

## **Máster Dual en Industria 4.0**

Universidad de Córdoba  
Instituto de Estudios de Posgrado

### **Trabajo Fin de Máster**

# **Adaptación de un Supermercado a la Industria 4.0**

**Autor: Cristian Cuesta Sánchez**

**Director: José Manuel Palomares Muñoz**

Córdoba, Julio de 2020



# **Trabajo Fin de Máster**

Adaptación de un Supermercado a la Industria 4.0

**Autor :** Cristian Cuesta Sánchez

**Tutor :** José Manuel Palomares Muñoz

La defensa del presente Trabajo Fin de Máster se realizó el día                      de  
de 2020, siendo calificada por el siguiente tribunal:

**Presidente:**

**Secretario:**

**Vocal:**

**Suplente 1:**

**Suplente 2:**

y habiendo obtenido la siguiente calificación:

**Calificación:**

En Córdoba, a                      de                      de 2020



*Dedicado a  
mi familia / mi abuelo / mi abuela*



# Agradecimientos

Aquí vienen los agradecimientos... Aunque está bien acordarse de la pareja, no hay que olvidarse de dar las gracias a tu madre, que aunque a veces no lo parezca disfrutará tanto de tus logros como tú... Además, la pareja quizás no sea para siempre, pero tu madre sí.





# Índice general

<b>Resumen</b>	<b>I</b>
<b>Abstract</b>	<b>III</b>
<b>Índice de figuras</b>	<b>IV</b>
<b>Índice de figuras</b>	<b>V</b>
<b>Índice de tablas</b>	<b>VII</b>
<b>Introducción, Objetivos, Metodología y Planificación</b>	<b>1</b>
1.    Objetivos . . . . .	3
2.    Metodología de trabajo . . . . .	3
3.    Estructura de la memoria . . . . .	4
<b>1. Estado del arte</b>	<b>5</b>
1.1. iPro y su pantalla . . . . .	7
1.1.1. Programación iPro . . . . .	8
1.1.2. Configuración de la pantalla . . . . .	10
1.2. Kiconex . . . . .	11
1.3. ESP32-PoE . . . . .	14
1.4. Protocolo Modbus . . . . .	15
<b>2. Desarrollo del proyecto</b>	<b>17</b>
2.1. Programación iPro . . . . .	17
2.2. Diseño Pantalla para el control . . . . .	25

2.3. Elaboración de librerías en Kiconex . . . . .	28
2.4. Programación dispositivo wireless . . . . .	32
2.4.1. Librería para interfaz de configuración . . . . .	33
2.4.2. Librería para maestro Modbus RTU . . . . .	37
2.4.3. Librería para esclavo Modbus TCP . . . . .	38
2.4.4. Código principal . . . . .	40
2.4.5. Integración de todos los elementos del sistema . . . . .	41
<b>3. Pruebas y Resultados</b>	<b>43</b>
3.1. Pruebas control UTA . . . . .	43
3.2. Pruebas kiwi . . . . .	43
3.3. Puesta en marcha de la instalación . . . . .	45
<b>4. Conclusiones y Trabajos Futuros</b>	<b>47</b>
<b>Anexo A. Las Unidades de Tratamiento de Aire</b>	<b>51</b>
<b>Anexo B. Programa Unidad de Tratamiento de Aire</b>	<b>53</b>
<b>Anexo C. Registros Control Unidad de Tratamiento de Aire</b>	<b>55</b>
<b>Anexo D. Librería KiWiModbusBridge</b>	<b>75</b>

# Resumen

Ha de constar de tres o cuatro párrafos, donde se presente de manera clara y concisa de qué va el proyecto. Han de quedar respondidas las siguientes preguntas:

- ¿De qué va este proyecto? ¿Cuál es su objetivo principal?
- ¿Cómo se ha realizado? ¿Qué tecnologías están involucradas?
- ¿En qué contexto se ha realizado el proyecto? ¿Es un proyecto dentro de un marco general?

Lo mejor es escribir el resumen al final.



# Abstract

Here comes a translation of the “Resumen” into English. Please, double check it for correct grammar and spelling. As it is the translation of the “Resumen”, which is supposed to be written at the end, this as well should be filled out just before submitting.



# Índice de figuras

1.1. Plano de equipos del supermercado . . . . .	5
1.2. iPro Series . . . . .	7
1.3. Aspecto tabla de variables en ISaGRAF . . . . .	9
1.4. Ventanas pantalla iPro . . . . .	10
1.5. Aspecto interfaz Visoprog . . . . .	11
1.6. Red kiconex . . . . .	12
1.7. Aspecto librería en la plataforma IoT . . . . .	13
1.8. Olimex ESP32-PoE . . . . .	14
1.9. Conversor UEXT-RS485 . . . . .	14
1.10. Pines ESP32-PoE y conector UEXT . . . . .	15
2.1. Modos de funcionamiento . . . . .	19
2.2. Diseño de las ventanas de la pantalla . . . . .	28
2.3. Añadir parámetro a librería kiconex - Pestaña general . . . . .	30
2.4. Añadir parámetro a librería kiconex - Pestaña categoría . . . . .	31
2.5. Añadir parámetro a librería kiconex - Pestaña unidades . . . . .	31
2.6. Añadir parámetro a librería kiconex - Pestaña modbus . . . . .	31
2.7. Añadir parámetro a librería kiconex - adición de metadato . . . . .	32
2.8. Ventanas interfaz kiwi . . . . .	33
2.9. Flujo de programa del código principal de kiwi . . . . .	40
2.10. Esquema conexión equipos supermercado . . . . .	41
3.1. Desconexiones kiwi en red de Intarcon . . . . .	45
3.2. Desconexiones kiwi en red de router kiconex . . . . .	45

3.3. Instalación supermercado conectada y funcionando . . . . .	46
A.1. Plano de partes de una UTA . . . . .	52
B.1. Plano de partes de una UTA . . . . .	54
D.1. Plano de partes de una UTA . . . . .	76



# Índice de tablas

1.1. Especificaciones de E/S para distintos modelos de iPro. . . . .	8
2.1. Especificaciones E/S UTA. . . . .	17
2.2. Parámetros de configuración de la UTA. . . . .	19
3.1. Direcciones Modbus equipos supermercado. . . . .	46



# Introducción, Objetivos, Metodología y Planificación

El término Industria 4.0 implica la innovación a través de los nuevos conceptos tecnológicos como son el IoT (Internet of Things), el Big Data, Machine Learning, etc. Se trata de buscar lo que busca cualquier empresa: reducir costes e incrementar beneficios. Es por ello que cualquier empresa que no entienda este concepto, corre el riesgo de perder su cuota de mercado en un mundo tan competitivo.

La recolección de datos en la industria, y en concreto en el sector de la refrigeración, supone, en la mayoría de casos, comunicarse con PLCs que emplean el protocolo Modbus. Para ello, son cada vez más las empresas que crean sistemas gateway que traduzcan la información que viaja a través de Modbus, a un protocolo más rápido, innovador y propio de la Industria 4.0, en la mayoría de casos para intercambiar información con la nube. Es así como surgen productos y servicios completamente nuevos, como es el caso de kiconex [1].

Kiconex es un grupo de ingenieros que busca aplicar esos nuevos conceptos de la Industria 4.0 a los equipos de climatización y refrigeración, mediante un sistema de monitorización, supervisión y control. Un sistema de este tipo permite la adquisición de datos de los equipos comercializados, mediante cuyo análisis se puede estudiar el comportamiento de los mismos ante distintos ambientes y crear modelos predictivos, así como proporcionar un soporte técnico rápido y eficaz a los clientes. A esto hay que añadir la comodidad y tranquilidad del cliente, que desde la palma de su mano y desde cualquier parte del mundo, puede comprobar que sus instalaciones están funcionando correctamente, en tiempo real y sin necesidad de una supervisión presencial de las mismas.

Para todo ello, kiconex dispone de distintas opciones hardware [2] que se conectan a una instalación con un controlador previamente instalado e intercambian información con sus re-

gistros. La información recogida es enviada a un servidor en la nube y puede ser visualizada en una plataforma IoT [1]. Para realizar esta operación, el hardware dispone de puertos para la conexión de equipos de climatización y refrigeración mediante el protocolo de comunicaciones industrial Modbus [3]. Para ello, el software ha sido diseñado para poder transmitir todos los datos obtenidos por Modbus TCP o RTU a través de MQTT.

En estas situaciones se tiene muy en cuenta el concepto inglés retrofit, que consiste en la adaptación de la maquinaria ya existente a las nuevas tecnologías, sin modificar lo que ya hay o con unos mínimos cambios. Esto, en el entorno en que se mueve kiconex, supone acceder a máquinas en muchas ocasiones aisladas, a las que el tendido de cableado hace que el proceso sea difícil o impracticable. Es por ello, por lo que se necesita desarrollar un nuevo dispositivo que permita mejorar las máquinas a través de una comunicación Modbus inalámbrica.

Parte del contenido del proyecto que en este documento se desarrolla, es precisamente el diseño de un equipo wireless cuya aplicación concreta será la de dotar de conectividad inalámbrica a una serie de equipos frigoríficos de un supermercado. Las aplicaciones de un sistema de este tipo son múltiples, dada la facilidad con la que se podrían conectar equipos distribuidos, ampliando el alcance a esas zonas aisladas que se mencionan anteriormente. Un sistema inalámbrico supone una instalación rápida, limpia y económica, téngase en cuenta lo que supondría establecer comunicaciones por cable con, por ejemplo, unas islas frigoríficas situadas en la parte central de un supermercado. Significaría desmantelar dicho supermercado para la instalación de los canales y cables necesarios, implicando mayor mano de obra y posiblemente horas de cierre del establecimiento para poder realizar las labores necesarias.

El resto del proyecto desarrollado está compuesto de la parte de climatización del supermercado, del diseño de todas las comunicaciones y su integración con la plataforma IoT de kiconex. Aquí es donde entra en juego *kicontrol*, otro servicio que da kiconex para aquellos clientes que prefieren un control diseñado a medida para su instalación, con el añadido de que ese control va acompañado de un dispositivo kiconex que lo conecta con la nube. En el supermercado que se trata, las Unidades de Tratamiento de Aire (en adelante UTAs, Véase Anexo xxx: ¿Qué es una UTA?) son nuevas pero carecen de controlador, por lo que se aprovechará el diseño de un control nuevo y a medida para facilitar su posterior implantación en la red de kiconex.

## 1. Objetivos

El objetivo de este proyecto es la implantación de una red IoT para la climatización y refrigeración de un supermercado, usando la plataforma kiconex.

El objetivo principal puede dividirse en los siguientes subobjetivos:

1. Realizar un análisis de todos los puntos a tener en cuenta en la implantación del sistema.
2. Programación de un controlador para una o varias UTAs.
3. Elaboración de las librerías necesarias para la integración de todos los elementos en kiconex.
4. Desarrollo de un nuevo sistema de comunicación inalámbrico vía WiFi.
5. Integración de todos los elementos del sistema.

## 2. Metodología de trabajo

Para alcanzar los objetivos planteados se ha seguido la siguiente metodología de trabajo:

1. Analizar cada uno de los componentes del supermercado, los controles empleados, como se comunicarán entre ellos, y qué es necesario para diseñar todo el sistema y sus componentes.
2. Estudio del funcionamiento del control programable *iPro* [4], de Dixell [5].
3. Estudiar el funcionamiento del software necesario para la programación del *iPro* y su pantalla: *ISaGRAF* y *Visoprog*.
4. Desarrollo de las librerías necesarias para integrar cada uno de los controladores programables en la red de kiconex [6].
5. Diseño completo de un dispositivo de comunicación inalámbrica basado en ESP32, con un sistema que facilite su configuración e implantación por parte de los instaladores.
6. Pruebas de funcionamiento de los nuevos desarrollos realizados, resultados y conclusiones.

7. Integración de los elementos y puesta en marcha.
8. Documentar todo el proyecto realizado.

### 3. Estructura de la memoria

Para la documentación del proyecto, la memoria se ha organizado en los siguiente capítulos:

- **Capítulo 1. Estado del Arte:** Se describe el estado del arte de cada parte del proyecto y los elementos necesarios para llevarla a cabo.
  - **1. Kiconex:** Se explica cómo se estructura una red con kiconex.
  - **2. iPro y Pantalla:** Se presenta el software usado para la programación del control y la pantalla de visualización, además de las librerías y códigos de los que se parte.
  - **3. ESP32-PoE:** Se presenta el hardware empleado para la programación del dispositivo kiconex wireless, así como las librerías que han ayudado a su desarrollo.
- **Capítulo 2. Desarrollo del proyecto:** En este capítulo se desarrolla el proyecto de forma estructurada:
  - **1. Programación iPro.**
  - **2. Diseño Pantalla para el control.**
  - **3. Elaboración de librerías en kiconex.**
  - **4. Programación dispositivo wireless.**
- **Capítulo 3. Pruebas y resultados:** Se describen las pruebas realizadas para verificar el correcto funcionamiento del control de la UTA y del dispositivo inalámbrico. También se comentan los resultados obtenidos.
- **Capítulo 4. Puesta en marcha:** Se describe el modo de instalación, la comunicación e integración de cada elemento y el resultado final.
- **Capítulo 5. Conclusiones y Trabajos Futuros:** Se indican las conclusiones extraídas del trabajo realizado, aportando nuevas y posibles líneas de desarrollo.

# Capítulo 1

## Estado del arte

El presente proyecto parte de un pedido realizado por un cliente que pretende comunicar con kiconex los distintos equipos de un supermercado: muebles frigoríficos y máquinas de climatización. La mayoría de estos equipos dispone de un controlador propio, sin embargo, en el caso de la climatización, se ha solicitado el diseño de un control a medida para el funcionamiento de una UTA. Aunque no se dispone de un plano específico del supermercado y su equipamiento, la Figura 1.1 es un buen ejemplo. La única diferencia está en la climatización.

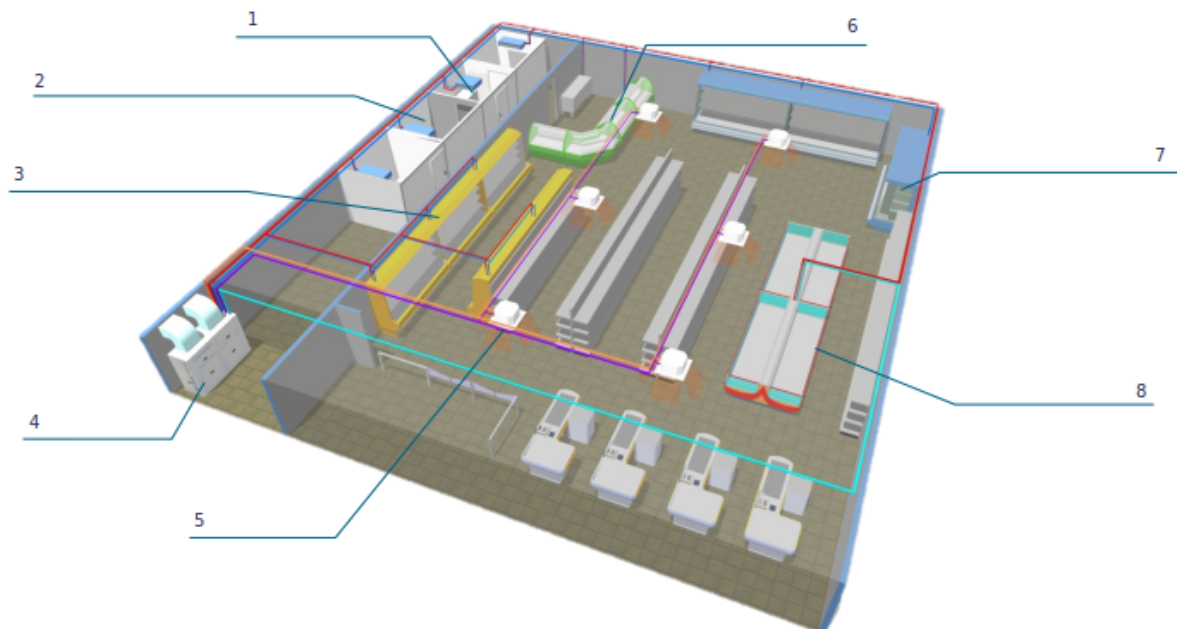


Figura 1.1: Plano de equipos del supermercado

1. Obradores.
2. Cámaras frigoríficas.
3. Murales de lácteos.
4. Bomba de calor.
5. Climatización\*.
6. Vitrinas expositoras.
7. Semimurales de carne, frutas y verduras.
8. Islas de congelados.

Kiconex emplea en sus desarrollos controladores de Dixell [5], en concreto los modelos *IPG208* e *IPG215* de *iPro* [4], con los módulos de expansión *IPX206* e *IPX215* [4]. El modelo a emplear dependerá de las especificaciones de entradas y salidas de la UTA que aparece en la Figura 1.1. A continuación, en el apartado 1 de este capítulo se describe el software necesario para la programación de un control de este tipo.

El dispositivo inalámbrico a diseñar, mencionado en el apartado de introducción anterior comunicará los muebles frigoríficos: Murales de lácteos, vitrinas expositoras, semimurales e islas de congelados. El hardware empleado, descrito en el apartado 3 de este capítulo, se basa en el chip ESP32 [7]. Para entender el papel de este dispositivo en la red, es necesario conocer el funcionamiento de una instalación desde el punto de vista de kiconex, atendiendo a cómo se integra cada elemento. El apartado 2 de este capítulo se dedica por completo a describir la estructura de un sistema basado en kiconex.

---

\*En la Figura 1.1 aparecen unidades tipo fancoil, pero lo que en realidad hay es una serie de conductos conectados a una UTA central



## 1.1. iPro y su pantalla



Figura 1.2: iPro Series

*iPro* (Figura 1.2) es la gama de controladores programables ofrecida por Dixell. La gama consta de controladores programables, ampliaciones de E/S, controladores para válvulas electrónicas e interfaces gráficas adaptadas para cubrir cualquier tipo de aplicación en el sector del aire acondicionado, el sector de la refrigeración y cualquier área relativa. Algunas de sus especificaciones son:

- Alimentación a 24Vac/dc.
- Microprocesador ARM9 de 32 bits (200MHz).
- El programa y los parámetros se almacenan en una memoria flash permanente. No se pierden datos en caso de fallo de alimentación.
- Servidor web interno.
- Hasta 80 Mb de memoria flash, dependiendo del modelo.

- Entradas y salidas completamente configurables.
- Conexiones:
  - Puerto Ethernet.
  - Puerto USB.
  - Conexión dedicada para un display LCD.
  - CANBus.
  - RS485 Master.
  - RS485 Slave.

Los modelos se diferencian en el tamaño (10 DIN o 4 DIN) y en el número de entradas y salidas (analógicas y digitales). La Tabla 1.1 recoge las especificaciones de los modelos empleados por kiconex.

	Controlador		Módulo de Expansión	
	IPG208	IPG215	IPX206	IPX215
<b>Entradas analógicas</b>	6	10	7	10
<b>Salidas analógicas</b>	4	6	3	6
<b>Entradas digitales</b>	11	20	3	20
<b>Salidas digitales (Relés)</b>	8	6	8	15

Tabla 1.1: Especificaciones de E/S para distintos modelos de iPro.

Dixell dispone de dos modelos de displays compatibles con el *iPro*: *VGIPG* y *VTIPG* [8]. Para el diseño de dichas pantallas, se emplea el software *Visoprolog* [9] de Dixell, que importa las variables creadas en la programación del controlador, para poder configurar en la pantalla la interacción con las mismas.

### 1.1.1. Programación iPro

Para la programación del *iPro* se emplea el software *ISaGRAF* [10], ya que ofrece un entorno de desarrollo estándar e internacional, soportando varios lenguajes de programación diferentes según las normas IEC61131. Se trata de un software usado en todo el mundo y que

también permite simular el sistema programado. Los estándares de programación soportados son:

- **(LD)** Escalera
- **(FBD)** Diagrama a Bloques
- **(SFC)** Tabla de Funciones Secuenciales
- **(ST)** Texto Estructurado
- **(IL)** Lista de Instrucciones
- **(FC)** Diagrama de flujo

Para comenzar a realizar un programa se parte de una plantilla ya preparada previamente por kiconex para configurar entradas, salidas, parámetros, alarmas, avisos, etc.. Es por ello que primero es necesario conocer las especificaciones del sistema a programar. En el código es necesario distinguir lo que es la configuración de entradas/salidas de lo que es el valor leído en las mismas.

A cada variable de parámetro, alarma, aviso, estado y entradas/salidas se le asigna una dirección de registro, como se puede ver en la Figura 1.3. Lo habitual es destinar distintos rangos de direcciones a cada tipo de variable.

ENTRADAS_SALIDAS							
Nombre	Tipo	Comentario	Dirección	Valor inicial	Dimensión	No volátil	0
Pb_T_deposito_1	DINT		1000			No	
Pb_T_deposito_2	DINT		1001			No	
Pb_T_deposito_3	DINT		1002			No	
Pb_T_deposito_4	DINT		1003			No	
Pb_T_ACS_impul	DINT		1004			No	
Pb_T_ACS_relor	DINT		1005			No	
Pb_T_REC_impul	DINT		1006			No	
Pb_T_REC_relor	DINT		1007			No	
AD_TEMP_CAL	DINT		1080			No	
DI_onoff	BOOL		1100			No	
DI_cal	BOOL		1101			No	
DI_cal_status	BOOL		1102			No	
DI_contador	BOOL		1103			No	
DI_B01	BOOL		1104			No	
DI_B02	BOOL		1105			No	
DI_B03	BOOL		1106			No	
DI_B04	BOOL		1107			No	
RELE_VALV_ACS	BOOL		1200			No	
RELE_VALV_EQUILIBRADO	BOOL		1201			No	
RELE_RESIS_DEPOS_ACS	BOOL		1202			No	

Figura 1.3: Aspecto tabla de variables en ISaGRAF

### 1.1.2. Configuración de la pantalla

Para la configuración de la pantalla, se emplea *Visoprog*, un software ofrecido por la misma marca Dixell, especialmente para su uso con los modelos *VGIPG* e *VTIPG*. En este caso también se ha partido de una plantilla previamente preparada por kiconex con las distintas pantallas que necesita un control como el *iPro*:

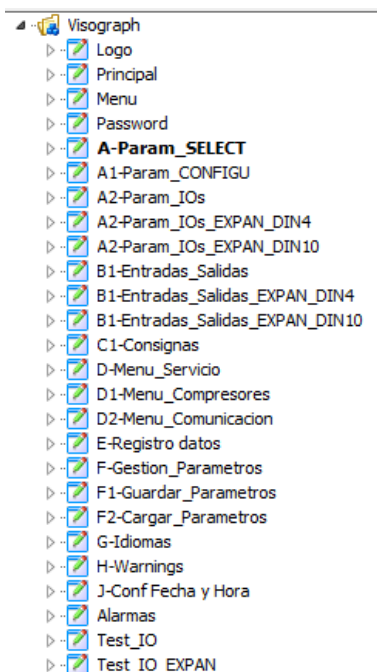


Figura 1.4: Ventanas pantalla iPro

Para usar este programa, en primer lugar se importa un fichero de registros generados por *ISaGRAF*. Así es como *Visoprog* los conoce y permite que el usuario interactúe con los mismos en el proceso de navegación por la pantalla, habiendo configurado previamente dicha interacción.

*Visoprog* tiene también una sección destinada a textos estándar, así se puede cambiar un texto en un solo sitio, y no en los cincuenta sitios en los que se está empleando. La Figura 1.5 representa el aspecto de la interfaz del programa.

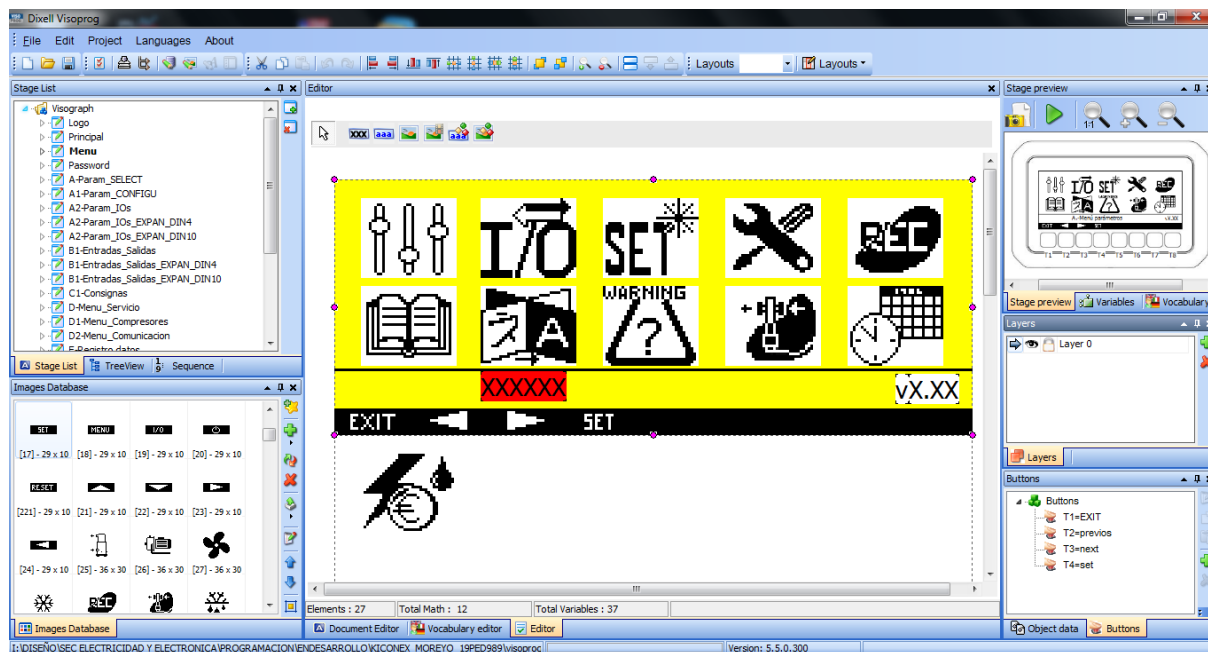


Figura 1.5: Aspecto interfaz Visoprog

## 1.2. Kiconex

Como ya se ha mencionado en la introducción, kiconex es una plataforma de supervisión y control para equipos de climatización y frío industrial. Entre sus funciones se encuentra:

- **Almacenamiento en la nube** de datos de temperatura, estado de funcionamiento, presiones, etc.
- **Gráficas** para visualizar la evolución de los datos en el tiempo, permitiendo compararlos.
- **Control remoto:** marcha/paro, cambio de consigna, etc.
- **Reglas** con programaciones horarias o acciones frente a condiciones.
- **Diagramas** para introducir un plano del edificio y sobre ese plano ver la temperatura de las salas y, por ejemplo, encender y apagar.
- **Alarmas** con posibilidad recibir alertas en caso de fallo del equipo.

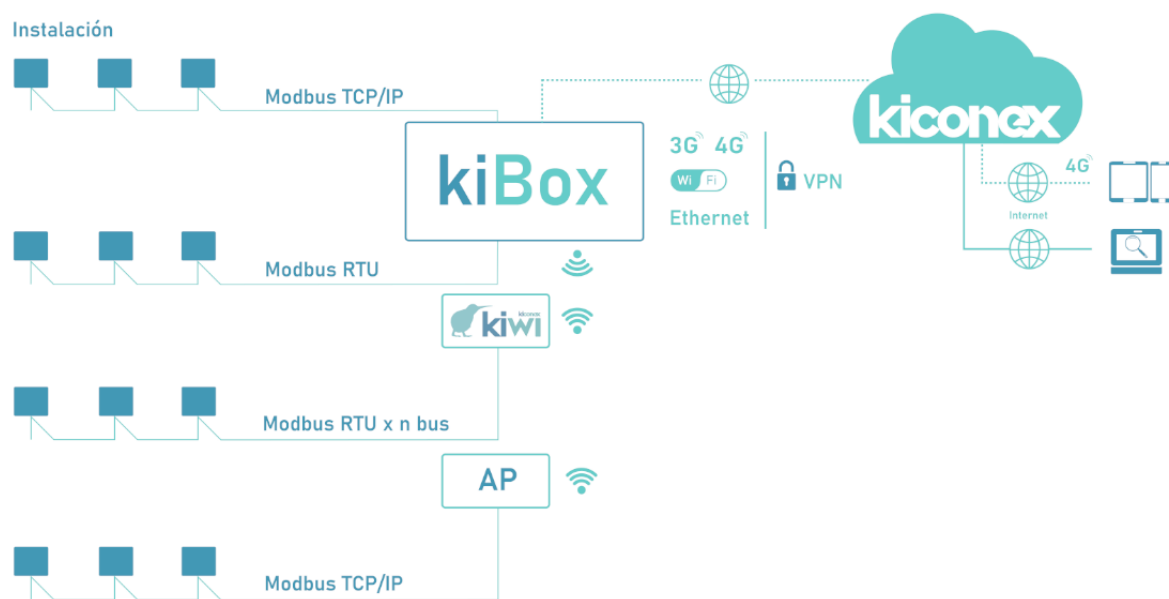


Figura 1.6: Red kiconex

Para su funcionamiento, kiconex se compone de los siguiente elementos:

1. **Servidor en la nube**, donde se almacenan los datos.
2. **Plataforma de visualización y control**.
3. **Kibox**: hardware que actúa como pasarela entre la nube y el equipo de frío o clima. Recibe mensajes de la nube a través de MQTT, los procesa y los envía en formato Modbus al equipo final. Este último, recibe el mensaje, lo procesa y envía una respuesta que sigue el camino inverso hasta que la nube recibe el dato solicitado.
4. **Equipos** de climatización o frío industrial: emplean el estándar industrial Modbus. La mayoría emplean Modbus RTU pero kiconex también funciona a través de Modbus TCP. Estos equipos reciben los mensajes del *kibox* y le envían una respuesta.
5. **Kiwi**: aún no está en venta, ya que se trata del dispositivo inalámbrico que se ha desarrollado en este proyecto, pero el objetivo es su actuación como esclavo Modbus, recibiendo mensajes del maestro *kibox* a través de TCP-IP. *Kiwi* retransmite estos mensajes actuando como maestro, a los dispositivos de frío y clima (esclavos). Es decir, *kiwi* es un maestro-esclavo: maestro de cara a los equipos y esclavo de cara al *kibox*.

La plataforma de supervisión, para poder comunicarse con los controles de los equipos de frío y clima, necesita conocer sus registros. Para ello, en la plataforma existe un apartado llamado "librerías", en el cual se pueden ver y crear los mapeados de registros de los controles que se necesiten. En la Figura 1.7 se puede ver el aspecto de una librería en la plataforma.

Nombre	Protocolo	Grupo	Intervalo	Visibilidad
<b>ENTRADAS ANALÓGICAS</b>				
<input type="checkbox"/> Transductor alta presión C1 (P) Transductor de alta presión circuito 1 (Presión)	1 (0x1)		0	<input type="checkbox"/> <input type="checkbox"/>
<input type="checkbox"/> Transductor alta presión C1 (T) Transductor de alta presión circuito 1 (Temperatura)	2 (0x2)		0	<input type="checkbox"/> <input type="checkbox"/>
<input type="checkbox"/> Lectura Escritura Val. mín.: 32768 Val. máx.: 32767 U. medida: °C Notas	Func. lectura 3 Func. escritura 0 Offset 0.1 Máscara 0 Valor 0 Longitud 16			
<input type="checkbox"/> Transductor alta presión C2 (P) Transductor de alta presión circuito 2 (Presión)	3 (0x3)		0	<input type="checkbox"/> <input type="checkbox"/>
<input type="checkbox"/> Transductor alta presión C2 (T) Transductor de alta presión circuito 2 (Temperatura)	4 (0x4)		0	<input type="checkbox"/> <input type="checkbox"/>

Figura 1.7: Aspecto librería en la plataforma IoT

Para el caso del *iPro*, en el cual el programa se diseña desde cero por kiconex, el mapeado de registros se extrae directamente del software *ISaGRAF*, y se introduce en una nueva librería en la plataforma. Cuando se trata de otros controles, es el cliente quien lo consigue a través del fabricante. kiconex, gracias a su trayectoria, ya ha recopilado una gran cantidad de librerías para multitud de controles, lo que facilita mucho el proceso.

### 1.3. ESP32-PoE

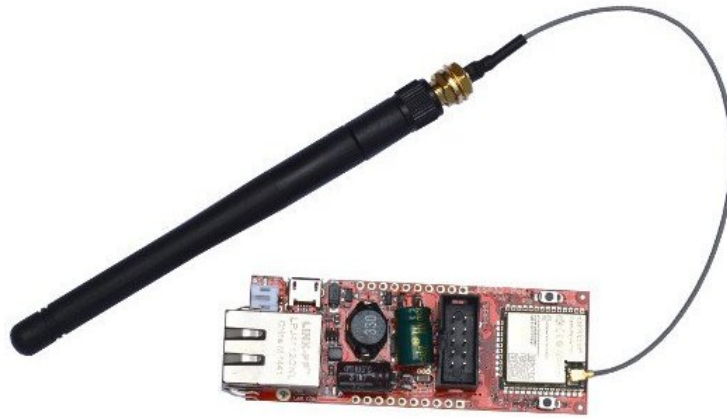


Figura 1.8: Olimex ESP32-PoE

El hardware empleado para el desarrollo del kiconex inalámbrico (*kiwi*) se basa en el chip ESP32 [7]. El modelo empleado es el Olimex ESP32-PoE (Figura 1.8). Olimex [11] es una empresa de Bulgaria líder en fabricación electrónica para el mercado integrado. Su modelo ESP32-PoE ha sido elegido por disponer de un puerto UEXT desde el que se puede obtener una interfaz RS-485 a través de un conversor [12] (Figura 1.9), para la comunicación a través de Modbus RTU. También dispone de puerto ethernet que puede usarse bien como conexión a internet vía ethernet o bien para conectar un equipo por modbus TCP. La Figura Figura 1.10 representa los pines de entrada y salida de la placa y del conector UEXT.

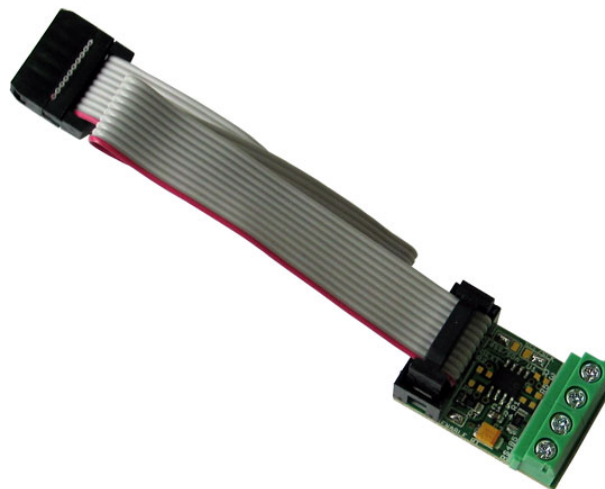
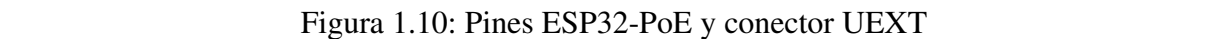


Figura 1.9: Conversor UEXT-RS485





-

ofrecer información si no se le ha solicitado antes.

- Para la comunicación, cada esclavo dispone de una dirección única de **8 bits**, por lo que un maestro puede tener hasta **254** esclavos.
- Se puede distinguir entre **Modbus TCP**, para comunicaciones vía ethernet a través de la red LAN, y **Modbus RTU**, para comunicaciones serie a través de RS232/RS485.
- Estructura del mensaje: el mensaje siempre se compone de: *dirección del esclavo*, código de *función Modbus*, *dirección de registro* y *cantidad de registros* a leer. En el caso del Modbus RTU, se añaden dos bytes que contienen un *código CRC* para la detección de errores.
- Funciones: dispone de distintas funciones en función de la naturaleza de la información compartida o de los registros con los que se desea compartir información:
  - **01 (0x01) Read Coils**: lee de 1 a 2000 bits de un dispositivo remoto.
  - **02 (0x02) Read Discrete Inputs**: lee de 1 a 2000 bits de registro en un dispositivo remoto.
  - **03 (0x03) Read Holding Registers**: lee de 1 a 125 registros de 16 bits continuos en un dispositivo remoto.
  - **04 (0x04) Read Input Registers**: lee de 1 a 125 registros de 16 bits continuos en un dispositivo remoto.
  - **05 (0x05) Write Single Coil**: escribe un solo bit de registro en el dispositivo remoto.
  - **06 (0x06) Write Single Register**: escribe un solo registro de 16bits en el dispositivo remoto.
  - **15 (0x0F) Write Multiple Coils**: escribe de 1 a 2000 bits de registro consecutivos en un dispositivo remoto.
  - **16 (0x10) Write Multiple registers**: escribe de 1 a 123 registros 16 bits consecutivos en un dispositivo remoto.

En el Anexo xxx, extraído directamente de la web de la Modbus Organization [3] se describe de forma detallada cada una de las funciones Modbus, y que información exacta contiene cada campo (byte) del mensaje.

## Capítulo 2

# Desarrollo del proyecto

En las siguientes secciones se detalla de forma estructurada el desarrollo de cada una de las partes que componen este proyecto.

### 2.1. Programación iPro

Como ya se ha explicado en los capítulos anteriores, el destino del iPro es la programación de la UTA existente en el supermercado. En el Anexo A se describe detalladamente qué es una UTA y todos los elementos de los que se puede componer. Para el caso que nos atañe, la Tabla 2.1 recoge las especificaciones concretas del control.

GRUPO	Entrada analógica	Salida analógica	Entrada digital	Salida digital
Ventiladores		Ventilador impulsión Ventilador retorno	Seguridad Ventilador impulsión Seguridad Ventilador retorno	
Filtros			Filtro entrada Filtro salida Filtro retorno	
Humectador	Humedad de retorno Humedad exterior		Indicador nivel agua	ON/OFF Humectador
Intercambiador de calor	Temp. impulsión agua Temp. retorno agua			Válvula de agua
E/S de aire	Temp. impulsión aire Temp. retorno aire Temp. extracción aire Temp. aire exterior			

Tabla 2.1: Especificaciones E/S UTA.

La lógica de funcionamiento sigue las siguientes pautas:

- **Compuertas:** No existen compuertas regulables.
- **Filtros:** La señal digital de los filtros indica si están sucios o no.
- **Ventiladores:**
  - La velocidad de los ventiladores se regula mediante una señal 0-10V, dicha velocidad debe ser la misma en cada momento.
  - La señal digital en los ventiladores indica fallo en los mismos.
  - La velocidad debe ser regulable de forma automática o de forma manual:
    - La regulación automática ajusta la velocidad del ventilador en función de la diferencia de temperatura con la consigna.
    - La regulación manual de la velocidad se establece en tres rangos parametrizables: alta, media y baja.
- **Humectador:**
  - Se establece una consigna de humedad.
  - A partir de las mediciones de humedad del aire que entra y sale se activa la humectación o no.
  - El indicador de nivel del humectador indica si éste está lleno o no.
- **Intercambiador de calor:**
  - Debe existir un aviso que indique si el fluido está a una temperatura acorde para alcanzar la consigna deseada.
  - Mientras la UTA esté en funcionamiento, la válvula que lleva el fluido hasta el intercambiador debe estar abierta.
- **E/S de aire:**
  - La temperatura de retorno es la usada como referencia para el control de temperatura del aire.

- El control de temperatura se establece según el modo de funcionamiento: verano (frío) o invierno (calor).
- Las curva de funcionamiento para cada uno de los modos es la de la Figura 2.1:

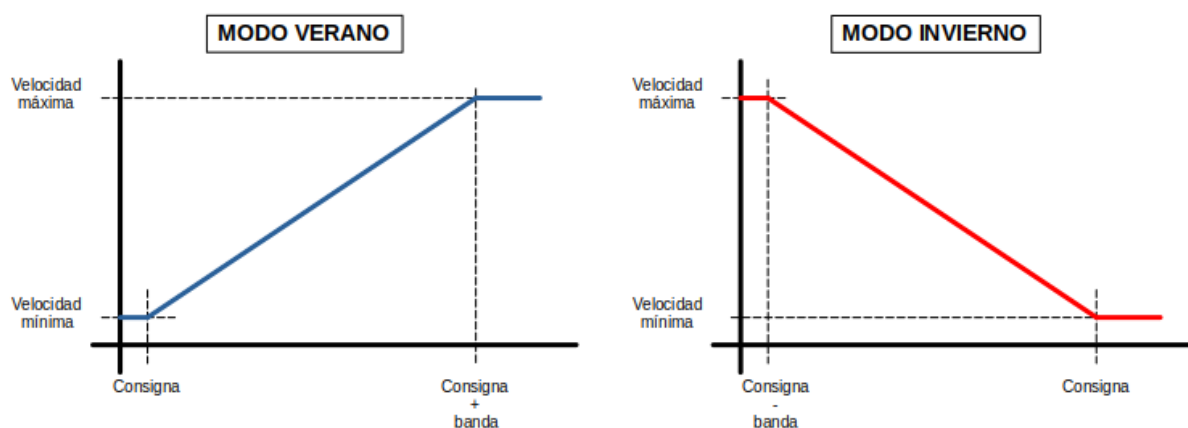


Figura 2.1: Modos de funcionamiento

El Anexo B recoge parte del programa realizado para el funcionamiento de la UTA. Los parámetros necesarios para la configuración del controlador son los de la Tabla 2.2.

Tabla 2.2: Parámetros de configuración de la UTA.

GRUPO	PARÁMETRO	DESCRIPCIÓN
Configuración Equipo	<b>CNF01</b>	Habilitar módulos de expansión (DIN4/DIN10)
	<b>HUM01</b>	Establecer consigna de humedad en %HR
	<b>HUM02</b>	Establecer banda para control de humedad, en %HR
	<b>TMP01</b>	Establecer consigna de temperatura en °C
	<b>TMP02</b>	Establecer banda de temperatura para la regulación en el modo invierno (°C)
	<b>TMP03</b>	Establecer banda de temperatura para la regulación en el modo verano (°C)

Tabla 2.2 continua en la página siguiente...

Tabla 2.2 ...continuación de la página anterior.

GRUPO	PARÁMETRO	DESCRIPCIÓN
Ventiladores	<b>FAN01</b>	Establecer valor de velocidad baja
	<b>FAN02</b>	Establecer valor de velocidad media
	<b>FAN03</b>	Establecer valor de velocidad alta
	<b>FAN04</b>	Establecer valor de velocidad mínima para la regulación automática
	<b>FAN05</b>	Establecer valor de velocidad máxima para la regulación automática
	<b>FAN06</b>	Establecer banda de velocidad para la regulación automática
	<b>FAN07</b>	Tiempo de retardo de activación de alarma en ventilador de impulsión en segundos
	<b>FAN08</b>	Tiempo de retardo de desactivación de alarma en ventilador de impulsión en segundos
	<b>FAN09</b>	Número de alarmas en el ventilador de impulsión hasta bloqueo
	<b>FAN10</b>	Tiempo de retardo de activación de alarma en ventilador de retorno en segundos
	<b>FAN11</b>	Tiempo de retardo de desactivación de alarma en ventilador de retorno en segundos
	<b>FAN12</b>	Número de alarmas en el ventilador de retorno hasta bloqueo
Filtros	<b>FIL01</b>	Retardo de activación de aviso en filtro de entrada de aire en segundos
	<b>FIL02</b>	Retardo de desactivación de alarma en filtro de entrada de aire en segundos
	<b>FIL03</b>	Retardo de activación de aviso en filtro de impulsión de aire en segundos

Tabla 2.2 continua en la página siguiente...

Tabla 2.2 ...continuación de la página anterior.

GRUPO	PARÁMETRO	DESCRIPCIÓN
	<b>FIL04</b>	Retardo de desactivación de alarma en filtro de impulsión de aire en segundos
	<b>FIL05</b>	Retardo de activación de aviso en filtro de retorno de aire en segundos
	<b>FIL06</b>	Retardo de desactivación de alarma en filtro de retorno de aire en segundos
Entradas digitales	<b>DIG01</b>	INV(FALSE)/DIR(TRUE): polaridad inversa o directa
	<b>DIG02</b>	0=No utilizado: entrada sin función asociada
	<b>DIG03</b>	1=ON-OFF: Interruptor de ON/OFF remoto
	<b>DIG04</b>	2=DI. Ventilador Impulsión: seguridad ventilador impulsión
	<b>DIG05</b>	3=DI. Ventilador retorno: seguridad ventilador retorno
	<b>DIG06</b>	4=DI. Nivel Humectador: seguridad nivel humectador
	<b>DIG07</b>	5=DI. Filtro entrada aire: seguridad filtro entrada aire
	<b>DIG08</b>	6=DI. Filtro impulsión aire: seguridad filtro impulsión aire
	<b>DIG09</b>	7=DI. Filtro retorno aire: seguridad filtro retorno aire
	<b>DIG10</b>	
	<b>DIG11</b>	
	<b>DIG12</b>	
	<b>DIG13</b>	
	<b>DIG14</b>	
	<b>DIG15</b>	
	<b>DIG16</b>	
	<b>DIG17</b>	
	<b>DIG18</b>	
	<b>DIG19</b>	
	<b>DIG20</b>	
	<b>DIG21</b>	
	<b>DIG22</b>	

Tabla 2.2 continua en la página siguiente...

Tabla 2.2 ...continuación de la página anterior.

GRUPO	PARÁMETRO	DESCRIPCIÓN
	DIG23	
	DIG24	
	DIG25	
	DIG26	
	DIG27	
	DIG28	
	DIG29	
	DIG30	
	DIG31	
	DIG32	
	DIG33	
	DIG34	
	DIG35	
	DIG36	
	DIG37	
	DIG38	
	DIG39	
	DIG40	
	DIG41	
	DIG42	
	DIG43	
Sondas analógicas	PBS01	0=no usado: sonda sin función
	PBS02	1=Ta impulsión aire
	PBS03	2=Ta retorno aire
	PBS04	3=Ta impulsión agua
	PBS05	4=Ta retorno agua
	PBS06	5=Ta extracción aire
	PBS07	6=Ta aire extraído

Tabla 2.2 continua en la página siguiente...



Tabla 2.2 ...continuación de la página anterior.

GRUPO	PARÁMETRO	DESCRIPCIÓN
	PBS08	7=Ta extracción aire
	PBS09	8=Humedad de retorno
	PBS10	9=Humedad exterior
	PBS11	
	PBS12	
	PBS13	
	PBS14	
	PBS15	
	PBS16	
	PBS17	
	PBS18	
	PBS19	
	PBS20	
	PBS21	
	PBS22	
	PBS23	
	PBS24	
	PBS25	
	PBS26	
	PBS27	
Rele salida	RLO01	0=No usado:relé sin función asociada
	RLO02	1=Válvula de 3 vías: activación válvula de agua
	RLO03	2=Humectador: activación humectador
	RLO04	3=Alarma: activación alarma
	RLO05	
	RLO06	
	RLO07	
	RLO08	

Tabla 2.2 continua en la página siguiente...

**Tabla 2.2 ...continuación de la página anterior.**

<b>GRUPO</b>	<b>PARÁMETRO</b>	<b>DESCRIPCIÓN</b>
	RLO09	
	RLO10	
	RLO11	
	RLO12	
	RLO13	
	RLO14	
	RLO15	
	RLO16	
	RLO17	
	RLO18	
	RLO19	
	RLO20	
	RLO21	
	RLO22	
	RLO23	
	RLO24	
	RLO25	
	RLO26	
	RLO27	
	RLO28	
	RLO29	
	RLO30	
	RLO31	
	RLO32	
	RLO33	
	RLO34	
	RLO35	

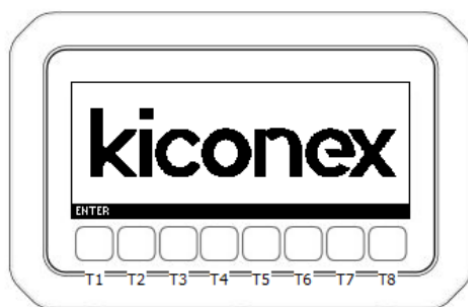
**Tabla 2.2 continua en la página siguiente...**

Tabla 2.2 ...continuación de la página anterior.

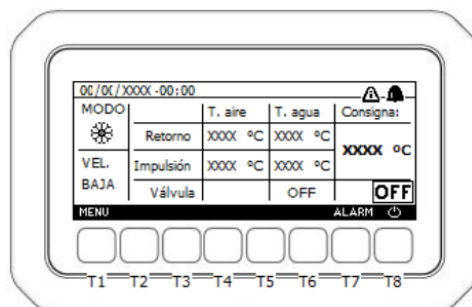
GRUPO	PARÁMETRO	DESCRIPCIÓN
	<b>RLO36</b>	
Salidas analógicas	<b>ANA01</b>	<b>0=No utilizado:</b> salida sin función asociada
	<b>ANA02</b>	<b>1=Vel. Ventilador imp.:</b> salida analógica para regular la
	<b>ANA03</b>	velocidad del ventilador de impulsión de aire
	<b>ANA04</b>	<b>2=Vel. Ventilador ret.:</b> salida analógica para regular la
	<b>ANA05</b>	velocidad del ventilador de retorno de aire
	<b>ANA06</b>	
	<b>ANA07</b>	
	<b>ANA08</b>	
	<b>ANA09</b>	
	<b>ANA10</b>	
	<b>ANA11</b>	
	<b>ANA12</b>	
	<b>ANA13</b>	
	<b>ANA14</b>	
	<b>ANA15</b>	

## 2.2. Diseño Pantalla para el control

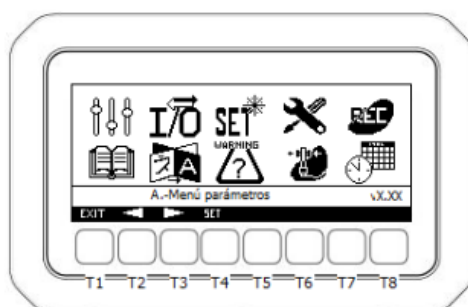
Desde el programa Visoprog, donde se diseña la pantalla, se importan las variables del programa creado con el software ISaGRAF. Para ello, Visoprog tiene dos opciones de importación: directamente desde un proyecto ISaGRAF o desde una hoja excel o fichero csv. El diseño realizado se compone de las ventanas de la Figura 2.2, su funcionamiento se describe en el manual del anexo xxx:



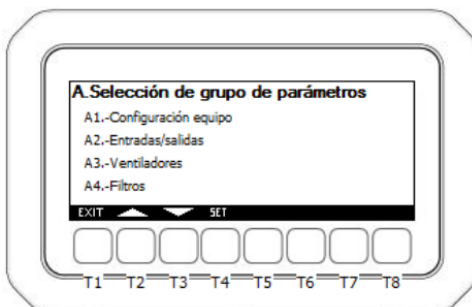
(a) Logo de inicio



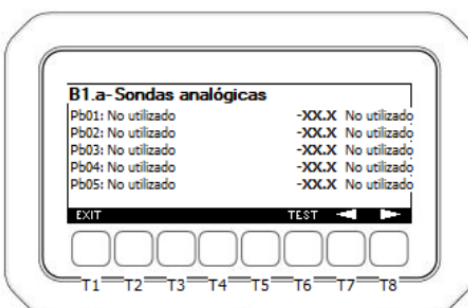
(b) Pantalla principal



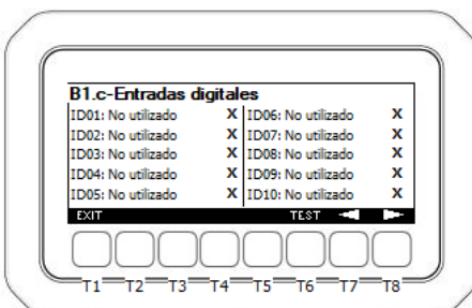
(c) Menú



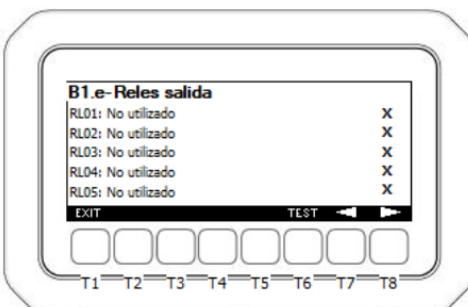
(d) Menú de parámetros



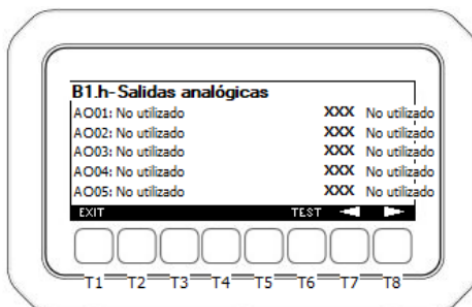
(e) Configuración de sondas analógicas



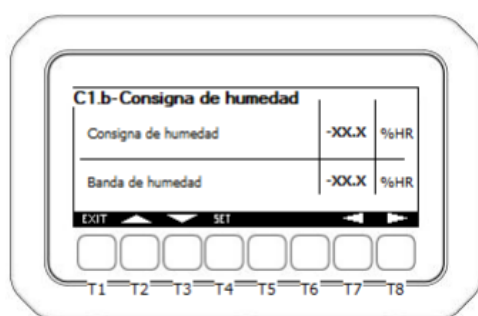
(f) Configuración de entradas digitales



(g) Configuración de relés de salida



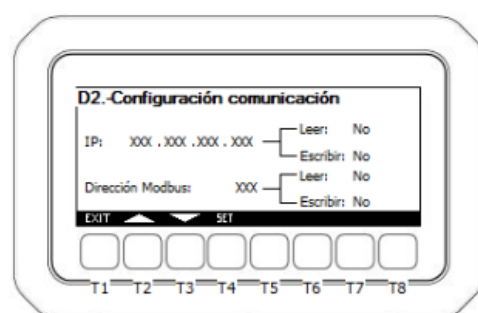
(h) Configuración de salidas analógicas



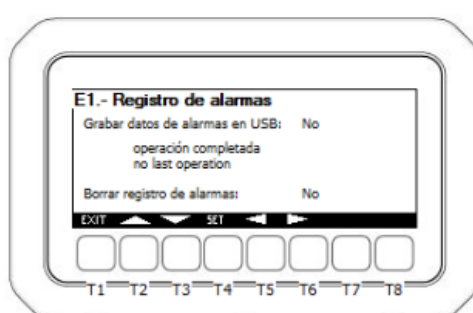
(i) Consigna de humedad



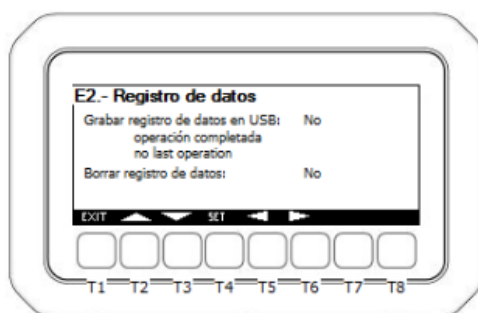
(j) Menú de servicio



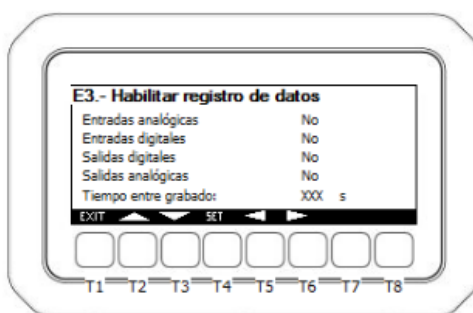
(k) Configuración de comunicación



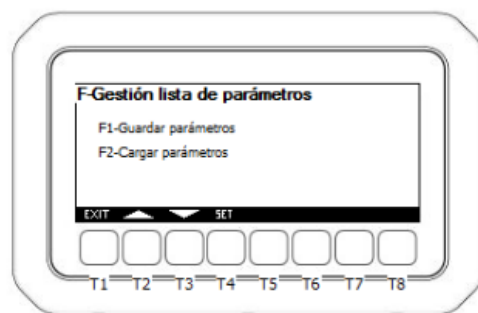
(l) Registro de alarmas



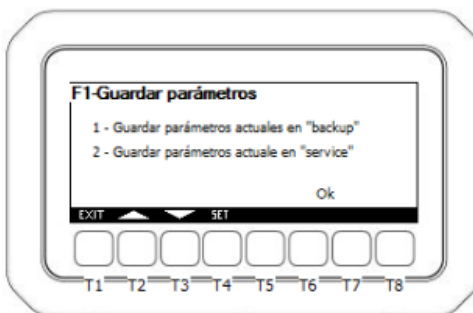
(m) Registro de datos



(n) Habilitar el registro de datos



(ñ) Menú de carga/descarga de parámetros

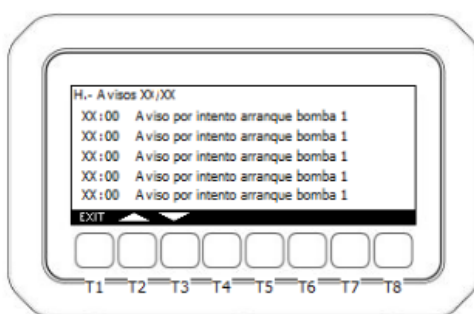


(o) Guardar parámetros



(p) Cargar parámetros

(q) Cambio de idioma



(r) Visualización de avisos

Figura 2.2: Diseño de las ventanas de la pantalla

## 2.3. Elaboración de librerías en Kiconex

En el capítulo anterior se explicaba que son necesarias librerías que recopilen los registros de cada control para la integración del mismo en la plataforma IoT de kiconex. Los controles que componen la instalación del supermercado son los siguientes:

- **Dixell XW60VS:** para cámaras frigoríficas y obradores.
- **Dixell XR-75:** para murales, semimurales, vitrinas e islas de congelados.
- **Danfoss AK-PC-551:** para la central de refrigeración.
- **Nuevo control basado en iPro IPG208:** el control programado en el apartado anterior, destinado a la UTA.

Para todos los controles existen ya librerías preparadas en kiconex, excepto para el iPro de la UTA, para el que hay que crearla a partir del programa realizado.

ISaGRAF permite exportar un recopilatorio de variables en formato PDF. La mayoría de lo que aparece en dicho PDF se puede omitir en la librería por tratarse de parámetros por defecto del control, por lo que se han recopilado en el Tabla C los registros que sí hay que tener en cuenta. Para añadir dichos registros a una librería, es necesario completar los siguientes campos de la pestaña librerías en la plataforma IoT:

- Nombre<sup>\*</sup>
- Descripción<sup>\*</sup>
- Categoría<sup>\*</sup>
- Grupo
- Unidad de medida
- Rango
- **Registro**<sup>\*</sup>
- Función de lectura
- Función de escritura
- Offset
- Longitud (bits)<sup>\*</sup>
- Máscara
- Valor
- Metadatos

La *categoría* clasifica al registro en función de su naturaleza (E/S, Parámetro, Alarma, etc.), y el *grupo* lo clasifica en función de la aplicación (ejemplo: ventiladores, compresores, etc.). Los campos *offset*, *máscara* y *valor* permiten hacer un tratamiento del valor que se lee o que se va a escribir, lo que es útil en situaciones en las que, por ejemplo, solo se quiere escribir

---

<sup>\*</sup>Campo obligatorio.

un bit concreto de un campo de 8 bits, o cuando se leen datos de temperatura con campos decimales (el valor leído necesita la aplicación de un offset que indique cuántos decimales tiene). El campo *icono* permite tener un icono asociado al valor de cara a los diagramas de la plataforma. Los *metadatos* asocian valores a un campo de texto o a un icono. Por ejemplo, el estado de un ventilador se asocia a dos iconos: uno de un ventilador apagado y otro de un ventilador encendido, lo que hace que en la ventana principal se indique el estado de dicho ventilador a través de un icono en lugar de un valor, lo que lo hace más visual. Esto se aplica también a los diagramas.

El aspecto de la interfaz para introducir los campos anteriores son los de las Figuras 2.3 a 2.7.

Datos del parámetro

GENERAL CATEGORÍA UNIDADES MODBUS METADATOS NOTAS

\* Nombre

\* Descripción

Tags

\* Intervalo

Visibilidad

Icono general


Figura 2.3: Añadir parámetro a librería kiconex - Pestaña general





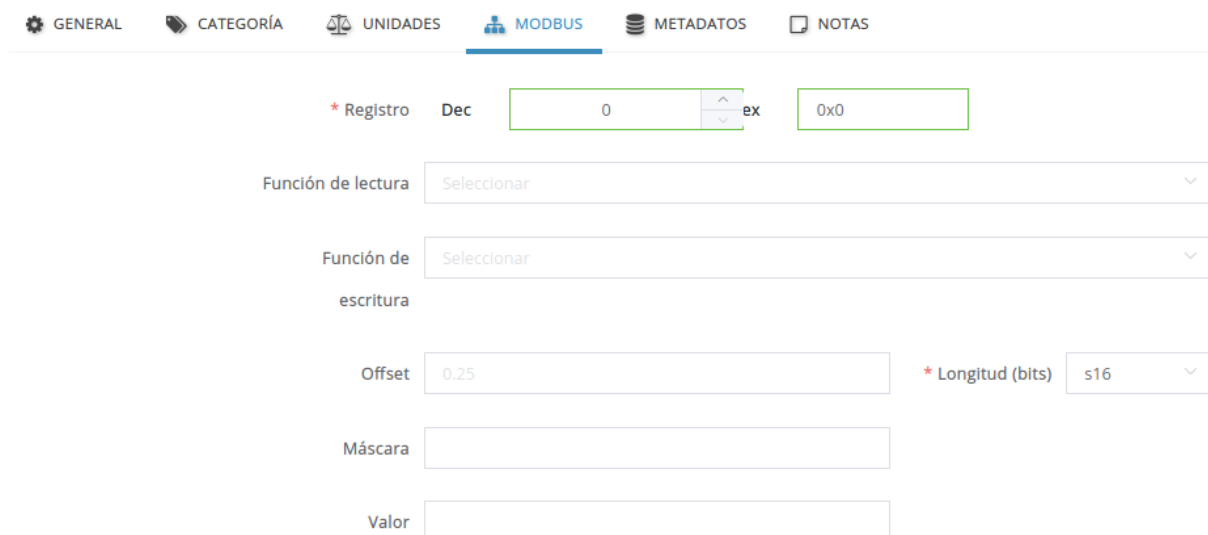
The screenshot shows the 'CATEGORÍA' tab in the Kiconex software. The tab is highlighted with a blue underline. Below the tab, there are two dropdown menus. The first is labeled '\* Categoría' and has 'Categoría' selected. The second is labeled 'Grupo' and has 'Grupo' selected.

Figura 2.4: Añadir parámetro a librería kiconex - Pestaña categoría



The screenshot shows the 'UNIDADES' tab in the Kiconex software. The tab is highlighted with a blue underline. Below the tab, there are two input fields. The first is labeled 'U. de medida' and has 'Unidad' selected. The second is labeled 'Rango' and has '0' and '1' selected, with a range of '(De -65535 a 65535)'.

Figura 2.5: Añadir parámetro a librería kiconex - Pestaña unidades



The screenshot shows the 'MODBUS' tab in the Kiconex software. The tab is highlighted with a blue underline. Below the tab, there are several input fields. The first is labeled '\* Registro' and has 'Dec' selected, with '0' and 'ex' entered. The second is labeled 'Función de lectura' and has 'Seleccionar' selected. The third is labeled 'Función de escritura' and has 'Seleccionar' selected. The fourth is labeled 'Offset' and has '0.25' entered. The fifth is labeled '\* Longitud (bits)' and has 's16' selected. The sixth is labeled 'Máscara' and is empty. The seventh is labeled 'Valor' and is empty.

Figura 2.6: Añadir parámetro a librería kiconex - Pestaña modbus

Metadato

Operador Igual a

\* Valor

Etiqueta

Icono Seleccionar

Nivel de alarma Ninguno

Cancelar Guardar

Figura 2.7: Añadir parámetro a librería kiconex - adición de metadato

## 2.4. Programación dispositivo wireless

Llegados a este punto, hay que buscar la forma de conectar de forma inalámbrica los muebles de refrigeración: murales de lácteos, vitrinas expositoras, semimurales e islas de congelados. Para ello se ha diseñado un nuevo dispositivo que se conecta de forma cableada al control del equipo y de forma inalámbrica a través de una red WiFi. Ya se describió en el capítulo anterior el hardware a emplear, el ESP32-PoE de Olimex, y se explicó el funcionamiento de una red kiconex. Es por ello que el software del ESP32 debe tener una serie de requisitos:

- Comunicación: debe disponer de una interfaz de configuración de red, tanto de la red Modbus como de la red TCP/IP.
- Esclavo Modbus TCP: debe ser capaz de recibir las tramas Modbus TCP del maestro kibox, entenderlas y procesarlas para extraer cada campo.
- Maestro Modbus RTU: a través de la información extraída de la trama TCP recibida, enviar una nueva trama a través de Modbus RTU.

- Respuesta: es necesario repetir el proceso a la inversa: recibir la respuesta a la trama enviada por Modbus RTU, extraer la información y reenviarla como respuesta a la trama Modbus TCP recibida previamente desde el Maestro kibox.

### 2.4.1. Librería para interfaz de configuración

Para la interfaz se parte de la librería WiFiManager de Khoi Hoang [14] (consultar su enlace en la bibliografía para más información). Dicha librería ha sido configurada por completo para adaptarla a las necesidades de diseño del kiwi:

1. Interfaz en cualquier modo: accesible tanto en modo AP (para realizar la configuración), como en modo cliente WiFi (para ver y cambiar la configuración realizada).
2. Medio de conexión: permite seleccionar Ethernet o WiFi.
3. Diseño: logo de kiwi y colores de kiconex.
4. Nuevas ventanas con más opciones, como se muestra en la Figura 2.8:

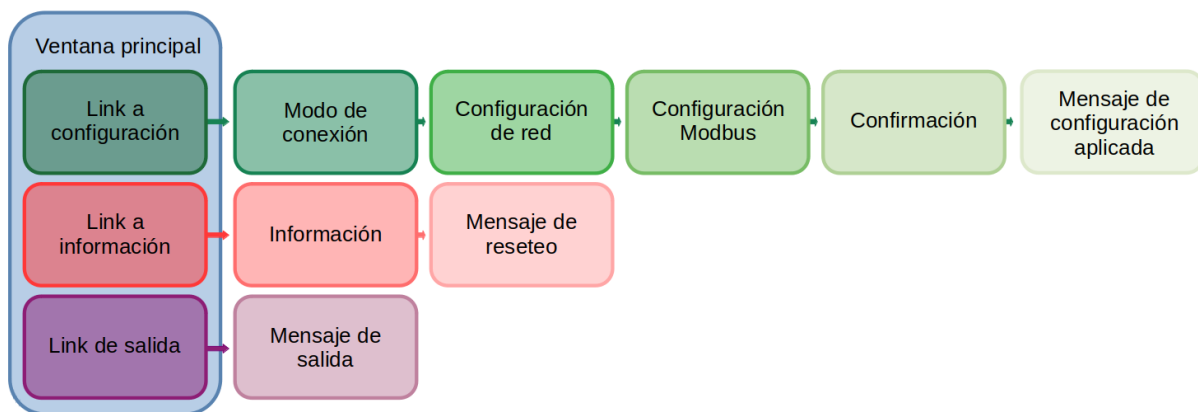


Figura 2.8: Ventanas interfaz kiwi

La nueva librería resultado de aplicar los anteriores cambios se ha bautizado con el nombre de KiWiManager. Su uso dentro del código principal es el siguiente:

- Configuración inicial:

```
#include "../KiWiManager/src/KiWiManager.h"
#include "../KiWiManager/src/KiWiManager.cpp"

#define USE_AVAILABLE_PAGES    false
#define SHOW_AP_DEBUG          true
#define CONFIGURATION_TIMEOUT  90      //time in seconds

// For some unknown reason webserver can only be started once per boot
// up
// so webserver can not be used again in the sketch.
#define WIFI_CONNECT_TIMEOUT    60000L
#define WHILE_LOOP_DELAY        5000L
#define WHILE_LOOP_STEPS        (WIFI_CONNECT_TIMEOUT / ( 3 *
    WHILE_LOOP_DELAY ))

// SSID and PW for Config Portal
const char* HOSTNAME = "kiwigw";
String ssid = "KiWi_" + String(ESP_getChipId(), HEX);
const char* password = "*****";
byte mac[6];

// SSID and PW for your Router
String Router_SSID;
String Router_Pass;

KiWiManager KiWiPortal(HOSTNAME);

void setup()
{
    KiWiPortal.begin();
}
```

### ■ Acceso al modo AP:

```
KiWiPortal.setDebugOutput(SHOW_AP_DEBUG);
KiWiPortal.showAvailablePages(USE_AVAILABLE_PAGES);
KiWiPortal.setMinimumSignalQuality(15);
```

```
Router_SSID = KiWiPortal.getSSID();
Router_Pass = KiWiPortal.WiFi_Pass();

if (Router_SSID != "")
{
    KiWiPortal.setConfigPortalTimeout(CONFIGURATION_TIMEOUT); //If no
        access point name has been previously entered disable timeout.
#ifdef MBMainDebug
    Serial.println("Stored: SSID = " + Router_SSID + ", Pass = " +
        Router_Pass);
    Serial.print("Timeout ");
    Serial.print(CONFIGURATION_TIMEOUT);
    Serial.println("s");
#endif
}
else{
#ifdef MBMainDebug
    Serial.println("No timeout");
#endif
}

if (!KiWiPortal.startConfigPortal((const char *) ssid.c_str()/*,
    password*/)) {
#ifdef MBMainDebug
    Serial.println("Not connected to WiFi but continuing anyway.");
#endif
} else{
#ifdef MBMainDebug
    Serial.println("WiFi connected in WiFiManager.");
#endif
}
```

#### ■ Conexión en modo cliente WiFi:

```
KiWiPortal.setDebugOutput (SHOW_AP_DEBUG);
KiWiPortal.showAvailablePages (USE_AVAILABLE_PAGES);
KiWiPortal.setMinimumSignalQuality (40);
```

```
Router_SSID = KiWiPortal.getSSID();
Router_Pass = KiWiPortal.getPASS();
IPAddress ip = KiWiPortal.getIP();
IPAddress gw = KiWiPortal.getGW();
IPAddress sn = KiWiPortal.getSN();
IPAddress dns1 = KiWiPortal.getDNS1();
IPAddress dns2 = KiWiPortal.getDNS2();

WiFi.mode(WIFI_STA);
WiFi.config(ip,gw,sn,dns1,dns2);
WiFi.begin(Router_SSID.c_str(),Router_Pass.c_str());

while ( (WiFi.status() != WL_CONNECTED) && (millis() - startedAt <
WIFI_CONNECT_TIMEOUT ) )
{
    #ifdef MBMainDebug
        Serial.print(".");
    #endif
    int i = 0;
    while ( (!WiFi.status() || WiFi.status() >= WL_DISCONNECTED) && i
        ++ < WHILE_LOOP_STEPS)
    {
        delay(WHILE_LOOP_DELAY);
    }
}
```

### ■ Conexión en modo cliente Ethernet:

```
KiWiPortal.setDebugOutput(SHOW_AP_DEBUG);
KiWiPortal.showAvailablePages(USE_AVAILABLE_PAGES);
KiWiPortal.setMinimumSignalQuality(15);

configState = KiWiPortal.startEthernetClientConfigPortal((const char
    *) ssid.c_str());
```

## 2.4.2. Librería para maestro Modbus RTU

Para esto se ha empleado la librería Modbus RTU del usuario smarmengol (Samuel Marco) en github. Para usarla dentro del código y poder enviar mensajes Modbus y recibir las respuestas, se emplea el siguiente código:

### ■ Inicialización:

```
#include "../KiWiModbusRTU/KiWiModbusRtu.h"
#include "../KiWiModbusRTU/KiWiModbusRtu.cpp"

Modbus master(u8id, u8serno, u8txenpin);

master->begin(u32speed);           // baud-rate
master->setTimeout(timeOut);       // Tiempo entre reintentos
```

### ■ Uso:

```
uint8_t rtuCommunication = 1;
uint8_t ret = 0;

modbus_t telegrama[2];
uint8_t w_id = READ_index;

if (rw1 == 1) w_id = WRITE_index;

this->_u8state = 0;

/**
 * Struct with Modbus RTU parameters
 * telegrama[0].u8id      slave ID
 * telegrama[0].u8fct     Modbus Function Code
 * telegrama[0].u16RegAdd Start address
 * telegrama[0].u16CoilsNo Number of registers or coils to read/write
 * telegrama[0].au16reg   Pointer to memory array
 */
telegrama[rw1].u8id = au16data[ID_index];           // slave ID
telegrama[rw1].u8fct = au16data[FN_index];          // function code
telegrama[rw1].u16RegAdd = au16data[START_index];  // star address
```

```
telegrama[rwl].ul6CoilsNo = aul6data[LONG_index]; // number of
    registers/coils
telegrama[rwl].aul6reg = aul6data + w_id;          // pointer to
    memory array

while (rtuCommunication)
{
    switch (this->_u8state)
    {
        case 0:
            this->_u8state++;
            break;
        case 1:
            master->query(telegrama[rwl]); // send query (only once)
            this->_u8state++;
            break;
        case 2:
            ret = master->poll(); // check message reception
            /**
                When generating the request to the slave, the master is in
                COM_IDLE mode, and after that its status would be
                COM_WAITING
                COM_IDLE=0 ; COM_WAITING=1
            */
            if (master->getState() == COM_IDLE)
            {
                rtuCommunication = 0;
            }
            break;
    }
}
```

### 2.4.3. Librería para esclavo Modbus TCP

En este caso se ha diseñado por completo una librería cuya función es esperar la recepción de mensajes a través de TCP/IP a la IP del dispositivo, procesar esos mensajes y enviarlos vía Modbus RTU empleando la librería anterior. Es decir se trata de una librería puente, por lo que



se ha bautizado con el nombre de KiWiModbusBridge. Su uso dentro del código principal es el siguiente:

```
// Modbus bridge library
#include "../KiWiModbusBridge/KiWiModbusBridge.h"
#include "../KiWiModbusBridge/KiWiModbusBridge.cpp"

/**
 * ModBus RTU Configuration
 * u8id : id = 0 for master, = 1..247 for slaves
 * u8serno : serial port (0 for Serial)
 * u8txenpin : 0 for RS-232 and USB-FTDI
 *
 * any value > 1 para RS-485
 */
uint8_t u8id = 0;
uint8_t u8serno = 0;
uint8_t u8txenpin = 5;

KiWiModbusBridge slave(u8id, u8txenpin, u8txenpin);

long rtu_br;
unsigned int tcp_port;

void setup()
{
    rtu_br=9600;
    tcp_port=502;
    slave.begin(tcp_port, rtu_br);
}

void loop()
{
    slave.run();
}
```

En el Anexo D se describe la lógica de funcionamiento de esta librería.

### 2.4.4. Código principal

El código principal del software de kiwi se encarga de integrar cada una de las librerías anteriores y de gestionar el flujo de programa. Su lógica de funcionamiento se recoge en el siguiente diagrama:

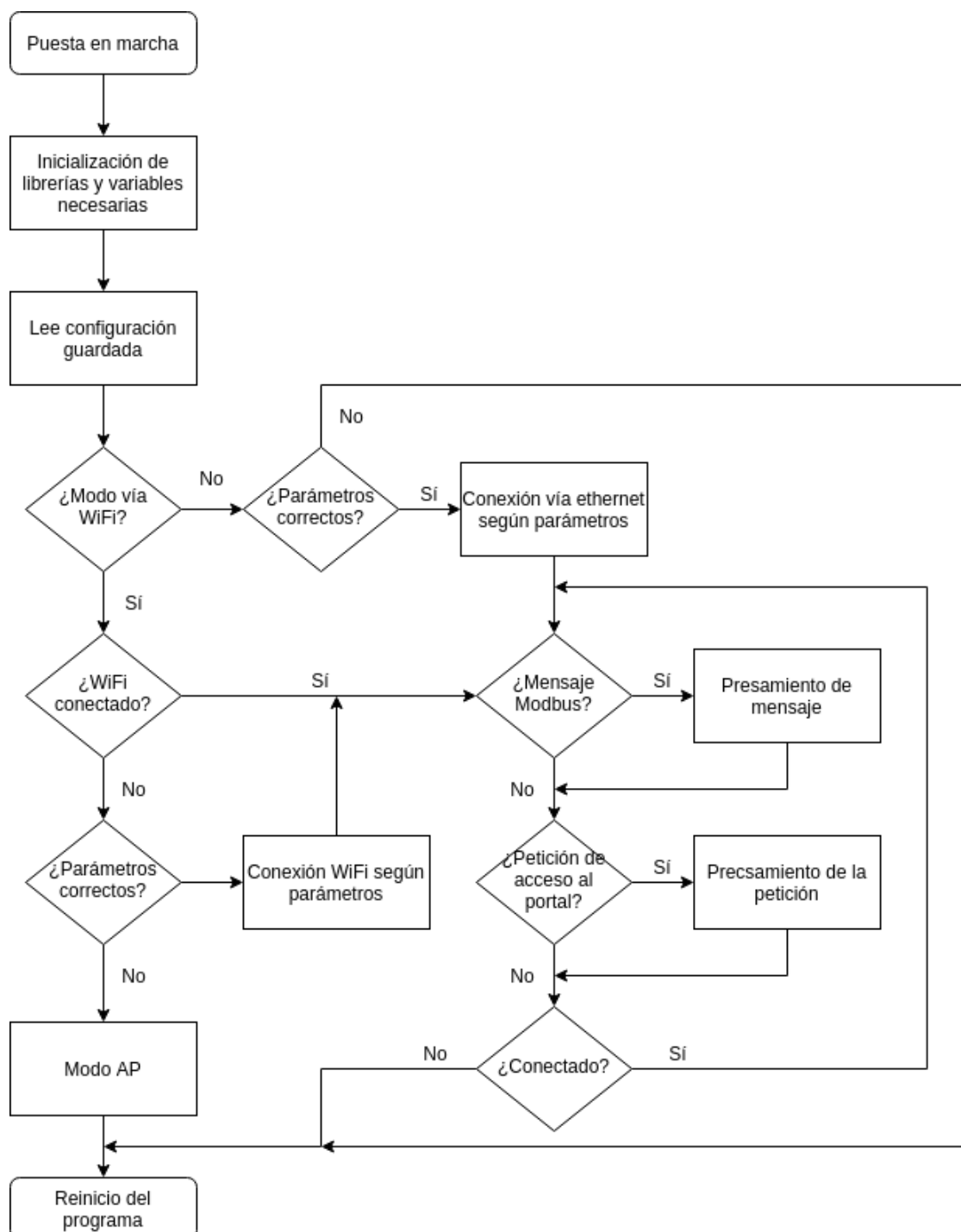


Figura 2.9: Flujo de programa del código principal de kiwi

### 2.4.5. Integración de todos los elementos del sistema

Ya esta todo listo para conectarlo a la red kiconex. Como ya se ha mencionado, los equipos del supermercado se comunican con la plataforma a través del kibox y del kiwi. Por ello, lo primero es crear la instalación del supermecado en la plataforma IoT, utilizando un id único de cada kibox, y después se añaden a dicha instalación cada uno de los equipos:

- Las centrales, las cámaras de refrigeración y los equipos de clima, se conectan de forma cableada al kibox a través de Modbus RTU, por lo que en la plataforma se añaden indicando su librería de registros y el puerto y la dirección Modbus.
- En el caso de los muebles de refrigeración, se conectan de forma cableada con el kiwi a través de Modbus RTU. Dado que la comunicación se realiza a través del mismo kiwi, vía TCP, cada mueble se añade a la instalación indicando la librería de registros del control, la IP del kiwi, y la dirección Modbus del control.

La Figura 2.10 siguiente, indica el esquema de conexión física de cada equipo.

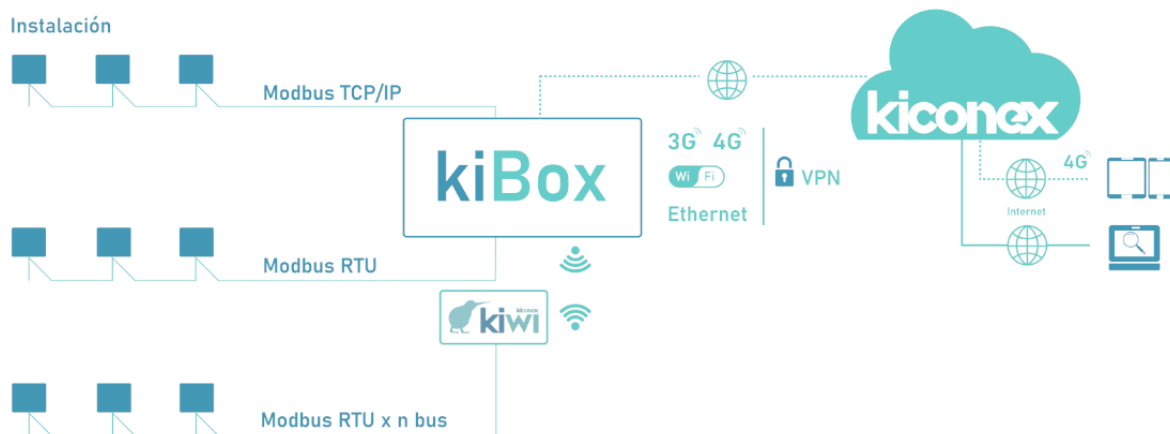


Figura 2.10: Esquema conexión equipos supermercado

La Figura 2.10 a Figura 2.10 muestra el proceso de creación de una instalación en la plataforma IoT de kiconex y la Figura 2.10 a Figura 2.10 el proceso de adición de equipos a la misma. Finalmente, la instalación tiene el aspecto de la Figuraxxx 2.10.



## Capítulo 3

# Pruebas y Resultados

### 3.1. Pruebas control UTA

Tras realizar el programa y el diseño de la pantalla para el control de la UTA, estos se cargan en un simulador que se compone de un iPro y una pantalla para pruebas. Dado que el programa se ha simulado previamente en el software ISaGRAF, no se han encontrado problemas de funcionamiento en él, sin embargo, es habitual y así ha sido en este caso, encontrar errores de diseño en la pantalla, que se han pasado por alto durante su creación. Estos problemas se han corregido desde el software Visoprog y se han vuelto a cargar.

Cuando el resultado de las pruebas es bueno, desde fábrica realizan un cuadro nuevo con iPro, al que se le carga el programa realizado tras comprobar que el iPro funciona correctamente. Este cuadro se envía al cliente, en este caso el instalador al cargo del supermercado.

### 3.2. Pruebas kiwi

En las primeras pruebas del kiwi, se han descubierto varios problema y/o necesidades:

- La necesidad de borrar la configuración guardada sin necesidad de acceder al portal web de configuración. Para ello se han añadido al programa las líneas necesarias para detectar la pulsación del botón integrado en la placa durante 5 segundos, lo que indica que se quiere resetear el kiwi y borrar la configuración realizada.
- En esa primera versión, no se disponía de la posibilidad de conexión por cable, pero se

decidió añadir dicha opción dado que dependiendo del equipo y la instalación puede ser más rentable usar un dispositivo como el kiwi conectado vía ethernet.

- Timeout de mensajes. Cada vez que se recibían mensajes para un dispositivo, kiwi abría un canal para los mensajes destinados a ese dispositivo, pero no volvía a cerrarlo, por lo que se ha añadido un tiempo (timeout) tras el cual, si no se vuelven a recibir mensajes a través de ese canal, se vuelve a cerrar.
- Problema de cambio de modo (AP y cliente). Para la gestión de la conexión de red, WiFi o Ethernet, se han empleado las librerías estándar para el ESP32: WiFi.h, ETH.h, Wi-FiClient.h. Al parecer estas librerías no gestionan bien el paso de un modo a otro, por lo que el ESP32 parecía dejar de funcionar. La solución ha sido reiniciar el kiwi tras la configuración, para volver a ponerse en marcha desde cero, en el nuevo modo, a partir de los parámetros de configuración guardados en la EEPROM.
- Desconexiones puntuales de la red WiFi. En el kiwi instalado para pruebas en la red de la empresa Intarcon, se detectaron desconexiones continuas de la red WiFi. Este problema dio muchos dolores de cabeza, ya que el kiwi parecía funcionar sin problemas en otras redes. Para analizar este caso, se conectó un segundo kiwi en una red aislada, empleando un router de los empleados por kiconex para aquellas instalaciones sin acceso a la red, y se registraron en una hoja las desconexiones de ambos kiwis.

El resultado fue el de la Figura 3.1 y la Figura 3.2, donde se puede ver que el kiwi de la red del router no tuvo ningún problema, mientras que el kiwi de la red de Intarcon tenía desconexiones continuas durante el horario de trabajo. Fue este último dato el decisivo para saber que el problema estaba en la red de Intarcon, por lo que lo consultamos con el informático al cargo de la misma: nos confirmó que tras unos cambios en la red, había problemas con las DNS, y muchos equipos eran expulsados de la red. Este problema aun persiste, aunque en menor medida, pero el kiwi conectado al router sirve de muestra del buen funcionamiento del mismo.

The screenshot shows the 'CATEGORÍA' tab selected in a configuration interface. The top navigation bar includes icons and labels for GENERAL, CATEGORÍA, UNIDADES, MODBUS, METADATOS, and NOTAS. The main area contains two dropdown menus: the first is labeled '\* Categoría' and the second is labeled 'Grupo'. Both dropdowns currently display the text 'Categoría' and 'Grupo' respectively, with a downward arrow on the right of each.

Figura 3.1: Desconexiones kiwi en red de Intarcon

This screenshot is identical to the one in Figure 3.1, showing the 'CATEGORÍA' tab in a configuration interface. It features the same top navigation bar and two dropdown menus labeled '\* Categoría' and 'Grupo', both showing their respective default values.

Figura 3.2: Desconexiones kiwi en red de router kiconex

### 3.3. Puesta en marcha de la instalación

Cuando se conectan todos los dispositivos de la instalación, es momento de su puesta en marcha y de comprobar que todo está correctamente. Se trata de un trámite necesario, ya que aunque normalmente no da problemas, existen ocasiones en las que falla la conexión por causas ajenas a kiconex: resistencias de extremo de cable modbus, direcciones Modbus no configuradas en los equipos correctamente, etc. En este caso, el mapa de direcciones Modbus es el de la Tabla 3.1.

EQUIPO	DIRECCIÓN
Ventiladores	1
Filtros	2
Humectador	3
Intercambiador de calor	4
E/S de aire	5

Tabla 3.1: Direcciones Modbus equipos supermercado.

La Figura 3.3 muestra como todos los elementos de la instalación se han conectado correctamente.

GENERAL CATEGORÍA UNIDADES MODBUS METADATOS NOTAS

\* Categoría Categoría

Grupo Grupo

Figura 3.3: Instalación supermercado conectada y funcionando



## Capítulo 4

# Conclusiones y Trabajos Futuros

Tras realizar el programa y el diseño de la pantalla para el control de la UTA, estos se cargan en un simulador que se compone de un iPro y una pantalla para pruebas. Dado que el programa se ha simulado previamente en el software ISaGRAF, no se han encontrado problemas de funcionamiento en él, sin embargo, es habitual y así ha sido en este caso, encontrar errores de diseño en la pantalla, que se han pasado por alto durante su creación. Estos problemas se han corregido desde el software Visoprog y se han vuelto a cargar.

Cuando el resultado de las pruebas es bueno, desde fábrica realizan un cuadro nuevo con iPro, al que se le carga el programa realizado tras comprobar que el iPro funciona correctamente. Este cuadro se envía al cliente, en este caso el instalador al cargo del supermercado.



## Bibliografía

- [1] *Plataforma IoT Kiconex*. dirección: <https://app.kiconex.com/>.
- [2] *Productos Kiconex*. dirección: <https://www.kiconex.com/producto/>.
- [3] *Protocolo Modbus*. dirección: <http://www.modbus.org/>.
- [4] *Manual iPro*. dirección: <https://climate.emerson.com/documents/ipro-series-en-4923358.pdf>.
- [5] *Marca Dixell*. dirección: <https://climate.emerson.com/es-es/brands/dixell>.
- [6] *¿Qué es kiconex?* Dirección: <https://www.kiconex.com/que-es-kiconex/>.
- [7] *Web de Espressif sobre su ESP32*. dirección: <https://www.espressif.com/en/products/socs/esp32/overview>.
- [8] *Pantalla VTIPG*. dirección: <https://climate.emerson.com/es-es/shop/1/dixell-electronics-sku-vtipg-hmi-es-es>.
- [9] *Programa Visoprogram*. dirección: <https://climate.emerson.com/es-es/shop/1/dixell-electronics-sku-visoprogram-es-es>.
- [10] *Empresa Logicbus hablando sobre ISaGRAF*. dirección: <https://www.logicbus.com.mx/isagraf.php>.
- [11] *Página de producto de la placa Olimex ESP32-PoE*. dirección: <https://www.olimex.com/Products/IoT/ESP32/ESP32-POE/open-source-hardware>.
- [12] *Página de producto del conversor UEXT-RS485*. dirección: <https://www.olimex.com/Products/Modules/Interface/MOD-RS485/open-source-hardware>.
- [13] *S. Marco, Librería Modbus RTU para ESP32 en GitHub*. dirección: <https://github.com/smarmengol/Modbus-Master-Slave-for-Arduino>.

## BIBLIOGRAFÍA

---

- [14] K. Hoang, *Librería de la interfaz de conexión WiFi en GitHub*. dirección: [https://github.com/khoih-prog/ESP\\_WiFiManager](https://github.com/khoih-prog/ESP_WiFiManager).

---

# **ANEXOS**



## Anexo A

# Las Unidades de Tratamiento de Aire

Una UTA es un equipo encargado de tratar el aire atendiendo a distintos aspectos de la climatización:

- Ventilación
- Filtrado
- Control de temperatura
- Control de humedad

Su funcionamiento da la posibilidad de regular el caudal de ventilación en función de la medición del CO<sub>2</sub> y de las condiciones térmicas del local, así como recuperar parte de la energía térmica del aire que se expulsa al exterior.

Las partes de las que se compone una UTA son:

- Compuertas o tomas de aire para la entrada y salida del aire.
- Zona de mezcla del aire de retorno y del aire exterior para permitir recuperar energía mediante lo que se conoce como *free-cooling*. La recuperación puede ser de distintos tipos:
  - Recuperación aire-agua: el aire extraído transfiere su calor al agua a través de un intercambiador. El mismo agua transfiere ese calor al aire introducido desde el exterior, el cual fluye por otro conducto. Este sistema es más salubre al impedir que se mezclen ambas corrientes de aire, pero tiene menor rendimiento.

- Recuperación mediante placas aire-aire: a través de un cubo metálico pasan ambas corrientes de aire a modo de flujo cruzado. El calor se transfiere a través del contacto del aire con las placas metálicas de los canales.
- Intercambiador de calor, por donde fluye agua a la temperatura deseada (frío o calor). La temperatura del agua se transfiere al aire cuando éste pasa a través del intercambiador.
- Filtros, que pueden ser de distintos tipos en función de las especificaciones de filtrado.
- Humectador: el paso del aire seco y caliente por un panel mojado hace que se evapore parte de su agua y se produzca aire húmedo y frío. Dispone de una bandeja de almacenamiento y recogida del agua.
- Zona de ventilación con ventilador y motor.

La siguiente figura ilustra a modo de ejemplo todos los elementos mencionados:

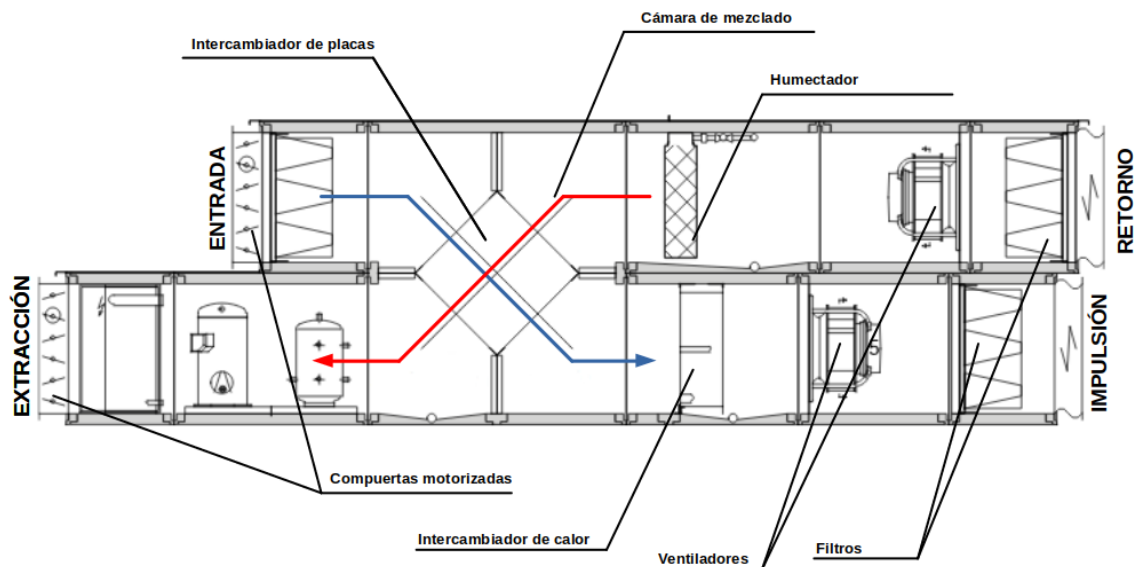


Figura A.1: Plano de partes de una UTA



## Anexo B

# Programa Unidad de Tratamiento de Aire

Una UTA es un equipo encargado de tratar el aire atendiendo a distintos aspectos de la climatización:

- Ventilación
- Filtrado
- Control de temperatura
- Control de humedad

Su funcionamiento da la posibilidad de regular el caudal de ventilación en función de la medición del CO<sub>2</sub> y de las condiciones térmicas del local, así como recuperar parte de la energía térmica del aire que se expulsa al exterior.

Las partes de las que se compone una UTA son:

- Compuertas o tomas de aire para la entrada y salida del aire.
- Zona de mezcla del aire de retorno y del aire exterior para permitir recuperar energía mediante lo que se conoce como *free-cooling*. La recuperación puede ser de distintos tipos:
  - Recuperación aire-agua: el aire extraído transfiere su calor al agua a través de un intercambiador. El mismo agua transfiere ese calor al aire introducido desde el exterior, el cual fluye por otro conducto. Este sistema es más salubre al impedir que se mezclen ambas corrientes de aire, pero tiene menor rendimiento.

- Recuperación mediante placas aire-aire: a través de un cubo metálico pasan ambas corrientes de aire a modo de flujo cruzado. El calor se transfiere a través del contacto del aire con las placas metálicas de los canales.
- Intercambiador de calor, por donde fluye agua a la temperatura deseada (frío o calor). La temperatura del agua se transfiere al aire cuando éste pasa a través del intercambiador.
- Filtros, que pueden ser de distintos tipos en función de las especificaciones de filtrado.
- Humectador: el paso del aire seco y caliente por un panel mojado hace que se evapore parte de su agua y se produzca aire húmedo y frío. Dispone de una bandeja de almacenamiento y recogida del agua.
- Zona de ventilación con ventilador y motor.

La siguiente figura ilustra a modo de ejemplo todos los elementos mencionados:

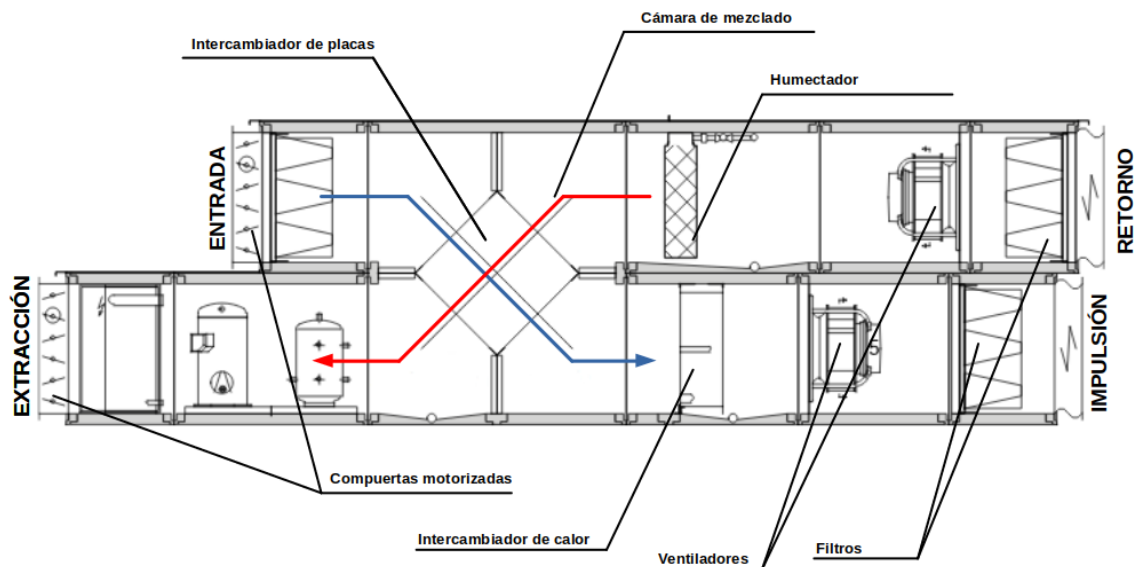


Figura B.1: Plano de partes de una UTA

## Anexo C

# Registros Control Unidad de Tratamiento de Aire

Este anexo recoge los parámetros necesarios para el control de la Unidad de Tratamiento de Aire programada en el proyecto:

### **Pb\_T\_IMP\_AIRE**

(\* temperatura impulsión de aire \*)

Grupo: ENTRADAS\_SALIDAS

Tipo: DINT

Atributo: Libre

Sentido: Memory

Dirección: 1000

### **Pb\_T\_RET\_AIRE**

(\* temperatura retorno de aire \*)

Grupo: ENTRADAS\_SALIDAS

Tipo: DINT

Atributo: Libre

Sentido: Memory

Dirección: 1001

**Pb\_T\_IMP\_AGUA**

(\* temperatura impulsión fluido \*)

Grupo: ENTRADAS\_SALIDAS

Tipo: DINT

Atributo: Libre

Sentido: Memory

Dirección: 1002

**Pb\_T\_RET\_AGUA**

(\* temperatura retorno fluido \*)

Grupo: ENTRADAS\_SALIDAS

Tipo: DINT

Atributo: Libre

Sentido: Memory

Dirección: 1003

**Pb\_T\_EXTRAC\_AIRE**

(\* temperatura extracción de aire \*)

Grupo: ENTRADAS\_SALIDAS

Tipo: DINT

Atributo: Libre

Sentido: Memory

Dirección: 1004

**Pb\_T\_EXT\_AIRE**

(\* temperatura aire exterior \*)

Grupo: ENTRADAS\_SALIDAS

Tipo: DINT

Atributo: Libre

Sentido: Memory

---

Dirección: 1005

**Pb\_H\_RET**

(\* humedad exterior \*)

Grupo: ENTRADAS\_SALIDAS

Tipo: DINT

Atributo: Libre

Sentido: Memory

Dirección: 1006

**Pb\_H\_EXT**

(\* humedad retorno \*)

Grupo: ENTRADAS\_SALIDAS

Tipo: DINT

Atributo: Libre

Sentido: Memory

Dirección: 1007

**AO\_VENT\_IMP**

(\* Ventilador impulsión de aire \*)

Grupo: ENTRADAS\_SALIDAS

Tipo: DINT

Atributo: Libre

Sentido: Memory

Dirección: 1080

**AO\_VENT\_RET**

(\* Ventilador retorno aire \*)

Grupo: ENTRADAS\_SALIDAS

Tipo: DINT

Atributo: Libre

Sentido: Memory

Dirección: 1081

### **DI\_ONOFF**

(\* On-OFF remoto \*)

Grupo: ENTRADAS\_SALIDAS

Tipo: BOOL

Atributo: Libre

Sentido: Memory

Dirección: 1100

### **DI\_VENT\_IMP**

(\* Seguridad ventilador imp. aire \*)

Grupo: ENTRADAS\_SALIDAS

Tipo: BOOL

Atributo: Libre

Sentido: Memory

Dirección: 1101

### **DI\_VENT\_RET**

(\* Seguridad ventilador ret. aire \*)

Grupo: ENTRADAS\_SALIDAS

Tipo: BOOL

Atributo: Libre

Sentido: Memory

Dirección: 1102

### **DI\_NIVEL\_HUM**

(\* Sensor nivel humectador \*)

---

Grupo: ENTRADAS\_SALIDAS

Tipo: BOOL

Atributo: Libre

Sentido: Memory

Dirección: 1103

### **DI\_FILT\_ENT**

(\* Filtro de entrada de aire sucio \*)

Grupo: ENTRADAS\_SALIDAS

Tipo: BOOL

Atributo: Libre

Sentido: Memory

Dirección: 1104

### **DI\_FILT\_IMP**

(\* Filtro de impulsión de aire sucio \*)

Grupo: ENTRADAS\_SALIDAS

Tipo: BOOL

Atributo: Libre

Sentido: Memory

Dirección: 1105

### **DI\_FILT\_RET**

(\* Filtro de retorno de aire sucio \*)

Grupo: ENTRADAS\_SALIDAS

Tipo: BOOL

Atributo: Libre

Sentido: Memory

Dirección: 1106

**RELE\_VALVE3V**

(\* Conmutación válvula \*)

Grupo: ENTRADAS\_SALIDAS

Tipo: BOOL

Atributo: Libre

Sentido: Memory

Dirección: 1200

**RELE\_HUMECT**

(\* Activación/Desactivación de la humectación \*)

Grupo: ENTRADAS\_SALIDAS

Tipo: BOOL

Atributo: Libre

Sentido: Memory

Dirección: 1201

**RELE\_COMP\_IMPUL**

(\* Apertura/Cierre Compuerta de Impulsión \*)

Grupo: ENTRADAS\_SALIDAS

Tipo: BOOL

Atributo: Libre

Sentido: Memory

Dirección: 1202

**RELE\_COMP\_RET**

(\* Apertura/Cierre Compuerta de retorno \*)

Grupo: ENTRADAS\_SALIDAS

Tipo: BOOL

Atributo: Libre

Sentido: Memory



---

Dirección: 1203

### **RELE\_ALARM**

(\* Activación / Desactivación Alarma \*)

Grupo: ENTRADAS\_SALIDAS

Tipo: BOOL

Atributo: Libre

Sentido: Memory

Dirección: 1204

### **ONOFF\_GEN**

Grupo: ESTADOS

Tipo: BOOL

Atributo: Libre

Sentido: Memory

Dirección: 1402

### **PAR\_SET**

(\* Consigna \*)

Grupo: PARAMETROS\_CONFIGURACION

Tipo: DINT

Atributo: Libre

Sentido: Memory

No volátil: SI

Dirección: 1508

### **PAR\_SET\_HUM**

(\* Consigna de diferencia de humedad con el exterior \*)

Grupo: PARAMETROS\_CONFIGURACION

Tipo: DINT

Atributo: Libre

Sentido: Memory

No volátil: SI

Dirección: 1509

### **PAR\_VEL\_BAJA**

(\* Configuración de velocidad baja \*)

Grupo: PARAMETROS\_CONFIGURACION

Tipo: DINT

Atributo: Libre

Sentido: Memory

No volátil: SI

Dirección: 150B

### **PAR\_VEL\_MEDIA**

(\* Configuración de velocidad media \*)

Grupo: PARAMETROS\_CONFIGURACION

Tipo: DINT

Atributo: Libre

Sentido: Memory

No volátil: SI

Dirección: 150C

### **PAR\_VEL\_ALTA**

(\* Configuración de velocidad alta \*)

Grupo: PARAMETROS\_CONFIGURACION

Tipo: DINT

Atributo: Libre

Sentido: Memory

No volátil: SI

---

Dirección: 150D

### **PAR\_VEL\_MIN**

(\* Velocidad mínima en los ventiladores \*)

Grupo: PARAMETROS\_CONFIGURACION

Tipo: DINT

Atributo: Libre

Sentido: Memory

No volátil: SI

Dirección: 150E

### **PAR\_VEL\_MAX**

(\* Velocidad máxima en los ventiladores \*)

Grupo: PARAMETROS\_CONFIGURACION

Tipo: DINT

Atributo: Libre

Sentido: Memory

No volátil: SI

Dirección: 150F

### **PAR\_BANDA\_VEL**

(\* Ancho de banda para velocidad máxima \*)

Grupo: PARAMETROS\_CONFIGURACION

Tipo: DINT

Atributo: Libre

Sentido: Memory

No volátil: SI

Dirección: 1510

### **PAR\_BAND\_INV**

(\* Temperatura de banda en modo invierno \*)

Grupo: PARAMETROS\_CONFIGURACION

Tipo: DINT

Atributo: Libre

Sentido: Memory

No volátil: SI

Dirección: 1512

### **PAR\_BAND\_VER**

(\* Temperatura de banda en modo verano \*)

Grupo: PARAMETROS\_CONFIGURACION

Tipo: DINT

Atributo: Libre

Sentido: Memory

No volátil: SI

Dirección: 1513

### **PAR\_BAND\_HUM**

(\* Banda humedad \*)

Grupo: PARAMETROS\_CONFIGURACION

Tipo: DINT

Atributo: Libre

Sentido: Memory

No volátil: SI

Dirección: 1514

### **PAR\_TEMP\_FIL1**

(\* Temporización para alarma filtro entrada aire(no usada) \*)

Grupo: PARAMETROS\_CONFIGURACION

Tipo: DINT

---

Atributo: Libre

Sentido: Memory

Dirección: 1515

### **PAR\_TEMP\_FIL2**

(\* Temporización para alarma filtro impulsión aire(no usada) \*)

Grupo: PARAMETROS\_CONFIGURACION

Tipo: DINT

Atributo: Libre

Sentido: Memory

Dirección: 1516

### **PAR\_TEMP\_FIL3**

(\* Temporización para alarma filtro impulsión aire(no usada) \*)

Grupo: PARAMETROS\_CONFIGURACION

Tipo: DINT

Atributo: Libre

Sentido: Memory

Dirección: 1517

### **PAR\_TOFF\_AL\_FAN1**

Grupo: PARAMETROS\_CONFIGURACION

Tipo: DINT

Atributo: Libre

Sentido: Memory

No volátil: SI

Dirección: 1518

### **PAR\_TON\_AL\_FAN1**

Grupo: PARAMETROS\_CONFIGURACION

Tipo: DINT

Atributo: Libre

Sentido: Memory

No volátil: SI

Dirección: 1519

### **PAR\_NUM\_INTENTOS\_FAN1**

Grupo: PARAMETROS\_CONFIGURACION

Tipo: DINT

Atributo: Libre

Sentido: Memory

No volátil: SI

Dirección: 151A

### **PAR\_TOFF\_AL\_FAN2**

Grupo: PARAMETROS\_CONFIGURACION

Tipo: DINT

Atributo: Libre

Sentido: Memory

No volátil: SI

Dirección: 151B

### **PAR\_TON\_AL\_FAN2**

Grupo: PARAMETROS\_CONFIGURACION

Tipo: DINT

Atributo: Libre

Sentido: Memory

No volátil: SI

Dirección: 151C

---

### **PAR\_NUM\_INTENTOS\_FAN2**

Grupo: PARAMETROS\_CONFIGURACION

Tipo: DINT

Atributo: Libre

Sentido: Memory

No volátil: SI

Dirección: 151D

### **PAR\_TON\_AL\_FIL1**

Grupo: PARAMETROS\_CONFIGURACION

Tipo: DINT

Atributo: Libre

Sentido: Memory

No volátil: SI

Dirección: 151E

### **PAR\_TON\_AL\_FIL2**

Grupo: PARAMETROS\_CONFIGURACION

Tipo: DINT

Atributo: Libre

Sentido: Memory

No volátil: SI

Dirección: 151F

### **PAR\_TON\_AL\_FIL3**

Grupo: PARAMETROS\_CONFIGURACION

Tipo: DINT

Atributo: Libre

Sentido: Memory

No volátil: SI

Dirección: 1520

**PAR\_TOFF\_AL\_FIL1**

Grupo: PARAMETROS\_CONFIGURACION

Tipo: DINT

Atributo: Libre

Sentido: Memory

No volátil: SI

Dirección: 1521

**PAR\_TOFF\_AL\_FIL2**

Grupo: PARAMETROS\_CONFIGURACION

Tipo: DINT

Atributo: Libre

Sentido: Memory

No volátil: SI

Dirección: 1522

**PAR\_TOFF\_AL\_FIL3**

Grupo: PARAMETROS\_CONFIGURACION

Tipo: DINT

Atributo: Libre

Sentido: Memory

No volátil: SI

Dirección: 1523

**PAR\_NUM\_INTENTOS\_FIL1**

Grupo: PARAMETROS\_CONFIGURACION

Tipo: DINT

Atributo: Libre



---

Sentido: Memory

No volátil: SI

Dirección: 1524

### **PAR\_NUM\_INTENTOS\_FIL2**

Grupo: PARAMETROS\_CONFIGURACION

Tipo: DINT

Atributo: Libre

Sentido: Memory

No volátil: SI

Dirección: 1525

### **PAR\_NUM\_INTENTOS\_FIL3**

Grupo: PARAMETROS\_CONFIGURACION

Tipo: DINT

Atributo: Libre

Sentido: Memory

No volátil: SI

Dirección: 1526

### **SONDAS**

Dimensión: [1..27]

Grupo: PARAMETROS\_CONFIGURACION

Tipo: DINT

Atributo: Libre

Sentido: Memory

No volátil: SI

Dirección: 1600

### **ENTRADAS\_DIGITALES**

Dimensión: [1..43]

Grupo: PARAMETROS\_CONFIGURACION

Tipo: DINT

Atributo: Libre

Sentido: Memory

No volátil: SI

Dirección: 1630

### **POLARIDAD\_DI**

Dimensión: [1..43]

Grupo: PARAMETROS\_CONFIGURACION

Tipo: BOOL

Atributo: Libre

Sentido: Memory

No volátil: SI

Dirección: 1660

### **RELE\_SALIDA**

Dimensión: [1..36]

Grupo: PARAMETROS\_CONFIGURACION

Tipo: DINT

Atributo: Libre

Sentido: Memory

No volátil: SI

Dirección: 1690

### **SALIDAS\_ANALOG**

Dimensión: [1..15]

Grupo: PARAMETROS\_CONFIGURACION

Tipo: DINT

---

Atributo: Libre

Sentido: Memory

No volátil: SI

Dirección: 16C0

### **ALARM\_VENT\_IMPULSION**

Grupo: ALARMAS

Tipo: BOOL

Atributo: Libre

Sentido: Memory

Dirección: 1703

### **ALARM\_VENT\_RETORNO**

Grupo: ALARMAS

Tipo: BOOL

Atributo: Libre

Sentido: Memory

Dirección: 1704

### **WARNING\_VENT\_IMPULSION**

Grupo: WARNINGS

Tipo: BOOL

Atributo: Libre

Sentido: Memory

Dirección: 1A06

### **WARNING\_VENT\_RETORNO**

Grupo: WARNINGS

Tipo: BOOL

Atributo: Libre

Sentido: Memory

Dirección: 1A07

### **WARNING\_FILTRO\_ENTRADA**

Grupo: WARNINGS

Tipo: BOOL

Atributo: Libre

Sentido: Memory

Dirección: 1A08

### **WARNING\_FILTRO\_IMPULSION**

Grupo: WARNINGS

Tipo: BOOL

Atributo: Libre

Sentido: Memory

Dirección: 1A09

### **WARNING\_FILTRO\_RETORNO**

Grupo: WARNINGS

Tipo: BOOL

Atributo: Libre

Sentido: Memory

Dirección: 1A0A

### **WARNING\_H\_FLUID\_TMP**

Grupo: WARNINGS

Tipo: BOOL

Atributo: Libre

Sentido: Memory

Dirección: 1A0B

---

### **WARNING\_L\_FLUID\_TMP**

Grupo: WARNINGS

Tipo: BOOL

Atributo: Libre

Sentido: Memory

Dirección: 1A0C

### **KB\_ONOFF**

Grupo: COMANDOS

Tipo: BOOL

Atributo: Libre

Sentido: Memory

Dirección: 1C00

### **RESET\_ALARM\_MANUAL**

Grupo: COMANDOS

Tipo: BOOL

Atributo: Libre

Sentido: Memory

Dirección: 1C01

### **KB\_VEL**

(\* Selección de velocidad: 0,1,2,3(AUTO) \*)

Grupo: COMANDOS

Tipo: DINT

Atributo: Libre

Sentido: Memory

Dirección: 1C06

### **KB\_MODAL**

(\* Modo verano/invierno \*)

Grupo: COMANDOS

Tipo: BOOL

Atributo: Libre

Sentido: Memory

Dirección: 1C07

## Anexo D

# Librería KiWiModbusBridge

Una UTA es un equipo encargado de tratar el aire atendiendo a distintos aspectos de la climatización:

- Ventilación
- Filtrado
- Control de temperatura
- Control de humedad

Su funcionamiento da la posibilidad de regular el caudal de ventilación en función de la medición del CO<sub>2</sub> y de las condiciones térmicas del local, así como recuperar parte de la energía térmica del aire que se expulsa al exterior.

Las partes de las que se compone una UTA son:

- Compuertas o tomas de aire para la entrada y salida del aire.
- Zona de mezcla del aire de retorno y del aire exterior para permitir recuperar energía mediante lo que se conoce como *free-cooling*. La recuperación puede ser de distintos tipos:
  - Recuperación aire-agua: el aire extraído transfiere su calor al agua a través de un intercambiador. El mismo agua transfiere ese calor al aire introducido desde el exterior, el cual fluye por otro conducto. Este sistema es más salubre al impedir que se mezclen ambas corrientes de aire, pero tiene menor rendimiento.

- Recuperación mediante placas aire-aire: a través de un cubo metálico pasan ambas corrientes de aire a modo de flujo cruzado. El calor se transfiere a través del contacto del aire con las placas metálicas de los canales.
- Intercambiador de calor, por donde fluye agua a la temperatura deseada (frío o calor). La temperatura del agua se transfiere al aire cuando éste pasa a través del intercambiador.
- Filtros, que pueden ser de distintos tipos en función de las especificaciones de filtrado.
- Humectador: el paso del aire seco y caliente por un panel mojado hace que se evapore parte de su agua y se produzca aire húmedo y frío. Dispone de una bandeja de almacenamiento y recogida del agua.
- Zona de ventilación con ventilador y motor.

La siguiente figura ilustra a modo de ejemplo todos los elementos mencionados:

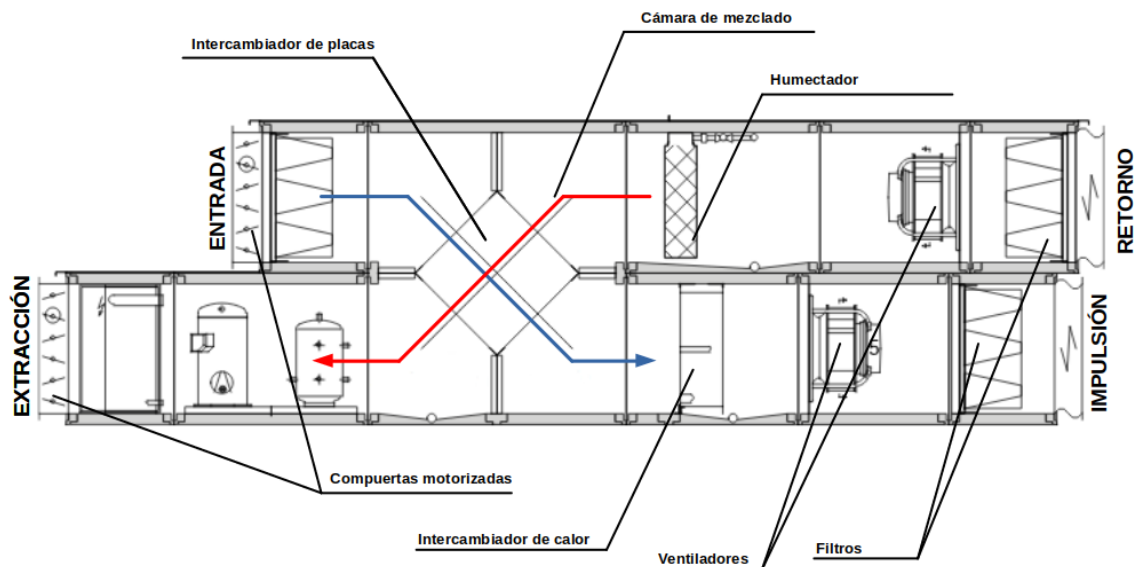


Figura D.1: Plano de partes de una UTA