



Trabajo de Fin de Grado
Grado en Ingeniería Informática

**Sistema para la gestión y control de sensor ITH en
explotaciones ganaderas**
**System for the management and control of THI sensors
in livestock farms**

Manual de Usuario (Tipología B)

Autor: Javier Romero Ramos

Directores: Ezequiel Herruzo Gómez
Miguel Ángel Montijano Vizcaíno



NIVERSIDAD DE CÓRDOBA

Índice general

1	Introducción	6
2	Requisitos técnicos del sistema	7
2.1	Sistema de adquisición de datos	7
2.2	Sistema de recolección y envío de datos	7
2.3	Sistema de gestión de datos	8
2.4	Sistema de almacenamiento de datos	8
3	Requisitos finales del sistema	9
3.1	Sistema de adquisición de datos	9
3.2	Sistema de recolección y envío de datos	9
3.3	Sistema de gestión de datos y sistema de almacenamiento de datos . . .	10
3.4	Módulo de accionamiento de dispositivo conectado	10
3.5	Aplicación móvil y notificaciones	10
4	Instalación y configuración del sistema	11
4.1	Sistema de adquisición de datos	11
4.1.1	Conexionado	11
4.1.2	Descarga del código y del entorno de programación	11
4.1.3	Configuración del entorno de programación y programación del Arduino	15
4.1.4	Configurar y habilitar la puerta de salida	15
4.2	Sistema de recepción y envío de datos	23
4.2.1	Creación de la aplicación	24
4.2.2	Registro de dispositivo en la aplicación	24
4.2.3	Configurar dispositivo en la aplicación	30
4.2.4	El código de Arduino	31
4.2.5	Integración de Aplicación con sistema de adquisición de datos .	33
4.3	Sistema de gestión de datos	37
4.3.1	Preparación del entorno	37
4.3.1.1	Instalación de Node.js	37
4.3.1.2	Creación del proyecto Node.js	39
4.3.1.3	Instalación de dependencias	39
4.3.1.4	Desarrollo del servidor	40
4.3.1.5	Despliegue del servidor en Railway (con integración GitHub)	41
4.3.2	Configuración y uso de API Key en TTN	46
4.3.2.1	Creación de una API Key en TTN	46
4.4	Sistema de almacenamiento de datos	48



Índice general

4.4.1	Instalación de base de datos	48
4.4.1.1	Creación de MySQL en Railway	48
5	Desinstalación del sistema	50
5.1	Sistema de adquisición de datos	50
5.1.1	Eliminación de código de Arduino	50
5.1.2	Desinstalación de librerías de Arduino	50
5.1.3	Eliminación de archivos descargados	50
5.1.4	Configurar a valores de fábrica la puerta de enlace	53
5.2	Sistema de recepción y envío de datos	53
5.2.1	Eliminar dispositivo en The Things Network	53
5.2.2	Eliminar aplicación en The Things Network	54
5.3	Sistema de gestión de datos	54
5.3.1	Detener el servicio en Railway	55
5.4	Eliminar archivos de código de Node.js	57
5.5	Sistema de almacenamiento de datos	57
5.5.1	Eliminar esquema completo de MySQL	57
5.5.2	Eliminar el servicio MySQL en Railway	57
6	Instalación y configuración de la aplicación móvil	60
6.1	Instalación a través de Google Play Store	60
6.2	Instalación manual mediante archivo APK	60
6.3	Configuración inicial	61
7	Descripción y uso de la aplicación móvil	62
7.1	Pantalla de bienvenida	62
7.2	Inicio de sesión	63
7.3	Registro de usuario	65
7.4	Pantalla principal	67
7.5	Alertas y notificaciones	69
7.6	Generación de informes en PDF	71
7.7	Menú desplegable de navegación	74
7.8	Pantalla de administración de perfil	76
7.9	Pantalla de preferencias	77
7.10	Pantalla de edición de sensor	80
8	Desinstalación de la aplicación	84

Índice de figuras

4.1	Foto del conexionado del sistema de adquisición de datos, hecho con la herramienta Fritzing	12
4.2	Foto del conexionado del sistema de adquisición de datos	13
4.3	Descarga de Arduino IDE	13
4.4	Pasos para abrir el proyecto en Arduino IDE.	14
4.5	Pasos para instalar librerías en Arduino IDE.	16
4.6	Pasos para ejecutar programa en Arduino IDE.	17
4.7	Etiqueta del The Things Indoor Gateway	17
4.8	Registro en The Things Network.	18
4.9	Acceder a The Things Network.	19
4.10	Registrar gateway en The Things Network.	21
4.11	Botón reset gateway	22
4.12	Botón setup gateway	22
4.13	Configuración completa de red Wi-fi con gateway	23
4.14	Crear aplicación en The Things Network.	24
4.15	Registrar dispositivo en aplicación en The Things Network.	26
4.16	Datos de formulario para registrar dispositivo en The Things Network .	27
4.17	Ejemplo para obtener JoinEUI	28
4.18	Monitor serie para obtener EUI	29
4.19	Datos completos en formulario de dispositivo en The Things Network .	29
4.20	Listado de dispositivos en The Things Network	30
4.21	Acceso a Uplink en dispositivo registrado en The Things Network . .	30
4.22	Selección de “Uplink“ en dispositivo registrado en The Things Network	31
4.23	Código en JavaScript en el dispositivo en The Things Network	31
4.24	Código Arduino de appEui y appKey	32
4.25	Salida de datos en monitor serie en Arduino	33
4.26	Datos en vivo en The Things Network	33
4.27	Selección de webhook en The Things Network	34
4.28	Creación de webhook en The Things Network.	35
4.29	Relleno de datos del formulario al crear webhook en The Things Network	36
4.30	Registro y guardado de webhook en The Things Network.	36
4.31	Instalación y descarga de Node.js.	38
4.32	Versión instalada de Node.js	39
4.33	Conjunto de archivos generados para la creación del proyecto	39
4.34	Cabecera del archivo por parte del servidor de paquetes necesarios .	41
4.35	Fragmento de código encargado de insertar mediciones en MySQL . .	41
4.36	Repositorio del Node.js en Github	42
4.37	Creación de proyecto en Railway	43



Índice de figuras

4.38 Creación de servicio MySQL en Railway	44
4.39 Generación de dominio público en Railway	45
4.40 Logs de registro de respuesta de HTTP en Railway	45
4.41 Logs de registro de peticiones de datos recibidos en Railway	46
4.42 Acceder a The Things Network.	47
4.43 Lista de API Keys creadas en TTN	48
4.44 Servicio MySQL provisionado en Railway y credenciales de conexión.	49
5.1 Conexión de microcontrolador Arduino al ordenador	51
5.2 Eliminación del código de Arduino	52
5.3 Eliminación de librerías de Arduino	53
5.4 Eliminación del dispositivo en The Things Network	54
5.5 Eliminación de la aplicación en The Things Network	54
5.6 Eliminación del servicio de Node.js en Railway	56
5.7 Eliminación del esquema completo de MySQL en Railway	58
5.8 Eliminación del servicio MySQL en Railway	59
7.1 Pantalla de bienvenida de app móvil	63
7.2 Pantalla de inicio de sesión de app móvil	65
7.3 Pantalla de registro de usuario de app móvil	67
7.4 Pantalla principal de app móvil	69
7.5 Pantalla de notificación de alerta	71
7.6 Pantalla de selección de “Abrir con“ el PDF generado	73
7.7 Pantalla de PDF generado	74
7.8 Pantalla principal seleccionado el menú desplegable	75
7.9 Pantalla de Administrar perfil	77
7.10 Pantalla de Preferencias (1/2)	79
7.11 Pantalla de Preferencias (2/2)	80
7.12 Editar sensor: información del dispositivo y ubicación.	82
7.13 Editar sensor: modo de operación y ajuste del umbral ITH.	83
8.1 Desinstalación de la aplicación: Paso 1	84
8.2 Desinstalación de la aplicación: Paso 2	85
8.3 Desinstalación de la aplicación: Paso 3	86
8.4 Desinstalación de la aplicación: Paso 4	87

CAPÍTULO 1

Introducción

En este documento se proporciona una explicación clara y sencilla, junto con los pasos a seguir, para que el usuario técnico pueda poner en marcha el sistema desarrollado, permitiendo al usuario final usarlo con correcto funcionamiento, siguiendo todos los pasos y ayudas necesarias escritas en este manual.

Además, tanto el usuario técnico como el usuario final podrán conocer los requisitos mínimos necesarios para el funcionamiento del sistema, así como los pasos para su instalación, configuración, utilización y apagado, incluyendo también las indicaciones necesarias para su posible desinstalación.

El proyecto se compone de varios subsistemas que garantizan su correcto funcionamiento continuo. La documentación está estructurada de forma clara y detallada, permitiendo que cualquier usuario pueda desempeñar su papel en el uso o mantenimiento del sistema sin dificultades adicionales.

CAPÍTULO 2

Requisitos técnicos del sistema

En este capítulo se detallan los requisitos técnicos que el sistema debe tener para poder ser instalado y ejecutado de forma correcta.

2.1. Sistema de adquisición de datos

- Módulo DHT11 para medir temperatura y humedad mediante salida digital calibrada.
- Arduino MKR WAN 1310 con radio LoRa/LoRaWAN para leer el DHT11 y enviar datos por LoRaWAN.
- Arduino IDE versión 2.3.6.
- Librería DHT (2.1.0) y librería MKRWAN (1.1.1).
- The Things Indoor Gateway conectado a Internet para entregar los paquetes a TTN.
- Accionamiento de ventilador:
 - Ventilador DC 5 V (modelo de referencia: JZK-79255) con conector de 2 pines.
 - Transistor NPN BC337 en configuración *low-side* como interruptor.
 - Diodo 1N4004 en antiparalelo con el ventilador (rueda libre).
 - Resistencia de base 220Ω entre GPIO y base del BC337.
 - Fuente de 5 V adecuada para el ventilador y cableado/protoboard.

2.2. Sistema de recolección y envío de datos

- Cuenta y aplicación en The Things Network (TTN) con dispositivo dado de alta por OTAA.
- Configuración del *Payload Formatter* para obtener temperatura, humedad e ITH.



- Gateway LoRaWAN con salida a Internet (Ethernet/Wi-Fi/4G) y consola TTN accesible.
- Webhook de TTN apuntando al dominio público del backend en Railway.

2.3. Sistema de gestión de datos

- Backend en Node.js 18+ con Express y cliente MySQL `mysql2`.
- Despliegue en Railway con dominio público y variables de entorno configuradas (`MYSQLHOST`, `MYSQLUSER`, `MYSQLPASSWORD`, `MYSQLDATABASE`, `MYSQLPORT`, `PORT`).
- Endpoint de verificación `/health` y consulta de logs desde el panel de Railway.
- Visual Studio Code versión 1.92.1 para edición y depuración del código.

2.4. Sistema de almacenamiento de datos

- MySQL gestionada en Railway para persistencia de mediciones, sensores, granjas y usuarios.

CAPÍTULO 3

Requisitos finales del sistema

En esta sección se detallan los requisitos que el usuario final debe cumplir para que el sistema funcione correctamente en su instalación real. La arquitectura actual se apoya en TTN para la red LoRaWAN y en un backend y base de datos desplegados en Railway, por lo que no es necesario mantener servidores locales encendidos.

3.1. Sistema de adquisición de datos

Para que el nodo sensor pueda enviar lecturas con normalidad se requiere:

- Alimentación estable para el microcontrolador y el sensor (por ejemplo, 5 V mediante adaptador de corriente o batería).
- Presencia de una pasarela LoRaWAN cercana que reciba los datos del nodo.
- Conectividad a Internet en la pasarela (backhaul) por alguno de estos medios: Wi-Fi, Ethernet o datos móviles 4G/5G mediante punto de acceso del teléfono. No es imprescindible disponer de Wi-Fi fijo si hay cobertura móvil.
- Ubicación adecuada del nodo y del gateway para asegurar cobertura radio como priorizar zonas elevadas.

3.2. Sistema de recolección y envío de datos

Para el envío de lecturas hacia la nube:

- La aplicación y dispositivo deben estar dados de alta en The Things Network y vinculados al gateway utilizado.
- La función de decodificación en TTN debe estar activada para producir el payload con temperatura, humedad e ITH.
- El webhook de TTN debe apuntar al dominio público del backend en Railway y encontrarse operativo.



3.3. Sistema de gestión de datos y sistema de almacenamiento de datos

Este bloque se ejecuta íntegramente en la nube:

- El backend y la base de datos están alojados en Railway, puesto que el usuario no debe mantener equipos locales. Solo se requiere conexión a Internet para acceder a la API desde la app.
- El servicio debe encontrarse accesible y con credenciales configuradas.

3.4. Módulo de accionamiento de dispositivo conectado

Si se desea activar un ventilador desde la aplicación:

- Ventilador DC de 5 V con su fuente de alimentación adecuada.
- Etapa de conmutación con transistor NPN (por ejemplo, BC337) en configuración low-side, resistencia de base de $220\ \Omega$ y diodo 1N4004 en antiparalelo con la carga.
- Cableado correcto y disipación mínima según corriente del ventilador.

3.5. Aplicación móvil y notificaciones

Para el uso de la app en el dispositivo del usuario:

- Teléfono con acceso a Internet (Wi-Fi o datos móviles).
- Permitir notificaciones del sistema para recibir avisos cuando el ITH supere el umbral configurado.
- Credenciales de acceso a la aplicación y, en su caso, pertenencia a la granja/sensores asignados.

CAPÍTULO 4

Instalación y configuración del sistema

4.1. Sistema de adquisición de datos

4.1.1. Conexionado

Con el modulo DHT11 y el arduino MKR WAN 1310 adquiridos y todos los componentes necesarios para la activación del dispositivo ventilador de 5V DC, se conecta a la protoboard siguiendo el esquema de la Figura 4.1

Posteriormente, se muestra en la Figura 4.2, el conexionado de cables de todos los componentes.

Una vez conectado todos los componentes, se procede a la programación del proyecto.

4.1.2. Descarga del código y del entorno de programación

Para comenzar, es necesario obtener el código fuente del proyecto, el cual se encuentra disponible en el repositorio de GitHub. Además, se debe descargar el entorno de desarrollo Arduino IDE desde su página web oficial, como se muestra en laFigura 4.3.

Una vez descargado e instalado Arduino IDE, se debe ejecutar el programa. Para abrir el proyecto, hay que acceder al menú “Archivo” y seleccionar la opción “Abrir”. A continuación, se debe buscar la ubicación del proyecto y cargarlo. Si el proceso se ha realizado correctamente, el código del proyecto aparecerá visible en la interfaz de Arduino IDE, tal como se aprecia en la Figura 4.4.

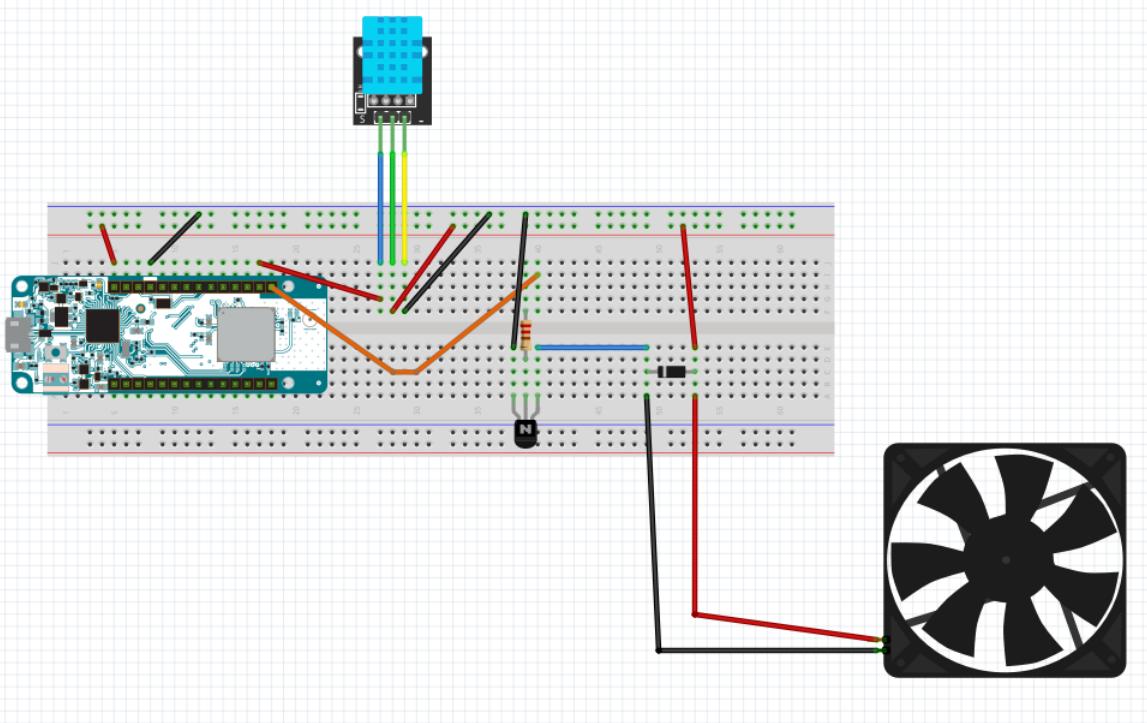


Figura 4.1: Foto del conexionado del sistema de adquisición de datos, hecho con la herramienta Fritzing

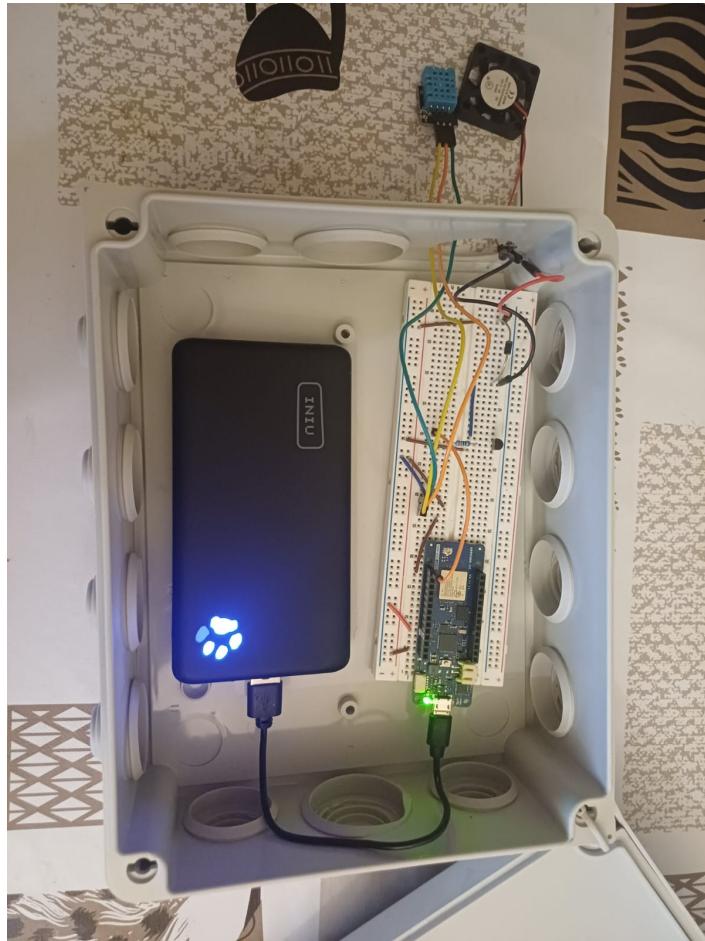


Figura 4.2: Foto del conexionado del sistema de adquisición de datos

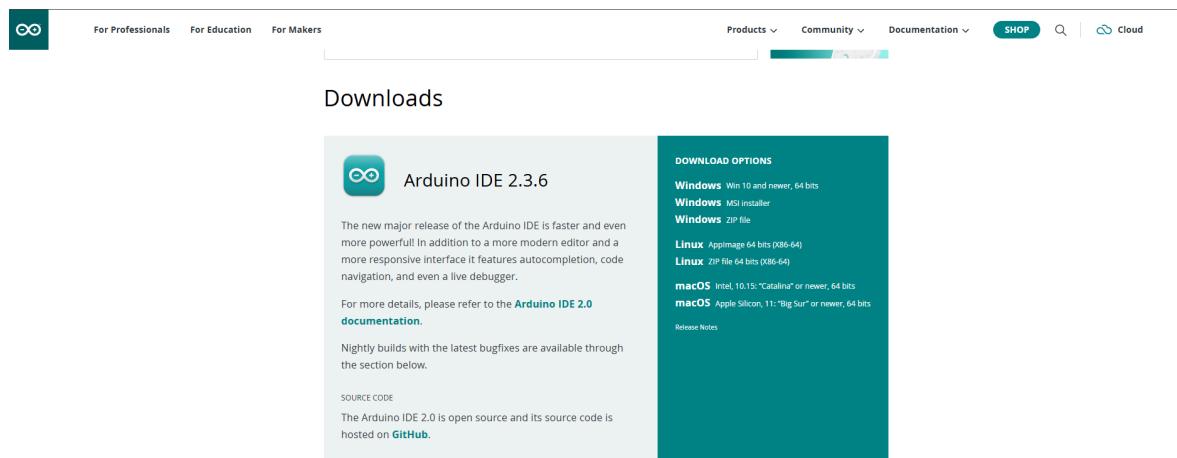


Figura 4.3: Descarga de Arduino IDE



Capítulo 4. Instalación y configuración del sistema

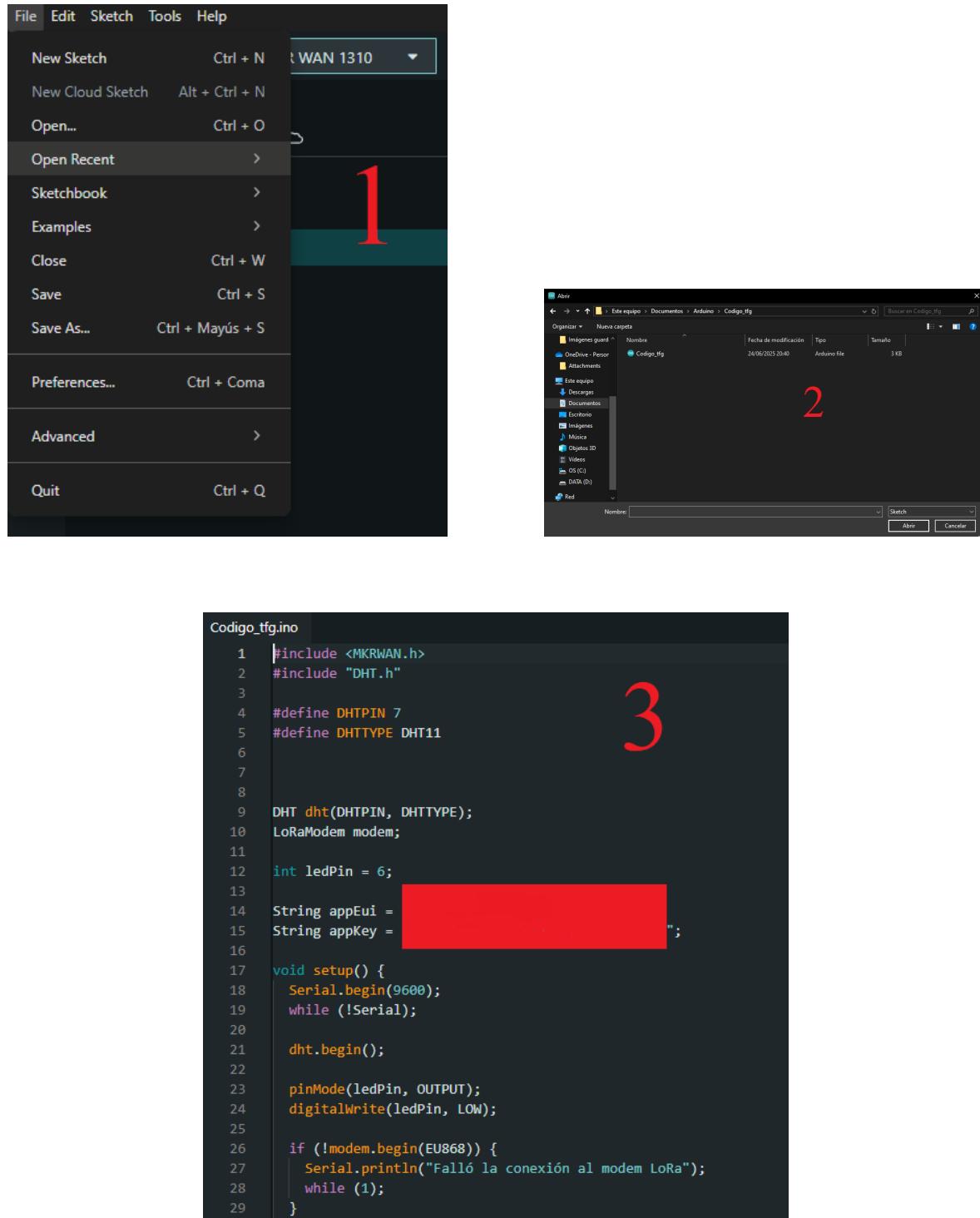


Figura 4.4: Pasos para abrir el proyecto en Arduino IDE.



4.1.3. Configuración del entorno de programación y programación del Arduino

Antes de comenzar la programación del dispositivo, es necesario instalar las librerías que utilizará el sistema. Para ello, en el entorno de desarrollo Arduino IDE, se debe acceder al gestor de librerías, identificado con un ícono de varios libros situado en el panel lateral izquierdo. Una vez dentro, se deben buscar las librerías necesarias escribiendo DHT11 y MKRWAN en la barra de búsqueda, y se procede a su instalación. Este proceso se muestra en la Figura 4.5.

A continuación, se debe conectar el microcontrolador Arduino MKR WAN 1310 al ordenador utilizando un cable micro USB.

Una vez conectado, es importante seleccionar correctamente la placa desde el menú superior del Arduino IDE. Finalmente, se debe pulsar el botón de carga para cargar el código al dispositivo. El proceso de programación puede tardar unos segundos, como se aprecia en la Figura 4.6.

Una vez, hecho todo esto se puede comprobar el funcionamiento y visualizar los mensajes que genera el programa de Arduino, donde se abre el Monitor Serie, que se accede desde la esquina superior derecha del entorno de Arduino, con el ícono de una lupa.

4.1.4. Configurar y habilitar la puerta de salida

Para configurar correctamente la puerta de enlace LoRaWAN, es necesario seguir las indicaciones proporcionadas en el manual de referencia. A continuación, se describen los pasos que se han seguido durante el proceso de configuración en este proyecto. Es posible que el orden de los pasos pueda variar ligeramente, ya que la puerta de enlace utilizada fue previamente registrada en una cuenta distinta de The Things Network.

Como primer paso, se recomienda realizar una fotografía de la etiqueta que viene adherida al dispositivo, tal como se muestra en la Figura 4.7. En esta etiqueta se encuentran los datos esenciales que serán necesarios más adelante para completar la configuración de la puerta de enlace.

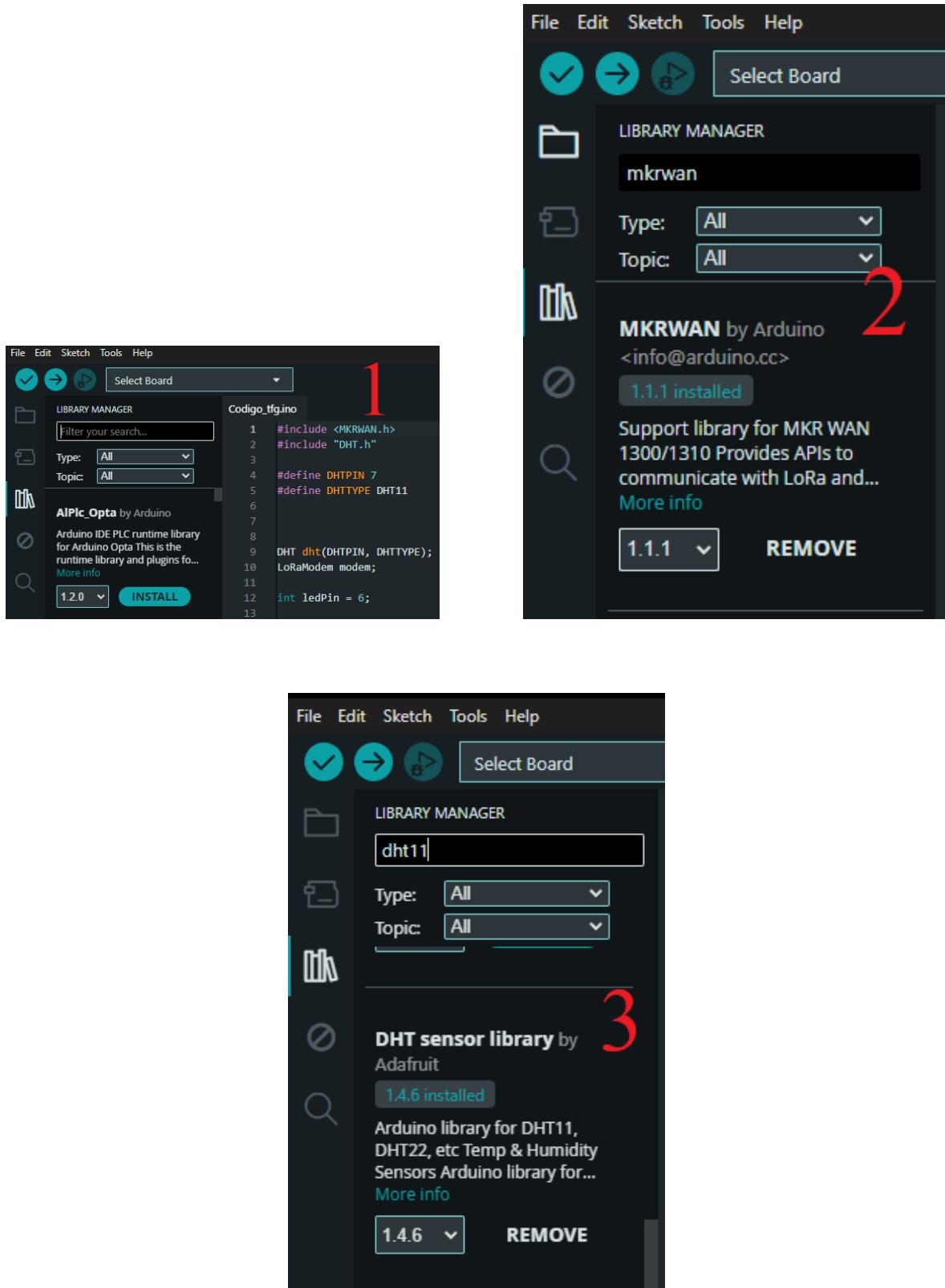


Figura 4.5: Pasos para instalar librerías en Arduino IDE.



Capítulo 4. Instalación y configuración del sistema

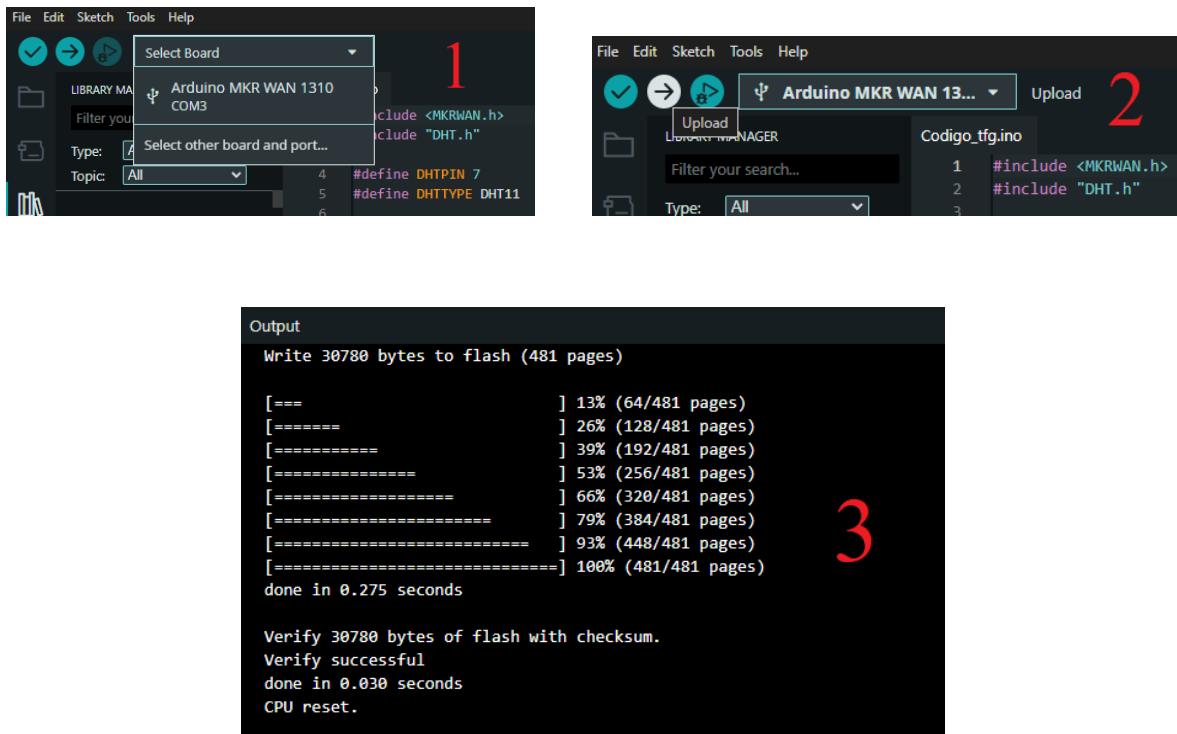


Figura 4.6: Pasos para ejecutar programa en Arduino IDE.



Figura 4.7: Etiqueta del The Things Indoor Gateway

El siguiente paso es crear una cuenta en la plataforma The Things Network (TTN). Para ello, se debe acceder a su página web oficial y seleccionar la opción “Sign up”. Posteriormente, se debe hacer clic en “Join The Things Network” y “Sign up for free”. A continuación, se completan los datos solicitados en el formulario de registro, tal como se muestra en la Figura 4.8.



Capítulo 4. Instalación y configuración del sistema

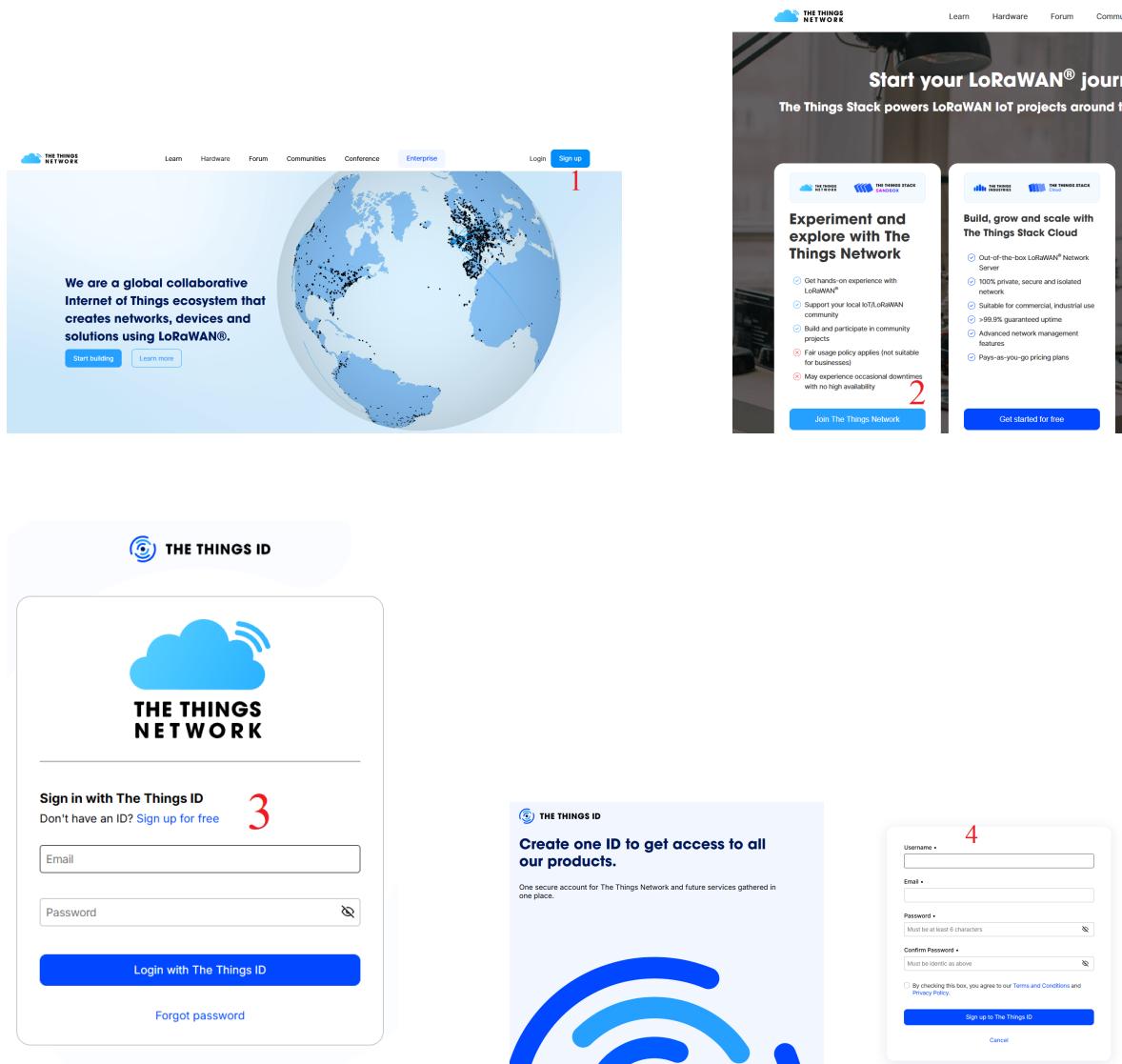


Figura 4.8: Registro en The Things Network.

Una vez creada la cuenta, es necesario iniciar sesión. Tras acceder, se debe pulsar sobre la opción “Console” y seleccionar el clúster deseado. Después, se continua con la sesión iniciada y se debe hacer clic en “Gateways”, como se visualiza en la Figura 4.9.

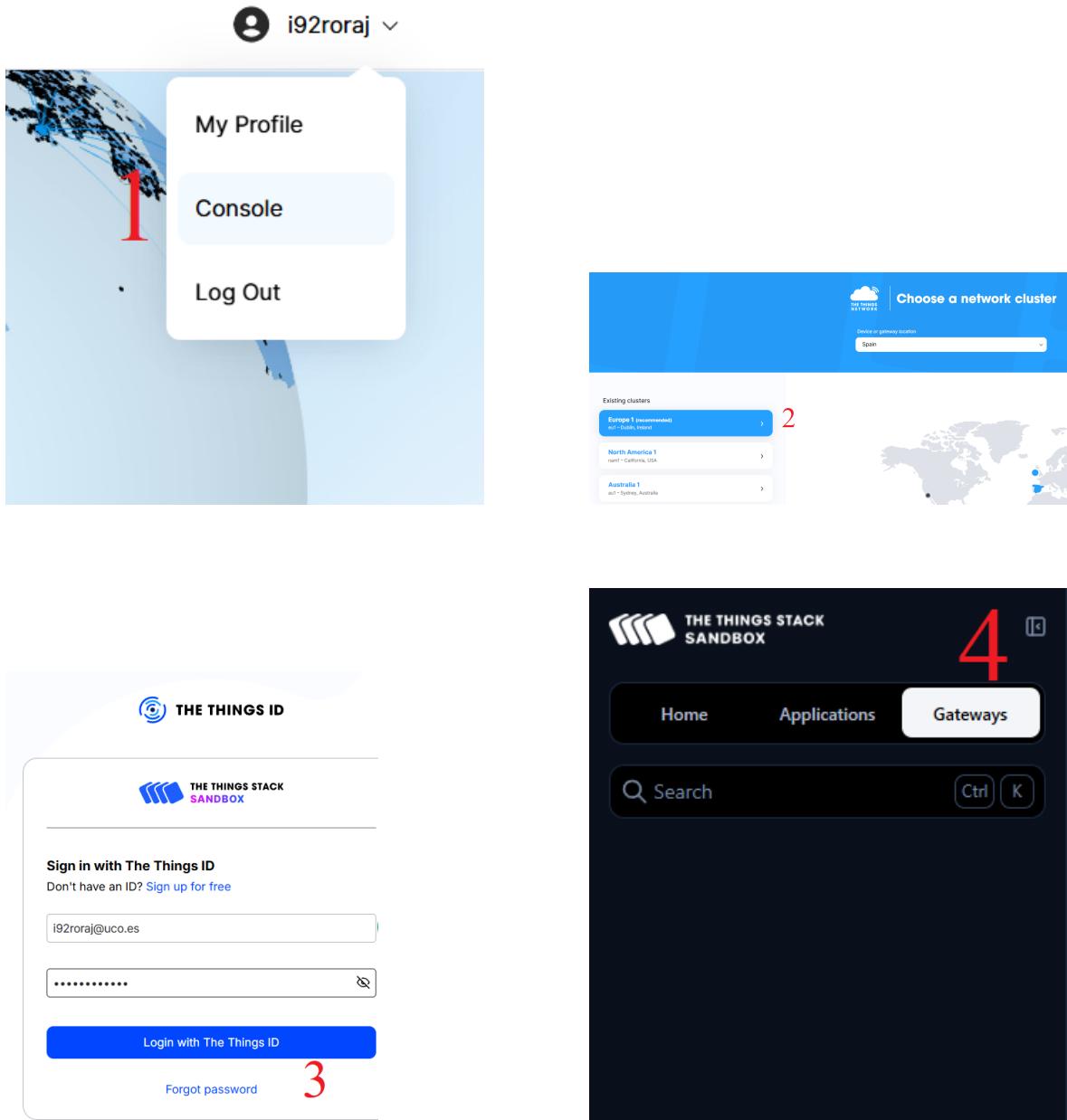


Figura 4.9: Acceder a The Things Network.

El siguiente paso consiste en registrar la puerta de enlace. Para ello, se debe seleccionar “Register gateway” y completar la información solicitada siguiendo las indicaciones del manual de configuración. Finalmente, se debe pulsar en “Claim gateway”. Si durante este proceso aparece un error como el que se muestra en la tercera subfigura de la Figura 4.10, significa que la puerta de enlace ya ha sido asociada a otra cuenta, por lo que no se podrá reclamar directamente desde la plataforma web.

En este caso, sería necesario utilizar el cliente oficial proporcionado por The Things Industries, según el manual de referencia. No obstante, en este proyecto este inconveniente no afecta, ya que la puerta de enlace funciona correctamente y ha sido cedida



Capítulo 4. Instalación y configuración del sistema

por la universidad, por lo que es probable que haya sido configurada previamente por otro alumno.



Capítulo 4. Instalación y configuración del sistema

The figure consists of three vertically stacked screenshots from the The Things Network web interface.

Screenshot 1: Shows the main 'Gateways' page with a single registered gateway. The gateway has a status of '1' and was created at the current time. It includes a 'Scan gateway QR code' button.

Screenshot 2: Shows the 'Register gateway' form. Fields include 'Gateway EUI' (58 A0 CB FF FE 80 4F E5), 'Owner token' (meypyC3r), 'Gateway ID' (eui-58a0cbffe804fe5), and 'Frequency plan' (Europe 863-870 MHz). A 'Claim gateway' button is present.

Screenshot 3: Shows the same 'Register gateway' form, but with an error message: 'Submit failed' and 'Gateway with EU 58A0CBFFE804FE5 already exists'. It also includes a 'Scan gateway QR code' button and a 'Gateway EUI' field (58 A0 CB FF FE 80 4F E5).

Figura 4.10: Registrar gateway en The Things Network.



A continuación, se debe configurar la conexión Wi-Fi de la puerta de enlace. Para ello, es necesario conectarla a la corriente eléctrica o al ordenador mediante un cable USB-C. Después, se debe mantener pulsado durante 5 segundos el botón “Reset”, situado en la parte inferior del dispositivo (ver Figura 4.11), hasta que el LED comience a parpadear rápidamente en verde y rojo.

Una vez que el LED muestre ese parpadeo, se debe mantener pulsado el botón “Setup” (ubicado en la parte superior del dispositivo, como se aprecia en la Figura 4.12) durante 10 segundos, hasta que el LED empiece a parpadear rápidamente en color rojo.



Figura 4.11: Botón reset gateway



Figura 4.12: Botón setup gateway

En ese momento, la puerta de enlace creará un punto de acceso Wi-Fi con el nombre



"MINIHUB-" seguido de los últimos seis dígitos del identificador EUI del dispositivo. Se debe conectar a esta red utilizando la contraseña que aparece en la etiqueta del dispositivo. Una vez conectados, hay que introducir en el navegador la dirección 192.168.4.1 para acceder a la página de configuración y acceder con la configuración de la conexión Wi-Fi. Todo este proceso se muestra en la Figura 4.13.

Tras finalizar este proceso, el siguiente paso será crear la aplicación correspondiente en The Things Network y registrar en ella los dispositivos que se van a utilizar en el proyecto.

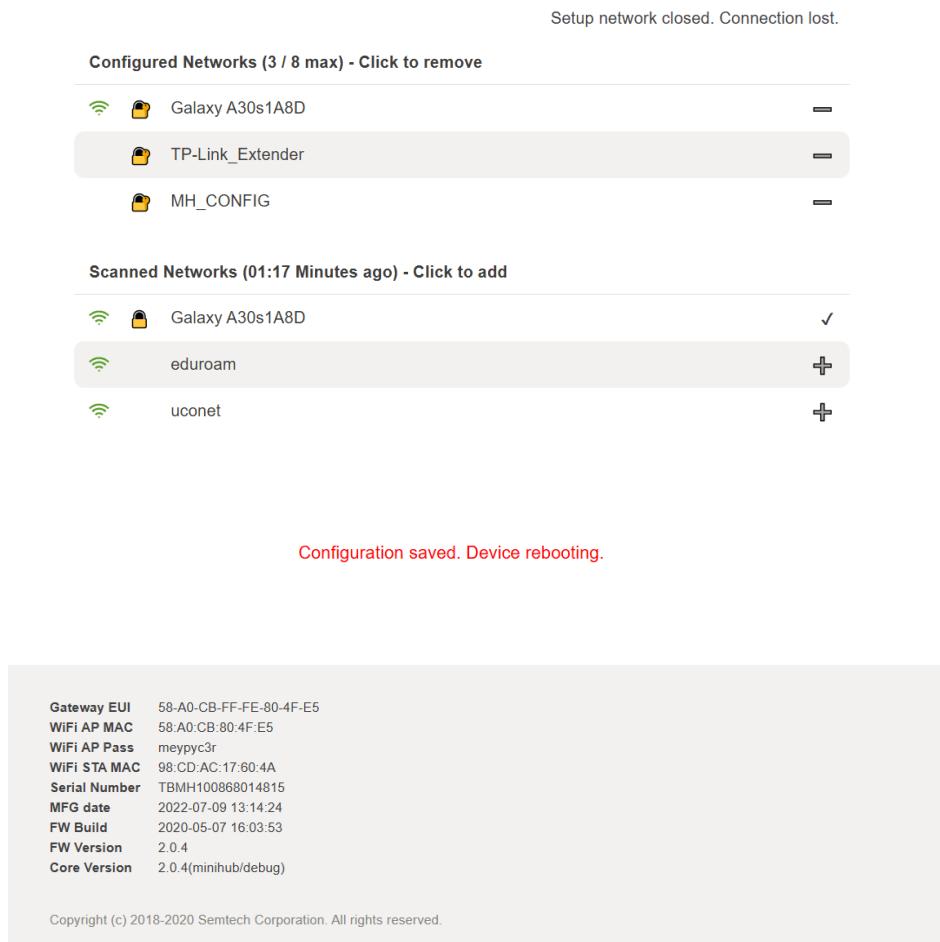


Figura 4.13: Configuración completa de red Wi-fi con gateway

4.2. Sistema de recepción y envío de datos

En esta sección, se detalla los pasos a seguir en la pagina de The Things Network y los pasos de la aplicación a crear.



4.2.1. Creación de la aplicación

Primero, accedemos a la pagina principal de The Things Network, en la consola de nuestra cuenta, después, en la pestaña de “Home“ se pulsa sobre “Create Application“, y después se rellena el formulario con los datos pedidos y se pulsa sobre crear aplicación, tal como se observa en la Figura 4.14.

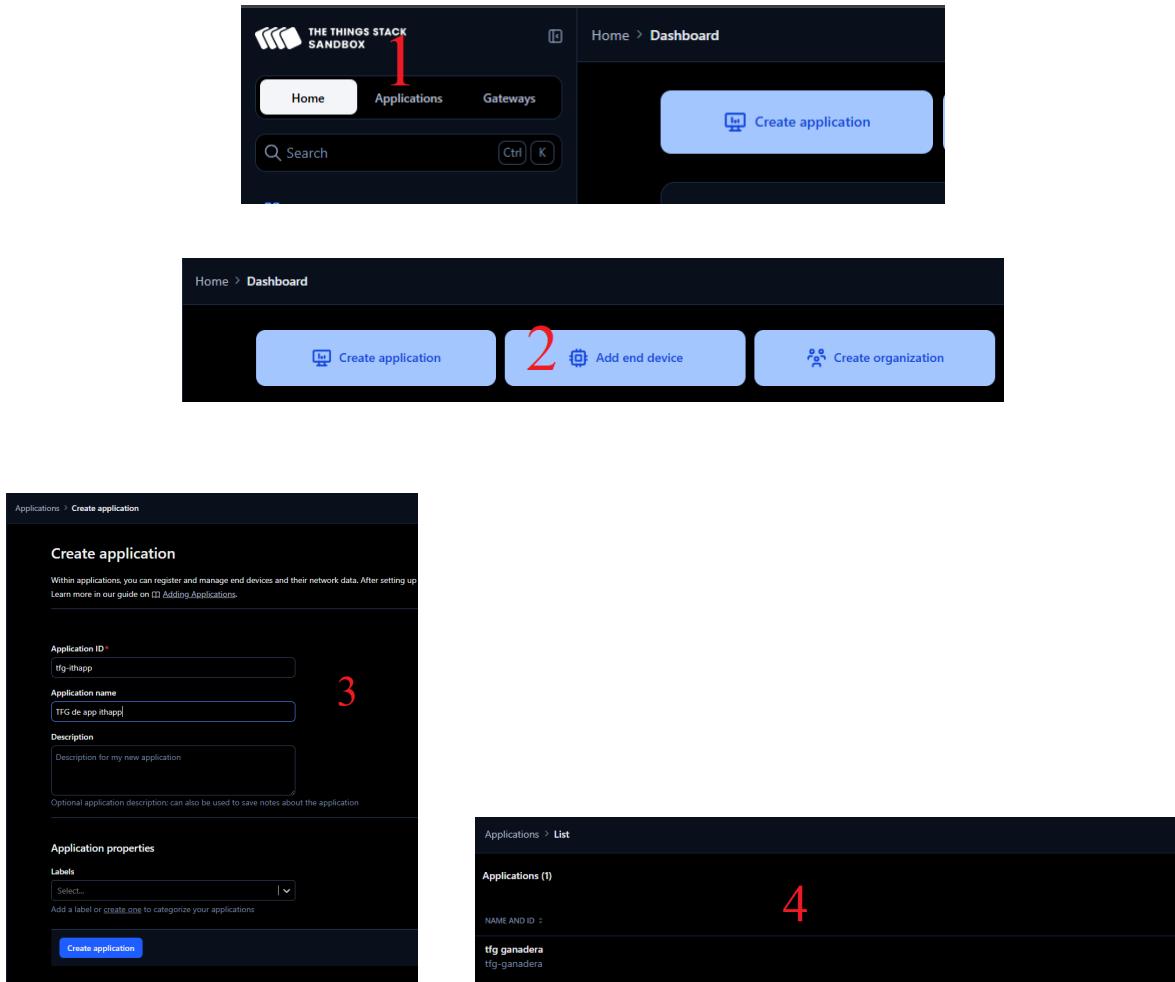


Figura 4.14: Crear aplicación en The Things Network.

4.2.2. Registro de dispositivo en la aplicación

Una vez que la aplicación ha sido creada en The Things Network, el siguiente paso es acceder a ella para iniciar su configuración. Dentro de la aplicación, se debe seleccionar la opción “End devices“ y posteriormente “Register end devices“. En esta sección, será necesario completar los datos solicitados en función del sistema de adquisición de datos que se está utilizando, tal como se muestra en la Figura 4.15.

Al llegar al apartado “Provisioning information“, es importante indicar en el campo



“JoinEUI” un valor personalizado que servirá como identificador de la aplicación dentro del código Arduino. Por otro lado, en “DevEUI” se debe introducir el identificador único del dispositivo, que se obtiene ejecutando un código específico desde el entorno Arduino IDE. La Figura 4.16 muestra los datos requeridos en este paso.

Para obtener el ”DevEUI” correspondiente, en Arduino IDE se debe ir a “File”, seleccionar “Examples”, luego “MKRWAN” y finalmente abrir el ejemplo llamado “First-Configuration”. Esto abrirá una nueva ventana con el código preparado para esta configuración, como se observa en la Figura 4.17. Una vez abierto, se debe cargar el código en el dispositivo pulsando el botón de programación, representado por una flecha hacia la derecha en el menú superior.

Después de la carga, en el Monitor Serie aparecerá el resultado de la ejecución, donde se mostrará el "DevEUI" buscado. Este valor se muestra en la línea donde se imprime el mensaje "Your device EUI is:", como se observa en la Figura 4.18.

Este identificador debe ser copiado y pegado en el formulario correspondiente dentro de The Things Network. A continuación, se deben completar los campos restantes con la información deseada, como se presenta en la Figura 4.19 . Para finalizar, se pulsa sobre “Register device” y, si todo es correcto, el nuevo dispositivo aparecerá listado en la pantalla de dispositivos registrados, como se puede ver en la Figura 4.20.



Capítulo 4. Instalación y configuración del sistema

The figure consists of three screenshots illustrating the device registration process in The Things Stack Sandbox:

- Screenshot 1:** Application overview page for "tfg-ithapp". It shows 1 end device registered. The sidebar includes links for Application overview, End devices, Live data, Webhooks, Message storage, and Payload formatters.
- Screenshot 2:** End devices registration page. It shows 2 last activities. The page includes a search bar, a "Labels" dropdown, an "Import end devices" button, and a "+ Register end device" button.
- Screenshot 3:** Register end device page for an Arduino MKR WAN 1310. The page includes fields for End device type (selected as "Select the end device in the LoRaWAN Device Repository"), Input method (selected as "Enter end device specifics manually"), and device details like Brand (Arduino SA), Model (MKR WAN 1310), Hardware Ver. (1.0), Firmware Ver. (1.2.3), and Profile (EU_863_870). A detailed description of the MKR WAN 1310 board is provided, along with a product image and a link to the Product website. The Frequency plan is set to "Europe 863-870 MHz (SF9 for RX2 - recommended)".

Figura 4.15: Registrar dispositivo en aplicación en The Things Network.



Provisioning information

JoinEUI ⓘ *

11 22 33 44 55 66 77 88 **Reset**

This end device can be registered on the network

DevEUI ⓘ *

Generate 1/50 used

AppKey ⓘ *

Generate

End device ID ⓘ *

my-new-device

Device properties

Labels

Select... | ▾

Add a label or [create one](#) to categorize your devices

After registration

View registered end device
 Register another end device of this type

Register end device

Figura 4.16: Datos de formulario para registrar dispositivo en The Things Network



Capítulo 4. Instalación y configuración del sistema

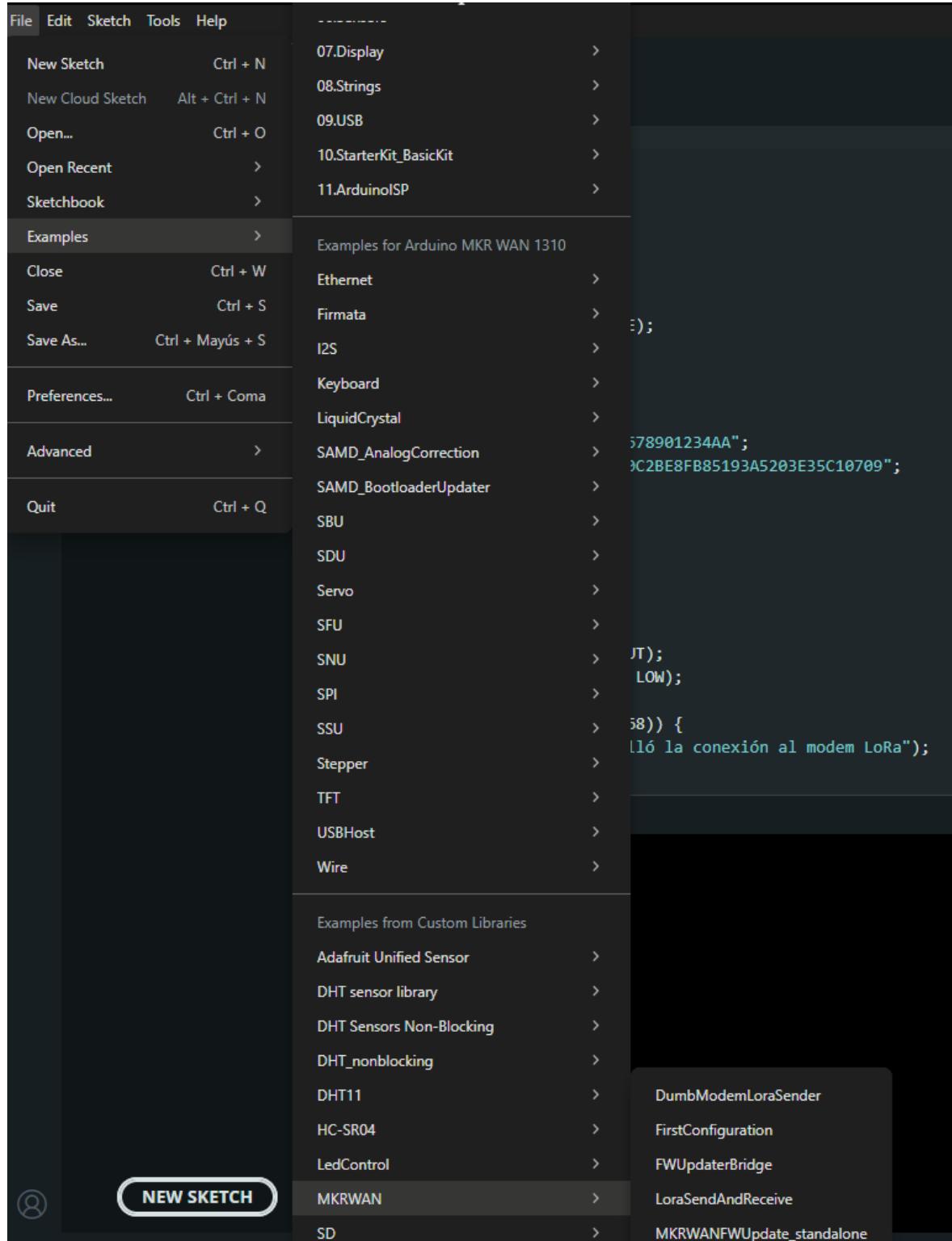


Figura 4.17: Ejemplo para obtener JoinEUI



Capítulo 4. Instalación y configuración del sistema

```
Welcome to MKR WAN 1300/1310 first configuration sketch
Register to your favourite LoRa network and we are ready to go!
Your module version is: ARD-078 1.2.3
Please make sure that the latest modem firmware is installed.
To update the firmware upload the 'MKRWANFWUpdate_standalone.ino' sketch.
Your device EUI is: a8
Are you connecting via OTAA (1) or ABP (2)?
```

Figura 4.18: Monitor serie para obtener EUI

Provisioning information

JoinEUI ⓘ *

11 22 33 44 55 66 77 88

This end device can be registered on the network

DevEUI ⓘ *

 1/50 used

AppKey ⓘ *

End device ID ⓘ *

eui-

Device properties

Labels

Select... | ▾

Add a label or [create one](#) to categorize your devices

After registration

View registered end device

Register another end device of this type

Figura 4.19: Datos completos en formulario de dispositivo en The Things Network



End devices (1)			Search
NAME AND ID	DEVEUI	JOINER	
[REDACTED]	[REDACTED]	12 34 56 78 90 12 34 AA	

Figura 4.20: Listado de dispositivos en The Things Network

4.2.3. Configurar dispositivo en la aplicación

El siguiente paso consiste en configurar el dispositivo dentro de la aplicación en The Things Network para que, al recibir datos del sensor, la plataforma sea capaz de interpretarlos correctamente. Para ello, es necesario acceder al dispositivo que se ha registrado previamente y seleccionar la opción “Payload formatters“, como se muestra en la Figura 4.21.

Una vez dentro de esta pestaña se pincha en la sección de “Uplink“, en el desplegable “Formatter type“, se debe escoger la opción “Custom Javascript formatter“, tal como se observa en la Figura 4.22.

Una vez seleccionada esta opción, aparecerá un cuadro de texto donde será necesario insertar el código correspondiente, que se encuentra disponible en el repositorio GitHub del proyecto.

Concretamente, este código se encuentra en el archivo llamado “Custom Javascript formatter.js“ dentro de la carpeta “The Things Network“, como se ilustra en la Figura 4.23.

The screenshot shows the 'Device overview' page for a registered device. At the top, there are tabs for 'Device overview', 'Live data', 'Messaging', 'Location', 'Payload formatters' (which is highlighted with a red box), and 'Settings'. Below these tabs, there are two main sections: 'End device info' (showing an Arduino S.A. connected via MKR WAN 1310) and 'Latest decoded payload' (displaying a JSON object with fields: humedad: 48.92, ilit: 77.42, temperatura: 29.24). The 'Payload formatters' section contains a code editor with the following JavaScript code:

```
1 [j]
2   "humedad": 48.92,
3   "ilit": 77.42,
4   "temperatura": 29.24
5 }
```

Figura 4.21: Acceso a Uplink en dispositivo registrado en The Things Network



Capítulo 4. Instalación y configuración del sistema

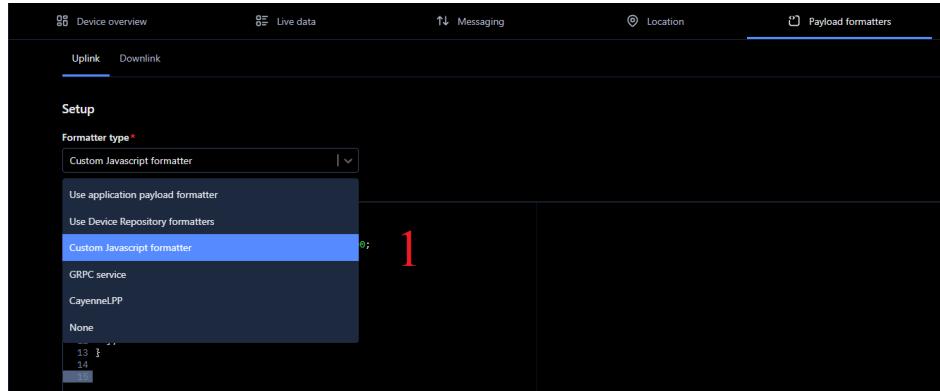


Figura 4.22: Selección de “Uplink“ en dispositivo registrado en The Things Network

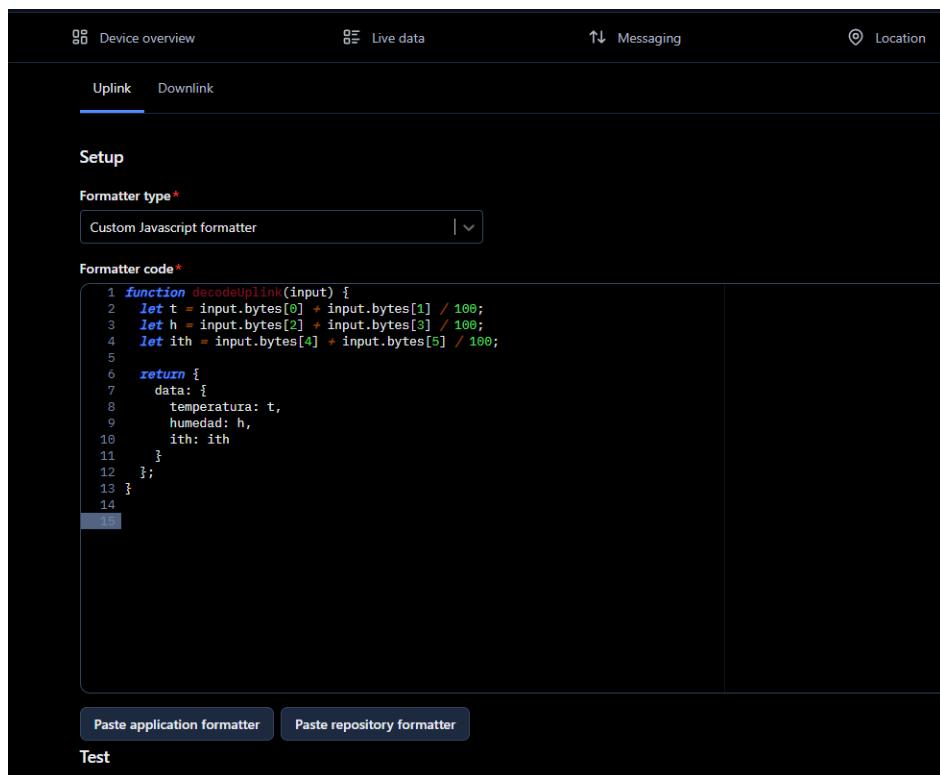


Figura 4.23: Código en JavaScript en el dispositivo en The Things Network

4.2.4. El código de Arduino

El siguiente paso es adaptar el código del Arduino para que pueda conectarse correctamente a la aplicación que se ha creado en The Things Network y enviar los datos. Para ello, se debe abrir el archivo “Codigo-tfg“, el mismo que se utilizó anteriormente para programar el Arduino, y que se encuentra disponible en el repositorio de GitHub del proyecto.

En este código es necesario actualizar los valores de los campos appEui y appKey, reemplazándolos por los datos específicos que aparecen en la sección “Overview“ del



dispositivo que se ha registrado en The Things Network. Este se muestra en la Figura 4.24.

Una vez realizada la configuración, se debe cargar el código en el Arduino pulsando sobre el botón de carga, que está representado por la flecha hacia la izquierda en la parte superior del entorno Arduino IDE. Si todo está configurado correctamente, el resultado que se debería visualizar en el Monitor Serie será similar al que se muestra en la Figura 4.25

Posteriormente, al acceder a la pestaña “Live data“ dentro de la aplicación en The Things Network, se deberían visualizar el envío de datos transmitidos, tal y como se muestra en la Figura 4.26.

```
Codigo_tfg.ino
1 #include <MKRWAN.h>
2 #include "DHT.h"
3
4 #define DHTPIN 7
5 #define DHTTYPE DHT11
6
7
8
9 DHT dht(DHTPIN, DHTTYPE);
10 LoRaModem modem;
11
12 int ledPin = 6;
13
14 String appEui = "XXXXXXXXXXXX";
15 String appKey = "XXXXXXXXXXXXXX";
16
17 void setup() {
18     Serial.begin(9600);
19     while (!Serial);
20
21     dht.begin();
22
23     pinMode(ledPin, OUTPUT);
24     digitalWrite(ledPin, LOW);
25
26     if (!modem.begin(EU868)) {
27         Serial.println("Falló la conexión al modem LoRa");
28         while (1);
29     }
30 }
```

Figura 4.24: Código Arduino de appEui y appKey



The screenshot shows the Arduino Serial Monitor window titled "Serial Monitor X". It has two tabs: "Output" (selected) and "Input". The message area contains the following text:
Conectado a TTN
Enviado: T=29.50 H=48.00 ITH=77.35
No hay downlink en este ciclo.

Figura 4.25: Salida de datos en monitor serie en Arduino

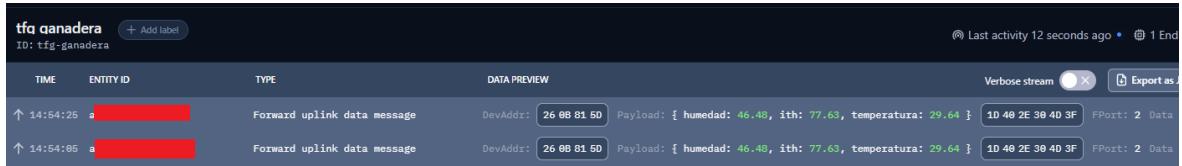


Figura 4.26: Datos en vivo en The Things Network

4.2.5. Integración de Aplicación con sistema de adquisición de datos

A continuación, es necesario configurar la dirección del servidor al que se enviarán los datos, es decir, la URL correspondiente al Sistema de gestión de datos.

La comunicación entre The Things Network y este subsistema se realiza mediante webhooks. Para configurarlo, hay que acceder a la sección “Webhooks” desde el menú lateral de la aplicación en The Things Network, como se muestra en la Figura 4.27.

Dentro de esta sección, se debe crear un nuevo webhook de tipo personalizado, completando los campos requeridos con la información correspondiente, tal y como se observa en la Figura 4.28 .

Uno de los aspectos más importante es seleccionar el formato de transmisión como “JSON” y asegurarse de introducir correctamente la URL base del backend que recibirá los datos. Además, es importante añadir las cabeceras necesarias, como las que se muestran en la Figura 4.29, seleccionar el tipo de evento que se desea recibir y especificar la ruta donde se deben enviar los datos cada vez que The Things Network reciba un nuevo mensaje.

Esta configuración garantiza que los datos se transmitan correctamente en formato JSON, incluyendo en las cabeceras la información necesaria para que el backend pueda interpretarlos sin problemas. También se ajusta el filtro de eventos para que únicamente se reenvíen los mensajes que hayan sido previamente procesados en The Things Network y que sean relevantes para el sistema.

Es fundamental que esta configuración se mantenga sin modificaciones (excepto la



URL del servidor) ya que el backend está diseñado para recibir los datos con estas especificaciones y rutas establecidas.

Una vez configurado, se deben guardar los cambios. El webhook aparecerá en el panel correspondiente, como se aprecia en la Figura 4.30. Es posible que inicialmente se muestre el mensaje “Status: Request failed”. Esto ocurre porque The Things Network ha intentado enviar un mensaje al webhook, pero todavía no está operativo el subsistema de recepción, que será precisamente el siguiente componente que se configurará.

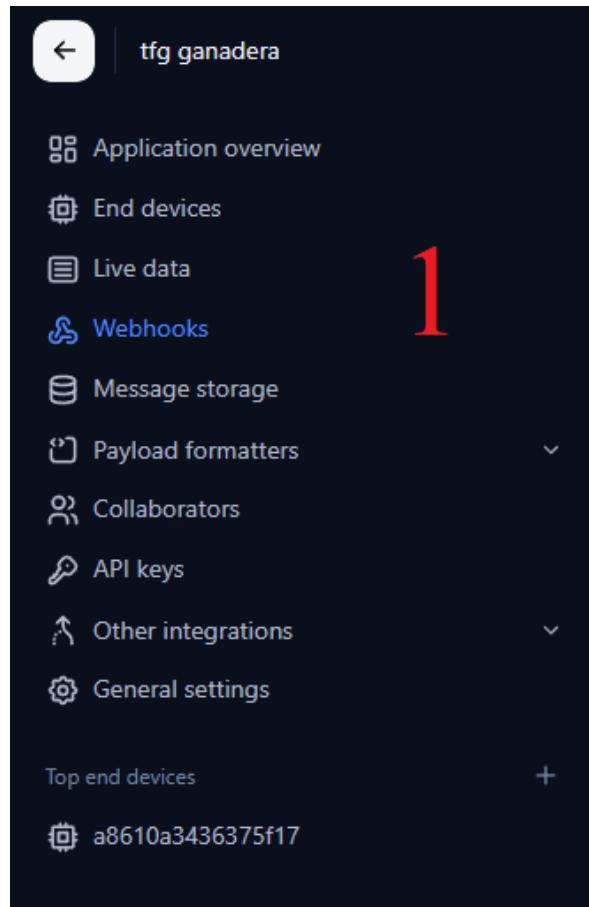


Figura 4.27: Selección de webhook en The Things Network



The figure consists of three screenshots illustrating the steps to create a webhook:

- Screenshot 1 (Left):** Shows the main interface with a large red number "2" and a blue button labeled "+ Add webhook".
- Screenshot 2 (Center):** Shows the "Choose webhook template" screen. It features a large red number "3" and a "Custom webhook" section with the sub-instruction "Create a custom webhook without template".
- Screenshot 3 (Right):** Shows the "Add webhook" configuration screen. It includes fields for "Webhook ID" (set to "my-new-webhook"), "Webhook format" (set to "JSON"), "Base URL" (set to "https://example.com/webhooks"), and "Downlink API key" (an empty field). It also includes sections for "Request authentication", "Additional headers", and "Filter event data".

Figura 4.28: Creación de webhook en The Things Network.



Capítulo 4. Instalación y configuración del sistema

General settings

Webhook ID*
prueba6

Webhook format*
JSON

Base URL*
https://direct-aware-quagga.ngrok-free.app

Downlink API key
[redacted]
The API key will be provided to the endpoint using the "X-Downlink-Apikey" header

Request authentication
 Use basic access authentication (basic auth)

Additional headers
+ Add header entry

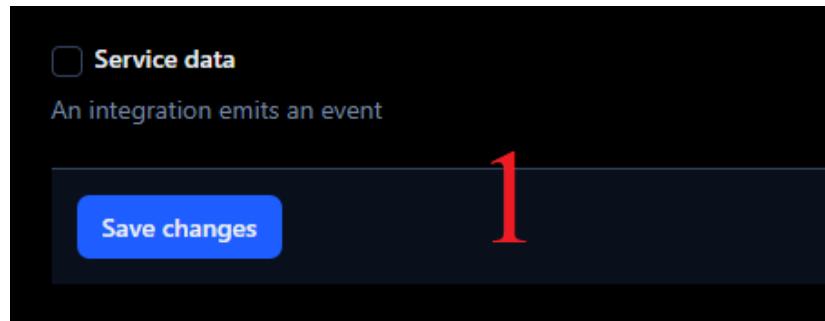
Filter event data
up.uplink_message.decoded_payload |

+ Add filter path

Enabled event types
For each enabled event type an optional path can be defined which will be appended to the base URL

Uplink message /webhook
An uplink message is received by the application

Figura 4.29: Relleno de datos del formulario al crear webhook en The Things Network



2

Webhooks (1)			
ID	BASE URL	TEMPLATE ID	STATUS
prueba6	https://direct-aware-quagga.ngrok-free.app	None	Healthy *

Figura 4.30: Registro y guardado de webhook en The Things Network.



4.3. Sistema de gestión de datos

En esta sección se configura el sistema encargado de recibir los datos enviados desde The Things Network, donde se procesarán y se almacenarán en la base de datos en la nube. Este sistema actúa como puente entre la red LoRa y LoRaWAN y el sistema final, garantizando que las lecturas del sensor lleguen de forma correcta y estén disponibles para la aplicación móvil a través de la API.

4.3.1. Preparación del entorno

Para este proyecto, el servidor que recibe y procesa los datos no se ejecuta en un equipo local, sino que está publicado en Railway con un dominio accesible públicamente desde Internet.

En este entorno se emplea Node.js como tecnología principal para desarrollar el servicio backend que gestiona la recepción mediante una API REST y realiza la inserción en MySQL, también alojada en Railway.

Dado que el servicio está expuesto de forma pública y segura, se configura en la consola de The Things Network un webhook que apunta directamente al dominio del backend en Railway, de modo que cada uplink se entrega a la API para su validación y almacenamiento.

Para ello, se deben seguir los pasos de configuración indicados en las secciones posteriores de este manual.

4.3.1.1. Instalación de Node.js

Primero, se descarga e instala Node.js, que es un entorno de ejecución para JavaScript que permite trabajar con servidores y aplicaciones web por parte del servidor. Para ello, se accede a la página oficial de Node.js y se descarga la versión LTS (Long Term Support), lo que garantiza un soporte prolongado. Estos pasos se pueden ver la Figura 4.31.

Una vez completada la descarga, se procede a instalar el software aceptando las opciones por defecto del instalador. Finalizada la instalación, se puede comprobar su correcto funcionamiento ejecutando el siguiente comando en la terminal:

```
node -v
```

Este comando muestra la versión instalada, confirmando que Node.js está listo para usarse, tal como se muestra en la Figura 4.32.



The screenshot shows the 'Descarga Node.js®' (Download Node.js) page. It features a search-like interface where you can specify 'Obtiene Node.js®' (Get Node.js), 'versión' (version) 'v22.17.0 (LTS)', 'para' (for) 'Windows', 'usando' (using) 'Docker', and 'con' (with) 'npm'. A blue info bar at the top says: 'Info Want new features sooner? Get the [latest Node.js version](#) instead and try the latest improvements!' Below this is a code block showing Docker commands to run Node.js. The 'PowerShell' tab is selected. A 'Copiar al portapapeles' (Copy to clipboard) button is available. A note at the bottom states: 'Docker is a containerization platform. If you encounter any issues please visit [Docker's website](#)'.

Obtiene Node.js® para v22.17.0 (LTS) para Windows usando Docker con npm

Info Want new features sooner? Get the [latest Node.js version](#) instead and try the latest improvements!

```
1 # Docker provee instrucciones dedicadas para cada sistema operativo.
2 # Por favor consulta la documentación oficial en https://www.docker.com/get-started/
3
4 # Descarga la imagen de Docker de Node.js:
5 docker pull node:22-alpine
6
7 # Crea un contenedor de Node.js e inicia una sesión shell:
8 docker run -it --rm --entrypoint sh node:22-alpine
9
10 # Verifica la versión de Node.js:
11 node -v # Debería mostrar "v22.17.0".
12
13 # Verifica versión de npm:
14 npm -v # Debería mostrar "10.9.2".
```

PowerShell Copiar al portapapeles

Docker is a containerization platform. If you encounter any issues please visit [Docker's website](#)

O obtiene una versión pre compilada de Node.js® para Windows usando la arquitectura x64 . 2

[Windows Installer \(.msi\)](#) [Standalone Binary \(.zip\)](#)

Figura 4.31: Instalación y descarga de Node.js.



```
PROBLEMS    OUTPUT    DEBUG CONSOLE    TERMINAL    PORTS    SQ
PS C:\Users\Usuario\Desktop\sensor\ttn-server2> node -v
v22.14.0
PS C:\Users\Usuario\Desktop\sensor\ttn-server2>
```

Figura 4.32: Versión instalada de Node.js

4.3.1.2. Creación del proyecto Node.js

El siguiente paso consiste en crear una carpeta que funcionará como contenedor del proyecto. Para ello, se ejecutaron los siguientes comandos en la terminal:

```
mkdir ttn-server2
cd ttn-server2
npm init -y
```

El comando `npm init -y` genera automáticamente el archivo `package.json`, que contiene la configuración básica del proyecto.

El resultado obtenido a ejecutar estos comandos generara un arbol de archivos, como se muestra en la Figura 4.33.

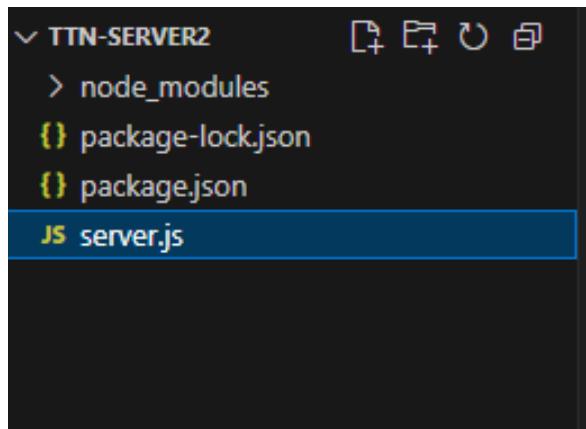


Figura 4.33: Conjunto de archivos generados para la creación del proyecto

4.3.1.3. Instalación de dependencias

Una vez, dentro del proyecto, se instalan los paquetes necesarios para montar el servidor web y gestionar las peticiones entrantes, además de conectarse con la base de datos y permitir el acceso desde otras aplicaciones:



```
npm install express body-parser mysql2 cors
```

Todos estos paquetes puesto en el comando anterior para instalar tienen la siguiente función:

- **express**: framework para crear el servidor HTTP.
- **body-parser**: permite procesar los cuerpos de las peticiones en formato JSON.
- **mysql2**: cliente para conectarse a la base de datos MySQL.
- **cors**: habilita el intercambio de recursos entre diferentes orígenes (Cross-Origin Resource Sharing).

4.3.1.4. Desarrollo del servidor

A continuación, se crea un archivo llamado server.js, que contendrá toda la lógica necesaria para gestionar la recepción de datos, su almacenamiento y la comunicación con TTN. El archivo incluye:

- Conexión a la base de datos MySQL.
- Rutas para recibir datos del sensor cuya ruta es “/webhook”.
- Rutas para registrar granjas, sensores y usuarios.
- Una ruta para comprobar si el servidor en la nube esta activo.
- Una ruta para consultar la última medición registrada.
- La activación del servidor en el puerto 8080.

En la Figura 4.34 se muestra como cabecera del archivo los paquetes instalados mencionados en la anterior sección, para el correcto funcionamiento del servidor.

Y, como resto del archivo server.js, todas las acciones o lógica necesaria para hacer determinadas acciones para el ingreso de datos recibidos desde The Things Network en la base de datos de MySQL, tal como se muestra en la Figura 4.35.



Capítulo 4. Instalación y configuración del sistema

```
JS server.js > ...
JS server.js ...
1  const express = require("express");
2  const mysql = require("mysql2/promise");
3  const cors = require("cors");
4
5  const app = express();
6
7  // 1) Middleware
8  app.use(express.json());
9  app.use(cors({ origin: "*" })); // si usas Flutter Web, pon tu dominio en vez de "*"
10
11 // 2) Puerto (Railway coloca PORT)
12 const port = process.env.PORT || 3000;
13
14 // 3) Config MySQL (Railway inyecta estas vars al crear la DB)
15 const pool = mysql.createPool({
16   host: process.env.MYSQLHOST || "localhost",
17   port: process.env.MYSQLPORT ? Number(process.env.MYSQLPORT) : 3306,
18   user: process.env.MYSQLUSER || "root",
19   password: process.env.MYSQLPASSWORD || "",
20   database: process.env.MYSQLDATABASE || "ganaderapp",
21   waitForConnections: true,
22   connectionLimit: 10,
23 });
24
25 // 4) Salud
26 app.get("/health", async (_req, res) => {
27   try {
28     const conn = await pool.getConnection();
29     await conn.ping();
30     conn.release();
31     res.json({ ok: true });
32   } catch (e) {
33     console.error(e);
34     res.status(500).json({ ok: false, error: "DB ping failed" });
35   }
36 });

// 5) Endpoints
```

Figura 4.34: Cabecera del archivo por parte del servidor de paquetes necesarios

```
// TTN webhook → inserta mediciones
app.post("/webhook", async (req, res) => {
  try {
    const uplink = req.body.uplink_message;
    if (!uplink?.decoded_payload) {
      return res.status(400).send("Payload invalido");
    }

    const { temperatura, humedad, ith } = uplink.decoded_payload;
    const sql = "INSERT INTO mediciones (temperatura, humedad, ith) VALUES (?, ?, ?)";
    await pool.query(sql, [temperatura, humedad, ith]);

    console.log("✓ Datos recibidos:", { temperatura, humedad, ith });
    res.send("OK");
  } catch (err) {
    console.error("✗ Error insertando mediciones:", err);
    res.status(500).send("Error en base de datos");
  }
});
```

Figura 4.35: Fragmento de código encargado de insertar mediciones en MySQL

4.3.1.5. Despliegue del servidor en Railway (con integración GitHub)

En esta sección se explica cómo publicar el servidor de Node.js en Railway, partiendo de un repositorio en GitHub conectado a Railway y terminando con la API accesible por HTTPS para conectar el webhook de The Things Network.



Para ello, se necesita como requisito previo tener cuenta en Github y Railway, y ser conectada desde Railway, donde esto te lo pide con solo entrar por primera vez a Railway, se conecta y se hacen los siguientes pasos:

- Crear un repositorio nuevo (público o privado) y subir el código del servidor.
- Incluir los siguientes archivos: `server.js`, `package.json`, `package-lock.json`

Esto se muestra el repositorio y los archivos necesarios en la Figura 4.36

File	Description	Time Ago
i92rroraj Sincroniza lockfile y añade cors	✓	856bb99 · 4 days ago
package-lock.json	Sincroniza lockfile y añade cors	4 days ago
package.json	Sincroniza lockfile y añade cors	4 days ago
server.js	Proyecto TFG	4 days ago

Figura 4.36: Repositorio del Node.js en Github

El siguiente paso es acceder a Railway, una vez conectado a Github y crear un proyecto nuevo. Después elegir la opción de desplegar desde Github y autorizar el acceso al repositorio, y por ultimo, seleccionar el repositorio del backend y confirmar el despliegue del servicio. Donde Railway detectará Node.js y quedará como activo el servicio en el proyecto de Railway. Todo esto se muestra en la Figura 4.37.

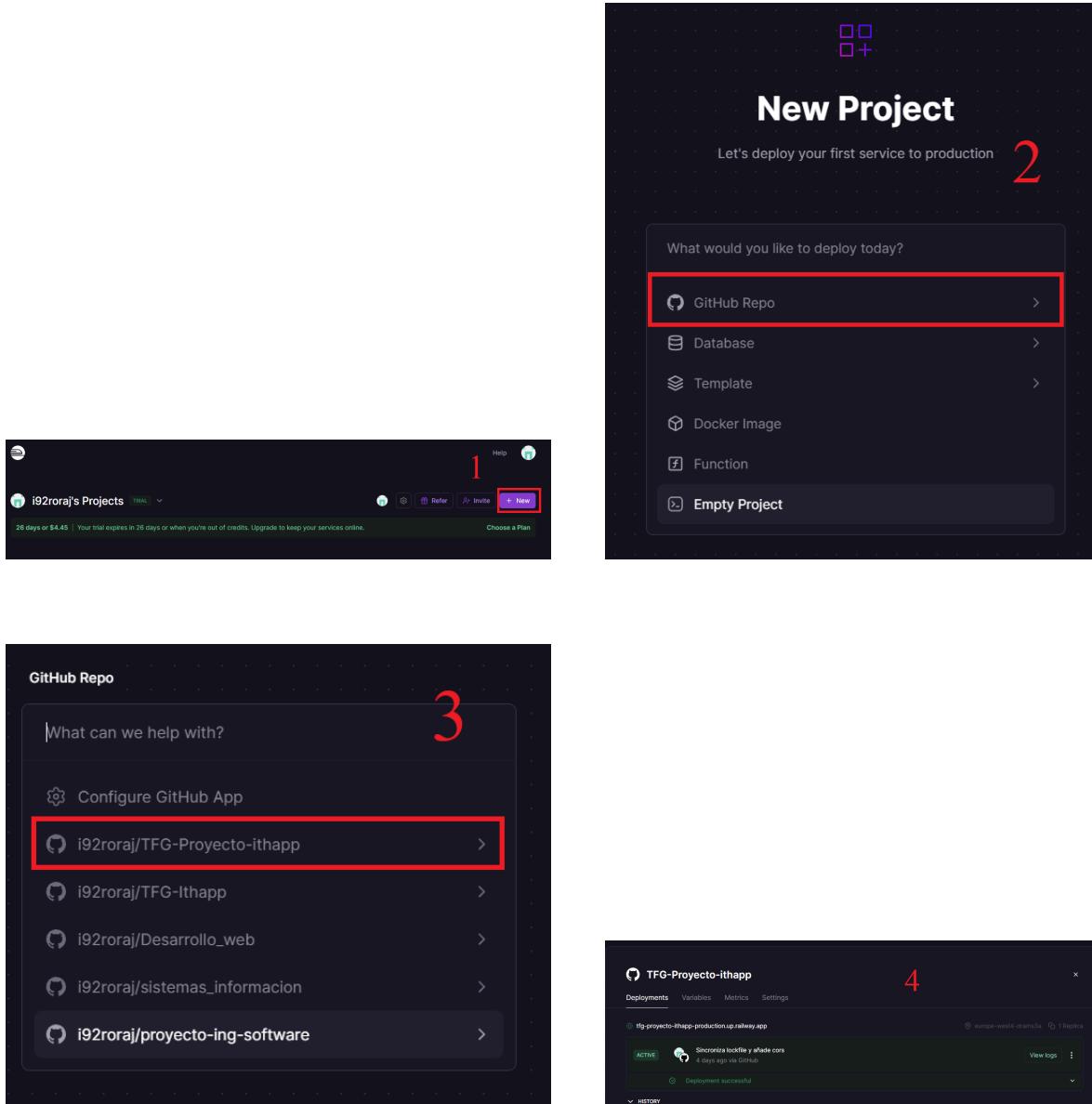


Figura 4.37: Creación de proyecto en Railway

A continuación, una vez conectado todo lo anterior, debemos crear el servicio de MySQL en Railway, para ello se siguen los siguientes pasos:

1. En el mismo proyecto de Railway, añadir un servicio de base de datos MySQL.
2. Anotar las credenciales generadas (host, puerto, usuario, contraseña, base de datos).
3. En el servicio del backend, definir las variables de entorno necesarias:
 - MYSQLHOST, MYSQLUSER, MYSQLPASSWORD, MYSQLDATABASE, MYSQLPORT, donde no es necesario establecer PORT, ya que Railway la crea automáticamente.



Todo estos pasos se muestra en la Figura 4.38.

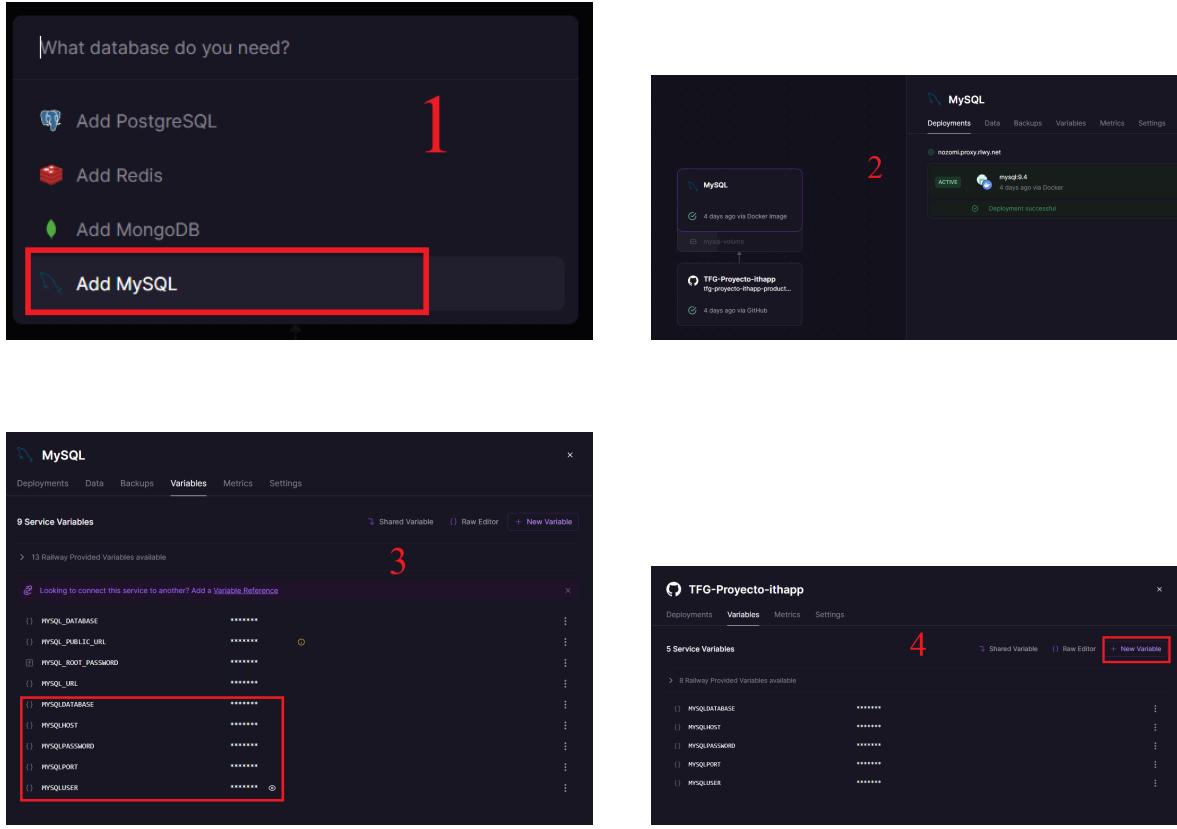


Figura 4.38: Creación de servicio MySQL en Railway

A continuación, debemos hacer el siguiente proceso de obtener un dominio público para hacer las operaciones de envío de y gestión de los datos, para ello se siguen los siguientes pasos :

1. En el servicio de nuestro backend de Node.js, debemos asignar un dominio de HTTPS público, pulsando en “Settings” y en la sección de “Networking”, pulsar sobre “New Domain” para crear un dominio público que utilizaremos en el webhook de The Things Network (ver Figura 4.39).
2. Verificar que los logs del servicio se generan correctamente en el panel de Railway.

Como ultimo paso, debemos configurar el webhook en TTN, donde debemos modificar el webhook creado, como se ha explicado anteriormente, y poner la URL en los campos de configuración del webhook.

Una vez, hecho todo esto, se procede a enviar datos con un uplink de prueba y confirmar respuesta HTTP 200 en TTN (ver Figura 4.40) y que haya un registro de la petición en los logs de Railway (ver Figura 4.41).



Capítulo 4. Instalación y configuración del sistema

The screenshot shows the Railway app interface for the project "TFG-Proyecto-ithapp". The "Settings" tab is selected. Under the "Networking" section, the "Public Networking" subsection is shown, which allows access via HTTP. It displays a card for "tfg-proyecto-ithapp-production.up.railway.app" on port 8080, labeled as "Metal Edge". Below this are buttons for "+ Custom Domain" and "+ TCP Proxy". The "Private Networking" subsection is also visible, showing an internal IP address "tfg-proyecto-ithapp.railway.internal" with an IPv6 link.

Figura 4.39: Generación de dominio público en Railway

The screenshot shows the Railway app interface for the project "TFG-Proyecto-ithapp / 47d42bd". The "HTTP Logs" tab is selected. It displays a table of log entries for the endpoint "/mediciones". The table has columns for Time (CEST), Method, Path, HTTP Status, and Total Duration. The log entries show multiple requests at different times between Aug 27 2025 11:42:21 and Aug 27 2025 11:57:53, all resulting in a 200 status code with varying response times.

Time (CEST)	Method	Path	HTTP Status	Total Duration
Aug 27 2025 11:42:21	GET	/mediciones	200	4ms
Aug 27 2025 11:43:28	GET	/mediciones	200	3ms
Aug 27 2025 11:43:21	GET	/mediciones	200	3ms
Aug 27 2025 11:43:21	GET	/mediciones	200	2ms
Aug 27 2025 11:43:25	GET	/mediciones	200	3ms
Aug 27 2025 11:43:26	GET	/mediciones	200	2ms
Aug 27 2025 11:43:26	GET	/mediciones	200	3ms
Aug 27 2025 11:57:53	GET	/mediciones	200	3ms
Aug 27 2025 11:57:53	GET	/mediciones	200	2ms
Aug 27 2025 11:57:53	GET	/mediciones	200	2ms
Aug 27 2025 11:57:53	GET	/mediciones	200	3ms
Aug 27 2025 11:57:53	GET	/mediciones	200	3ms
Aug 27 2025 11:57:53	GET	/mediciones	200	3ms

Figura 4.40: Logs de registro de respuesta de HTTP en Railway



The screenshot shows a log entry from a Railway application instance named 'tfg-proyecto-ithapp-production.up.railway.app'. The log details a series of data reception events. Each event is timestamped and contains a JSON object with 'temperatura', 'humedad', and 'ith' fields. The data is consistently received at 29.24, 31.28, and 74.46 respectively.

Time (CEST)	Data
Aug 27 2025 11:33:50	Datos recibidos: { temperatura: 29.24, humedad: 31.28, ith: 74.46 }
Aug 27 2025 11:34:10	Datos recibidos: { temperatura: 29.24, humedad: 31.28, ith: 74.46 }
Aug 27 2025 11:34:30	Datos recibidos: { temperatura: 29.24, humedad: 31.28, ith: 74.46 }
Aug 27 2025 11:34:50	Datos recibidos: { temperatura: 29.24, humedad: 32.28, ith: 74.61 }
Aug 27 2025 11:35:10	Datos recibidos: { temperatura: 29.34, humedad: 32.28, ith: 74.72 }
Aug 27 2025 11:35:30	Datos recibidos: { temperatura: 29.34, humedad: 32.28, ith: 74.72 }
Aug 27 2025 11:35:50	Datos recibidos: { temperatura: 29.34, humedad: 31.28, ith: 74.57 }
Aug 27 2025 11:36:10	Datos recibidos: { temperatura: 29.34, humedad: 31.28, ith: 74.57 }
Aug 27 2025 11:36:30	Datos recibidos: { temperatura: 29.34, humedad: 31.28, ith: 74.57 }
Aug 27 2025 11:36:50	Datos recibidos: { temperatura: 29.34, humedad: 31.28, ith: 74.57 }
Aug 27 2025 11:37:10	Datos recibidos: { temperatura: 29.24, humedad: 31.28, ith: 74.46 }
Aug 27 2025 11:37:30	Datos recibidos: { temperatura: 29.24, humedad: 31.28, ith: 74.46 }
Aug 27 2025 11:37:50	Datos recibidos: { temperatura: 29.24, humedad: 31.28, ith: 74.46 }
Aug 27 2025 11:38:10	Datos recibidos: { temperatura: 29.34, humedad: 31.28, ith: 74.57 }
Aug 27 2025 11:38:30	Datos recibidos: { temperatura: 29.34, humedad: 31.28, ith: 74.57 }
Aug 27 2025 11:38:50	Datos recibidos: { temperatura: 29.34, humedad: 31.28, ith: 74.57 }
Aug 27 2025 11:39:10	Datos recibidos: { temperatura: 29.34, humedad: 31.28, ith: 74.57 }

Figura 4.41: Logs de registro de peticiones de datos recibidos en Railway

4.3.2. Configuración y uso de API Key en TTN

Para que la aplicación de Flutter pueda interactuar con The Things Network (TTN) y enviar comandos de control (downlinks) hacia un dispositivo LoRaWAN, es necesario tener con una API Key válida y configurada con los permisos adecuados. Esta clave de acceso permite que la app se comunique directamente con la plataforma TTN sin necesidad de intervención manual.

4.3.2.1. Creación de una API Key en TTN

En esta sección, se detallan los pasos a seguir para generar una API Key valida. Los pasos son los siguientes:

1. Acceder al panel de control de The Things Network.
2. Entrar en la sección de aplicación creada anteriormente.
3. En el menú lateral, seleccionar “API Keys“.
4. Pulsar en el botón “+ Add API key“ para crear una nueva clave.
5. Darle un nombre identificativo en el campo de nombre de API key.
6. Seleccionar los permisos necesarios. Para enviar datos al dispositivo, es necesario marcar los siguientes campos:
 - Write downlink application traffic



- Read application traffic (uplink and downlink)

7. Finalmente, pulsar “Create API Key“.

La plataforma generará una cadena larga de texto que debe guardarse en ese momento, ya que no volverá a mostrarse. Todos estos pasos, se muestra en la Figura 4.42.

Una vez creado, sale en la lista de API Keys en la aplicación que se ha creado en TTN, tal como se muestra en la Figura 4.43.

Esta clave será utilizada internamente por el sistema para comunicarse de forma segura con el dispositivo conectado a la red TTN.

The figure consists of four screenshots illustrating the process of creating an API key in The Things Network:

- Screenshot 1:** Shows the "Applications" tab selected in the navigation bar. The sidebar menu includes "API keys", which is highlighted with a red box.
- Screenshot 2:** Shows the main application overview page with 1 End device. A large red number "2" is overlaid on the screen. A blue button labeled "+ Add API key" is highlighted with a red box.
- Screenshot 3:** Shows the "Add API key" form. It has fields for "Name" (My new API key) and "Expiry date" (dd/mm/aaaa). Under "Rights*", the "Grant individual rights" option is selected. A large red number "3" is overlaid on the screen.
- Screenshot 4:** Shows the "Create API key" button at the bottom of the "Add API key" form. A large red number "4" is overlaid on the screen. The "Create API key" button is highlighted with a red box.

Figura 4.42: Acceder a The Things Network.



KEY ID	NAME	GRANTED RIGHTS
W [REDACTED]	KRQF mqtt-password-key-1747309929928	Read application traffic (uplink and downlink) Write downlink application traffic

Figura 4.43: Lista de API Keys creadas en TTN

4.4. Sistema de almacenamiento de datos

Este subsistema guarda de forma estructurada y persistente todas las lecturas de los sensores, así como la información de usuarios, sensores y granjas. En la versión actual del proyecto la base de datos es MySQL desplegada en la nube con Railway, y su administración durante el desarrollo se realiza con HeidiSQL desde el equipo del usuario.

4.4.1. Instalación de base de datos

Para almacenar datos correctamente es necesario que el servicio MySQL de Railway esté operativo y que podamos conectarnos a él con un cliente como HeidiSQL. A continuación se indican los pasos, aunque se han explicado anteriormente esta parte en la anterior sección como paso esencial para ejecutar correctamente el servidor en la nube:

4.4.1.1. Creación de MySQL en Railway

1. En el proyecto de Railway, crear un nuevo servicio de tipo MySQL.
2. Una vez creado, anotar las credenciales de conexión que muestra el panel: host, puerto, usuario, contraseña y nombre de la base de datos.
3. Vincular estas credenciales al servicio del backend copiarlas como variables de entorno en el backend: MYSQLHOST, MYSQLUSER, MYSQLPASSWORD, MYSQLDATABASE, MYSQLPORT.
4. Importar el esquema: desde HeidiSQL o cualquier cliente SQL, ejecutar el script con las tablas del proyecto (granjas, usuarios, sensores y mediciones).

En la Figura 4.44 puede verse un ejemplo del panel donde se muestran las credenciales del servicio y el enlace al backend.



The screenshot shows the Railway MySQL service configuration page. At the top, there are tabs for Deployments, Data, Backups, Variables (which is selected), Metrics, and Settings. Below the tabs, it says "9 Service Variables". There is a note: "Looking to connect this service to another? Add a Variable Reference". A large red box highlights the following variables:

Variable	Type	Value
MYSQLDATABASE	String	*****
MYSQLHOST	String	*****
MYSQLPASSWORD	String	*****
MYSQLPORT	String	*****
MYSQLUSER	String	*****

Figura 4.44: Servicio MySQL provisionado en Railway y credenciales de conexión.

CAPÍTULO 5

Desinstalación del sistema

En este capítulo se explica, de forma práctica, cómo desinstalar todo el sistema dejando el menor rastro posible. Para ello, se deben seguir los siguientes pasos para cada sistema:

5.1. Sistema de adquisición de datos

En esta sección, para conseguir la desinstalación completa de este sistema, el usuario final debe desconectar los sensores y el gateway.

5.1.1. Eliminación de código de Arduino

En el sistema utilizado con elementos hardware, se deberá conectar el microcontrolador Arduino MKR WAN 1310 al ordenador mediante cable micro USB a USB, tal como se muestra en la Figura 5.1.

Después, se debe abrir el proyecto de “Código_tfg.ino” en Arduino IDE y vaciar todo código del archivo, o también cargar cualquier otro código vacío sin funcionalidad, tal como se muestra en la Figura 5.2.

5.1.2. Desinstalación de librerías de Arduino

Para ello, una vez eliminado el código del microcontrolador Arduino, se procede a borrar del entorno de programación de Arduino IDE.

Primero, debemos ejecutar el entorno para eliminar las librerías “MKRWAN” Y “DHT11”, que han sido utilizadas para la instalación y correcto funcionamiento del proyecto de Arduino. Esto se muestra en la Figura 5.3.

5.1.3. Eliminación de archivos descargados

Para ello, debemos eliminar los archivos descargados, en este caso, “Código_tfg.ino” puede ser eliminado desde el repositorio de la plataforma de GitHub, ya que no se va



a hacer uso de estos archivos.

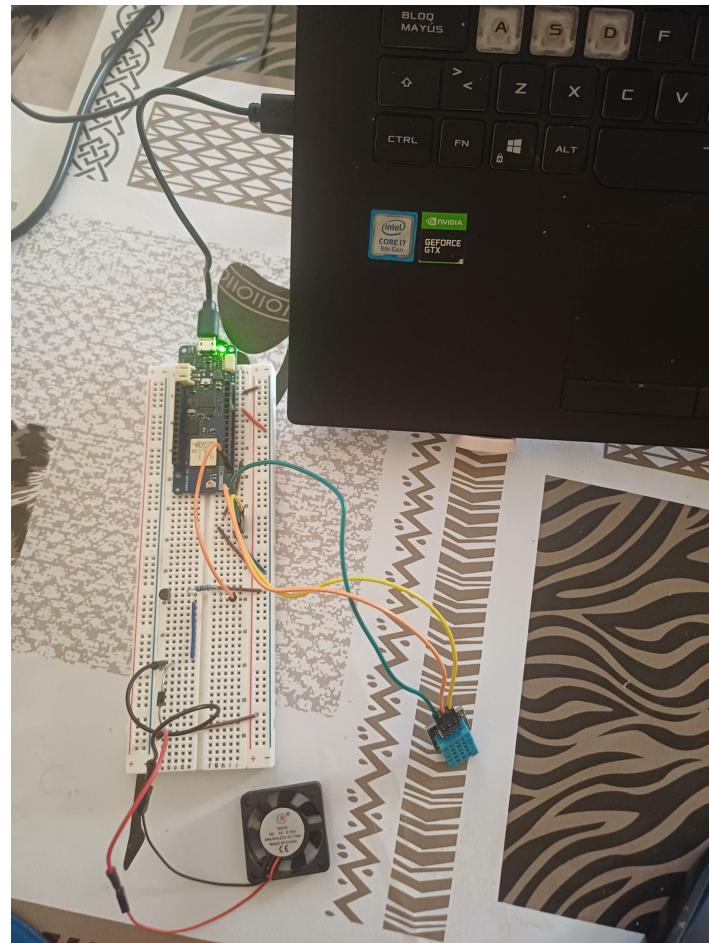


Figura 5.1: Conexión de microcontrolador Arduino al ordenador

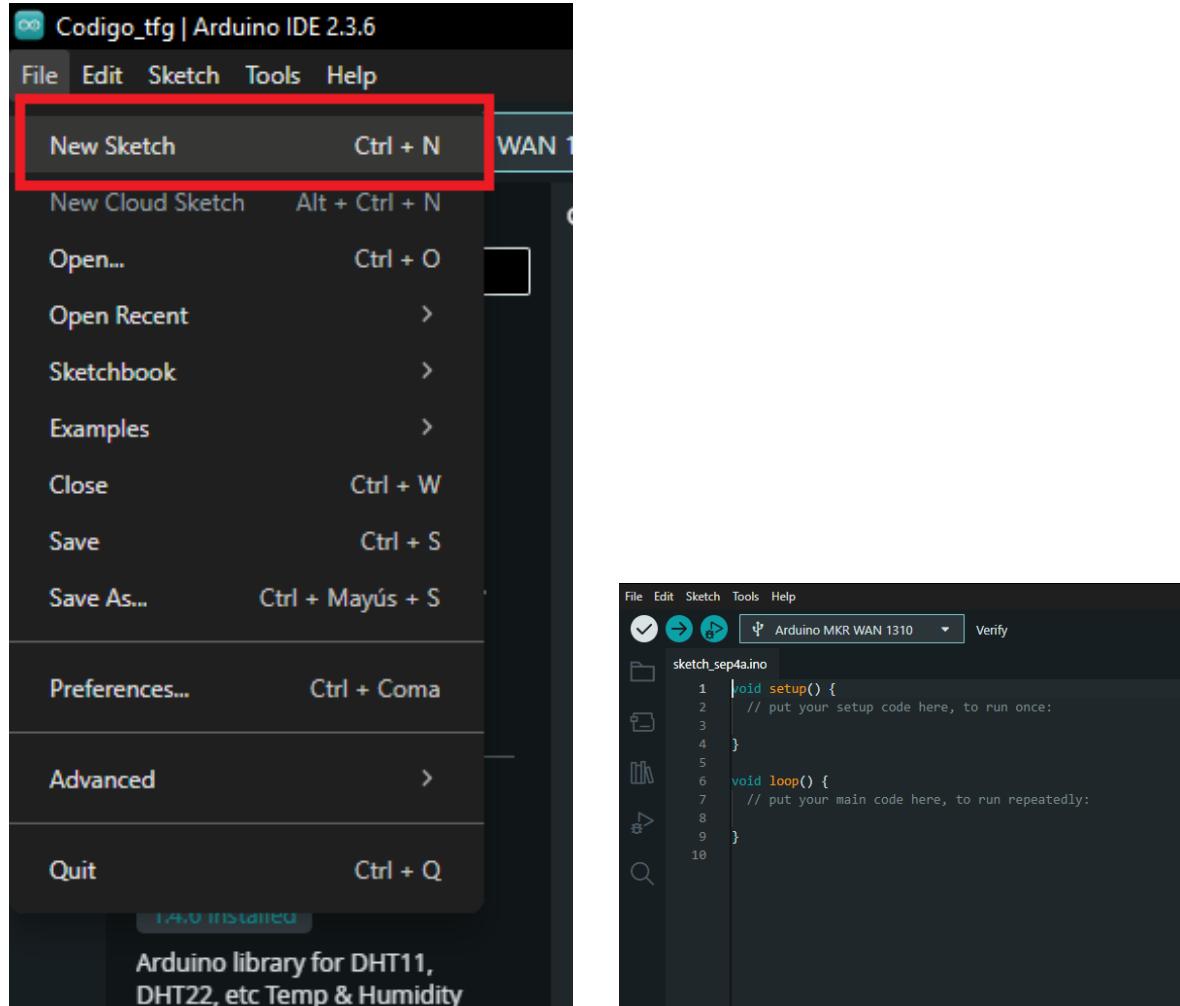


Figura 5.2: Eliminación del código de Arduino

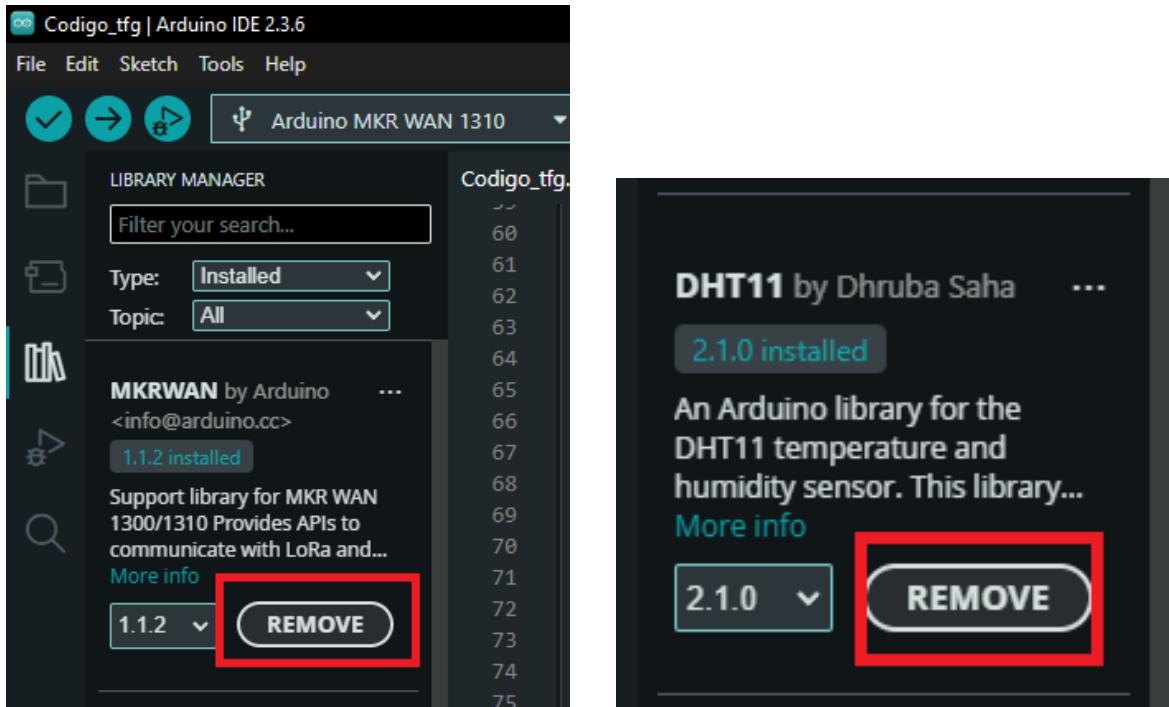


Figura 5.3: Eliminación de librerías de Arduino

5.1.4. Configurar a valores de fábrica la puerta de enlace

Para ello, para configurar a valores por defecto el gateway, se debe tener conectado a la corriente eléctrica el gateway y mantener pulsado el botón de “RESET” durante 5 segundos hasta que el LED parpadee rápidamente de verde a rojo.

5.2. Sistema de recepción y envío de datos

La desinstalación de este sistema de recepción y envío de datos en la que es utilizada mayormente la plataforma de The Things Network, se deben seguir los siguientes pasos para su desinstalación.

5.2.1. Eliminar dispositivo en The Things Network

Para eliminar el dispositivo, se debe acceder a la cuenta de The Things Network, despues, se accede a la pestaña de dispositivos registrados en “End Devices“, pinchar en “Settings“ y para eliminar el Arduino MKRWAN 1310 se pulsa en “Delete end device“, tal como se muestra en Figura 5.4.



Capítulo 5. Desinstalación del sistema

The screenshot shows the 'General settings' page for an end device named 'a8610a3436375f17'. The 'Delete end device' button at the bottom right is highlighted with a red box.

Figura 5.4: Eliminación del dispositivo en The Things Network

5.2.2. Eliminar aplicación en The Things Network

Para eliminar la aplicación creada que se tiene en la cuenta de The Things Network, se debe acceder a la pestaña en el menú lateral izquierdo de TTN y pulsar en “General Settings”, y una vez dentro pulsar el botón “Delete application”, tal como se muestra en la Figura 5.5.

The screenshot shows the 'General settings' page for an application named 'tfq ganadera'. The 'Delete application' button at the bottom right is highlighted with a red box.

Figura 5.5: Eliminación de la aplicación en The Things Network

5.3. Sistema de gestión de datos

Este sistema corresponde a la API desarrollada en Node.js (Express) desplegada en Railway y a cualquier integración asociada (webhook de TTN, claves, dominios). El



objetivo de la desinstalación es detener el servicio, retirar credenciales y eliminar el despliegue sin dejar endpoints públicos activos.

5.3.1. Detener el servicio en Railway

Para detener y eliminar el servidor en la nube se deben seguir los siguientes pasos:

1. Acceder a Railway → Proyecto de Railway → Servicio de Node.js.
2. En “Settings”, pausar o detener el servicio, pulsando “Disconnect” .
3. Quitar el dominio personalizado desde “Domains“.
4. Eliminar servicio pulsando en “Delete service“ del servicio TFG-Proyecto-ithapp.

Todos estos pasos se muestra en la Figura 5.6.



Capítulo 5. Desinstalación del sistema

The figure consists of three screenshots illustrating the process of deleting a service in Railway:

- Screenshot 1:** Shows the "Source" configuration page. It displays a GitHub repository connection (i92roraJ/TFG-Proyecto-ithapp) with a "Disconnect" button highlighted by a red box. Below it, a "Branch connected to production" section shows the "main" branch selected.
- Screenshot 2:** Shows the "Networking" configuration page. It lists a public domain entry: "tfg-proyecto-ithapp-production.up.railway.app" (Port 8080 - Metal Edge), with a "Delete" button highlighted by a red box.
- Screenshot 3:** Shows the "Settings" page. Under "Config-as-code", there's a "Railway Config File" section with a "Delete Service" button highlighted by a red box. This section also contains a warning about permanent deletion and a note about retries.

Figura 5.6: Eliminación del servicio de Node.js en Railway



5.4. Eliminar archivos de código de Node.js

Para eliminar cualquier rastro de los archivos almacenados en el repositorio de GitHub de los archivos del servidor se debe hacer lo siguiente:

1. En GitHub, archivar el repositorio o eliminarlo si no se va a reutilizar.
2. Si se mantiene, eliminar los archivos `server.js`, `package.json`, `package-lock.json`.

5.5. Sistema de almacenamiento de datos

Este sistema comprende la base de datos MySQL gestionada en Railway y los clientes que se han usado para administrarla. Para ello, se deben seguir los siguientes pasos:

5.5.1. Eliminar esquema completo de MySQL

Para ello, se hace de la siguiente manera, donde desde una terminal de nuestro ordenador tal como “CMD“ se ejecuta los siguientes comandos, que se muestra en la Figura 5.7.

```
railway link  
railway connect  
DROP DATABASE railway;
```

5.5.2. Eliminar el servicio MySQL en Railway

1. En el proyecto de Railway, seleccionar el servicio MySQL y elegir *Delete Service*, tal como se muestra en la Figura 5.8.
2. Confirmar que no quedan *Backups* retenidos si se desea una eliminación completa.



Capítulo 5. Desinstalación del sistema

The screenshot shows two windows side-by-side. The left window is a command prompt (cmd.exe) with the title 'C:\Windows\system32\cmd.exe'. It displays the following text:

```
C:\Windows\system32\cmd.exe
Microsoft Windows [Versión 10.0.19045.6216]
(c) Microsoft Corporation. Todos los derechos reservados.

C:\Users\Usuario>railway link
> Select a workspace i92rroraj's Projects
> Select a project rare-purpose
> Select an environment production
> Select a service <esc to skip> MySQL

Project rare-purpose linked successfully! ☀

C:\Users\Usuario>
```

The right window is a MySQL terminal with the title 'C:\Windows\system32\cmd.exe - "node" "C:\Users\Usuario\AppData\Roaming\npm\node_modules\railway\cli\bin\railway.js" connect'. It displays the following text:

```
Microsoft Windows [Versión 10.0.19045.6216]
(c) Microsoft Corporation. Todos los derechos reservados.

C:\Users\Usuario>railway link
> Select a workspace i92rroraj's Projects
> Select a project rare-purpose
> Select an environment production
> Select a service <esc to skip> MySQL

Project rare-purpose linked successfully! ☀

C:\Users\Usuario>railway connect
> Select service MySQL
mysql: [Warning] Using a password on the command line interface can be insecure.
Welcome to the MySQL monitor. Commands end with ; or \g.
Your MySQL connection id is 829
Server version: 9.4.0 MySQL Community Server - GPL

Copyright (c) 2000, 2025, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql>
```

Below the MySQL prompt, the text 'mysql> DROP TABLE railway;' is visible.

Figura 5.7: Eliminación del esquema completo de MySQL en Railway



The screenshot shows the Railway interface for managing a MySQL service. At the top, there's a navigation bar with tabs: Deployments, Database, Backups, Variables, Metrics, and Settings (which is currently selected). Below the navigation is a search bar labeled "Filter Settings...". A note about serverless mode is displayed: "while the container is sleeping will be queued and served when the container wakes up. [Docs ↗](#)". A toggle switch is shown as "Enable Serverless" (disabled). Under "Restart Policy", it says "On Failure" and "Restart the container if it exits with a non-zero exit code." A note indicates "Your plan only supports up to 10 retries" with an "Upgrade" button. The number of retries is set to "10". On the left side, there's a circular icon with a flame symbol and the text "Delete Service". A warning message states: "Deleting this service will permanently delete all its deployments and remove it from this environment. This cannot be undone." A prominent red "Delete service" button is located at the bottom of this section.

Figura 5.8: Eliminación del servicio MySQL en Railway

CAPÍTULO 6

Instalación y configuración de la aplicación móvil

La app desarrollada para el todo el sistema desarrollado puede obtenerse de dos formas distintas, según la forma de instalación:

- **Distribución pública (recomendado).** Descarga directa desde **Google Play Store**. Ideal cuando la versión estable ya está publicada y cualquier usuario final necesita instalarla.
- **Distribución interna o versión beta.** Instalación manual mediante el **archivo .apk** proporcionado por el usuario técnico de desarrollo. Este paso, se puede obtener de entornos sin acceso a Play Store o cuando se quiera usar la aplicación con una versión preeliminar.

6.1. Instalación a través de Google Play Store

1. **Abrir Play Store.** En el móvil Android, acceda a la aplicación *Play Store* y busque “*IthApp*”.
2. **Descargar e instalar.** Pulse *Instalar* y el sistema descargará la versión firmada más reciente y gestionará futuras actualizaciones de forma automática.
3. **Primer arranque.** Una vez completada la instalación, pulse *Abrir* para lanzar la aplicación y continuar con la configuración inicial.

Este método tiene como ventaja que simplifica la distribución y asegura que los usuarios finales reciban actualizaciones sin intervención manual.

6.2. Instalación manual mediante archivo APK

1. **Obtener el APK.** Descargue el fichero *IthApp-vXX.apk* desde un portal interno o mediante un código QR proporcionado.
2. **Permitir orígenes externos.**
Ajustes → Seguridad → Instalar apps desconocidas → Habilitar para el gestor de archivos del dispositivo móvil.



3. **Ejecutar el APK.** Seleccionar el archivo descargado, revisar los permisos solicitados y finalizar la instalación.
4. **Verificar versión.** *Ajustes → Aplicaciones → IthApp* para comprobar que la versión coincide con la indicada por el responsable técnico.

6.3. Configuración inicial

1. **Permisos.** Al primer arranque se solicitarán: acceso a Internet, permiso de recepción de notificaciones y la ubicación para registrar automáticamente la granja.
2. **Crear cuenta o iniciar sesión.** *Registrarse* para un nuevo usuario (campos: nombre, e-mail, contraseña, granja) o *Iniciar sesión* si ya dispone de credenciales.
3. **Vincular sensor.** Menú *Sensores → Añadir* y escanear el DevEUI del nodo o escribirlo manualmente.
4. **Definir umbral ITH.** *Ajustes → Notificaciones* → fijar el valor crítico (74 por defecto). La aplicación enviará alertas push cuando lo supere.

Completados estos pasos, la app comenzará a solicitar datos al endpoint `/mediciones` cada 10s, mostrará temperatura, humedad e índice ITH actuales y permitirá al usuario emitir comandos *downlink* al servidor para activar o desactivar los dispositivos conectados.

CAPÍTULO 7

Descripción y uso de la aplicación móvil

Este capítulo detalla cómo instalar, configurar y manejar la app desarrollada para la supervisión ganadera. Por ello, se explica en cada sección las funcionalidades y pantallas en las que consiste la aplicación móvil.

7.1. Pantalla de bienvenida

Al abrir la aplicación se muestra una pantalla muy simple: en el centro en la parte superior aparece el logotipo y el nombre de la herramienta. Si el dispositivo ya conserva un token de validación de sesión iniciada, al pulsar el botón se salta directamente a la pantalla de medidas (Sección 7.4), de esta forma el usuario no tiene que autenticarse cada vez que accede.

En cambio, cuando se instala la app por primera vez o si el token de autenticación ha caducado, el botón redirige a la pantalla de inicio de sesión descrita en la Sección 7.2.

Así, la pantalla de bienvenida actúa como punto de entrada y decide, de manera transparente, si continuar con la autenticación o llevar al usuario a la visualización en tiempo real de las variables ambientales. En la Figura 7.1 se muestra la interfaz gráfica de la pantalla de bienvenida.

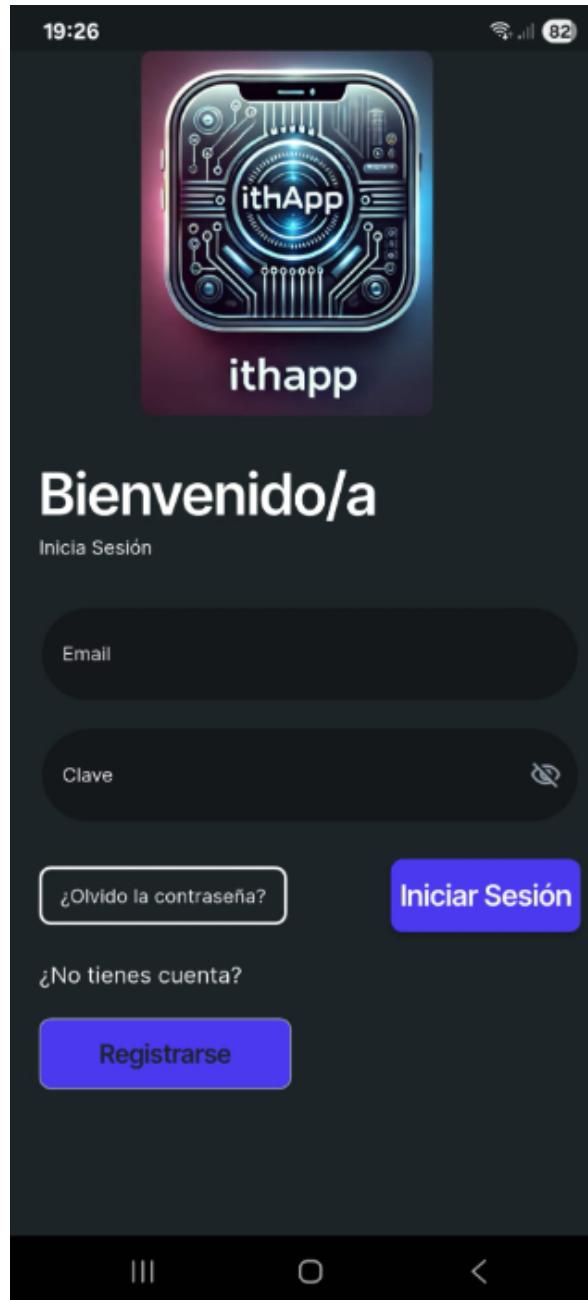


Figura 7.1: Pantalla de bienvenida de app móvil

7.2. Inicio de sesión

La pantalla de inicio de sesión permite que cada usuario se identifique con sus credenciales antes de acceder a los datos de su explotación. Consta de dos campos de texto **Correo electrónico** y **Contraseña** y un botón denominado ‘‘Iniciar sesión’’.

Bajo los campos se muestra un enlace “‘Olvidaste la contraseña?’ que lleva al flujo de recuperación mediante correo electrónico. En la Figura 7.2 se muestra la pantalla de inicio de sesión.



Para ello, se hacen los siguientes pasos para cumplir todo este proceso de autenticación:

1. El usuario introduce su dirección de correo y la contraseña definida en el registro.
2. Al pulsar “Iniciar sesión” la aplicación envía una petición POST al endpoint del backend.
3. Si las credenciales son correctas, el servidor responde con un token de autenticación que la app almacena de forma segura.
4. Inmediatamente se navega a la pantalla principal (Sección 7.4) donde se muestran las medidas en tiempo real.

Por el contrario, si sucede algún problema o error que se muestran a continuación, se muestra una alerta y, por ende, no permite continuar a la siguiente pantalla. Los errores mas comunes son:

- **Correo o contraseña incorrectos.** Se muestra un mensaje emergente indicando que las credenciales no son válidas y se alerta al usuario a reintentarlo.
- **Servidor inaccesible.** Si la API no responde (por ejemplo, caída del backend o fallo de conexión), la pantalla informa de la imposibilidad de iniciar sesión y sugiere comprobar la conexión al servidor.

Gracias a esto método de autenticación, la aplicación se asegura de que sólo usuarios autorizados puedan consultar las medidas y enviar comandos a los sensores.

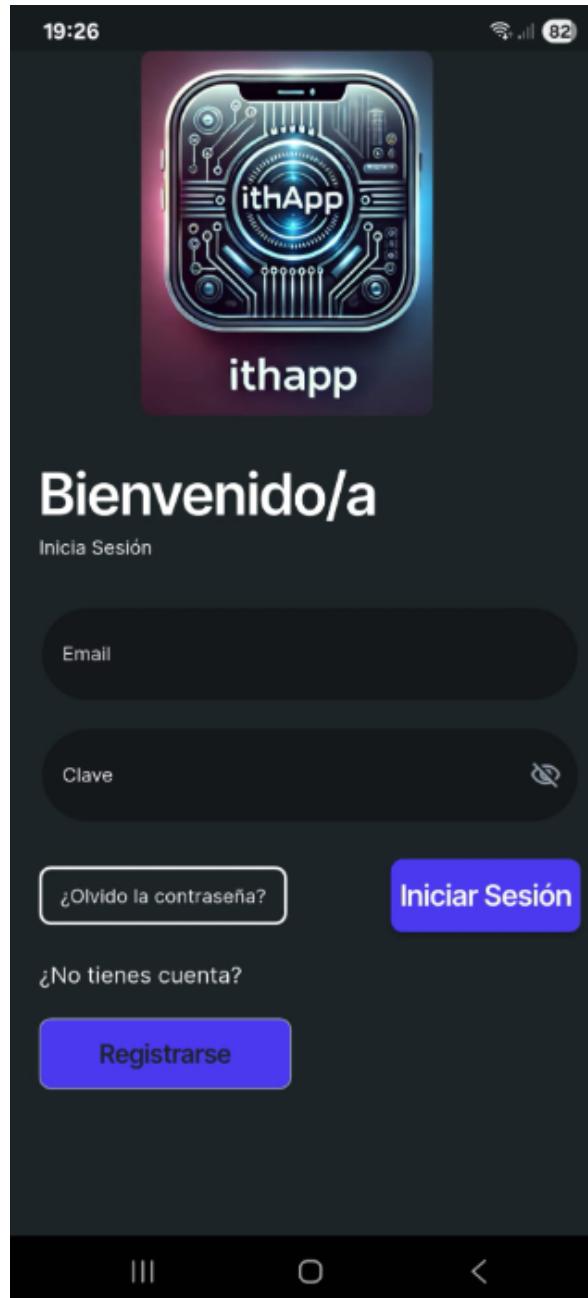


Figura 7.2: Pantalla de inicio de sesión de app móvil

7.3. Registro de usuario

La pantalla de registro permite crear una nueva cuenta y asociarla a un sensor y una explotación. Consta de los siguientes campos:

- Email
- Nombre



- Contraseña
- Fecha de nacimiento
- Nombre del sensor
- Nombre de la granja
- Dirección de la granja

Para ello, se sigue los siguientes pasos para completar el proceso de registro del usuario:

1. Rellenar todos los campos del formulario.
2. Pulsar el botón Guardar.
3. La aplicación envía una petición POST al servicio de autenticación del servidor
4. El servidor valida los datos y crea la cuenta. Si todo es correcto, devuelve un código de éxito junto con el identificador del nuevo usuario.
5. Tras la confirmación, la aplicación redirige a la pantalla de inicio de sesión para que el usuario acceda con sus nuevas credenciales.

En caso de error, se muestra un mensaje de error, para poder corregir el error y así completar el proceso. Los errores más comunes son:

- Si falta algún campo obligatorio, la aplicación muestra un aviso indicando la información pendiente.
- Si el correo ya está registrado, el servidor muestra un mensaje de error y se informa al usuario de que debe emplear otra dirección.
- Ante fallos de conexión, se notifica el mensaje en la app y se invita a reintentar la operación más tarde.

En la Figura 7.3 se ilustra el aspecto de la pantalla de registro en la aplicación.

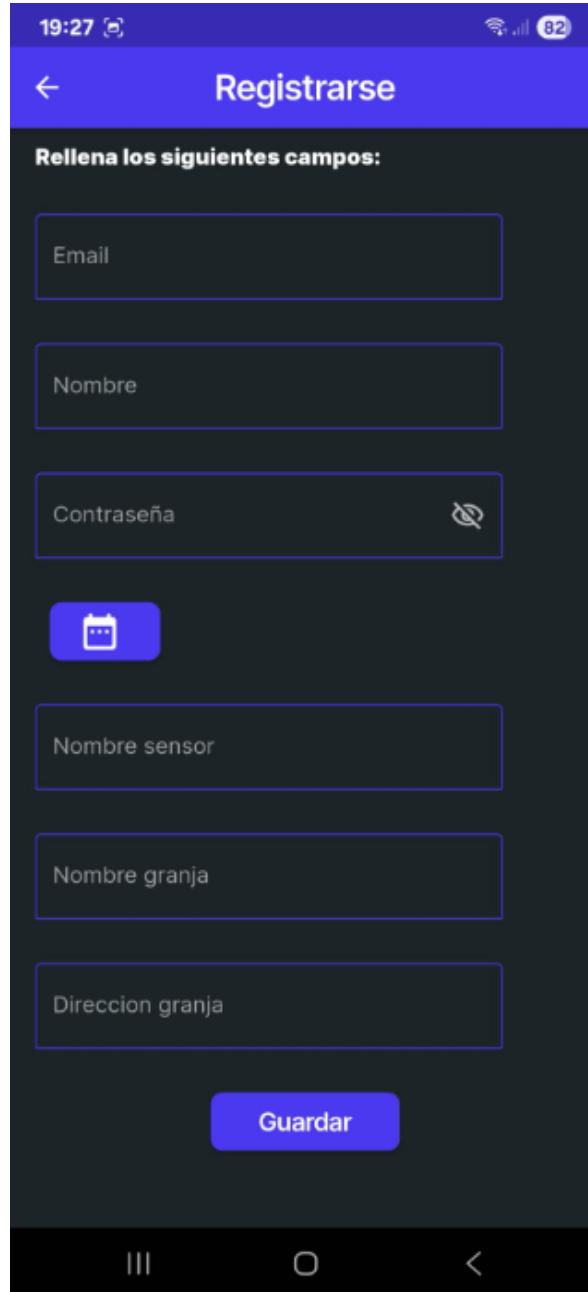


Figura 7.3: Pantalla de registro de usuario de app móvil

7.4. Pantalla principal

La pantalla principal reúne toda la información operativa del sistema. Su diseño se organiza en cuatro bloques verticales:

1. **Condiciones actuales.** Tarjeta de color que muestra, en tiempo real, la temperatura (°C), la humedad relativa (%), y el índice ITH. Los valores se actualizan cada diez segundos mediante una petición GET al servidor. Si el ITH supera el



umbral crítico, la cifra se colorea en rojo y se genera una notificación automática.

2. Estado del sistema. Panel resumen con tres líneas:

- Estado del dispositivo: Conectado o desconectado según la última respuesta del backend.
- Última actualización: Mostrada en formato relativo dependiendo desde la ultima medición cogida.
- Estado de alertas: Activas o silenciadas según la preferencia del usuario.

3. Acciones. Conjunto de botones:

- Generar informe: consulta el histórico y crea un PDF con la evolución de ITH.
- Activar dispositivo: envía un downlink para encender el actuador conectado al sistema.
- Desactivar dispositivo: envía el comando para desactivar el dispositivo conectado.

4. Configuración rápida del sensor. Acceso directo para configurar el modo de activación del dispositivo conectado al sistema y posibilidad de llenar mas información acerca del sensor y establecer el umbral del ITH para su activación automática.

Cada ciclo de actualización sigue estos pasos:

1. La aplicación solicita la última medición al endpoint `/mediciones`.
2. El backend responde con los valores en formato JSON.
3. La interfaz actualiza los tres indicadores y comprueba si el ITH supera el umbral.
4. En caso crítico, se dispara la función de notificación interna y se registra el evento en la base de datos.

El menú desplegable, situado en la esquina superior derecha, permite acceder a las secciones de informe, perfil y ajustes generales sin abandonar la pantalla principal.

En caso de error en la pantalla principal se puedan mostrar mensajes de error como los siguientes:

- Fallo de conexión: se muestra un banner rojo indicando *Servidor no disponible*.
- JSON malformado: se ocultan los datos y se solicita al usuario reintentar la operación.



La Figura 7.4 presenta la apariencia de esta pantalla en ejecución.



Figura 7.4: Pantalla principal de app móvil

7.5. Alertas y notificaciones

La aplicación está diseñada para avisar al usuario cuando se detectan condiciones que pueden suponer un riesgo para el ganado. Estos avisos se presentan como notificaciones del sistema Android, visibles en la barra superior del dispositivo, de manera que el usuario pueda actuar de forma rápida incluso si la aplicación se encuentra cerrada o en segundo plano.



Las notificaciones se envían cuando el valor del Índice de Temperatura y Humedad (ITH) alcanza o supera un umbral predefinido, establecido por defecto en 74.

Este valor puede modificarse desde el apartado de configuración de la aplicación. La detección del valor crítico puede producirse tanto si la app está abierta como si está cerrada, en cuyo caso el servidor envía el aviso mediante un sistema de notificaciones push.

El contenido de la notificación incluye un título breve, como “Estrés térmico alto”, acompañado de un mensaje que muestra el valor exacto del ITH y una recomendación de actuación. Al pulsar sobre la notificación, el usuario es redirigido automáticamente a la pantalla principal de la aplicación, donde se pueden consultar las medidas más recientes. Todo esto se muestra en la Figura 7.5.

La aplicación permite personalizar la llegada de notificaciones desde su menú de ajustes, donde el usuario puede activar o desactivar las notificaciones, así como modificar el umbral de ITH que las genera.

Adicionalmente, desde la configuración del propio sistema Android es posible ajustar opciones como el sonido, la vibración o el modo “No molestar” para el sistema de notificaciones de la app.

Para que las notificaciones funcionen correctamente, es necesario los siguientes requisitos:

- Permiso de notificaciones concedido en el dispositivo.
- Conexión a Internet activa (Wi-Fi o datos móviles).
- Sesión iniciada en la app.

Si no se reciben los avisos, se recomienda seguir los siguientes pasos:

- Revise que las notificaciones estén activadas en Android para esta app.
- Desactive restricciones de batería que impidan la ejecución en segundo plano.
- Compruebe la conectividad del teléfono y del sensor.
- Cierre y vuelva a abrir la sesión para renovar el token de mensajería.

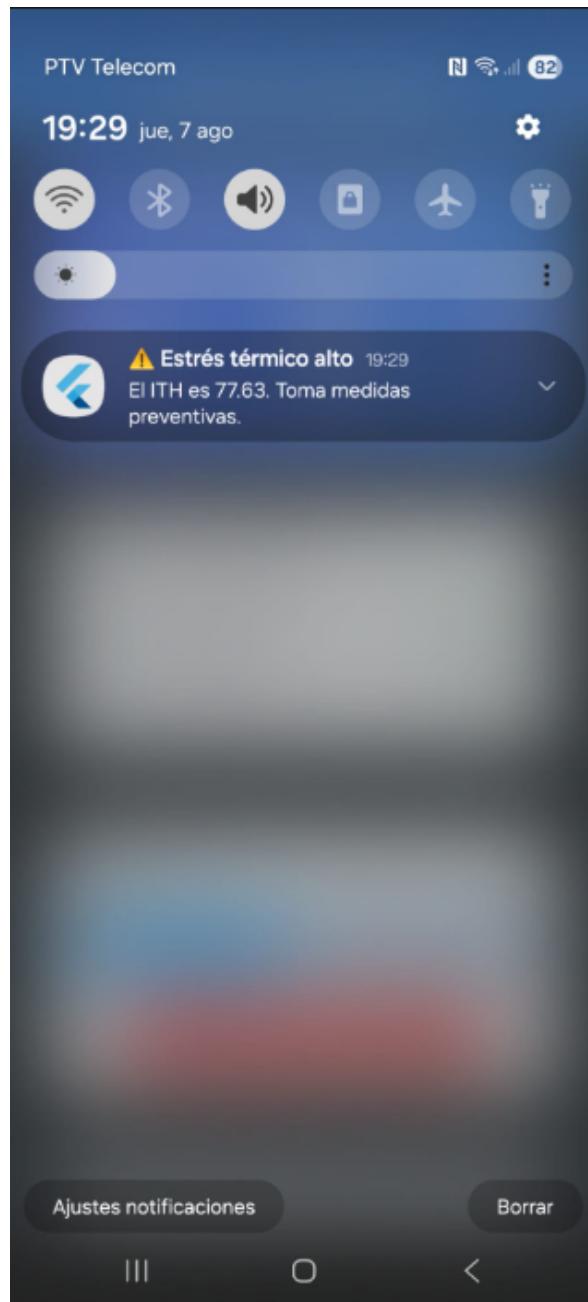


Figura 7.5: Pantalla de notificación de alerta

7.6. Generación de informes en PDF

La aplicación incluye una función que permite generar un informe en formato PDF con las condiciones ambientales registradas en un momento determinado. Este documento recoge de forma estructurada la información clave medida por el sistema, incluyendo la temperatura, la humedad relativa, el valor del Índice de Temperatura y Humedad (ITH) y la fecha y hora de la medición.

El proceso de creación del informe es sencillo. Desde la pantalla principal, el usuario



dispone de un botón identificado como “Generar informe”. Al pulsarlo, la aplicación obtiene la última medición disponible del servidor y compone automáticamente un documento PDF con un diseño claro y legible.

Este informe está pensado para ser utilizado tanto como registro personal como para compartirlo con otros miembros del equipo de la explotación ganadera.

Tal como se muestra la Figura 7.6 se debe seleccionar la opción de como abrir el pdf.

Una vez generado, el PDF se guarda en la memoria interna del dispositivo y se muestra al usuario una opción para abrirlo directamente o compartirlo a través de aplicaciones de correo electrónico o servicios en la nube.

La estructura del documento incluye un encabezado con el título “Informe de condiciones ambientales”, seguido de los valores de las mediciones, un resumen de la fecha y hora de captura de datos y, opcionalmente, el nombre de la granja asociada. Tal como se muestra en la Figura 7.7.

Para que la función de generación de informes funcione correctamente, es necesario que la aplicación disponga de acceso a los datos más recientes y tenga permisos de almacenamiento en el dispositivo. En caso de que no haya conexión con el servidor o no se disponga de mediciones recientes, se mostrará un mensaje informativo indicando que no es posible generar el informe en ese momento.

Esta herramienta resulta especialmente útil para documentar incidencias y llevar un registro histórico de las condiciones de la granja.

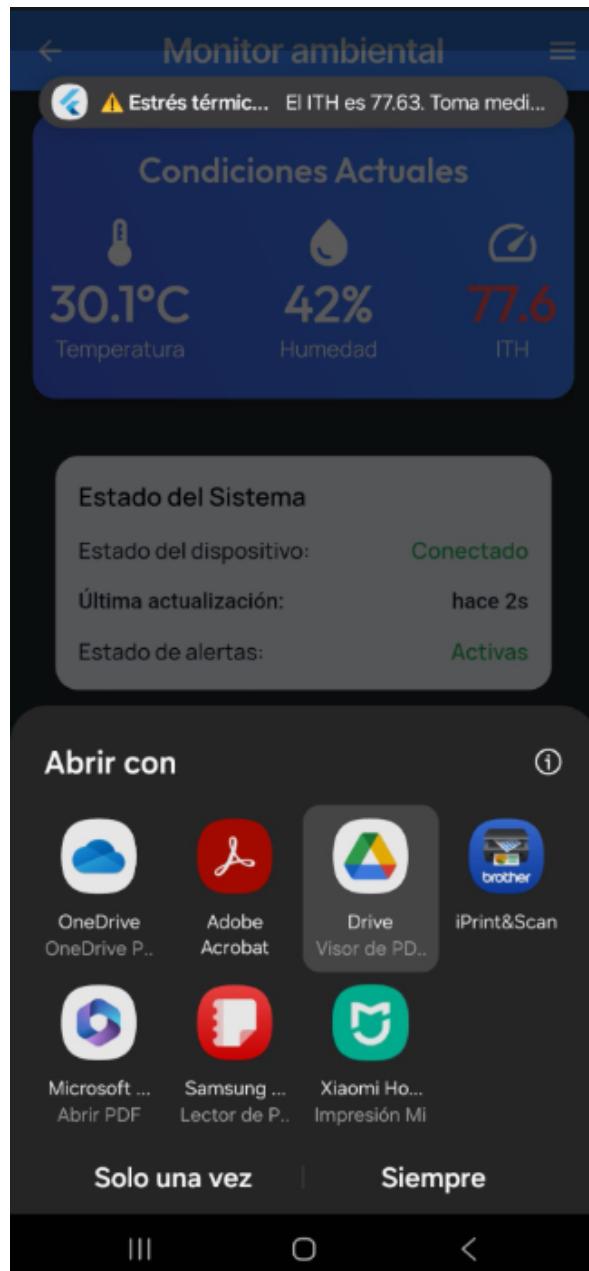


Figura 7.6: Pantalla de selección de “Abrir con” el PDF generado

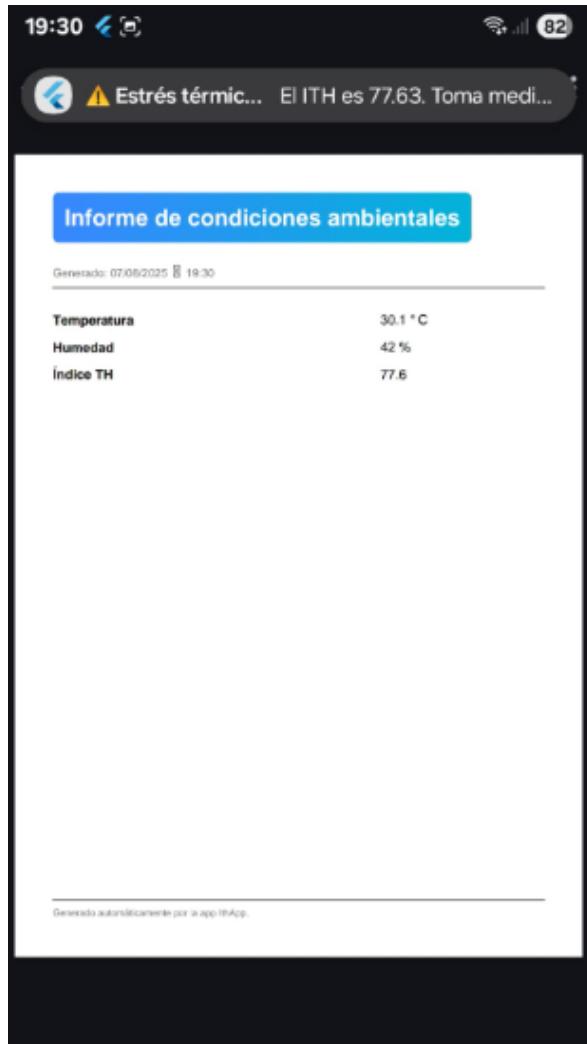


Figura 7.7: Pantalla de PDF generado

7.7. Menú desplegable de navegación

La aplicación dispone de un menú lateral desplegable que facilita el acceso a las opciones de gestión de usuario y a funciones adicionales. Este menú se abre al pulsar el ícono de menú ubicado en la esquina superior derecha de la pantalla principal.

En la parte superior del menú se muestra la dirección de correo electrónico con la que el usuario ha iniciado sesión, lo que permite identificar fácilmente la cuenta activa. Debajo de esta información se presentan varias opciones:

- **Administrar perfil:** Permite visualizar la información personal registrada en la cuenta, como nombre, email, datos de la granja y sensores asociados.
- **Modificar perfil:** abre un formulario donde el usuario puede actualizar sus datos personales o de la granja, cambiar la contraseña o modificar el nombre de los sensores vinculados.



- **Cerrar sesión:** finaliza la sesión actual, eliminando el token de autenticación guardado en el dispositivo y devolviendo al usuario a la pantalla de inicio de sesión.

La navegación a través del menú es sencilla: basta con pulsar sobre la opción deseada para que la aplicación redirija automáticamente a la pantalla correspondiente. Además, el menú puede cerrarse en cualquier momento deslizando de nuevo hacia la izquierda o tocando en cualquier parte fuera del panel. Esta parte de la pantalla se muestra en la Figura 7.8.

Este diseño de menú permite que todas las funciones relacionadas con la gestión del perfil estén siempre accesibles, sin necesidad de abandonar la pantalla principal.

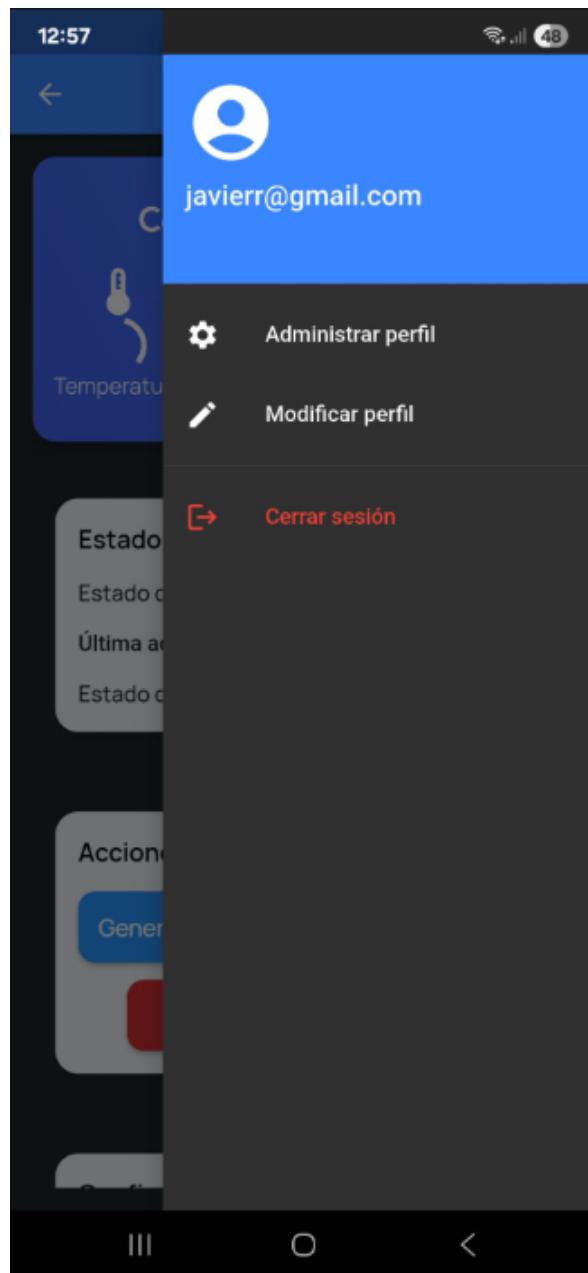


Figura 7.8: Pantalla principal seleccionado el menú desplegable



7.8. Pantalla de administración de perfil

La pantalla de administración de perfil permite al usuario gestionar y personalizar su cuenta de manera sencilla. Se accede a esta sección desde el menú lateral desplegable, pulsando en la opción *Administrar perfil*.

En la parte superior se muestra la dirección de correo electrónico asociada a la cuenta, junto con una imagen de perfil. Este espacio ayuda al usuario a identificar rápidamente la sesión activa.

A continuación, se presentan las opciones disponibles:

- **Modificar perfil:** abre un formulario para actualizar datos personales como el nombre, información de la granja o datos de los sensores.
- **Preferencias:** permite ajustar configuraciones personalizadas de la aplicación, como opciones de visualización o ajustes relacionados con notificaciones.
- **Cambiar contraseña:** ofrece un proceso seguro para modificar la clave de acceso. Suele solicitar la contraseña actual antes de establecer la nueva.
- **Cerrar sesión:** Cierra la sesión actual y devuelve al usuario a la pantalla de inicio de sesión.

La navegación en esta pantalla es directa: basta con pulsar sobre la opción deseada para acceder a su función específica. El diseño está pensado para que todas las acciones importantes relacionadas con la gestión de la cuenta estén centradas en un único lugar. Todo esto se muestra en la Figura 7.9.

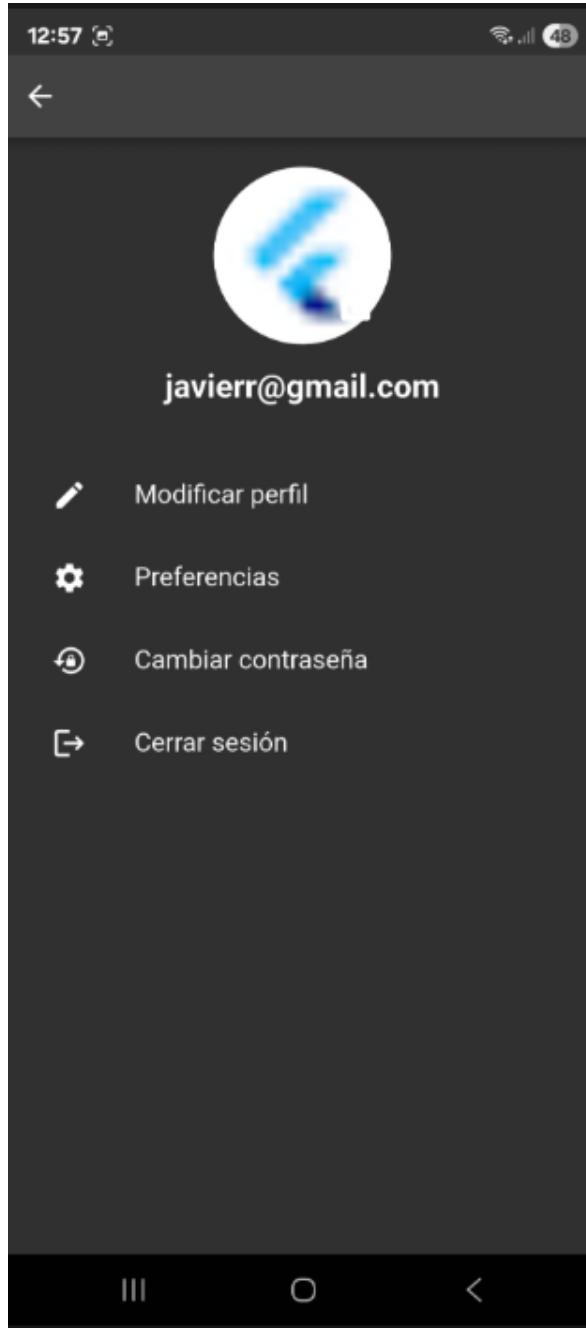


Figura 7.9: Pantalla de Administrar perfil

7.9. Pantalla de preferencias

La pantalla de *Preferencias* permite al usuario ajustar la experiencia de uso de la aplicación según sus necesidades. Se accede desde el menú lateral desplegable, pulsando en la opción *Preferencias*.

En la parte superior se muestran bloques agrupados por temática (Apariencia, Notificaciones, Privacidad, Idioma y Sistema), de forma que la configuración resulte clara



y accesible.

A continuación, se describen las opciones disponibles:

- **Tema:** permite seleccionar el modo de color de la aplicación. Puede seguir el del *Sistema* o fijarse en *Claro/Oscuro*.
- **Unidades de temperatura:** define la unidad a mostrar en la app.
- **Notificaciones push:** interruptor para activar o desactivar la recepción de avisos en segundo plano (por ejemplo, alertas de ITH alto).
- **Analíticas de uso:** permite el envío de métricas anónimas para mejorar la calidad de la aplicación.
- **Abrir ajustes del sistema:** acceso directo a la pantalla de configuración del dispositivo (permisos como notificaciones o ubicación).
- **Idioma de la app:** selecciona el idioma de la interfaz (por defecto, *Español*).
- **Vaciar caché:** elimina datos temporales para liberar espacio y resolver posibles problemas de visualización.
- **Restablecer ajustes:** devuelve las preferencias a sus valores por defecto.
- **Sobre la app:** muestra información de compilación y versión de la aplicación.

La navegación es directa e intuitiva: cada opción se activa con un toque o despliega un selector cuando corresponde. El diseño agrupa las configuraciones más habituales para que el usuario pueda localizar y modificar cada ajuste rápidamente. Todo esto se muestra en las Figura 7.10 y Figura 7.11.

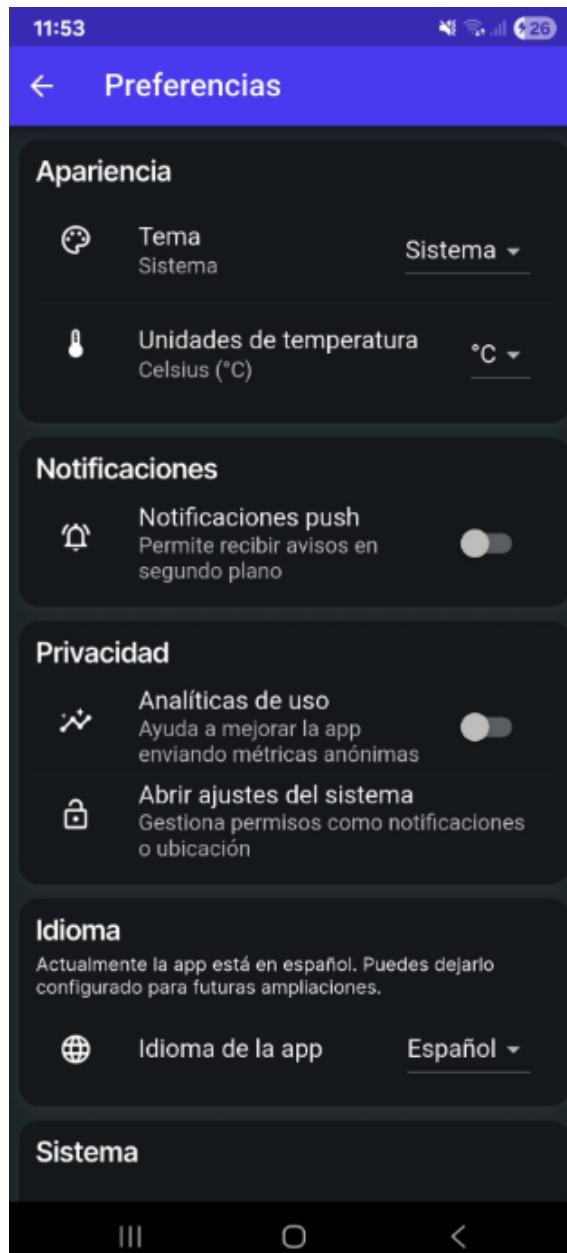


Figura 7.10: Pantalla de Preferencias (1/2)



Figura 7.11: Pantalla de Preferencias (2/2)

7.10. Pantalla de edición de sensor

La pantalla *Editar sensor* permite ajustar la información básica de un sensor ya registrado y definir cómo se comporta el sistema frente al estrés térmico. Se accede desde la pantalla principal.

La vista se organiza en tres bloques:

- **Información del dispositivo.** Incluye el campo Número de serie DEV_EUI) y *Modelo*. Junto al número de serie hay un icono que intenta obtener automática-



mente el último DEV_EUI detectado por el backend (requiere que el nodo haya enviado un *uplink* recientemente).

- **Ubicación.** Campos *Granja*, *Zona* y *Sala/Habitación* para organizar los sensores y facilitar búsquedas y filtros.
- **Modo de operación.** Tiene dos opciones:
 - **Automático (activa ventilador por ITH):** el usuario define un *Umbral ITH* con un control deslizante (por defecto 75). Cuando el ITH calculado supere ese valor, el sistema puede activar el ventilador de manera automática.
 - **Manual:** el ventilador se controla desde la pantalla de medidas (botón de encendido/apagado). Aquí sólo se guarda la preferencia para ese sensor.

Las acciones disponibles en esta página son:

- **Guardar cambios.** Valida los campos mínimos (DEV_EUI y *Modelo*) y actualiza la fila del sensor en la base de datos. Los campos que se persisten son: `dev_eui`, `modelo`, `area`, `zona`, `sala`, `modo` (auto o manual) y `umbral_ith` (sólo si el modo es automático). No se crea un sensor nuevo, simplemente se modifican los datos del existente.
- **Cancelar.** Vuelve a la pantalla anterior sin realizar cambios.

Las Figuras 7.12 y 7.13 muestran la pantalla con sus secciones principales y el selector de modo con el control del umbral ITH.



Figura 7.12: Editar sensor: información del dispositivo y ubicación.



Figura 7.13: Editar sensor: modo de operación y ajuste del umbral ITH.

CAPÍTULO 8

Desinstalación de la aplicación

En caso de que el usuario necesite eliminar la aplicación de su dispositivo, el proceso puede realizarse de forma sencilla desde los ajustes del sistema. A continuación se detalla el procedimiento paso a paso, teniendo en cuenta que el nombre de algunas opciones puede variar ligeramente según el modelo del teléfono y la versión de Android utilizada.

1. Acceder al menú **Ajustes** del dispositivo. Una vez dentro, localizar y pulsar la opción **Aplicaciones** tal y como se muestra en la figura correspondiente. Esta sección contiene la lista completa de aplicaciones instaladas en el dispositivo.

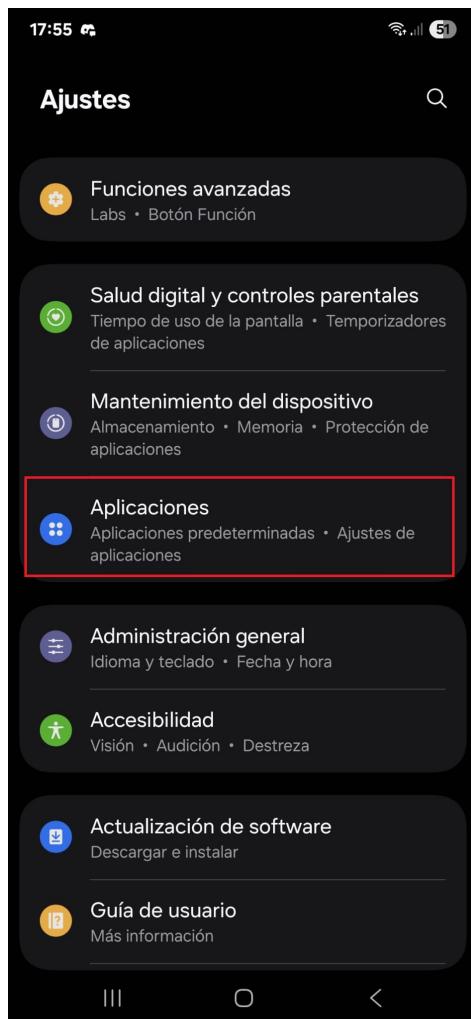


Figura 8.1: Desinstalación de la aplicación: Paso 1



2. Buscar el nombre de la aplicación en la lista. Para facilitar la búsqueda, se puede usar la barra de búsqueda escribiendo el nombre exacto.

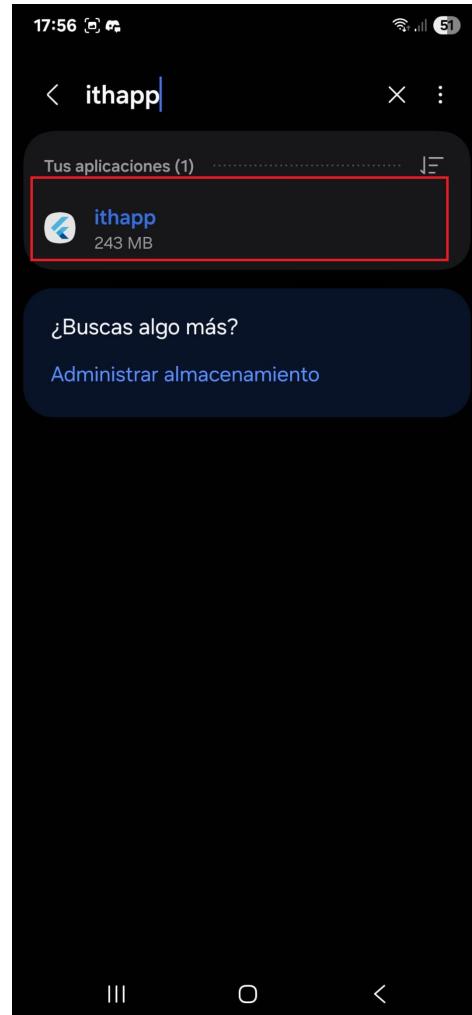


Figura 8.2: Desinstalación de la aplicación: Paso 2

3. Una vez seleccionada la aplicación, se abrirá una pantalla con información detallada sobre su uso. En este apartado, pulsar el botón **Desinstalar**.

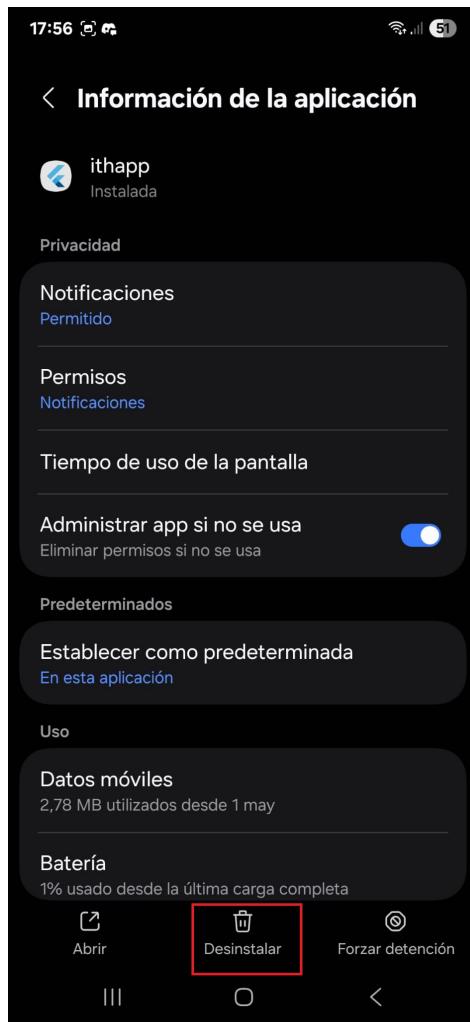


Figura 8.3: Desinstalación de la aplicación: Paso 3

4. El sistema pedirá una confirmación para proceder con la eliminación. Confirmar la acción pulsando nuevamente en **Aceptar** o **Cancelar**.

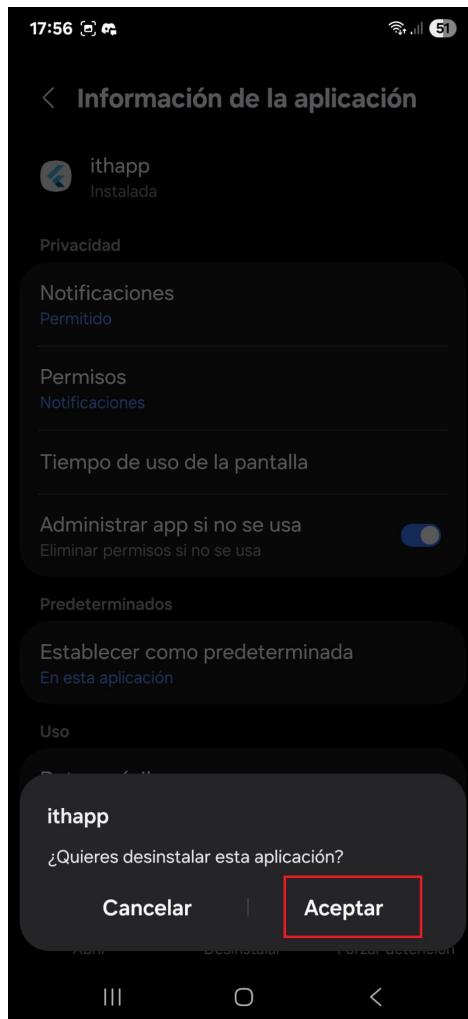


Figura 8.4: Desinstalación de la aplicación: Paso 4

5. El proceso se completará en pocos segundos y la aplicación dejará de estar disponible en el dispositivo.

Este método es válido para la mayoría de dispositivos Android. En versiones más recientes, también se puede desinstalar la aplicación directamente desde la pantalla de inicio, manteniendo pulsado el ícono y seleccionando la opción **Desinstalar**.