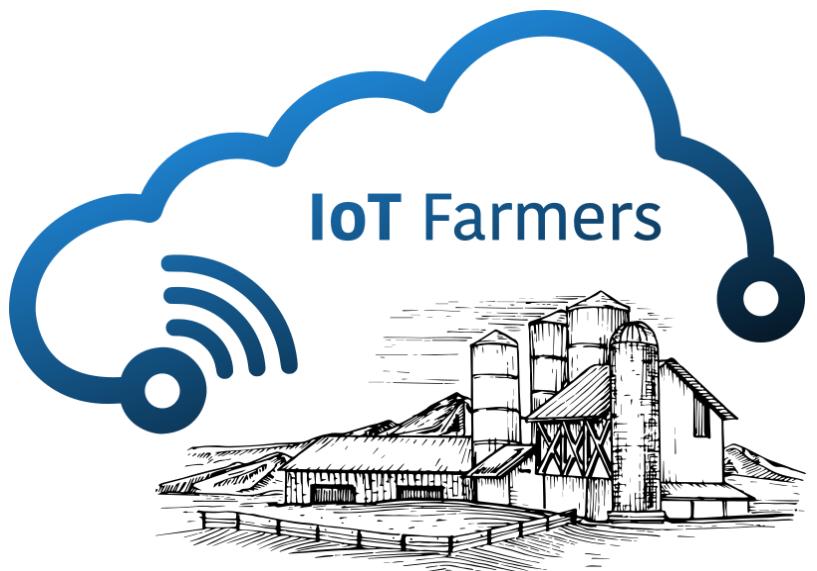


## **PLATAFORMA DE SUPERVISIÓN DE ACTIVOS DE CADENA DE SUMINISTRO Y TRATAMIENTO DE LECHE CIMILK**

Empresa: IoT Farmers.  
Ingenieros: Bonastre, Marc.  
Busing, Ricardo





## ÍNDICE

<b>1. INTRODUCCIÓN</b>	<b>3</b>
1.1 Antecedentes	3
1.2 Objetivos	3
1.2.1 Granjas y Vacas	4
1.2.2 Camiones	4
1.2.3 Planta de Producción	4
<b>2. PROPUESTA TÉCNICA</b>	<b>5</b>
2.1 Cloud	5
2.1.1 Hardware	5
2.1.2 Software	7
2.1.3 Database	9
2.1.4 Dashboard	10
2.2 Granja y Ganadería	11
2.2.1 Arquitectura	11
2.2.2 Hardware	13
Mota ganadería	13
Autonomía	13
Mensaje LoRaWAN	14
Gateway LoRa	15
Unidad de Control	15
Sensor de Temperatura Cuba	15
Sensor de Nivel	15
Sonda de Temperatura, Humedad y CO2	16
Contactor	16
2.2.3 Software	17
Arduino IDE	17
Telegram	17
MQTT	18
2.2.4 Dashboards	19
<b>2.3 CAMIONES</b>	<b>21</b>
2.3.1 Arquitectura	21
2.3.2 Hardware	22
Unidad de Control - Cisterna	22
Sensor de Temperatura	22
Sensor de Nivel	22
Unidad de Control - Camión	22
Módulo 4G	22
Pantalla Táctil	23
Interfaz OBD2	23



2.3.3 Software	23
Arduino IDE C++	24
Python	24
MQTT	25
2.3.4 Dashboards	26
<b>2.4 FÁBRICA</b>	<b>27</b>
2.4.1 Arquitectura	27
Máquinas de climatización	27
Puerta de Acceso a la Planta Productiva	29
2.4.2 Hardware	29
Unidad de Control	29
Sonda de Temperatura, Humedad y CO2	30
Medidor de Energía para las Unidades Interiores	30
Sensor Fotoeléctrico	30
Contactor	30
2.4.3 Software	31
Arduino IDE C++	31
MQTT	32
2.4.4 Dashboards	32
<b>3. PRESUPUESTO</b>	<b>34</b>
3.1 Coste Material	34
3.2 Costes de Ingeniería	36
3.3 Costes de puesta en marcha	36
3.4 Costes de explotación	36
3.5 Viabilidad económica	37
<b>ANEXO: Lista Variables de Supervisión</b>	<b>39</b>
<b>ANEXO: Github</b>	<b>55</b>

# 1. INTRODUCCIÓN

*IoTFarmers* es una empresa joven especializada en el mundo del IoT (Internet of Things) en el que su principal campo de trabajo es el de la interconexión de múltiples dispositivos utilizando herramientas de código abierto y fácilmente escalables con el fin de digitalizar los datos más críticos y con más valor. Esto abre un mundo de nuevas posibilidades a los clientes ya que además de permitirles tener un control completo y en tiempo real de sus procesos pueden optimizar y sacar un mayor provecho de estos.

En este documento se explica la propuesta para el proyecto *CIMILK*, un sistema de monitorización de activos ramaderos en el formato “llaves en mano” empleando las tecnologías de Industrial Internet of Things para la empresa *Receive All Signals Perfectly (RASP)*.

El equipo encargado de este proyecto estará formado por dos especialistas en el campo de la Industria 4.0 y IoT.



## 1.1 Antecedentes

Para la elaboración de esta propuesta técnica se han tenido en cuenta las especificaciones de cliente facilitadas en el pliego de condiciones “Proyecto Transversal IIoT v1.3.pdf” disponible en la plataforma Google Classroom desde el 21 de Febrero de 2021.

## 1.2 Objetivos

El objetivo del proyecto *CIMILK* se basa en la obtención de la información de los elementos clave de todo el proceso. Los objetivos del proyecto se pueden dividir en 3 campos distintos, Granjas y Vacas, Camiones y Planta de Producción.

La solución planteada deberá ser modular y fácilmente escalable para que en un futuro se puedan añadir nuevas soluciones de monitorización.

### 1.2.1 Granjas y Vacas

Se requiere de información de monitorización de todas las vacas de cada una de las 5 granjas que tiene RASP bajo su control, para permitir un futuro análisis sobre el estado de salud de las mismas además de otra información relevante de las granjas. Se requiere una cadencia de información de 15 minutos o menos por cada vaca.

Los puntos a tener en cuenta serán:

- Posicionamiento de las vacas
- Temperatura de la vaca
- Alarma si alguna vaca sufre de algún problema de salud y avisar al granjero
- Estado de las vallas
- Gestión de la refrigeración en verano
- Monitorización de la cuba de la granja
- Consulta de información mediante un bot

### 1.2.2 Camiones

Se requiere de información de monitorización de los camiones que transportan la leche hasta el centro productivo de RASP. Se requiere una cadencia de información de 5 minutos o menos por cada camión.

Los puntos a tener en cuenta serán:

- Posicionamiento de los camiones
- Temperatura de la leche
- Alarma si algún camión lleva algún retraso
- Alarma de revisión del vehículo
- Cantidad de leche que transportan
- Consulta de información mediante un bot

### 1.2.3 Planta de Producción

Se requiere de información de monitorización de los equipos de clima de la fábrica/camiones

que transportan la leche hasta el centro productivo de RASP. Se requiere una cadencia de información de 5 segundos o menos.

Los puntos a tener en cuenta serán:

- Consumo de los equipos de clima
- Temperatura y humedad de las oficinas
- Pulsador en cada camión que haga abrir la puerta del control de acceso solo si está cerca de la fábrica
- Temperatura de las cubas de almacenamiento de la leche
- Alarma cuando una máquina de clima deje de funcionar
- Consulta de información mediante un bot



## 2. PROPUESTA TÉCNICA

La propuesta técnica realizada por *IoTFarmers* está compuesta por 4 categorías distintas:

- Cloud
- Granja y ganadería
- Camiones
- Fábrica

Cada una de estas categorías incluirá un esquema de la solución, una descripción del Hardware y Software empleado y una muestra de los dashboards planteados para la monitorización de los activos así como las diferentes alarmas y métodos de interacción.

### 2.1 Cloud

Los datos se centralizan en el CLOUD de AZURE, se crea una máquina virtual con sistema operativo Ubuntu y dimensionada para trabajar correctamente con el volumen de datos que se manejan en este proyecto. La ventaja de trabajar con un servidor cloud es que a medida que el proyecto vaya creciendo la máquina virtual se puede ir redimensionando para las nuevas necesidades.

Además se ha tenido en cuenta que gracias a que se trabajará en el CLOUD no deberemos tener un mantenimiento de equipos Hardware.

La máquina virtual permite también una alta fiabilidad al estar asegurado por los estándares de calidad de una empresa contrastada como MICROSOFT y en caso de existir cualquier problema podría restituirse en cuestión de minutos.

#### 2.1.1 Hardware

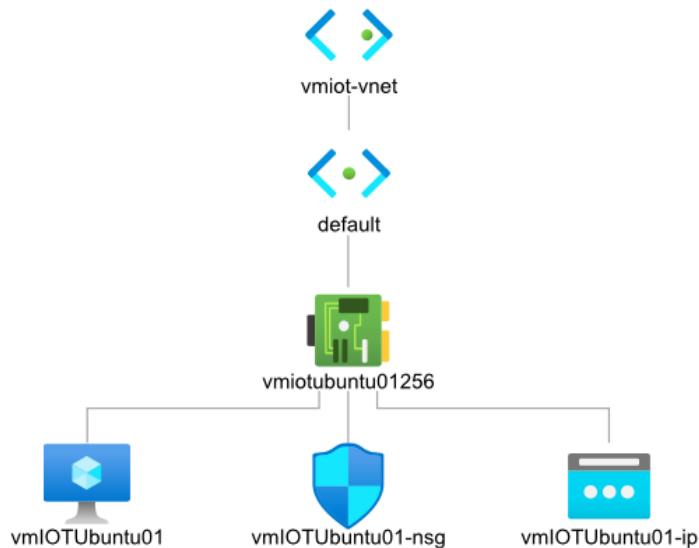
En este apartado no se describe un Hardware convencional dado que por su naturaleza de "cloud" no existe un Hardware al uso. No obstante se detalla la elección de la máquina virtual empleada así como su capacidad.

Para este proyecto se ha escogido en primera instancia una máquina virtual ampliable de la serie B ya que brinda la posibilidad de adquirir un tamaño de máquina virtual con un rendimiento de línea de base que puede acumular créditos si utiliza un rendimiento que no llega a esta línea de base. Cuando la máquina virtual ha acumulado créditos se puede ampliar por encima de la línea de base de esta con un uso de hasta un 100 % de la vCPU si la aplicación necesita el mayor rendimiento de CPU posible.

El hardware virtual empleado es el Standard B2S

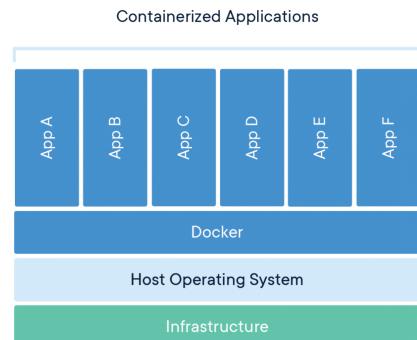
<b>vCPU</b>	2
<b>Memoria: GiB</b>	4
<b>GiB de almacenamiento temporal (SSD)</b>	8
<b>Rendimiento base de CPU de la máquina virtual</b>	40%
<b>Rendimiento máximo de CPU de la máquina virtual</b>	200%
<b>Créditos iniciales</b>	60
<b>Créditos ingresados/hora</b>	24
<b>Créditos máximos ingresados</b>	576
<b>Discos de datos máx.</b>	4
<b>Rendimiento máximo de almacenamiento temporal y en caché: IOPS/Mbps</b>	1600/15
<b>Rendimiento máximo del disco sin almacenamiento en la caché: IOPS/Mbps</b>	1280/15
<b>Nº máx. NIC</b>	3

Con la arquitectura de red virtual mostrada a continuación:



## 2.1.2 Software

En la máquina virtual instalaremos Docker y Docker-Compose para poder trabajar con containers. Un container es una unidad estándar de software que empaqueta el código y todas sus dependencias para que la aplicación se ejecute de forma rápida y confiable de un entorno informático a otro. Una imagen de contenedor de Docker es un paquete de software ligero, independiente y ejecutable que incluye todo lo necesario para ejecutar una aplicación: código, tiempo de ejecución, herramientas del sistema, bibliotecas del sistema y configuraciones.



Las imágenes de containers se convierten en containers en tiempo de ejecución y, en el caso de los containers de Docker, las imágenes se convierten en contenedores cuando se ejecutan en Docker Engine. Disponible para aplicaciones basadas en Linux y Windows, el software en contenedores siempre se ejecutará de la misma manera, independientemente de la infraestructura. Los contenedores aíslan el software de su entorno y garantizan que funcione de manera uniforme a pesar de las diferencias, por ejemplo, entre el desarrollo y la puesta en escena.

Compose es una herramienta para definir y ejecutar aplicaciones Docker de varios contenedores. Con Compose se usa un archivo YAML para configurar los distintos servicios de la aplicación. Luego, con un solo comando, crea e inicia todos los servicios desde su configuración.

Compose funciona en todos los entornos: producción, puesta en escena, desarrollo, pruebas y flujos de trabajo de CI.

El uso de Compose es básicamente un proceso de tres pasos:

1. Definir el entorno de la aplicación con un Dockerfile para que pueda reproducirse en cualquier lugar.
2. Definir los servicios que componen su aplicación en docker-compose.yaml para que puedan ejecutarse juntos en un entorno aislado.
3. Ejecute docker-compose up para iniciar y ejecutar toda su aplicación.

Los containers que instalaremos son Node-RED, Eclipse Mosquitto™, TimeScale DB y Grafana.

### Node-RED

Node-RED proporciona un editor de flujo basado en navegador que facilita la conexión de flujos utilizando la amplia gama de nodos de la paleta. Luego, los flujos se pueden implementar en el tiempo de ejecución con un solo clic.



Las funciones de JavaScript se pueden crear dentro del editor. Además incorpora una biblioteca que permite guardar funciones, plantillas o flujos útiles para su reutilización.

Desde Node-RED gestionaremos toda la información que queremos mandar a las motas y nos suscribiremos a todos los topics que las motas publican para guardarlas en la base de datos y así poder tener un histórico.

### Eclipse Mosquitto™

Eclipse Mosquitto es un agente de mensajes de código abierto (con licencia EPL / EDL) que implementa las versiones 5.0, 3.1.1 y 3.1 del protocolo MQTT. Mosquitto es liviano y adecuado para su uso en todos los dispositivos, desde computadoras de placa única de baja potencia hasta servidores completos.



El protocolo MQTT proporciona un método ligero para realizar mensajes mediante un modelo de publicación / suscripción. Esto lo hace adecuado para la mensajería de Internet de las cosas, como con sensores de baja potencia o dispositivos móviles como teléfonos, computadoras integradas o microcontroladores.

### TimeScale DB

TimescaleDB es una base de datos relacional de código abierto para datos de series de tiempo. Habla "SQL completo" y, en consecuencia, es fácil de usar como una base de datos relacional tradicional, pero escala de formas previamente reservadas para las bases de datos NoSQL.



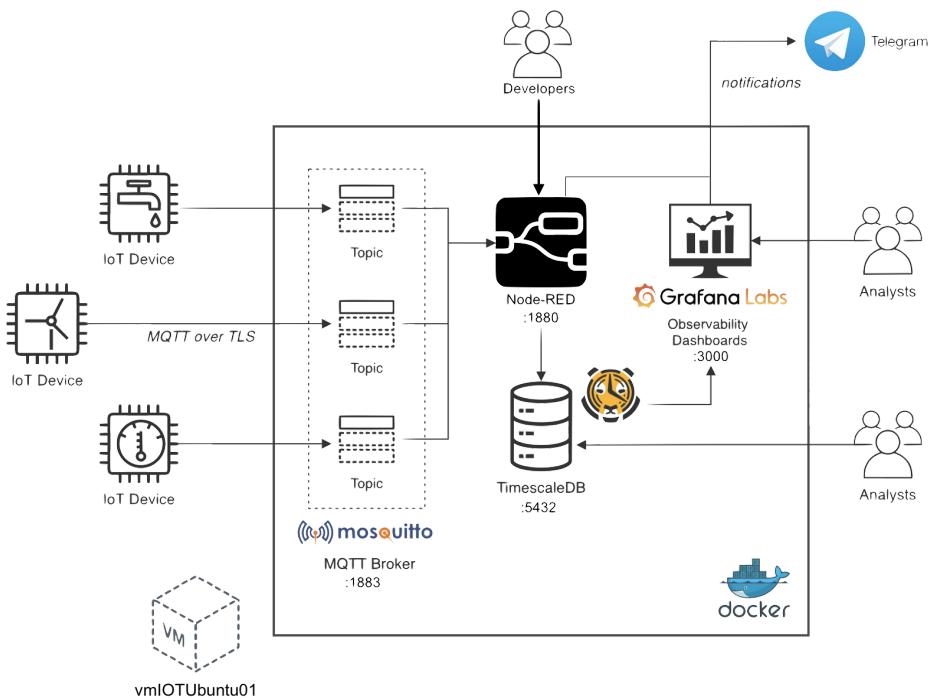
En comparación con las compensaciones exigidas por estas dos alternativas (relacional frente a NoSQL), TimescaleDB ofrece lo mejor de ambos mundos para los datos de series de tiempo.

### Grafana

Grafana es una herramienta de código abierto para el análisis y visualización de métricas.

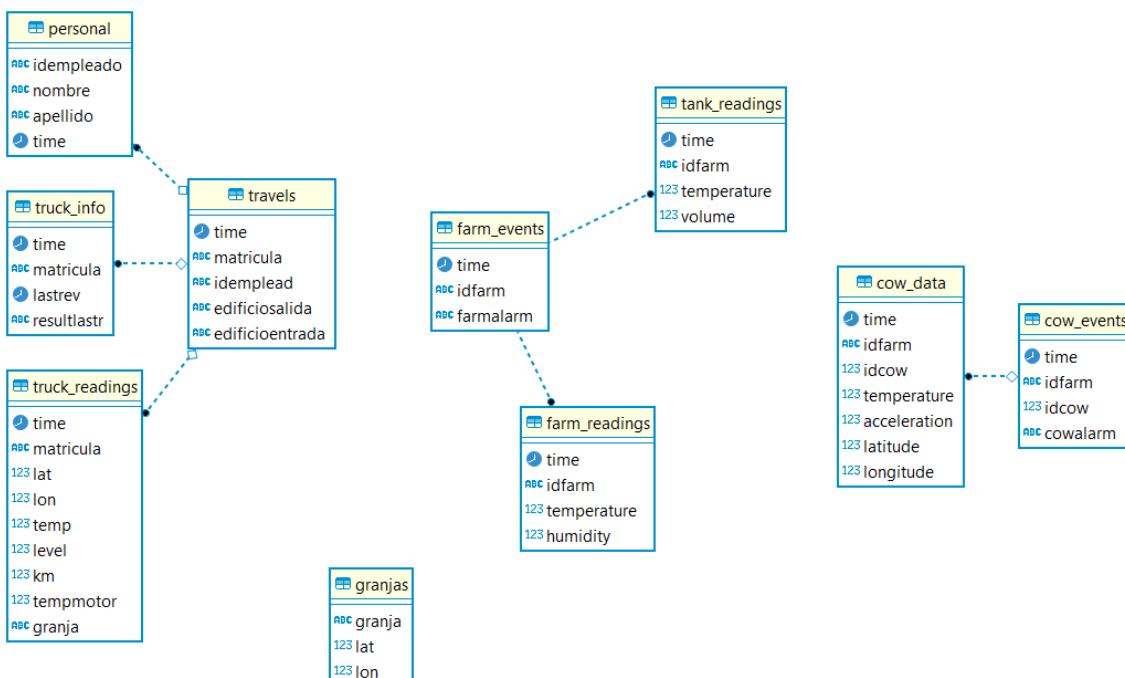


A continuación se muestra un esquema para entender cómo quedará estructurada toda la parte de los containers con los que se trabajará y los puertos por los que podremos acceder a ellos. Todos los puertos usados en los containers también los tenemos abiertos en la máquina virtual para poder acceder a ellos en cualquier momento.



## 2.1.3 Database

Como se ha comentado en el apartado de Software la BBDD empleada es un TimeScaleDB. De momento el esquema planteado de la DDBB es el siguiente:





## **Granja**

La información se almacena en dos tablas, una referente a los tanques de almacenamiento y otra a los controles de clima de la zona de ordeño.

En caso de producirse una alarma en cualquiera de las 2 tablas esta se vuela en la tabla de "eventos".

Los valores almacenados son temperatura y humedad de la zona, para posteriormente explotarlos y poder realizar un estudio de si existe relación entre la temperatura de ordeño y la calidad final del producto.

Lo mismo ocurre con la tabla del ganado, cada vez que se genera una alarma referente a los parámetros que se están monitorizando se genera una alarma en la tabla de eventos.

## **Camiones**

En el caso de los camiones la estructura es un poco más compleja, registrando en todo momento los parámetros que se leen en tiempo real en una tabla y cruzandolos con la tabla de registro de empleados así como el histórico de revisiones del camión.

## **Fábrica**

En el caso de la fábrica se almacena en una tabla los consumos de los diferentes equipos de clima y la temperatura de las diferentes salas.

### **2.1.4 Dashboard**

Se crea un dashboard en Grafana para poder visualizar el estado de la máquina virtual de Azure. De este modo podremos ver con facilidad si necesitamos aumentar la capacidad de nuestra máquina o de si está funcionando correctamente.



## 2.2 Granja y Ganadería

### 2.2.1 Arquitectura

La automatización de la captación de datos y de la toma de decisiones en la granja se compone de dos sistemas diferentes. Estos subsistemas son el sistema de ganadería, en el cual se monitoriza uno de los activos más importantes de la empresa RASP, que es el ganado, y la zona del edificio en la cual se efectúan las actividades de ordeño y almacenaje.

El subsistema de ganadería se basa en uno de los principios del Industrial Internet of Things que es la monitorización de un sistema aislado que debe cumplir con el requisito de autonomía energética y por tanto también debe minimizar en todo momento el consumo de los componentes.

Para cumplir estas dos premisas uno de los elementos claves es la elección del protocolo de comunicación que se emplea, al ser este uno de los aspectos que mayor consumo comporta. Por ello la tecnología elegida se pasa en las Low Power Wide Area Networks(LPWAN), concretamente en la tecnología LoRa y el protocolo LoRaWAN.

LoRa es una tecnología inalámbrica que utiliza un tipo de modulación en radiofrecuencia, como la AM o la FM o el PSK; pero patentado por Semtech una importante empresa fabricante de chips de radio.

Entre las principales ventajas de LoRa se encuentran las siguientes:

1. Alta tolerancia a las interferencias
2. Alta sensibilidad para recibir datos (-168dB)
3. Basado en modulación “chirp”
4. Bajo Consumo
5. Largo alcance 10 a 20km
6. Baja transferencia de datos (hasta 255 bytes)
7. Conexión punto a punto

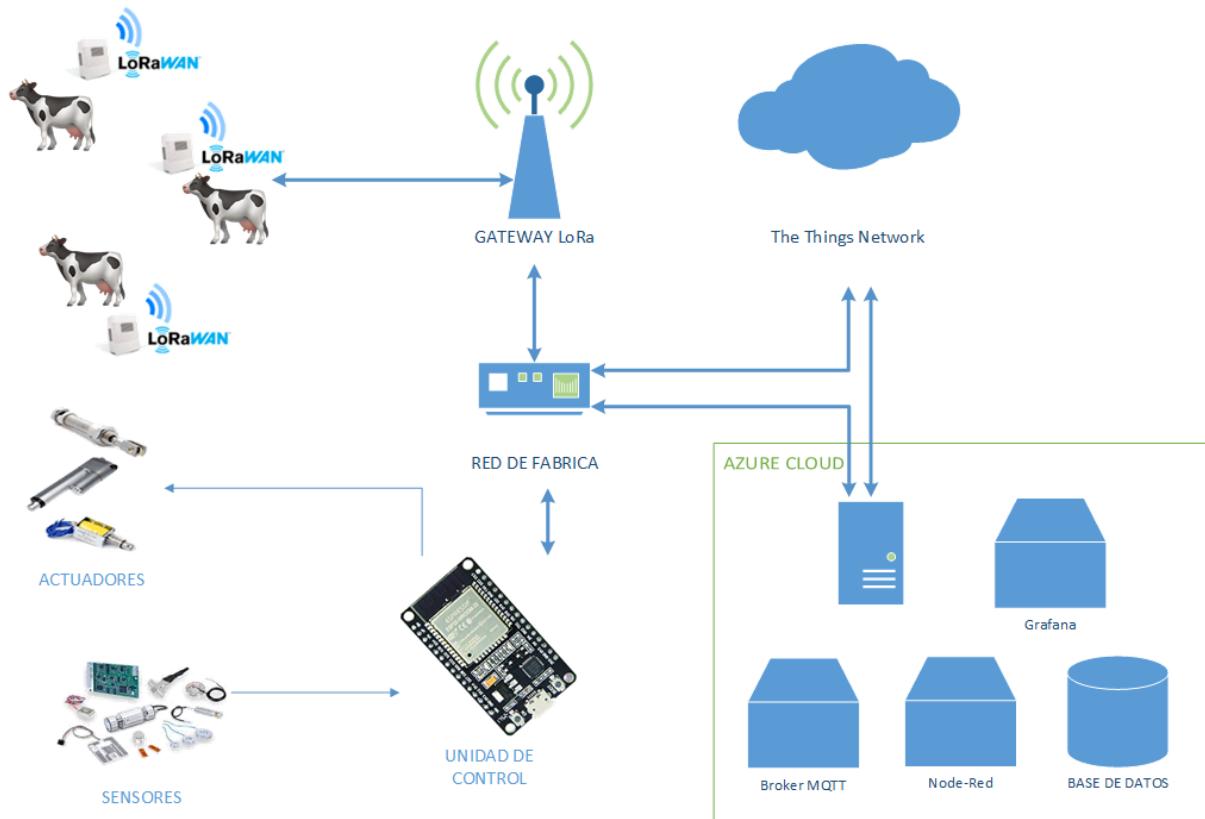
LoRaWAN es un protocolo de red que usa la tecnología LoRA integrando gateways (antenas) y nodos (dispositivos).

1. nodos (dispositivos): son los dispositivos finales que envían y reciben información hacia el gateway. Los nodos en este caso serán los sensores inteligentes que portará el ganado.
2. gateways (antenas): son los encargados de recibir y enviar información a los nodos desde el servidor. Es decir traducen la información de una red ya totalmente implantada a nivel mundial como es “el Internet” (popularmente hablando) a la red LoRaWAN.

Por otro lado nos encontramos con el subsistema del edificio de la granja, el cual consta básicamente de una unidad de control central que gestiona los diferentes sensores y actuadores repartidos por la granja y se ocupa de gestionar el envío de dichos datos al

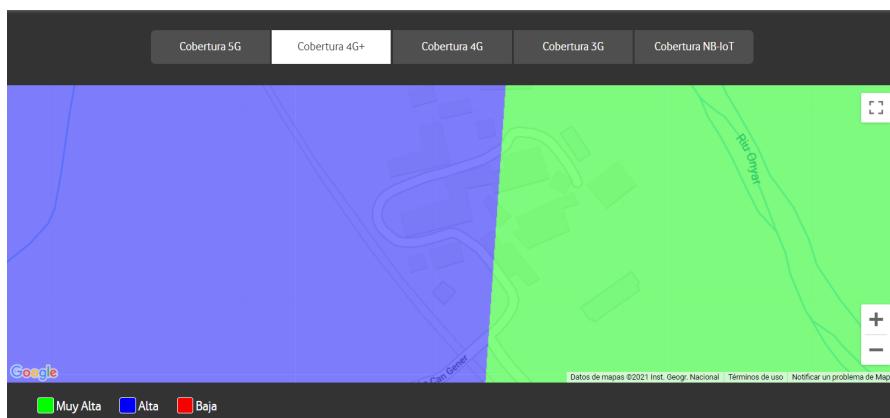
subsistema de gestión centralizado de la información (Cloud) explicado en el apartado anterior.

La arquitectura explicada anteriormente se ve plasmada en la siguiente figura.



Esta arquitectura se repetiría en cada una de las granjas, siendo la transmisión TCP/IP interna sobre WiFi, y entre la red de fábrica-cloud por la conexión a internet existente en la granja.

En el caso de la granja de Mas Gener, que no cuenta con conexión a internet, el único cambio sería que se contraría un módem 4G para la transmisión de datos de planta y de esta forma evitar la instalación de un cableado físico que elevaría considerablemente el presupuesto. Esto sería posible dada la gran cobertura existente en la zona como se puede observar en la siguiente figura.



## 2.2.2 Hardware

En el apartado de HW encontramos 5 grupos:

1. Mota ganadería.
2. Gateway LoRa.
3. Unidad de control
4. Sensores
5. Actuadores

### Mota ganadería

La mota de ganadería es una solución de la empresa SODAQ, concretamente el Solar Powered LoRa Cattle tracker V2. Este dispositivo cuenta con GPS, para realizar el seguimiento de los activos, chip de transmisión LoRaWAN para enviar los datos a la nube, sensor de temperatura integrado así como un acelerómetro y magnetómetro. En cuanto a aprovisionamiento energético la mota permite una autonomía total al integrar una batería de 2200mAh y un panel solar de 0.5W para autoabastecerse durante un período de 3 años.



El equipo cuenta con una correa incorporada testeada en diferentes tipos de ganado (vacas, caballos y cabras), así como un contrapeso para asegurar que el dispositivo se posicione siempre con la placa solar bien orientada.

GPS	LoRa	Sensor Temperatura	Acelerómetro	Energía
EVA 8M	RNE 868	BME 280	LSM303AGR	Batería de 2200 mAh Panel Solar de 0.5W

Para este dispositivo es de especial importancia el cálculo de la autonomía en caso de encontrarnos con días sin sol o encontrarse averiado el sistema de recolección de energía.

### Autonomía

Para efectuar dicho cálculo se remite a cada uno de los componentes del dispositivo, obtenidos de la web del fabricante SODAQ, y posteriormente a los datasheet de los componentes.

Componentes	Corriente en funcionamiento ( $\mu$ A)	Corriente en DEEP SLEEP ( $\mu$ A)
TEMPERATURA	1,80E+00	1,00E-01
ACELERÓMETRO	5,00E+01	2,00E+00

MICRO	3,98E+03	1,00E+02
LORA	1,21E+05	3,20E+00
GPS	1,70E+04	2,00E+01
Total ( $\mu$ A)	1,42E+05	1,25E+02
Total (mA)	142,03	0,13

Considerando la cadencia mínima de datos en 15 min y un tiempo de reconexión del gps de 30s, el dispositivo estaría trabajando un 3,4% del tiempo y un 96,6% en deep sleep.

Por tanto el dispositivo consumirá 4,86 mAh, lo que nos daría una autonomía de 453h lo que equivale a 18 días. En caso de disponer de energía solar el dispositivo puede funcionar a un régimen de mensajes más elevado, cuyo límite vendrá determinado por el siguiente apartado relativo a los mensajes LoRa.

#### Mensaje LoRaWAN

El uso de este protocolo de comunicación implica seguir una serie de normas para la correcta explotación de la banda de frecuencia de radio sobre la que se sustenta. TheThings network establece que el límite de mensajes en el aire es de 30s/día.

Teniendo en cuenta la cadencia de datos requerida se estaría hablando de una tasa de interacción de 96 mensajes diarios y en caso de tener disponibilidad del recurso solar de 384 actualizaciones diarias.

Teniendo esto en cuenta esto nos encontramos con que el TimeOnAir para los 96 mensajes diarios es de 312ms y para los 288 (mensaje cada 5min) de 104,17ms.

A un SpreadingFactor SF7, la región de emisión EU868 y el ancho de banda de 125kHz, esto implicaría que podríamos transmitir respectivamente un payload de 182 bytes y de 41 bytes respectivamente.

Restando los bytes de cabeceras propios de LoRaWAN (13 bytes) quedarían en 169 bytes y 28 bytes respectivamente.

	Low Power Mode	High Performance Mode
Tasa de actualización [min]	15	5
Mensajes diarios [nº]	96	288
TimeOnAir/mensaje [ms]	312,5	104,17
MAX Payload size [bytes]	182	41
Useful Payload size [bytes]	169	28

En este caso el payload debe estructurarse para el caso más crítico que son los 28 bytes. En caso de estar trabajando en LowPower mode se podrían añadir datos como podrían ser el tiempo transcurrido desde la última carga o el nivel de batería actual.

## Gateway LoRa

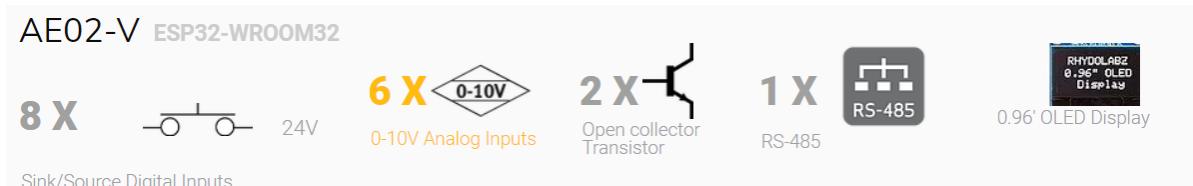
El hardware empleado es The Things Outdoor Gateway - 868 Mhz. TTOG es una puerta de entrada robusta para exteriores y está diseñada para la longevidad en todos los climas. Cuenta con backhaul Ethernet y 3G / 4G, con GPS integrado. Con grado de protección IP67 e incluyendo todo el kit de montaje, y compatible con cualquier servidor LoRaWAN.



Esta puerta de enlace se configurará vía la interfaz web que integra.

## Unidad de Control

Como unidad de control se empleará una **Norvi IIOT-AE02-V**, una solución basada en ESP32 que está preparada para entornos industriales. Dispone de entradas y salidas compatibles con los niveles de voltaje que se pueden encontrar en la industria y un puerto de comunicación Modbus RS485. Además viene encapsulado y con la posibilidad de montaje en carril DIN. Incorporando una pantalla para mostrar mensajes para pequeñas interacciones con el usuario.



## Sensor de Temperatura Cuba

El sensor de temperatura que se usará para medir la temperatura de la leche en la cuba de almacenamiento es una Pt100 con tubo de cuello. Se trata del modelo **HTFB3/MUV/100** de la marca **TITEC**. Es IP54 con lo cual puede instalarse en cualquier entorno de una granja, incluyendo ala intemperie. El rango de medición es de -50°C a 600°C y se lee mediante una señal 0-10V.



## Sensor de Nivel

El sensor usado para medir el nivel de la cuba de almacenamiento es un sensor de ultrasonidos con un rango de medición de 350 a 6000mm. La medición se puede hacer desde una señal 0-10V. Este sensor es del fabricante LOXONE.



## Sonda de Temperatura, Humedad y CO2

Para controlar las condiciones de la zona de ordeño se colocará una sonda de Temperatura, Humedad y CO2. En este caso todo y no ser estrictamente necesario el sensor de CO2 para estandarizar con la instalación de fábrica se instalará el mismo sensor que se verá



posteriormente. La sonda seleccionada es el modelo **HDH-M-RH** de la marca **Produal**, una sonda de pared que dispone de salidas 0-10V para las señales pero que además incorpora comunicación por Modbus RS485.

### **Contactor**

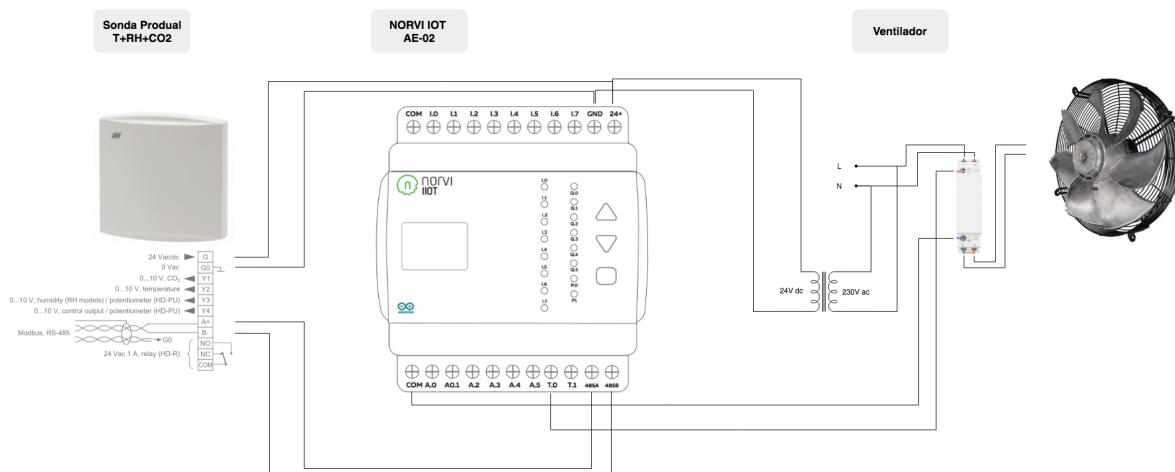
Un contactor es un elemento electromecánico que tiene la capacidad de establecer o interrumpir la corriente eléctrica de una carga, con la posibilidad de ser accionado a distancia mediante la utilización de elementos de comando, los cuales están compuesto por un circuito bobina / electroimán por la cual circula una menor corriente que la de carga en sí (incluso podría utilizarse baja tensión para el comando).

En este proyecto para activar la carga (ventilador) se empleará un contactor de Schneider Electric LC1D09BD, cuyas especificaciones son:

- Corriente asignada para el contactor en AC3: 9A.
- Corriente asignada para el contactor en AC1 (I<sub>th</sub>): 25A.
- Potencia máxima (kW) 380-400Vac: 4kW.
- Potencia máxima (kW) 220-230Vac: 2,2kW.
- Contactos auxiliares: 2 NA+NC (1 abierto y 1 cerrado)



En el siguiente esquema eléctrico se puede observar la conexión de los elementos



### 2.2.3 Software

#### **Arduino IDE**

Arduino IDE es un conjunto de herramientas de software multiplataforma (Windows, Linux, iOS) que permiten a los programadores desarrollar y grabar todo el código necesario en la placa de control. El IDE de Arduino nos permite escribir, depurar, editar y grabar nuestro programa (llamados



“sketches” en el mundo Arduino) de una manera sumamente sencilla, lo cual ha hecho de esta plataforma todo un éxito gracias a la accesibilidad.

Arduino no solo permite interaccionar con placas propias, sino que con la ayuda de núcleos de terceros, se puede usar con placas de desarrollo de otros proveedores como es el caso de este proyecto, en el cual se programará una placa de “ESPRESSIF” empleando este IDE.

Arduino IDE será la plataforma de desarrollo de la ESP32 y por tanto se programará toda la gestión de los sensores y actuadores de la granja, las comunicaciones ya sean WiFi (en el caso de los dispositivos de los edificios) o LoRa en el caso de las motas del ganado. De igual manera será en esta plataforma donde se programará toda la parte de comunicación en los protocolos MQTT y LoRaWAN respectivamente.

### **Telegram**

Telegram es una aplicación de mensajería enfocada en la velocidad y seguridad, que permite al usuario utilizarlo desde diferentes plataformas. Actualmente esta aplicación cuenta con más de 500 millones de usuarios activos mensuales y es una de las 10 apps más descargadas del mundo.



La aplicación permite enviar mensajes, fotos, videos y archivos de cualquier tipo (doc, zip, mp3, etc.), como también crear grupos de hasta 200.000 personas o canales para hacer difusiones a audiencias ilimitadas. Puedes escribir a tus contactos del teléfono y encontrar personas a través de sus nombres de usuario. Como resultado, Telegram es como el SMS y el correo electrónico combinados, y puede satisfacer todas las necesidades de mensajería personal o de negocios.

No obstante lo que ha llevado a Telegram a ser empleado en este proyecto es que nos permite interaccionar con otros entornos con una gran facilidad y seguridad, lo que nos brinda la oportunidad de enviar notificaciones desde nuestros entornos de monitorización o actuar como interfaz de usuario para enviar órdenes al sistema.

Por tanto se empleará telegram para ejecutar la orden de encendido “manual” de los ventiladores de la zona de ordeño.



¿Qué puede hacer este bot?  
Este bot es propiedad de la empresa iotFarmers. Se usa para avisos de alarmas y para dar órdenes en la planta.

/MasAlmarFAN - Turn ON/OFF FAN  
/MasBadosaFAN -Turn ON/OFF FAN  
/MasGenerFAN -Turn ON/OFF FAN  
/MasColomFAN -Turn ON/OFF FAN  
/Temperature - GetTemperature

You are welcome: [REDACTED]  
Your chat id is [REDACTED]

/MasAlmarFAN 16:20 ✓✓

Ricardo  
/MasAlmarFAN  
QUIERES ENCEDER o APAGAR EL VENTILADOR? 16:20

TURN ON      TURN OFF

Esta interacción vía telegram es recibida en el Cloud y reenviada vía MQTT a los dispositivos de fábrica de forma que el ventilador se podría accionar de forma remota.

## MQTT

En el apartado de “Cloud” se ha comentado que es el protocolo MQTT, en este apartado se definirán los topics que se han desarrollado para la gestión de las granjas.

En el caso del control del edificio el Topic principal será:

`FARM/<idGranja>/<TipoElemento>/<idElemento>`

- `<idGranja>`: este elemento es el identificador único de granja donde se ubicará el elemento física a monitorizar, es decir : “MasAlmar”, “MasBadosa”, “CanSiset”, “MasAlmar” y “MasBadosa”. En caso de abrirse una nueva granja únicamente se debería publicar en un nuevo id.
- `<TipoElemento>`: identificador del elemento físico que se va a medir u accionar, en este caso los disponibles son: “Cuba”, “Estancia”, “Fan”, “ESP” pero sería fácilmente ampliable en caso de requerirse un nuevo elemento de monitorización. El elemento “ESP” se emplearía para enviar algún mensaje de configuración o actualización a la unidad de control.

- <idElemento>: este elemento no se emplea para la prueba de concepto, pero sí se desarrollaría en el caso de implementar la solución definitiva, permitiendo escalabilidad en caso de ampliar el número de cubas, el número de estancias de ordeñado o la cantidad de ventiladores. Este identificador permitiría actuar u obtener los datos de un elemento concreto.

Un ejemplo sería:

`Farm/MasAlmar/Cuba/1` → Suscripción/Publicación a la cuba 1 de la granja MasAlmar.

En caso de querer suscribirse a todas las cubas de todas las granjas:

`Farm/+/Cuba/#`

En el caso de las vacas el topic no estaría tan bien definido, dado que viene dado por el formato que permite el broker MQTT de TTN.

`<AppID>/devices/<DevID>/up`

`<AppID>` sería el nombre de la aplicación en TTN, en este caso se propone usar el nombre del proyecto como AppID “CIMILK”.

`<DevID>` nombre del dispositivo, en este caso sería la MAC del dispositivo.

Un ejemplo por tanto sería: `CIMILK/devices/00:1B:44:11:3A:B7/up`

Para suscribirse a todas las motas tendríamos por tanto:

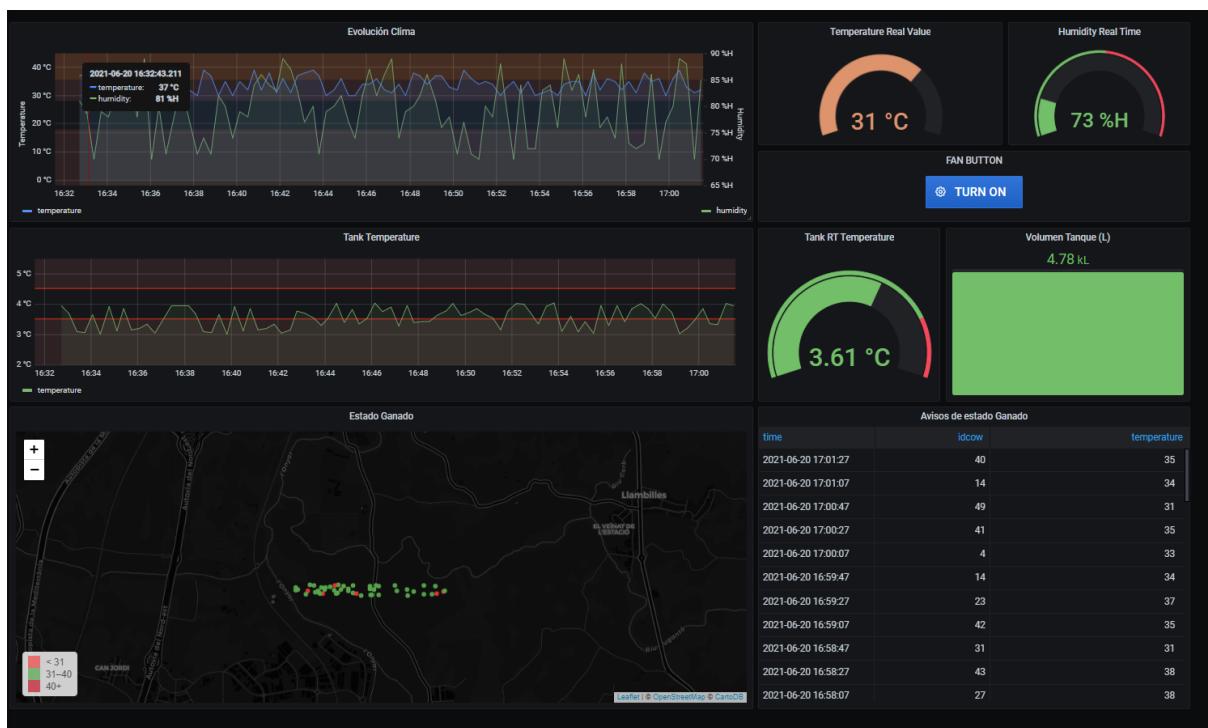
`CIMILK/devices/+/up/#`

Para enviar comandos se deberían publicar al topic:

`CIMILK/devices/<DevID>/down/`

## 2.2.4 Dashboards

Se ha creado un dashboard para cada granja, en el cual se pueden observar los parámetros de monitorización en tiempo real de la temperatura y la humedad de la estancia así como su evolución temporal. También se permite al usuario activar el ventilador de forma manual y se monitoriza el tanque de almacenamiento de leche, tanto en cuanto a nivel de producto como su temperatura.



Por otro lado también se muestra un mapa con la ubicación de cada una de las vacas de la granja, pintando de diferentes colores en función de si tiene un problema de salud o se encuentran fuera del perímetro marcado.

Por último se muestra una tabla donde se ven únicamente los valores de monitorización del ganado que se han desviado de lo considerado un buen indicador de salud del animal.

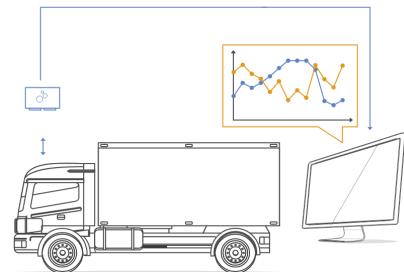
## 2.3 CAMIONES

### 2.3.1 Arquitectura

En el caso de los camiones lo vamos a diferenciar en dos elementos distintos, el control de la cisterna, que se utilizará tecnología basada en ESP32 y luego el control del camión en sí que utilizará como sistema de control una Raspberry Pi.

El camión dispondrá de una Raspberry Pi que tendrá conexión a internet gracias a un HAT de 4G. Este módulo además incorpora localización GPS con lo cual conoceremos la ubicación de los camiones en todo momento. Además la Raspberry Pi la encapsularemos en un estuche con un monitor de 7”, donde se le mostrará al conductor la ruta que debe seguir y unos botones para mandar la señal de abrir y cerrar la puerta de la planta de producción y la de las granjas.

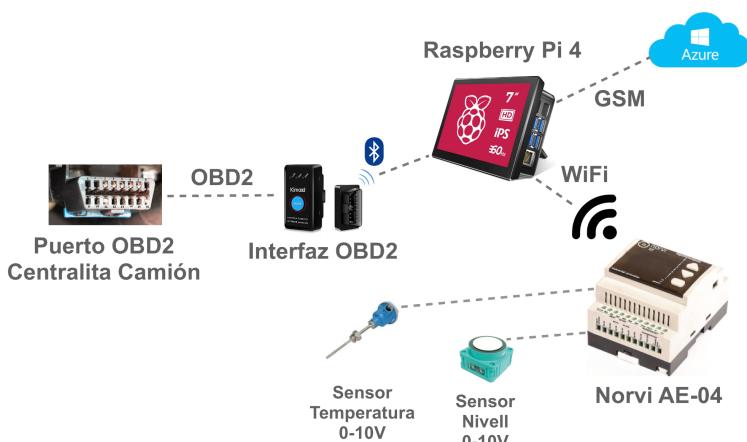
Todos los vehículos incorporan un conector OBD2. El OBD2 es un sistema integrado en el vehículo para hacer autodiagnósticos. Por lo tanto, desde este conector se podrá leer fácilmente la información del camión, extrayendo información como kilómetros recorridos, velocidad, consumo, temperatura del motor, posibles fallos del vehículo, etc..



Para extraer toda la información, usaremos una interfaz OBD2 con bluetooth incorporado que se conectará a la centralita del vehículo. Mediante la Raspberry, como dispone de bluetooth podremos leer todos los datos y mandarlos al Cloud.

En la cisterna se colocará una sonda de temperatura para poder controlar que no se rompa la cadena de frío de la leche y un sensor de nivel para saber cuantos litros está transportando el camión. Estos datos los va a leer una unidad de control basada en ESP32 que a posteriori se mandarán al Cloud. Para poder mandar la información conectaremos el controlador mediante WiFi a la Raspberry Pi que también estará trabajando como router.

En la siguiente imagen se muestra como quedará la arquitectura del sistema de monitorización y control del camión.



### 2.3.2 Hardware

## Unidad de Control - Cisterna

Se usará una unidad **Norvi IIOT AE-04**, una solución basada en ESP32 que está preparada para entornos industriales. Dispone de entradas y salidas compatibles con los niveles de voltaje que se pueden encontrar en la industria y un puerto de comunicación Modbus RS485. Además viene encapsulado y con la posibilidad de montaje en carril DIN. Este modelo en concreto lleva una antena de WiFi exterior para poder tener una mejor comunicación.



AE04-V ESP32-WROOM32



## Sensor de Temperatura

El sensor de temperatura que se usará para medir la temperatura de la leche en la cisterna es una Pt100 con tubo de cuello. Se trata del modelo **HTFB3/MUV/100** de la marca **TITEC**. Es IP54 con lo que no hay ningún problema con que se encuentre a la intemperie. El rango de medición es de -50°C a 600°C y se lee mediante una señal 0-10V.



## Sensor de Nivel

El sensor usado para medir el nivel de del tanque es un sensor de ultrasonidos con un rango de medición de 350 a 6000mm. La medición se puede hacer desde una señal 0-10V. Este sensor es del fabricante LOXONE.



## Unidad de Control - Camión

Se usará una Raspberry Pi 4 de 4GB Quad core Cortex-A72 (ARM v8) 64-bit SoC @ 1.5GHz. Incorpora conectividad WiFi 2.4GHz y 5.0GHz, Bluetooth 5.0 y BLE. También tiene un bloque de 40 pines GPIO que se puede usar para conectar distintos dispositivos. Se conectará una SSD de 32GB donde se instalará el sistema operativo Raspberry Pi OS.



## Módulo 4G

Para dar conectividad a la Raspberry Pi usaremos periférico **SIM7600G-H** que se monta encima de la placa. Este módulo permite conectividad 4G/3G/2G y además posicionamiento GNSS (Sistema global de navegación por satélite).



Las características del módulo son las siguientes:

- SIM card slot, supports 1.8V/3V SIM card
- TF card slot for storing data like files, messages, etc.
- Onboard audio jack and audio decoder for making telephone call
- 2x LED indicators, easy to monitor the working status

- Baudrate: 300bps ~ 4Mbps (default: 115200bps)
- Autobauding baudrate: 9600bps ~ 115200bps

#### Características GNSS:

- 16-channel
- C/A code

#### Sensitivity

- Tracking: -159 dBm (GPS) / -158 dBm (GLONASS) / TBD (BD)
- Cold starts: -148 dBm

#### Time-To-First-Fix (open air)

- Cold starts: <35s
- Hot starts: <1s

#### Precisión

- Posicionamiento: <2.5m CEP

#### Pantalla Táctil

El otro periférico que usaremos será un monitor con pantalla táctil de 7". Este además nos servirá como estuche para la Raspberry Pi.



#### Interfaz OBD2

Para poder leer toda la información del ordenador a bordo del camión usaremos un escáner OBD2 de Kimood equipado con Bluetooth 4.0. Tiene solo 3 milisegundos de baja latencia. Se puede usar para las siguientes funcionalidades:



- Lectura de los códigos de problemas de diagnóstico, tanto genéricos como específicos del fabricante, y muestre su significado.
- Borrar los códigos de falla y apague la luz MIL ("Check Engine" light).
- Muestra los datos de sensores a tiempo real como:
  - RPM del motor.
  - Valor de carga calculado.
  - Temperatura del refrigerante.
  - Estado del sistema de combustible.
  - Velocidad del vehículo.
  - Presión del colector de admisión.
  - Temperatura del aire de admisión.
  - Tasa de flujo de aire.
  - Voltajes del sensor de oxígeno / ajustes de combustible a corto plazo asociados.
  - Muchos otros datos más.

### 2.3.3 Software

Para el desarrollo del software de la parte de los camiones usaremos 2 herramientas. La primera Arduino IDE para programar en C++ la unidad de control Norvi IOT. Para programar la Raspberry Pi usaremos Python, un lenguaje tremadamente versátil, claro, fácil de aprender y que fomenta la creación de código de calidad. Además, dispone de librerías y frameworks de código abierto para cubrir una gran variedad de necesidades.

A parte para poder mandar los datos de Norvi IOT al broker en el Cloud usaremos la Raspberry Pi como bridge para que pueda publicar directamente sin que este necesite tener conexión 4G.

### Arduino IDE C++

Lo que vamos a programar en C++ va a ser la conexión a la red wifi que generaremos con la raspberry y al broker de MQTT montado en Azure. Además se programará la lectura de las sondas de temperatura y presión y la publicaremos al broker de mosquitto montado en el Cloud.



### Python

Lo primero que vamos a programar en python va a ser la lectura mediante bluetooth de los distintos parámetros de la centralita del camión. Para ello deberemos conocer como solicitar y cómo recibiremos la información del interfaz OBD2. Los mensajes OBD2 están formados por una trama en HEX donde hay un identificador y el dato. Los datos se separan en Modo, PID y datos (A, B, C, D) tal y como se muestra a continuación.



OBD2 frame

Identifier (e.g. 7E8)	#bytes (e.g. 03)	Mode (e.g. 41)	PID (e.g. 0D)	A (e.g. 32)	B (e.g. AA)	C (e.g. AA)	D (e.g. AA)	(Unused) (e.g. AA)
CAN ID	CAN data							

Un ejemplo de request/response de un mensaje CAN para el PID 'Velocidad del Vehículo' con un valor de 50 km/h sería de la siguiente forma:

Request: 7DF 02 01 0D 55 55 55 55 55

Response: 7E8 03 41 0D 32 AA AA AA AA

**Identificador:** para mensajes OBD2, el identificador es estándar de 11 bits y se utiliza para distinguir entre "mensajes de solicitud" (ID 7DF) y "mensajes de respuesta" (ID 7E8 a 7EF). Tenga en cuenta que el 7E8 normalmente será donde responde el motor principal o la ECU.

**Longitud:** esto simplemente refleja la longitud en número de bytes de los datos restantes (03 a 06). Para el ejemplo de Velocidad del vehículo, es 02 para la solicitud (ya que solo siguen 01 y 0D), mientras que para la respuesta es 03, ya que siguen 41, 0D y 32.

**Modo:** para solicitudes, será entre 01-0A. Para las respuestas, el 0 se reemplaza por 4 (es decir, 41, 42,..., 4A). Hay 10 modos como se describe en el estándar SAE J1979 OBD2. El modo 1 muestra los datos actuales y es, por ejemplo, utilizado para observar la velocidad del vehículo en tiempo real, RPM, etc. Otros modos se utilizan para, por ejemplo, muestra o borra los códigos de diagnóstico de problemas almacenados y muestra los datos del cuadro congelado.

**PID:** Para cada modo, existe una lista de PID OBD2 estándar, p. Ej. en el modo 01, PID 0D es la velocidad del vehículo. Para la lista completa, ver la wiki [OBDII-PID's](#). Cada PID tiene una descripción y algunos tienen un mínimo / máximo específico y una fórmula de conversión.

La fórmula para la velocidad es, por ejemplo, simplemente A, lo que significa que el byte de datos A (que está en HEX) se convierte a decimal para obtener el valor convertido en km/h (es decir, 32 se convierte en 50 km/h arriba). Por ejemplo RPM (PID 0C), la fórmula es  $(256 * A + B) / 4$ .

**A, B, C, D:** Estos son los bytes de datos en HEX, que deben convertirse a forma decimal antes de que se utilicen en los cálculos de la fórmula PID. Tener en cuenta que no se utiliza el último byte de datos (después de Dh).

Teniendo en cuenta como se solicita la información y como la recibiremos se creará un programa para poder leer todos los PID.

A parte de la lectura de los parámetros OBD2 también crearemos un programa en el que leeremos la geolocalización del camión y la mostraremos en un mapa en la pantalla de 7". En la pantalla también añadiremos unos botones para que cuando el camión esté cerca de la fábrica los pueda pulsar para abrir la puerta.

Si se cree oportuno también se podría llegar a crear un panel con gráficos y tendencias en la pantalla usando Dash para visualizar datos mediante python.

## MQTT

Los topics definidos para monitorizar los camiones serán los siguientes:

`TRUCK/<idCamion>/<TipoElemento>/<idElemento>`

`<idCamion>`: identificador del camión que está realizando la ruta. En este caso se ha nombrado a los camiones en función de la ruta que realizan. Por tanto los identificadores actuales serían: "MasAlmar", "MasBadosa", "CanSiset", "MasAlmar" y "MasBadosa". En caso de ampliar o modificar los identificadores únicamente sería dar de alta.

`<TipoElemento>`: identificador del tipo de elemento a monitorizar. En este caso se han creado dos topics, uno para el estado del vehículo y otro para el estado del tanque de almacenamiento, es decir: "TruckState", "Tank". En caso de requerirse alguna otra monitorización se añadiría un nuevo Tipo.

`<idElemento>`: este parámetro no se ha contemplado.

Ejemplo de suscripción a un único camión:

`TRUCK/MasAlmar/Tank/1`

`TRUCK/MasAlmar/TruckState/1`

Ejemplo de suscripción a los parámetros de salud de todos los vehículos:

`TRUCK/+/TruckState/#`

### 2.3.4 Dashboards

Se crea un dashboard en grafana donde podemos ver la posición de los camiones en un mapa, el nivel de cada cisterna de los camiones, la temperatura a la que se encuentra la leche y un gráfico de cómo ha ido evolucionando la temperatura de la leche.



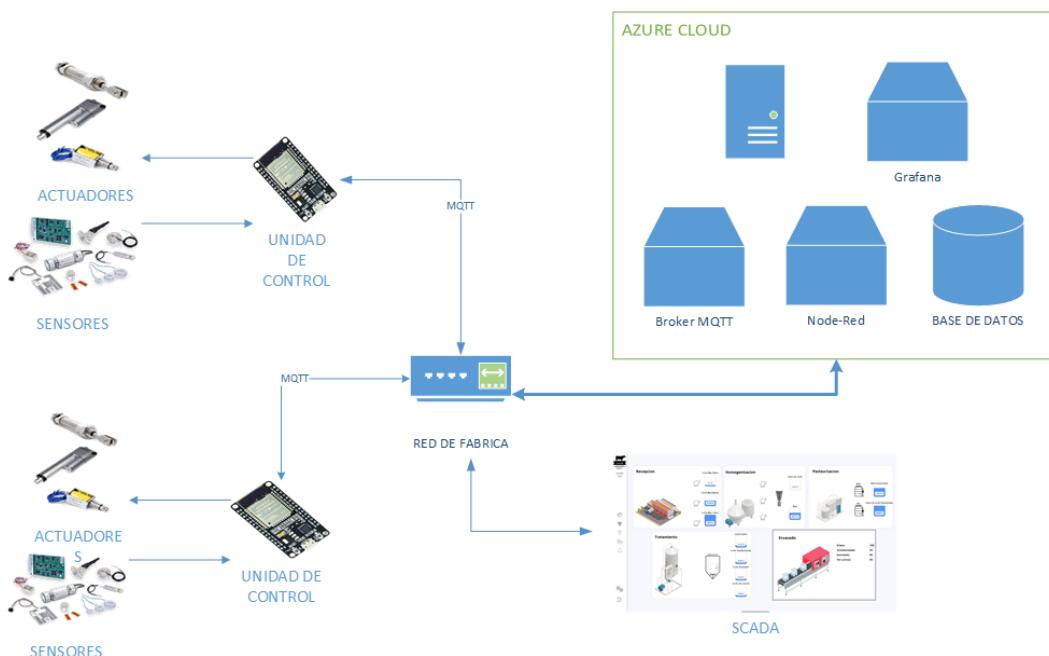
En el gráfico de serie temporal se le han marcado unos thresholds para que nos avise cuando la temperatura de la leche de un camión sale del rango establecido. En el caso que la temperatura esté fuera del rango durante X tiempo nos mandará un aviso al bot de telegram.



## 2.4 FÁBRICA

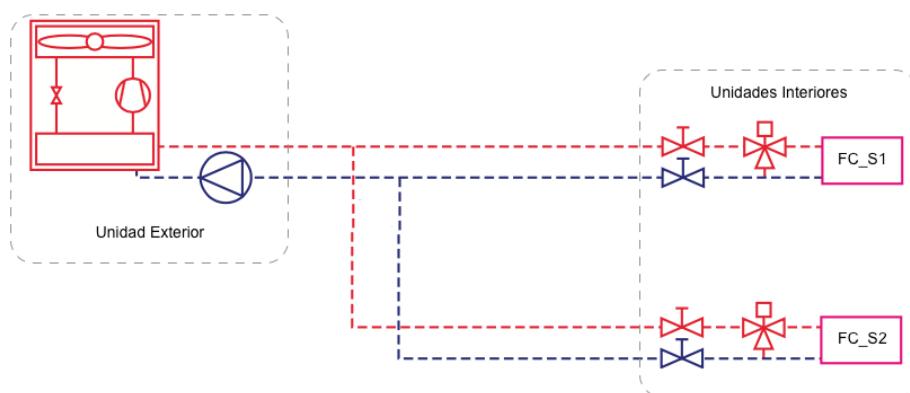
### 2.4.1 Arquitectura

La arquitectura de fábrica es la que presenta menor diversidad en cuanto a redes se refiere en lo que atañe a este proyecto. Al estar todo la cadena de producción previamente automatizada únicamente se contempla la comunicación entre los dispositivos de adquisición de datos climáticos de oficinas, los medidores de consumo de las máquinas climáticas, su comunicación con el cloud y con el SCADA vía OPC-UA. Toda esta comunicación se efectúa mediante la red ethernet ya existente en la planta productiva.



#### Máquinas de climatización

El sistema de climatización existente en la planta productiva está compuesto por una unidad exterior y 5 unidades interiores (una por cada sala). A continuación se muestra una representación gráfica de la instalación.



La unidad exterior es una enfriadora que suministra agua caliente o fría a las unidades interiores dependiendo del modo de funcionamiento en el que esté trabajando. Esta

enfriadora dispone de conexión modbus, desde donde podremos dar los siguientes comandos:

- Paro-Marcha de la unidad.
- Modo de Funcionamiento.
- Consigna del modo frío y modo calor.
- Reset de alarmas.

Además a parte de los comandos podremos leer la siguiente información de la unidad:

- Alarmas.
- Estados de los compresores, ventiladores, bomba de agua, etc.
- Contador de arranques y horas de funcionamiento de los distintos elementos.
- Lectura de las sondas de temperatura.
- Lectura de las sondas de presión del circuito refrigerante.
- Consumo eléctrico de la unidad.

En los anexos podremos encontrar la lista completa de las variables de supervisión.

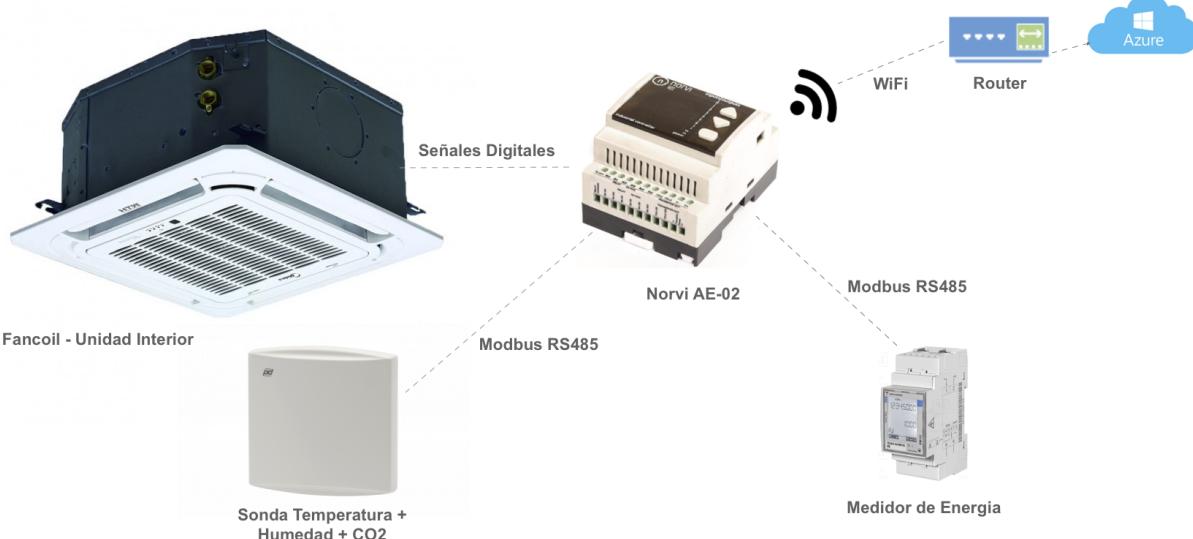
Al tener poder obtener toda la información necesaria a través de modbus la arquitectura quedará de la siguiente manera:



Como se ha comentado anteriormente cada sala dispondrá de un fancoil como unidad interior, que incorpora una placa de control. Esta placa dispone de una entrada digital para controlar el paro/marcha y otra entrada digital para la selección del modo de funcionamiento. Estas unidades las controlaremos de forma autónoma y para eso se colocará una sonda de temperatura, humedad y co2 para así saber cuando debemos arrancar la unidad. Además se supervisará el consumo de energía de este con el fin de controlar su correcto funcionamiento.

Tanto la sonda como el medidor escogidos llevan incorporado modbus rs485, facilitando el conexionado con la unidad de control que añadiremos.

La arquitectura quedará de la siguiente manera:



### Puerta de Acceso a la Planta Productiva

Para poder controlar la puerta de acceso a la planta productiva se utilizará una unidad Norvi IIOT AE01-R, que estará conectado a la red wifi de la planta. A esta unidad de control le conectaremos un contactor para controlar el paro marcha del motor y un sensor fotoeléctrico para detectar si el camión ya ha entrado o no.



## 2.4.2 Hardware

### Unidad de Control

Tanto para las unidades interiores como la unidad exterior se usará la misma unidad de control. Esta es una **Norvi IIOT**, una solución basada en ESP32 que está preparada para entornos industriales. Dispone de entradas y salidas compatibles con los niveles de voltaje que se pueden encontrar en la industria y un puerto de comunicación Modbus RS485. Además viene encapsulado y con la posibilidad de montaje en carril DIN. Para la zona de climatización se usará una **AE-02-V**



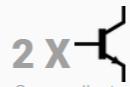
## AE02-V ESP32-WROOM32

**8 X**



24V

**6 X**   
0-10V Analog Inputs

**2 X**   
Open collector Transistor

**1 X**   
RS-485

  
0.96' OLED Display

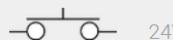
Sink/Source Digital Inputs

Para la apertura de la puerta se instalará una AE01-R.

## AE01- R ESP32-WROOM32

**R**

**8 X**



24V

**6 X**   
5A Relay outputs

**2 X**   
Open collector Transistor

**1 X**   
RS-485

  
0.96' OLED Display

Sink/Source Digital Inputs

### Sonda de Temperatura, Humedad y CO<sub>2</sub>

Para controlar las condiciones de las distintas salas se colocará una sonda de Temperatura, Humedad y CO<sub>2</sub>. Se ha creído oportuno que disponga de sensor de CO<sub>2</sub> para poder detectar cuando se necesita renovar el aire. La sonda seleccionada es el modelo **HDH-M-RH** de la marca **Produal**, una sonda de pared que dispone de salidas 0-10V para las señales pero que además incorpora comunicación por Modbus RS485.



### Medidor de Energía para las Unidades Interiores

Para poder detectar un posible mal funcionamiento de las unidades interiores de climatización, se colocarán medidores de energía. Así en el caso que haya una falla en la unidad lo podremos detectar viendo un incremento en el consumo eléctrico. El medidor seleccionado es el modelo **EM112DINAV01XS1X** del fabricante **CARLO GAVAZZI**, el cual dispone de comunicación Modbus RS485.



### Sensor Fotoeléctrico

Se añadirá un sensor fotoeléctrico que nos mandará una señal digital conforme si hay un objeto o no en la puerta. Mediante esta señal sabremos si podemos o no cerrar la puerta de acceso a la planta de producción. El sensor escogido es el modelo **XUK2ARCNL2R** del fabricante **SCHNEIDER**.

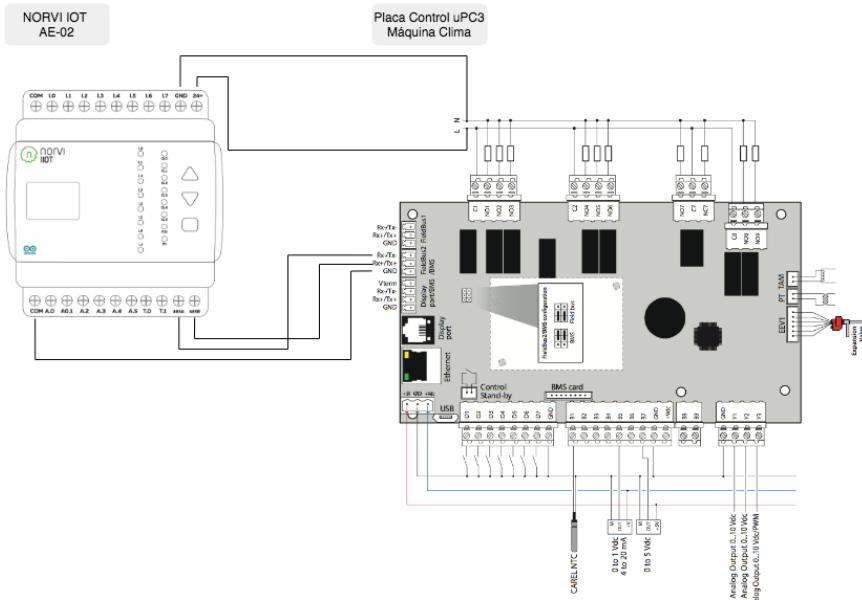


### Contacto

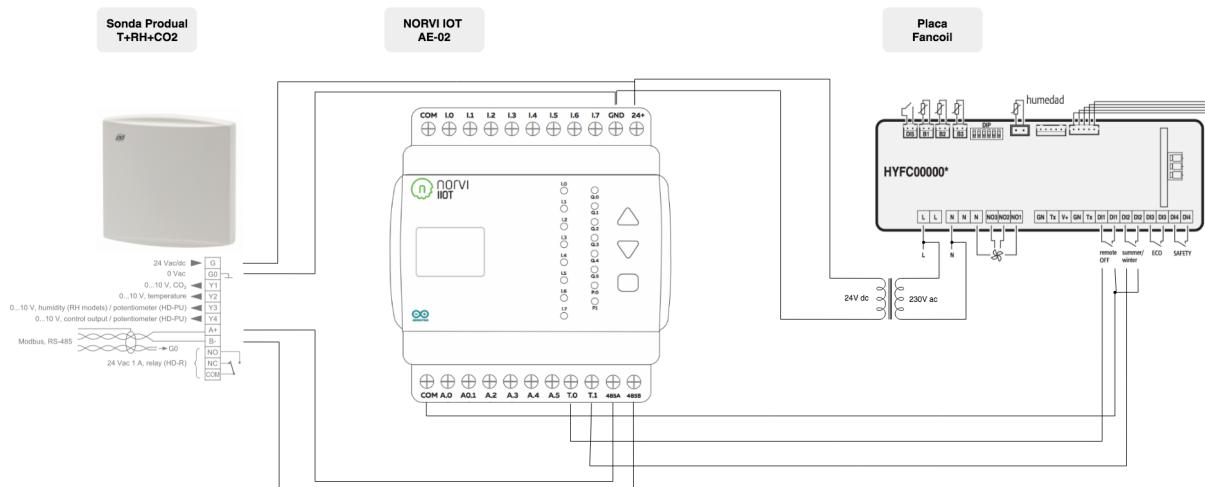
Usaremos el contacto para activar el motor de la puerta de acceso a la fábrica. Se empleará un contactor de Schneider Electric LC1D09BD, cuyas especificaciones son:

- Corriente asignada para el contactor en AC3: 9A.
- Corriente asignada para el contactor en AC1 (Ith): 25A.
- Potencia máxima (kW) 380-400Vac: 4kW.
- Potencia máxima (kW) 220-230Vac: 2,2kW.
- Contactos auxiliares: 2 NA+NC (1 abierto y 1 cerrado)

El esquema eléctrico para la unidad exterior quedará de la siguiente manera:



El esquema eléctrico para la unidad interior quedará de la siguiente forma:



### 2.4.3 Software

El software que usaremos para todas las unidades de control añadidas en la fábrica al estar basadas en ESP32 las programaremos con **C++** y la plataforma de Arduino IDE.

#### Arduino IDE C++

A todos los controladores se les creará acceso al WiFi de la fábrica y al broker MQTT instalado en la máquina virtual de Azure.

Para controlar la unidad exterior de climatización usaremos el protocolo de comunicación Modbus RS485. Para ello se hará uso de librerías Modbus ya existentes.



Para las unidades interiores también utilizaremos las librerías Modbus para comunicar con los sensores de temperatura, humedad y CO<sub>2</sub> y con el medidor de energía. Además se programará la gestión de los fancoils mediante salidas digitales con las que podremos encender y parar la unidad y seleccionar el modo de funcionamiento.

A la puerta de entrada se programará la salida digital que arrancará y parará el motor de la puerta y se leerá una entrada digital del sensor fotoeléctrico, la cual dará permiso para cerrar la puerta una vez no haya presencia de objetos en ella.

## **MQTT**

El topic definido

```
Factory/<TipoElemento>/<idElemento>/<VariableFisica>
```

<**TipoElemento**>: Elemento físico que se está midiendo. En este caso se ha considerado “Sala” para las salas, “Ext” para las medidas de en la unidad exterior de clima, “Door” para la interacción con la puerta de entrada, para referirse directamente a la ESP se usará el “ESP”.

<**idElemento**>: identificador del elemento al cual se desea suscribir.

<**VariableFisica**>: variable que se está monitorizando, por ejemplo “temperatura”, “presión”, “humedad”, “consumo”,etc..

Por ejemplo para suscribirse a la sala 1:

```
Factory/Sala/1
```

Por ejemplo para suscribirse a todas las salas:

```
Factory/Sala/#
```

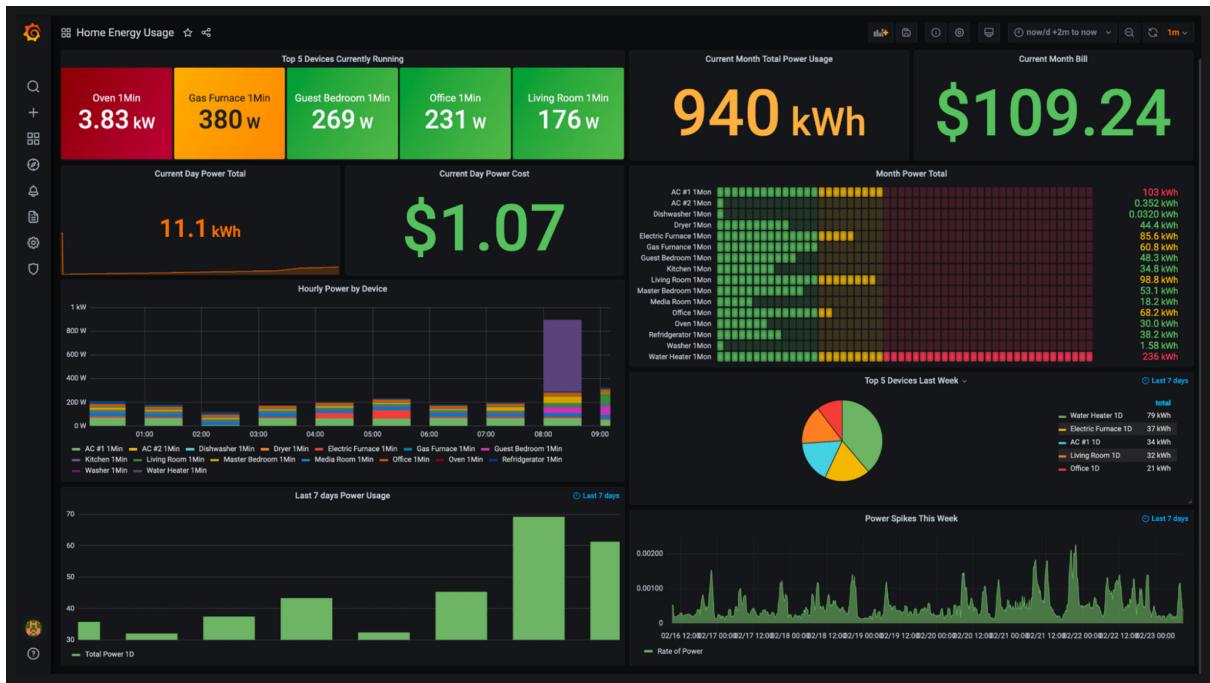
Por ejemplo para suscribirse a la puerta de entrada:

```
Factory/Door/1
```

## 2.4.4 Dashboards

Se creará un dashboard donde podamos ver el consumo de los distintos dispositivos que tenemos supervisando. En él se mostrará el consumo actual de los dispositivos, el consumo total de lo que llevamos de día y el coste de ello y después un resumen por horas, semanas y mes.

Además se añadirán thresholds para que nos avisen cuando un dispositivo está consumiendo más de la cuenta.



### 3. PRESUPUESTO

El presupuesto se divide en 4 conceptos principales: material, costes de ingeniería, costes de puesta en marcha y costes de explotación.

#### 3.1 Coste Material

El coste material se desglosa por cada subsistema, a continuación se puede observar el presupuesto para el subsistema de monitorización del camión.

DESCRIPCIÓN	MARCA/MODELO	PRECIO	UNIT	TOTAL
PANEL PC RASPBERRY	EVICIV	130,00 €	5	650,00 €
OBD CONNECTOR	KIMOOD	16,00 €	5	80,00 €
MÓDULO 4G+GPS	IBest Raspberry Pi	38,00 €	5	190,00 €
MÓDULO DE CONTROL	NORVI IIOT AE-04-V	82,69 €	5	413,45 €
SENSOR TEMPERATURA	TiTEC HTFB3/MUV/100	131,00 €	5	655,00 €
SENSOR NIVEL	LOXONE	229,59 €	5	1147,95 €
				3.136,4 €

A continuación se puede observar los costes del subsistema de granja (Recinto):

DESCRIPCIÓN	MARCA/MODELO	PRECIO	UNIT	TOTAL
MÓDULO DE CONTROL	NORVI IIOT AE02-V	75,04 €	5	375,21 €
SENSOR TEMPERATURA TANQUE	TiTEC HTFB3/MUV/100	131,00 €	5	655,00 €
SENSOR TEMPERATURA-H UMEDAD	HDH-M-RH	145,00 €	5	725,00 €
SENSOR NIVEL	LOXONE	229,59 €	5	1.147,95 €
CONTACTOR	SCHNEIDER	35,00 €	5	175,00 €
ROUTER 4G (MAS GENER)+SIM	HUAWEI	160	1	160,00 €
FUENTE DE ALIMENTACIÓN 24V	RS-PRO	25€	5	125,00 €

ARMARIO ELÉCTRICO		100€	5	500,00 €
				3.078,16 €

A continuación el subsistema de monitorización del ganado:

DESCRIPCIÓN	MARCA/MODELO	PRECIO	UNIT	TOTAL
MOTA VACA	SODAQ	181,80 €	287	52.176,60 €
GATEWAY LoRaWAN	The Things Industries	469,88 €	5	2.349,40 €
				54.526,00 €

Para terminar con el subsistema de fábrica:

DESCRIPCIÓN	MARCA/MODELO	PRECIO	UNIT	TOTAL
MÓDULO DE CONTROL	NORVI IIOT AE02-V	75,04 €	5	413,45 €
SENSOR TEMPERATURA - HUMEDAD-CO2	HDH-M-RH	145,00 €	5	725,00 €
SENSOR DE CORRIENTE	CARLO GAVAZZI	130,00 €	5	650,00 €
MÓDULO DE CONTROL	NORVI IIOT AE01-R	69,03 €	1	69,03 €
CONTACTOR	SCHNEIDER	35,00 €	2	70,00 €
DETECTOR FOTOELECTRICO (EMISOR-RECEPTOR)	TELEMECANIQUE	100€	1	100,00 €
FUENTE DE ALIMENTACIÓN 24V	RS-PRO	25€	6	150,00 €
ARMARIO ELÉCTRICO		100€	6	600,00 €
				2.777,48 €

### 3.2 Costes de Ingeniería

DESCRIPCIÓN	HORAS	COSTE
SW CAMION	60	3.000,00 €
SW GRANJA-EDIFICIO	40	2.000,00 €
SW GRANJA-ganadería	75	3.750,00 €
SW FABRICA	50	2.500,00 €
SW AZURE	100	5.000,00 €
	325	16.250,00 €

### 3.3 Costes de puesta en marcha

DESCRIPCIÓN	HORAS	COSTE
CAMION	40	2.000,00 €
GRANJA-EDIFICIO	30	1.500,00 €
GRANJA-ganadería	50	2.500,00 €
FÁBRICA	35	1.750,00 €
AZURE	25	1.250,00 €
	180	9.000,00 €

### 3.4 Costes de explotación

El cliente deberá hacerse cargo de los gastos de explotación de la plataforma. a razón de los datos consumidos vía 4G por los camiones y la granja de Mas Gener 4,5€/mes. Así como de los gastos por la explotación de los servicios de azure que ascienden a 33 €/mes.

DESCRIPCIÓN	UNIT	COSTE MENSUAL	COSTE ANUAL
SIM-CAMIÓN	6	4,5	324
SIM-GRANJA	1	6	72
Azure	1	33	396
			792

### 3.5 Viabilidad económica

El monto final de la oferta asciende a 88.768,03 € excluido IVA.

Teniendo en cuenta las pérdidas por el fallecimiento prematuro de un animal y que en la Unión Europea la recolección y trazabilidad de animales de abasto que mueren en la explotación es obligatoria desde 2002, la implementación de un sistema de monitorización del ganado se erige como un punto crítico para reducir las pérdidas ocasionadas por la pérdida de un activo, a la vez que reducir los costes incurridos por tareas obligatorias por ley pero de no valor añadido a nivel económico como es la búsqueda del animal fallecido en el campo.

Según los médicos veterinarios y zoootecnistas una morbilidad del ganado de entre el 10% y el 30% es un claro síntoma de que no se está gestionando correctamente la salud de los activos y que la mortalidad se debe mantener siempre por debajo del 3%.

Extrapolando estos datos al ganado tendríamos que con una tasa de mortalidad del 10% y una rotación del 1% trimestral pasan por la granja 294 vacas anualmente de las cuales 30 pueden fallecer, lo que implicaría una pérdida anual de unos 25k€ (suponiendo que ya se haya amortizado parte de la animal), cantidad que ya por sí sola justificaría la inversión en el ganado a 3 años que es el tiempo garantizado de las baterías.

Reducir las pérdidas por lotes de leche defectuosos al tener la cadena de frío garantizada y trazada desde el momento de la extracción es también un elemento vital, no olvidemos que la leche es la fuente de ingreso principal de la empresa y la pérdida de un lote de producción 32,43 €/t (precio medio de la leche en granjas españolas) incurre no sólo en el coste del material (324,3€ en el caso del camión más grande) sino que también puede significar perder una entrega de producto y con ello incumplir los compromisos con cliente.

Al tratarse de una solución basada en OPEN SOURCE el cliente no deberá abonar importes por licencia para ninguno de los SW especificados en el proyecto, tal es la ideología de nuestra empresa que incluso la realización de esta memoria técnica ha sido llevada a cabo con software libre.

La solución propuesta por IOTFarmers garantiza la durabilidad de los componentes al tratarse de elementos robustos de calidad industrial en los elementos de supervisión con suministro eléctrico constante, exceptuando únicamente las motas del ganado que por estar basadas en el concepto de “wearable” difieren de sus homólogos de fábrica y granja.

A su vez la solución propuesta es perfectamente escalable para más granjas, fábricas, elementos de logística, activos o monitorización de alguna otra variable que no haya sido tomado en consideración en esta propuesta, permitiendo a RASP la oportunidad de evolucionar en la captación de datos y no necesariamente sujetos al suministro de IOTFarmers, al ser una solución totalmente abierta y transparente.



## ANEXO: Lista Variables de Supervisión

Tipo Dato	Permiso	Dirección	Variable	Descripción
Discret Input	R	1	Pump_1.HW_Val	
Discret Input	R	2	Pump_2.HW_Val	
Discret Input	R	3	Comp_1.HW_Val	
Discret Input	R	4	Comp_2.HW_Val	
Discret Input	R	5	Comp_3.HW_Val	
Discret Input	R	6	Comp_4.HW_Val	
Discret Input	R	7	V4V_Circ1.HW_Val	
Discret Input	R	8	V4V_Circ2.HW_Val	
Discret Input	R	9	OnOff_VE1_VE2.HW_Val	
Discret Input	R	10	OnOff_VE3_VE4.HW_Val	
Discret Input	R	11	Resist_Deposito.HW_Val	
Discret Input	R	12	Resist_Placas_Circ1.HW_Val	
Discret Input	R	13	Resist_Placas_Circ2.HW_Val	
Discret Input	R	14	Resist_Carter_Circ1.HW_Val	
Discret Input	R	15	Resist_Carter_Circ2.HW_Val	
Discret Input	R	16	Alarm.HW_Val	
Discret Input	R	17	AI_Prb_Temp_Entrada_Agua	Unit - Prb - Sonda temp entrada agua
Discret Input	R	18	AI_Prb_Temp_Salida_Agua_Circ1	Circuito 1 - Prb - Sonda temp salida agua
Discret Input	R	19	AI_Prb_Temp_Salida_Agua_Circ2	Circuito 2 - Prb - Sonda temp salida agua
Discret Input	R	20	AI_Prb_Temp_Salida_Deposito	Unit - Prb - Sonda temp salida deposito
Discret Input	R	21	AI_Prb_Temp_Exterior	Unit - Prb - Sonda temp exterior
Discret Input	R	22	AI_Prb_Caudalimetro	Unit - Prb - Sonda caudalimetro
Discret Input	R	23	AI_Prb_Pres_Agua	Unit - Prb - Sonda presion de agua
Discret Input	R	24	AI_Prb_Temp_Descarga_Circ1	Circuito 1 - Prb - Sonda temp descarga
Discret Input	R	25	AI_Prb_Pres_Descarga_Circ1	Circuito 1 - Prb - Sonda presion descarga
Discret Input	R	26	AI_Prb_Temp_Liquido_Circ1	Circuito 1 - Prb - Sonda temp liquido
Discret Input	R	27	AI_Prb_Pres_Aspiracion_Circ1	Circuito 1 - Prb - Sonda presion aspiracion
Discret Input	R	28	AI_Prb_Temp_Aspiracion_Circ1	Circuito 1 - Prb - Sonda temp aspiracion
Discret Input	R	29	AI_Prb_Temp_Descarga_Circ2	Circuito 2 - Prb - Sonda temp descarga
Discret Input	R	30	AI_Prb_Pres_Descarga_Circ2	Circuito 2 - Prb - Sonda presion descarga

Discret Input	R	31	AI_Prb_Temp_Liquido_Circ2	Circuito 2 - Prb - Sonda temp liquido
Discret Input	R	32	AI_Prb_Pres_Aspiracion_Circ2	Circuito 2 - Prb - Sonda presion aspiracion
Discret Input	R	33	AI_Prb_Temp_Aspiracion_Circ2	Circuito 2 - Prb - Sonda temp aspiracion
Discret Input	R	34	AI_Presostato_Manual_AltaPresion_Circ1	Circuito 1 - DIN - Presostato Manual de Alta Presión
Discret Input	R	35	AI_Presostato_Manual_AltaPresion_Circ2	Circuito 1 - DIN - Presostato Manual de Alta Presión
Discret Input	R	36	AI_Pump1_Overload	Unit - DIN - TÁrmico Bomba 1
Discret Input	R	37	AI_Pump2_Overload	Unit - DIN - TÁrmico Bomba 2
Discret Input	R	38	AI_Comp1_Overload	Circuito 1 - DIN - TÁrmico Compresor 1
Discret Input	R	39	AI_Comp2_Overload	Circuito 1 - DIN - TÁrmico Compresor 2
Discret Input	R	40	AI_Comp3_Overload	Circuito 2 - DIN - TÁrmico Compresor 3
Discret Input	R	41	AI_Comp4_Overload	Circuito 2 - DIN - TÁrmico Compresor 4
Discret Input	R	42	AI_VE1_VE2_Overload	Circuito 1 - DIN - TÁrmico Ventilador Exterior 1 o 2
Discret Input	R	43	AI_VE3_VE4_Overload	Circuito 2 - DIN - TÁrmico Ventilador Exterior 3 o 4
Discret Input	R	44	AI_DscgHiP_Circ1	Circuito 1 - Envelope - High discharge pressure
Discret Input	R	45	AI_DscgHiTemp_Circ1	Circuito 1 - Envelope - High discharge temperature
Discret Input	R	46	AI_DscgLowP_Circ1	Circuito 1 - Envelope - Low discharge pressure
Discret Input	R	47	AI_HiCurr_Circ1	Circuito 1 - Envelope - High current
Discret Input	R	48	AI_HiRatioP_Circ1	Circuito 1 - Envelope - High pressure ratio
Discret Input	R	49	AI_LowDeltaP_Circ1	Circuito 1 - Envelope - Low delta pressure
Discret Input	R	50	AI_LowRatioP_Circ1	Circuito 1 - Envelope - Low pressure ratio
Discret Input	R	51	AI_SuctHiP_Circ1	Circuito 1 - Envelope - High suction pressure
Discret Input	R	52	AI_SuctLowP_Circ1	Circuito 1 - Envelope - Low suction pressure
Discret Input	R	53	AI_DscgHiP_Circ2	Circuito 2 - Envelope - High discharge pressure
Discret Input	R	54	AI_DscgHiTemp_Circ2	Circuito 2 - Envelope - High discharge temperature
Discret Input	R	55	AI_DscgLowP_Circ2	Circuito 2 - Envelope - Low discharge pressure
Discret Input	R	56	AI_HiCurr_Circ2	Circuito 2 - Envelope - High current

Discret Input	R	57	AI_HiRatioP_Circ2	Circuito 2 - Envelope - High pressure ratio
Discret Input	R	58	AI_LowDeltaP_Circ2	Circuito 2 - Envelope - Low delta pressure
Discret Input	R	59	AI_LowRatioP_Circ2	Circuito 2 - Envelope - Low pressure ratio
Discret Input	R	60	AI_SuctHiP_Circ2	Circuito 2 - Envelope - High suction pressure
Discret Input	R	61	AI_SuctLowP_Circ2	Circuito 2 - Envelope - Low suction pressure
Discret Input	R	62	AI_Batt_EVD_1	Unit - EVD - Battery discharge
Discret Input	R	63	AI EEPROM_EVD_1	Unit - EVD - EEPROM Alarm
Discret Input	R	64	AI_IncompleteClosing_EVD_1	Unit - EVD - Incomplete closing
Discret Input	R	65	AI_EmergencyClosing_EVD_1	Unit - EVD - Emergency closing
Discret Input	R	66	AI_FW_CompatibErr_EVD_1	Unit - EVD - Firmware not compatible
Discret Input	R	67	AI_ConfigErr_EVD_1	Unit - EVD - Configuration Error
Discret Input	R	68	AI_Low_SH_A_EVD_1	Circuito 1 - EEV VlvA - Low SupeHeat
Discret Input	R	69	AI_LOP_A_EVD_1	Circuito 1 - EEV VlvA - LOP
Discret Input	R	70	AI_MOP_A_EVD_1	Circuito 1 - EEV VlvA - MOP
Discret Input	R	71	AI_EEV_A_EVD_1	Circuito 1 - EEV VlvA - EEV Motor Error
Discret Input	R	72	AI_LowSuct_A_EVD_1	Circuito 1 - EEV VlvA - Low suct. temp.
Discret Input	R	73	AI_HiT_Cond_EVD_1	Unit - EVD - High condensing temp.
Discret Input	R	74	AI_Low_SH_B_EVD_1	Circuito 2- EEV VlvB - Low SupeHeat
Discret Input	R	75	AI_LOP_B_EVD_1	Circuito 2- EEV VlvB - LOP
Discret Input	R	76	AI_MOP_B_EVD_1	Circuito 2- EEV VlvB - MOP
Discret Input	R	77	AI_EEV_B_EVD_1	Circuito 2- EEV VlvB - EEV Motor Error
Discret Input	R	78	AI_LowSuct_B_EVD_1	Circuito 2- EEV VlvB - Low suct. temp.
Discret Input	R	79	AI_SelfTuning_1	Ineffective adaptive control
Discret Input	R	80	AI_EmergClos_1	Emergency closing alarm
Discret Input	R	81	AI_RangeError_1	Parameter range error (some children in the data structure are out of bounds)
Discret Input	R	82	AI_HiTempCond_2	High condensation temperature protection
Discret Input	R	83	AI_SelfTuning_2	Ineffective adaptive control
Discret Input	R	84	AI_EmergClos_2	Emergency closing alarm
Discret Input	R	85	AI_RangeError_2	Parameter range error (some children in the data structure are out of bounds)
Discret Input	R	86	AI_AFreeze_Glb	Unit - AntiFreeze Alarm (Unit On)

Discret Input	R	87	Warn_AFreeze	Unit - AntiFreeze Warn (Unit Off)
Discret Input	R	88	AI_AFreeze_Circ1	Circuito 1 - AntiFreeze Alarm (Unit On)
Discret Input	R	89	AI_AFreeze_Circ2	Circuito 2 - AntiFreeze Alarm (Unit On)
Discret Input	R	90	AI_HP_Circ1	Circuito 1 - Alta presiÃ³n de descarga
Discret Input	R	91	AI_LP_Circ1	Circuito 1 - Baja presiÃ³n de descarga
Discret Input	R	92	AI_MP_Circ1	Circuito 1 - Minima presiÃ³n de descarga
Discret Input	R	93	AI_Fuga_Circ1	Circuito 1 - Fuga de presiÃ³n
Discret Input	R	94	AI_HP_Circ2	Circuito 2 - Alta presiÃ³n de descarga
Discret Input	R	95	AI_LP_Circ2	Circuito 2 - Baja presiÃ³n de descarga
Discret Input	R	96	AI_MP_Circ2	Circuito 2 - Minima presiÃ³n de descarga
Discret Input	R	97	AI_Fuga_Circ2	Circuito 2 - Fuga de presiÃ³n
Discret Input	R	98	AI_Temp_Descarga_Circ1	Circuito 1 - Alta temperatura de descarga
Discret Input	R	99	AI_Temp_Descarga_Circ2	Circuito 2 - Alta temperatura de descarga
Discret Input	R	100	AI_FlwOutLim	Unit - Flw - Caudal fuera de rango
Discret Input	R	101	AI_WaterPresOutLim	Unit - Water Presure Out of Limit
Discret Input	R	102	AI_FlwSw_Pmp_On	Unit - DIN - FlwSw Salta la alarma si al encender bomba NO detecta flujo
Discret Input	R	103	AI_FlwSw_Pmp_Off	Unit - DIN - FlwSw Salta la alarma si al parar bomba Si detecta flujo
Discret Input	R	104	AI_EVD_Offline	Unit - EVD offline
Discret Input	R	105	AI_Offline_EM3SE_1	EM3SE offline alarm
Discret Input	R	106	AI_Fan1_Offline	Fan1 offline alarm
Discret Input	R	107	AI_Fan2_Offline	Fan2 offline alarm
Discret Input	R	108	AI_Fan3_Offline	Fan3 offline alarm
Discret Input	R	109	AI_Fan4_Offline	Fan4 offline alarm
Discret Input	R	110	AI_Hrs_Pump1	Unit - Horas funcionamiento bomba 1
Discret Input	R	111	AI_Hrs_Pump2	Unit - Horas funcionamiento bomba 2
Discret Input	R	112	AI_Hrs_Comp1	Circuito 1 - Horas funcionamiento compresor 1
Discret Input	R	113	AI_Hrs_Comp2	Circuito 1 - Horas funcionamiento compresor 2
Discret Input	R	114	AI_Hrs_Comp3	Circuito 2 - Horas funcionamiento compresor 3

Discret Input	R	115	AI_Hrs_Comp4	Circuito 2 - Horas funcionamiento compresor 4
Discret Input	R	116	AI_Hrs_VE1_VE2	Circuito 1 - Horas funcionamiento ventilador exterior 1 y 2
Discret Input	R	117	AI_Hrs_VE3_VE4	Circuito 2 - Horas funcionamiento ventilador exterior 3 y 4
Discret Input	R	118	AI_Hrs_ResistPlacas1	Circuito 1 - Horas funcionamiento resistencia placas
Discret Input	R	119	AI_Hrs_ResistPlacas2	Circuito 2 - Horas funcionamiento resistencia placas
Discret Input	R	120	AI_Hrs_ResistDeposito	Unit - Horas funcionamiento resistencia deposito
Discret Input	R	121	AI_Hrs_ResistCarter1	Circuito 1 - Horas funcionamiento resistencia carter
Discret Input	R	122	AI_Hrs_ResistCarter2	Circuito 2 - Horas funcionamiento resistencia carter
Discret Input	R	123	AI_HP_Grave_Circ1	Circuito 1 - Alta presión de descarga - Grave
Discret Input	R	124	AI_LP_Grave_Circ1	Circuito 1 - Baja presión de descarga - Grave
Discret Input	R	125	AI_MP_Grave_Circ1	Circuito 1 - Minima presión de descarga - Grave
Discret Input	R	126	AI_Fuga_Grave_Circ1	Circuito 1 - Fuga de presión - Grave
Discret Input	R	127	AI_HP_Grave_Circ2	Circuito 2 - Alta presión de descarga - Grave
Discret Input	R	128	AI_LP_Grave_Circ2	Circuito 2 - Baja presión de descarga - Grave
Discret Input	R	129	AI_MP_Grave_Circ2	Circuito 2 - Minima presión de descarga - Grave
Discret Input	R	130	AI_Fuga_Grave_Circ2	Circuito 2 - Fuga de presión - Grave
Discret Input	R	131	AI_HT_Grave_Circ1	Circuito 1 - Alta temperatura de descarga - Grave
Discret Input	R	132	AI_HT_Grave_Circ2	Circuito 2 - Alta temperatura de descarga - Grave
Discret Input	R	133	AI_FlwSw_Pmp_On_Grave	Unit - DIN - FlwSw Salta la alarma si al encender bomba NO detecta flujo - Grave
Discret Input	R	134	AI_LowRatioP_Circ1_Grave	Circuito 1 - Envelope - Low pressure ratio - Grave
Discret Input	R	135	AI_LowRatioP_Circ2_Grave	Circuito 2 - Envelope - Low pressure ratio - Grave
Discret Input	R	136	AI_FlwOutLim_Grave	Unit - Flw - Caudal fuera de rango - Grave
Discret Input	R	137	AI_AFreeze_Glb_Grave	Unit - AntiFreeze Alarm (Unit On) - Grave

Discret Input	R	138	AI_AFreeze_Circ1_Grave	Circuito 1 - AntiFreeze Alarm (Unit On) - Grave
Discret Input	R	139	AI_AFreeze_Circ2_Grave	Circuito 2 - AntiFreeze Alarm (Unit On) - Grave
Discret Input	R	140	AI_WaterPresOutLim_Grave	Unit - Water Presure Out of Limit - Grave
Discret Input	R	141	AI_S1_EVD_1	Circuito 1 - Prb - Sonda temperatura aspiracion (S1 error)
Discret Input	R	142	AI_S2_EVD_1	Circuito 1 - Prb - Sonda temperatura aspiracion (S2 error)
Discret Input	R	143	AI_S3_EVD_1	Circuito 1 - Prb - Sonda temperatura aspiracion (S3 error)
Discret Input	R	144	AI_S4_EVD_1	Circuito 2 - Prb - Sonda temperatura aspiracion (S4 error)
Discret Input	R	145	AI_ZiehlFan1_COMErr	ZiehlFan MB 1 - Comunication Error
Discret Input	R	146	AI_ZiehlFan1_MotorStart	ZiehlFan MB 1 - Motor Start Error
Discret Input	R	147	AI_ZiehlFan1_TempErr	ZiehlFan MB 1 - Temperature Error
Discret Input	R	148	AI_ZiehlFan1_SafetyShutdown	ZiehlFan MB 1 - Safety Shutdown
Discret Input	R	149	AI_ZiehlFan1_Sinefilter	ZiehlFan MB 1 - Sinefilter
Discret Input	R	150	AI_ZiehlFan1_PeakCurrent	ZiehlFan MB 1 - Peak Current
Discret Input	R	151	AI_ZiehlFan1_MotorBlock	ZiehlFan MB 1 - Motor Blocked
Discret Input	R	152	AI_ZiehlFan1_HallSensor	ZiehlFan MB 1 - Hallsensor Error
Discret Input	R	153	AI_ZiehlFan1_TB	ZiehlFan MB 1 - TB Error
Discret Input	R	154	AI_ZiehlFan1_LineFault	ZiehlFan MB 1 - Line Fault
Discret Input	R	155	AI_ZiehlFan1_UInLO	ZiehlFan MB 1 - UIN LO
Discret Input	R	156	AI_ZiehlFan1_UInHI	ZiehlFan MB 1 - UIN HI
Discret Input	R	157	AI_ZiehlFan1_UZKLO	ZiehlFan MB 1 - UZK LO
Discret Input	R	158	AI_ZiehlFan1_UZKHI	ZiehlFan MB 1 - UZK HI
Discret Input	R	159	AI_ZiehlFan1_EarthToGroundFault	ZiehlFan MB 1 - Earth to ground fault
Discret Input	R	160	AI_ZiehlFan1_IGBTFault	ZiehlFan MB 1 - IGBT Fault
Discret Input	R	161	AI_ZiehlFan1_FunctionalSafetyErr	ZiehlFan MB 1 - Functional Safety - Full Safe Mode
Discret Input	R	162	AI_ZiehlFan1_MotorcontrolSystemErr	ZiehlFan MB 1 - Motorcontrol System Warning
Discret Input	R	163	AI_ZiehlFan1_UserAppSystemErr	ZiehlFan MB 1 - User Application System Failure
Discret Input	R	164	AI_ZiehlFan1_VibrationErr	ZiehlFan MB 1 - Vibration Error
Discret Input	R	165	AI_ZiehlFan1_LifetimeErr	ZiehlFan MB 1 - Lifetime Error
Discret Input	R	166	AI_ZiehlFan1_DirectionErr	ZiehlFan MB 1 - Direction Error

Discret Input	R	167	AI_ZiehlFan1_LimitsErr	ZiehlFan MB 1 - Limitations active
Discret Input	R	168	AI_ZiehlFan2_COMErr	ZiehlFan MB 2 - Communication Error
Discret Input	R	169	AI_ZiehlFan2_MotorStart	ZiehlFan MB 2 - Motor Start Error
Discret Input	R	170	AI_ZiehlFan2_TempErr	ZiehlFan MB 2 - Temperature Error
Discret Input	R	171	AI_ZiehlFan2_SafetyShutdown	ZiehlFan MB 2 - Safety Shutdown
Discret Input	R	172	AI_ZiehlFan2_Sinefilter	ZiehlFan MB 2 - Sinefilter
Discret Input	R	173	AI_ZiehlFan2_PeakCurrent	ZiehlFan MB 2 - Peak Current
Discret Input	R	174	AI_ZiehlFan2_MotorBlock	ZiehlFan MB 2 - Motor Blocked
Discret Input	R	175	AI_ZiehlFan2_HallSensor	ZiehlFan MB 2 - Hallsensor Error
Discret Input	R	176	AI_ZiehlFan2_TB	ZiehlFan MB 2 - TB Error
Discret Input	R	177	AI_ZiehlFan2_LineFault	ZiehlFan MB 2 - Line Fault
Discret Input	R	178	AI_ZiehlFan2_UInLO	ZiehlFan MB 2 - UIN LO
Discret Input	R	179	AI_ZiehlFan2_UInHI	ZiehlFan MB 2 - UIN HI
Discret Input	R	180	AI_ZiehlFan2_UZKLO	ZiehlFan MB 2 - UZK LO
Discret Input	R	181	AI_ZiehlFan2_UZKHI	ZiehlFan MB 2 - UZK HI
Discret Input	R	182	AI_ZiehlFan2_EarthToGroundFault	ZiehlFan MB 2 - Earth to ground fault
Discret Input	R	183	AI_ZiehlFan2_IGBTFault	ZiehlFan MB 2 - IGBT Fault
Discret Input	R	184	AI_ZiehlFan2_FunctionalSafetyErr	ZiehlFan MB 2 - Functional Safety - Full Safe Mode
Discret Input	R	185	AI_ZiehlFan2_MotorcontrolSystemErr	ZiehlFan MB 2 - Motorcontrol System Warning
Discret Input	R	186	AI_ZiehlFan2_UserAppSystemErr	ZiehlFan MB 2 - User Application System Failure
Discret Input	R	187	AI_ZiehlFan2_VibrationErr	ZiehlFan MB 2 - Vibration Error
Discret Input	R	188	AI_ZiehlFan2_LifetimeErr	ZiehlFan MB 2 - Lifetime Error
Discret Input	R	189	AI_ZiehlFan2_DirectionErr	ZiehlFan MB 2 - Direction Error
Discret Input	R	190	AI_ZiehlFan2_LimitsErr	ZiehlFan MB 2 - Limitations active
Discret Input	R	191	AI_ZiehlFan3_LimitsErr	ZiehlFan MB 3 - Limitations active
Discret Input	R	192	AI_ZiehlFan3_COMErr	ZiehlFan MB 3 - Communication Error
Discret Input	R	193	AI_ZiehlFan3_MotorStart	ZiehlFan MB 3 - Motor Start Error
Discret Input	R	194	AI_ZiehlFan3_TempErr	ZiehlFan MB 3 - Temperature Error
Discret Input	R	195	AI_ZiehlFan3_SafetyShutdown	ZiehlFan MB 3 - Safety Shutdown
Discret Input	R	196	AI_ZiehlFan3_Sinefilter	ZiehlFan MB 3 - Sinefilter
Discret Input	R	197	AI_ZiehlFan3_PeakCurrent	ZiehlFan MB 3 - Peak Current
Discret Input	R	198	AI_ZiehlFan3_MotorBlock	ZiehlFan MB 3 - Motor Blocked

Discret Input	R	199	AI_ZiehlFan3_HallSensor	ZiehlFan MB 3 - Hallsensor Error
Discret Input	R	200	AI_ZiehlFan3_TB	ZiehlFan MB 3 - TB Error
Discret Input	R	201	AI_ZiehlFan3_LineFault	ZiehlFan MB 3 - Line Fault
Discret Input	R	202	AI_ZiehlFan3_UInLO	ZiehlFan MB 3 - UIN LO
Discret Input	R	203	AI_ZiehlFan3_UInHI	ZiehlFan MB 3 - UIN HI
Discret Input	R	204	AI_ZiehlFan3_UZKLO	ZiehlFan MB 3 - UZK LO
Discret Input	R	205	AI_ZiehlFan3_UZKHI	ZiehlFan MB 3 - UZK HI
Discret Input	R	206	AI_ZiehlFan3_EarthToGroundFault	ZiehlFan MB 3 - Earth to ground fault
Discret Input	R	207	AI_ZiehlFan3_IGBTFault	ZiehlFan MB 3 - IGBT Fault
Discret Input	R	208	AI_ZiehlFan3_FunctionalSafetyErr	ZiehlFan MB 3 - Functional Safety - Full Safe Mode
Discret Input	R	209	AI_ZiehlFan3_MotorcontrolSystemErr	ZiehlFan MB 3 - Motorcontrol System Warning
Discret Input	R	210	AI_ZiehlFan3_UserAppSystemErr	ZiehlFan MB 3 - User Application System Failure
Discret Input	R	211	AI_ZiehlFan3_VibrationErr	ZiehlFan MB 3 - Vibration Error
Discret Input	R	212	AI_ZiehlFan3_LifetimeErr	ZiehlFan MB 3 - Lifetime Error
Discret Input	R	213	AI_ZiehlFan3_DirectionErr	ZiehlFan MB 3 - Direction Error
Discret Input	R	214	AI_ZiehlFan4_DirectionErr	ZiehlFan MB 4 - Direction Error
Discret Input	R	215	AI_ZiehlFan4_LimitsErr	ZiehlFan MB 4 - Limitations active
Discret Input	R	216	AI_ZiehlFan4_COMErr	ZiehlFan MB 4 - Comunication Error
Discret Input	R	217	AI_ZiehlFan4_MotorStart	ZiehlFan MB 4 - Motor Start Error
Discret Input	R	218	AI_ZiehlFan4_TempErr	ZiehlFan MB 4 - Temperature Error
Discret Input	R	219	AI_ZiehlFan4_SafetyShutdown	ZiehlFan MB 4 - Safety Shutdown
Discret Input	R	220	AI_ZiehlFan4_Sinefilter	ZiehlFan MB 4 - Sinefilter
Discret Input	R	221	AI_ZiehlFan4_PeakCurrent	ZiehlFan MB 4 - Peak Current
Discret Input	R	222	AI_ZiehlFan4_MotorBlock	ZiehlFan MB 4 - Motor Blocked
Discret Input	R	223	AI_ZiehlFan4_HallSensor	ZiehlFan MB 4 - Hallsensor Error
Discret Input	R	224	AI_ZiehlFan4_TB	ZiehlFan MB 4 - TB Error
Discret Input	R	225	AI_ZiehlFan4_LineFault	ZiehlFan MB 4 - Line Fault
Discret Input	R	226	AI_ZiehlFan4_UInLO	ZiehlFan MB 4 - UIN LO
Discret Input	R	227	AI_ZiehlFan4_UInHI	ZiehlFan MB 4 - UIN HI
Discret Input	R	228	AI_ZiehlFan4_UZKLO	ZiehlFan MB 4 - UZK LO
Discret Input	R	229	AI_ZiehlFan4_UZKHI	ZiehlFan MB 4 - UZK HI
Discret Input	R	230	AI_ZiehlFan4_EarthToGroundFault	ZiehlFan MB 4 - Earth to ground fault
Discret Input	R	231	AI_ZiehlFan4_IGBTFault	ZiehlFan MB 4 - IGBT Fault
Discret Input	R	232	AI_ZiehlFan4_FunctionalSafetyErr	ZiehlFan MB 4 - Functional Safety - Full Safe Mode

Discret Input	R	233	AI_ZiehlFan4_MotorcontrolSystemErr	ZiehlFan MB 4 - Motorcontrol System Warning
Discret Input	R	234	AI_ZiehlFan4_UserAppSystemErr	ZiehlFan MB 4 - User Application System Failure
Discret Input	R	235	AI_ZiehlFan4_VibrationErr	ZiehlFan MB 4 - Vibration Error
Discret Input	R	236	AI_ZiehlFan4_LifetimeErr	ZiehlFan MB 4 - Lifetime Error
Coil	R/W	1	KeybOnOff	Unit On/Off by keyboard
Coil	R/W	2	AlrmResByBms	Alarm reset by BMS
Input Register	R	1	Snd_Reg_Final_MSK	°C
Input Register	R	2	Unit_Power_Request	%
Input Register	R	3	Temp_Entrada_Agua.Val	°C
Input Register	R	4	Temp_Salida_Aqua_C1.Val	°C
Input Register	R	5	Temp_Salida_Deposito	°C
Input Register	R	6	Temp_Exterior	°C
Input Register	R	7	Caudalimetro	m3/h
Input Register	R	8	Pres_Agua	bar
Input Register	R	9	Temp_Descarga_Circ1	°C
Input Register	R	10	Pres_Descarga_Circ1	bar
Input Register	R	11	Temp_Liquido_Circ1	°C
Input Register	R	12	Pres_Aspiracion_Circ1	bar
Input Register	R	13	Temp_Aspiracion_Circ1	°C
Input Register	R	14	Temp_Descarga_Circ2	°C
Input Register	R	15	Pres_Descarga_Circ2	bar
Input Register	R	16	Temp_Liquido_Circ2	°C
Input Register	R	17	Pres_Aspiracion_Circ2	bar
Input Register	R	18	Temp_Aspiracion_Circ2	°C
Input Register	R	19	Salto_Termico	°C
Input Register	R	20	Recalentamiento_Circ1	°C
Input Register	R	21	Recalentamiento_Circ2	°C

Input Register	R	22	Recalentamiento_Descarga_Circ1	°C
Input Register	R	23	Recalentamiento_Descarga_Circ2	°C
Input Register	R	24	Subcooling_Circ1	°C
Input Register	R	25	Subcooling_Circ2	°C
Input Register	R	26	EEV_A_PosPerc	%
Input Register	R	27	EEV_A_PosSteps	stp
Input Register	R	28	EEV_B_PosPerc	%
Input Register	R	29	EEV_B-PosSteps	stp
Input Register	R	30	PotenciaFrigorifica	kW
Input Register	R	31	EER_COP	
Input Register	R	32	Freq_EM3SE	Hz
Input Register	R	33	Phase1_EM3SE.ApparentPower	
Input Register	R	34	Phase1_EM3SE.Current	
Input Register	R	35	Phase1_EM3SE.Phase_NeutralVoltage	
Input Register	R	36	Phase1_EM3SE.Phase_NextPhaseVoltage	
Input Register	R	37	Phase1_EM3SE.Power	
Input Register	R	38	Phase1_EM3SE.PowerFactor	
Input Register	R	39	Phase1_EM3SE.ReactivePower	
Input Register	R	40	Phase2_EM3SE.ApparentPower	
Input Register	R	41	Phase2_EM3SE.Current	
Input Register	R	42	Phase2_EM3SE.Phase_NeutralVoltage	
Input Register	R	43	Phase2_EM3SE.Phase_NextPhaseVoltage	
Input Register	R	44	Phase2_EM3SE.Power	
Input Register	R	45	Phase2_EM3SE.PowerFactor	
Input Register	R	46	Phase2_EM3SE.ReactivePower	
Input Register	R	47	Phase3_EM3SE.ApparentPower	

Input Register	R	48	Phase3_EM3SE.Current		
Input Register	R	49	Phase3_EM3SE.Phase_NeutralVoltage		
Input Register	R	50	Phase3_EM3SE.Phase_NextPhaseVoltage		
Input Register	R	51	Phase3_EM3SE.Power		
Input Register	R	52	Phase3_EM3SE.PowerFactor		
Input Register	R	53	Phase3_EM3SE.ReactivePower		
Input Register	R	54	TotPhase_EM3SE.ApparentPower		
Input Register	R	55	TotPhase_EM3SE.Current		
Input Register	R	56	TotPhase_EM3SE.Phase_NeutralVoltage		
Input Register	R	57	TotPhase_EM3SE.Phase_NextPhaseVoltage		
Input Register	R	58	TotPhase_EM3SE.Power		
Input Register	R	59	TotPhase_EM3SE.PowerFactor		
Input Register	R	60	TotPhase_EM3SE.ReactivePower		
Input Register	R	61	TotEnergy_EM3SE.Energy		
Input Register	R	62	TotEnergy_EM3SE.ReactiveEnergy		
Input Register	R	63	Outdoor_Fan1_DCVoltage	V	
Input Register	R	64	Outdoor_Fan1_IBTTemp	°C	
Input Register	R	65	Outdoor_Fan1_InsideTemp	°C	
Input Register	R	66	Outdoor_Fan1_LineVoltage	V	
Input Register	R	67	Outdoor_Fan1_MCUtemp	°C	
Input Register	R	68	Outdoor_Fan1_Motorcurrent	A	
Input Register	R	69	Outdoor_Fan1_MotorcurrPhU	A	
Input Register	R	70	Outdoor_Fan1_MotorcurrPhV	A	
Input Register	R	71	Outdoor_Fan1_MotorPower	W	
Input Register	R	72	Outdoor_Fan1_MotorVoltage	V	
Input Register	R	73	Outdoor_Fan1_PowerFactor		



Input Register	R	74	Outdoor_Fan1_Speed	rpm
Input Register	R	75	Outdoor_Fan1_VibrationAxisX	mm/s
Input Register	R	76	Outdoor_Fan1_VibrationAxisY	mm/s
Input Register	R	77	Outdoor_Fan1_VibrationAxisZ	mm/s
Input Register	R	78	Outdoor_Fan2_DCVoltage	V
Input Register	R	79	Outdoor_Fan2_IBTTemp	°C
Input Register	R	80	Outdoor_Fan2_InsideTemp	°C
Input Register	R	81	Outdoor_Fan2_LineVoltage	V
Input Register	R	82	Outdoor_Fan2_MCUtemp	°C
Input Register	R	83	Outdoor_Fan2_Motorcurrent	A
Input Register	R	84	Outdoor_Fan2_MotorCurrPhU	A
Input Register	R	85	Outdoor_Fan2_MotorCurrPhV	A
Input Register	R	86	Outdoor_Fan2_MotorPower	W
Input Register	R	87	Outdoor_Fan2_MotorVoltage	V
Input Register	R	88	Outdoor_Fan2_PowerFactor	
Input Register	R	89	Outdoor_Fan2_Speed	rpm
Input Register	R	90	Outdoor_Fan2_VibrationAxisX	mm/s
Input Register	R	91	Outdoor_Fan2_VibrationAxisY	mm/s
Input Register	R	92	Outdoor_Fan2_VibrationAxisZ	mm/s
Input Register	R	93	Outdoor_Fan3_DCVoltage	V
Input Register	R	94	Outdoor_Fan3_IBTTemp	°C
Input Register	R	95	Outdoor_Fan3_InsideTemp	°C
Input Register	R	96	Outdoor_Fan3_LineVoltage	V
Input Register	R	97	Outdoor_Fan3_MCUtemp	°C
Input Register	R	98	Outdoor_Fan3_Motorcurrent	A
Input Register	R	99	Outdoor_Fan3_MotorCurrPhU	A



Input Register	R	100	Outdoor_Fan3_MotorCurrPhV	A
Input Register	R	101	Outdoor_Fan3_MotorPower	W
Input Register	R	102	Outdoor_Fan3_MotorVoltage	V
Input Register	R	103	Outdoor_Fan3_PowerFactor	
Input Register	R	104	Outdoor_Fan3_Speed	rpm
Input Register	R	105	Outdoor_Fan3_VibrationAxisX	mm/s
Input Register	R	106	Outdoor_Fan3_VibrationAxisY	mm/s
Input Register	R	107	Outdoor_Fan3_VibrationAxisZ	mm/s
Input Register	R	108	Outdoor_Fan4_DCVoltage	V
Input Register	R	109	Outdoor_Fan4_IGBTTemp	°C
Input Register	R	110	Outdoor_Fan4_InsideTemp	°C
Input Register	R	111	Outdoor_Fan4_LineVoltage	V
Input Register	R	112	Outdoor_Fan4_MCUtemp	°C
Input Register	R	113	Outdoor_Fan4_Motorcurrent	A
Input Register	R	114	Outdoor_Fan4_MotorCurrPhU	A
Input Register	R	115	Outdoor_Fan4_MotorCurrPhV	A
Input Register	R	116	Outdoor_Fan4_MotorPower	W
Input Register	R	117	Outdoor_Fan4_MotorVoltage	V
Input Register	R	118	Outdoor_Fan4_PowerFactor	
Input Register	R	119	Outdoor_Fan4_Speed	rpm
Input Register	R	120	Outdoor_Fan4_VibrationAxisX	mm/s
Input Register	R	121	Outdoor_Fan4_VibrationAxisY	mm/s
Input Register	R	122	Outdoor_Fan4_VibrationAxisZ	mm/s
Input Register	R	123		
Input Register	R	124		
Input Register	R	125	Defrost_Status_Circ1	Estado desescarche circuito 1

Input Register	R	126	Defrost_Status_Circ2	Estado desescarche circuito 2
Input Register	R	127		Ej. Version: 1.5.21 (Valor 1)
Input Register	R	128	Current_SW_Ver_X	Ej. Version: 1.5.21 (Valor 5)
Input Register	R	129	Current_SW_Ver_Y	Ej. Version: 1.5.21 (Valor 12)
Input Register	R	130	Current_SW_Ver_Z	Ej. Version: 1.5.21 (Valor 12)
Input Register	R	131	Current_OS_Ver_1_MSB	Ej. Version: 1.5.21 (Valor 1)
Input Register	R	132	Current_OS_Ver_1_LSB	Ej. Version: 1.5.21 (Valor 5)
Input Register	R	133	Current_OS_Ver_2_MSB	Ej. Version: 1.5.21 (Valor 5)
Input Register	R	134	Current_OS_Ver_2_LSB	Ej. Version: 1.5.21 (Valor 12)
Input Register	R	135	Current_OS_Ver_3_MSB	Ej. Version: 1.5.21 (Valor 12)
Input Register	R	136	Current_OS_Ver_3_LSB	Ej. Version: 1.5.21 (Valor 1)
Input Register	R	137	Current_BOOT_Ver_1_MSB	Ej. Version: 1.5.21 (Valor 5)
Input Register	R	138	Current_BOOT_Ver_1_LSB	Ej. Version: 1.5.21 (Valor 1)
Input Register	R	139	Current_BOOT_Ver_2_MSB	Ej. Version: 1.5.21 (Valor 5)
Input Register	R	140	Current_BOOT_Ver_2_LSB	Ej. Version: 1.5.21 (Valor 5)
Input Register	R	141	Current_BOOT_Ver_3_MSB	Ej. Version: 1.5.21 (Valor 12)
Input Register	R	142	Current_BOOT_Ver_3_LSB	Ej. Version: 1.5.21 (Valor 12)
Input Register	R	143	Serial_Number_1	

Input Register	R	144	Serial_Number_2
Input Register	R	145	Serial_Number_3
Input Register	R	146	Serial_Number_4
Input Register	R	147	Contador_Horas_Comp1_MSB
Input Register	R	148	Contador_Horas_Comp1_LSB
Input Register	R	149	Contador_Horas_Comp2_MSB
Input Register	R	150	Contador_Horas_Comp2_LSB
Input Register	R	151	Contador_Horas_Comp3_MSB
Input Register	R	152	Contador_Horas_Comp3_LSB
Input Register	R	153	Contador_Horas_Comp4_MSB
Input Register	R	154	Contador_Horas_Comp4_LSB
Input Register	R	155	Contador_Horas_Pump1_MSB
Input Register	R	156	Contador_Horas_Pump1_LSB
Input Register	R	157	Contador_Horas_Pump2_MSB
Input Register	R	158	Contador_Horas_Pump2_LSB
Input Register	R	159	Contador_Horas_Pump3_MSB
Input Register	R	160	Contador_Horas_Pump3_LSB
Input Register	R	161	Contador_Horas_RDeposito_MSB
Input Register	R	162	Contador_Horas_RDeposito_LSB
Input Register	R	163	Contador_Horas_RPlacas_Circ1_MSB
Input Register	R	164	Contador_Horas_RPlacas_Circ1_LSB
Input Register	R	165	Contador_Horas_RPlacas_Circ2_MSB
Input Register	R	166	Contador_Horas_RPlacas_Circ2_LSB
Input Register	R	167	Contador_Horas_Ventilador_Exterior_Circ1_MSB
Input Register	R	168	Contador_Horas_Ventilador_Exterior_Circ1_LSB
Input Register	R	169	Contador_Horas_Ventilador_Exterior_Circ2_MSB

Input Register	R	170	Contador_Horas_Ventilador_Exterior_Circ2_LSW	
Input Register	R	171	Contador_Arranques_Comp1_MSB	
Input Register	R	172	Contador_Arranques_Comp1_LSW	
Input Register	R	173	Contador_Arranques_Comp2_MSB	
Input Register	R	174	Contador_Arranques_Comp2_LSW	
Input Register	R	175	Contador_Arranques_Comp3_MSB	
Input Register	R	176	Contador_Arranques_Comp3_LSW	
Input Register	R	177	Contador_Arranques_Comp4_MSB	
Input Register	R	178	Contador_Arranques_Comp4_LSW	
Input Register	R	179	Contador_Arranques_Pump1_MSB	
Input Register	R	180	Contador_Arranques_Pump1_LSW	
Input Register	R	181	Contador_Arranques_Pump2_MSB	
Input Register	R	182	Contador_Arranques_Pump2_LSW	
Input Register	R	183	Contador_Arranques_Pump3_MSB	
Input Register	R	184	Contador_Arranques_Pump3_LSW	
Input Register	R	185	Contador_Arranques_RDeposito_MSB	
Input Register	R	186	Contador_Arranques_RDeposito_LSW	
Input Register	R	187	Contador_Arranques_RPlacas_Circ1_MSB	
Input Register	R	188	Contador_Arranques_RPlacas_Circ1_LSW	
Input Register	R	189	Contador_Arranques_RPlacas_Circ2_MSB	
Input Register	R	190	Contador_Arranques_RPlacas_Circ2_LSW	
Input Register	R	191	Contador_Arranques_Ventilador_Exterior_Circ1_MSB	
Input Register	R	192	Contador_Arranques_Ventilador_Exterior_Circ1_LSW	
Input Register	R	193	Contador_Arranques_Ventilador_Exterior_Circ2_MSB	
Input Register	R	194	Contador_Arranques_Ventilador_Exterior_Circ2_LSW	
Holding Register	R/W	1	Modo_Reg	3: Modo Frio, 4: Modo Bomba



Holding Register	R/W	2	SetP_Modbus	Consigna unidad (Modifica la consigna del modo en el que se está trabajando)	°C
Holding Register	R/W	3	Setpoint_Reg_Frio	Consigna Modo Frio	°C
Holding Register	R/W	4	Setpoint_Reg_Calor	Consigna Modo Bomba	°C

## ANEXO: Github

Se ha creado un repositorio en Github donde se ha colgado toda la información del proyecto, información sobre la planificación del mismo y además una aplicación para poder simular una pequeña parte de lo que vamos a desarrollar en este proyecto.

El repositorio lo podréis encontrar en el siguiente link:

<https://github.com/busingricardo/iotFarmers>