

## **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>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>7</b>
1.1. iPro y su pantalla . . . . .	9
1.1.1. Programación iPro . . . . .	10
1.1.2. Configuración de la pantalla . . . . .	12
1.2. Kiconex . . . . .	13
1.3. ESP32-PoE . . . . .	16
1.4. Protocolo Modbus . . . . .	17
<b>2. Desarrollo del proyecto</b>	<b>19</b>
2.1. Programación iPro . . . . .	19
2.2. Diseño Pantalla para el control . . . . .	19
2.3. Elaboración de librerías en Kiconex . . . . .	20
2.4. Programación dispositivo wireless . . . . .	20

<b>3. Pruebas y Resultados</b>	<b>21</b>
<b>4. Puesta en marcha</b>	<b>23</b>
<b>5. Conclusiones y Trabajos Futuros</b>	<b>25</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.	Página con enlaces a hilos . . . . .	5
1.1.	Plano de equipos del supermercado . . . . .	7
1.2.	iPro Series . . . . .	9
1.3.	Aspecto tabla de variables en ISaGRAF . . . . .	11
1.4.	Ventanas pantalla iPro . . . . .	12
1.5.	Aspecto interfaz Visoprog . . . . .	13
1.6.	Red kiconex . . . . .	14
1.7.	Aspecto librería en la plataforma IoT . . . . .	15
1.8.	Olimex ESP32-PoE . . . . .	16
1.9.	Conversor UEXT-RS485 . . . . .	16
1.10.	Pines ESP32-PoE y conector UEXT . . . . .	17





# 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) 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.

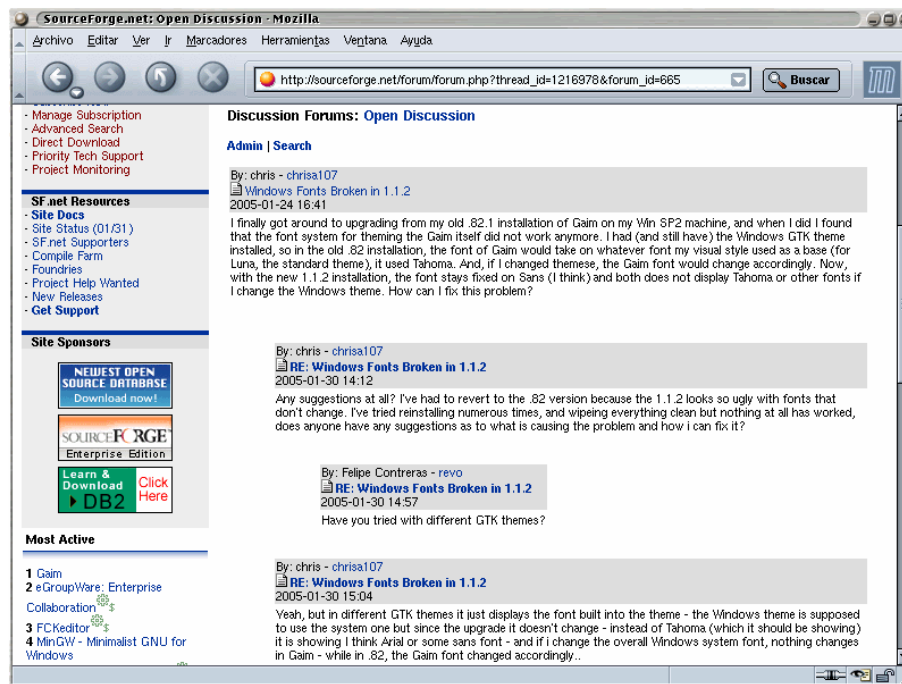


Figura 1: Página con enlaces a hilos

Recomiendo leer los consejos prácticos sobre escribir documentos científicos en  $\text{\LaTeX}$  de Diomidis Spinellis<sup>1</sup>.

Lee sobre el uso de las comas<sup>II</sup>. Las comas en español no se ponen al tuntún. Y nunca, nunca entre el objeto y el predicado (p.ej. en “Yo, hago el TFG” sobre la coma).

A continuación, viene una figura, la Figura 1. Observarás que el texto dentro de la referencia es el identificador de la figura (que se corresponden con el “label” dentro de la misma). También habrás tomado nota de cómo se ponen las “comillas dobles” para que se muestren correctamente. Nota que hay unas comillas de inicio (“) y otras de cierre (”), y que son diferentes. Volviendo a las referencias, nota que al compilar, la primera vez se crea un diccionario con las referencias, y en la segunda compilación se “rellenan” estas referencias. Por eso hay que compilar dos veces tu memoria. Si no, no se crearán las referencias.

A continuación un bloque “verbatim”, que se utiliza para mostrar texto tal cual. Se puede utilizar para ofrecer el contenido de correos electrónicos, código, entre otras cosas.

<sup>1</sup><https://github.com/dspinellis/latex-advice>

<sup>II</sup><http://narrativabreve.com/2015/02/opiniones-de-un-corrector-de-estilo-11-recetas-par.html>



# 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, ha solicitado el diseño de un control a medida para el funcionamiento de una UTA. La instalación cuenta con los elementos de la Figura 1.1:

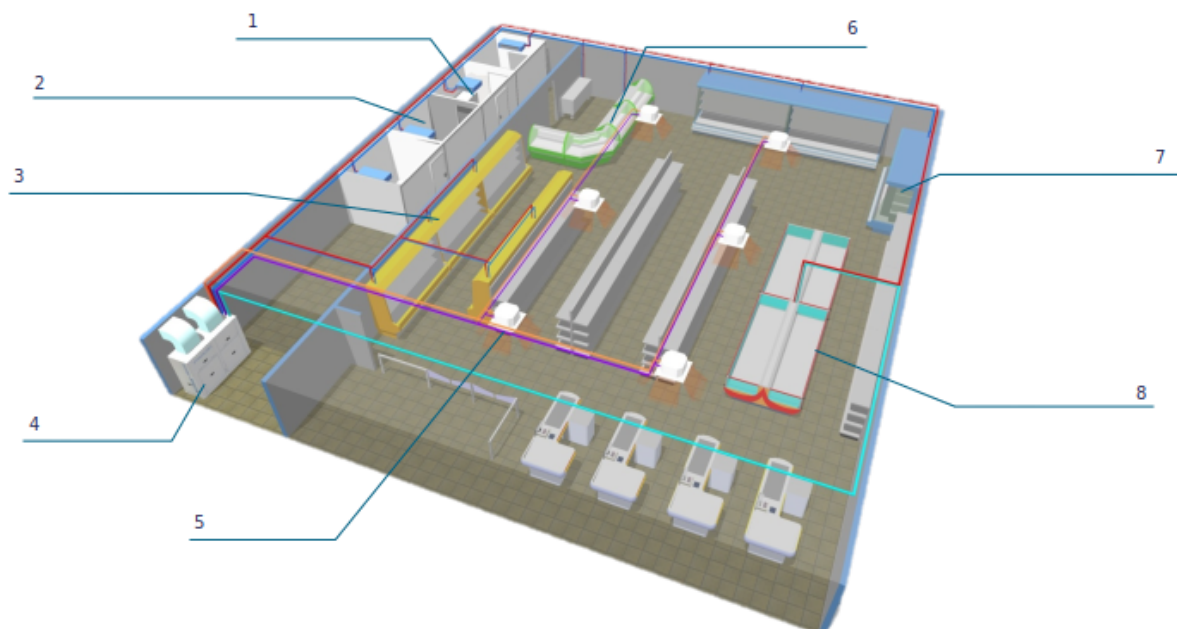


Figura 1.1: Plano de equipos del supermercado

1. Obradores.
2. Cámaras frigoríficas.
3. Murales de lácteos.
4. Bomba de calor.
5. Unidades de Tratamiento de Aire.
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.



## 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*:

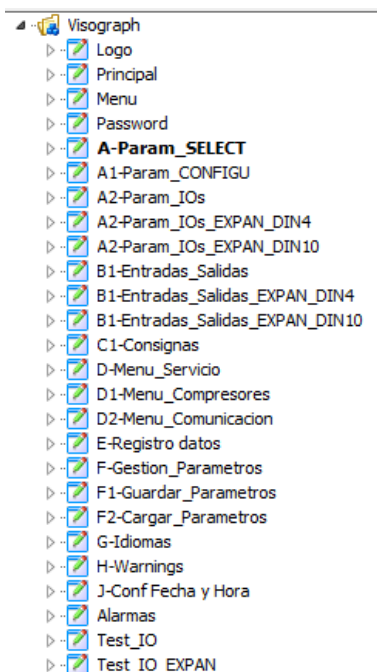


Figura 1.4: Ventanas pantalla iPro

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.

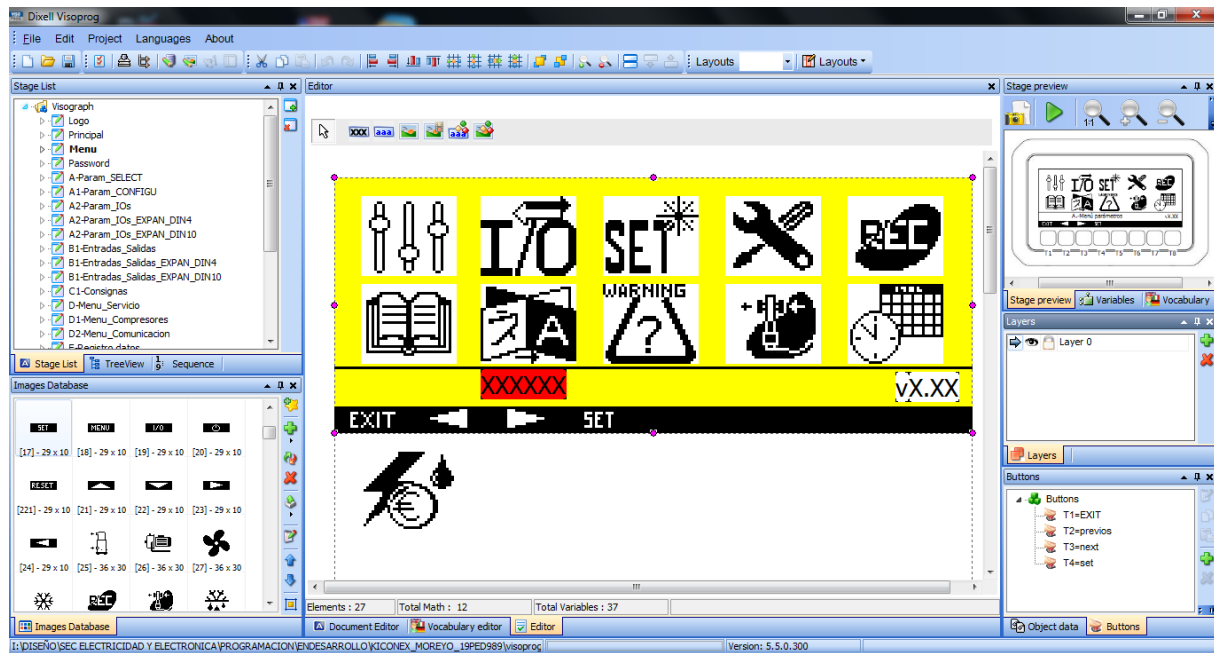


Figura 1.5: Aspecto interfaz Visoprog

## 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.

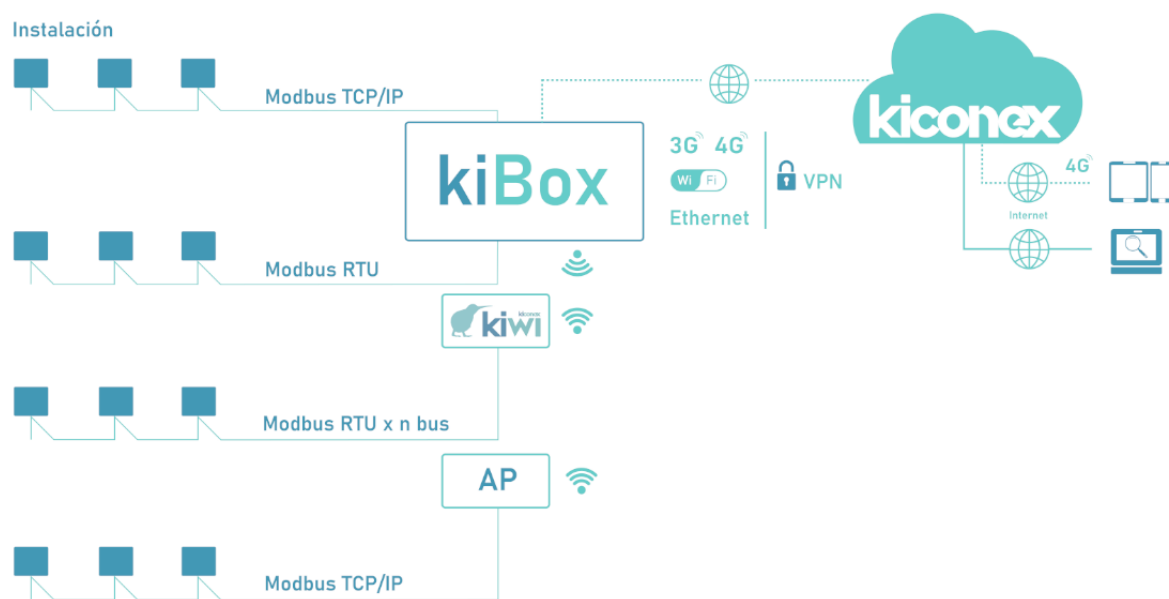
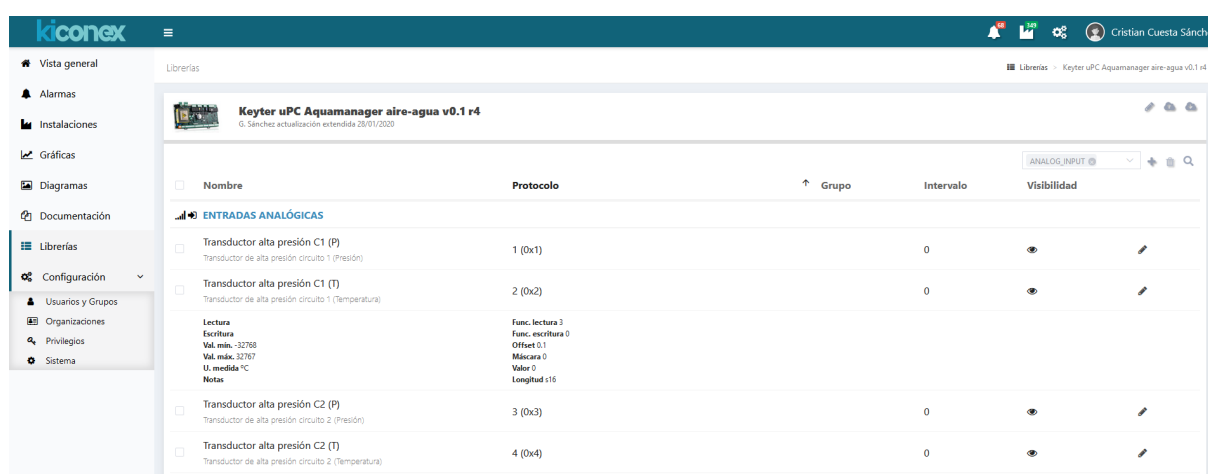


Figura 1.6: Red kiconex

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

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.

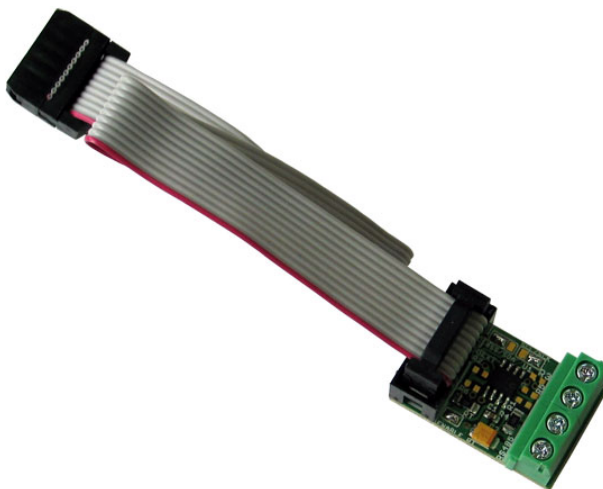
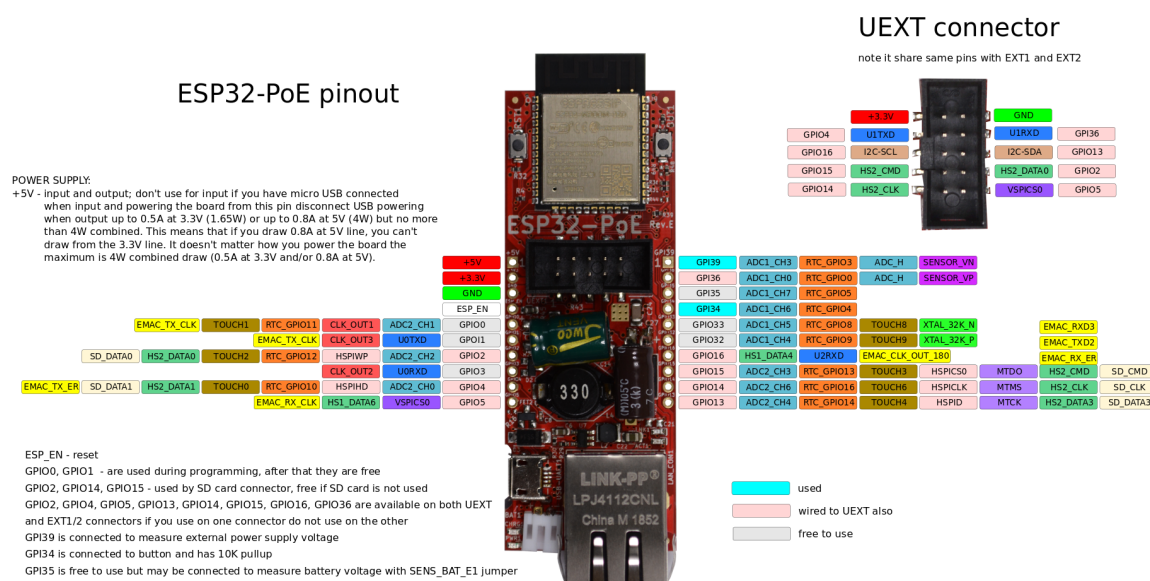


Figura 1.9: Conversor UEXT-RS485





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

Capítulo de desarrollo de cada uno de los apartados del proyecto.

### 2.1. Programación iPro

Como ya se ha explicado en los capítulos anteriores, el destino del iPro es la programación de varias UTAs existentes en el supermercado. Existen dos tipos de UTA, la que solo coge el aire de la estancia para volver a introducirlo a la temperatura deseada, y la que tiene un sistema de renovación de aire: conduce el aire de la estancia al exterior e introduce el aire desde el exterior, calentándolo o enfriándolo primero. La figura xxx representa ambos sistemas.

El objetivo es realizar un programa estándar que se pueda usar para ambos tipos de UTA, para así evitar realizar nuevos programas en futuros proyectos que incluyan una UTA. Para ello, en la tabla xxx se han recogido las diferencias técnicas entre ambas, para saber hasta que punto se puede adaptar el programa a cualquier caso.

### 2.2. Diseño Pantalla para el control

indican las conclusiones generales obtenidas a partir del estudio realizado, se proponen líneas de trabajo futuro, y se indican las publicaciones asociadas al trabajo realizado en este TFM.

## **2.3. Elaboración de librerías en Kiconex**

indican las conclusiones generales obtenidas a partir del estudio realizado, se proponen líneas de trabajo futuro, y se indican las publicaciones asociadas al trabajo realizado en este TFM.

## **2.4. Programación dispositivo wireless**

indican las conclusiones generales obtenidas a partir del estudio realizado, se proponen líneas de trabajo futuro, y se indican las publicaciones asociadas al trabajo realizado en este TFM.

## Capítulo 3

# Pruebas y Resultados

indican las conclusiones generales obtenidas a partir del estudio realizado, se proponen líneas de trabajo futuro, y se indican las publicaciones asociadas al trabajo realizado en este TFM.



## Capítulo 4

### Puesta en marcha

indican las conclusiones generales obtenidas a partir del estudio realizado, se proponen líneas de trabajo futuro, y se indican las publicaciones asociadas al trabajo realizado en este TFM.





## Capítulo 5

# Conclusiones y Trabajos Futuros

indican las conclusiones generales obtenidas a partir del estudio realizado, se proponen líneas de trabajo futuro, y se indican las publicaciones asociadas al trabajo realizado en este TFM.



## 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).