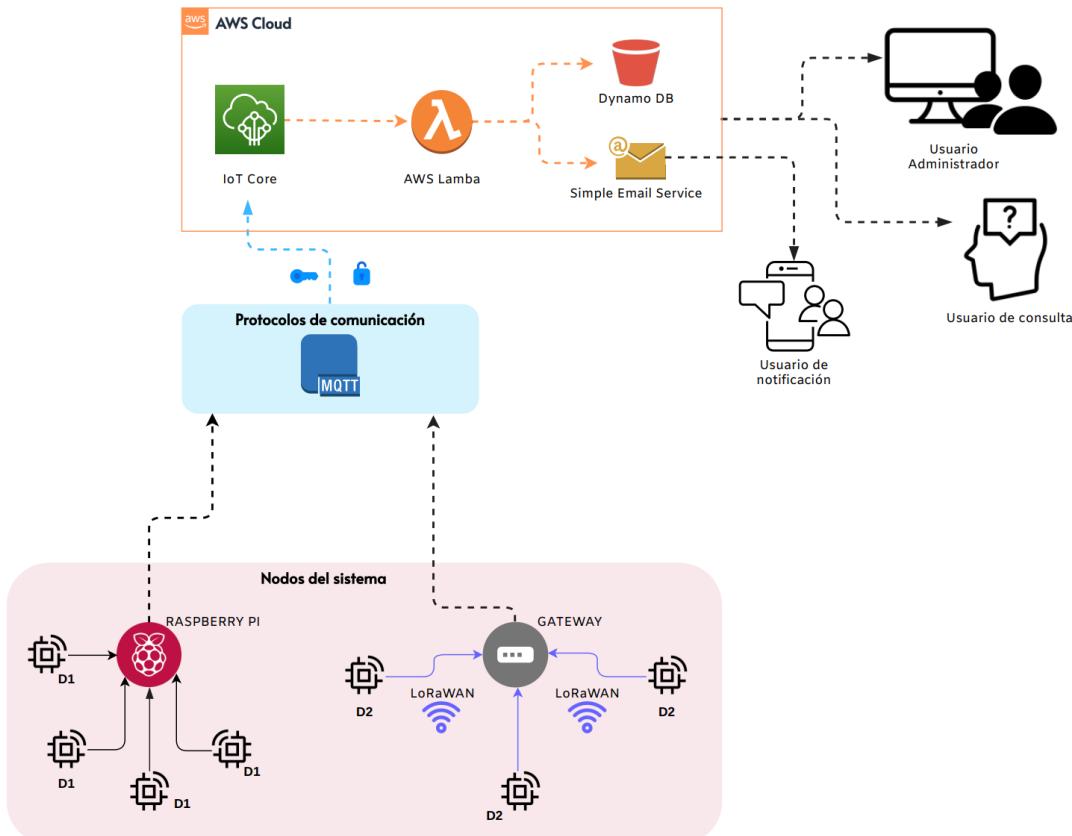


Figura 11. Vista global del sistema. Fuente: Elaboración propia.

5.2 Sensorización del entorno

Para realizar una buena evaluación del estado del terreno y automatizar otras posibles tareas, es necesario distribuir una serie de nodos de sensorización sobre los terrenos de cultivo y en otras instalaciones. Estos nodos, como se ha mencionado repetidas veces a lo largo de la memoria, están formados por los diferentes sensores LoRa en un caso y por nodos basados en Raspberry Pi en otro caso. La cantidad de sensores empleados puede ser variable, permitiendo así al propietario incrementar o decrementar los nodos según sus necesidades, creando una solución escalable en todo momento sin necesidad de reprogramación del sistema.

A continuación, se explicará de manera detallada y concreta los dispositivos utilizados en cada caso. Se ha de destacar que, a la hora de seleccionar los sensores adecuados, se tuvo en cuenta que algunos deben estar preparados para trabajar en el exterior.

5.2.1 Hardware

5.2.1.1 Raspberry Pi

Para construir los dispositivos que forman el sistema de control de fertilizante, se utilizará un ordenador de placa reducida, también conocidos como SBC (Single Board Computer) con conectividad WiFi.

En este caso concreto se utilizará la Raspberry Pi 4 Model B. En un primer momento, el objetivo principal de su fabricación fue la de estimular la enseñanza de la informática a las aulas pero, debido a su gran éxito, se empezó a implementar tanto en ámbitos que van desde la robótica a aplicaciones de IoT [40].



Figura 12. Raspberry Pi 4 Model B. Fuente: [41].

Entre sus características, destacan las siguientes:

- Microprocesador ARM v8 64-bit, 4 núcleos, @1.5GHz
- 4GB RAM (hay variantes con otras cantidades de memoria RAM)
- Wifi 802.11ac
- Bluetooth 5.0
- Gigabit Ethernet
- 2 puertos USB 3.0 y 2 puertos USB 2.0
- GPIO 40 pines
- 2 puertos micro-HDMI, hasta 4K @60Hz
- Micro-SD

5.2.1.2 Gateway

Como se ha concluido en el capítulo anterior, para la transmisión de datos desde los dispositivos instalados en los terrenos de cultivo se utilizará la tecnología de comunicación LoRaWAN. Para hacer esto posible, es necesario emplear un Gateway LoRa, que no es más que una puerta de enlace que en este caso sirve de unión entre los dispositivos y la plataforma en la nube. Los nodos del sistema utilizan la tecnología inalámbrica de bajo consumo para conectarse al Gateway correspondiente, y este utiliza una red de alto ancho de banda, como Ethernet o Wi-Fi, para enviar finalmente los datos a la plataforma AWS. Dicho de otra forma, las puertas de enlace se pueden definir como enrutadores con un concentrador LoRa, permitiéndoles así recibir paquetes de dicho tipo, para luego dirigirlos al punto de destino.

Dentro de los Gateway LoRa, en cuanto al software, se pueden encontrar dos tipos:

- Puertas de enlace que solo ejecutan el software necesario para reenviar los paquetes. Esto hace que su coste sea más bajo y el firmware donde se ejecutan mínimo.
- Puertas de enlace que se ejecutan como un servicio sobre un sistema operativo y un ordenador convencional. Esto encarece su precio, pero le da libertad al propietario para poder instalar en el mismo dispositivo otros componentes software.

A mayores del sistema de funcionamiento interno, también existen ciertas características importantes en un Gateway LoRa:

- Protocolos utilizados para el reenvío de paquetes de datos. Hoy en día la mayoría de Gateways utilizan una versión del reenviador de paquetes UDP de Semtech, o el protocolo MQTT.
- Planes de frecuencia de envío utilizados. Dado que la tecnología LoRaWAN opera en el espectro radioeléctrico sin licencia, aparecen ciertas limitaciones específicas de cada país.
- Ciclos de trabajo. Se refiere a la porción de tiempo con la que se produce la transmisión de datos. Dependiendo del país donde se encuentre, este

- parámetro también puede estar regulado, por lo que es necesario informarse y comprobar si hay que cumplir ciertas medidas.
- Identificación del dispositivo. Las puertas de enlace LoRaWAN cuentan con 4 valores, formados por bits, para identificarse, de los cuales solo el DevAddr no es único del Gateway.
 - DevEUI: identificador de dispositivo final
 - DevAddr: dirección de dispositivo
 - AppEUI: identificador de aplicación
 - GatewayEUI: identificador de puerta de enlace

Otra decisión importante es respecto la propiedad de la puerta de enlace, pudiendo utilizar una pública o adquirir una propia. En el caso de emplear una red pública, una de las más conocidas es The Things Network. Esta es una iniciativa creada con el objetivo de crear una comunidad global IoT utilizando la tecnología LoRaWAN con gateways instalados en diferentes países. Esto permite a los usuarios utilizar una puerta de enlace ya instalada y que se encuentre en su zona de cobertura, además de poder acceder a información de otros usuarios que sea publicada en dicha pasarela.

En dicho trabajo se ha optado por la instalación de un Gateway propio debido a la importancia que se le dará a la seguridad y privacidad de la información y al control total sobre la solución por parte del cliente.



Figura 13. Esquema sobre el funcionamiento de un Gateway. Fuente: Elaboración propia.

Finalmente, el Gateway seleccionado y adquirido para el presente proyecto fue el modelo WisGate Edge Lite 2, de RAK Wireless. La elección se ha tomado en base a la relación calidad – precio y teniendo en cuenta las facilidades de integración con la plataforma AWS, ya que existe una lista de puertas de enlace completamente compatibles con el recurso dedicado a la tecnología LoRaWAN.

Como ya se ha mencionado antes, a la hora de adquirir la puerta de enlace es necesario tener claro en qué plan de frecuencia se va a trabajar. Esta información es proporcionada por la propia compañía LoRa Alliance y, en este caso concreto, ya que el país es España, el rango de frecuencia será 863-870 MHz.

Las características más relevantes de este producto son las siguientes:

- Compatibilidad completa con LoRaWAN Stack con Semtech SX1302.
- Ethernet Base-T de 100 M.
- Multi back-haul con Ethernet, Wi-Fi y Celular.
- Red LET Cat 4.
- Permite la integración tanto con servidores de red privados, como ChirpStack, como con públicos, que es el caso de TTN.
- Servidor de red integrado para una fácil implementación de aplicaciones e integración de puertas de enlace.



Figura 14. WisGate Edge Lite 2. Fuente: [42].

5.2.1.3 Temperatura y pH del suelo

Para la toma de la temperatura se va a utilizar el mismo dispositivo que para medir el pH del suelo, el Dragino LSPH01, por lo que cuenta con dos sondas impermeables. El propio dispositivo cuenta con una antena LoRa y una batería de Li-SOCl₂, proporcionándole una independencia total.

Respecto a los rangos y la resolución de la medición, para la acidez del suelo proporciona valores entre 3 pH y 10 pH, y con una resolución de 0.01 pH, mientras que para la temperatura variará entre -40°C y 85°C y una resolución de 0.1°C.



Figura 15. Dragino LSPH01. Fuente: [43].

5.2.1.4 Temperatura y humedad ambiente

Para medir las características ambientales también se utilizará un único sensor, el Milesight IoT EM500-CO₂, el cual está diseñado para medir 4 parámetros en total; la temperatura, humedad, CO₂ y presión barométrica del ambiente.

Entre los parámetros más relevantes, decir que cuenta con un rango de -30°C – 70°C y resolución de 0.1°C para la temperatura, mientras que respecto a la humedad abarca desde 0% hasta 100% y una resolución de 0.5% [44].

Tiene incorporada una antena LoRa y una batería de 19000 mAh, lo que le permitirá tener un funcionamiento interrumpido hasta 10 años en función de la frecuencia de transmisión de datos.



Figura 16. Milesight EM-500 CO2. Fuente: [45]

5.2.1.5 Iluminancia

A la hora de medir la iluminancia se van a utilizar dos dispositivos; Kit RAK Wireless WisBlock GPS y el sensor de iluminancia RAK Wireless WisBlock RAK1903, los cuales se van a juntar para poder formar un sistema propio. Esto se debe a la imposibilidad de obtener un sensor único ya centrado en medir este parámetro.



Figura 17. Kit WisBlock GPS. Fuente: [46].



Figura 18. RAK1903. Fuente: [47].

El contenido Kit WisBlock GPS es el siguiente:

- RAK5005-O: placa base donde se conectarán el resto de los dispositivos.
- RAK4631: módulo que permite al sistema emparejarse con receptores LoRa.
- RAK1904: sensor de aceleración.
- RAK1910: módulo GPS.
- Antena LoRa, BLE y GPS.
- RAKBox-B2: carcasa donde se van a montar todos los elementos necesarios. Tiene una placa solar que permite cargar una batería, aumentando así la autonomía del sistema.

La elección de este kit ha sido motivada por el aprovechamiento de la placa base RAK5005-O, el módulo RAK4631 y la antena LoRa. Por otro lado, se ha tenido en cuenta la carcasa con panel solar, ya que permite al sistema alimentarse a través de una batería que será recargada mediante una fuente renovable.

Respecto al sensor de luz ambiental, este tiene un rango de medición de entre 0,01 lux hasta 83865 lux.

5.2.2 Software

5.2.2.1 Raspberry Pi OS

Como cualquier otro PC, Raspberry necesita un sistema operativo para funcionar. Una gran cualidad es la versatilidad de su software, lo que le permite ser utilizada con múltiples fines, desde miniordenador, como es el caso de este proyecto, hasta una plataforma de videojuegos.

Esto es posible gracias a que su sistema operativo, Raspberry Pi OS, es una variante de Debian y, como tal, una distribución del sistema operativo de código libre Linux [48]. Este será instalado y se ejecutará en una tarjeta microSD insertada en la placa.

5.2.2.2 WisGateOS

En el caso de la puerta de enlace utilizada para los dispositivos LoRa, el sistema operativo que hace realidad su funcionamiento es WisGateOS [49], también basado en Linux, en concreto, en el proyecto OpenWrt. Su función es ser el software que interactúa con el dispositivo físico concreto a través de su firmware, creando una capa de abstracción común para el acceso a los dispositivos concretos.

Para configurar la puerta de enlace está disponible el software WisDM, que se ejecuta sobre WisGateOs y ofrece una interfaz gráfica de usuario amigable para los administradores de la instalación.

5.2.2.3 Milesight ToolBox

El dispositivo que mide los parámetros ambientales utiliza ToolBox Software para su configuración, o Milesight ToolBox en el caso de emplear la opción mediante NFC. Esto es una aplicación proporcionada por la propia compañía que permite configurar el dispositivo desde lo básico, como la recopilación de datos, calibración, intervalos de tiempo de lectura; hasta todos los parámetros necesarios para utilizar la tecnología LoRaWAN. A mayores, también permite hacer un seguimiento y mantenimiento del dispositivo, permitiendo al usuario descargar y actualizar el firmware si es necesario.



Figura 19. Software Milesight ToolBox. Fuente: Elaboración propia.

5.2.3 Montaje

5.2.3.1 WisGate Edge Lite 2

Para realizar el montaje correctamente se siguen las instrucciones proporcionadas por el fabricante [50] [51].

Paso 1. En un primer lugar es necesario conectar la antena LoRa al conector RP-SMA que se encuentra en la parte posterior del Gateway. Para encender la puerta de enlace se utiliza el adaptador CC de 12 V proporcionado por el fabricante.

Paso 2. Una vez listas las configuraciones previas explicadas en el manual, se procede a realizar las configuraciones para utilizar el dispositivo como puerta de enlace LoRa. Para ello se accede a la interfaz gráfica mediante su dirección IP. Primero es necesario definir correctamente la banda de frecuencia regional que se va a utilizar. Para esto se accede a Channel Plan y en el apartado Region se selecciona EU863-870.

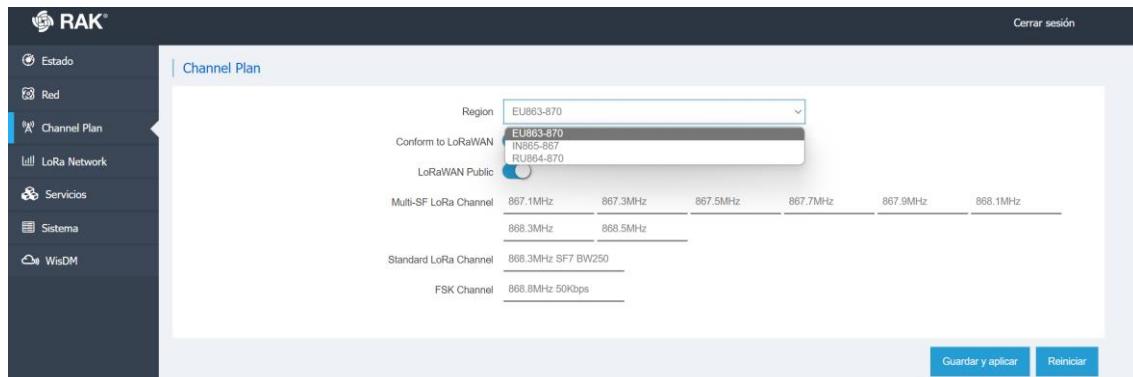


Figura 20. Configuración de la banda de frecuencia del Gateway. Fuente: Elaboración propia.

Ahora se configurarán los ajustes de la red LoRa accediendo a LoRa Network. En la parte de arriba se muestra el EUI, que es un identificador único para la puerta de enlace. En la pestaña desplegable Mode se elegirá el modo de funcionamiento del Gateway, en este caso Basic Station con servidor LNS y autenticación de cliente y servidor TLS.

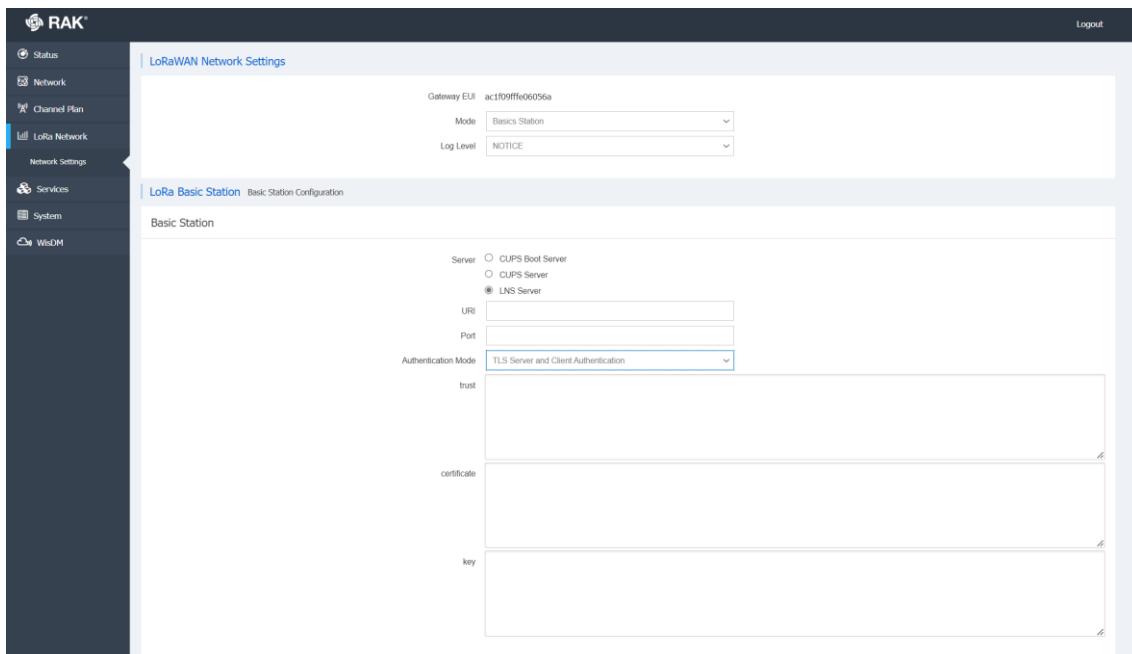


Figura 21. Configuración LoRa Basic Station del Gateway. Fuente: Elaboración propia.

El resto de los parámetros a cubrir dependen de las credenciales que proporcione la plataforma AWS una vez se registre el Gateway en la plataforma.

5.2.3.2 Milesight EM500-CO2

Para realizar el montaje correctamente se siguen las instrucciones proporcionadas por el fabricante.

Paso 1. Antes de iniciar la configuración es necesario asegurarse de que la batería del dispositivo está colocada correctamente y la lámina aislante que lleva en su interior ha sido retirada. Para ello se desatornillará la tapa inferior y se harán las comprobaciones necesarias, quitando la lámina aislante.

Para encender el dispositivo se mantiene pulsado durante más de 3 segundos el botón que hay en la placa, activándose un led de color verde.



Figura 22. Botón de encendido de Milesight EM500-CO2. Fuente: Elaboración propia.

Paso 2. La configuración del dispositivo se puede hacer mediante NFC, a través de un teléfono inteligente, o conectándolo por USB a un ordenador, en este caso se ha optado por la primera opción. Para ello es necesario descargar e instalar la aplicación Milesight ToolBox en nuestro móvil, la cual permite realizar configuraciones en el

dispositivo y actualizaciones de software, aparte de mostrar por pantalla información relevante sobre el mismo.



Figura 23. Interfaz gráfica de Milesight EM500-CO2. Fuente: Elaboración propia.

De manera predeterminada el dispositivo viene configurado con un intervalo de informes de 10 minutos, esto se ha cambiado a 120 minutos, aunque es de libre elección. A mayores, en *Data Collection Settings*, se configuran los parámetros que se quieren monitorear, en este caso simplemente temperatura y humedad, permitiendo así que el sensor omita los 2 restantes.

Respecto a la configuración de frecuencia de LoRaWAN, se accede a LoRaWAN Settings para seleccionar la frecuencia con la que se va a trabajar y los canales mediante los cuales se enviarán los enlaces.



Figura 24. Configuración de la banda de frecuencia de Milesight EM500-CO2. Fuente: Elaboración propia.

5.2.3.3 Dragino LSPH01

Para realizar el montaje correctamente se siguen las instrucciones proporcionadas por el fabricante [52].

Paso 1. El primer paso es conectar la antena LoRa en el conector que se encuentra en la parte superior del dispositivo.

Paso 2. Ahora se realiza la configuración del dispositivo. Respecto a la configuración LoRaWAN ya viene de manera predeterminada en modo LoRaWAN OTTA clase A, utilizando la región EU868, por lo que no es necesario realizar ningún cambio en el mismo.

La única variación es respecto al intervalo de transmisión de datos. Este viene predefinido cada 20 minutos, por lo que es necesario cambiarlo a 2 horas. Para ello se conecta el dispositivo al ordenador mediante un conversor USB a TTL y con la ayuda de cables de puente.



Figura 25. Conexión del conversor USB-TTL al dispositivo LSPH01. Fuente: Elaboración propia.

En este paso hay que prestar especial atención a las conexiones, donde RXD y TXD son los conectores correspondientes a recepción y transmisión de datos, respectivamente:

- GND LSPH01 --> USB TTL GND
- UART_TXD LSPH01 --> USB TTL RXD
- UART_RXD LSPH01 --> USB TTL TXD

Se conecta el USB al puerto del ordenador y mediante una herramienta software, en este caso Tera Term, se le proporcionará el siguiente comando AT, expresado en milisegundos:

AT+TDC=7200000

Figura 26. Comando AT para cambiar el intervalo de transmisión en LSPH01. Fuente: Elaboración propia.

Paso 3. A continuación se procede al encendido del dispositivo. Para ello se coloca el jumper amarillo de manera que cree un puente entre los dos pines, y asegurándose de que el interruptor SW1 se encuentra en modo Flash.



Figura 27. Encendido de Dragino LSPH01. Fuente: Elaboración propia.

Paso 4. Antes de realizar las primeras mediciones, es necesario colocar la sonda en el terreno durante más de 24 horas, ya que los primeros valores pueden ser erróneos.



Figura 28. Colocación de la sonda de Dragino LSPH01. Fuente: Elaboración propia.

5.2.3.4 RAK Wireless Kit WisBlock + RAK1903

Para realizar el montaje correctamente se siguen las instrucciones proporcionadas por el fabricante [53][54][55][56].

Paso 1. En un primer lugar se instala la placa con dos tornillos en la base de la carcasa, lo cual permite agregar el resto de los componentes encima sujetos a la misma.



Figura 29. Montaje de la placa en Box-B2. Fuente: Elaboración propia.

Paso 2. A continuación, se coloca el módulo WisBlock. Este se encuentra formado por el WisBlock Base RAK5005-O, que es la placa base que conecta todos los módulos necesarios entre sí y crea la fuente de alimentación al sistema, y WisBlock Core RAK4630 que es el centro de procesamiento de datos del sistema y contiene protocolos LoRaWAN, lo que permite la comunicación mediante dicha tecnología.

Antes de fijarlo a la placa, hay que añadir el sensor de luz RAK1903. Para esto se hacen coincidir los conectores del sensor con los que se encuentran en la placa base sección A, y luego se ejerce una ligera fuerza hasta que se escuche un sonido que confirma que se ha unido correctamente [57][58]. Gracias a un tornillo se fija el sensor a la base y el conjunto se agrega a la caja fijándolo nuevamente con unos tornillos.

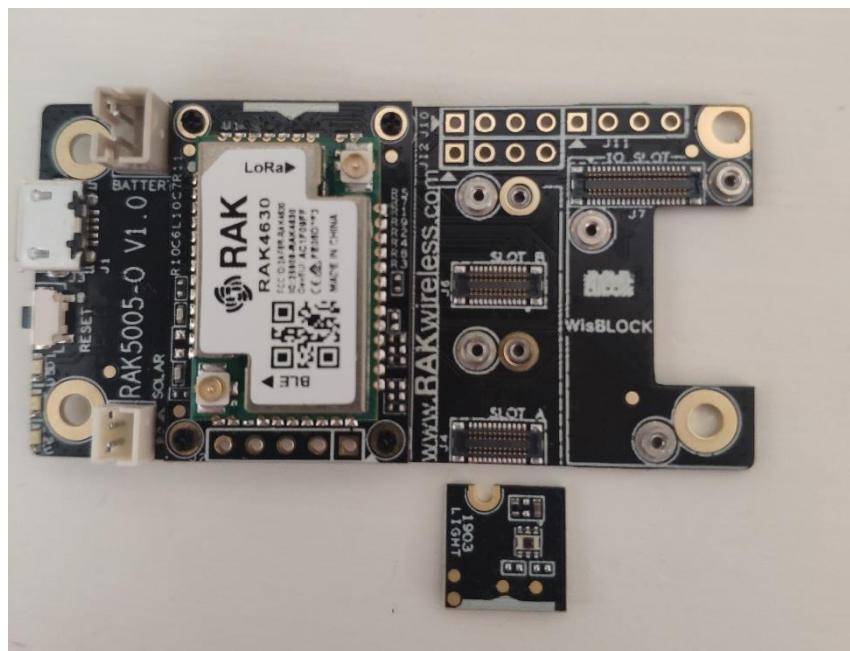


Figura 30. Conectores A de la placa base RAK5005-O. Fuente: Elaboración propia.

Paso 3. A continuación, se conectan las antenas LoRa y BLE al módulo RAK4631. Hay que prestar atención a la etiqueta del dispositivo, ya que muestra en qué conector hay que colocar cada una de ellas.

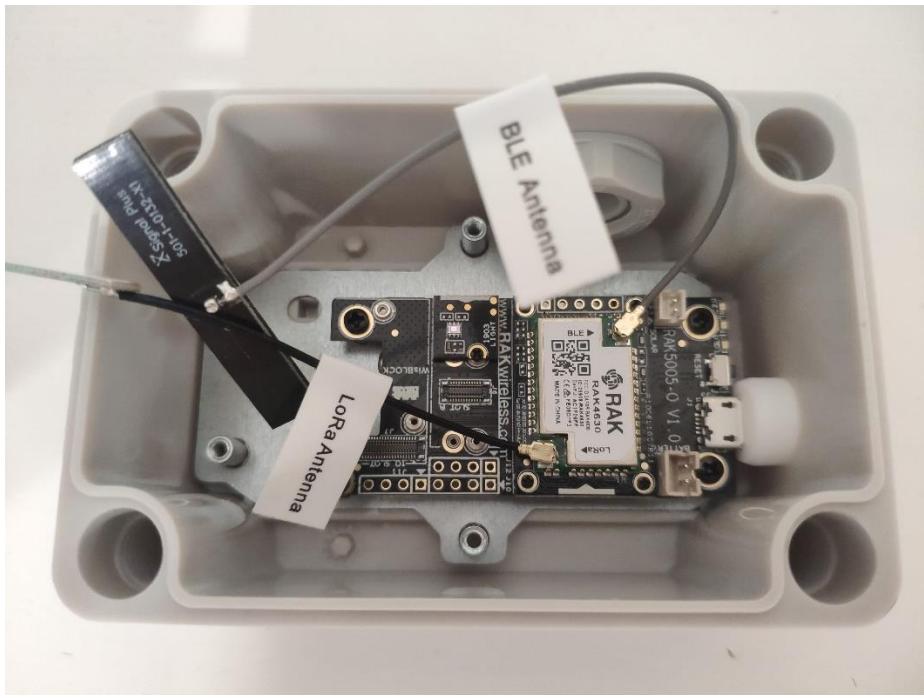


Figura 31. Conexión de las antenas LoRa y BLE al módulo RAK4631. Fuente: Elaboración propia.

Paso 4. Por último, se procede a conectar el cable de la batería y panel solar en los conectores PCB correspondientes de WisBlock Base, asegurándose de qué pines corresponden al positivo y cuáles al negativo. En este paso hay que prestar especial atención ya que no todas las baterías vienen con la misma distribución.



Figura 32. Conexión de la batería y placa solar a RAK5005-O. Fuente: Elaboración propia.

Paso 5. Finalmente, se coloca el sistema montado en una carcasa transparente. Esto se debe a que la carcasa incluida en el kit es completamente opaca y estanca, impidiendo así el correcto flujo de lux y provocando que el sensor devuelva valores muy pequeños o casi nulos.

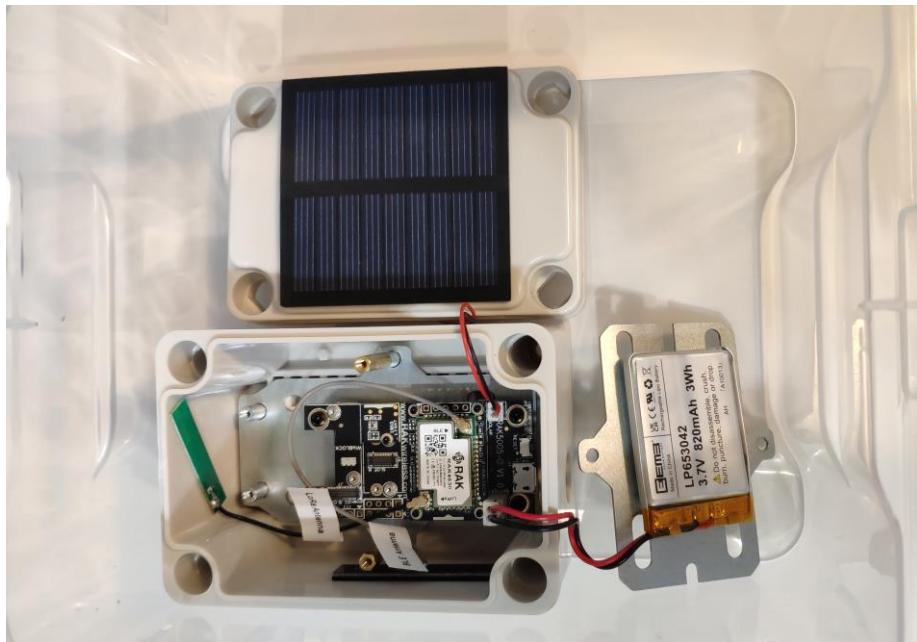


Figura 33. Colocación final del sistema RAK1903. Fuente: Elaboración propia.

Paso 6. Para configurar la placa WisBlock RAK4631 y el sensor RAK1903 es necesario conectarlo al PC y seguir el procedimiento explicado en el manual. Para ello, en la placa RAK5005-O hay un puerto USB al que se puede acceder sin necesidad de sacar los dispositivos fuera de la carcasa. De esta manera el software Arduino IDE [59] reconoce la placa y un led rojo indica que la conexión ha sido realizada con éxito. Antes de realizar las configuraciones referentes a la comunicación LoRa se hacen una serie de configuraciones previas [60][61][62], como la actualización del firmware para soportar el uso de la API RUI3 o la incorporación de bibliotecas, explicadas en el manual.

Paso 7. Para enviar los valores de lux mediante LoRaWAN hay que instalar la biblioteca llamada SX126x-Arduino.

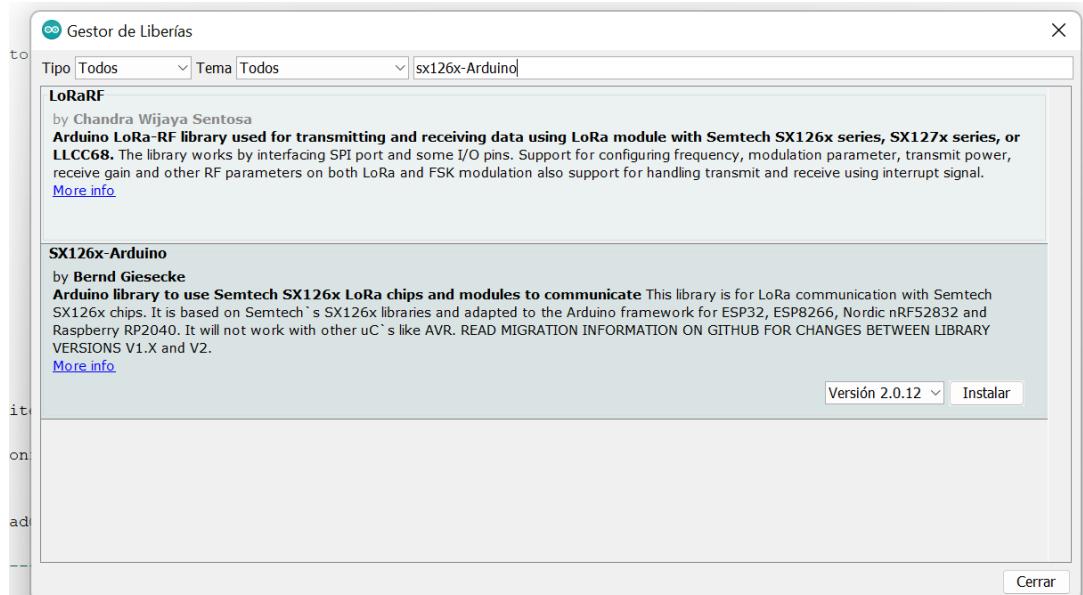


Figura 34. Gestor de librerías en Arduino IDE. Fuente: Elaboración propia.

Como la placa RAK4631-R, que es la que se comunica mediante LoRaWAN con la puerta de enlace, funciona como un microcontrolador, es necesario grabar en la misma el código de arranque correspondiente para que se conecte al Gateway, reciba los datos

procedentes del sensor y posteriormente los envíe a la plataforma AWS. En el monitor serie de Arduino se puede comprobar si efectivamente se envían paquetes a la puerta de enlace y el valor de estos, siempre y cuando se tenga el dispositivo conectado al ordenador.

```
09:47:54.648 -> El valor lux es: 0.72
09:47:54.648 -> Lux 0.72
09:47:54.648 -> Data Packet:
09:47:54.692 -> 0x01 0x65 0x00 0x00
09:47:54.692 -> Sending is requested
09:47:55.799 -> Successfully sent
09:47:55.846 -> +EVT:SEND_CONFIRMED_OK
09:52:55.816 -> El valor lux es: 0.48
09:52:55.816 -> Lux 0.48
09:52:55.816 -> Data Packet:
09:52:55.863 -> 0x01 0x65 0x00 0x00
09:52:55.910 -> Sending is requested
09:52:56.994 -> Successfully sent
09:52:56.994 -> +EVT:SEND_CONFIRMED_OK
09:57:56.986 -> El valor lux es: 0.48
09:57:56.986 -> Lux 0.48
09:57:56.986 -> Data Packet:
09:57:57.033 -> 0x01 0x65 0x00 0x00
09:57:57.080 -> Sending is requested
09:57:58.152 -> Successfully sent
09:57:58.152 -> +EVT:SEND_CONFIRMED_OK
10:02:58.179 -> El valor lux es: 0.48
10:02:58.179 -> Lux 0.48
```

Figura 35. Comprobación del código de arranque de RAK4631. Fuente: Elaboración propia.

5.3 Plataforma IoT global

En base a todos los estudios y análisis previos que se han realizado sobre el proyecto, se ha decidido utilizar la plataforma IoT de AWS para la gestión e integración de todos los datos proporcionados por los diferentes nodos que conforman el sistema.

De esta manera, y como ya se ha mencionado anteriormente, el nodo de sensorización basado en Raspberry Pi se comunicará con la plataforma mediante el protocolo MQTT, mientras que los dispositivos restantes utilizarán la tecnología LoRaWAN para comunicarse con la puerta de enlace, y esta mediante el protocolo de comunicación HTTPs con AWS.

En este capítulo se explicarán de manera detallada todas las configuraciones y recursos utilizados a la hora de implementar el sistema en la plataforma AWS [63], así como los resultados obtenidos en base a la realización de diversas pruebas.

En este proyecto concreto se ha utilizado el servicio *IoT Core* de AWS como recurso principal del sistema, en el cual se definirán los dispositivos virtuales que se corresponden con los dispositivos reales basados en Raspberry Pi, y mediante *IoT Core for LoRaWAN* los nodos de sensorización que utilizan dicha tecnología y la puerta de enlace correspondiente. Además de los anteriores, hay otros servicios proporcionados por AWS que son de interés para el proyecto:

- Cliente de prueba de MQTT, para la organización de los mensajes enviados por dispositivos en *topics* y su visualización y monitorización desde la consola.
- Dynamo DB, para el almacenamiento persistente de los datos y la consulta de históricos.
- Amazon SES, para la notificación de alertas mediante una dirección de correo electrónico.

- AWS Lambda, para la programación y ejecución de procesos genéricos en la nube, que en este caso nos permite programar funciones para el procesamiento de los datos recibidos en la plataforma y la toma de decisiones.

5.3.1 Envío y recepción de mensajes de sensorización con MQTT

Para validar la correcta organización de los datos enviados en temas o *topics* según su contenido, siguiendo la filosofía del patrón publicación/subscripción, se empleará el cliente de prueba de MQTT, que permite visualizar los mensajes enviados según al tema que se hayan dirigido en origen.

En este proyecto concreto se han definido tres temas principales, *Datos_Raspberry*, *Parámetros_Ambientales* y *Parámetros_Terreno*. De esta manera, los valores relacionados con el nivel de fertilizantes irán dirigidos al primero, los que hacen referencia a condiciones del ambiente (temperatura, humedad e iluminancia) al segundo y, siguiendo la misma lógica, se encuentran las propiedades del terreno (pH y temperatura), que serán enviadas al último tema. A mayores se encuentran dos temas adicionales que permiten la visualización de los mensajes decodificados enviados por los nodos de sensorización de tipo LoRa: *Parámetros_Ambientales_Decoder* y *Parámetros_Terreno_Decoder*.

El esquema que siguen los mensajes es diferente según el tipo de nodo del que procedan, y los pasos seguidos para concluir la comunicación también, por lo que a continuación se explicarán brevemente ambos casos.

5.3.1.1 Publicación y visualización de mensajes enviados por Raspberry Pi

Para conectar la Raspberry Pi a AWS se utilizará el recurso IoT Core, donde se definirá el dispositivo.

Una vez configurada y definida la Raspberry, es momento de gestionar los mensajes enviados, que serán publicados en el tema *Datos_Raspberry*. Para ello es necesario establecer una conexión con el broker MQTT de la plataforma, lo cual se realizará mediante código como se muestra en la siguiente figura. En este fragmento de código se aprecia cómo es necesario especificar el punto de enlace proporcionado por AWS, *endpoint*, que será el utilizado por el dispositivo IoT para conectarse. Por otro lado, se especificará la ruta donde se encuentran los archivos correspondientes al certificado del dispositivo, la clave privada y el certificado de entidad.

```
1 # Conexión MQTT
2 def setup_mqtt_connection():
3     endpoint = "xxxxxxxxxxxx.iot.eu-west-1.amazonaws.com"
4     client_id = socket.gethostname()
5     # An event-loop is a thread for doing async work, such as I/O
6     event_loop_group = io.EventLoopGroup(num_threads=1)
7     # DNS host resolver to use
8     host_resolver = io.DefaultHostResolver(event_loop_group)
9     # Handles creation and setup of client socket connections
10    client_bootstrap = io.ClientBootstrap(event_loop_group, host_resolver)
11    mqtt_connection = mqtt_connection_builder.mtls_from_path(
12        endpoint=endpoint,
13        client_id=client_id,
14        cert_filepath="/home/pi/TFG/certificate.pem.crt",
15        pri_key_filepath="/home/pi/TFG/private.pem.key",
16        ca_filepath="/home/pi/TFG/AmazonRootCA1.pem",
17        client_bootstrap=client_bootstrap,
18    )
19    print(f"Conectando a {endpoint} con id {client_id}...")
20    mqtt_connection.connect().result()
21    print("Conectado")
22    return mqtt_connection
```

Figura 36. Fragmento de código conexión broker MQTT. Fuente: Elaboración propia.

Una vez establecida la comunicación se enviará el mensaje al tema seleccionado. En la siguiente ilustración se muestra un ejemplo de envío de datos a la plataforma, generando un valor aleatorio correspondiente al nivel de fertilizante disponible.

```
1 # Conseguir información del nivel de fertilizante disponible
2 def get_info():
3     # Se adquiere un valor aleatorio entre 5000 y 25000 L
4     nivel = str(randrange(5000, 25000))
5     return nivel
6
7 # Publicar la información en un topic de AWS
8 def publish(mqtt_connection):
9     nivel = get_info()
10    # Transformación del mensaje a JSON para poder enviar al topic correspondiente
11    mensaje = json.dumps(
12        {
13            "Nivel_fertilizante": nivel,
14        }
15    )
16    # Publicación en el topic "Datos_Raspberry"
17    mqtt_connection.publish(topic="Datos_Raspberry",
18                            payload=mensaje, qos= mqtt.QoS.AT_LEAST_ONCE)
19    print(f"El nivel de fertilizante es {nivel} L")
```

Figura 37. Fragmento de código envío de mensajes. Fuente: Elaboración propia.

Una vez concluida exitosamente la comunicación se pueden visualizar los datos enviados al tema *Datos_Raspberry* mediante el cliente de prueba de MQTT. Para ello es necesario crear una suscripción al tema previamente.

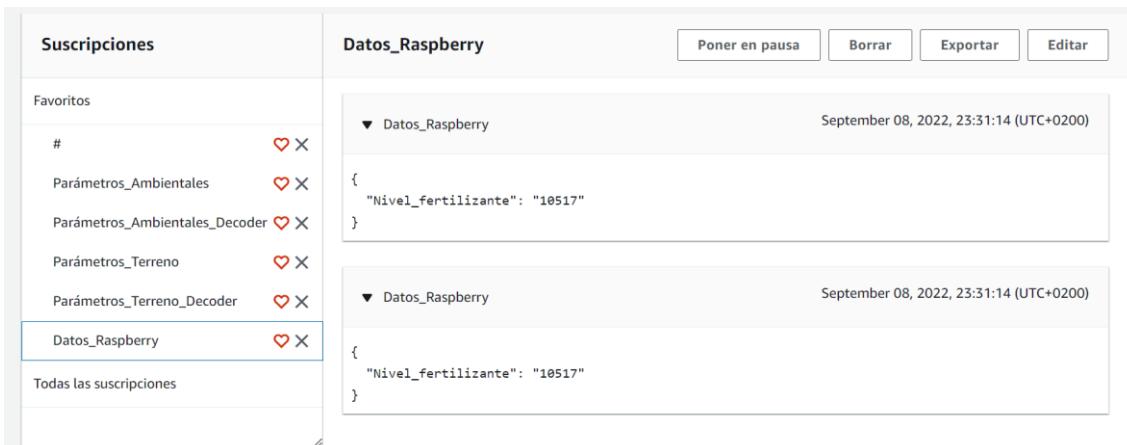


Figura 38. Visualización de mensajes en tema "Datos_Raspberry". Fuente: Elaboración propia.

5.3.1.2 Publicación y visualización de mensajes enviados mediante dispositivos LoRaWAN

A diferencia del caso anterior, los dispositivos dotados de la tecnología de comunicación LoRaWAN se integrarán en el sistema mediante el recurso de AWS denominado *IoT Core for LoRaWAN*, por lo que la lógica seguida es distinta.

En este caso se encuentran tres nodos de sensorización, EM500-CO2, LSPH01 y RAK1903, y una puerta de enlace, RAK7268. Las configuraciones realizadas para una conexión con el *network server* se ilustran en el siguiente diagrama [64][65][66].

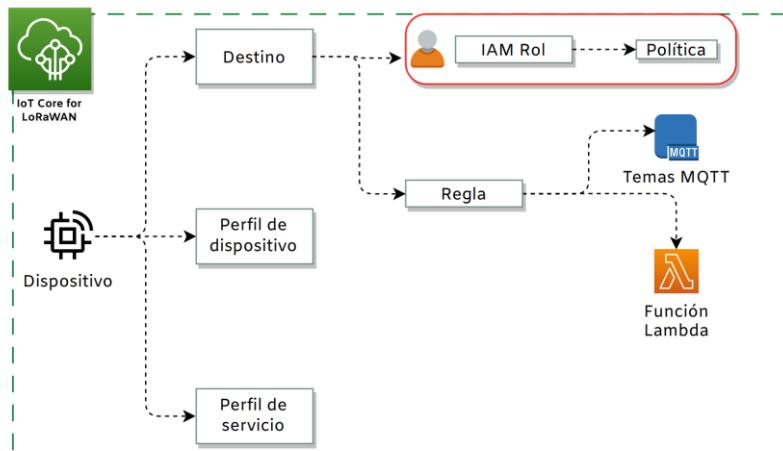


Figura 39. Arquitectura IoT Core for LoRaWAN. Fuente: Elaboración propia.

Para realizar la publicación de los valores en los diferentes temas MQTT es necesario crear una regla por dispositivo, que se asociará posteriormente al destino de este. Dicha regla permite a los dispositivos interactuar con el resto de los recursos de AWS y será la encargada de redirigir los datos recibidos desde los nodos de sensorización, con independencia de su ubicación o número.

Esta regla sigue una configuración similar en los tres tipos de dispositivos, diferenciándose en las acciones que se deben producir, las cuales hacen referencia a las tareas que debe ejecutar la regla cuando el dispositivo envíe un mensaje. Para los dispositivos se diferencian simplemente en los nombres de los recursos que invocan, es decir, para todos los nodos se realizará la publicación de mensajes en un tema concreto y la llamada a una función Lambda concreta [67][68], que será la encargada de decodificar la carga útil enviada y publicarla en un tema destinado a la visualización de mensajes decodificados.

En el caso concreto del sensor EM500-CO2 la regla *DecoderEM500* enviará los mensajes al tema *Parámetros_Ambientales* y la función Lambda invocada será *EM500CO2Decoder*.

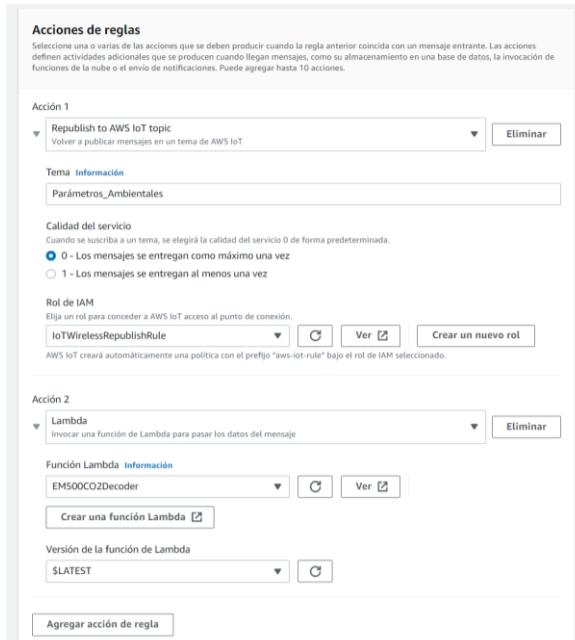


Figura 40. Acciones de la regla DecoderEM500. Fuente: Elaboración propia.

Para poder visualizar los mensajes enviados por los dispositivos se accede al cliente de prueba de MQTT. En ellos se puede apreciar como el esquema que sigue cada dispositivo es distinto.

The screenshot shows a MQTT visualization interface. On the left, there's a sidebar titled 'Suscripciones' (Subscriptions) with a 'Favoritos' (Favorites) section containing several entries like '#', 'Parámetros_Terreno', 'Parámetros_Ambientales', 'Datos_Raspberry', 'Parámetros_Terreno_Decoder', and 'Parámetros_Ambientales_Decoder'. The 'Parámetros_Ambientales_Decoder' entry is highlighted with a blue border. On the right, the main panel is titled 'Parámetros_Ambientales_Decoder' and shows a message from 'Parámetros_Ambientales_Decoder' at 'September 10, 2022, 20:42:34 (UTC+0200)'. The message content is a JSON object: { "temperature": 25.2, "humidity": 67.5 }.

Figura 41. Visualización de mensajes en tema "Parámetros_Ambientales_Decoder".
Fuente: Elaboración propia.

This screenshot shows the same MQTT visualization interface. The 'Suscripciones' sidebar now lists 'Parámetros_Terreno_Decoder' as a favorite. The main panel displays a message from 'Parámetros_Terreno_Decoder' at 'September 09, 2022, 00:57:58 (UTC+0200)'. The message content is a JSON object: { "Bat": 3.35, "TempC_DS18B20": "0.00", "phSoil": "4.50", "temperatureSoil": "24.50", "Interrupt_flag": 0, "Message_type": 1 }.

Figura 42. Visualización de mensajes en tema "Parámetros_Terreno_Decoder". Fuente: Elaboración propia.

This screenshot shows the MQTT visualization interface again. The 'Suscripciones' sidebar has 'Parámetros_Ambientales_Decoder' selected. The main panel shows a message from 'Parámetros_Ambientales_Decoder' at 'September 09, 2022, 00:59:18 (UTC+0200)'. The message content is a JSON object: { "devEui": "ac1f09ffffe06d3f3", "fPort": 2, "freq": "867700000", "timestamp": "2022-09-08T22:59:18Z", "illuminance": 9 }. Below this, another message from 'Parámetros_Ambientales_Decoder' at 'September 09, 2022, 00:54:17 (UTC+0200)' is partially visible.

Figura 43. Visualización de mensajes en tema "Parámetros_Ambientales_Decoder".
Fuente: Elaboración propia.

5.4 Registro de datos en la nube

Una vez medidos y enviados los datos ambientales, del terreno y medición de fertilizante a la plataforma AWS, es necesario tratarlos para que resulten útiles a la hora de realizar una evaluación de las condiciones de la zona y agilizar la toma de decisiones. Para ello se ha optado, en primer lugar, por su almacenamiento en una base de datos

de históricos, para poder realizar un seguimiento de manera rápida y flexible y detectar las variaciones más relevantes.

El recurso ofrecido por AWS utilizado para ello es Dynamo DB, que es una base de datos NoSQL que permite almacenar y ordenar una gran colección de datos de la manera que le indique el usuario [69]. Para entender mejor su funcionamiento y los pasos a realizar es preciso tener claros los componentes que conforman este recurso y su funcionamiento.

De manera simple, DynamoDB trabaja la elaboración de tablas, elementos y atributos, donde cada componente engloba al siguiente. Esto quiere decir que la tabla no es más que una colección de datos, y los elementos que hay en la tabla son las diferentes características de los datos que se están almacenando. En último nivel se encuentran los atributos, que hacen referencia a las características fundamentales de los datos.

5.4.1 Implementación y pruebas

A continuación, se muestran los pasos seguidos para comenzar a utilizar este recurso en el proyecto. En primer lugar, es necesario acceder a la consola de DynamoDB y seleccionar crear tabla.

En las propiedades de dicha tabla se cubrirá el nombre y la clave de partición, que en este caso se ha optado por utilizar la fecha y hora a la que se han generado los datos.

The screenshot shows the 'Create table' wizard in the AWS DynamoDB console. The top navigation bar shows 'DynamoDB > Tablas > Crear tabla'. The main title is 'Crear tabla'. Below it, a section titled 'Detalles de la tabla' has a 'Información' link. It states: 'DynamoDB es una base de datos sin esquemas que solo requiere un nombre de tabla y una clave principal al crear la tabla.' Under 'Nombre de la tabla', there is a text input field containing 'monitorizacion_tfg'. A note below says: 'Entre 3 y 255 caracteres. Solo se pueden usar letras, números, guiones bajos (-), guiones (-) y puntos (.).' Below this, there is a section for 'Clave de partición' with two dropdown menus: 'Time' and 'Número'. A note says: 'La clave de partición forma parte de la clave principal de la tabla. Se trata de un valor hash que se utiliza para recuperar elementos de la tabla, así como para asignar datos entre hosts por cuestiones de escalabilidad y disponibilidad.' Under 'Clave de ordenación - opcional', there is another dropdown menu 'Cadena' with a note: 'Puede utilizar una clave de ordenación como segunda parte de la clave principal de una tabla. La clave de ordenación le permite ordenar o buscar entre todos los elementos que comparten la misma clave de partición.' A note at the bottom says: 'Escribir el nombre de la clave de ordenación' and 'De 1 a 255 caracteres, distingue entre mayúsculas y minúsculas.'

Figura 44. Creación de una tabla DynamoDB. Fuente: Elaboración propia.

Una vez creada la tabla de datos se creará una regla que permita enviar los datos de los sensores recibidos a través de IoT Core para su almacenamiento en dicha base de datos. En este caso se hacen tres reglas para redirigir los valores enviados previamente a cada topic, *Dynamo_Rule_Ambiente*, *Dynamo_Rule_Terreno* y *Dynamo_Rule_Raspberry*. Ambas comparten la misma configuración, diferenciándose en la instrucción SQL, que especificará que datos se desean almacenar en la tabla y su procedencia.

De esta manera en el caso de *Dynamo_Rule_Ambiente*, la regla está orientada a los datos *temperature*, *humidity* y *illuminance*, procedentes del tema *Parámetros_Ambientales_Decoder*, por lo que la sintaxis SQL es la siguiente:

Configurar la instrucción SQL [Información](#)
Agregue una sintaxis SQL simplificada para filtrar los mensajes recibidos en un tema MQTT y enviar los datos a otro lugar.

Instrucción SQL

Versión de SQL
La versión del motor de reglas SQL que se utilizará al evaluar la regla.
2016-03-23 ▾

Instrucción SQL
Ingrese una instrucción SQL con lo siguiente: SELECT <Attribute> FROM <Topic Filter> WHERE <Condition>. Por ejemplo: SELECT temperature FROM 'iot/topic' WHERE temperature > 50. Para obtener más información, consultar Referencia de SQL de AWS IoT.

```
1 SELECT temperature, humidity, illuminance FROM "Parámetros_Ambientales_Decoder"
```

SQL Línea 1, columna 79

C Cancelar Anterior Siguiente

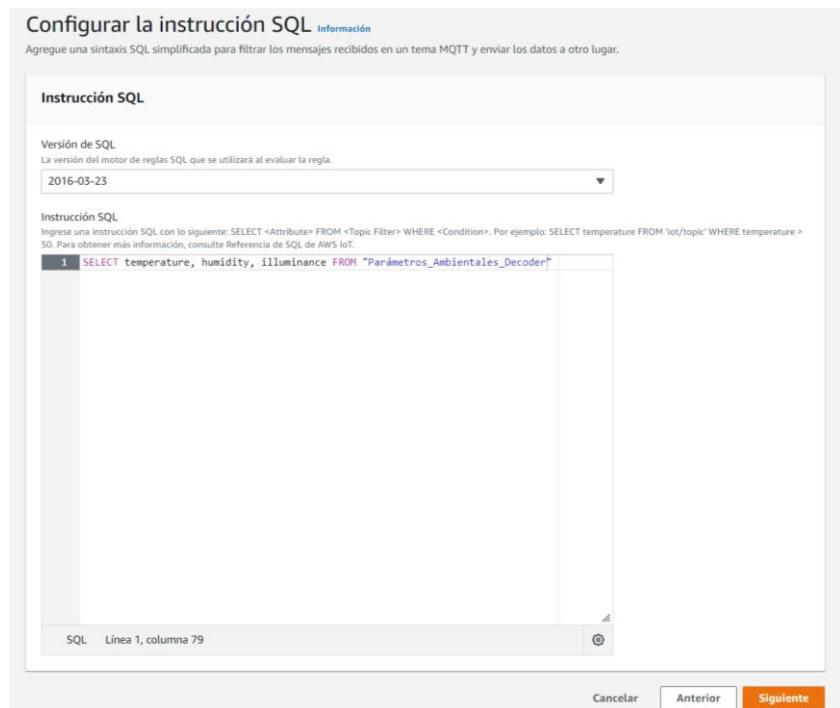


Figura 45. Ejemplo de instrucción SQL para la regla Dynamo_Rule_Ambiente. Fuente: Elaboración propia.

La acción que va a ejecutar la regla es la de enviar los datos a la tabla *monitorización_tfg*, previamente creada en DynamoDB, y para ello es necesario proporcionar la clave de partición que va a seguir la misma.

Acciones de reglas
Seleccione una o varias de las acciones que se deben producir cuando la regla anterior coincide con un mensaje entrante. Las acciones definen actividades adicionales que se producen cuando llegan mensajes, como su almacenamiento en una base de datos, la invocación de funciones de la nube o el envío de notificaciones. Puede agregar hasta 10 acciones.

Acción 1

DynamoDB
Insertar un mensaje en una tabla de DynamoDB ▾ Eliminar

Nombre de la tabla [Información](#)
monitorizacion_tfg ▾ Ver Crear una tabla de DynamoDB

Clave de partición
La clave de partición (también denominada clave hash) debe coincidir con la clave de partición de la tabla de DynamoDB que ha creado.
Time

Tipo de clave de partición
El tipo de clave de partición (también denominada clave hash) puede ser STRING o NUMBER. El valor predeterminado es STRING.
NUMBER ▾

Valor de clave de partición
El valor de clave de partición (también denominada clave hash) admite plantillas de sustitución que proporcionan datos en tiempo de ejecución.
\${timestamp()}

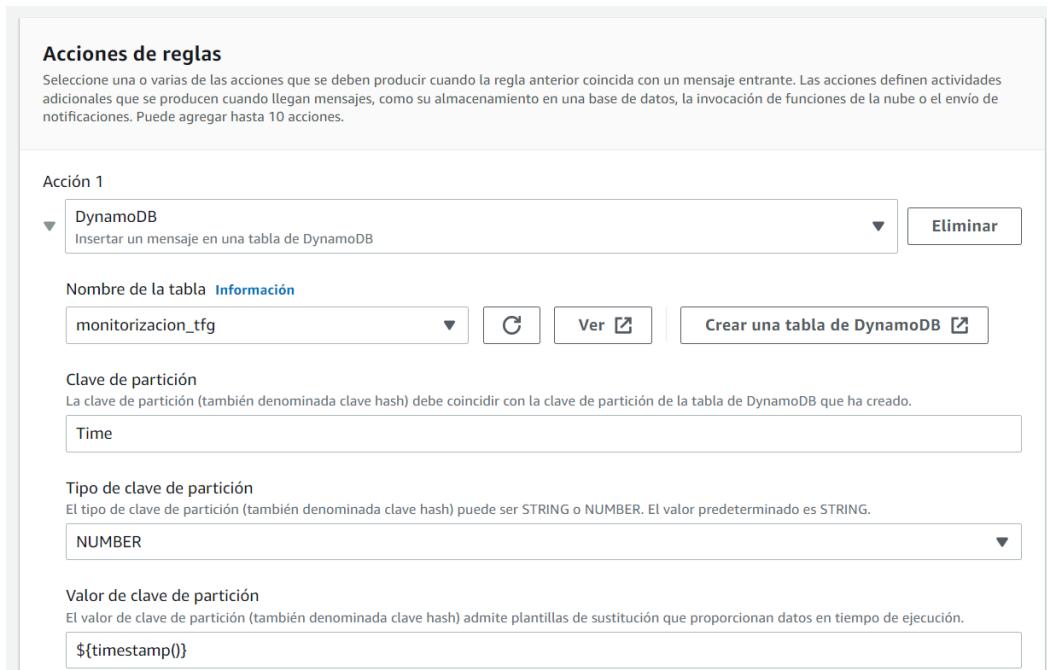


Figura 46. Acciones de la regla Dynamo_Rule_Ambiente. Fuente: Elaboración propia.

Es necesario crear y seleccionar un rol de IAM al que se le conceden automáticamente permisos para realizar la acción seleccionada. Este rol *DynamoRole* será común para las tres reglas.

Realizando las configuraciones adaptadas en las dos reglas restantes, los valores devueltos y almacenados en la tabla quedan de la siguiente manera:

Elementos devueltos (955)		Acciones ▾	Crear elemento
□	Time	payload	▼
□	1662709079212	{"phSoil": {"S": "4.46"}, "temperatureSoil": {"S": "21.90"} }	
□	1662709277649	{"illuminance": {"N": "9"} }	
□	1662709578871	{"illuminance": {"N": "9"} }	
□	1662709880011	{"illuminance": {"N": "9"} }	
□	1662709886559	{"phSoil": {"S": "4.57"}, "temperatureSoil": {"S": "24.40"} }	
□	1662710083017	{"temperature": {"N": "25.5"}, "humidity": {"N": "66.5"} }	
□	1662710184150	{"illuminance": {"N": "9"} }	
□	1662710279629	{"phSoil": {"S": "4.46"}, "temperatureSoil": {"S": "21.80"} }	
□	1662710347848	{"temperature": {"N": "23.8"}, "humidity": {"N": "66.5"} }	
□	1662710427451	{"temperature": {"N": "23.8"}, "humidity": {"N": "66"} }	
□	1662710485437	{"illuminance": {"N": "9"} }	
□	1662710516374	{"Nivel_fertilizante": {"S": "20958"} }	

Figura 47. Tabla de datos DynamoDB. Fuente: Elaboración propia.

5.5 Notificación de alertas

Una vez se hayan alcanzado los objetivos de visualizar y almacenar los datos, es momento de analizarlos para enviar una serie de notificaciones si alguno no se encuentra en un rango de valores apropiados. Para ejecutar dichos avisos se empleará el recurso SES de AWS, que es un servicio que permite enviar y recibir mensajes de correo electrónico utilizando un servidor propio de AWS [70].

Su funcionamiento se basa en que una aplicación cliente solicita al recurso SES permiso para enviar un correo electrónico, y esta acepta o no la solicitud. Cuando se envía la notificación esta llega al destinatario a través de Internet.

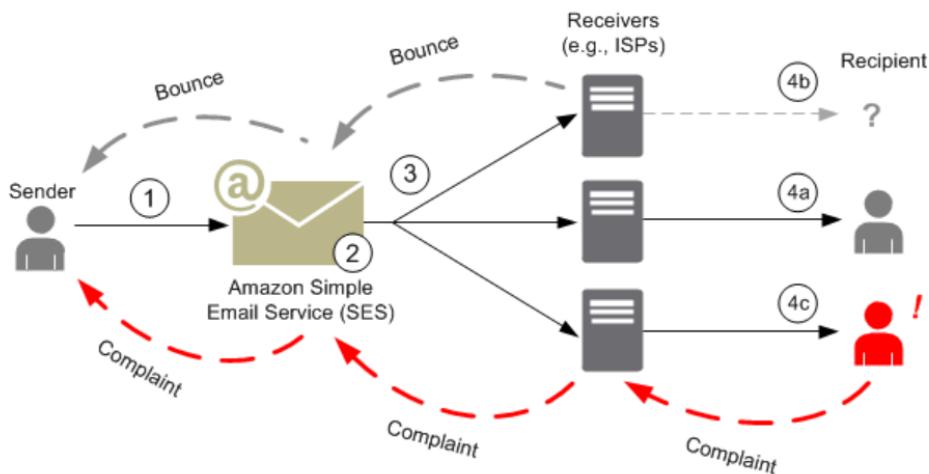


Figura 48. Arquitectura Amazon SES. Fuente [70].

Por lo tanto, teniendo en cuenta los parámetros medibles de los que se recogen los valores, se ha configurado un ejemplo concreto basado en un cultivo de hortalizas, donde los rangos óptimos son los siguientes:

- La temperatura del suelo oscilará entre 16°C y 24°C.
- El valor de pH del suelo oscilará entre 6 y 7.
- La temperatura del ambiente oscilará entre 18°C y 28°C.
- La humedad del ambiente oscilará entre 50% y 70%.
- La iluminancia oscilará entre 10.000 lux y 40.000 lux.

Por otro lado, respecto al nivel disponible de fertilizante en las instalaciones, para que salte una notificación de aviso será necesario que se encuentre en un valor que se ha configurado a modo de ejemplo por debajo de 500L.

5.5.1 Implementación y pruebas

A continuación, se explicarán brevemente los procesos necesarios para implementar Amazon SES y funciones AWS Lambda en el proyecto [71][72][73]. El primer paso es crear las credenciales SMTP para poder utilizar el servidor de correo electrónico proporcionado por el recurso SES. En este momento se abrirá un formulario donde se creará un usuario IAM con una política ya definida

The screenshot shows the 'Crear un usuario para SMTP' (Create new user for SMTP) step of the AWS IAM wizard. It includes fields for 'Nombre de usuario de IAM' (IAM user name) set to 'ses-smtp-user.2022' and 'IAM' set to '64 caracteres como mínimo'. Below the form, there is explanatory text about IAM and a JSON policy document for the user.

Este formulario le permite crear un usuario de IAM para la autenticación SMTP con Amazon SES. Escriba el nombre de un nuevo usuario de IAM o acepte la opción predeterminada. A continuación, haga clic en Crear para configurar las credenciales de SMTP.

Nombre de usuario de IAM: ses-smtp-user.2022

IAM: 64 caracteres como mínimo

Ocultar más información

Amazon SES utiliza AWS Identity and Access Management (IAM) para administrar las credenciales de SMTP. El nombre de usuario de IAM distingue entre mayúsculas y minúsculas; además, solo puede contener caracteres alfanuméricos y los símbolos +#=_,@_-.

Las credenciales de SMTP se componen de un nombre de usuario y una contraseña. Al hacer clic en el botón Crear, más abajo, las credenciales de SMTP se generan automáticamente.

Se concederá la política de IAM siguiente al nuevo usuario:

```
"Statement": [{"Effect": "Allow", "Action": "ses:SendRawEmail", "Resource": "*"}]
```

Figura 49. Creación de un usuario IAM para Amazon SES. Fuente: Elaboración propia.

Se mostrarán las credenciales asignadas para garantizar la seguridad de este y es importante descargarlas en el momento ya que no se podrá volver a tener acceso a ellas.

Finalmente será esencial verificar las identidades utilizadas con las que se va a utilizar SES. Estas identidades pueden ser dominios o direcciones de correo electrónico desde las cuales se van a enviar y recibir mensajes, coincidiendo esto último con la identidad empleada en este proyecto. Para ello simplemente se escoge el tipo de identidad como correo electrónico y se escribe la dirección de este, a continuación, se abre el email que llega a la dirección para finalizar el proceso de verificación.

Para poder enviar correos electrónicos utilizando AWS Lambda y Amazon SES como primer paso hay que crear un IAM Rol, *SESLambdaRole*, al que se le vincula la siguiente política, *SESLambdaPolicy*.

Editar SESLambdaPolicy

Una política define los permisos de AWS que puede asignar a un usuario, grupo o rol. Puede crear y editar una política en el editor visual y con JSON. [Más información](#)

Editor visual JSON Importar la política administrada

```
1  {
2      "Version": "2012-10-17",
3      "Statement": [
4          {
5              "Effect": "Allow",
6              "Action": "logs:CreateLogGroup",
7              "Resource": "arn:aws:logs:eu-west-1:449926048867:*log-group:/aws/lambda/NotificationPrueba:*"
8          },
9          {
10             "Effect": "Allow",
11             "Action": [
12                 "logs:CreateLogStream",
13                 "logs:PutLogEvents"
14             ],
15             "Resource": [
16                 "arn:aws:logs:eu-west-1:449926048867:log-group:/aws/lambda/NotificationPrueba:*"
17             ]
18         },
19         {
20             "Effect": "Allow",
21             "Action": [
22                 "ses:SendEmail",
23                 "ses:SendRawEmail"
24             ],
25             "Resource": "*"
26         }
27     ]
28 }
```

Figura 50. Política asociada al IAM Rol SESLambdaRole. Fuente: Elaboración propia.

Esto otorgará al rol de IAM el permiso de componer un mensaje de correo electrónico y hacer referencia a los valores publicados en el tema MQTT correspondiente. A la hora de crear el rol es importante seleccionar correctamente el caso de uso al que está destinado, en este caso se permitirá que acceda a recursos de AWS, en concreto Lambda. Por último, se le adjunta la política creada y un nombre.

A continuación, se crea la función Lambda mediante la cual se activará el envío de notificaciones. Ya que los mensajes son enviados a los temas en paquetes diferentes, según cual sea su dispositivo de origen, es necesario crear cuatro funciones Lambda en función de la casuística; *EmailNotificationTerreno*, *EmailNotificationAmbiental*, *EmailNotificationluminancia* y *EmailNotificationFertilizante*.

Todas las funciones utilizan el lenguaje de programación Python y el rol de ejecución creado anteriormente. A continuación, se muestra el fragmento de código empleado para la función *EmailNotificationTerreno*, pero todas seguirán la misma estructura utilizando las variables correspondientes [74]. Se puede ver como se han barajado todos los casos posibles para que en la notificación se sepa tanto que parámetro produce el error como su valor. Por otro lado, se ha añadido la condición de que si la desviación de los parámetros ocurre fuera de una franja horaria no se produzca ninguna notificación. Esto está pensado especialmente para las alertas generadas por los valores de iluminancia, ya que se asume y acepta que en horas nocturnas esos valores sean muy bajos o incluso nulos.

```

1 import boto3
2 from datetime import datetime
3 from datetime import time
4 import sys
5
6 client = boto3.client('ses')
7
8 def lambda_handler(event, context):
9     inicio=time(9)
10    final=time(21)
11    if datetime.now().time()>inicio and datetime.now().time()<final:
12        ph=float(str(event['phSoil']))
13        temperature=float(str(event['temperatureSoil']))
14
15        # Format text message from data
16        if ph<6 and 16<temperature<24:
17            message_text = "El valor del pH es muy ácido, pH= {0}.".format(
18                str(event['phSoil']))
19        elif ph>7 and 16<temperature<24:
20            message_text = "El valor del pH es muy básico, pH= {0}.".format(
21                str(event['phSoil']))
22
23        elif temperature<16 and 6<ph<7:
24            message_text = "La temperatura en el suelo es muy baja, T= {0} °C.".format(
25                str(event['temperatureSoil']))
26
27        elif temperature>24 and 6<ph<7:
28            message_text = "La temperatura en el suelo es muy alta, T= {0} °C.".format(
29                str(event['temperatureSoil']))
30
31    else:
32        message_text = "Los parámetros del terreno se encuentran fuera del rango óptimo. El valor del pH es {0} y de la temperatura {1} °C.".format(
33            str(event['phSoil']), str(event['temperatureSoil']))
34
35    client.send_email(
36        Source = 'xxxxxxxx@correo1.com',
37        Destination = [
38            {'ToAddresses': ['xxxxxxxx@correo2.com']}
39        ],
40        Message = {
41            'Subject': {
42                'Data': 'Amazon SES Notification',
43                'Charset': 'UTF-8'
44            },
45            'Body': {
46                'Text': {
47                    'Data': message_text,
48                    'Charset': 'UTF-8'
49                }
50            }
51        }
52    else:
53        sys.exit()

```

Figura 51. Fragmento de código empleado para la notificación de alertas. Fuente: Elaboración propia.

Para invocar a la función Lambda cuando algún parámetro publicado en los temas MQTT no se encuentre en el rango de valores óptimos es necesario crear una regla.

De igual manera que ocurre con las funciones, se empleará una regla para cada dispositivo, con la misma configuración, pero llamando a funciones Lambda diferentes y una instrucción SQL pensada para el suceso concreto. Por lo que se tendrá; *Notification_Terreno_Rule*, *Notification_Ambiental_Rule*, *Notification_Illuminance_Rule* y *Notification_Fertilizante_Rule*.

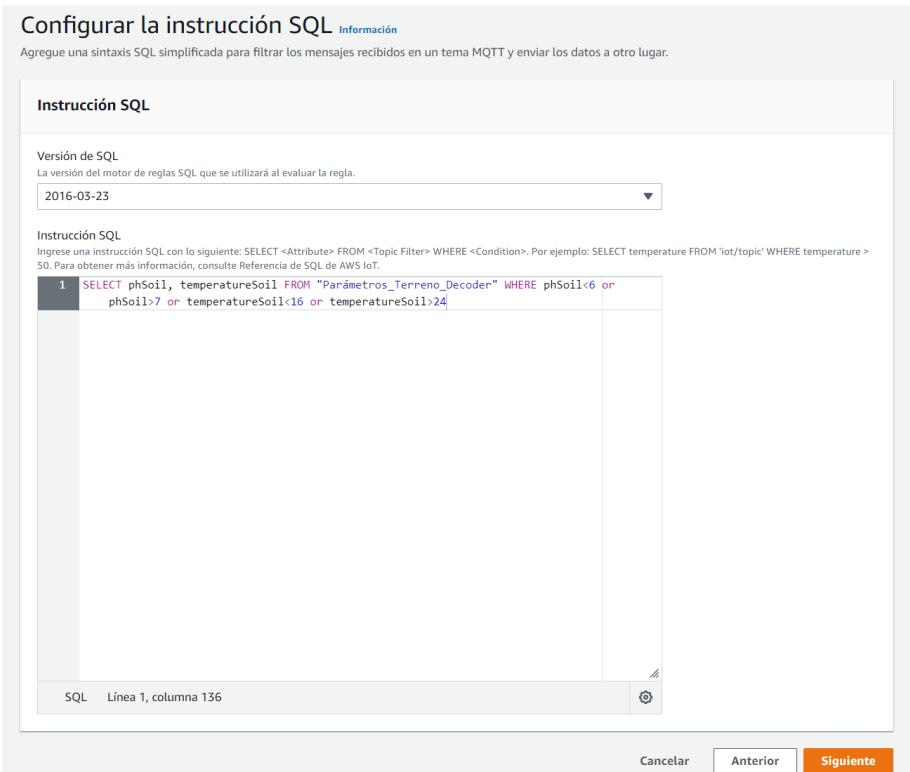


Figura 52. Ejemplo de instrucción SQL para Notification_Terreno_Rule. Fuente: Elaboración propia.

Una vez configurados todos los recursos empleados, se han realizado diferentes pruebas y se concluye que las configuraciones dan el resultado esperado. De esta manera, cuando el parámetro que se encuentra fuera del rango óptimo es perteneciente a los parámetros del terreno, el mensaje que se envía a la dirección de correo electrónico establecida es el siguiente.



Figura 53. Notificación de alerta vía correo electrónico. Fuente: Elaboración propia.

5.6 Pruebas del sistema global

En este proyecto, se desarrolla y configura un sistema prototipo de monitorización de cultivos siguiendo una arquitectura característica de un sistema IoT actual, implementando un limitado escenario de pruebas del mismo en el que se incluye un pequeño número de nodos de sensorización de distinto tipo. Se miden propiedades útiles del terreno, concretamente temperatura y pH del suelo, temperatura y humedad del ambiente, e iluminancia, mediante los dispositivos hardware con capacidad de comunicación LoRa que se adquirieron explícitamente para el proyecto. Por otro lado, también se ilustra el uso de un nodo de sensorización basado en Raspberry Pi para emular la medida remota del nivel de fertilizante que se encuentra disponible en los depósitos de la explotación.

Todos estos datos se almacenan y gestionan en una plataforma en la nube basada en AWS, dónde se integran, a través de la programación correspondiente, los nodos de sensorización disponibles para el escenario de pruebas. En dicha plataforma, se crea al menos un usuario con acceso para la consulta de datos actuales e históricos. Esto hará

que el usuario tenga control sobre dichas propiedades, y pueda tomar decisiones más informadas y tempranas referentes al mantenimiento de los cultivos.

Para que la comunicación entre los nodos sensorizados y la plataforma Cloud sea efectiva, hay que emplear una tecnología de comunicación adecuada en cada casuística. Esto dependerá directamente de la distribución del problema, utilizando LoRaWAN para los dispositivos que se encuentran a una distancia considerable de las instalaciones locales, como es el caso de LSPH01, EM500-CO2 y RAK1903, y Wi-Fi para dispositivos que pertenezcan a las mismas, concretamente la Raspberry Pi.

En todo este proceso también se tiene en cuenta la posibilidad de que existan cambios drásticos en los parámetros muestreados, y no se encuentren dentro de un rango de valores favorable para los cultivos. En este caso se dará un aviso a un usuario autorizado para ello mediante una dirección de correo electrónico.

A continuación, se explica el procedimiento para la realización de una prueba del sistema completo simulando una implantación real. Se podrá comprobar la eficacia de la tecnología de comunicación elegida, LoRaWAN, ya que los dispositivos se encontrarán distanciados y no en las mismas estancias, a diferencia de como ocurrió en las pruebas iniciales realizadas a medida que se configuraba el sistema. Por otro lado, también se podrá destacar la flexibilidad del sistema, ya que los cuatro dispositivos estarán conectados y funcionando simultáneamente.

La distribución escogida para esta demostración fue la siguiente:

- La puerta de enlace y el nodo de sensorización Raspberry Pi se encuentran en una estancia de la vivienda con cobertura Wi-Fi.
- Los dispositivos con soporte LoRa EM500-CO2 y LSPH01 se ubican en el terreno de cultivo con la ayuda de un poste y bridas o abrazaderas. La sonda de medición del dispositivo LSPH01 se coloca lo más cerca posible del dispositivo principal para evitar el despliegue de cableado.



Figura 54. Implementación real de los dispositivos EM500-CO2 y LSPH01. Fuente: Elaboración propia.

- El nodo LoRa RAK1903 se sitúa en las inmediaciones del terreno, donde la incidencia de luz es considerable.



Figura 55. Implementación real del dispositivo RAK1903. Fuente: Elaboración propia.

Respecto a las pruebas ejecutadas en AWS de manera individual no se ha realizado ningún cambio significativo, ya que las configuraciones de los dispositivos y la lógica seguida en la plataforma permiten escalar la solución a un número distinto de dispositivos sin necesidad de programación adicional.

En una primera prueba se sitúan los dispositivos en los lugares explicados y se procede a su análisis mediante la plataforma cloud. En este escenario concreto los avisos de valores fuera del rango configurado que aparecen son respecto a los valores del pH, temperatura del suelo, temperatura ambiental y humedad ambiental. Esto es una reacción esperada ya que esta primera prueba se realiza un día soleado y con temperaturas elevadas.

En las siguientes figuras se ve el correo electrónico enviado a través de SES para la notificación del evento tanto en lo referido a parámetros ambientales como a parámetros del terreno. Prestando atención a los procesos descritos sobre el código de notificación se puede observar como la función Lambda programada detecta que los parámetros que se desvían del valor óptimo son ambos, y ha enviado una notificación acorde a dicha situación.



Figura 56. Notificación de alerta en los parámetros de temperatura y humedad. Fuente: Elaboración propia.

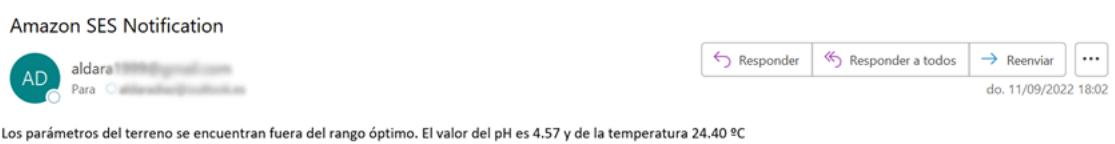


Figura 57. Notificación de alerta en los parámetros de pH y temperatura. Fuente: Elaboración propia.

Si se accede desde la consola a la tabla de Dynamo DB *monitorización_tfg* también se puede concluir que el almacenamiento de los datos se realiza de manera correcta, ya que se almacenan los valores en todo momento, indiferentemente de que sean óptimos o no, añadiendo además una marca temporal. A continuación, se muestra un

fragmento de la tabla donde se puede observar que el valor de temperatura y humedad que produjo una notificación se encuentra correctamente almacenado. Que estos valores se encuentren tabulados confirma que la publicación en el tema de MQTT se ha realizado también correctamente.

Elementos devueltos (294)		Acciones ▾	Crear elemento
<input type="checkbox"/>	Time	payload	▼
<input type="checkbox"/>	1662912340106	{ "temperature" : { "N" : "29" }, "humidity" : { "N" : "49.5" } }	▼

Figura 58. Ejemplo de mensajes almacenados en la tabla DynamoDB. Fuente: Elaboración propia.

A continuación, se elabora otro escenario forzando la aparición de valores fuera del rango normal por parte de la iluminancia y el nivel de fertilizante. A mayores también se publica un mensaje en el tema *Parámetros_Ambientales_Decoder*, donde solo se desvía de los valores establecidos la temperatura. En el caso del sensor de luz RAK1903 se encuentra colocado en un lugar sombrío donde no le incide poca luz solar.

Se puede observar cómo los dispositivos restantes también funcionan de manera correcta y como la función Lambda detecta con éxito el parámetro donde se produce el evento significativo aportando su valor exacto.



Figura 59. Notificación de alerta en el nivel de fertilizante disponible. Fuente: Elaboración propia.



Figura 60. Notificación de alerta en el parámetro de temperatura. Fuente: Elaboración propia.



Figura 61. Notificación de alerta en el parámetro de iluminancia. Fuente: Elaboración propia.

5.7 Código fuente

Todos los códigos fuente programados a lo largo del proyecto se encuentran en un repositorio público de GitHub con la siguiente URL (como se comentó previamente, se utilizaron los lenguajes Python, JavaScript y C++ según las necesidades):

https://github.com/GII/student_projects/2022_aldara_diaz

6 REFERENCIAS

- [1] «Una población en crecimiento. Naciones Unidas.,» [En línea]. Available: <https://www.un.org/es/global-issues/population>.
- [2] F. Panis, *La agricultura mundial en la perspectiva del año 2050.*, Italia, 2009.
- [3] «World: Total Population. United Nations.,» 2022. [En línea]. Available: <https://population.un.org/wpp/Graphs/Probabilistic/POP/TOT/900>.
- [4] J. Sesma, «Los retos de la urbanización: las tendencias del crecimiento urbano mundial. El mundo.,» 13 Abril 2015. [En línea]. Available: <https://www.elmundo.es/economia/2015/04/13/552b8fdae2704e8c7d8b4583.html>.
- [5] «Cómo el cambio climático afecta el agro. Agronegocios.,» 20 Diciembre 2016. [En línea]. Available: <https://www.agronegocios.co/agricultura/como-el-cambio-climatico-afecta-el-agro-2622315#:~:text=Seg%C3%BAn%20lo%20dijo%20el%20Ministerio%20de%20Ambiente%20y,en%20la%20composici%C3%B3n%20de%20las%20pasturas%20y%20forrajes..>
- [6] «Acción por el Clima. Naciones unidas.,» [En línea]. Available: <https://www.un.org/es/climatechange/what-is-climate-change>.
- [7] «IoT En agricultura. Cultivos y dispositivos inteligentes. IAT.,» 2020. [En línea]. Available: <https://iat.es/tecnologias/internet-de-las-cosas-iot/agricultura/#:~:text=Los%20beneficios%20de%20la%20aplicaci%C3%B3n%20del%20IoT%20en,los%20cultivos.%20Automatiza%20el%20trabajo%20en%20%C3%A1reas%20rurales..>
- [8] P. D. R. S. K. Nikesh Gondchawar, *IoT bases Smart Agriculture*, 2016.
- [9] «What is smart agriculture and why smart agriculture is the future?. Smart Farming Solutions.,» 28 Mayo 2020. [En línea]. Available: https://ondo.io/what_is_smart_agriculture.
- [10] A. C. Giacomo Veneri, *Hands-On Industrial Internet of Things*, Packt, 2018.
- [11] Y. A. S. Casadiego, *Importance of meteorological stations for decision making in agriculture.*, 2020..
- [12] «Temperatura del suelo para la siembra y el cultivo. Eos Data Analytics.,» 28 Julio 2021. [En línea]. Available: <https://eos.com/es/blog/temperatura-del-suelo/>.
- [13] «El control de la humedad del suelo: un factor clave. Eos Data Analytics.,» 9 Septiembre 2020. [En línea]. Available: <https://eos.com/es/blog/humedad-del-suelo/#:~:text=Unos%20niveles%20suficientes%20de%20humedad%20del%20suelo%20son,humedad%2C%20sino%20tambi%C3%A9n%20como%20regulador%20de%20la%20temperatura..>

- [14] S. Parent., «¿Cómo influye la humedad en la calidad de los cultivos? Promix.,» 9 Noviembre 2021. [En línea]. Available: <https://www.pthorticulture.com/es/centro-de-formacion/como-influye-la-humedad-en-la-calidad-de-los-cultivos/>.
- [15] «La luz como factor ambiental para las plantas. Nutricontrol.,» 11 Febrero 2020. [En línea]. Available: <https://nutricontrol.com/es/la-luz-como-factor-ambiental-para-las-plantas/>.
- [16] «La agricultura inteligente y el problemas de las redes en el campo. Agriculturers.,» 26 Septiembre 2017. [En línea]. Available: <https://agriculturers.com/la-agricultura-inteligente-y-el-problema-de-las-redes-en-el-campo/>.
- [17] «6 Leading Types of IoT Wireless Tech and Their Best Use Cases. Behrtech.,» [En línea]. Available: <https://behrtech.com/blog/6-leading-types-of-iot-wireless-tech-and-their-best-use-cases/>.
- [18] V. Ferrer, 2020. [En línea]. Available: <https://vicentferrer.com/sigfox/>.
- [19] B. S. Chaudhari. Marco Zennaro, LPWAN Technologies for IoT and M2M Applications, Elsevier., 2020.
- [20] V. Ferrer, 2020. [En línea]. Available: <https://vicentferrer.com/lora-lorawan/>.
- [21] «AWS IoT Core. Amazon Web Services,» 2022. [En línea]. Available: <https://aws.amazon.com/es/iot-core/>.
- [22] «Solución para productos inteligentes. Amazon Web Services.,» 2022. [En línea]. Available: <https://aws.amazon.com/es/solutions/implementations/smart-product-solution/>.
- [23] «Azure IoT. Microsoft Azure,» 2022. [En línea]. Available: <https://azure.microsoft.com/es-es/solutions/iot/#overview>.
- [24] «Azure Architecture Center. Microsoft.,» 2022. [En línea]. Available: <https://docs.microsoft.com/en-us/azure/architecture/>.
- [25] «Google Cloud. Wikipedia,» 2 Junio 2021. [En línea]. Available: https://es.wikipedia.org/wiki/Google_Cloud.
- [26] «IoT Core. Google Cloud.,» 2022. [En línea]. Available: <https://cloud.google.com/iot-core/>.
- [27] «AWS IoT Core for LoRaWAN. Amazon Web Services.,» 2022. [En línea]. Available: <https://aws.amazon.com/es/iot-core/lorawan/>.
- [28] D. Prodanova, «¿How do LoRaWAN gateways work?,» 31 Agosto 2021. [En línea]. Available: <https://news.rakwireless.com/how-do-lorawan-gateways-work/>.
- [29] «LoRa Basics Station. LoRa Developer Portal.,» 2022. [En línea]. Available: <https://lora-developers.semtech.com/build/software/lora-basics/lora-basics-for-gateways>.

- [30] «Embedded Network Server Bunfiles MQTT In LoRaWAN Gateway. Milesight IoT.,» 2022. [En línea]. Available: <https://www.milesight-iot.com/blog/mqtt-forwarding/>.
- [31] «Developer Survey,» 2021. [En línea]. Available: <https://insights.stackoverflow.com/survey/2021#most-popular-technologies-language>.
- [32] «Access SQL: conceptos básicos, vocabulario y sintaxis. Microsoft.,» 2022. [En línea]. Available: <https://support.microsoft.com/es-es/office/access-sql-conceptos-b%C3%A1sicos-vocabulario-y-sintaxis-444d0303-cde1-424e-9a74-e8dc3e460671#bm1>.
- [33] B. IoT, «Protocolos de comunicación en IoT que deberías conocer. Barbara,» 29 Abril 2021. [En línea]. Available: <https://barbaraiot.com/blog/protocolos-iot-que-deberias-conocer/#:~:text=Respecto%20a%20la%20elecci%C3%B3n%20entre%20los%20diferentes%20protocolos,protocolos%20m%C3%A1s%20habituales%20son%20MQTT%2C%20CoAP%20y%20HTTP..>
- [34] «¿Qué importancia tienen los protocolos de comunicación?. Cuovadis,» 2022. [En línea]. Available: <https://cuovadis.es/faq/que-importancia-tienen-los-protocolos-de-comunicacion/>.
- [35] «MQTT communication patterns. Amazon Web Services.,» 2022. [En línea]. Available: https://docs.aws.amazon.com/fr_fr/whitepapers/latest/designing-mqtt-topics-aws-iot-core/mqtt-communication-patterns.html.
- [36] «¿Qué es el HTTP?. IONOS Digital Guide,» 14 Julio 2020. [En línea]. Available: <https://www.ionos.es/digitalguide/hosting/cuestiones-tecnicas/protocolo-http/>.
- [37] «¿Qué es una API Rest?,» 8 Mayo 2022. [En línea]. Available: <https://www.redhat.com/es/topics/api/what-is-a-rest-api>
- [38] «MQTT, ¿Qué es?, ¿Cómo se puede usar? y ¿Cómo funciona?. DescubreArduino.,» [En línea]. Available: <https://descubreadduino.com/mqtt-que-es-como-se-puede-usar-y-como-funciona/#:~:text=C%C3%B3mo%20funciona%20MQTT%20Una%20sesi%C3%B3n%20MQTT%20se%20divide,puerto%20personalizado%20definido%20por%20los%20operadores%20del%20broker..>
- [39] «MQTT Basics. HIVEMQ.,» 2022. [En línea]. Available: https://www.hivemq.com/mqtt-essentials/?utm_source=bing&utm_medium=ppc&utm_campaign=EU%20%7C%20MQTT%20Main&utm_content=MQTT%20Essentials&utm_term=mqt&msclkid=80c55a966867157b1b80384017502ee6.
- [40] «Raspberry Pi. Wikipedia,» 16 Mayo 2022. [En línea]. Available: https://es.wikipedia.org/wiki/Raspberry_Pi.
- [41] «Raspi pi 4B. Reichelt elektronik.,» [En línea]. Available: <https://www.reichelt.de/de/de/raspberry-pi-4-b-4x-1-5-ghz-4-gb-ram-wlan-bt-rasp-pi-4-b-4gb-p259920.html?r=1>.

- [42] «WisGate Edge Lite 2. RAKwireless Store.,» [En línea]. Available: <https://store.rakwireless.com/products/wisgate-edge-lite-2-rak7268-rak7268c?variant=39942861914310>.
- [43] «LSPH01-LoRaWAN Soil pH Sensor. Dragino.,» [En línea]. Available: <https://www.dragino.com/products/agriculture-weather-station/item/184-lsp01.html>.
- [44] «Carbon Dioxide Sensor (4 in 1). Milesight IoT,» 2022. [En línea]. Available: <https://www.milesight-iot.com/lorawan/sensor/em500-co2/>.
- [45] «Milesight IoT EM500-CO2 Sensor 4en1 LoRaWAN. MakerStore.,» [En línea]. Available: <https://maker-store.es/iot-lora-nb-iot-rfid-m2m-co/lora/lora-sensoren/4789/milesight-iot-em500-co2-sensor-lorawan>.
- [46] « WisBlock Kit2. RAKwireless Store.,» [En línea]. Available: <https://store.rakwireless.com/products/wisblock-kit-2-lora-based-gps-tracker-with-solar-panel?variant=37637925175494>.
- [47] «Ligh Sensor Texas Instruments. RAKwireless Store.,» [En línea]. Available: <https://store.rakwireless.com/products/rak1903-opt3001dnpr-ambient-light-sensor>.
- [48] «Software de Raspberry Pi. Raspberry.,» [En línea]. Available: <https://raspberrypi.cl/software-de-raspberry/>.
- [49] D. Prodanova, «The future of RAK's gateways is WisGateOS. RAK News Hub.,» 12 Abril 2021. [En línea]. Available: https://news.rakwireless.com/the-future-of-raks-gateways-is-wisgateos/?utm_source=RAK7268&utm_medium=Footer&utm_campaign=RAKwireless.
- [50] «WisGate OS User Manual. RAK Documentation Center.,» 29 Julio 2022. [En línea]. Available: <https://docs.rakwireless.com/Product-Categories/Software-APIs-and-Libraries/WisGateOS/Overview/>.
- [51] «RAK7268 WisGate Edge Lite 2. RAK Documentation Center.,» 29 Julio 2022. [En línea]. Available: <https://docs.rakwireless.com/Product-Categories/WisGate/RAK7268/Overview/>.
- [52] «LSPH01- LoRaWAN Soil pH Sensor User Manual. Dragino Wiki,» 18 Agosto 2022. [En línea]. Available: <http://wiki.dragino.com/xwiki/bin/view/Main/User%20Manual%20for%20LoRaWAN%20End%20Nodes/LSPH01-LoRaWAN%20Soil%20pH%20Sensor%20User%20Manual/#H200B1.2Features>.
- [53] «WisBlock RAKBox-B2 Enclosure. RAK Documentation Center.,» 29 Julio 2022. [En línea]. Available: <https://docs.rakwireless.com/Product-Categories/Accessories/RAKBox-B2/Overview/>.
- [54] «RAK4631 Quick Start Guide. RAK Documentation Center,» 29 Julio 2022. [En línea]. Available: <https://docs.rakwireless.com/Product-Categories/WisBlock/RAK4631/Quickstart/#prerequisite>.

- [55] «RAK1903 WisBlock Ambient Light Sensor Module. RAK Documentation Center.,» 29 Julio 2022. [En línea]. Available: <https://docs.rakwireless.com/Product-Categories/WisBlock/RAK1903/Overview/>.
- [56] «RAK1903 Quick Start Guide. RAK Documentation Center,» 29 Julio 2022. [En línea]. Available: <https://docs.rakwireless.com/Product-Categories/WisBlock/RAK1903/Quickstart/#hardware-setup>.
- [57] «RAK5005-O WisBlock Base Board. RAK Documentation Center.,» 29 Julio 2022. [En línea]. Available: <https://docs.rakwireless.com/Product-Categories/WisBlock/RAK5005-O/Overview/#product-description>.
- [58] «RAK5005-O Baseboard Installation Guide. RAK Documentation Center,» 10 Noviembre 2021. [En línea]. Available: <https://docs.rakwireless.com/Knowledge-Hub/Learn/RAK5005-O-Baseboard-Installation-Guide/>.
- [59] B. Giesecke, «Installation of Board Suppor Package in Arduino IDE. RAK Documentation Center.,» 27 Diciembre 2021. [En línea]. Available: <https://docs.rakwireless.com/Knowledge-Hub/Learn/Installation-of-Board-Support-Package-in-Arduino-IDE/>.
- [60] «RAKwireless Unifies Interfae V3 (RUI3). RAK Documentation Center,» 5 Agosto 2022. [En línea]. Available: <https://docs.rakwireless.com/RUI3/#overview>.
- [61] «RAK4631-R WisBlock LPWAN Module. RAK Documentation Center,» 5 Agosto 2022. [En línea]. Available: <https://docs.rakwireless.com/Product-Categories/WisBlock/RAK4631-R/Overview/>.
- [62] «RAKWireless BSP Support For The Arduino Board Manager. Github.,» 2022. [En línea]. Available: <https://github.com/RAKWireless/RAKwireless-Arduino-BSP-Index>.
- [63] A. W. Services, «Getting Started with AWS IoT Core for LoRaWAN. Youtube,» 26 Febrero 2021. [En línea]. Available: <https://www.youtube.com/watch?v=6-ZrdRjqdTk>.
- [64] H. Hua, «How to Connect Milesight IoT End devices to AWS IoT Core for LoRaWAN,» 12 Junio 2022. [En línea]. Available: [https://support.milesight-iot.com/support/solutions/articles/73000514172-how-to-connect-milesight-iot-end-devices-to-aws-iot-core-for-lorawan#Create%20lambda%20function%20for%20destination%20rule](https://support.milesight-iot.com/support/solutions/articles/73000514172-how-to-connect-milesight-iot-end-devices-to-aws-iot-core-for-lorawan>Create%20lambda%20function%20for%20destination%20rule).
- [65] «Notes for AWS-IoT-Core. Dragino Wiki.,» 9 Septiembre 2022. [En línea]. Available: <http://wiki.dragino.com/xwiki/bin/view/Main/AWS%20IoT%20Core%20for%20LoRaWAN/#H3.1.1CREATEDEVICEPROFILE>.
- [66] «AWS IoT Core Integration. RAK Documentation Center.,» 10 Noviembre 2021. [En línea]. Available: <https://docs.rakwireless.com/Knowledge-Hub/Learn/AWS-IoT-Core-Integration/>.
- [67] «Milesight IoT. Github.,» 2022. [En línea]. Available: <https://github.com/Milesight-IoT/SensorDecoders>.

- [68] «LoRa. Cayenne Docs.,» [En línea]. Available: <https://developers.mydevices.com/cayenne/docs/lora/#lora-cayenne-low-power-payload-data-types>.
- [69] «¿Qué es Amazon DynamoDB?Amazon,» 2022. [En línea]. Available: https://docs.aws.amazon.com/es_es/amazondynamodb/latest/developerguide/Introduction.html.
- [70] «Simple Email Service. Amazon Web Services.,» 2022. [En línea]. Available: https://docs.aws.amazon.com/es_es/ses/latest/dg/Welcome.html.
- [71] S. Paul., «Sending Email using AWS Lambda and Amazon SES. JavaScript in Plain English.,» 30 Octubre 2020. [En línea]. Available: <https://javascript.plainenglish.io/sending-email-using-aws-lambda-and-amazon-ses-3eab39517ca9>.
- [72] I. Maselesele, «Sending an email with AWS Lambda Function (Python). JavaScript in Plain English.,» 20 Abril 2020. [En línea]. Available: <https://ivhani.medium.com/sending-an-email-with-aws-lambda-function-python-c4533aabf4af>.
- [73] «¿Cómo puedo enviar emails con Lambda y Amazon SES?. Amazon,» 15 Septiembre 2021. [En línea]. Available: <https://aws.amazon.com/es/premiumsupport/knowledge-center/lambda-send-email-ses/>.
- [74] «Boto3 documentation.,» 2022. [En línea]. Available: <https://boto3.amazonaws.com/v1/documentation/api/latest/index.html>.

ANEXOS

1. Presupuestos

HARDWARE			
PRODUCTO	Unidades	Precio/Unidad	Importe
RAK Wireless WisGate Edge Lite 2	1	140,97 €	140,97 €
Dragino LSPH01	1	127,85 €	127,85 €
RAK Wireless WisBlock Kit GPS con panel solar	1	80,09 €	80,09 €
RAK Wireless WisBlock Sensor de luz	1	4,69 €	4,69 €
Batería recargable 3.7 V Li-Ion	1	10,29 €	10,29 €
Milesight IoT EM500-CO2	1	246,40 €	246,40 €
Raspberry Pi 4 Model B 4GB	1	69,95 €	69,95 €
Raspberry Pi SenseHat	1	36,95 €	36,95 €
Caja de protección Raspberry Pi 4	1	9,95 €	9,95 €
Alimentador Raspberry Pi 4 USB-C	1	8,95 €	8,95 €
Sandisk MicroSD 32 GB	1	11,34 €	11,34 €
AZDelivery convertidor USB a TTL	1	7 €	7 €
IMPORTE TOTAL			754,43 €

SOFTWARE			
Producto	Unidades	Precio/Unidad	Importe
Plataforma AWS (Capa Gratuita)	1	0,00 €	0,00 €
AWS IoT Core for LoRaWAN	1	2,30 €	2,30 €
Amazon SES	1	0,12 €	0,12 €
Librerías y herramientas de programación	1	0,00 €	0,00 €
IMPORTE TOTAL			2,42 €

MANO DE OBRA			
Producto	Horas	Precio/Hora	Importe
Desarrollo del proyecto	400	35 €	14.000,00 €
IMPORTE TOTAL			14.000,00 €

2. Manuales

Configuración del Gateway LoRa

Para configurar la puerta de enlace RAK7286 es necesario acceder a ella. Para esto hay dos métodos, mediante el puerto WAN del enrutador o utilizando el modo Wi-Fi AP, que es lo que se ha hecho en este proyecto. Para esto se accede a la configuración Wi-Fi del ordenador y habrá una red llamada RAK7268_XXXX, donde los últimos cuatro dígitos corresponden a la dirección MAC del propio Gateway.

Una vez aquí, se conecta el ordenador a dicha red y se accede en el navegador a la dirección IP asignada de manera predeterminada a la puerta de enlace, “192.168.230.1”. En esta dirección aparecerá la página principal de la interfaz gráfica, donde habrá que proporcionar el usuario y la contraseña correspondiente.

Después de introducir las credenciales correctamente, se abre la interfaz de configuración y supervisión de la puerta de enlace. Aquí aparece información básica, como el número de mensajes recibidos, transmitidos o nodos; también se podrán consultar las estadísticas proporcionadas sobre el comportamiento del Gateway.

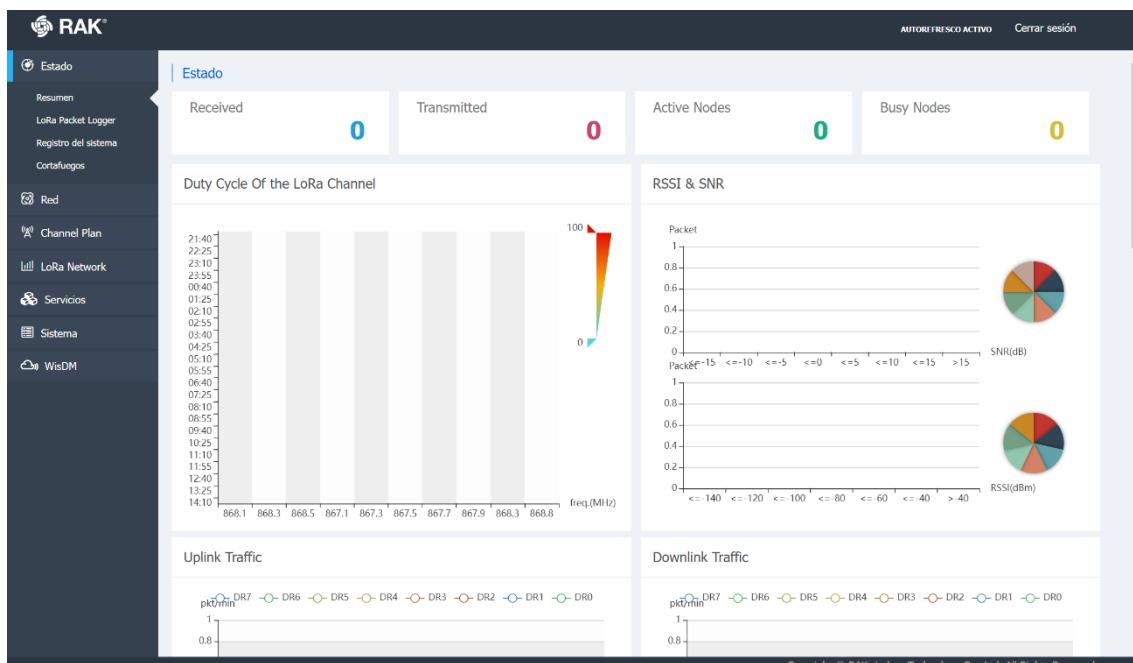


Figura 62. Interfaz de configuración y supervisión del Gateway. Fuente: Elaboración propia.

Ahora es necesario conectarse a través de la red Wi-Fi. Para esto, se accede a Red y Wi-Fi. La puerta de enlace puede funcionar en tres modos diferentes, punto de acceso, cliente o ambas. El punto de acceso permite conectarse a la puerta de enlace mediante la dirección IP asignada, y el modo cliente hace que dicho dispositivo se conecte a una red inalámbrica, pero sin la opción AP disponible. En este caso se ha elegido que funcione en ambos modos, aunque lo básico para el proyecto es el modo cliente.

Para eso se selecciona Punto de acceso + Cliente en el campo Modo. Ahora aparecerán dos zonas a cubrir, por un lado, Wireless Access Point, correspondiente al modo Punto de acceso; y Wireless Client, correspondiente al modo Cliente.

Se empieza cubriendo lo relacionado con el modo AP, donde el campo ESSID (nombre del punto de acceso) viene cubierto de manera predeterminada como RAK7268_XXXX. En el campo Encriptación se elige el tipo de cifrado utilizado para proteger esta red, en este proyecto concreto se ha utilizado WPA-PSK/WPA2-PSK

Mixed Mode, y por último un campo Clave, donde se escribe una contraseña para proteger la red.

Ahora se cubrirá lo necesario para el modo Cliente. En este caso en el campo ESSID se selecciona la red a la que se quiere que se conecte el Gateway. Una vez completado este paso, se selecciona el cifrado que utiliza dicha red y la contraseña que la protege. Por último, dentro de este modo aparecen opciones respecto a la dirección IP, ya que esta puede ser una Dirección estática o la que el servidor DHCP del router le proporcione.

Una vez guardados y aplicados los cambios, se ve como en este apartado aparece la dirección IP que se le asignó al Gateway, la cual permitirá a partir de ahora acceder a esta interfaz.

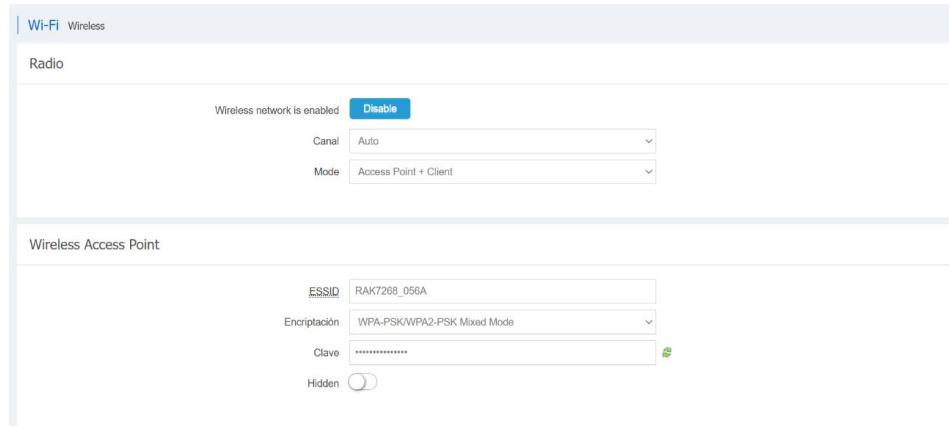


Figura 63. Configuración del modo AP del Gateway. Fuente: Elaboración propia.

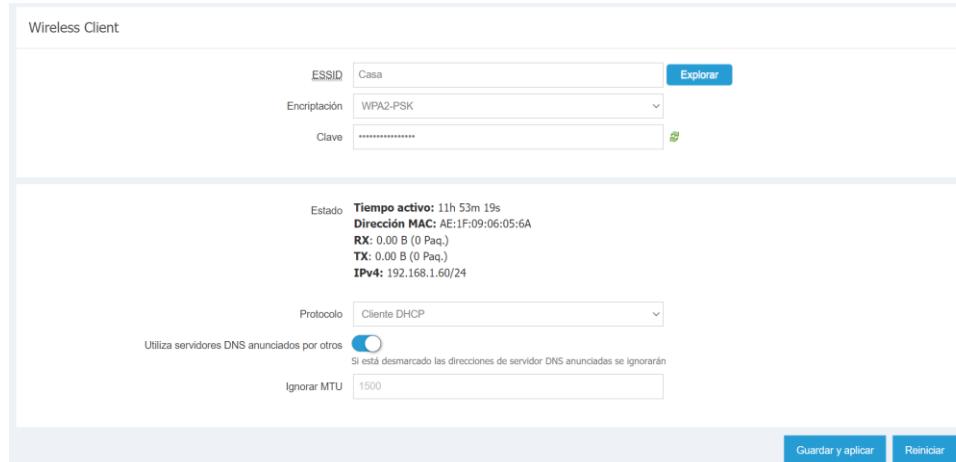


Figura 64. Configuración del modo cliente del Gateway. Fuente: Elaboración propia.

Otra cuestión de gran relevancia es la actualización del firmware. Buscando en la documentación, se ha encontrado que el último firmware disponible, WisGateOS_1.3.4, es más reciente que el que trae el dispositivo preinstalado, WisGateOS_1.2.3.

Para actualizarlo primero se es necesario descargar el archivo desde la documentación oficial RAK7268, para instalarlo a través de la interfaz. Despues de realizar la actualización correctamente, aparece por pantalla una suma de

comprobación, el tamaño del archivo y un mensaje sobre si se mantiene la configuración previamente realizada en el dispositivo.

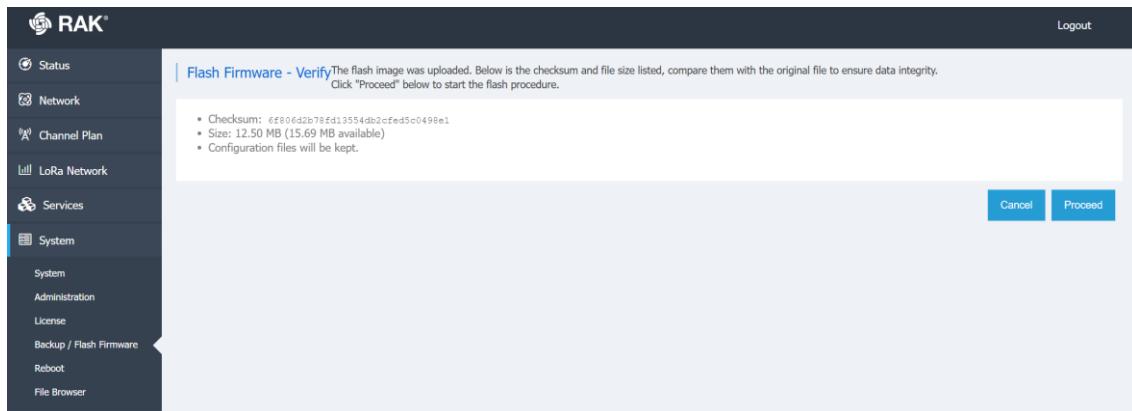


Figura 65. Actualización del firmware del Gateway. Fuente: Elaboración propia.

Configuración RAK4631-R + RAK1903

Para configurar RAK4631 es necesario descargar e instalar Arduino IDE¹.



Figura 66. Arduino IDE 1.8.19. Fuente: Elaboración propia.

La actualización del firmware del dispositivo para pasar de RAK4631 a 4631-R permite utilizar la API de RUI3².

Primero se actualiza el gestor de arranque, para ello hay que descargar los archivos adafruit-nrfutil.ex y rui3_nrf52840_bootloader_latest.zip de la documentación oficial y guardarlos en una carpeta creada en el PC, en este caso ubicada en el disco C. Una vez hecho esto, se ejecutan los siguientes comandos en el terminal del PC.

```
cd C:\RAK4631_Bootloader_to_RUI_v3  
adafruit-nrfutil.exe --verbose dfu serial -- package rui3_nrf52840_bootloader_latest.zip --port COM5 -b 115200 -- singlebank -- touch 1200
```

Figura 67. Comando para la actualización del gestor de arranque de RAK4631. Fuente: Elaboración propia.

En ellos es necesario tener en cuenta el puerto al que está conectado el dispositivo al ordenador, comprobándolo en el administrador de dispositivos.

¹ [Installation of Board Support Package in Arduino IDE | RAKwireless Documentation Center](#)

² [RAKwireless Unified Interface V3 \(RUI3\) | RAKwireless Documentation Center](#)

Una vez actualizado el gestor de arranque se procede a la actualización del firmware³. Ahora se descargan los archivos nrfutil.exe y rui3_rak4631_latest.zip y se guardan en la carpeta creada anteriormente. Para actualizarlos se ejecuta el siguiente comando comprobando previamente el puerto COM, ya que al actualizar el gestor de arranque este pudo cambiar.

```
nrfutil.exe dfu serial -pkg rui3_rak4631_latest.zip -p COM6
```

Figura 68. Comando para la actualización del firmware de RAK4631. Fuente: Elaboración propia.

Finalmente, el dispositivo RAK4631 ahora es RAK4631-R. Para verificar si la carga del firmware se hizo correctamente se utiliza el monitor serie de Arduino.

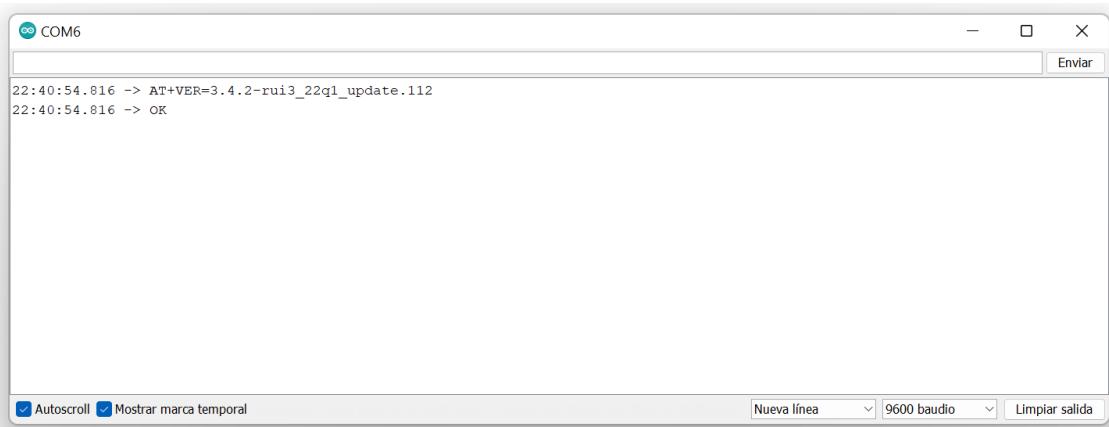


Figura 69. Actualización de firmware RAK4631-R realizada con éxito. Fuente: Elaboración propia.

Una vez listo se instala RAK Wireless Arduino BSP para WisBlock⁴. En el propio programa de Arduino se accede a Archivo y Preferencias, y en el gestor de URLs Adicionales de Tarjetas se copia la siguiente URL.

https://raw.githubusercontent.com/RAKWireless/RAKwireless-Arduino-BSP-Index/main/package_rakwireless.com_rui_index.json

Se procede a la configuración en Arduino. Lo primero es acceder a Herramientas, y seleccionar Gestor de tarjetas.

Se abrirá la siguiente pestaña, donde se instalan los módulos necesarios, que en este caso es RAK Wireless RUI nRF Boards, ya que se utiliza RAK4631-R.

³ [RAK4631-R WisBlock LPWAN Module | RAKwireless Documentation Center](#)

⁴ [GitHub - RAKWireless/RAKwireless-Arduino-BSP-Index: RAKwireless BSP Support for the Arduino Board Manager](#)

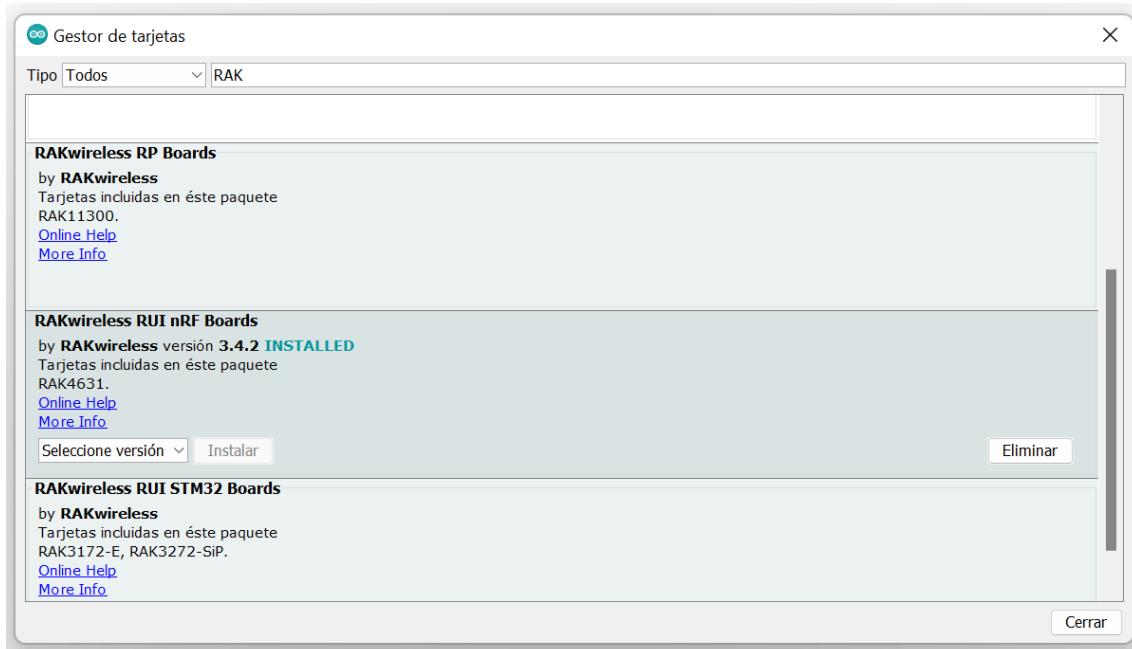


Figura 70. Gestor de tarjetas en Arduino IDE. Fuente: Elaboración propia.

Una vez instalado, se selecciona en Herramientas la placa WisBlock RAK4631.

Como el sensor que se utiliza es el RAK1903 es necesario instalar manualmente la biblioteca correspondiente. Para ello se accede a la documentación oficial y se descarga un archivo RAKwireless_RAK1903_Optical_OPT3001.zip, para a continuación añadir la librería a Arduino IDE.

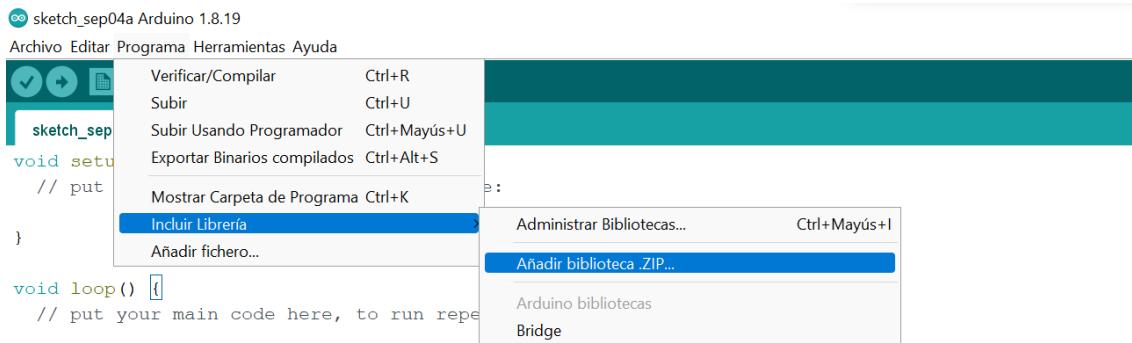


Figura 71. Instalación manual de la biblioteca de RAK1903. Fuente: Elaboración propia.

Creación de una cuenta gratuita de AWS

El primer paso para crear una cuenta de AWS es acceder a la página de la plataforma AWS y seleccionar la opción “Crear una cuenta gratuita”.

Ahora se cubrirán todos los datos iniciales, como el correo electrónico y el nombre de la cuenta. A continuación, AWS enviará un correo para verificar la dirección del email introducido.



Explore los productos de la capa gratuita con una cuenta de AWS nueva.

Para obtener más información, visite aws.amazon.com/free.



Registrarse en AWS

Dirección de correo electrónico del usuario raíz
Se utiliza para la recuperación de cuentas y algunas funciones administrativas

Nombre de la cuenta de AWS
Elija un nombre para la cuenta. Podrá cambiarlo en la configuración de la cuenta después de registrarse.

Verificar la dirección de correo electrónico

O
Iniciar sesión en una cuenta de AWS existente

Figura 72. Registro en AWS. Fuente: Elaboración propia.

Una vez finalizados estos dos primeros pasos, será necesario introducir cierta información que van pidiendo por pantalla, como adjudicarle una contraseña a la cuenta, un número de teléfono, el país y una tarjeta de crédito como método de pago. Esto último no implica que se esté creando una cuenta no gratuita, simplemente que, si se acceden a ciertos servicios que no se encuentran dentro de la capa gratuita, AWS cobrará a final de mes a la cuenta bancaria enlazada a dicha cuenta.

Para acceder a la nube, se hará como usuario raíz, lo cual quiere decir que se accede con acceso ilimitado, ya que es la persona que gestiona la cuenta. Para esto solo será necesario introducir tanto la dirección de correo como la contraseña adjudicada en la creación de la cuenta.

Por último, en la propia Consola, hay que definir las regiones y zonas en las que se va a trabajar, siendo las regiones ubicaciones geográficas específicas donde se alojan los recursos, que a su vez cuentan con zonas, que son las áreas de implementación para dichos recursos. Esto tiene especial importancia ya que ubicar los distintos recursos en regiones diferentes puede incurrir en gastos indeseados. Por esta razón la región que se utiliza en todos los servicios es “Europa (Irlanda) eu-west-1”, ya que cuenta con el recurso IoT Core for LoRaWAN.

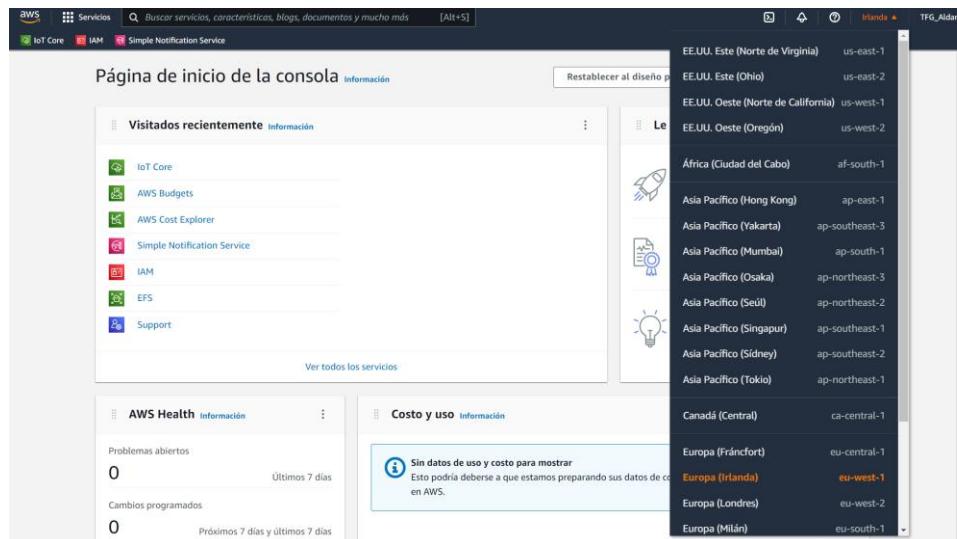


Figura 73. Definir la región de AWS. Fuente: Elaboración propia.

Creación de usuarios y asignar permisos

Ahora se explicará brevemente los pasos para crear un usuario y concederle permisos para administrar los recursos de AWS. En primer lugar, se accederá al servicio IAM, que es el que permite administrar los usuarios y el acceso al resto de recursos. En el panel de navegación se elegirá la opción “Usuarios” para posteriormente agregar uno. El tipo de acceso escogido será mediante programación.

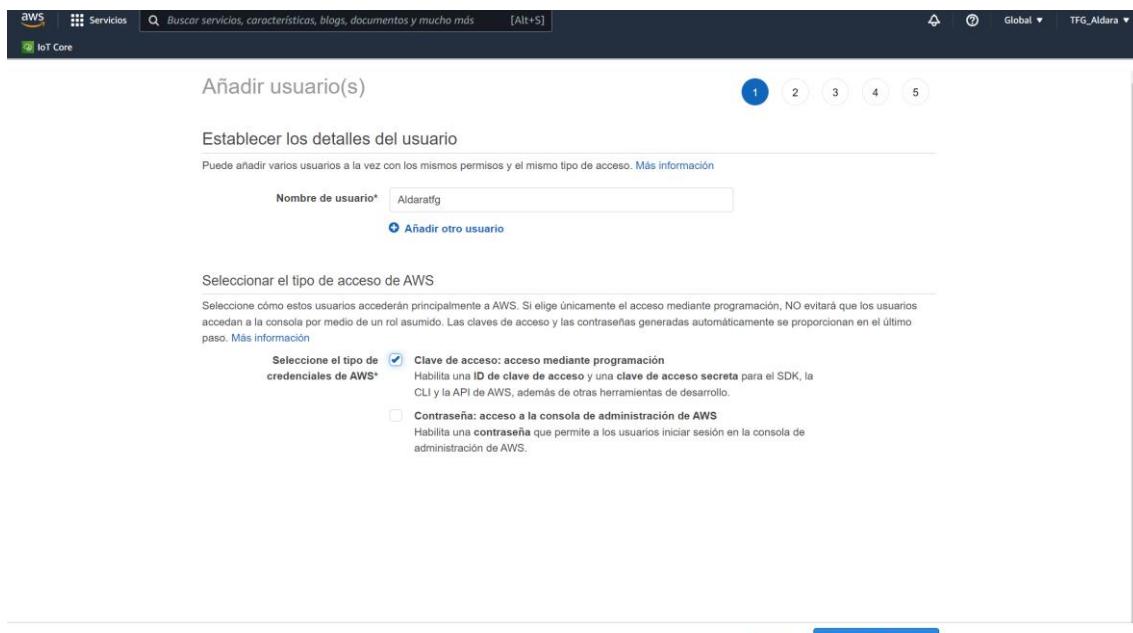


Figura 74. Creación de un usuario IAM. Fuente: Elaboración propia.

En el siguiente paso se establecerán los permisos al usuario. Para esto se asociarán las políticas existentes y se escogerá la opción *AdministratorAccess*.

Por último, antes de crear el usuario, AWS adjunta toda la información cubierta antes con el fin de asegurarse de que todo está correcto. Una vez revisado, se creará el usuario. Finalmente, se descargarán las credenciales de seguridad del usuario.

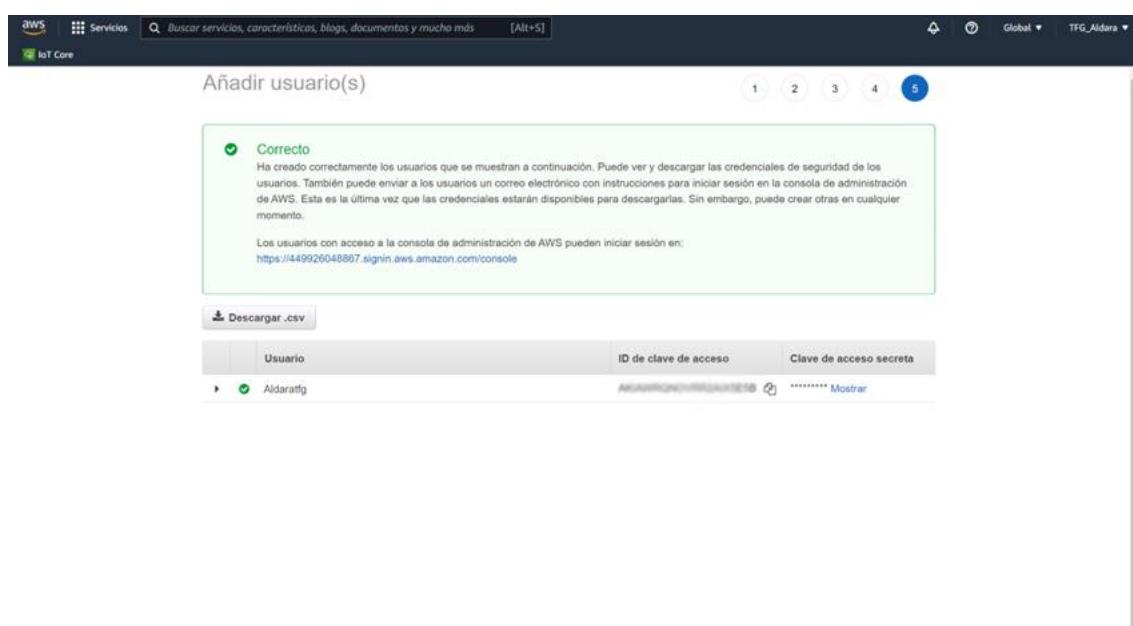


Figura 75. Credenciales de seguridad del usuario. Fuente: Elaboración propia.

Definición de dispositivos tipo Raspberry Pi en AWS

Dentro del servicio IoT Core se definirán dispositivos virtuales, correspondientes a dispositivos reales, aplicado a este proyecto una Raspberry Pi.

El primer paso es crear una política mediante IoT Core. Estas políticas concederán a los objetos permisos para obtener acceso a los diferentes recursos de AWS, como pueden ser otros objetos o los diferentes temas MQTT.

Para crearla no hay más que acceder a crear política y añadir en las instrucciones las acciones que podrá realizar el dispositivo.

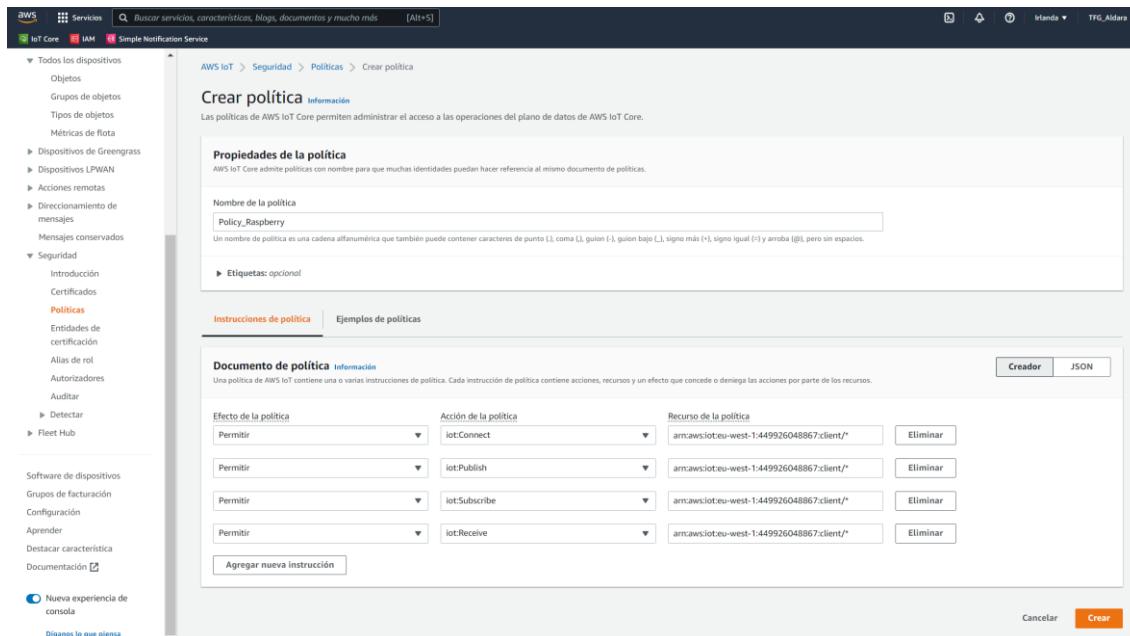


Figura 76. Política asociada a Raspberry Pi. Fuente: Elaboración propia.

La política *Connect* permite conectarse al agente de mensajes de AWS IoT Core, *Publish* hace que sea posible publicar en un tema de MQTT, *Subscribe* deja que el dispositivo se subscriba a un filtro de temas y *Receive* que reciba mensajes de AWS IoT Core. Posteriormente en el apartado Administración, en la sección Objetos se procederá a crear un único objeto.

A continuación, se deberán cubrir una serie de propiedades, donde la única obligatoria es el nombre, y a mayores en este caso se ha añadido el objeto a un grupo, *Instalaciones_Local*, al cual pertenecerán todos los dispositivos que se encuentren cerca del centro de control, en este caso la Raspberry.

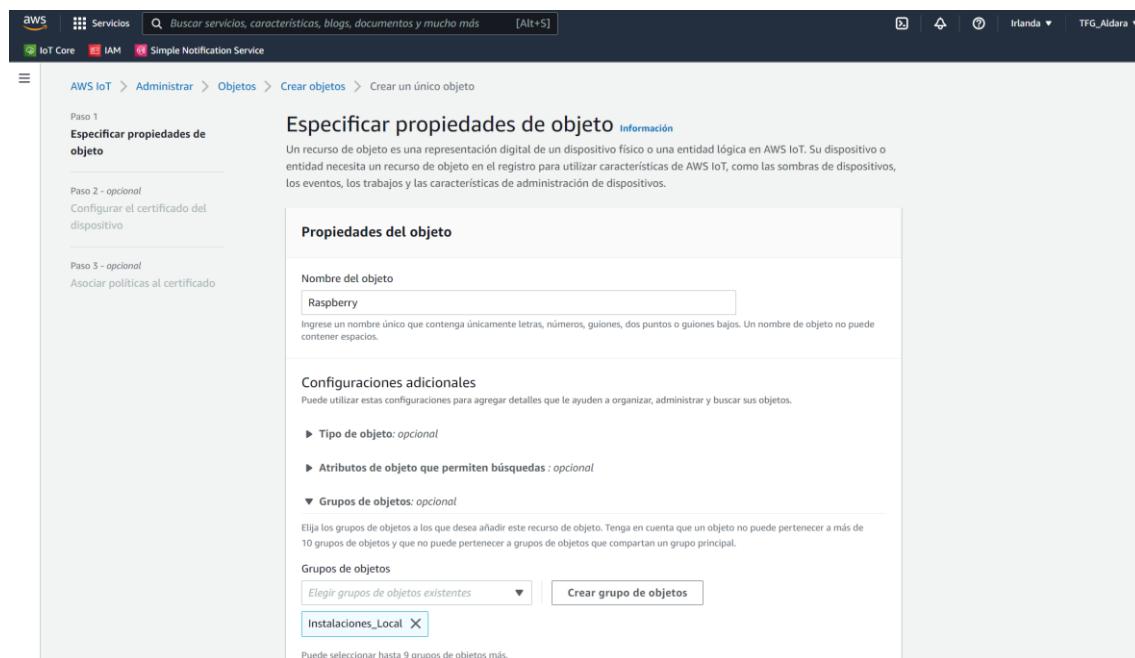


Figura 77. Propiedades del objeto Raspberry. Fuente: Elaboración propia.

Ahora se configura el certificado del dispositivo para conectarse con IoT Core, la opción escogida es que automáticamente la plataforma genere la clave pública y privada.

Es necesario asociarle al objeto una política, donde se escogerá la creada anteriormente, *Policy_Raspberry*.

Por último, se descargarán los certificados generados automáticamente. Esto es muy importante ya que es la única vez que se pueden descargar.



Figura 78. Certificados y claves del objeto Raspberry. Fuente: Elaboración propia.

AWS IoT Device SDK v2 para Python

Ahora al dispositivo virtual creado anteriormente es necesario asociarle su dispositivo real. Esto se realizará mediante el software cliente, SDK, que es una interfaz de línea de comandos que permite acceder a las bibliotecas de cliente de AWS.

Para instalar SDK en la Raspberry se ejecutan las siguientes líneas de comando en el terminal de esta:

```
pip3 install awsiotsdk
```

Definición de Gateway RAK7268 con AWS IoT Core for LoRaWAN

Para añadir la puerta de enlace a AWS mediante la consola hay que ir a IoT Core y acceder al recurso Dispositivos LPWAN, donde se agregará el Gateway.

El siguiente paso es proporcionar todos los detalles necesarios de la puerta de enlace e información sobre la banda de frecuencias. Entre esto se encuentra el EUI del Gateway, el nombre del dispositivo y la banda de frecuencia.

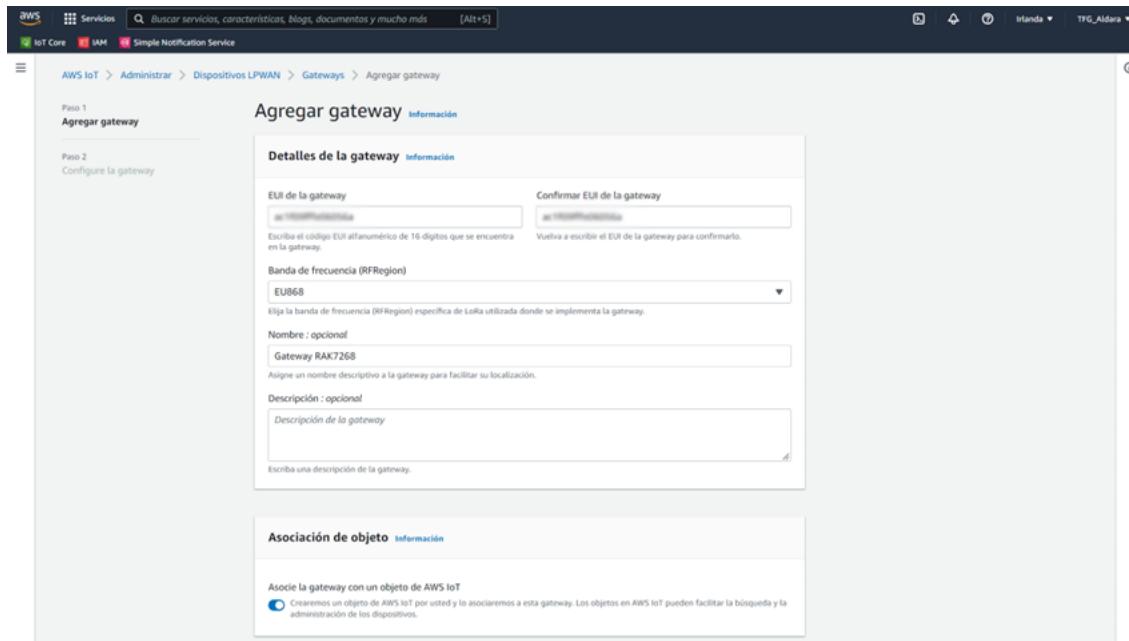


Figura 79. Definición del Gateway en AWS. Fuente: Elaboración propia.

El siguiente paso es crear y descargar todos los certificados del Gateway para que se comunique con la plataforma AWS de forma segura y autenticada, que consta de una clave privada y un certificado creado mediante el estándar X.509.

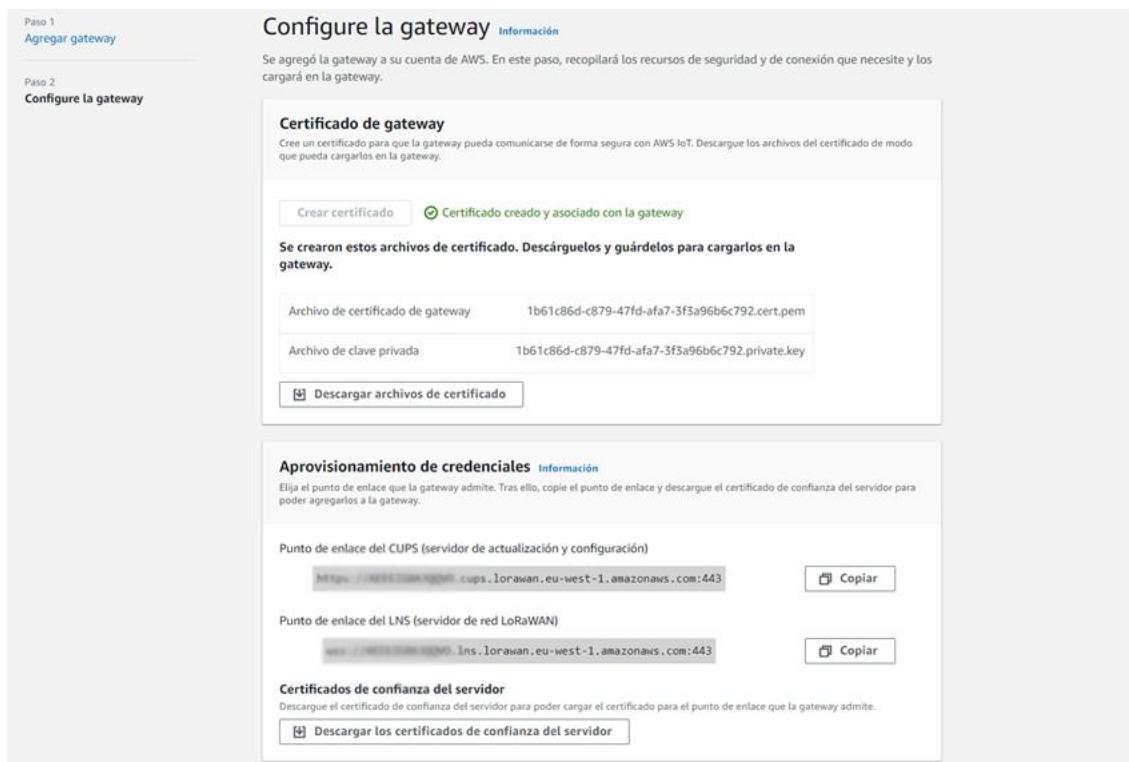


Figura 80. Certificados del Gateway. Fuente: Elaboración propia.

También es necesario crear el rol de IAM que permite al servidor de configuración y actualización (CUPS) administrar las credenciales de la puerta de enlace. Para ello se selecciona crear un rol, *IoTWirelessGatewayCertManagerRole*.

Una vez que está el Gateway añadido a AWS, se irá a la propia interfaz de RAK. En el apartado LoRa Network se cubrirá lo referente a URI, Port, trust, certificate y key de la siguiente manera:

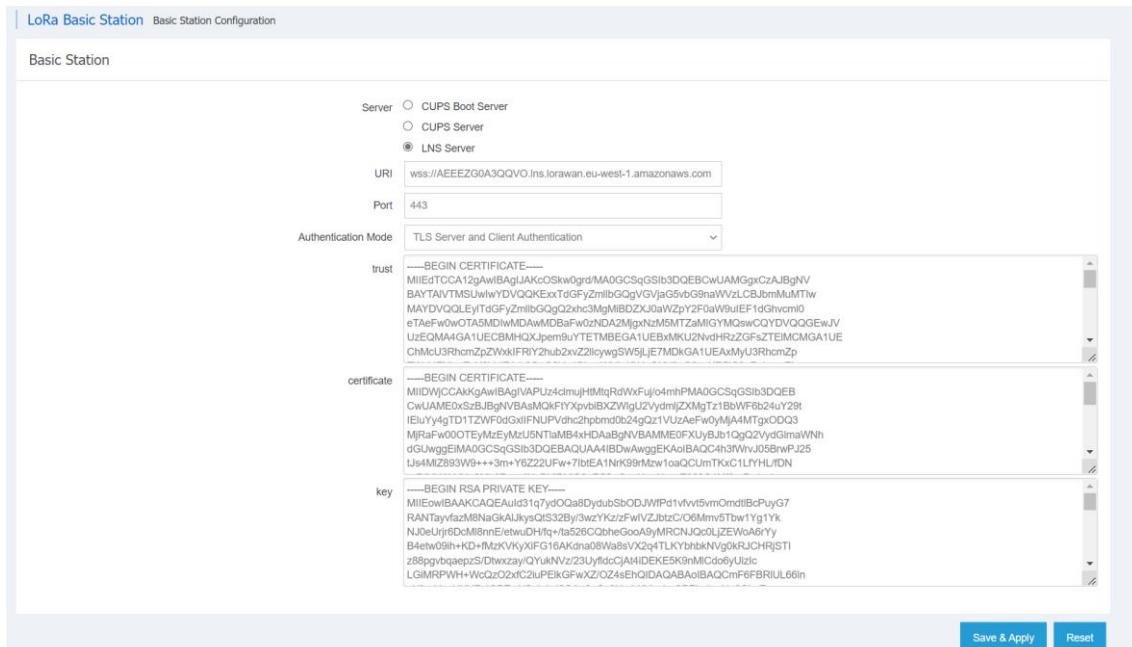


Figura 81. Configuración LoRa Basic Station del Gateway. Fuente: Elaboración propia.

Definición de dispositivos con AWS IoT Core for LoRaWAN

En este apartado se explicará el procedimiento para registrar un dispositivo LoRa mediante IoT Core for LoRaWAN. Estos pasos, explicados sobre el dispositivo Milesight EM500-CO2, son los mismos que se han llevado a cabo en los 2 nodos de sensorización restantes.

En primer lugar, se crea un rol de IAM, *IoTWirelessDestinationRole*. Este rol será utilizado por todos los dispositivos LoRa. La entidad de confianza seleccionada es la propia cuenta de AWS que se está utilizando, por esta razón se marca la opción cuenta de AWS y el ID correspondiente a la misma.

En el paso siguiente se le asignan los permisos al rol de IAM. En este caso, se creará una nueva política, *IoTWirelessDestinationPolicy*, destinada a este intercambio de mensajes entre dispositivos LoRaWAN, el Gateway y el recurso AWS IoT Core for LoRaWAN. El código de la política concreta es el siguiente.

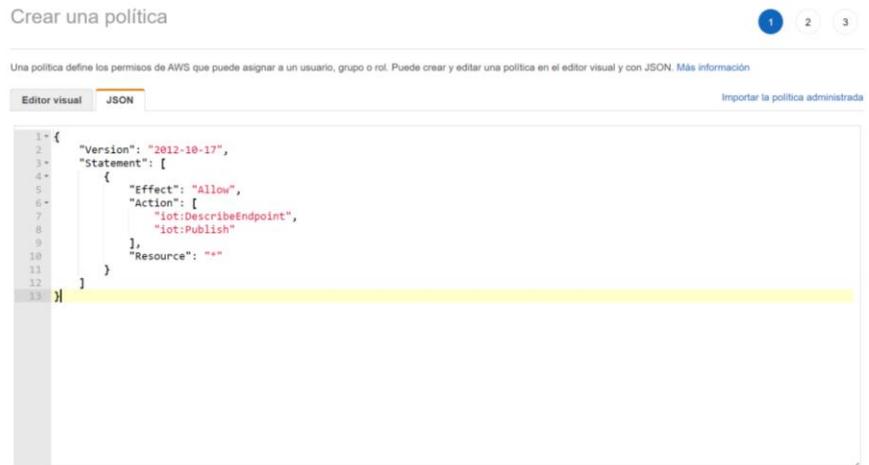


Figura 82. Política *IoTWirelessDestinationPolicy*. Fuente: Elaboración propia.

Por último, se hace un cambio en el código de relaciones de confianza del rol, para esto se accede al resumen de este y se cambia la propiedad como se muestra a continuación.

The screenshot shows the AWS IAM 'Resumen del rol' (Role summary) page for the role 'IoTWirelessDestinationRole'. In the 'Relaciones de confianza' (Trust relationships) section, there is a single JSON document:

```
1 {
2     "Version": "2012-10-17",
3     "Statement": [
4         {
5             "Effect": "Allow",
6             "Principal": {
7                 "Service": "iotwireless.amazonaws.com"
8             },
9             "Action": "sts:AssumeRole",
10            "Condition": {}
11        }
12    ]
13 }
```

Figura 83. Relaciones de confianza de *IoTWirelessDestinationRole*. Fuente: Elaboración propia.

El dispositivo envía los mensajes a la plataforma de manera codificada, por lo que para poder procesarlos es necesario crear una función Lambda.

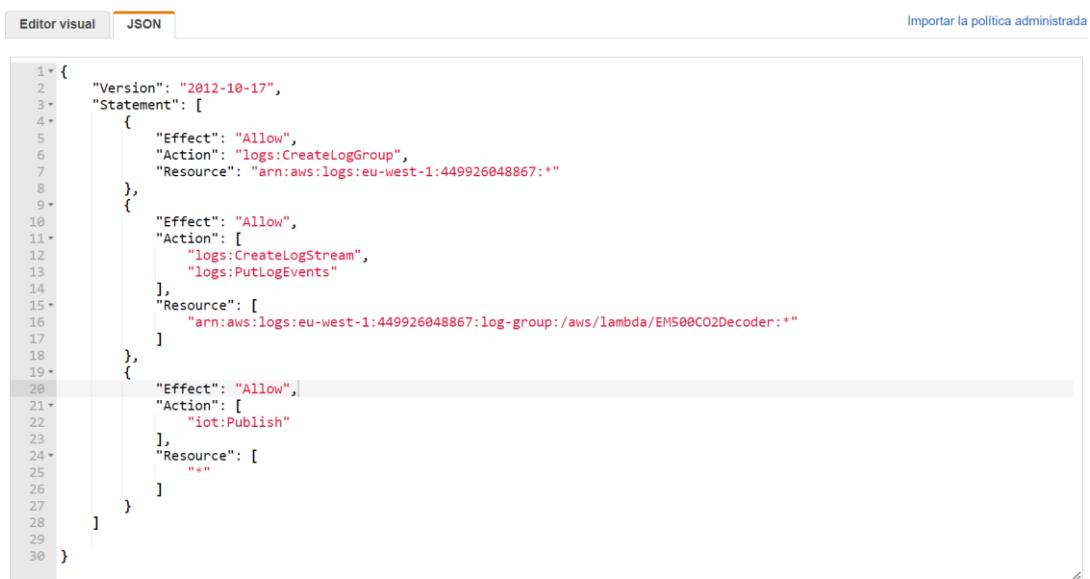
En la configuración de la función que se crea se seleccionará crear desde cero y con un tiempo de ejecución de:

- Node.js 12.x. si el dispositivo es Milesight EM500-CO2 o Dragino LSPH01.
- Python 3.8 si el dispositivo es RAK1903.

Una vez creada la función se hace clic en la misma y se introducen las líneas de código correspondientes al sensor se esté agregando.

Una vez implementado se accede a la configuración de la función y a permisos. Allí se selecciona el rol de ejecución y se modifica el código JSON hasta quedar de la siguiente manera.

Una política define los permisos de AWS que puede asignar a un usuario, grupo o rol. Puede crear y editar una política en el editor visual y con JSON. [Más información](#)

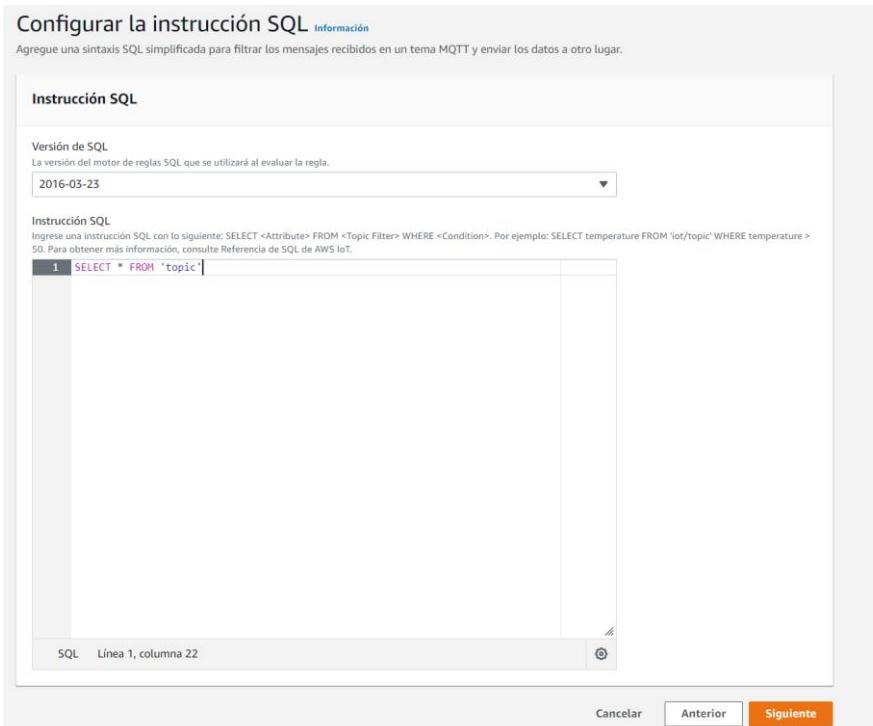


The screenshot shows the AWS Lambda function execution role policy editor. It has two tabs at the top: "Editor visual" (which is selected) and "JSON". There is also a link "Importar la política administrada". The main area contains a large block of JSON code representing a policy:

```
1  {
2    "Version": "2012-10-17",
3    "Statement": [
4      {
5        "Effect": "Allow",
6        "Action": "logs:CreateLogGroup",
7        "Resource": "arn:aws:logs:eu-west-1:449926048867:log-group:/aws/lambda/EM500CO2Decoder:*"
8      },
9      {
10        "Effect": "Allow",
11        "Action": [
12          "logs:CreateLogStream",
13          "logs:PutLogEvents"
14        ],
15        "Resource": [
16          "arn:aws:logs:eu-west-1:449926048867:log-group:/aws/lambda/EM500CO2Decoder:*"
17        ]
18      },
19      {
20        "Effect": "Allow",
21        "Action": [
22          "iot:Publish"
23        ],
24        "Resource": [
25          "*"
26        ]
27      }
28    ]
29  }
```

Figura 84. Rol de ejecución de la función Lambda. Fuente: Elaboración propia.

Es hora de hacer las configuraciones necesarias para que el resultado se publique automáticamente en el tema seleccionado cada vez que el dispositivo envía un mensaje. Para ello es necesario crear una regla, que posteriormente será asociada al destino del dispositivo, que sea la encargada de reenviar la carga útil del mismo.



The screenshot shows the AWS IoT Rule configuration interface. At the top, it says "Configurar la instrucción SQL" with a "Información" link. Below that, a note says "Agregue una sintaxis SQL simplificada para filtrar los mensajes recibidos en un tema MQTT y enviar los datos a otro lugar." The main area is titled "Instrucción SQL". It has a "Versión de SQL" section with a dropdown menu set to "2016-03-23". Below that is an "Instrucción SQL" section with a note about the syntax. A text input field contains the SQL query "SELECT * FROM 'topic'". At the bottom, there are buttons for "Cancelar", "Anterior", and "Siguiente".

Figura 85. Instrucción SQL para la regla. Fuente: Elaboración propia.

Ahora se dirá que acciones realizará la regla creada cuando llegue un mensaje entrante. En este caso la acción 1 será reenviar un mensaje en un tema de AWS IoT, y el rol que concede acceso al punto de conexión *IoTWirelessRepublishRule*. La acción 2 será invocar a la función Lambda creada para descodificar los mensajes enviados.

Una vez esto listo se agregar el destino y se introduce el nombre de la regla creada y el rol de servicio *IoTWirelessDestinationRole*.

The screenshot shows the 'Agregar destino' (Add Destination) interface in AWS IoT Core. The 'Detalles del destino' (Destination Details) section includes a 'Nombre de destino' (Destination Name) field containing 'EM500', a 'Descripción del destino' (Destination Description) field containing 'DecoderEM500', and a 'Copiar' (Copy) button. Below these are two configuration options:

- Escriba un nombre de regla**
Escriba el nombre de la regla o una regla/tema que procesará los mensajes enviados a este destino.
- Publicar en el agente de mensajes de AWS IoT Core**
Si necesita un agente de publicación/suscripción para distribuir mensajes a varios suscriptores

At the bottom left is a 'Avanzado' (Advanced) link.

Figura 86. Creación de un destino. Fuente: Elaboración propia.

El siguiente paso es crear un perfil para el dispositivo, con la ventaja de que todos los nodos de sensorización trabajan con en la misma banda de frecuencia y utilizan la misma versión MAC, por lo que será utilizado por el resto de los dispositivos.

La banda de frecuencia es EU868 ya que se trabaja en Europa, y la clase del dispositivo es clase A, como se muestra en las características de los mismos.

Agregar perfil de dispositivo

Perfil del dispositivo [Información](#)

Describa las capacidades del dispositivo y los parámetros de arranque que necesita el servidor de red para configurar el servicio de acceso de radio de LoRaWAN.

Seleccionar un perfil predeterminado y personalizar: *opcional*

Los perfiles predeterminados se basan en la clase de dispositivo LoRaWAN OTAA seleccionada y en la banda de radiofrecuencia LoRaWAN. Es posible que tenga que personalizar el perfil según las especificaciones del proveedor de dispositivos.

EU868 - A ▾

Nombre del perfil de dispositivo
Escriba un nombre descriptivo para este perfil de dispositivo.
EU868-A-OTAA

Banda de frecuencia (RFRegion)
Elija la banda de frecuencia compatible con LoRa para este perfil.
EU868 ▾

Versión de MAC
La versión de MAC de los dispositivos LoRaWAN que utilizan este perfil.
1.0.3 ▾

Versión de parámetros regionales
Seleccione el identificador de versión de los parámetros de región para este perfil.
RP002-1.0.1 (recomendado) ▾

MaxEIRP
Escriba el valor de MaxEIRP para este perfil de dispositivo.
5

Admite clase B
Elija esta opción para escribir los valores de compatibilidad con clase B.

Admite clase C
Elija esta opción para escribir los valores de compatibilidad con clase C.

Admite Join
Elija esta opción para escribir los valores para admitir Join (OTAA) o no (ABP).

▶ Configuración opcional

Figura 87. Creación de un perfil de dispositivo. Fuente: Elaboración propia.

Ahora se crea un perfil de servicio, que de igual manera que en el perfil de dispositivo será común para los tres nodos de sensorización LoRa.

Agregar perfil de servicio

Perfil del servicio [Información](#)

El perfil de servicio describe las características que están habilitadas para los usuarios y el índice de mensajes que pueden enviarse a través de la red.

Nombre de perfil de servicio: *opcional*
Escriba un nombre de perfil descriptivo.

AddGWMetaData
Agregue metadatos de gateway adicionales (RSSI, SNR, geolocalización de GW, etc.) a los paquetes enviados por los dispositivos.

Etiquetas: *opcional*
Una etiqueta es una marca que se asigna a un recurso de AWS. Cada etiqueta consta de una clave y un valor opcional. Puede utilizar etiquetas para buscar y filtrar los recursos o realizar un seguimiento de los costos de AWS.

Clave	Valor - <i>opcional</i>
<input type="text" value="Clave"/>	<input type="text" value="Valor"/>

Agregar nueva etiqueta
Puede agregar hasta 49 etiquetas.

Figura 88. Creación de un perfil de servicio. Fuente: Elaboración propia.

Una vez creado todo lo anterior, se accede a dispositivos para añadir finalmente el dispositivo LoRaWAN. En este punto es necesario proporcionar el DevEUI, AppKey y AppEui del dispositivo.

Agregar dispositivo

Especificación de LoRaWAN y configuración de dispositivo inalámbrico [Información](#)

Especificación de dispositivo inalámbrico
Las especificaciones del dispositivo se componen de la versión LoRaWAN (1.1 o 1.0.x) y el proceso de autenticación (autenticación Over The Air o autenticación por personalización). Una vez seleccionados, los datos se cifran con una clave que AWS posee y administra por usted.

OTAA v1.0.x

DevEUI
Valor de DevEUI hexadecimal de 16 dígitos que se encuentra en su dispositivo inalámbrico.

Confirmar DevEUI
Vuelva a escribir el DevEUI.

AppKey
Valor de AppKey hexadecimal de 32 dígitos proporcionado por el proveedor de dispositivos inalámbricos.

Confirmar AppKey
Vuelva a escribir la AppKey.

AppEUI
AppEUI hexadecimal de 16 dígitos proporcionado por el proveedor de dispositivos inalámbricos.

Confirmar AppEUI
Vuelva a escribir el AppEUI.

Nombre del dispositivo inalámbrico: *opcional*

Un nombre descriptivo para facilitar la localización del dispositivo inalámbrico.

Descripción del dispositivo inalámbrico: *opcional*

Una descripción útil para el dispositivo inalámbrico.

Figura 89. Agregar el dispositivo EM500-CO2 a AWS. Fuente: Elaboración propia.