



UNIVERSIDAD DE JAÉN

Escuela Politécnica Superior de Jaén

Trabajo Fin de Grado

DISEÑO Y PROGRAMACIÓN DE UNA INSTALACIÓN DOMÓTICA EN UNA VIVIENDA UNIFAMILIAR MEDIANTE TECNOLOGÍA KNX

Alumno: María Ramírez Díaz

Tutor: Prof. D. Diego Manuel Martínez Gila

Dpto.: Ingeniería Electrónica y Automática

Cotutor: Prof. D. Antonio Cano Ortega

Noviembre, 2023



Universidad de Jaén
Escuela Politécnica Superior de Jaén
Departamento de Ingeniería Electrónica y Automática

Don DIEGO MANUEL MARTÍNEZ GILA y Don ANTONIO CANO ORTEGA, tutores del Trabajo Fin de Grado titulado: DISEÑO Y PROGRAMACIÓN DE UNA INSTALACIÓN DOMÓTICA EN UNA VIVIENDA UNIFAMILIAR MEDIANTE TECNOLOGÍA KNX, que presenta MARÍA RAMÍREZ DÍAZ, autoriza su presentación para defensa y evaluación en la Escuela Politécnica Superior de Jaén.

Jaén, noviembre de 2023

El alumno:

Los tutores:

RESUMEN

En este Trabajo Fin de Grado se presenta el montaje y la programación de una instalación domótica mediante tecnología KNX en una vivienda unifamiliar.

Con el fin de conseguir un hogar más eficiente y cómodo, se controlan la iluminación y las persianas mediante KNX y se integran softwares gratuitos como HomeAssistant y Arduino para controlar los dispositivos y los sensores, teniendo en cuenta tanto la normativa vigente como otras justificaciones técnicas.

ABSTRACT

This thesis presents the installation and programming of a home automation system using KNX technology in a single-family house.

In order to achieve a more efficient and comfortable home, the lighting and blinds are controlled by KNX and free software such as HomeAssistant and Arduino are integrated to control the devices and sensors, taking into account both the current regulations and other technical justifications.

ÍNDICE GENERAL

1. INTRODUCCIÓN.....	1
1.1. ¿QUÉ SE ENTIENDE POR DOMÓTICA?	1
1.2. NORMATIVA RELACIONADA CON UNA INSTALACIÓN DOMÓTICA.....	3
1.3. ESTADO DE LA TECNOLOGÍA	4
1.4. EL ESTÁNDAR KNX	9
1.4.1. Definición de KNX	9
1.4.2. Ventajas del sistema KNX	10
1.4.3. KNX Association	10
1.4.4. Estructura	10
1.4.5. Funcionamiento	11
1.4.6. Telegramas.....	12
1.4.7. Configuración	14
1.4.8. Topologías.....	14
1.4.9. Medios de transmisión.....	15
1.4.10. Direcciones físicas.....	16
1.5. OBJETIVOS.....	16
1.6. ESTRUCTURA DE LA MEMORIA.....	17
2. METODOLOGÍA	18
2.1. HARDWARE EMPLEADO	18
2.1.1. Componentes KNX.....	18
2.1.2. Arduino	26
2.1.3. Sensor SHT31-D	27
2.2. SOFTWARE EMPLEADO	28
2.2.1. ETS.....	28
2.2.2. Home Assistant	30
2.3. PROTOCOLOS.....	32
2.3.1. MQTT.....	32
2.3.2. I2C	33
3. RESULTADOS.....	34
3.1. CASO DE ESTUDIO 1: PROGRAMACIÓN DE UNA INSTALACIÓN KNX	34
3.1.1. Descripción de la instalación.....	34
3.1.2. Montaje hardware.....	34
3.1.3. Programación software.....	39
3.2. CASO DE ESTUDIO 2: INTEGRACIÓN DE HOME ASSISTANT	40
3.2.1. Descripción de la instalación.....	40
3.2.2. Programación software.....	40
3.3. CASO DE ESTUDIO 2: INTEGRACIÓN DE ARDUINO Y EL SENSOR SHT31-D	43
3.3.1. Descripción de la instalación.....	43
3.3.2. Montaje hardware	43
3.3.3. Programación software.....	44
4. CONCLUSIONES Y TRABAJOS FUTUROS.....	46
5. BIBLIOGRAFÍA.....	47
6. ANEXOS	51
6.1. PROGRAMACIÓN KNX EN ETS5.....	51
6.1.1. Nuevo proyecto	51
6.1.2. Edificio	52

6.1.3.	<i>Dispositivos</i>	52
6.1.4.	<i>Direcciones de grupo</i>	54
6.1.5.	<i>Objetos de comunicación</i>	56
6.1.6.	<i>Dispositivos programados</i>	57
6.1.7.	<i>Esquema de la solución implementada en ETS5</i>	57
6.1.8.	<i>Ejemplos de funcionamiento</i>	58
6.2.	COHESIÓN CON HOME ASSISTANT	60
6.2.1.	<i>Instalación y configuración de la máquina virtual</i>	60
6.2.2.	<i>Acceso a la URL y creación de usuario</i>	61
6.2.3.	<i>Integraciones</i>	61
6.2.4.	<i>Complementos</i>	67
6.2.5.	<i>Instalación de app móvil</i>	67
6.2.6.	<i>Integración de KNX</i>	69
6.2.7.	<i>Integración de Arduino</i>	71
6.2.8.	<i>Configuración de pantalla</i>	74
6.2.9.	<i>Ejemplos de funcionamiento</i>	78
6.3.	COHESIÓN CON ARDUINO	79
6.3.1.	<i>Código de Arduino</i>	79
6.3.2.	<i>Ejemplos de funcionamiento</i>	80
6.4.	ENLACE A FOTOS.....	81
6.5.	ENLACE A VÍDEOS.....	81

ÍNDICE FIGURAS

Figura 1.4.1.1:	Logo KNX. Fuente: KNX Association	9
Figura 1.4.4.1:	Estructura línea KNX.....	11
Figura 1.4.4.2:	Ejemplo estructura línea KNX	11
Figura 1.4.6.1:	Longitud telegrama KNX	13
Figura 1.4.6.2:	Telegrama KNX	13
Figura 1.4.6.3:	EIS.....	14
Figura 1.4.8.1:	Topologías KNX.....	15
Figura 1.4.9.1:	Acoplador.....	16
Figura 2.1.1.1:	Fuentes de alimentación	18
Figura 2.1.1.2:	Conector bus.....	19
Figura 2.1.1.3:	Sensor de movimiento	19
Figura 2.1.1.4:	Actuador binario 4 canales.....	20
Figura 2.1.1.5:	Cable BUS	20
Figura 2.1.1.6:	Actuadores de persianas.....	21
Figura 2.1.1.7:	Actuador de regulación	22
Figura 2.1.1.8:	Actuador multifunción.....	22
Figura 2.1.1.9:	Dimmer	23
Figura 2.1.1.10:	IP Router Wi-Fi	24
Figura 2.1.1.11:	Pulsador 8 teclas.....	24
Figura 2.1.1.12:	Pantalla Flat V4 Zennio	25
Figura 2.1.1.13:	Pantalla Z35 V2 de Zennio.....	25
Figura 2.1.2.1:	Arduino UNO Wi-Fi Rev2	27

Figura 2.1.3.1: Sensor SHT31-D.....	27
Figura 2.2.1.1: Dongle de licencia USB KNX	29
Figura 2.2.2.1: Diferencias entre las versiones de Home Assistant.....	31
Figura 2.3.2.1. Protocolo I2C	33
Figura 3.1.2.1: Cableado KNX. Fuente: KNX Association	35
Figura 3.1.2.2: Montaje	36
Figura 3.1.2.3: Montaje final KNX	37
Figura 3.1.2.4: Montaje	37
Figura 3.1.2.5: Montaje final maletín Zennio	38
Figura 3.1.3.1: Programación.....	39
Figura 3.2.2.1: Archivo de configuración YAML	41
Figura 3.2.2.2: Archivo de configuración YAML	42
Figura 3.2.2.3: Archivo de configuración YAML	42
Figura 3.3.2.1: Placa con sensor	43
Figura 3.3.3.1: Archivo de configuración YAML	45
Figura 3.3.3.2: Interfaz de Home Assistant	45
Figura 6.1.1.1: Nuevo proyecto	51
Figura 6.1.1.2: Seguridad	51
Figura 6.1.2.1: Edificio	52
Figura 6.1.3.1: Catálogo ETS5.....	52
Figura 6.1.3.2: Dispositivos.....	53
Figura 6.1.3.3: Configuración parámetros	53
Figura 6.1.4.1: Direcciones de grupo	54
Figura 6.1.4.2: Direcciones de grupo	54
Figura 6.1.4.3: Direcciones de grupo	55
Figura 6.1.4.4: Direcciones de grupo	55
Figura 6.1.4.5: Topología proyecto	56
Figura 6.1.5.1: Objetos de comunicación	56
Figura 6.1.6.1: Programación KNX en ETS5.....	57
Figura 6.1.7.1: Solución KNX en ETS5	58
Figura 6.1.8.1: Funcionamiento	58
Figura 6.1.8.2: Funcionamiento	59
Figura 6.1.8.3: Funcionamiento	59
Figura 6.2.1.1: Configuración de máquina virtual	60
Figura 6.2.1.2: Máquina virtual.....	60
Figura 6.2.2.1: Creación de cuenta e inicio de sesión	61
Figura 6.2.2.2: Home Assistant.....	61
Figura 6.2.3.1: Integración de inversor.....	61
Figura 6.2.3.2: Integración de inversor.....	62
Figura 6.2.3.3: Integración de inversor.....	62
Figura 6.2.3.4: Integración de inversor.....	62
Figura 6.2.3.5: Integración de inversor.....	63
Figura 6.2.3.6: Integración de inversor.....	63
Figura 6.2.3.7: Integración de inversor.....	64
Figura 6.2.3.8: Integración de inversor.....	64
Figura 6.2.3.9: Integración de inversor.....	64
Figura 6.2.3.10: Integración de inversor.....	64
Figura 6.2.3.11: Integración de inversor.....	65

Figura 6.2.3.12: Integración de cargador	65
Figura 6.2.3.13: Integración de cargador	66
Figura 6.2.3.14: Integración de cargador	66
Figura 6.2.4.1: Tienda de complementos	67
Figura 6.2.5.1: Instalación app móvil.....	67
Figura 6.2.5.2: Instalación app móvil.....	68
Figura 6.2.5.3: Instalación app móvil.....	68
Figura 6.2.6.1: Cohesión KNX	69
Figura 6.2.6.2: Cohesión KNX	69
Figura 6.2.6.3: Cohesión KNX. Telegramas	70
Figura 6.2.6.4: Archivo de configuración YAML	70
Figura 6.2.6.5: Archivo de configuración YAML	71
Figura 6.2.6.6: Archivo de configuración YAML	71
Figura 6.2.7.1: Tienda de complementos	71
Figura 6.2.7.2: Mosquitto broker	72
Figura 6.2.7.3: Integración MQTT	72
Figura 6.2.7.4: Archivo de configuración YAML	73
Figura 6.2.7.5: Interfaz de Home Assistant	73
Figura 6.2.8.1: Interfaz de Home Assistant	74
Figura 6.2.8.2: Interfaz de Home Assistant	74
Figura 6.2.8.3: Interfaz de Home Assistant	75
Figura 6.2.8.4: Interfaz de Home Assistant	75
Figura 6.2.8.5: Interfaz de Home Assistant	76
Figura 6.2.8.6: Interfaz de Home Assistant	76
Figura 6.2.8.7: Interfaz de Home Assistant	77
Figura 6.2.8.8: Interfaz de Home Assistant	77
Figura 6.2.8.9: App móvil de Home Assistant	78
Figura 6.2.9.1: Funcionamiento	78
Figura 6.3.1.1: Código Arduino	79
Figura 6.3.1.2: Código Arduino	79
Figura 6.3.1.3: Código Arduino	80
Figura 6.3.1.4: Código Arduino	80
Figura 6.3.2.1: Arduino a través de MQTT	80
Figura 6.3.2.2: Arduino a través de MQTT	81
Figura 6.3.2.1: Código QR fotos	81
Figura 6.3.2.1: Código QR vídeos.....	81

1. INTRODUCCIÓN

1.1. ¿Qué se entiende por domótica?

Se define como instalación domótica al conjunto de sistemas y elementos que permiten controlar y automatizar de forma inteligente una vivienda, con el fin de conseguir mayor seguridad y confort y gestionar la energía de forma eficiente, estableciéndose así un enlace entre el sistema y el usuario. Se clasifica en función de si utilizan la instalación eléctrica, cables específicos o la red para la transmisión de la señal, y sus combinaciones.

El sistema domótico recoge información del entorno a través de sensores (entradas), la analiza y, mediante una línea de bus, emite órdenes a los actuadores (salidas) del sistema.

Una de las ventajas de la domótica es que se puede personalizar el diseño de las instalaciones en función de las necesidades del usuario, llegando incluso a modificarlo si fuese necesario en un futuro e interconectarlo con otras redes existentes, por lo que se puede afirmar que este sistema es completamente flexible.

En las últimas décadas, el sector domótico ha ido evolucionando y, gracias a los avances alcanzados, los componentes son cada vez más variados, económicos y de calidad, con una instalación y manipulación más accesible para todos los usuarios.

Invertir en domótica es una decisión acertada porque la implantación de este sistema conlleva poder amortizar la instalación en un período corto de tiempo, tener un gasto menor y a tiempo de mantenimiento de instalaciones y una rentabilidad mayor de la vivienda a largo plazo, sin olvidar la reducción de gases de efecto invernadero a la atmósfera.

En conclusión, la domótica contribuye a mejorar el día a día de las personas en el ámbito doméstico y a cuidar el medio ambiente. [1]

La domótica ayuda a mejorar la calidad de vida del usuario en el hogar gracias a las aplicaciones detalladas a continuación: [2]

Confort

El sistema regulará, tanto de forma manual como de forma automática y según unos criterios establecidos por el usuario, las salidas, ya sea apagando o encendiendo luminarias o subiendo o bajando persianas de las distintas estancias de la vivienda, para obtener la iluminación deseada. Lo mismo ocurre con el sistema de climatización, obteniéndose la temperatura deseada gestionando el funcionamiento de radiadores o aires acondicionados de forma conjunta o independiente. El confort no es solo activar salidas; es poder hacerlo en cualquier momento y lugar.

Seguridad

Se basa en dos pilares: prevención y detección. Es posible gestionar el sistema de seguridad del hogar gracias a los distintos sensores que activan las alarmas de intrusión, contra incendios, gas, agua, etc., informando al usuario mediante avisos acústicos o electrónicos, e incluso cortando el suministro de manera automática e instantánea. Además, si el usuario lo desea, es posible apagar las luminarias, bajar las persianas y cerrar las cerraduras de todas las estancias ante indicio de robo.

Comunicación

A través del teléfono móvil, tablet, PC o cualquier dispositivo que disponga de conexión inalámbrica es posible controlar, gestionar y recibir información sobre la instalación domótica de la vivienda desde cualquier lugar.

Accesibilidad

La domótica tiene como una de sus finalidades facilitar la vida doméstica de cualquier persona en cualquier entorno, ajustando el sistema a sus necesidades.

Gestión energética

Otra aplicación de la domótica es la gestión inteligente del uso de la energía en la vivienda, esto es, del encendido y apagado de la iluminación, la conexión y desconexión del sistema de climatización, la conexión y desconexión de electrodomésticos, el calentamiento del agua necesaria, etc., permitiendo así reducir gastos energéticos innecesarios y conseguir un ahorro energético en el consumo del hogar.

1.2. Normativa relacionada con una instalación domótica

Una instalación domótica debe cumplir la siguiente normativa:

- Real Decreto 614/2001: Disposiciones mínimas para la protección y seguridad de los trabajadores en instalaciones eléctricas.
- Reglamento Electrotécnico de Baja Tensión 2002 e instrucciones técnicas complementarias y sus modificaciones.
- Real Decreto 314/2006, de 17 de marzo, por el que se aprueba el Código Técnico de la Edificación.
- Ley 16/2012: Legislación de evaluación ambiental nacional.
- Real Decreto 186/2016, de 6 de mayo, por el que se regula la compatibilidad electromagnética de los equipos eléctricos y electrónicos.
- Real Decreto 346/2011, de 11 de marzo, por el que se aprueba el Reglamento regulador de las infraestructuras comunes de telecomunicaciones para el acceso a los servicios de telecomunicación en el interior de las edificaciones.
- Real Decreto 235/2013, de 5 de abril, por el que se aprueba el procedimiento básico para la certificación de la eficiencia energética de los edificios.
- UNE-EN 61000-6-3: Compatibilidad electromagnética (CEM). Parte 6-3: Normas genéricas. Norma de emisión en entornos residenciales, comerciales y de industria ligera.
- UNE-HD 60364-7-712: Instalaciones eléctricas de baja tensión.
- UNE-HD 60364-5-52: Instalaciones eléctricas de baja tensión.

- Especificación AENOR AE 0026:2006: Instalaciones de sistemas domóticos en viviendas.
- UNE-EN 50491 para Sistemas Electrónicos de Viviendas y Edificios (HBES) y Sistemas de Automatización y Control de Edificios (BACS).
- UNE-EN 50090, donde se describen los requisitos de los sistemas HBES (“Home and Building Electronic Systems”).
- AENOR UNE-CLC/TR 50491-6-3:2013 IN: Instalaciones HBES. Evaluación y definición de niveles.

1.3. Estado de la tecnología

Desde el comienzo de la domótica hubo una necesidad de establecer estándares de fabricación.

Se define estándar cerrado como el protocolo específico de una marca, de forma que solo el fabricante puede modificarlo y fabricar dispositivos que sigan la misma norma.

Se define estándar abierto como el protocolo definido entre varias marcas con el fin de unificar criterios, sin patentes. Cualquier fabricante puede desarrollar dispositivos para él.

Aunque los estándares más extendidos en la actualidad son KNX, Lonworks y X10, también se utilizaron otros. Todos ellos se detallan a continuación: [3]

EHS

El estándar EHS (European Home System) se creó en 1984 en Europa, bajo el manto de la Comisión Europea, con el fin de implantar la domótica de forma masiva en el sector residencial. Se basa en una tipología de niveles OSI (Open Standard Interconnection): físico, de enlace de datos, de red y de aplicación. La idea era crear un protocolo abierto que permitiese cubrir las necesidades de automatización de las viviendas europeas con un sistema asequible a la mayoría de la población.

BatiBUS

Se trata de un protocolo abierto que usa la técnica CSMA-CA (Carrier Sense Multiple Access with Collision Avoidance), similar a Ethernet, pero en este caso, si se produce una colisión entre dos telegramas, la orden del dispositivo prioritario continúa transmitiendo, mientras que se corta la señal del otro. Con esta tecnología, todos los dispositivos escuchan lo que ha enviado otro, procesan dicha información y solo actuarán los necesarios para ejecutar la orden.

EIB

EIB es un sistema desarrollado por la Unión Europea con el objetivo de competir con el mercado domótico japonés y norteamericano y crear un estándar europeo. Se basa en la tipología de niveles OSI, al igual que el sistema EHS, y tiene una arquitectura descentralizada.

Para su impulso se creó la EIB Association (EIBA), formada por 113 empresas, cuyas tareas principales son:

- Fijar las directrices técnicas para el sistema y los productos EIB, así como establecer los procedimientos de ensayo y certificación de calidad.
- Distribuir el conocimiento y las experiencias de las empresas que trabajan sobre el EIB.
- Conceder a los productos EIB y a los fabricantes de estos una licencia de marca EIB con la que se pueden distribuir los productos.
- Colaborar activamente con otros organismos europeos o internacionales en todas las fases de la normalización y adaptar el sistema EIB a las normas vigentes.

KNX

KNX surge de la demanda de crear un único estándar europeo flexible, fácil de instalar y de bajo consumo para el control de edificios inteligentes bajo la iniciativa de tres asociaciones europeas:

- EIBA (European Installation Bus Association).
- BatiBUS Club International.
- EHSA (European Home System Association).

Se trata de, partiendo de los sistemas EIB, EHS y BatiBUS, crear un único estándar europeo para la domótica e inmótica que cubra todas las necesidades y requisitos de las instalaciones profesionales y residenciales que sea capaz de competir en calidad, prestaciones y precios con otros sistemas norteamericanos como LonWorks o CEBus.

Con KNX las funciones operativas y todos los procedimientos pueden ser controlados, monitorizados y alimentados a través de un único bus, con lo que se reduce cableado.

KNX permite reparametrizar los dispositivos de bus sin necesidad de añadir un cable.

X10

X10 es uno de los protocolos más antiguos de aplicación domótica. Fue diseñado en Escocia entre los años 1976 y 1978 con el objetivo de transmitir datos por las líneas de baja tensión a muy baja velocidad (60 bps en EEUU y 50 bps en Europa) y muy bajo coste. Al usar las líneas eléctricas de la vivienda, no es necesario tender nuevos cables para conectar dispositivos. Este protocolo es abierto, aunque el fabricante está obligado a usar los circuitos del fabricante escocés que diseñó esta tecnología, siendo asequible para una instalación no muy compleja.

LonWorks

Se trata de un protocolo abierto inventado para cubrir los requisitos de las aplicaciones de control, homologado por Europa, EE. UU. y China. Actualmente, se ha impuesto como estándar para el control y la comunicación de las gasolineras.

Tiene una estructura descentralizada, extremo a extremo, que permite distribuir la inteligencia entre los sensores y actuadores del edificio.

Los dispositivos de este protocolo están basados en un microcontrolador que permite direccionar cualquier nodo de forma unívoca dentro de una red de control LonWorks, que además funciona enviando telegramas y con cualquier medio de transmisión.

CEBus

En 1984, varios miembros de la EIA norteamericana (Electronics Industry Association) llegaron a la conclusión de la necesidad de un bus domótico que aportara más funciones de las que aportaban sistemas de aquella época. Especificaron y desarrollaron un estándar abierto llamado CEBus (Consumer Electronic Bus), igual que el EHS.

BACnet

El BACnet es un protocolo norteamericano desarrollado bajo el patrocinio de una asociación norteamericana de fabricantes e instaladores de equipos de calefacción y aire acondicionado.

El principal objetivo era el de crear un protocolo abierto que permitiera interconectar los sistemas de aire acondicionado y calefacción de las viviendas y edificios con el único propósito de realizar una gestión energética inteligente de la vivienda.

Este protocolo implementaba la arquitectura OSI de niveles. Algunas propiedades son obligatorias y otras son opcionales, pero la que siempre se debe configurar es la dirección del dispositivo.

DMX

DMX512, abreviado como DMX (Digital MultipleX), es un protocolo electrónico utilizado en luminotecnia para gestión y control de iluminación espectacular permitiendo la comunicación entre los equipos de control de luces y las propias fuentes de luz.

Aparece como solución al problema de la incompatibilidad que existía entre marcas por la utilización de protocolos cerrados, lo cual obligaba a tener un control de manejo por cada marca de luces que se tenía; sin embargo, pronto se convirtió en el protocolo preferido no sólo para controladores de enlace y dimmers, sino también para controlar aparatos de iluminación como scanners, cabezas móviles y dispositivos de efectos especiales como máquinas de humo.

Sistemas IFTTT

En los últimos años, la domótica presentaba baja capacidad de procesamiento de elementos y dependencia de interconexiones cableadas, lo que era un impedimento para realizar sistemas con más de diez componentes.

Actualmente, se ha subsanado el problema y cualquier elemento domótico tiene gran capacidad de cálculo y de comunicación inalámbrica, lo que permite su utilización en cualquier parte de la vivienda.

En una vivienda con conexión a internet se pueden controlar luminarias, tomas de corriente, persianas, cerraduras, equipos de sonido, televisores, sistemas de climatización y sistemas de vigilancia; además, es posible programar acciones en determinadas horas o según estímulos gracias a los sistemas IFTTT (If This, Then That).

A pesar de los grandes avances, todavía no se ha conseguido que todos los elementos estén sincronizados bajo un mismo estándar debido a que no se pueden incorporar todas las formas de conectividad sin utilizar un hardware de alto precio y a que cada elemento dispone de su propio sistema operativo, obligando así a comprar todos los productos de la misma marca.

En un futuro cercano, el objetivo es crear un estándar universal que pueda ser adoptado por todos los fabricantes y que no sea complejo, ya que la mayoría de los usuarios no tienen conocimientos técnicos y no son capaces de montar un sistema de control del hogar completo, sino que simplemente utilizan objetos sencillos como sistemas de iluminación simples o tomas de corriente inteligentes.

Se espera que en las futuras viviendas de nueva construcción se adopten sistemas domóticos de forma masiva, lo que llevará a la reducción de costes y a la creación de nuevas tecnologías en este ámbito. [4]

1.4. El estándar KNX

Debido al tipo de sistema que se quiere conseguir para la vivienda y teniendo en cuenta los estándares libres existentes para estudiantes y universidades, se ha llegado a la conclusión de utilizar el estándar KNX. [5] [6]

Además, KNX permite llevar a cabo cualquier reforma o ampliación del sistema conectando los nuevos dispositivos al bus existente, por lo que, si en un futuro se desea domotizar aún más la vivienda, no habría inconveniente.

1.4.1. Definición de KNX

KNX es un sistema bus descentralizado, es decir, no necesita control central. También, se considera un sistema abierto, independiente de sus fabricantes y sin problema de compatibilidad, que, a través de un lenguaje común, envía datos de control a los componentes de las distintas marcas instaladas en la vivienda.

Se basa en los sistemas EIB, EHS y BatiBUS, consiguiendo así que la comunicación entre los dispositivos sea realizada mediante diferentes buses, como par trenzado, línea de fuerza, Ethernet o radiofrecuencia, donde se conectan los sensores y actuadores (iluminación, persianas, climatización, seguridad, etc.).

Este estándar dispone de la distinción internacional ISO/IEC 14543-3, europea CENELEC EN 50090 y CEN EN 13321-1 y china GB/Z 20965.

Todo lo mencionado anteriormente corrobora que dicho sistema seguirá siendo líder a nivel mundial en la gestión domótica.



Figura 1.4.1.1: Logo KNX. Fuente: KNX Association

1.4.2. Ventajas del sistema KNX

- Ahorro energético: Al diseñar el sistema para encender, apagar o regular la iluminación, las persianas y la temperatura de la vivienda en función de unos criterios, se evita el gasto innecesario de energía.
- Ahorro de tiempo: Gracias a la herramienta de software de KNX (ETS) se realiza el diseño y la configuración al mismo tiempo, estando enlazados para modificarse a la vez.
- Sistema libre: Es posible utilizar componentes de diferentes marcas, lo que garantiza disponer de una amplia gama de productos.
- Facilidad para mejoras futuras: No hay problema en modificar o ampliar las funcionalidades de un sistema ya creado.

1.4.3. KNX Association

El sistema KNX está respaldado por la KNX Association, una agrupación de empresas líderes asociadas al control de viviendas y edificios, cuyo objetivo es establecer KNX como estándar europeo para aplicaciones de domótica e inmótica.

Actualmente más de 370 empresas son miembros de la KNX Association, representando el 80% de todos los componentes de control y gestión de viviendas y edificios, con el objetivo de promover el desarrollo de los sistemas de control y gestión de edificios y viviendas.

Más de 44000 compañías de 128 países y más de 100 universidades con más de 300 centros de formación disponen de acuerdos con KNX Association.

1.4.4. Estructura

El sistema está formado por áreas, líneas y segmentos de líneas.

Si en el sistema hay más de un área, estas se conectan en una línea de áreas (backbone) por medio de acopladores de área, siendo 15 el número de áreas máximas permitidas.

El área está formada por un máximo de 15 líneas conectadas a una principal mediante un acoplador de área y una fuente de alimentación que suministra energía al acoplador.

La línea está formada por un máximo de 4 segmentos de línea utilizando para ello los amplificadores de línea. En la línea se conectan los dispositivos. Cada línea debe disponer de una fuente de alimentación. La longitud máxima del bus en una línea es de 1000 metros. Es posible conectar 2 fuentes de alimentación en 200 metros de forma totalmente distribuida y homogénea. La distancia máxima entre una fuente de alimentación y un dispositivo es de 350 metros, y entre dos componentes de 700 metros.

El segmento de línea es la unidad más pequeña de la estructura. Está compuesto por un máximo de 64 componentes bus.

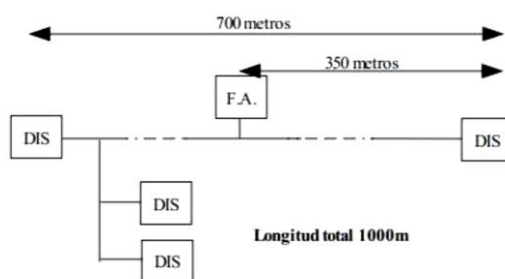


Figura 1.4.4.1: Estructura línea KNX

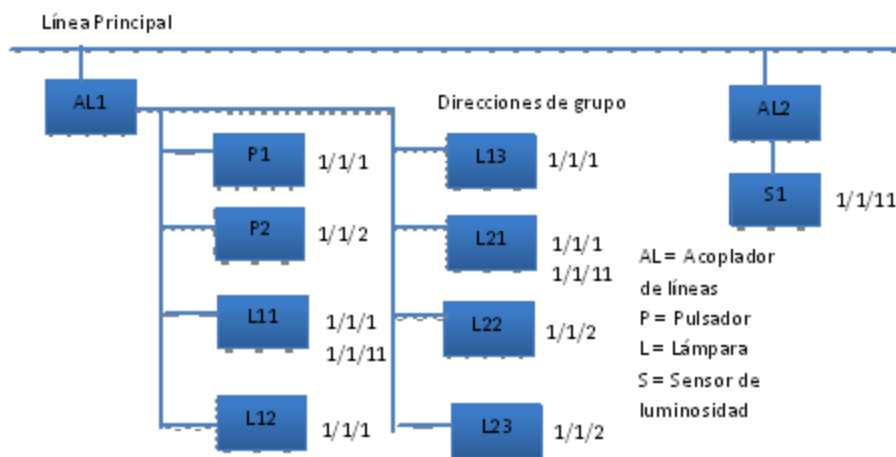


Figura 1.4.4.2: Ejemplo estructura línea KNX

1.4.5. Funcionamiento

Los sensores instalados en la vivienda detectan cambios en la luminosidad, temperatura, pulsadores, etc. Estos sensores envían, a través del cable bus donde están conectados los dispositivos, telegramas con la información empaquetada a todos los dispositivos, pero solo los actuadores a los que va dirigido el mensaje son

los que llevan a cabo las órdenes. Este traspaso de información solo es posible si el bus está libre, es decir, si ningún otro dispositivo está emitiendo telegramas, ya que sino solo tendrá acceso al bus el componente que sea más prioritario.

Cada actuador envía una señal confirmando que ha recibido el mensaje de forma satisfactoria. En caso de no serlo, el sensor repite el telegrama hasta tres veces. Si se agotan las tres veces sin respuesta satisfactoria, el mensaje queda anulado y el sistema notifica un error en el elemento receptor.

Los dispositivos de un mismo bus intercambian la información entre ellos, de ahí la importancia de diseñar una buena estructura.

1.4.6. Telegramas

En un bus se distinguen dos estados: estado lógico “1”, que significa que no circula corriente en el circuito; y el estado lógico “0”, que significa que sí circula corriente en el circuito.

Un telegrama es el estado que se produce cuando hay un cambio en el sistema y el componente envía dicha información al bus, como cuando un sensor detecta presencia.

En primer lugar, se espera a que el bus esté libre porque puede ser que en ese momento se esté enviando otro telegrama.

A continuación, se envía el telegrama y se espera un tiempo en el que el componente manda toda la información.

El tiempo total del telegrama es de 20 a 40 milisegundos, con una velocidad de 9600 bits/s.

El telegrama está formado por paquetes de caracteres formados por 11 bits con 2 bits de pausa distribuidos, cuya longitud está comprendida entre 8 y 23. Se configura de esta forma:

- Byte de control: indica la prioridad del telegrama según el tipo de mensaje mediante 8 bits.

- Byte de origen: indica la dirección física del dispositivo que envía el mensaje utilizando 4 bits para el área, 4 bits para la línea y 8 bits para el número.
- Byte de destino: el último bit de los 17 indica si es una dirección física si tiene valor “0” o si es una dirección de grupo si tiene valor “1”.
- Byte de contador: cuenta el número de saltos que ha dado el mensaje en su ruta.
- Byte de longitud: indica cuántos bytes tiene el campo de datos. Si su valor es “0” significa que tiene 1 bite, y si su valor es “15” significa que tiene 16 bytes.
- Byte de datos útiles: se configura de acuerdo con el EIB Interworking Standard (EIS).
- Byte de seguridad: se utiliza para comprobar que el mensaje ha sido recibido correctamente.

Todo esto se muestra de forma abreviada en las siguientes figuras:

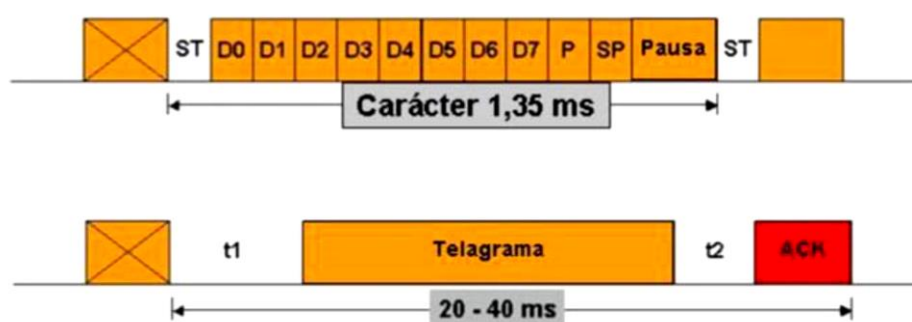


Figura 1.4.6.1: Longitud telegrama KNX

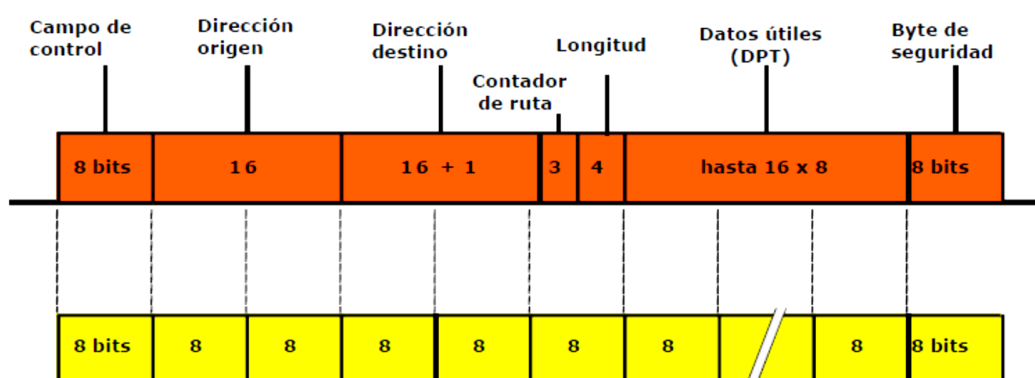


Figura 1.4.6.2: Telegrama KNX

EIS (EIB Interworking Standar).			
NºEIS	Función EIB	Nº bit	Descripción
EIS 1	Interruptor	1 bit	Encendido/apagado, habilitar/deshabilitar, alarma/no alarma, verdadero/falso.
EIS 2	Regulación	4 bit	Se puede utilizar de 3 formas, como interruptor, como valor relativo y como valor absoluto.
EIS 3	Hora	3 bytes	Día de la semana, hora, minutos y segundos.
EIS 4	Fecha	3 bytes	Día, mes, año (el margen es desde 1990 a 2089).
EIS 5	Valor	2 bytes	Para enviar valores físicos con representación.
EIS 6	Escala	8 bit	Se utiliza para enviar valores relativos con una resolución de 8bit.
EIS 7	Control motores	1 bit	Tiene dos usos: mover, arriba/abajo o extender/retraer y paso a paso.
EIS 8	Prioridad	1 bit	Se utiliza en conjunción con EIS 1 ó EIS 7.
EIS 9	Coma flotante	4 bytes	Codifica un valor en coma flotante según el formato definido por el IEEE 754
EIS 10	Contador 16bit	2 bytes	Representa valores de un contador de 16bit.
EIS 11	Contador 32bit	4 bytes	Representa valores de un contador de 32bit.
EIS 12	Acceso	4 bytes	Se usa para conceder accesos a distintas funciones.
EIS 13	Carácter ASCII	8 bit	Codifica según el formato ASCII
EIS 14	Contador 8bit	8 bit	Representa los valores de un contador de 8bit.
EIS 15	Cadena	14 bytes	Transmite una cadena de caracteres ASCII de hasta 14 bytes.

Figura 1.4.6.3: EIS

1.4.7. Configuración

Método E-Mode: pensado para un instalador cualificado con conocimientos básicos sobre la tecnología bus, debido a que la configuración se realiza mediante un controlador central o con microinterruptores. Los dispositivos son programados en fábrica, a falta de detalles. Similar a X-10.

Método A-Mode: pensado para el usuario final, ya que la configuración se realiza automáticamente al conectar los dispositivos.

Método S-Mode: pensado para proyectistas e instaladores KNX certificados porque la configuración se lleva a cabo mediante el software ETS. Los dispositivos solo pueden ser comprados a través de distribuidos especializados.

1.4.8. Topologías

En los sistemas KNX existen diferentes estructuras de comunicación y conexión para los diferentes componentes que lo integran. Pueden ser de tres tipos: línea, estrella y árbol. A su vez, cada topología se divide en tres niveles en los que se designan el número de área, el número de línea y el número de componente.

Topología en línea: los componentes del sistema se conectan a un mismo canal o línea principal, compartiendo el mismo bus para comunicarse entre ellos. Se suele utilizar en sistemas domóticos descentralizados.

Topología en estrella: los componentes del sistema se conectan por medio de un nodo central. Es el más utilizado para sistemas domóticos descentralizados.

Topología en árbol: es una variante de la topología en estrella con nodos secundarios.



Figura 1.4.8.1: Topologías KNX

1.4.9. Medios de transmisión

Para transmitir la información entre los componentes existen varios medios de transmisión que se pueden combinar para hacer el sistema más flexible. Esto es lo que permite la combinación del uso de componentes de distintas marcas.

- Par trenzado 1 (TP1): es el más utilizado, con una velocidad de transmisión de 9600 bits/s. Necesita previa instalación de bus. Los dispositivos se conectan al mismo bus.
- Par trenzado 0 (TP0): tiene una velocidad de transmisión de 4800 bits/s. Necesita previa instalación de bus. Los dispositivos se conectan al mismo bus.
- Línea de fuerza (PL110): tiene una velocidad de 1200 bits/s. Sin bus. La red eléctrica de la instalación se utiliza como bus.
- Línea de fuerza (PL132): tiene una velocidad de 2400 bits/s. Sin bus. La red eléctrica de la instalación se utiliza como bus.
- Ethernet: la transmisión se realiza mediante el cable Ethernet. Tiene una velocidad de 10 Mbits/s.
- Radiofrecuencia: funciona a una frecuencia de 868 MHz con una velocidad de 16,384 Kbits/s. Instalación sin cableado.

Para unir distintos medios de transmisión se utilizan los acopladores, en los que en la parte superior se conecta el cable bus y en la parte inferior lo hace el cable Ethernet:



Figura 1.4.9.1: Acoplador

1.4.10. Direcciones físicas

Una dirección física es el “nombre y apellidos” de un componente. Se compone de tres números:

- Primer número: indica el área donde se encuentra el dispositivo. Del 1 al 15. Si es 0 está indicando que el dispositivo se encuentra en la línea de áreas.
- Segundo número: indica la línea donde se encuentra el dispositivo. Del 1 al 15. Si es 0 está indicando que el dispositivo se encuentra en la línea principal.
- Tercer número: indica el número de dispositivo. Del 1 al 255. Si es 0 está indicando que el dispositivo es un acoplador de línea o de área.

1.5. Objetivos

El objetivo general del TFG consiste en desarrollar una instalación domótica de una vivienda unifamiliar basada en el estándar KNX, desde la instalación hardware de los componentes hasta la programación que permita ejecutar diferentes funcionalidades.

Es también objetivo del TFG la integración del estándar KNX con plataformas de código abierto (hardware y software) para extender funcionalidades a un coste más económico.

Los objetivos específicos son:

- Analizar el estado actual de la tecnología domótica.
- Estudiar las características de cada uno de los componentes hardware disponibles en el laboratorio.
- Automatizar funciones del hogar, tales como subir y bajar persianas, encender y apagar luces, regular la intensidad de las luces y conocer la temperatura y humedad.
- Utilizar programas de libre acceso para controlar el sistema domótico de forma económica.
- Gestionar la vivienda de forma remota, utilizando para ello un programa de libre acceso.
- Integrar diferentes tecnologías existentes como son KNX y Arduino.
- Validar el correcto funcionamiento de la instalación.

1.6. Estructura de la memoria

El presente trabajo fin de grado describe el proyecto de domotización de una vivienda, según el protocolo KNX, y su integración con herramientas open-hardware y open-software. Para ello la memoria se ha estructurado de la siguiente forma:

En el punto metodología se describe el hardware empleado, es decir, los componentes KNX utilizados, como la fuente de alimentación, el cable bus, los actuadores de persianas y los pulsadores, entre otros; el software empleado para la programación del sistema, como son ETS y Home Assistant; y los protocolos de comunicación, como es MQTT.

Después, se detallan los resultados obtenidos, donde se puede ver una descripción detallada del montaje y programación de los dispositivos KNX, así como la integración de Arduino en el sistema domótico y el sensor de humedad y temperatura SHT31-D.

Finalmente, se incorpora un punto de conclusiones y trabajos futuros.

2. METODOLOGÍA

2.1. Hardware empleado

En este punto se describen los componentes empleados tanto para la instalación KNX como para la parte open-hardware.

2.1.1. Componentes KNX

Los diferentes componentes, y sus características, utilizados para el montaje de la maqueta de este proyecto, que se encuentran en el material disponible en el laboratorio, se detallan a continuación:

Fuente de alimentación

Representadas en la Figura 2.1.1.1, la fuente de alimentación de Schneider Electric suministra una tensión de 30 V en corriente continua necesaria para el funcionamiento de los componentes bus de una línea del sistema. Su amperaje dependerá del número de componentes que alimente, pudiendo ser de 160, 320 o 640 mA. Mientras que la fuente de alimentación de Zennio suministra una tensión de 29 V y proporciona hasta 320 mA a los dispositivos de su línea. Ambas integran un filtro que separa los telegramas de datos de la fuente de alimentación. Están protegida contra cortocircuitos y cuenta con un límite de tensión y de corriente. Disponen de un botón de RESET y de 2 LEDs que indican su estado: funciona correctamente (verde) y sobrecarga debido a cortocircuito y tensión superior a 30V en el bus (rojo). [7]

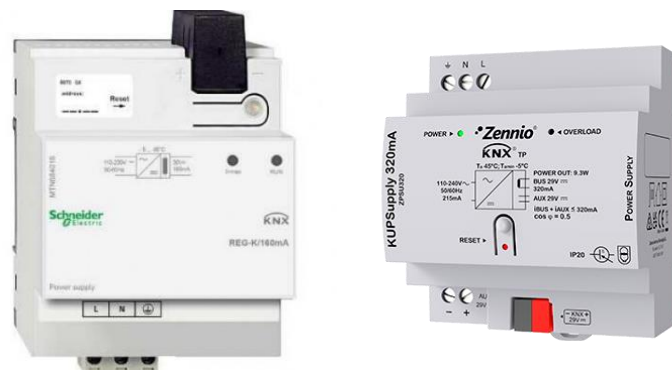


Figura 2.1.1.1: Fuentes de alimentación

Conector bus

Representado en la Figura 2.1.1.2, se trata de un pequeño elemento que ramifica, extiende y protege el bus. En él se realiza el conexionado de los dispositivos al bus. La parte roja es para el polo positivo y la parte negra para el negativo.

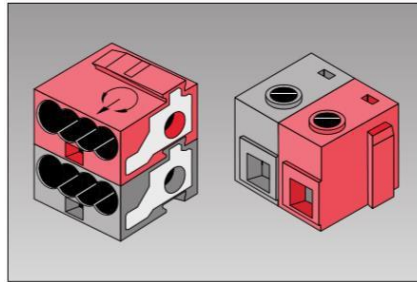


Figura 2.1.1.2: Conector bus

Sensor

Existen tanto sensores de movimiento como sensores de presencia. Ambos envían la información para que las salidas actúen según lo programado.

En el proyecto se ha utilizado el sensor de movimiento de la Figura 2.1.1.3, que dispone de tecnología de detección infrarroja y de un sensor de luminosidad integrado que mejora la funcionalidad del dispositivo. [8]



Figura 2.1.1.3: Sensor de movimiento

Actuador binario

Es un dispositivo que conecta y desconecta las cargas del sistema que pueden tener dos estados posibles: encendido o apagado, actuando como un interruptor inteligente que permite el control y la automatización eficiente de dispositivos eléctricos. Pueden ser mecánicos o electrónicos, con o sin detección de sobretensiones. Se clasifican según su número de canales.

El actuador binario utilizado en este proyecto, mostrado en la Figura 2.1.1.4: Actuador binario 4 canales, es un relé inteligente compatible con el bus comúnmente utilizado para el control de luminarias. Cuenta con cuatro canales independientes (A, B, C, D) de relés conmutables, LEDs que muestran el estado de operación: verde para indicar que el programa se ha cargado correctamente en el dispositivo, amarillo para mostrar el estado de conmutación del relé de los canales y rojo para informar de sobretensión en el bus. Además, incluye la opción de uso manual, con la que se puede activar el interruptor de salida sin alimentación del bus. [9]

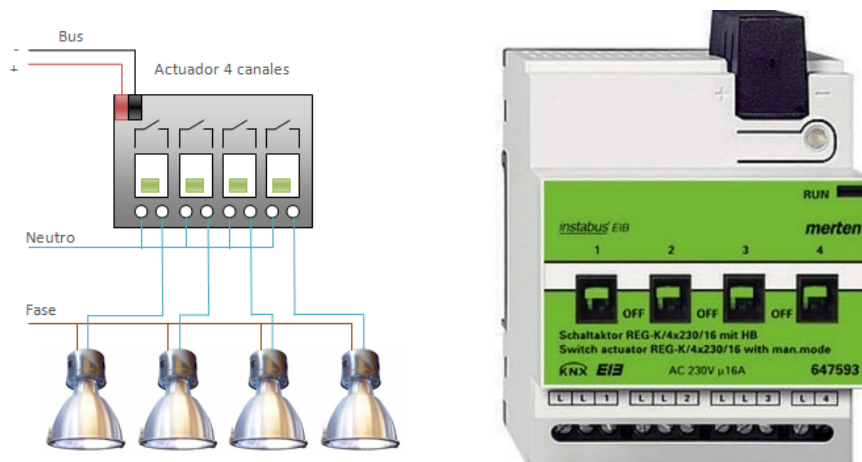


Figura 2.1.1.4: Actuador binario 4 canales

Cable bus

Se trata de un cable especial, que alimenta a los dispositivos y transmite la información en el sistema. Está formado por un cable rojo (positivo), uno negro (negativo), uno amarillo (positivo EIB) y uno blanco (negativo EIB).

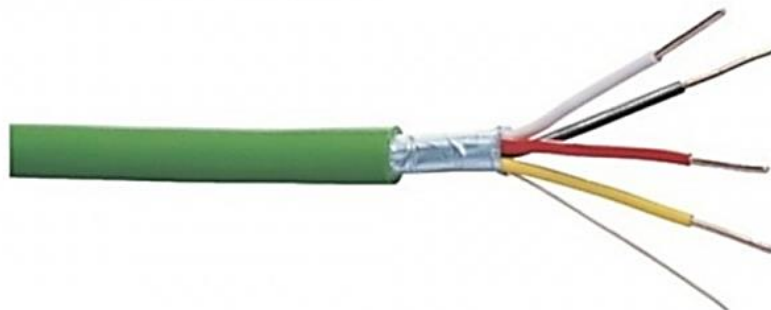


Figura 2.1.1.5: Cable BUS

Actuador de persianas

Es un dispositivo diseñado para controlar el movimiento de persianas, cortinas o toldos en una vivienda. Existen de 2, 4, 8 o 16 canales.

Mediante LEDs indica si su funcionamiento es correcto o si hay algún problema con la conexión al bus o fallo en su programación. Cuenta con un acoplador de bus integrado en el que recibe la alimentación del bus. Además, disponen del modo manual utilizando las teclas de los canales y sin estar programado por ETS.

En este proyecto se han utilizado dos modelos diferentes de actuadores de persianas: el modelo REG-K/4x/10 mit de merten, que puede controlar hasta 4 motores de persianas de hasta 1000 W de forma independiente; y el modelo REG-K/2x/10 con accionamiento manual de Schneider Electric, con 2 canales independientes para motores de hasta 1000 W, mostrados en la Figura 2.1.1.6. [10]



Figura 2.1.1.6: Actuadores de persianas

Actuador de regulación

Es un dispositivo que se utiliza para controlar y ajustar la cantidad de un cierto parámetro en un sistema. Es comúnmente utilizado en viviendas para regular la luminosidad de luminarias y la temperatura según condiciones ambientales.

En este proyecto se ha utilizado el DIMinBOX DX2 de Zennio, mostrado en la Figura 2.1.1.7. Este dispositivo es un actuador dimmer universal de 2 entradas analógico-digitales que pueden ser configuradas como entradas binarias binarias

multifunción, para sensores y pulsadores libres de potencial, como entradas de sondas de temperatura o como entradas de sensores de movimiento. Detecta cortocircuitos, falta de alimentación del bus, sobretensiones, frecuencia anómala y mala configuración en ETS. Además, permite el control manual a través de pulsadores. [11]



Figura 2.1.1.7: Actuador de regulación

Actuador multifunción

Se trata de un dispositivo cuyas salidas tienen configuración múltiple, pudiendo utilizarse para persianas, luminarias o salidas individuales. Se ha utilizado el mostrado en la Figura 2.1.1.8, que dispone de 4 salidas y permite un control manual a través de sus pulsadores. [12]



Figura 2.1.1.8: Actuador multifunción

Dimmer

Es un dispositivo que regula la potencia de energía que recibe una luminaria; sustituye al interruptor convencional que se utiliza de forma general para el encendido y apagado de la iluminación.

Se ha utilizado un controlador LED de 4 canales (RGBW o independientes) de la marca Zennio, mostrado en la Figura 2.1.1.9. Los canales de regulación podrán configurarse como 4 canales independientes, 1 canal RGBW o 1 canal RGB + 1 canal independiente. Incluye 6 entradas analógico-digitales que pueden ser configuradas como entradas binarias multifunción, para sensores y pulsadores libres de potencial o como entradas de sensores de movimiento. [13]



Figura 2.1.1.9: Dimmer

IP Router Wi-Fi

Se ha utilizado el Router IP de Carril DIN SpaceLogic KNX de Schneider Electric, mostrado en la Figura 2.1.1.10. Este dispositivo se alimenta a través del bus KNX y permite el reenvío de telegramas entre diferentes líneas a través de una IP, asignada en la configuración de ETS, que actúa como eje central. Se utiliza como interfaz de programación entre el bus KNX y ETS. Funciona conforme a la especificación KNXnet/IP utilizando los servicios básicos, la administración de dispositivos, la tunelización y el enrutamiento. Dispone de varios LEDs para indicar estados: LED IP (multicolor), LED de modo (multicolor) LED KNX (multicolor) y LED de programación (rojo).

La característica destacable de este dispositivo es que es compatible con KNX Security, por lo que puede utilizarse como router seguro y evitar los accesos no autorizados a la función de tunelización. [14]



Figura 2.1.1.10: IP Router Wi-Fi

Pulsador

Es un dispositivo que permite el encendido y apagado manual de las distintas salidas del sistema programadas. Existen modelos de 2, 4 o 8 teclas y de 8 teclas con receptor de infrarrojos para su manejo con mando a distancia. A cada tecla se le asigna una función, como activar o desactivar una luminaria, activar o desactivar el sistema de climatización, activar y desactivar escenas y controlar persianas.

Se han utilizado cuatro pulsadores de 8 teclas de Schneider Electric para controlar la iluminación y persianas de distintas estancias del hogar, como el de la Figura 2.1.1.11. [15]



Figura 2.1.1.11: Pulsador 8 teclas

Pantalla Flat v4

Esta pantalla está formada por cuatro pulsadores capacitivos retroiluminados, con detector de proximidad y totalmente personalizables. Incorpora 2 entradas analógico-digitales que pueden ser configuradas como entrada binaria para sensor o pulsador libre de potencial, como sonda de temperatura o sensor de movimiento.

En este proyecto se ha utilizado la que aparece en la Figura 2.1.1.12 para encender, apagar y regular la luminosidad de una luminaria empotrada en pared y para subir y bajar una persiana. [16]



Figura 2.1.1.12: Pantalla Flat V4 Zennio

Pantalla Z35 v2

Es una pantalla táctil capacitiva con display retroiluminado, sensor de proximidad, sensor de temperatura y de humedad internos y termostatos para dos zonas independientes. Puede contener hasta 56 controles/indicadores distribuidos en hasta 7 páginas con un menú de navegación. Además, incorpora 4 entradas analógico-digitales que pueden ser configuradas como entrada binaria para sensor o pulsador libre de potencial, como sonda de temperatura o sensor de movimiento.

En este proyecto se ha utilizado la que aparece en la Figura 2.1.1.13. [17]



Figura 2.1.1.13: Pantalla Z35 V2 de Zennio

2.1.2. *Arduino*

Arduino es una plataforma basada en hardware y software libre y extensible de código abierto muy utilizada en electrónica, programación y robótica. Permite crear proyectos de manera fácil y económica.

El hardware de Arduino se basa en una placa con un microcontrolador y pines de entrada y salida. La placa es una PCB (“Printed Circuit Board”) con un circuito interno integrado. Existen diferentes versiones de la placa, tanto oficiales como compatibles con Arduino, utilizándose en la mayoría un microcontrolador tipo AVR de la marca ATMEL. Los pines permiten conectar al microcontrolador sensores, como los de temperatura, humedad, ultrasonidos; actuadores, como motores; y otros dispositivos, como LEDs o interruptores. También, incluye puertos de comunicación para su programación y conexión a otros dispositivos, como USB.

El software de Arduino se basa en lenguaje de programación C++. Existe un entorno de desarrollo integrado de Arduino, llamado Arduino IDE, que simplifica la programación de las placas. Esta plataforma es muy sencilla e intuitiva y puede ejecutarse tanto en Windows como en macOS y Linux. Arduino IDE incluye librerías que facilitan el desarrollo de proyectos gracias a los códigos pre-escritos de ejemplo. También dispone de un monitor serie que resulta muy útil para ver el envío y recibimiento de datos entre la placa y el sistema. Los usuarios escriben el código en el área de trabajo de IDE y cargan el programa en la placa a través del USB.

Arduino se utiliza en una amplia gama de aplicaciones, desde proyectos educativos hasta profesionales y prototipos comerciales. Su uso está bastante extendido en sistemas de control doméstico, robots y dispositivos IoT, entre muchos otros. [18]

En este proyecto se ha utilizado la placa Arduino UNO Wi-Fi Rev 2, mostrada en la Figura 2.1.2.1. Esta placa incorpora el microcontrolador ATmega4809 de 8 bits, con una unidad de medición inercial LSM6DS3TR integrada. Puede conectarse a Bluetooth o una red Wi-Fi o actuar como punto de acceso gracias al chip criptográfico ECC608. Trabaja a 5 V. Tiene 14 pines de entrada/salida digital, siendo 5 salidas PWM, 6 entradas analógicas, una conexión USB, un conector de alimentación, un cabezal ICSP y un botón de RESET. [19]

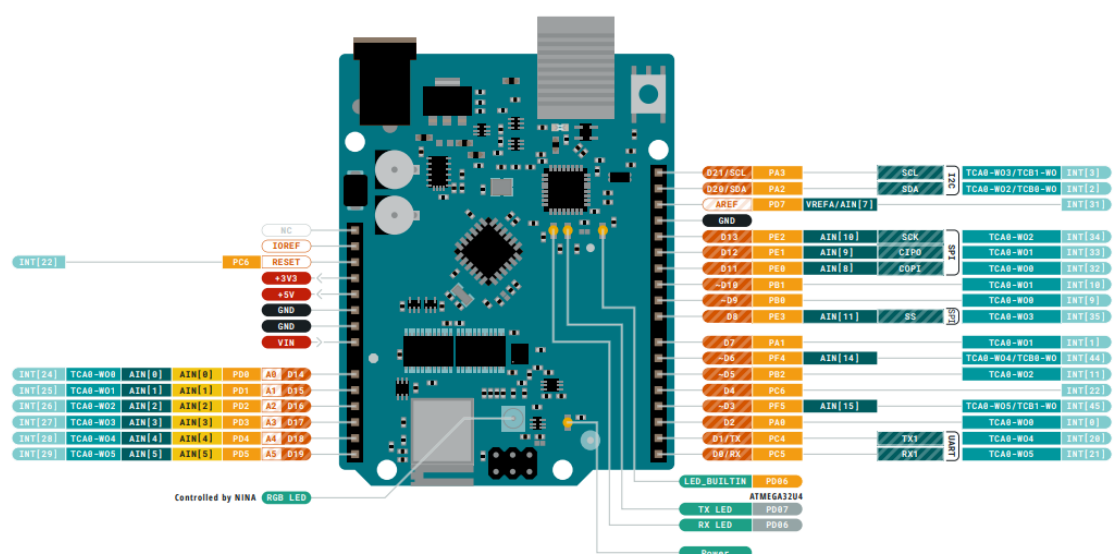


Figura 2.1.2.1: Arduino UNO Wi-Fi Rev2

2.1.3. Sensor SHT31-D

El sensor SHT31-D, mostrado en la Figura 2.1.3.1, es un sensor digital que mide la temperatura y la humedad del ambiente en el que se encuentre. Tiene una precisión de $\pm 0,3$ °C para la temperatura y del $\pm 2\%$ de la humedad relativa. Gracias al filtro de PTFE, se mantendrá limpio y al mismo tiempo permitirá que las mediciones sean tan exactas.

Este sensor se comunica a través del protocolo de comunicación I2C (“Inter-Integrated Circuit”), el cual permite la conexión de varios dispositivos utilizando únicamente dos cables: uno de datos (SDA) y otro de reloj (SCL). Es compatible con 3 V o 5 V, por lo que puede alimentarse y comunicarse con casi cualquier microcontrolador. No es necesario soldarlo a la placa, ya que incluye conectores. Dispone de una librería para su uso en Arduino IDE. [20]

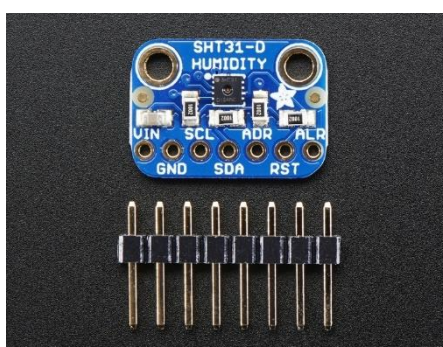


Figura 2.1.3.1: Sensor SHT31-D

2.2. Software empleado

2.2.1. ETS

ETS (Engineering Tool Software) es una herramienta de software inteligente para Windows, desarrollada por la empresa KNX Association, líder en el campo de la automatización, diseñada para la configuración y programación de sistemas de control y gestión de edificios con dispositivos de cualquier fabricante.

Concretamente, ETS proporciona una interfaz centralizada para la configuración sistemas KNX y permite a los usuarios integrar y gestionar una amplia gama de dispositivos, como iluminación, calefacción, ventilación y sistemas de seguridad.

Además, cuenta con una interfaz de programación visual que permite a los usuarios diseñar de sistemas complejos que pueden controlar y supervisar diversos aspectos del entorno y crear escenarios personalizados para adaptar el sistema a sus necesidades específicas.

ETS dispone de herramientas de diagnóstico que facilitan la identificación y resolución de problemas en la red KNX, así como funciones para el mantenimiento y la actualización de los sistemas.

Existen diferentes versiones, que se detallan a continuación:

- ETS Lite: Esta versión de ETS trabaja con un máximo de 20 productos KNX por proyecto. Es adecuada para proyectos pequeños y ofrece a los usuarios una introducción básica al entorno de programación KNX. Tiene disponible una versión demo.
- ETS Professional: Es una versión de pago que ofrece todas las funcionalidades avanzadas del software y puede tener licencia en la nube o con dongle. Es adecuada para proyectos de mayor envergadura y proporciona herramientas más completas para la programación, diagnóstico y mantenimiento de sistemas KNX. Tiene disponible una versión demo.
- ETS Home: Esta versión es de uso privado, para diseñar y programar un edificio con KNX utilizando la interfaz clásica de usuario del software ETS. Solo es posible crear un solo proyecto que contenga un máximo de 64

dispositivos KNX de cualquier fabricante. Está diseñado para instaladores expertos. Es compatible con Windows 10 y 11 y solo puede tener licencia en la nube. Dispone de una versión demo.

- KNX Virtual: Es un software gratuito para probar el funcionamiento de KNX con ETS de manera virtual. No es necesario realizar ninguna inversión en hardware para su uso.

Todas las versiones demo permiten un máximo de 25 dispositivos y un solo proyecto. [21]

Este proyecto, al principio, se diseñó en la versión demo de ETS6, pero dada la envergadura de este, se decidió utilizar una versión con licencia, optándose finalmente por ETS5, ya que el dongle utilizado solo permitía su uso hasta esta versión del software. Por tanto, se instaló ETS5 demo, que se convierte en profesional utilizando una licencia de USB KNX dongle, como la mostrada en la Figura 2.2.1.1. Si el dongle no está conectado al ordenador, el software trabajará en modo demo. [22]



Figura 2.2.1.1: Dongle de licencia USB KNX

2.2.2. *Home Assistant*

Home Assistant es una plataforma de código abierto en la que se puede gestionar un sistema de automatización de forma flexible y segura. Integra una amplia gama de dispositivos y servicios, permitiendo así un control exhaustivo y eficiente del hogar.

Sus características principales se basan en su compatibilidad con una gran variedad de protocolos, dispositivos y estándares de comunicación, su automatización personalizada para crear reglas y escenarios mediante su lenguaje basado en YAML, su flexibilidad y versatilidad para configurar su panel de control, y su seguridad y privacidad con el usuario.

Existen varias versiones, como Home Assistant OS, diseñado específicamente para ejecutarse en miniordenadores o de forma virtualizada; Home Assistant Container, para contenedores; Home Assistant Core, cuya instalación se realiza a través de Python; y Home Assistant Supervised, solo para sistemas Linux. En la Figura 2.2.2.1 se muestran las diferencias entre dichas versiones.

Además, Home Assistant se puede instalar en una variedad de plataformas, como Raspberry Pi, contenedores Docker, Linux y Windows, lo que la hace una plataforma accesible para usuarios con diferentes niveles de experiencia en domótica. Además, dispone de foros para el soporte y mejora constante del software.

El sistema utiliza componentes llamados "integraciones" para admitir diferentes dispositivos y servicios. Además, admite la creación de "automatizaciones", que son reglas o scripts que permiten la ejecución de acciones automáticamente en función de ciertos eventos o condiciones. Esto facilita la creación de escenarios personalizados para adaptarse a las preferencias individuales de los usuarios.

Los usuarios pueden acceder a esta interfaz desde cualquier dispositivo con un navegador web, lo que facilita el control remoto de la automatización del hogar. A través de dicha interfaz, los usuarios pueden conocer la información en tiempo real y controlar los dispositivos conectados al sistema.

Sin embargo, Home Assistant no solo permite controlar los estados de los dispositivos y avisar de ello, sino que, además, obtiene datos de sensores y los muestra en gráficas, actualizando los valores constantemente.

Gracias a su sencillez y a que permite controlar una gran cantidad de dispositivos en una misma red local de forma privada, Home Assistant se ha convertido en la plataforma popular para automatizar hogares. [23]

	OS	Container	Core	Supervised
Automations	✓	✓	✓	✓
Dashboards	✓	✓	✓	✓
Integrations	✓	✓	✓	✓
Blueprints	✓	✓	✓	✓
Uses container	✓	✓	✗	✓
Supervisor	✓	✗	✗	✓
Add-ons	✓	✗	✗	✓
Backups	✓	✓	✓	✓
Managed Restore	✓	✗	✗	✓
Managed OS	✓	✗	✗	✗

Figura 2.2.2.1: Diferencias entre las versiones de Home Assistant

2.3. Protocolos

2.3.1. MQTT

MQTT (Message Queuing Telemetry Transport) es un protocolo de comunicación M2M (Machine-To-Machine) sencillo, ligero y fiable diseñado para redes limitadas, comúnmente utilizado para monitorizar dispositivos en remoto, en la automatización del hogar y en espacios reducidos de alta seguridad.

Algunas características de este protocolo son:

- **Publicación/Suscripción:** Los clientes pueden publicar mensajes y recibir a través su suscripción al tema.
- **Broker:** El componente central en MQTT es el broker, que es responsable de enrutar los mensajes entre los clientes. Los clientes se conectan al broker para intercambiar mensajes.
- **Retención de mensajes:** Los mensajes pueden retenerse en el servidor MQTT y entregarse a los clientes cuando se conecten.
- **Entrega de mensajes:** MQTT ofrece tres niveles de servicio (QoS) para la entrega de mensajes:
 - QoS 0: Entrega al menos una vez, sin confirmación.
 - QoS 1: Entrega al menos una vez, con confirmación.
 - QoS 2: Entrega solo una vez mediante un flujo de diálogo de cuatro etapas.
- **Sistema ligero:** MQTT es adecuado para sistemas con dispositivos limitados.
- **Seguridad:** MQTT es un protocolo de comunicación seguro. Se puede utilizar mecanismos de autenticación como nombre de usuario y contraseña.

En este proyecto, el protocolo MQTT se utiliza para la comunicación entre la placa Arduino UNO Wi-Fi Rev2, que recopila datos a través del sensor SHT31-D, y Home Assistant. Para ello, se utilizan las librerías `WiFiNINA` y `Adafruit\_SHT31` para conectar la placa a la red WiFi y leer los datos del sensor. Una vez realizada la conexión, Arduino envía a Home Assistant dichos datos a través de temas MQTT creados específicamente para ello. Home Assistant es un suscriptor de MQTT, que recibe y escucha los temas y los representa como datos de temperatura y humedad obtenidos por un sensor. [24]

2.3.2. I2C

El protocolo de comunicación “Inter-Integrated Circuit” o “Inter-I2C”, conocido como I2C, presenta las siguientes características:

- Arquitectura tipo maestro-esclavo: El dispositivo que inicia la comunicación es el maestro, mientras que los que responden a sus mensajes son los esclavos.
- Bidireccionalidad: Los datos viajan en ambas direcciones entre el maestro y los esclavos.
- Direcciones únicas: Cada dispositivo dentro del sistema I2C tiene su propia dirección.
- Sincronización: El reloj SCL es proporcionado por el maestro y utilizado para sincronizar la transmisión de datos entre el maestro y los esclavos.
- Start-Stop: Cada transferencia de datos comienza con una señal de inicio y termina con una señal de parada.

En este proyecto, el protocolo I2C se utiliza para comunicarse con el sensor de temperatura y humedad SHT31-D. Se hace en Arduino IDE, implementándolo a través de la biblioteca `Wire.h` y utilizando la dirección 0x44 como dirección del sensor en el sistema I2C. [25]

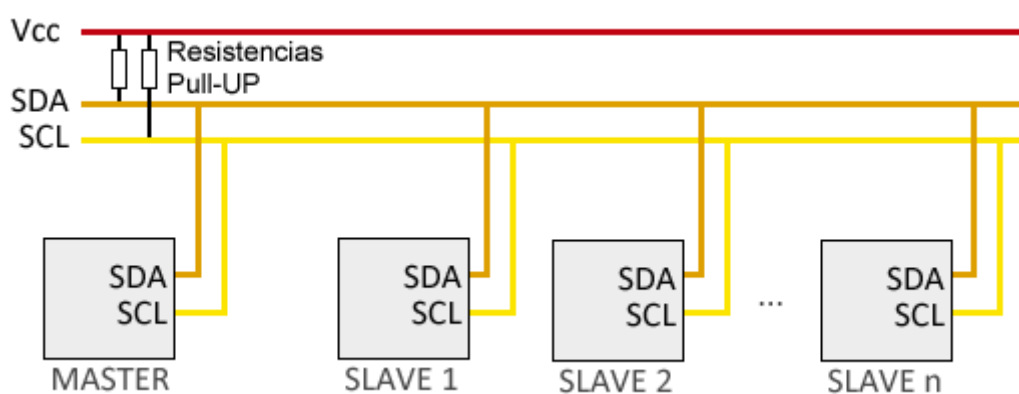


Figura 2.3.2.1. Protocolo I2C

3. RESULTADOS

3.1. Caso de estudio 1: Programación de una instalación KNX

3.1.1. Descripción de la instalación

- Planta baja:
 - Salón: 2 luminarias conmutadas y 2 persianas independientes.
 - Cocina: 2 luminarias conmutadas y 1 persiana.
 - Despacho: 2 luminarias independientes y 1 persiana.
 - Aseo: 2 luminarias independientes y 1 persiana.
- Planta primera:
 - Habitación 1: Luminaria independiente, luminaria con regulador de intensidad y persiana.
 - Habitación 2: Luminaria RGBW.
 - Habitación 3: No configurado por incompatibilidad de material.
 - Baño: Luminaria independiente.
- Sótano:
 - Garaje: No configurado por falta de material. Configurado en Home Assistant.

3.1.2. Montaje hardware

Para instalar los componentes del sistema domótico es necesario incluir un nuevo cableado en las canalizaciones internas de la instalación eléctrica de la vivienda a través del cual se interconectan todos los sensores y actuadores para intercambiar información entre ellos. Concretamente, se va a utilizar el cable NEXANS EIB-Y(ST)Y 2X2X0.8 mm, el llamado cable bus KNX. [26]

Para esta línea KNX TP se utiliza un par de hilos trenzados y apantallados. No se debe conectar la pantalla del cable a masa o tierra en ambos extremos porque su función es simplemente aislar. Debe tenerse especial cuidado en aquellos lugares en los que el cable bus pueda estar en contacto con cables de fuerza, como en cuadros eléctricos o cajas de empalme, para cumplir las distancias mínimas.

De forma general, entre la tensión de bus y la red de 230 V debe existir un doble aislamiento con una tensión de prueba de 4 kV. [27]

Dado que es una línea KNX TP con router KNXnet/IP, es indispensable instalar también cable de red o Ethernet, cuya longitud total no puede superar los 100 metros.

Por último, cabe destacar que, para que por la línea circule la tensión necesaria para permitir el tráfico de datos, es imprescindible instalar una fuente de alimentación KNX que convierta los 230V de la red en 29V en corriente continua.

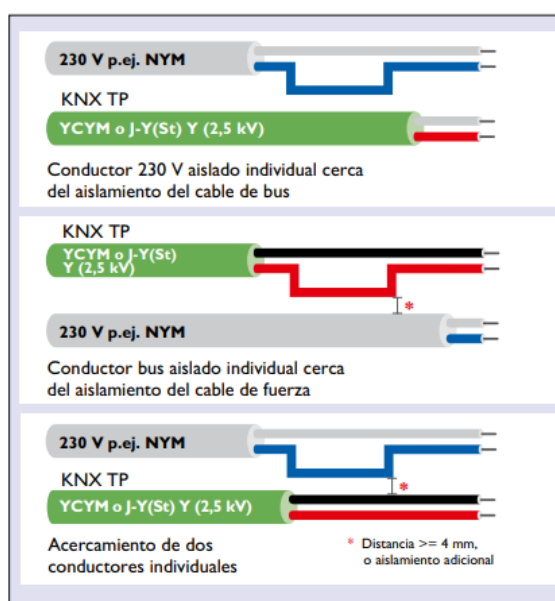


Figura 3.1.2.1: Cableado KNX. Fuente: KNX Association

Para el montaje de la maqueta que simboliza la planta baja de la vivienda, se han utilizado diferentes componentes, explicados anteriormente en el punto 2.1.1.

En primer lugar, se realiza la instalación de la fuente de alimentación del bus en el carril DIN. A continuación, se colocan todas las luminarias LEDs, junto con su fuente de alimentación. Seguidamente, se coloca el actuador binario que va a controlar dichas luminarias. En este paso se realiza la conexión de las luminarias con el actuador binario, obteniéndose el resultado obtenido en la Figura 3.1.2.2. Se incluyen pegatinas para diferenciar la ubicación de las luminarias en la vivienda.

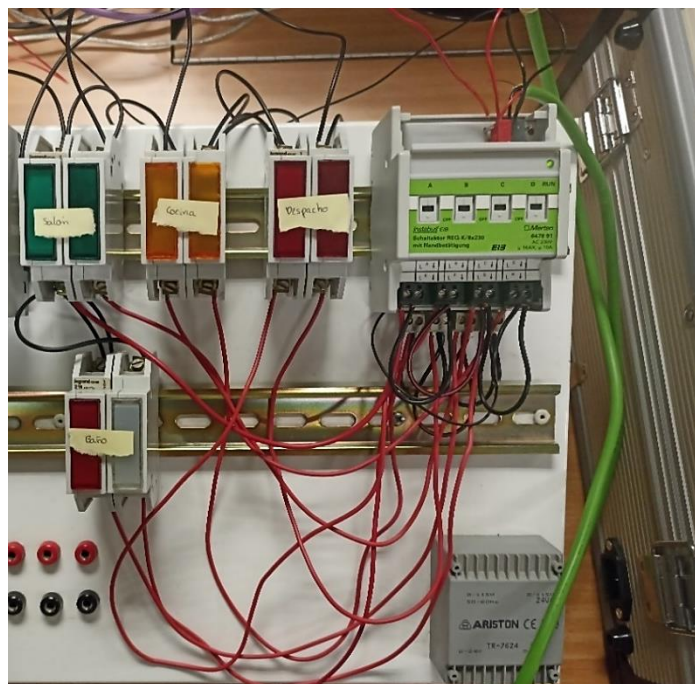


Figura 3.1.2.2: Montaje

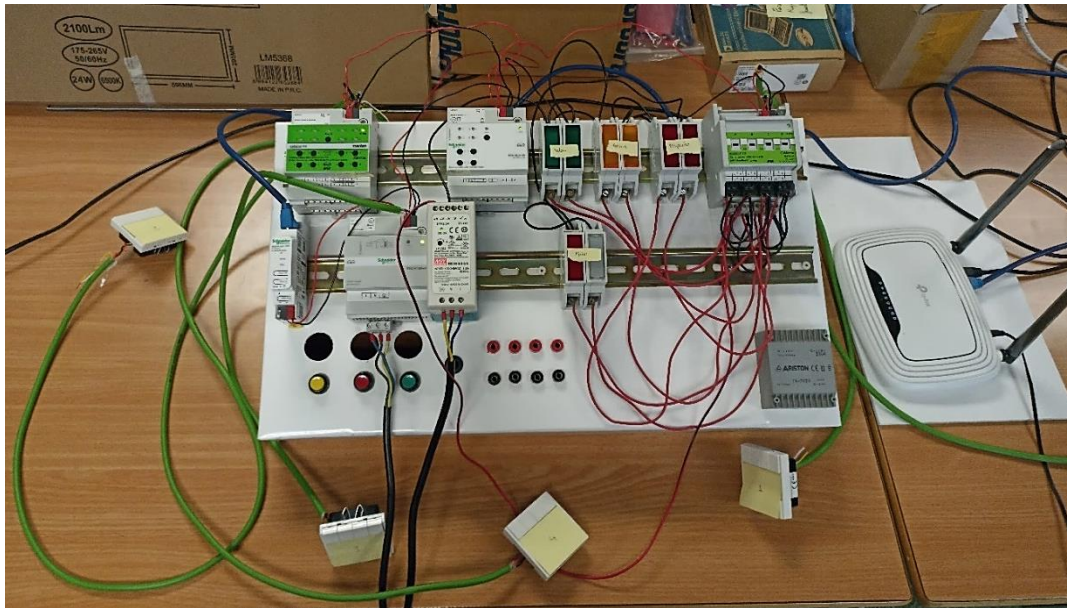
Después, se instalan los actuadores de persianas. En un primer momento solo se instaló el actuador de 4 canales, pero finalmente se optó también por añadir otro de 2 canales de otra marca, para comprobar que es posible utilizar dispositivos de cualquier fabricante.

Una vez instalados estos dispositivos, se añaden los pulsadores a través de los cuales se van a controlar las distintas acciones de esta parte del sistema, ya sea encender o apagar luminarias o subir y bajar persianas. Se incluyen pegatinas para diferenciar su ubicación en la vivienda.

Por último, se instala el router de KNX para conectar el sistema domótico al programa ETS y para que los telegramas se envíen vía Wi-Fi a todos los componentes. Este dispositivo se conecta a la red Wi-Fi de la vivienda mediante cable Ethernet.

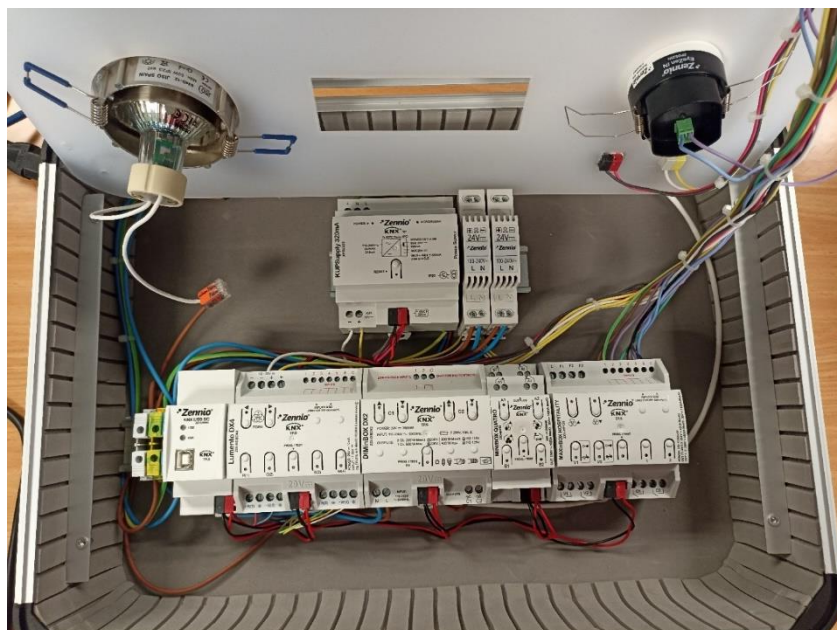
Para finalizar, se interconectan todos los dispositivos con el cable bus, mediante la conexión de los polos positivo (rojo) y negativo (negro).

El resultado final del montaje de esta parte del sistema se muestra en la Figura 3.1.2.3.

**Figura 3.1.2.3: Montaje final KNX**

Para ampliar el proyecto, se decide utilizar también el maletín ZDC23004 cedido por Zennio, cuyos dispositivos vienen incluidos, pero no conectados al bus. Representa la primera planta de la vivienda.

Para realizar la conexión de los dispositivos al bus, al igual que la otra parte de la maqueta, se comienza por la fuente de alimentación que alimenta a esta línea del sistema. A continuación, se conectan internamente los demás dispositivos, como el actuador de regulación, el actuador multifunción y el dimmer. Estas conexiones se muestran en la Figura 3.1.2.4.

**Figura 3.1.2.4: Montaje**

El maletín también dispone de interruptores que se conectan a los diferentes actuadores del sistema. Seguidamente, es turno de las luminarias y el sensor. Una vez conexas esta parte, solo queda incluir las pantallas que se van a utilizar en este proyecto.

El resultado final del montaje de esta parte del sistema se muestra en la Figura 3.1.2.5.

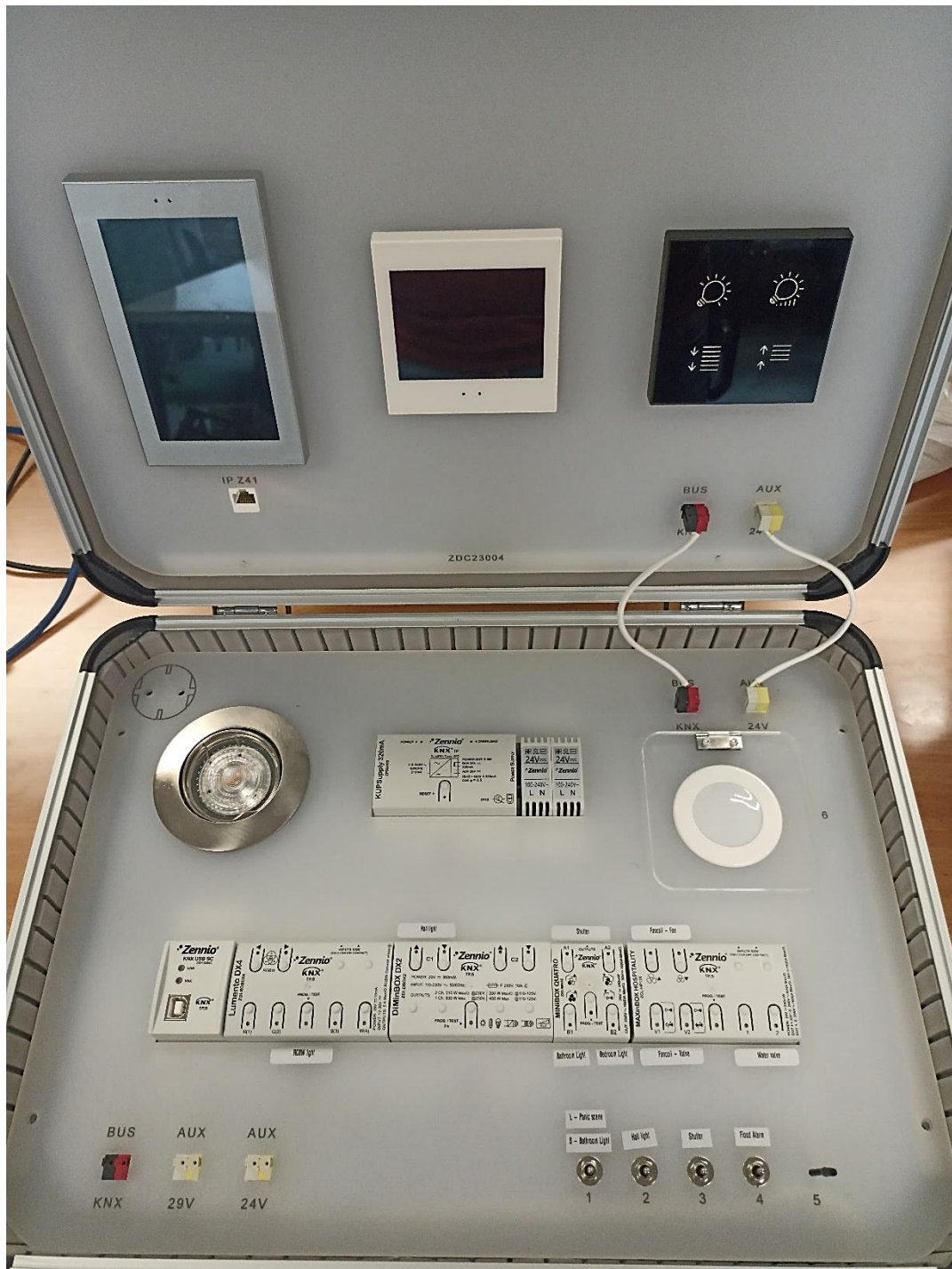


Figura 3.1.2.5: Montaje final maletín Zennio

3.1.3. Programación software

Para llevar a cabo la programación de los dispositivos, que se muestra detalladamente, paso a paso, en el punto 6.1, se lleva a cabo mediante el programa ETS, explicado anteriormente en el apartado 2.2.1.

En primer lugar, una vez abierto el programa, se instala y configura la licencia dongle y se crea un nuevo proyecto, donde se establece un nombre y contraseña. En el apartado de seguridad aparece el certificado que otorga el dispositivo KNX IP.

A continuación, se abre el proyecto y se crea el edificio, es decir, se crean las diferentes estancias que componen la vivienda: planta baja (salón, cocina, despacho, aseo y pasillo), planta primera (habitación 1, habitación 2, habitación 3, baño y pasillo) y garaje. En los pasillos es donde se van a ubicar los actuadores del sistema, mientras que en las estancias lo harán los sensores, luminarias, pulsadores y pantallas.

Después, se añaden los dispositivos a utilizar en cada estancia a través del catálogo, filtrando por fabricante. Se configura cada dispositivo, indicando el número de canales, su función y su dirección física única e individual.

El paso más importante es el de configurar las direcciones de grupo, para tener una topología correcta y que cada actuador esté vinculado y se comuniquen con el dispositivo adecuado. Esto se realiza mediante el apartado de objetos de comunicación en cada dispositivo creado.

Para finalizar, se programan todos los dispositivos con ayuda del software ETS, indicando cuál es cuál físicamente mediante el botón de programación, para lograr la configuración individual de cada uno, como se muestra en la Figura 3.1.3.1.



Figura 3.1.3.1: Programación

3.2. Caso de estudio 2: Integración de Home Assistant

3.2.1. Descripción de la instalación

- Integración de inversor HUAWEI SUN2000-6KTL-L1 (sin dispositivo físico).
- Integración de cargador V2C Trydan monofásico 7,4 kW (sin dispositivo físico).
- Control de dispositivos KNX a través de la app móvil.

3.2.2. Programación software

Para llevar a cabo la cohesión del proyecto KNX de la forma más económica posible con Home Assistant, es necesario hacerlo a través de una máquina virtual y su URL, configurando el software según las necesidades individuales.

Por tanto, siguiendo en todo momento las indicaciones de la página web oficial de Home Assistant para Windows [23], se lleva a cabo la instalación de la máquina virtual de Oracle VM VirtualBox. Una vez configurada, se inicia y se obtiene la pantalla de HomeAssistant, donde se indica la URL en la que se encuentra el software para su uso y modificación. [28]

Una vez se accede a la URL, es necesario crear una cuenta e iniciar sesión en la instancia. Dentro de la interfaz de Home Assistant, es posible configurar la cuenta, crear usuarios, añadir complementos, añadir integraciones, modificar sus archivos de configuración, etc.

Para instalar algunas integraciones se debe hacer a través de 'HACS' (Home Assistant Community Store), una tienda de componentes personalizados, plantillas y tarjetas desarrolladas por la comunidad de usuarios de Home Assistant que permite la instalación y actualización de forma rápida y sencilla de los componentes. [29]

Entre las integraciones objeto de instalación se encuentra el programa HUAWEI inversor monofásico HUAWEI SUN2000-6KTL-L1, cuyo proceso de instalación solicita varios parámetros, como son:

- Host: es la dirección IP del inversor HUAWEI, es decir, la dirección de red del inversor en la red local.
- Port: es el puerto para comunicarse con el inversor HUAWEI. Suele ser el 502.

Se configuran tanto el inversor como el power meter, para conocer los datos de consumo de la vivienda y de producción de la instalación fotovoltaica. Además, es necesario configurar el apartado 'Energía', que se encuentra en la barra lateral, para conocer el consumo real de la vivienda, siguiendo los puntos pedidos paso a paso.

Otra integración que destacar es la del cargador del vehículo eléctrico V2C Trydan, que se realiza a través del complemento 'Open Charge Point Protocol' de 'HACS'. [30]

Sin embargo, KNX es la integración más importante de Home Assistant en este proyecto. Basta con buscarla en la tienda de integraciones y seguir los pasos de su configuración. En este caso, al estar KNX conectado a la misma red que Home Assistant, este lo detecta y configura automáticamente. Después, se adjunta el archivo del programa ETS en el que están programadas las direcciones de grupo para que, en todo momento, se monitoricen los telegramas que se están llevando a cabo en tiempo real.

Al configurar la integración KNX, se detecta que los dispositivos no aparecen automáticamente en el visor principal. Para solucionar este problema, se ha consultado la página oficial de ayuda de Home Assistant [31], optando por programar los dispositivos que se desean controlar a través del archivo de configuración YAML, tal y como se muestra a continuación en Figura 3.2.2.1, Figura 3.2.2.2 y Figura 3.2.2.3:

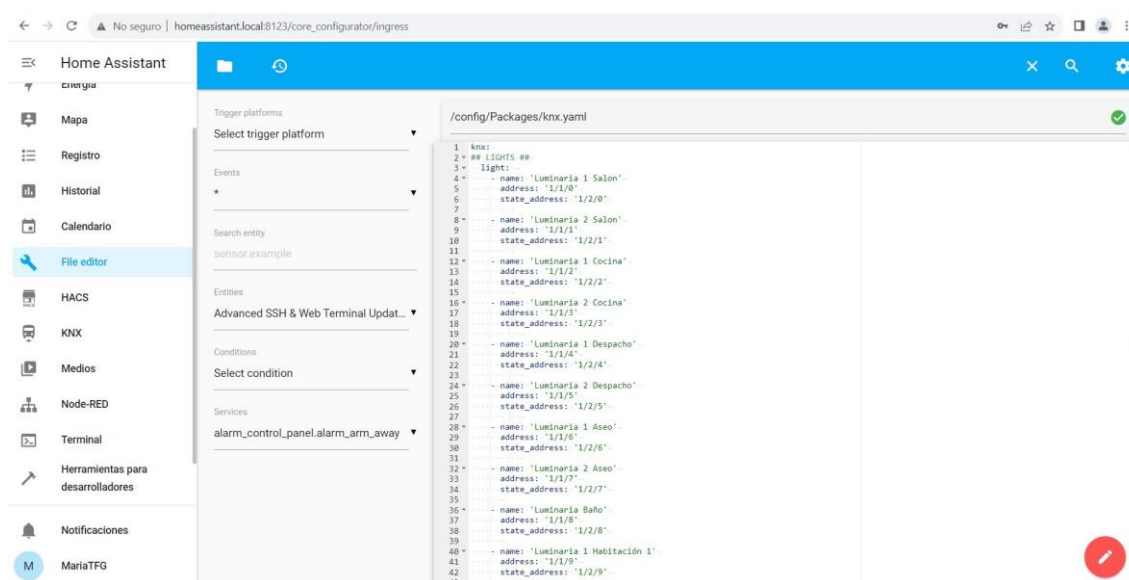


Figura 3.2.2.1: Archivo de configuración YAML

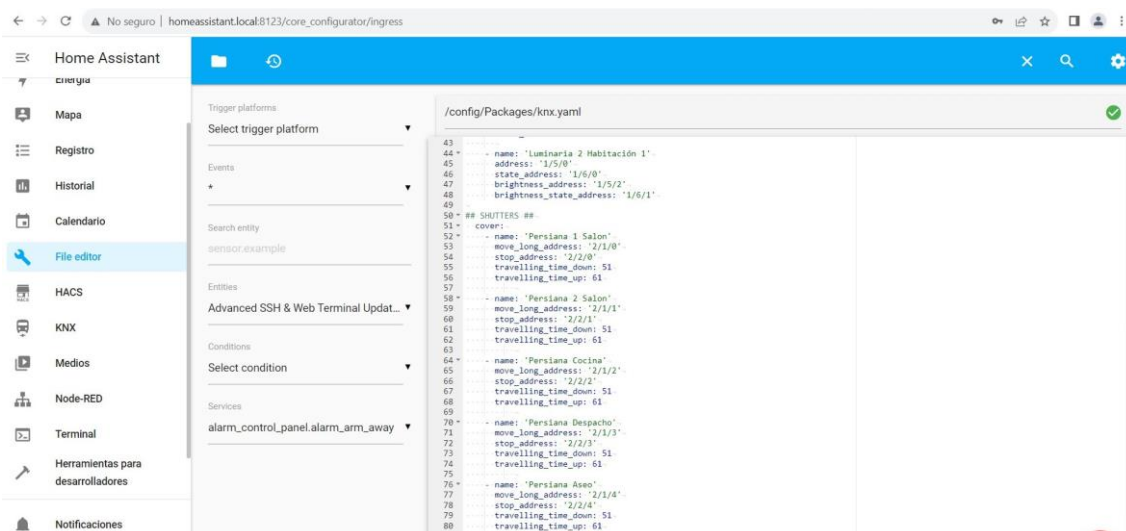


Figura 3.2.2.2: Archivo de configuración YAML

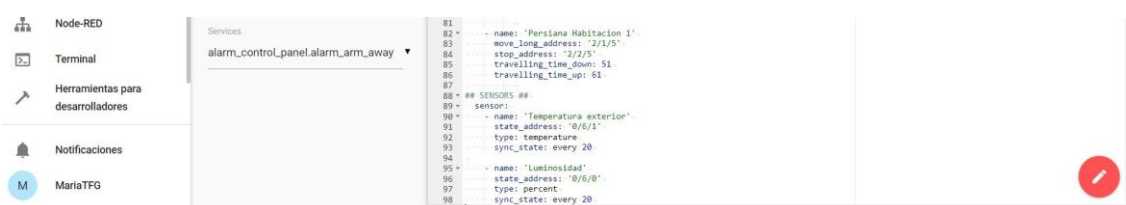


Figura 3.2.2.3: Archivo de configuración YAML

Por último, se añade todo el hardware de KNX a la interfaz de Home Assistant, concretamente en el apartado 'Resumen', donde se crea una página por cada estancia de la vivienda y en la que se incluyen los dispositivos configurados para las mismas.

También es posible la instalación de complementos. Simplemente hay que buscar el deseado lo en la tienda de complementos y pulsar el botón 'instalar', donde después aparecerá 'iniciar' y automáticamente se configurará.

Dado que el objetivo principal de esta cohesión es que se puedan controlar los dispositivos KNX de la instalación domótica de la vivienda a través de la app móvil de Home Assistant, se añade la integración 'aplicación móvil'.

Todo esto se muestra detalladamente, paso a paso, en el punto 6.2.

3.3. Caso de estudio 2: Integración de Arduino y el sensor SHT31-D

3.3.1. Descripción de la instalación

- Placa Arduino UNO Wi-Fi Rev2.
- Sensor de temperatura y humedad SHT31-D.

3.3.2. Montaje hardware

En primer lugar, lo primero es alimentar la placa Arduino, bien conectándola al ordenador mediante USB o bien a una fuente externa. En este proyecto se conecta con USB. Después, se conecta el sensor SHT31-D a la placa (mostrado en la Figura 3.3.2.1) de la siguiente forma:

- Vin del sensor con el pin 5V de la placa.
- GND del sensor con el pin GND de la placa.
- SCL del sensor con el pin SCL de la placa.
- SDA del sensor con el pin SDA de la placa.

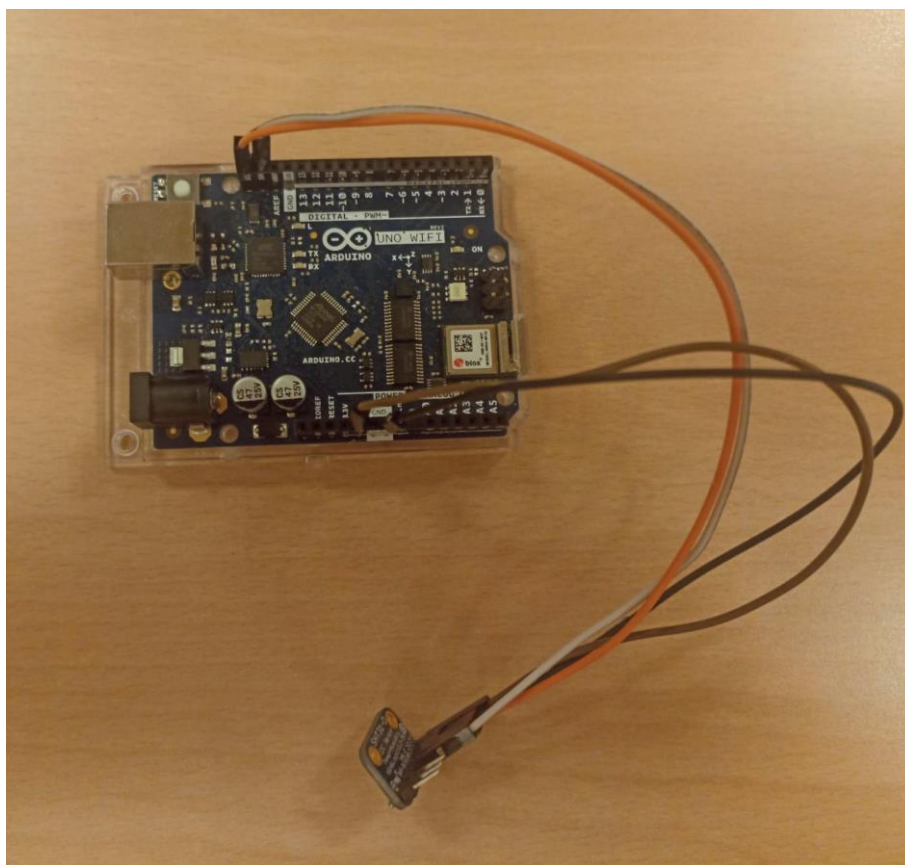


Figura 3.3.2.1: Placa con sensor

3.3.3. Programación software

Para programar el Arduino, se utiliza el software Arduino IDE. Para ello, se utilizan las librerías ‘Wire’ [32], que permite la comunicación a través del protocolo I2C; ‘Adafruit_SHT31’ [33], que permite al microcontrolador actuar con el sensor SHT31-D; ‘WiFiNINA’ [34], que permite la comunicación Wi-Fi en el Arduino Uno WiFi Rev2; y ‘PubSubClient’ [35], que permite que Arduino envíe mensajes a través de MQTT en temas al sistema domótico del hogar.

El procedimiento que sigue el código de Arduino (detallado en el punto 6.3.1) que permite enviar los valores medidos de temperatura y humedad por el sensor al bróker MQTT en Home Assistant a través de una conexión Wi-Fi es el siguiente:

En primer lugar, se incluyen las bibliotecas necesarias para la comunicación por I2C (Wire), el manejo del sensor SHT31-D (Adafruit_SHT31), la conexión Wi-Fi (WiFiNINA) y la comunicación MQTT (PubSubClient).

Seguidamente, se definen las constantes: dirección IP del servidor MQTT, puerto, nombre de usuario, contraseña, nombre del sensor y los temas de temperatura y humedad. También, se especifican el nombre y contraseña de la red Wi-Fi del sistema.

A continuación, se crea un identificador único para el dispositivo y se inicializan objetos para el cliente WiFi, el cliente MQTT y el sensor SHT31-D.

En la función *setup* se inicia la comunicación serial para la depuración, se inicializa el sensor SHT31-D y se verifica si está conectado correctamente, se realiza la conexión a la red Wi-Fi especificada y se configura el cliente MQTT con la dirección y puerto del servidor y la función de devolución de llamada para procesar mensajes recibidos desde el servidor MQTT. En este caso, esta función no se ha configurado porque no se desea recibir mensajes.

En la función *loop* se llama a `mqttClient.loop` para mantener la conexión MQTT activa, se lee la temperatura y la humedad del sensor SHT31-D, se publican los valores en los temas MQTT correspondientes, se imprime el mensaje en la consola serial y se espera 15 segundos antes de realizar otra lectura y envío de datos.

Para realizar la cohesión de Home Assistant con Arduino, se instala el complemento *Mosquitto broker*. Después, se instala la integración *MQTT*, la cual se configura automáticamente gracias a que todo el sistema domótico se comunica a través de la IP del router KNX.

Finalmente, dado que se desea integrar en el sistema el sensor SHT31-D, este se configura en el archivo de configuración YAML de Home Assistant, cuyo código se muestra a continuación en la Figura 3.3.3.1:

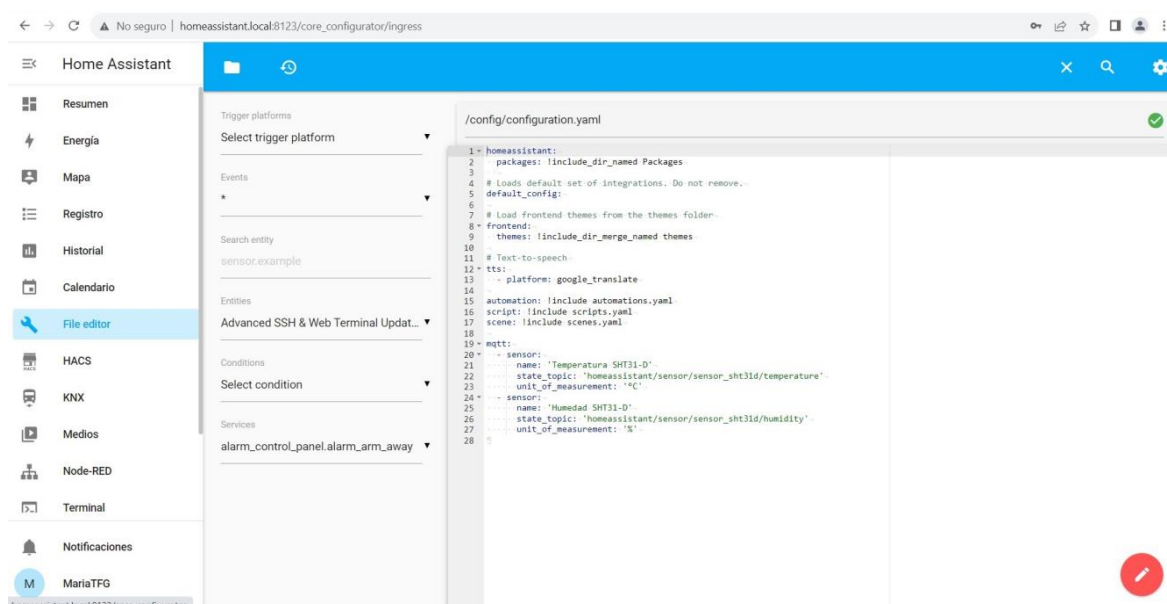


Figura 3.3.3.1: Archivo de configuración YAML

Para finalizar, se añaden los gráficos de temperatura y humedad a la interfaz de Home Assistant, quedando esta como se muestra en la Figura 3.3.3.2:

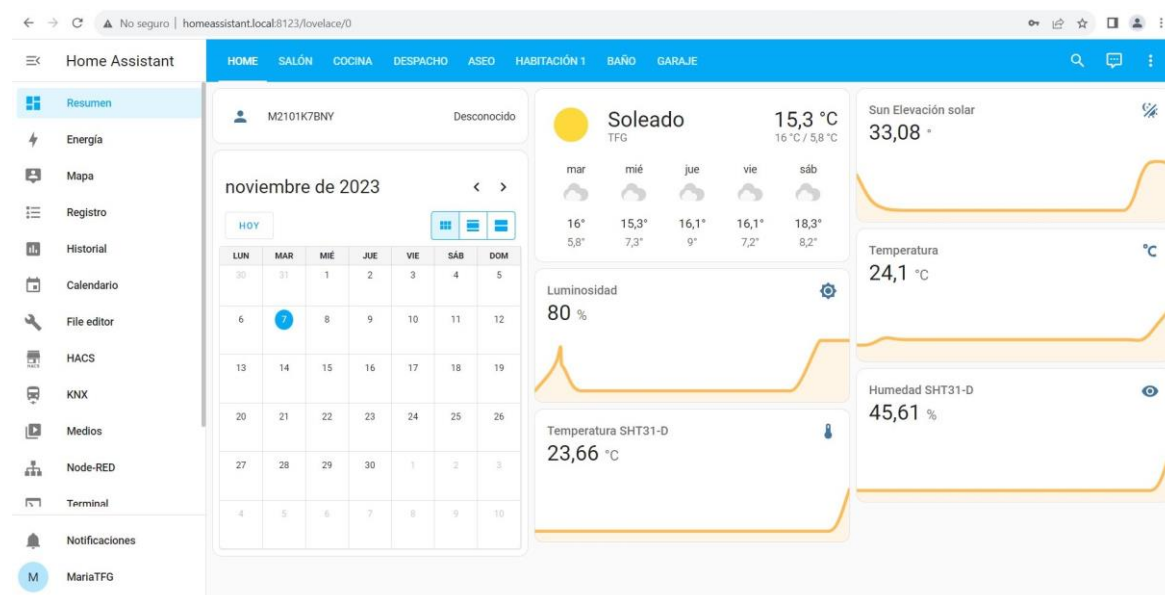


Figura 3.3.3.2: Interfaz de Home Assistant

4. CONCLUSIONES Y TRABAJOS FUTUROS

Este trabajo tiene como finalidad adquirir conocimientos en el campo de la domótica, ya que se trata de un campo con evolución y futuro, que conlleva muchos aspectos como puedan ser el confort, la seguridad y la gestión energética, entre otros.

Una vez probado el funcionamiento de la maqueta realizada, se ha comprobado que esta cumple con los objetivos propuestos porque es posible encender y apagar las luminarias de las diferentes estancias, tanto conmutadas como individuales, subir y bajar persianas, regular la intensidad de iluminación y controlar la luminaria RGBW.

Con Arduino se ha conseguido monitorizar la temperatura y la humedad de una estancia, a través del sensor SHT31-D, mediante el protocolo I2C y la comunicación MQTT para su integración en Home Assistant. Además de estas integraciones, existen una gama de posibilidades para la automatización y monitorización del hogar.

Por último, gracias a la app móvil de Home Assistant, es posible controlar todo el sistema domótico a través de un dispositivo móvil conectado a la misma red Wi-Fi. En el futuro se espera que también sea posible poder realizarlo a distancia.

Aunque con Home Assistant se ha llevado a cabo la integración del inversor HUAWEI SUN2000-6KTL-L1 y el cargador del vehículo eléctrico V2C Trydan monofásico 7,4 kW, no ha sido posible comprobar su funcionamiento ya que no se dispone físicamente de estos dispositivos.

De cara a futuro y a una ampliación de este proyecto, queda pendiente la comprobación del funcionamiento de la integración del inversor y del cargador del vehículo eléctrico y la domotización de los electrodomésticos de la vivienda, así como la implementación de un sistema de escenas y de alarmas con detectores de movimiento para cubrir la seguridad del hogar.

5. BIBLIOGRAFÍA

[1] Sistemas domóticos existentes, tipos y estándares. (s. f.). Domótica Sistemas. https://domoticasistemas.com/tienda/tutoriales/1_sistemas-existentes-tipos-y-estandares.html

[2] Aplicaciones de la domótica (Usos, funciones, utilidades) - IECOR. (2015, 29 diciembre). IECOR. <https://www.iecor.com/aplicaciones-de-la-domotica/>

[3] Estándares Internacionales de Domótica - IECOR. (2016, 2 enero). IECOR. <https://www.iecor.com/estandares-internacionales-de-domotica/>

[4] Webedia Brand Services. (2015). El estado actual de la domótica y las innovaciones que están por venir. Xataka Smart Home. <https://www.xatakahome.com/espacioprosegur/el-estado-actual-de-la-domotica-y-las-innovaciones-que-estan-por-venir>

[5] KNX iniciación. (s. f.). PPT. <https://es.slideshare.net/ert23/knx-iniciacin>

[6] Raul Carretero. (2020). Introduccion a KNX. Raul Carretero. <https://raulcarretero.com/introduccion-a-knx/>

[7] KUPSupply 320MA | Zennio. (2023, 18 enero). Zennio. <https://www.zennio.com/es/producto/kupsupply-320ma>

[8] EyeZen IN | Zennio. (2022, 24 junio). Zennio. <https://www.zennio.com/es/producto/eyezen-in>

[9] MTN647595 - KNX - Actuador binario Carril DIN 4 canales 230 V 16 A con DET corriente | Schneider Electric España. (s. f.). <https://www.se.com/es/es/product/MTN647595/knx-actuador-binario-carril-din-4-canales-230-v-16-a-con-det-corriente/>

[10] MTN649802 - KNX - Actuador de persiana carril DIN 2 canales 230 V 10 A con acc. manual | Schneider Electric España. (s. f.). <https://www.se.com/es/es/product/MTN649802/knx-actuador-de-persiana-carril-din-2-canales-230-v-10-a-con-acc-manual/>

[11] DIMinBOX DX2 | Zennio. (2022, 24 junio). Zennio.
<https://www.zennio.com/es/producto/diminbox-dx2>

[12] MINiBOX QUATRO | Zennio. (2022, 15 diciembre). Zennio.
<https://www.zennio.com/es/producto/minibox-quatro>

[13] Lumento DX4 | Zennio. (2023, 27 enero). Zennio.
<https://www.zennio.com/es/producto/lumento-dx4>

[14] MTN6500-0103 - KNX - SpaceLogic KNX/IP Router Carril DIN KNX Secure | Schneider Electric España. (s. f.). <https://www.se.com/es/es/product/MTN6500-0103/knx-spacelogic-knx-ip-router-carril-din-knx-secure/>

[15] MTN617419 - KNX - Pulsador de 4 elementos plus Marfil - Elegance | Schneider Electric España. (s. f.).
<https://www.se.com/es/es/product/MTN617419/knx-pulsador-de-4-elementos-plus-marfil-elegance/>

[16] Flat 1/2/4/6 v2 | Zennio. (2023, 8 marzo). Zennio.
<https://www.zennio.com/es/producto/flat-v2>

[17] Z35 V2 | Zennio. (2023, 6 septiembre). Zennio.
<https://www.zennio.com/es/producto/z35-v2>

[18] Aguayo, P. (2023, 12 mayo). ¿Qué es Arduino? | Arduino.cl - Compra tu arduino en línea. Arduino.cl - Compra tu Arduino en Línea. <https://arduino.cl/que-es-arduino/>

[19] ARDUINO UNO WiFi REV2. (s. f.). Arduino Official Store.
<https://store.arduino.cc/products/arduino-uno-wifi-rev2>

[20] Industries, A. (s. f.). Adafruit Sensirion SHT31-D.
<https://www.adafruit.com/product/2857>

[21] KNX para profesionales – KNX Association [Official Website]. (s. f.).
<https://www.knx.org/knx-es/para-profesionales/>

- [22] ETS6 licensing – KNX Association
- [23] Assistant, H. (s. f.). Home Assistant. Home Assistant.
<https://www.home-assistant.io/>
- [24] MQTT - the standard for IoT messaging. (s. f.). <https://mqtt.org/>
- [25] Llamas, L. (2016, 18 mayo). El bus I2C en Arduino. Luis Llamas.
<https://www.luisllamas.es/arduino-i2c/>
- [26] www.galileu.net. (n.d.). KNX España - KNX Association [Official website].
KNX España. <https://www.knx.es/es>
- [27] <https://www.areatecnologia.com>. (n.d.). El sistema KNX Domótica.
<https://www.areatecnologia.com/electricidad/knx.html>
- [28] Pedro González Gil. (2022, 1 marzo). Instalación y primeros pasos en Home Assistant [Vídeo]. YouTube. <https://www.youtube.com/watch?v=AZfO2sxmJIs>
- [29] Mark Watt Tech. (2023, 13 abril). HOW TO - Install HACS 2023 (Complete guide) [Vídeo]. YouTube. <https://www.youtube.com/watch?v=oXaGqjaKbeE>
- [30] Domótica Solar. (2022, 1 mayo). OCPP: Integra en Home Assistant tu cargador de vehículo eléctrico. [Vídeo]. YouTube.
<https://www.youtube.com/watch?v=7ngJjPKIUvg>
- [31] Assistant, H. (s. f.-a). Help. Home Assistant.
<https://www.home-assistant.io/help/>
- [32] Wire - Arduino reference. (s. f.).
<https://www.arduino.cc/reference/en/language/functions/communication/wire/>
- [33] Adafruit SHT31 Library - Arduino Reference. (s. f.).
<https://www.arduino.cc/reference/en/libraries/adafruit-sht31-library/>
- [34] WiFinINA - Arduino reference. (s. f.).
<https://www.arduino.cc/reference/en/libraries/wifinina/>

[35] PubSubClient - Arduino reference. (s. f.).
<https://www.arduino.cc/reference/en/libraries/pubsubclient/>

[36] Antonio Párraga. (2021, 12 junio). Crea tu CASA INTELIGENTE con Home Assistant [Vídeo]. YouTube. <https://www.youtube.com/watch?v=wFvIMPtK3dg>

[37] Domótica Solar. (2021, 7 abril). Home Assistant: NOTIFICACIONES via APP [Vídeo]. YouTube. <https://www.youtube.com/watch?v=8awXkgpw7ro>

6. ANEXOS

6.1. PROGRAMACIÓN KNX EN ETS5

6.1.1. Nuevo proyecto

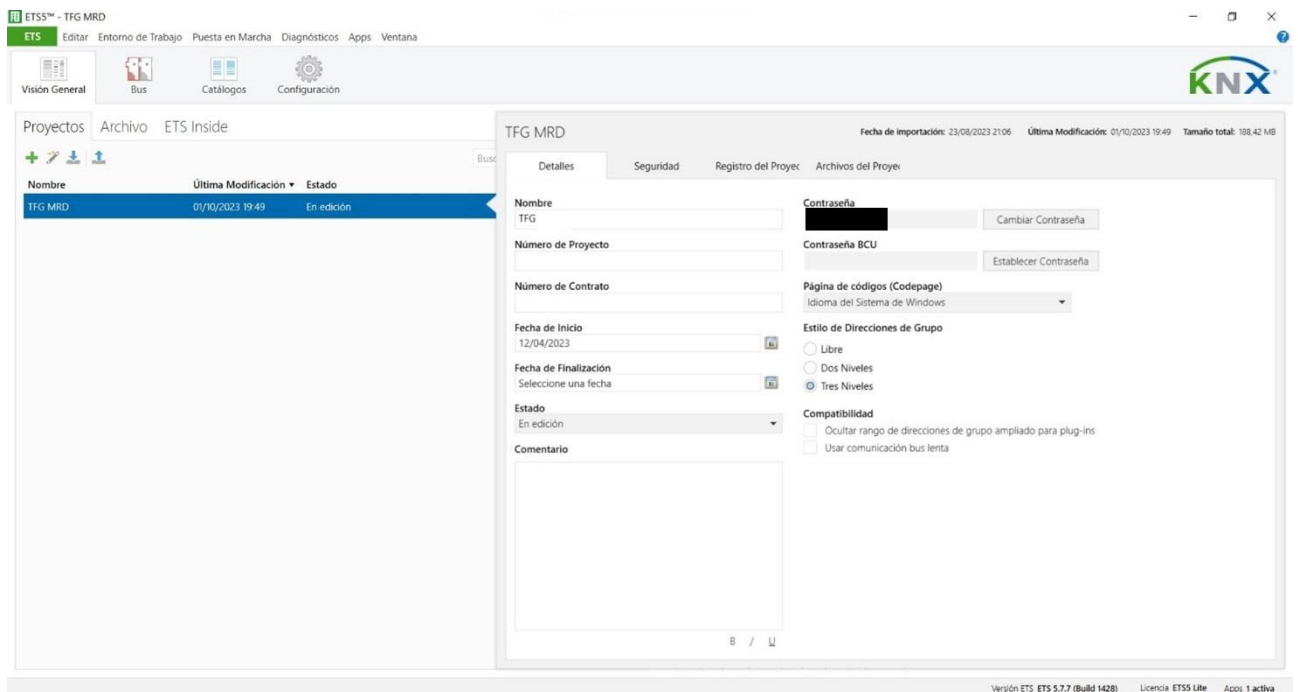


Figura 6.1.1.1: Nuevo proyecto

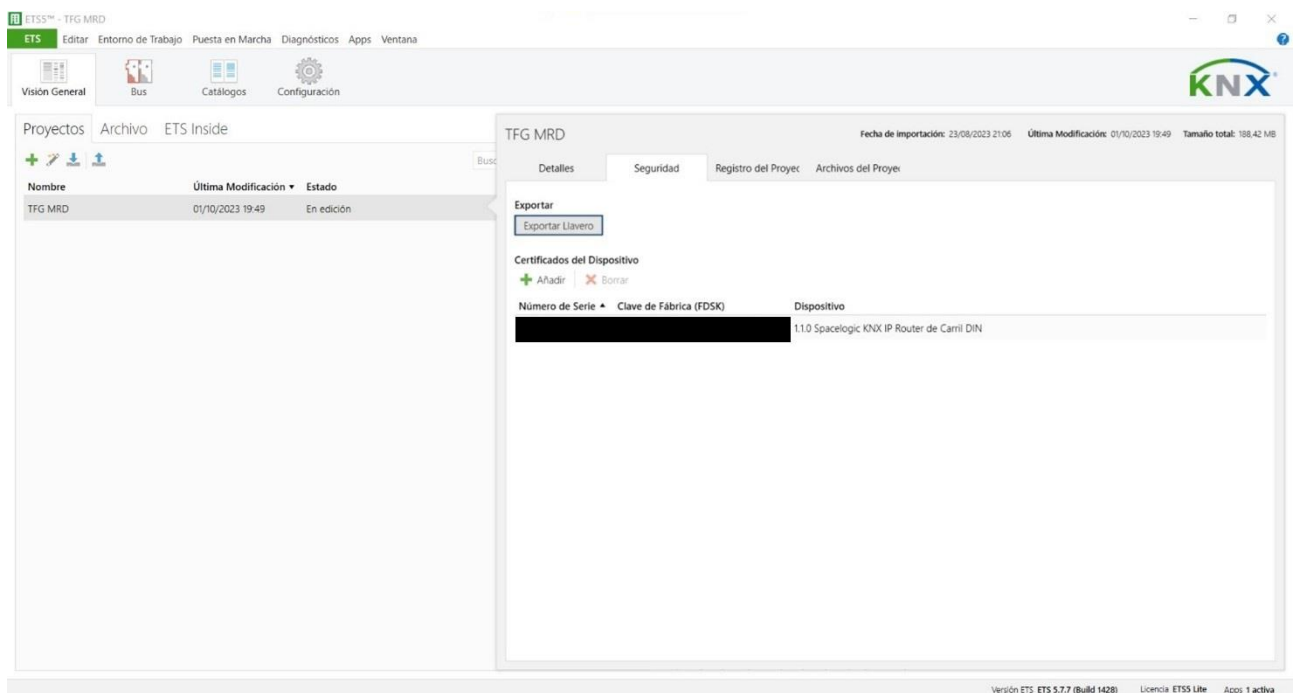


Figura 6.1.1.2: Seguridad

6.1.2. Edificio

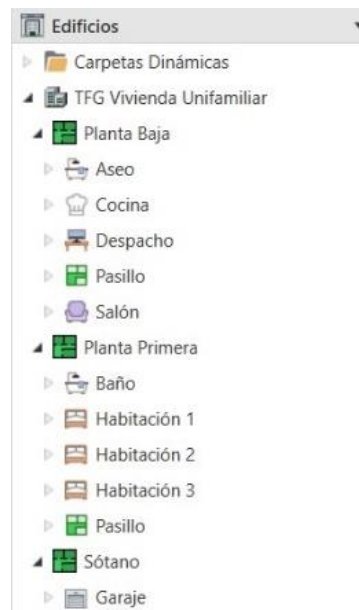


Figura 6.1.2.1: Edificio

6.1.3. Dispositivos

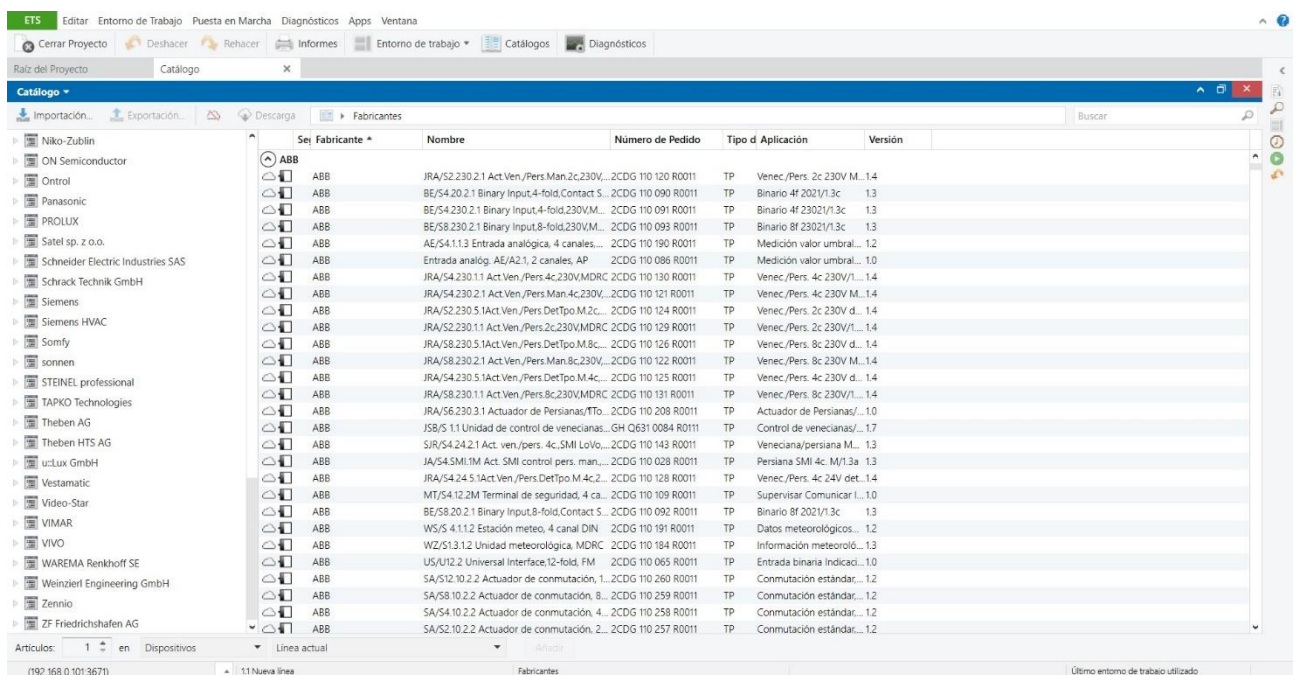


Figura 6.1.3.1: Catálogo ETS5

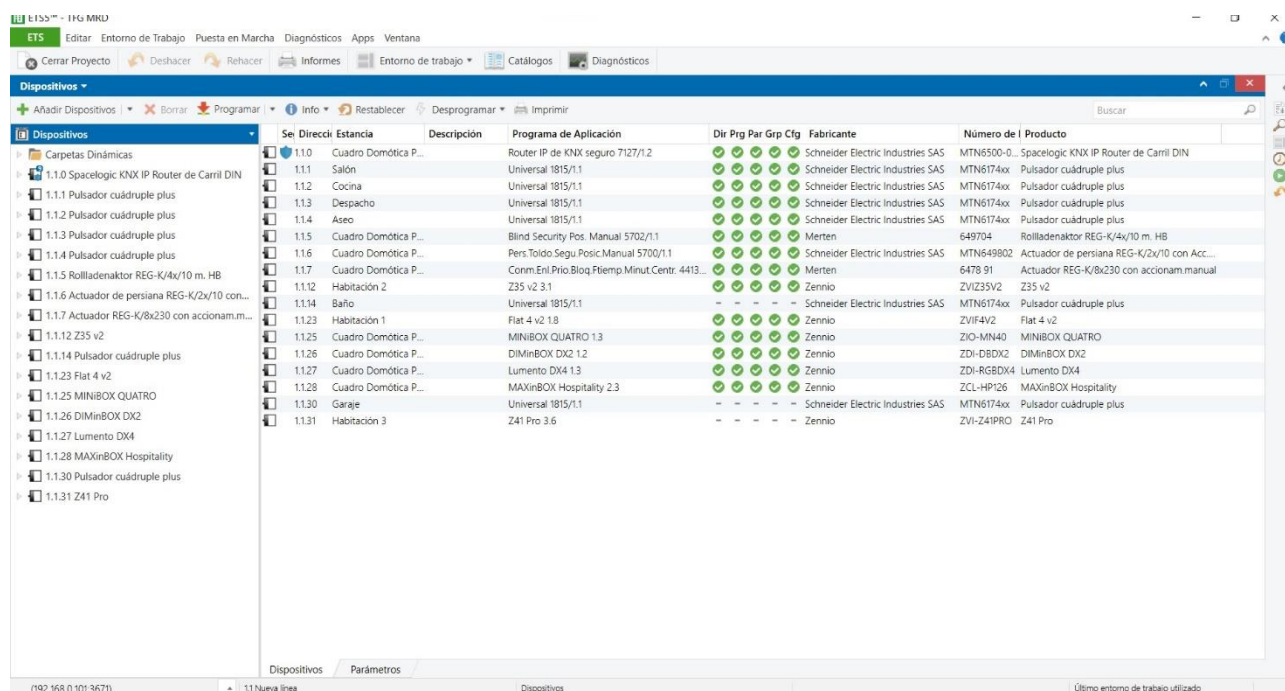


Figura 6.1.3.2: Dispositivos

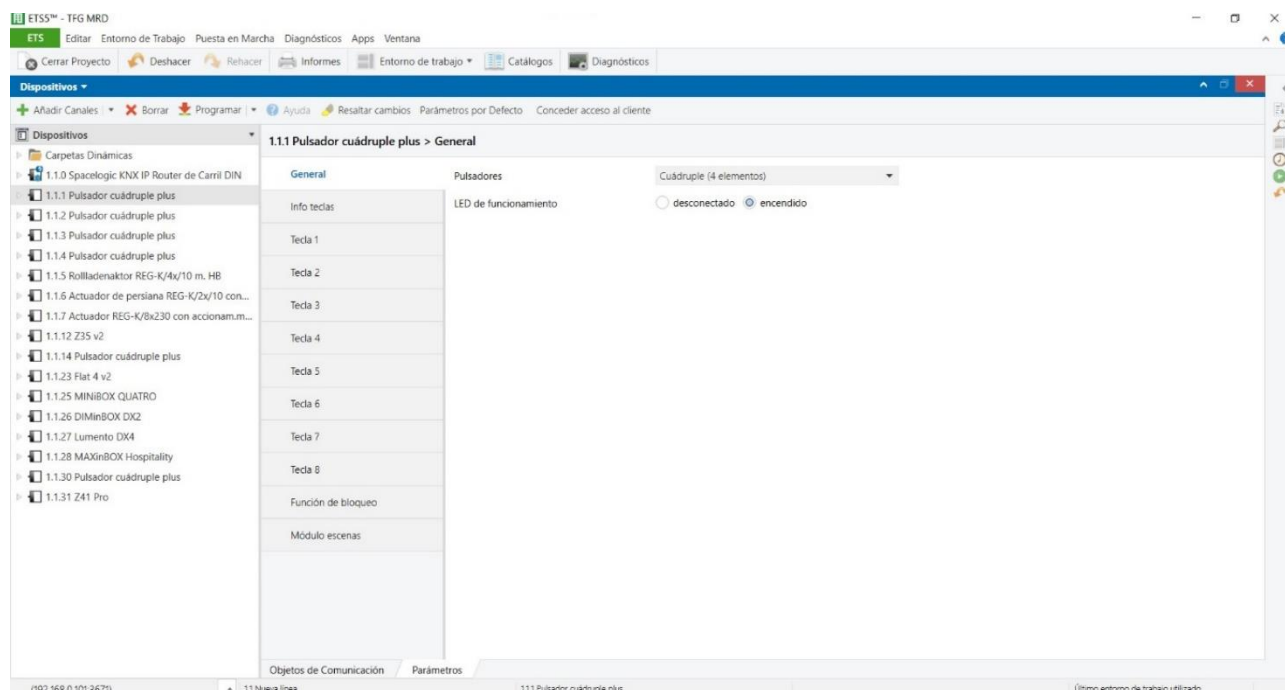


Figura 6.1.3.3: Configuración parámetros

6.1.4. Direcciones de grupo

Dirección *	Nombre	Descripción	Centra	Parar	Tipo de D.	Longitud	Nº de	Último Valor
0/0/1	OFF general		No	No	1 bit		1	
0/0/2	Bajar persianas		No	No			0	
0/0/3	ON/OFF Canal 1		No	No			0	
0/0/4	ON/OFF Canal 1		No	No			0	
0/0/5	Estado Canal 1		No	No			0	
0/0/6	Estado Canal 2		No	No			0	
0/0/7	Mover Persiana A		No	No			0	
0/0/8	STOP Persiana A		No	No			0	
0/0/9	OFF general		No	No			0	
0/0/10	Bajar persianas		No	No			0	
0/0/11	ON/OFF Canal 1		No	No			0	
0/0/12	ON/OFF Canal 1		No	No			0	
0/0/13	Estado Canal 1		No	No			0	
0/0/14	Estado Canal 2		No	No			0	
0/0/15	Mover Persiana A		No	No			0	
0/0/16	STOP Persiana A		No	No			0	
0/0/17	OFF general		No	No			0	
0/0/18	Bajar persianas		No	No			0	
0/0/19	ON/OFF Canal 1		No	No			0	
0/0/20	ON/OFF Canal 1		No	No			0	
0/0/21	Estado Canal 1		No	No			0	
0/0/22	Estado Canal 2		No	No			0	
0/0/23	Mover Persiana A		No	No			0	
0/0/24	STOP Persiana A		No	No			0	
0/0/25	OFF general		No	No			0	
0/0/26	Bajar persianas		No	No			0	
0/0/27	ON/OFF Canal 1		No	No			0	
0/0/28	ON/OFF Canal 1		No	No			0	
0/0/29	Estado Canal 1		No	No			0	
0/0/30	Estado Canal 2		No	No			0	
0/0/31	Mover Persiana A		No	No			0	

Figura 6.1.4.1: Direcciones de grupo

Dirección *	Nombre	Descripción	Centra	Parar	Tipo de D.	Longitud	Nº de	Último Valor
1/1/0	ON/OFF Canal 1 Salón		No	No	switch	1 bit	3	
1/1/1	ON/OFF Canal 2 Salón		No	No	switch	1 bit	4	
1/1/2	ON/OFF Canal 1 Cocina		No	No	1 bit		3	
1/1/3	ON/OFF Canal 2 Cocina		No	No	1 bit		3	
1/1/4	ON/OFF Canal 1 Despacho		No	No	1 bit		2	
1/1/5	ON/OFF Canal 2 Despacho		No	No	1 bit		2	
1/1/6	ON/OFF Canal 1 Aseo		No	No	switch	1 bit	2	
1/1/7	ON/OFF Canal 2 Aseo		No	No	1 bit		2	
1/1/8	ON/OFF Baño		No	No	switch	1 bit	4	
1/1/9	ON/OFF Habitación 1		No	No	switch	1 bit	4	
1/1/10	CONTROL TIMER - Baño		No	No	switch	1 bit	1	

Figura 6.1.4.2: Direcciones de grupo

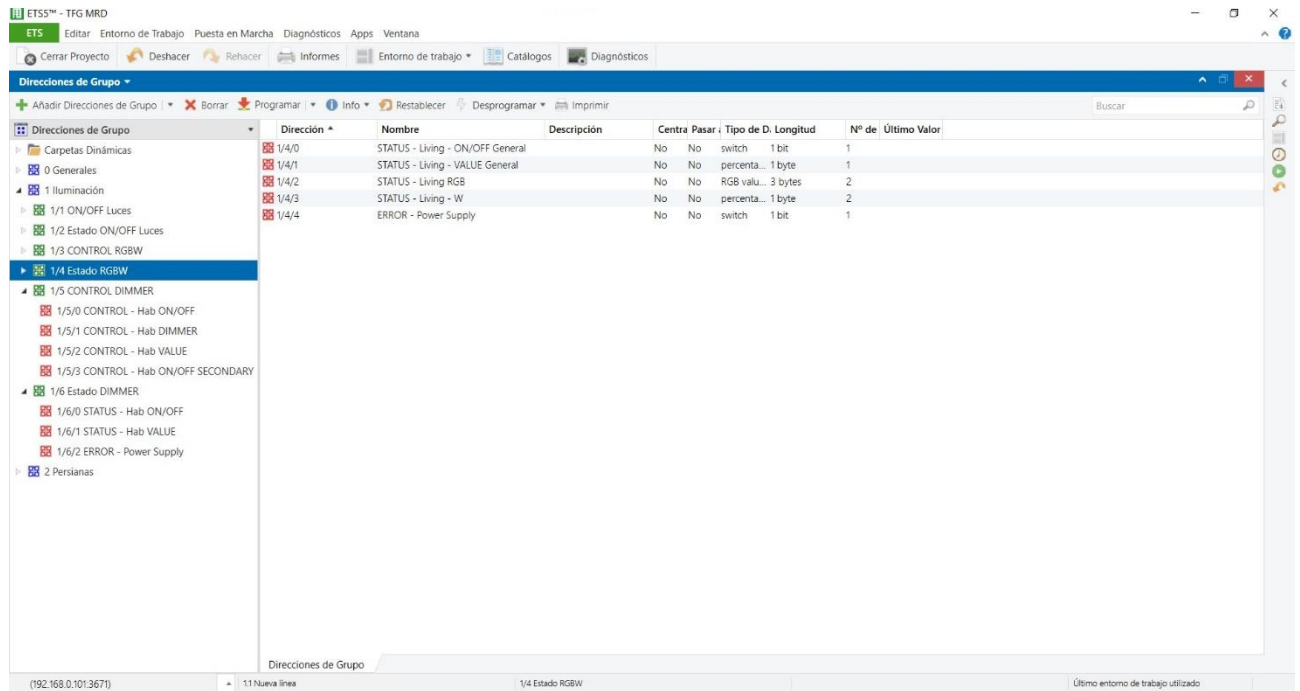


Figura 6.1.4.3: Direcciones de grupo

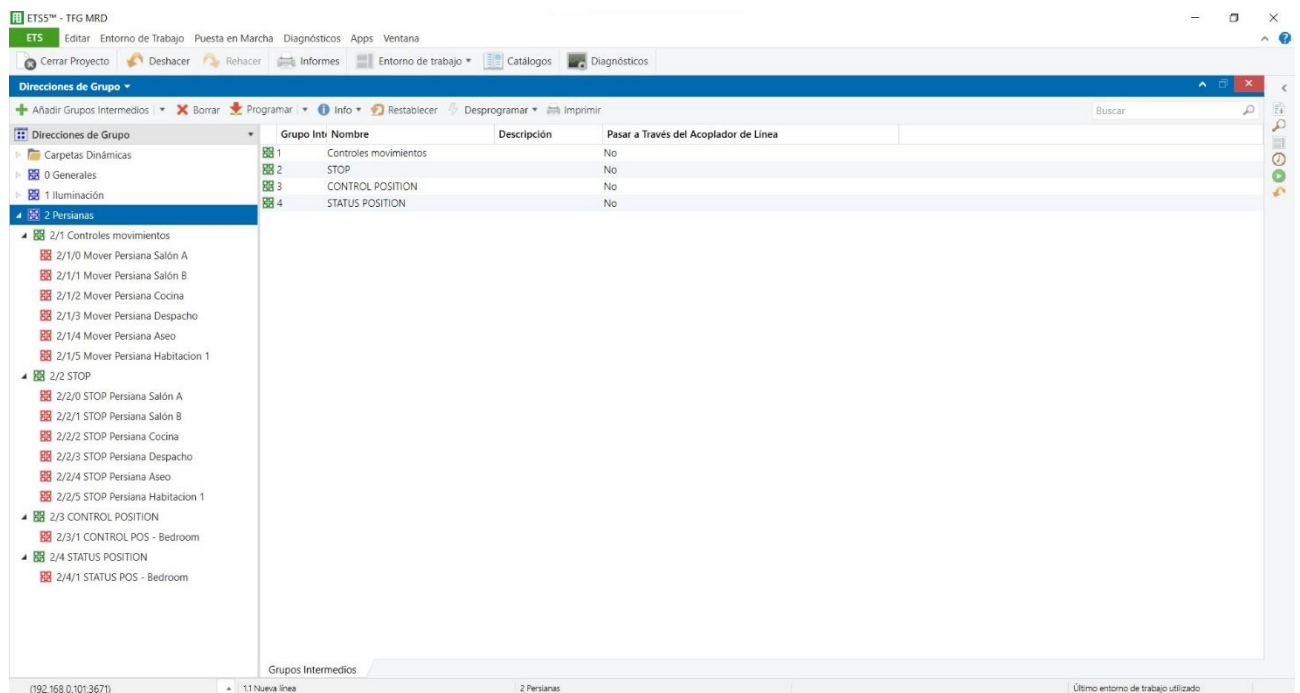


Figura 6.1.4.4: Direcciones de grupo

Sei	Dirección	Estancia	Descripción	Programa de Aplicación	Dir	Pr	Gr	Grp	Cfg	Fabricante	Número de Producto
1.1.0		Cuadro Domótica P...	Router IP de KNX seguro 7127/1.2	Universal 1815/1.1	✓	✓	✓	✓	✓	Schneider Electric Industries SAS	MTN6500... Spacelogic KNX IP Router de Carril DIN
1.1.1		Salón	Universal 1815/1.1	Universal 1815/1.1	✓	✓	✓	✓	✓	Schneider Electric Industries SAS	MTN6174xx Pulsador cuádruple plus
1.1.2		Cocina	Universal 1815/1.1	Universal 1815/1.1	✓	✓	✓	✓	✓	Schneider Electric Industries SAS	MTN6174xx Pulsador cuádruple plus
1.1.3		Despacho	Universal 1815/1.1	Universal 1815/1.1	✓	✓	✓	✓	✓	Schneider Electric Industries SAS	MTN6174xx Pulsador cuádruple plus
1.1.4		Aseo	Universal 1815/1.1	Universal 1815/1.1	✓	✓	✓	✓	✓	Schneider Electric Industries SAS	MTN6174xx Pulsador cuádruple plus
1.1.5		Cuadro Domótica P...	Blind Security Pos. Manual 5702/1.1	Universal 1815/1.1	✓	✓	✓	✓	✓	Merten	649704 Rollladenaktor REG-K/4x/10 m. HB
1.1.6		Cuadro Domótica P...	Pers.Toldo.Segu.Posic.Manual 5700/1.1	Universal 1815/1.1	✓	✓	✓	✓	✓	Schneider Electric Industries SAS	MTN649802 Actuador de persiana REG-K/2x/10 con Acc...
1.1.7		Cuadro Domótica P...	Corrim.Enl.Prio.Bloq.Ftiemp.Minut.Centr. 4413...	Universal 1815/1.1	✓	✓	✓	✓	✓	Merten	6478 91 Actuador REG-K/8x230 con accionam.manual
1.1.12		Habitación 2	Z35 v2 3.1	Universal 1815/1.1	✓	✓	✓	✓	✓	Zennio	ZVIZ35V2 Z35 v2
1.1.14		Baño	Universal 1815/1.1	Universal 1815/1.1	✓	✓	✓	✓	✓	Schneider Electric Industries SAS	MTN6174xx Pulsador cuádruple plus
1.1.23		Habitación 1	Flat 4 v2 1.8	Universal 1815/1.1	✓	✓	✓	✓	✓	Zennio	ZVIF4V2 Flat 4 v2
1.1.25		Cuadro Domótica P...	MINIBOX QUATRO 1.3	Universal 1815/1.1	✓	✓	✓	✓	✓	Zennio	ZIO-MN40 MINIBOX QUATRO
1.1.26		Cuadro Domótica P...	DIMinBOX DX2 1.2	Universal 1815/1.1	✓	✓	✓	✓	✓	Zennio	ZDI-DBDX2 DIMinBOX DX2
1.1.27		Cuadro Domótica P...	Lumento DX4 1.3	Universal 1815/1.1	✓	✓	✓	✓	✓	Zennio	ZDI-RGBD... Lumento DX4
1.1.28		Cuadro Domótica P...	MAXinBOX Hospitality 2.3	Universal 1815/1.1	✓	✓	✓	✓	✓	Zennio	ZCL-HP126 MAXinBOX Hospitality
1.1.30		Garaje	Universal 1815/1.1	Universal 1815/1.1	✓	✓	✓	✓	✓	Schneider Electric Industries SAS	MTN6174xx Pulsador cuádruple plus
1.1.31		Habitación 3	Z41 Pro 3.6	Universal 1815/1.1	✓	✓	✓	✓	✓	Zennio	ZVI-Z41PRO Z41 Pro

Figura 6.1.4.5: Topología proyecto

6.1.5. Objetos de comunicación

Número	Nombre	Función del Objeto	Descripción	Dirección de Grupo	Longitu	C	R	W	T	U	Tipo de Da	Prioridad
0	Objeto de conexión A	Tecla 1	ON/OFF Canal 1 Salón	1/1/0, 1/1/1, 1/2/0, 1/2/1	1 bit	C	-	W	T	-	Bajo	
3	Objeto de conexión A	Tecla 2	OFF general	0/0/1, 1/2/0, 1/2/1	1 bit	C	-	W	T	-	Bajo	
4	Objeto de conexión B	Tecla 2	Bajar persianas	0/2/1	1 bit	C	-	W	T	-	Bajo	
6	Objeto de conexión A	Tecla 3	ON/OFF Canal 2 Salón	1/1/1, 1/2/0, 1/2/1	1 bit	C	-	W	T	-	Bajo	
9	Objeto de conexión A	Tecla 4			1 bit	C	-	W	T	-	Bajo	
12	Objeto Parada/Orient.lamas	Tecla 5	STOP Persiana Salón A	2/2/0	1 bit	C	-	T	-	-	Bajo	
13	Objeto de movimiento	Tecla 5	Mover Persiana Salón A	2/1/0	1 bit	C	-	T	-	-	Bajo	
15	Objeto Parada/Orient.lamas	Tecla 6	STOP Persiana Salón B	2/2/1	1 bit	C	-	T	-	-	Bajo	
16	Objeto de movimiento	Tecla 6	Mover Persiana Salón B	2/1/1	1 bit	C	-	T	-	-	Bajo	
18	Objeto Parada/Orient.lamas	Tecla 7	STOP Persiana Salón A	2/2/0	1 bit	C	-	T	-	-	Bajo	
19	Objeto de movimiento	Tecla 7	Mover Persiana Salón A	2/1/0	1 bit	C	-	T	-	-	Bajo	
21	Objeto Parada/Orient.lamas	Tecla 8	STOP Persiana Salón B	2/2/1	1 bit	C	-	T	-	-	Bajo	
22	Objeto de movimiento	Tecla 8	Mover Persiana Salón B	2/1/1	1 bit	C	-	T	-	-	Bajo	

Figura 6.1.5.1: Objetos de comunicación

6.1.6. Dispositivos programados

Se	Dirección	Estancia	Descripción	Programa de Aplicación	Dir	Prg	Par	Grip	Cfg	Fabricante	Número de Producto
1.1.0		Cuadro Domótica P...	Router IP de KNX seguro 7127/1.2	Router IP de KNX seguro 7127/1.2	✓	✓	✓	✓	✓	Schneider Electric Industries SAS	MTN6500-... Spacelogic KNX IP Router de Carril DIN
1.1.1		Salón	Universal 1815/1.1	Universal 1815/1.1	✓	✓	✓	✓	✓	Schneider Electric Industries SAS	MTN6174xx Pulsador cuádruple plus
1.1.2		Cocina	Universal 1815/1.1	Universal 1815/1.1	✓	✓	✓	✓	✓	Schneider Electric Industries SAS	MTN6174xx Pulsador cuádruple plus
1.1.3		Despacho	Universal 1815/1.1	Universal 1815/1.1	✓	✓	✓	✓	✓	Schneider Electric Industries SAS	MTN6174xx Pulsador cuádruple plus
1.1.4		Aseo	Universal 1815/1.1	Universal 1815/1.1	✓	✓	✓	✓	✓	Schneider Electric Industries SAS	MTN6174xx Pulsador cuádruple plus
1.1.5		Cuadro Domótica P...	Blind Security Pos. Manual 5702/1.1	Blind Security Pos. Manual 5702/1.1	✓	✓	✓	✓	✓	Merten	649704 Rolladenaktor REG-K/4x/10 m. HB
1.1.6		Cuadro Domótica P...	Pers.Toldo Seg. Posic.Manual 5700/1.1	Pers.Toldo Seg. Posic.Manual 5700/1.1	✓	✓	✓	✓	✓	Schneider Electric Industries SAS	MTN649802 Actuador de persiana REG-K/2x/10 con Acc...
1.1.7		Cuadro Domótica P...	Conn.Enl.Prio Bloq.Pfemp.Minut.Centr. 4413...	Conn.Enl.Prio Bloq.Pfemp.Minut.Centr. 4413...	✓	✓	✓	✓	✓	Merten	6478 91 Actuador REG-K/8x230 con accionam.manual
1.1.8		Habitación 2	Z35 v2 3.1	Z35 v2 3.1	✓	✓	✓	✓	✓	Zennio	ZVI235V2 Z35 v2
1.1.14		Baño	Universal 1815/1.1	Universal 1815/1.1	✓	✓	✓	✓	✓	Schneider Electric Industries SAS	MTN6174xx Pulsador cuádruple plus
1.1.23		Habitación 1	Flat 4 v2 1.8	Flat 4 v2 1.8	✓	✓	✓	✓	✓	Zennio	ZVIF4V2 Flat 4 v2
1.1.25		Cuadro Domótica P...	MINIBOX QUATRO 1.3	MINIBOX QUATRO 1.3	✓	✓	✓	✓	✓	Zennio	ZIO-MN40 MINIBOX QUATRO
1.1.2		Cuadro Domótica P...	DiMinBOX DX2 1.2	DiMinBOX DX2 1.2	✓	✓	✓	✓	✓	Zennio	ZDI-DBDX2 DiMinBOX DX2
1.1.27		Cuadro Domótica P...	Luminto DX4 1.3	Luminto DX4 1.3	✓	✓	✓	✓	✓	Zennio	ZDI-RGBD... Luminto DX4
1.1.28		Cuadro Domótica P...	MAXinBOX Hospitality 2.3	MAXinBOX Hospitality 2.3	✓	✓	✓	✓	✓	Zennio	ZCL-HP126 MAXinBOX Hospitality
1.1.30		Garaje	Universal 1815/1.1	Universal 1815/1.1	✓	✓	✓	✓	✓	Schneider Electric Industries SAS	MTN6174xx Pulsador cuádruple plus
1.1.31		Habitación 3	Z41 Pro 3.6	Z41 Pro 3.6	✓	✓	✓	✓	✓	Zennio	ZVI-Z41PRO Z41 Pro

Figura 6.1.6.1: Programación KNX en ETS5

6.1.7. Esquema de la solución implementada en ETS5

0	Área principal (Backbone)
0.0	IP Línea principal (Backbone)
1	Nueva área
1.0	IP Línea principal
1.1	TP Nueva línea
17 dispositivos en la línea	
1.1.0	Schneider Electric MTN6500-0103 Spacelogic KNX IP Router de Carril DIN Router IP de KNX seguro 7127/1.2
Cuadro Domótica Planta Baja	
1.1.15	Interfaz KNXnet/IP tunneling
1.1.16	Interfaz KNXnet/IP tunneling
1.1.17	Interfaz KNXnet/IP tunneling
1.1.18	Interfaz KNXnet/IP tunneling
1.1.19	Interfaz KNXnet/IP tunneling
1.1.20	Interfaz KNXnet/IP tunneling
1.1.21	Interfaz KNXnet/IP tunneling
1.1.22	Interfaz KNXnet/IP tunneling
1.1.1	Schneider Electric MTN6174xx Pulsador cuádruple plus Universal 1815/1.1
Salón	
1.1.2	Schneider Electric MTN6174xx Pulsador cuádruple plus Universal 1815/1.1
Cocina	
1.1.3	Schneider Electric MTN6174xx Pulsador cuádruple plus Universal 1815/1.1
Despacho	
1.1.4	Schneider Electric MTN6174xx Pulsador cuádruple plus Universal 1815/1.1

Aseo				
1.1.5	Merten	649704	Rollladenaktor REG-K/4x/10 m. HB	Blind Security Pos. Manual 5702/1.1
Cuadro Domótica Planta Baja				
1.1.6	Schneider Electric Industries SAS	MTN649802	Actuador de persiana REG-K/2x/10 con Acc.Manual	Pers.Toldo.Segu.Posic.Manual 5700/1.1
Cuadro Domótica Planta Baja				
1.1.7	Merten	6478 91	Actuador REG-K/8x230 con accionam.manual	Conm.Enl.Prio.Bloq.Ftiemp .Minut.Centr. 4413/1.0
Cuadro Domótica Planta Baja				
1.1.12	Zennio	ZVIZ35V2	Z35 v2	Z35 v2 3.1
Habitación 2				
1.1.14	Schneider Electric Industries SAS	MTN6174xx	Pulsador cuádruple plus	Universal 1815/1.1
Baño				
1.1.23	Zennio	ZVIF4V2	Flat 4 v2	Flat 4 v2 1.8
Habitación 1				
1.1.25	Zennio	ZIO-MN40	MINiBOX QUATRO	MINiBOX QUATRO 1.3
Cuadro Domótica Planta Primera				
1.1.26	Zennio	ZDI-DBDX2	DIMinBOX DX2	DIMinBOX DX2 1.2
Cuadro Domótica Planta Primera				
1.1.27	Zennio	ZDI-RGBDX4	Lumento DX4	Lumento DX4 1.3
Cuadro Domótica Planta Primera				
1.1.28	Zennio	ZCL-HP126	MAXinBOX Hospitality	MAXinBOX Hospitality 2.3
Cuadro Domótica Planta Primera				
1.1.30	Schneider Electric Industries SAS	MTN6174xx	Pulsador cuádruple plus	Universal 1815/1.1
Garaje				
1.1.31	Zennio	ZVI-Z41PRO	Z41 Pro	Z41 Pro 3.6
Habitación 3				

Figura 6.1.7.1: Solución KNX en ETS5

6.1.8. Ejemplos de funcionamiento



Figura 6.1.8.1: Funcionamiento

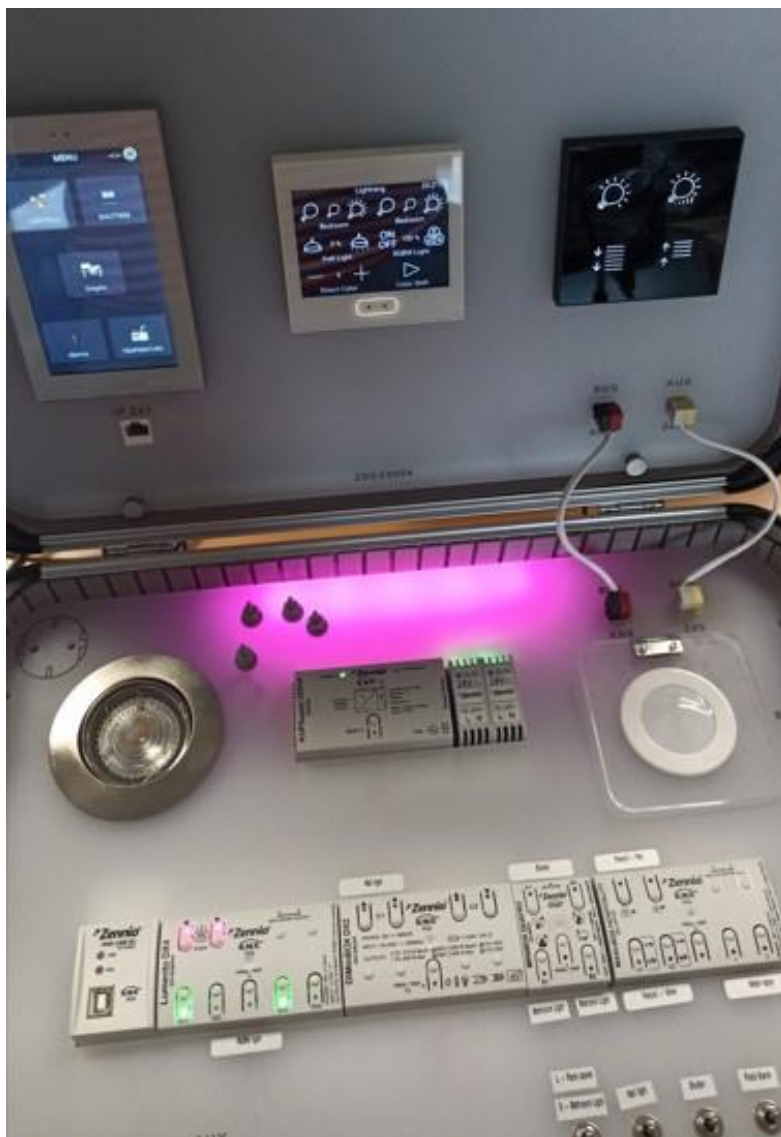


Figura 6.1.8.2: Funcionamiento

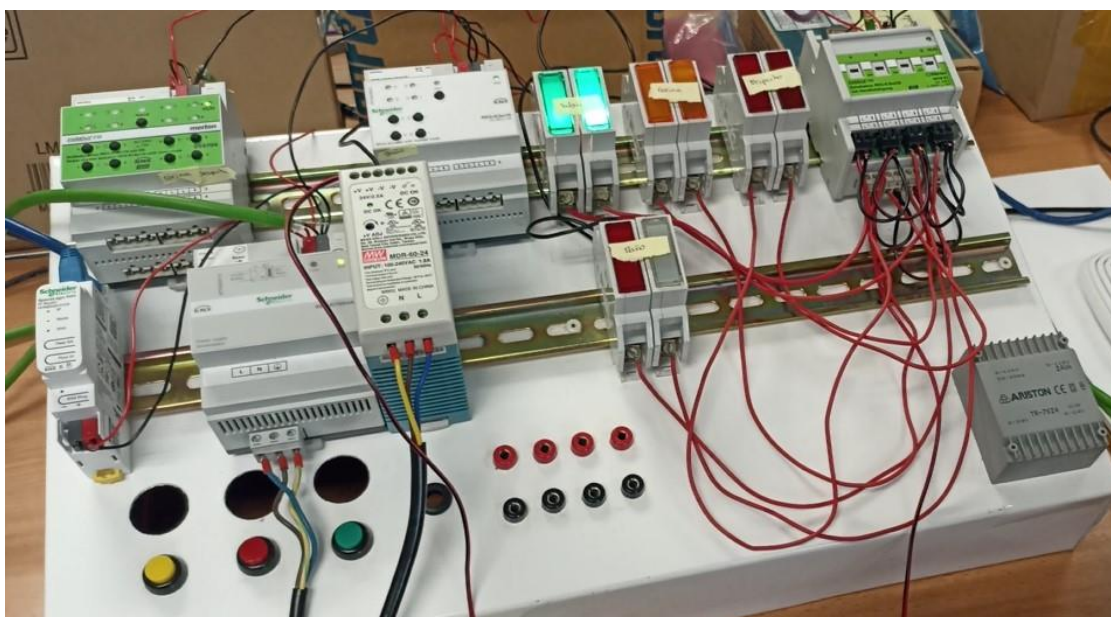


Figura 6.1.8.3: Funcionamiento

6.2. COHESIÓN CON HOME ASSISTANT

6.2.1. Instalación y configuración de la máquina virtual

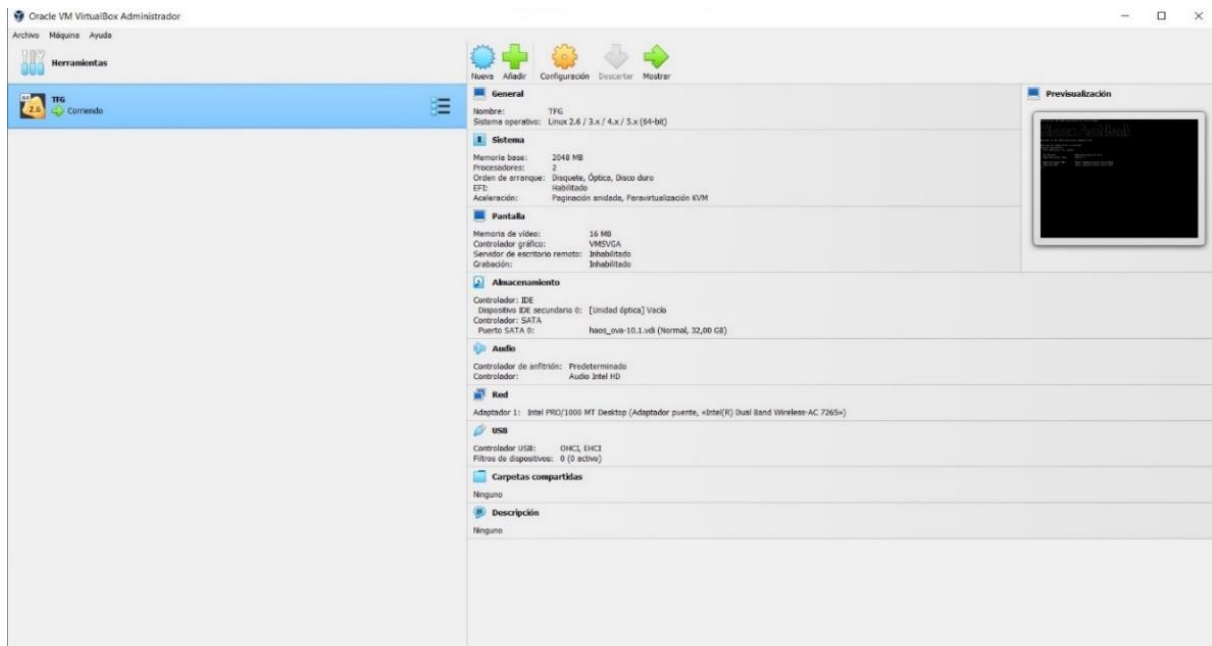


Figura 6.2.1.1: Configuración de máquina virtual



Figura 6.2.1.2: Máquina virtual

6.2.2. Acceso a la URL y creación de usuario

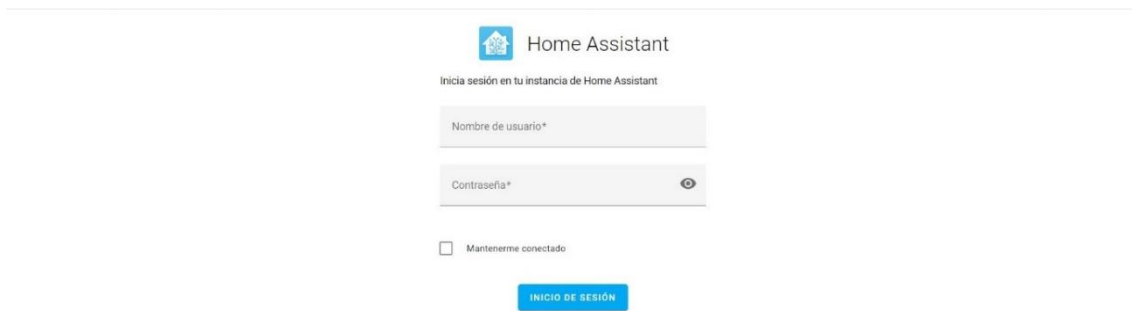


Figura 6.2.2.1: Creación de cuenta e inicio de sesión

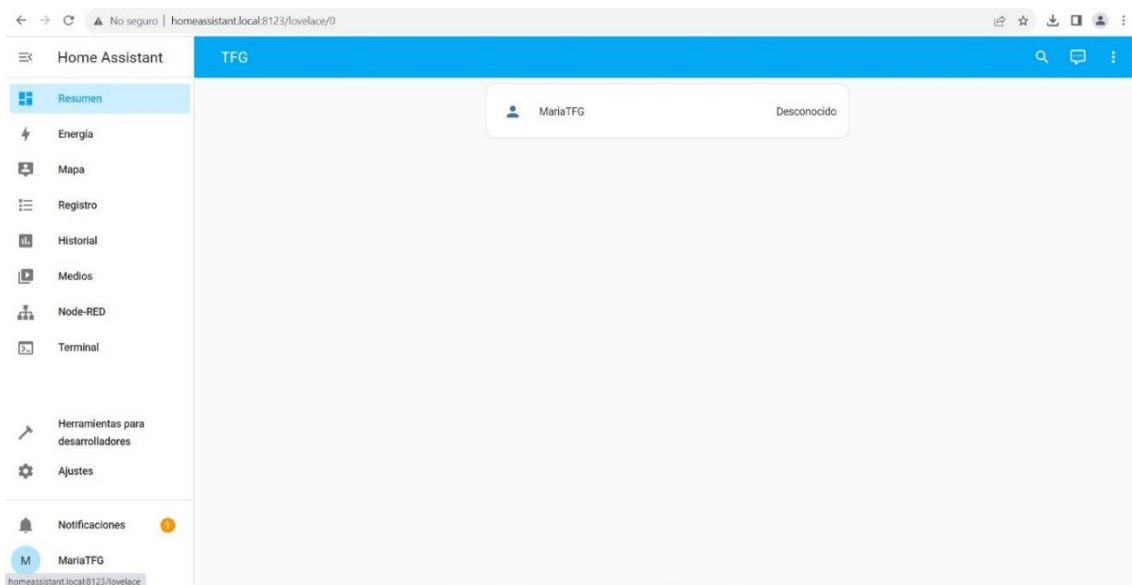


Figura 6.2.2.2: Home Assistant

6.2.3. Integraciones

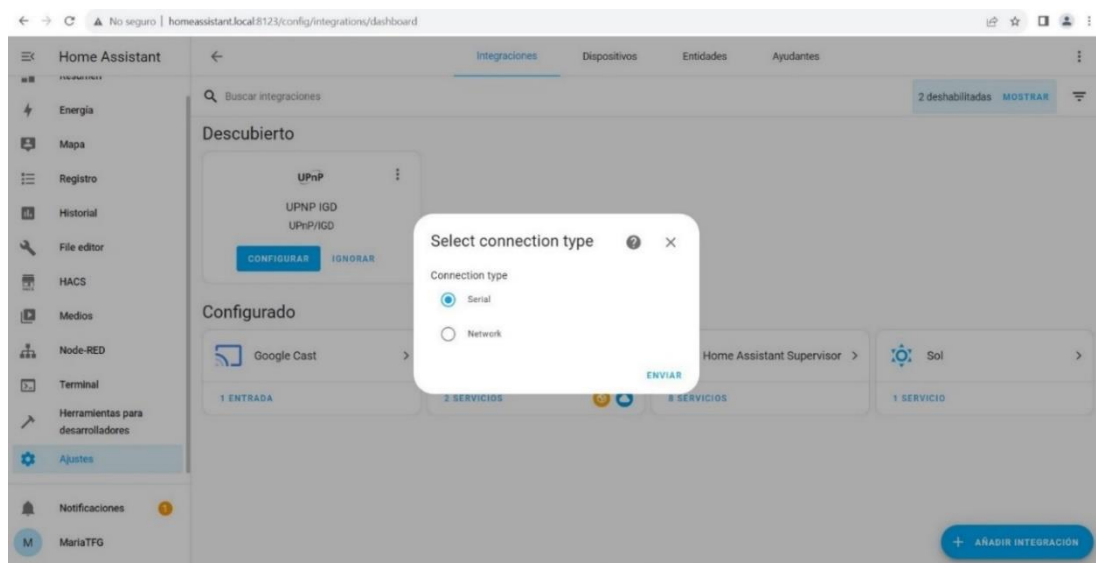


Figura 6.2.3.1: Integración de inversor

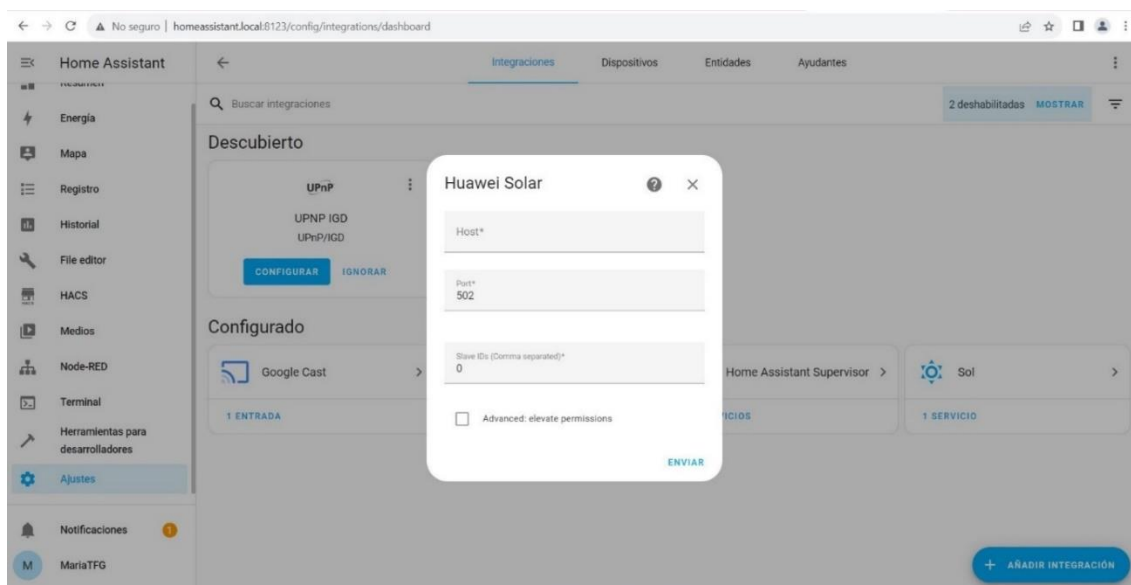


Figura 6.2.3.2: Integración de inversor

☰	Home Assistant	←	Integraciones	Dispositivos	Entidades	Ayudantes	⋮
☰	Resumen	🔍	Buscar dispositivos			Filtrando por integración "Huawei Solar: SUN2000-5KTL-L1" LIMPIAR	
⚡	Energía	↑	Dispositivo	Fabricante	Modelo	Área	Integración
📍	Mapa	🏠	Inverter	Huawei	SUN2000-5KTL-L1	—	Huawei Solar
📅	Registro	🏠	Power meter	<unknown>	<unknown>	—	Huawei Solar
📅	Historial						
📦	HACS						
📺	Medios						

Figura 6.2.3.3: Integración de inversor

Home Assistant	Inverter																				
Resumen	Información de Dispositivo SUN2000-5KTL-L1 por Huawei DESCARGAR DIAGNÓSTICOS Automatizaciones Todavía no se ha agregado ningún automatizaciones con este dispositivo. Puedes agregar uno haciendo clic en el botón + de arriba. Escenas Todavía no se ha agregado ningún escenas con este dispositivo. Puedes agregar uno haciendo clic en el botón + de arriba. Scripts Todavía no se ha agregado ningún scripts con este dispositivo. Puedes agregar uno haciendo clic en el botón + de arriba.																				
Energía	Sensores <table> <tr><td>Active power</td><td>1496 W</td></tr> <tr><td>Daily yield</td><td>1,34 kWh</td></tr> <tr><td>Day active power peak</td><td>1516 W</td></tr> <tr><td>Efficiency</td><td>98,38%</td></tr> <tr><td>Input power</td><td>1520 W</td></tr> <tr><td>PV 1 Current</td><td>3,21 A</td></tr> <tr><td>PV 1 Voltage</td><td>495,7 V</td></tr> <tr><td>PV 2 Current</td><td>0,0 A</td></tr> <tr><td>PV 2 Voltage</td><td>0,0 V</td></tr> <tr><td>Total yield</td><td>9400,28 kWh</td></tr> </table>	Active power	1496 W	Daily yield	1,34 kWh	Day active power peak	1516 W	Efficiency	98,38%	Input power	1520 W	PV 1 Current	3,21 A	PV 1 Voltage	495,7 V	PV 2 Current	0,0 A	PV 2 Voltage	0,0 V	Total yield	9400,28 kWh
Active power	1496 W																				
Daily yield	1,34 kWh																				
Day active power peak	1516 W																				
Efficiency	98,38%																				
Input power	1520 W																				
PV 1 Current	3,21 A																				
PV 1 Voltage	495,7 V																				
PV 2 Current	0,0 A																				
PV 2 Voltage	0,0 V																				
Total yield	9400,28 kWh																				
Mapa	Registro 6 de mayo de 2023 Inverter Alarms cambiado a None 10:34:52 - Hace 25 segundos Inverter Off-grid switch cambiado a Off-grid switch disabled 10:34:52 - Hace 25 segundos Inverter Off-grid status cambiado a On-grid 10:34:52 - Hace 25 segundos Inverter DSP data collection cambiado a DSP data collection 10:34:52 - Hace 25 segundos Inverter PV connection status cambiado a PV connected 10:34:52 - Hace 25 segundos																				
Registro																					
Historial																					
HACS																					
Medios																					
Herramientas para desarrolladores																					
Ajustes																					
Notificaciones																					

Figura 6.2.3.4: Integración de inversor

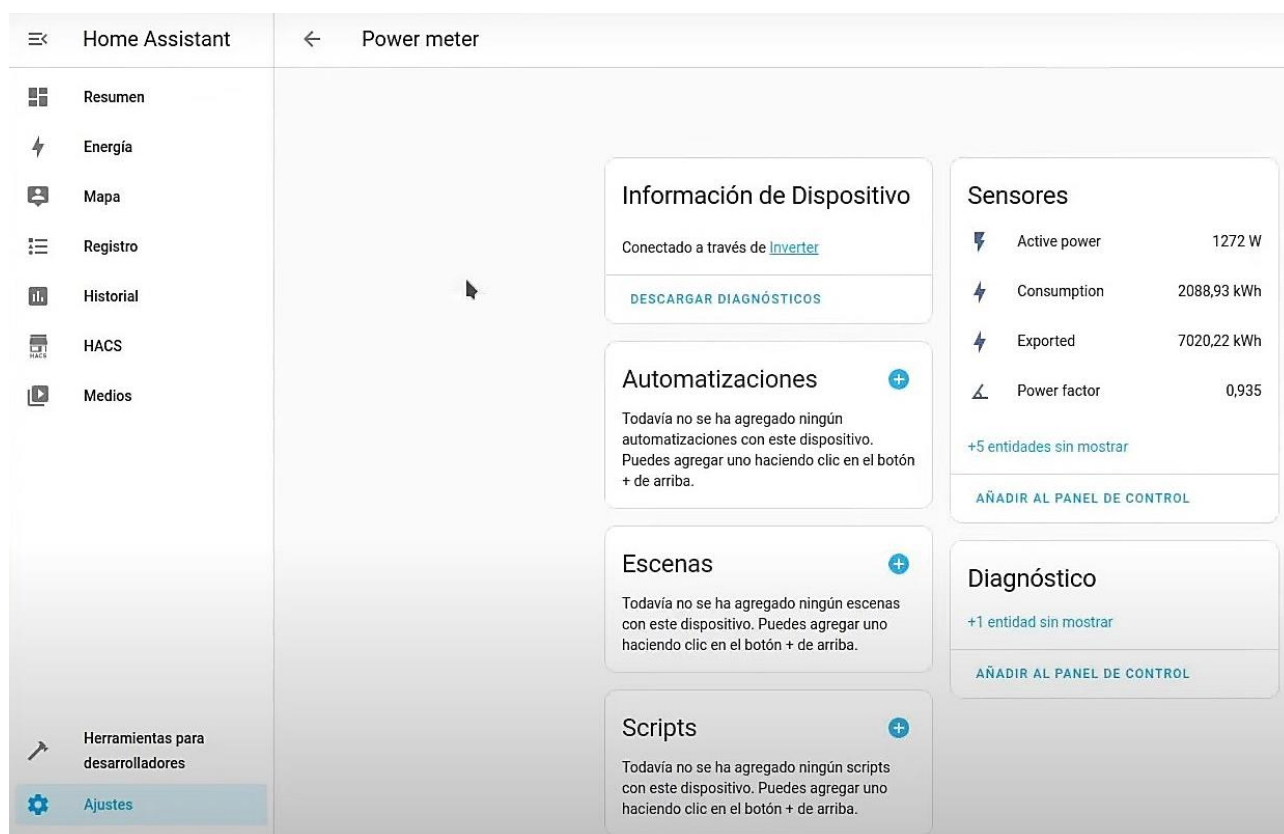


Figura 6.2.3.5: Integración de inversor

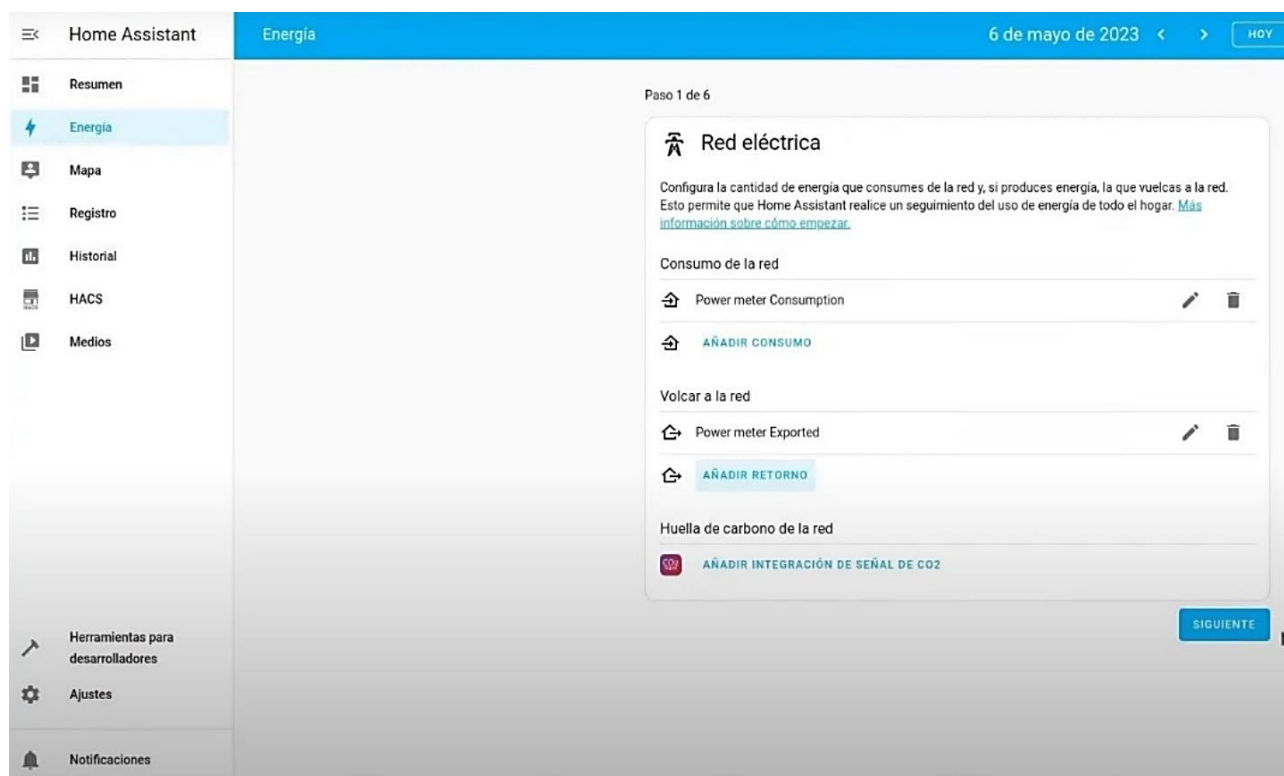


Figura 6.2.3.6: Integración de inversor



Figura 6.2.3.7: Integración de inversor

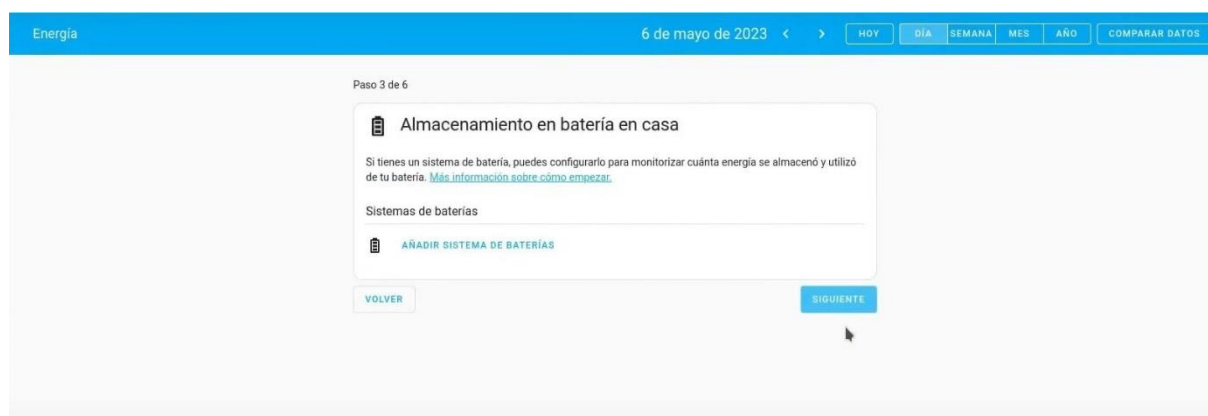


Figura 6.2.3.8: Integración de inversor



Figura 6.2.3.9: Integración de inversor



Figura 6.2.3.10: Integración de inversor

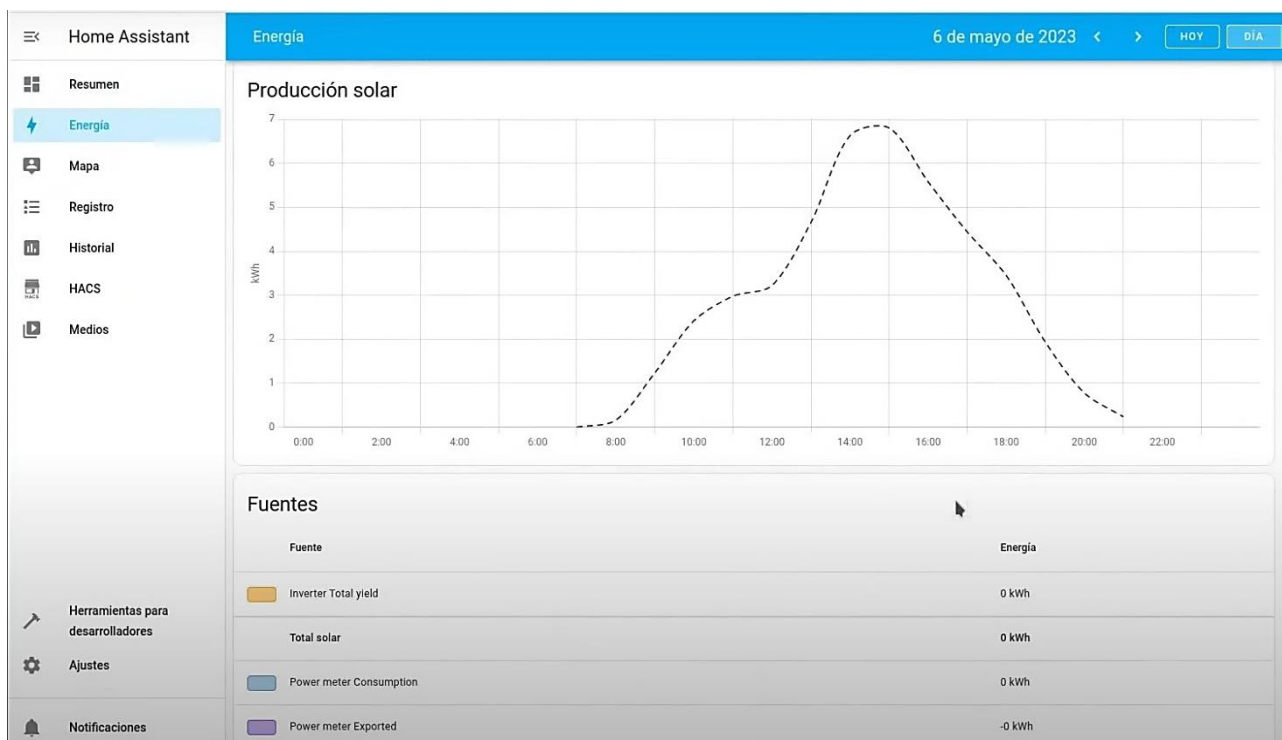


Figura 6.2.3.11: Integración de inversor

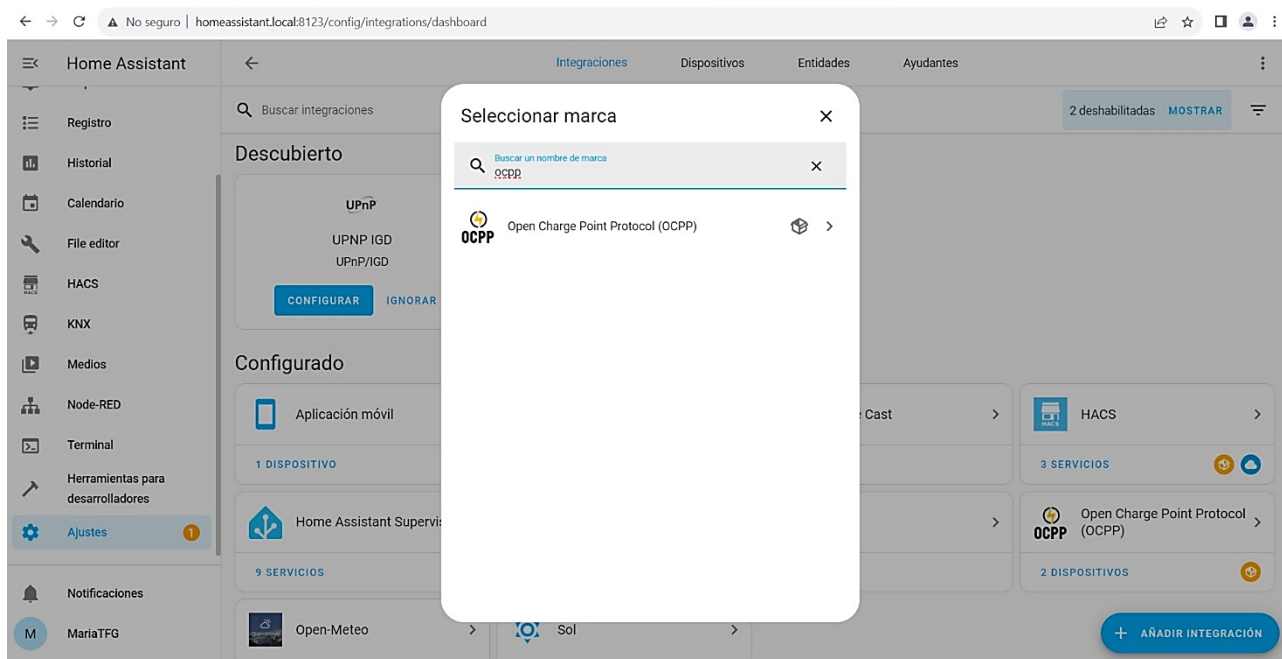


Figura 6.2.3.12: Integración de cargador

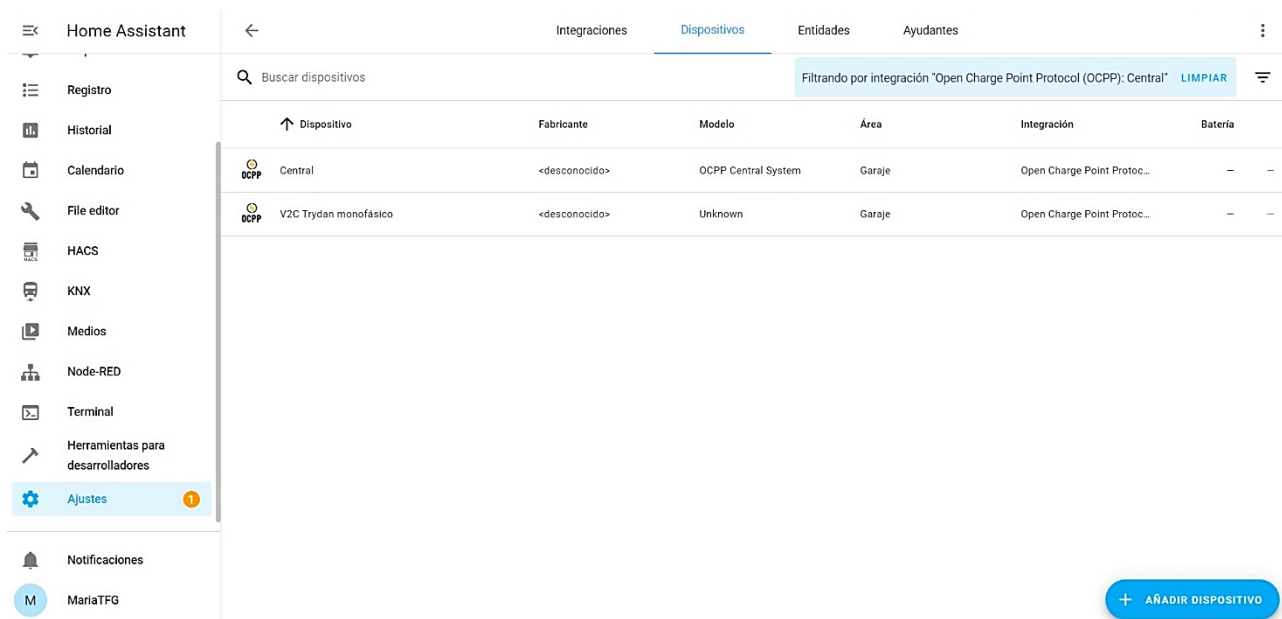


Figura 6.2.3.13: Integración de cargador

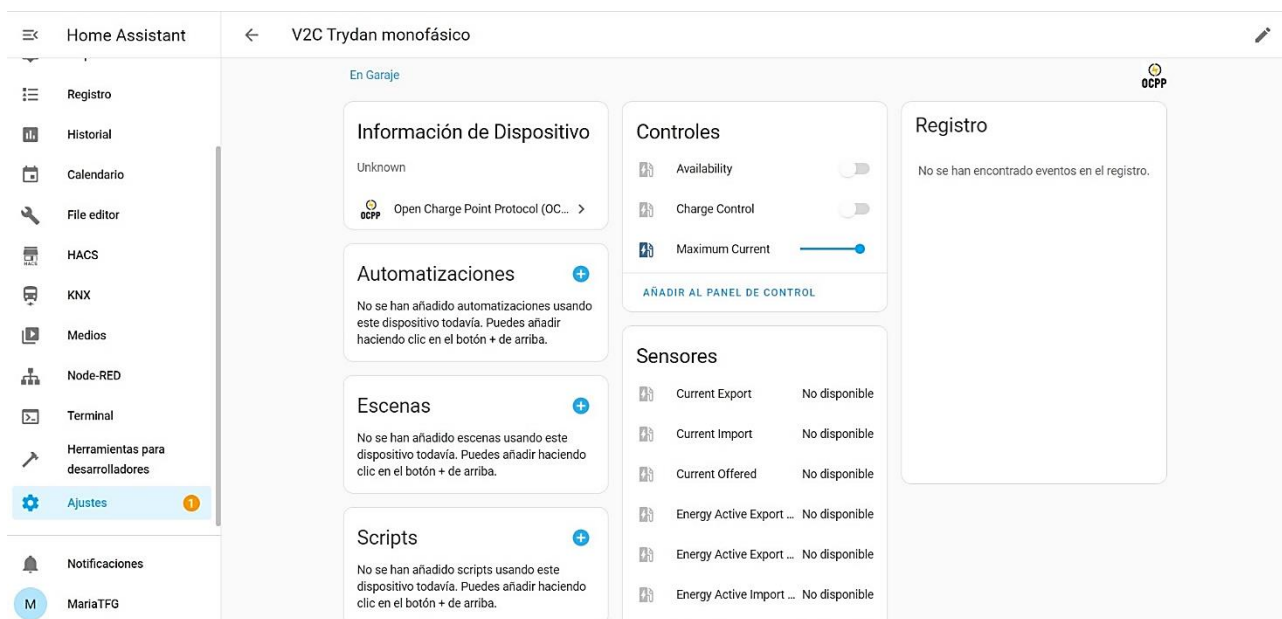


Figura 6.2.3.14: Integración de cargador

6.2.4. Complementos

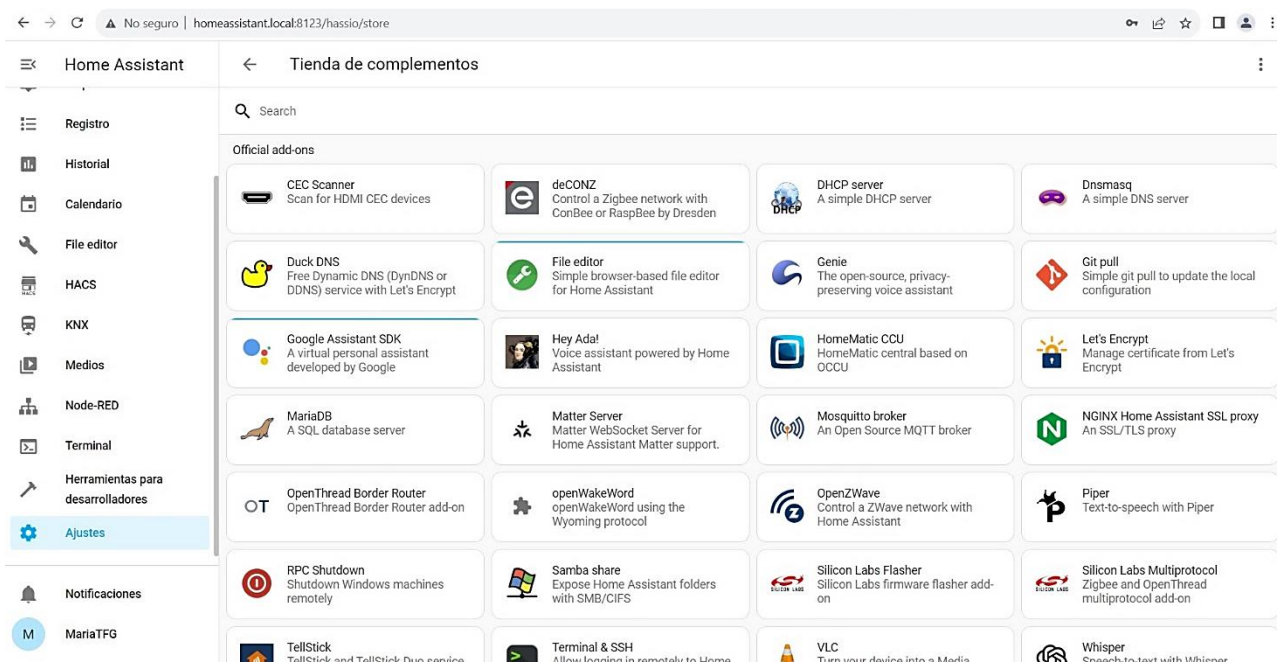


Figura 6.2.4.1: Tienda de complementos

6.2.5. Instalación de app móvil

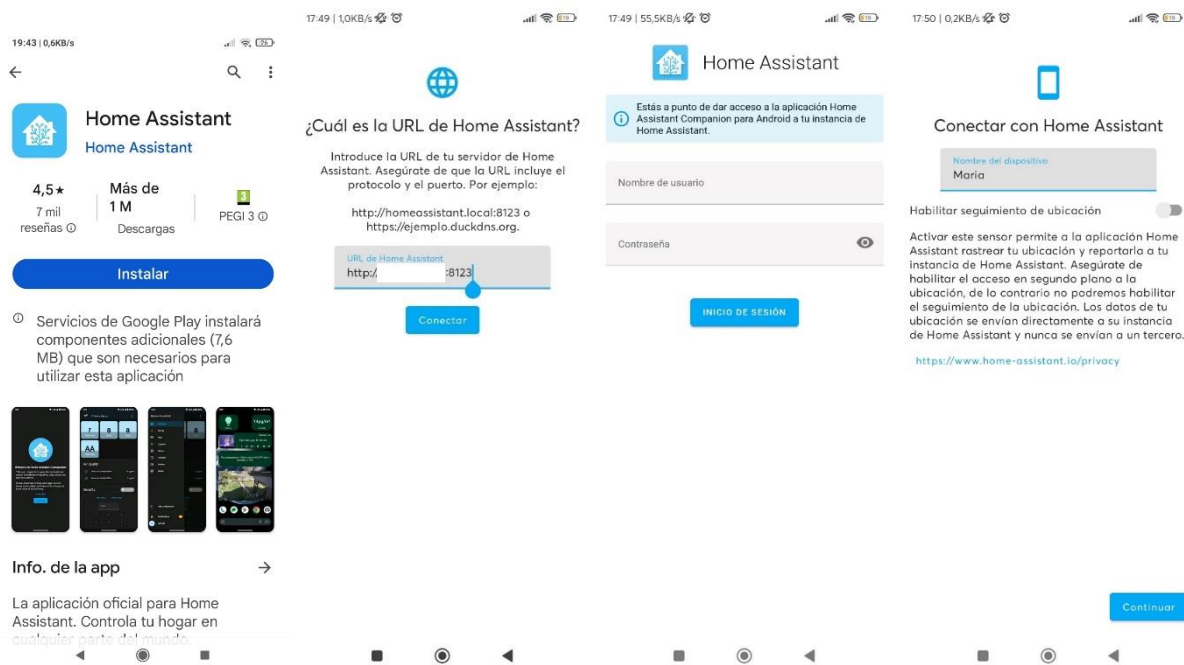


Figura 6.2.5.1: Instalación app móvil

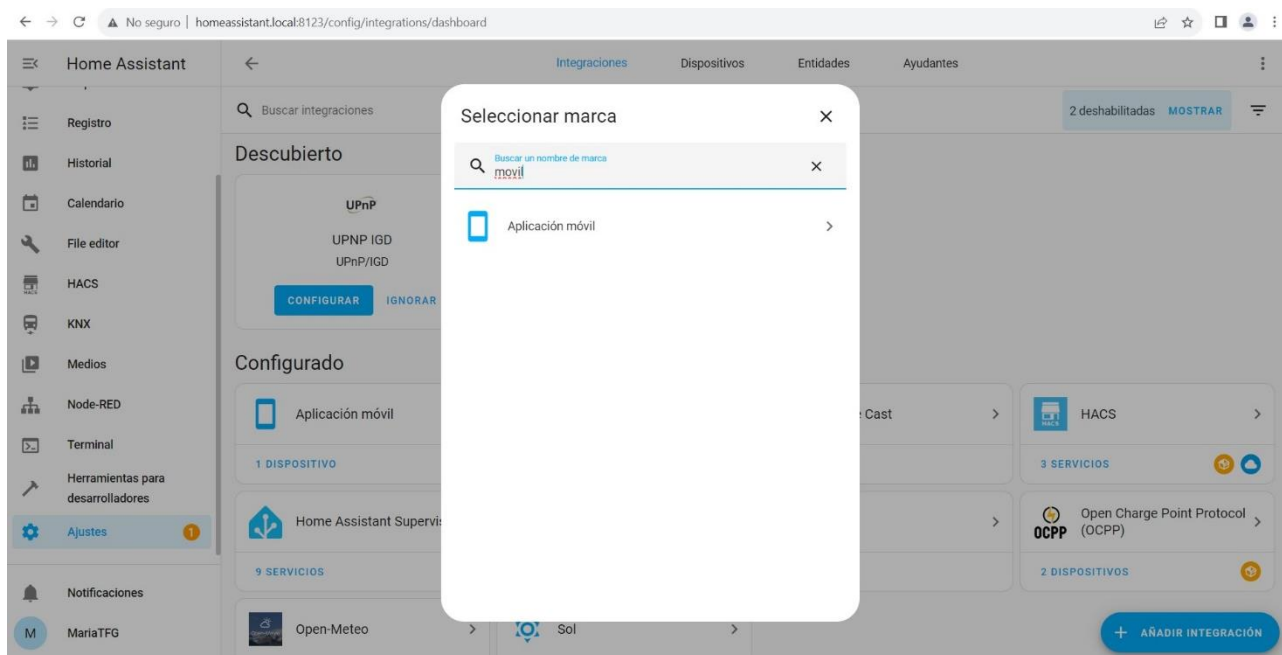


Figura 6.2.5.2: Instalación app móvil

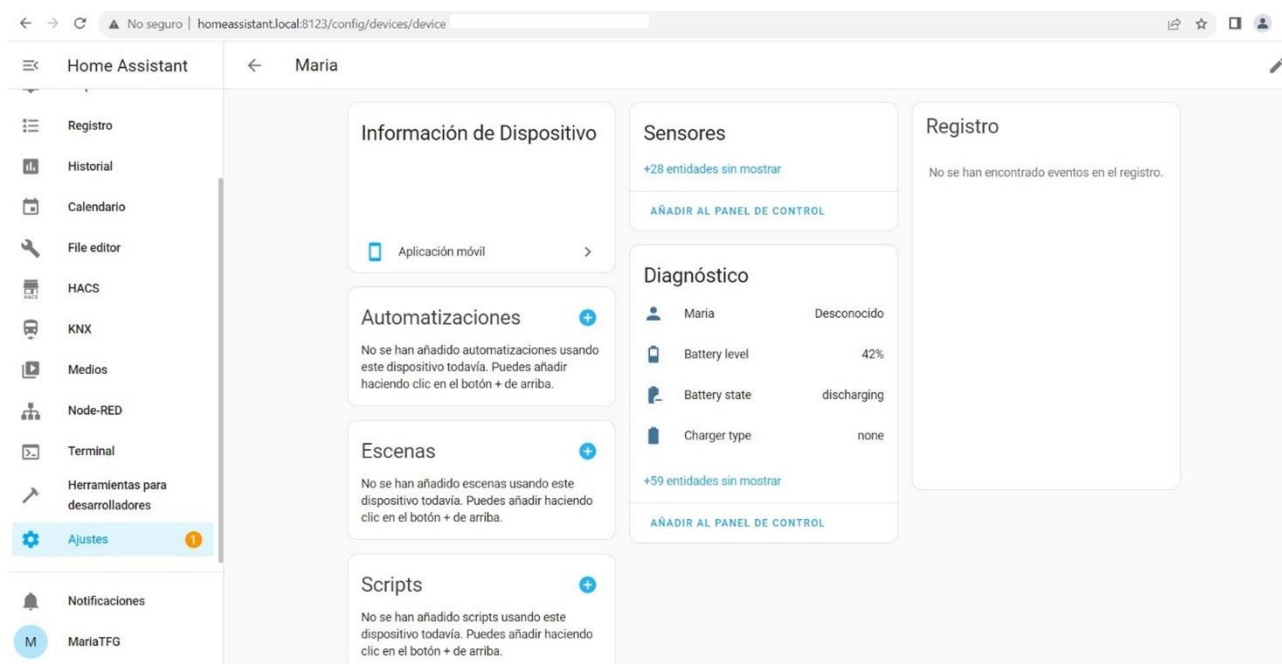


Figura 6.2.5.3: Instalación app móvil

6.2.6. Integración de KNX

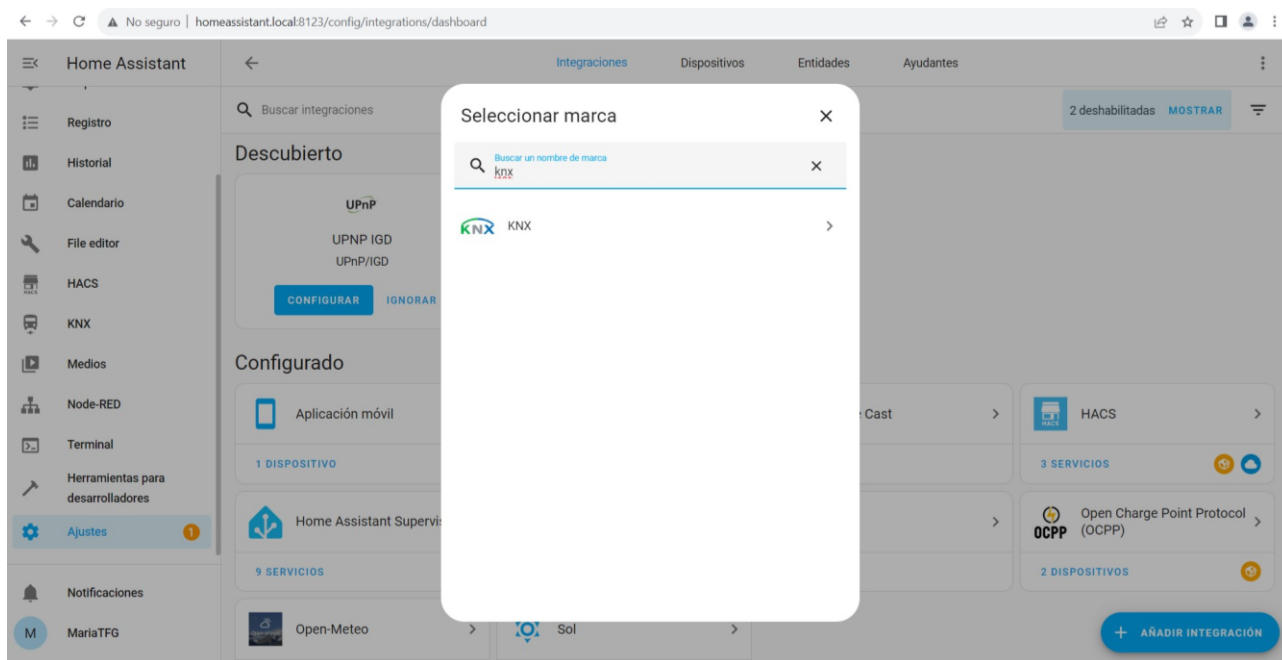


Figura 6.2.6.1: Cohesión KNX

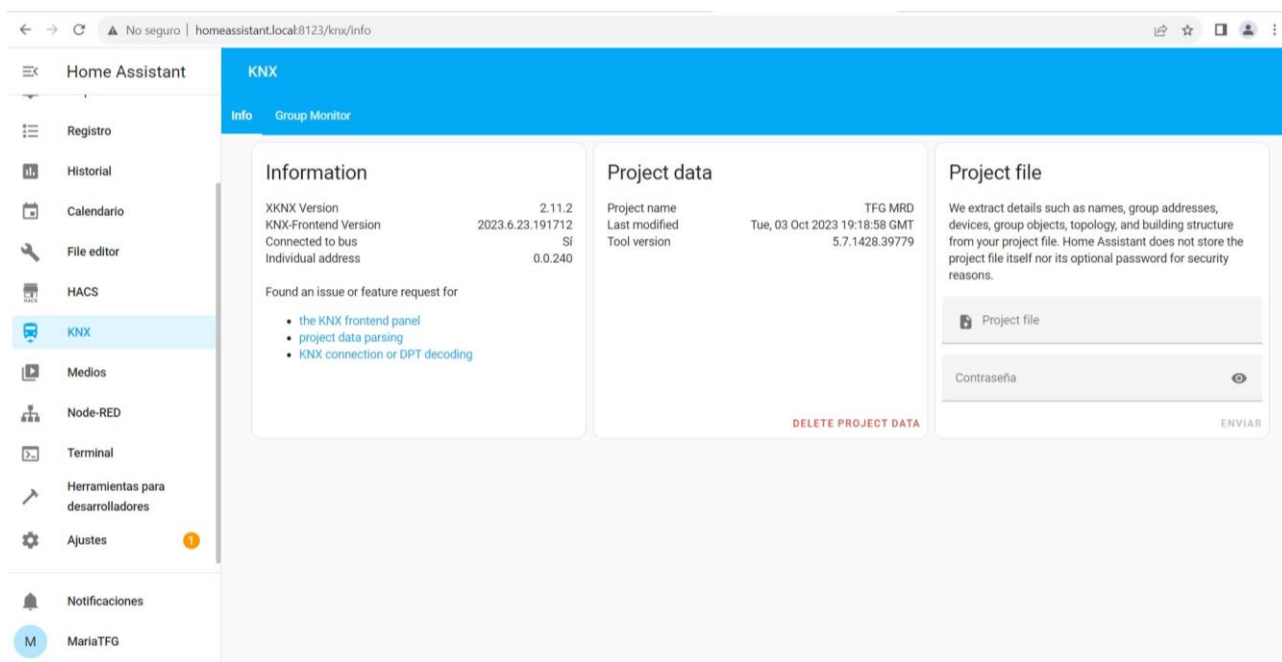
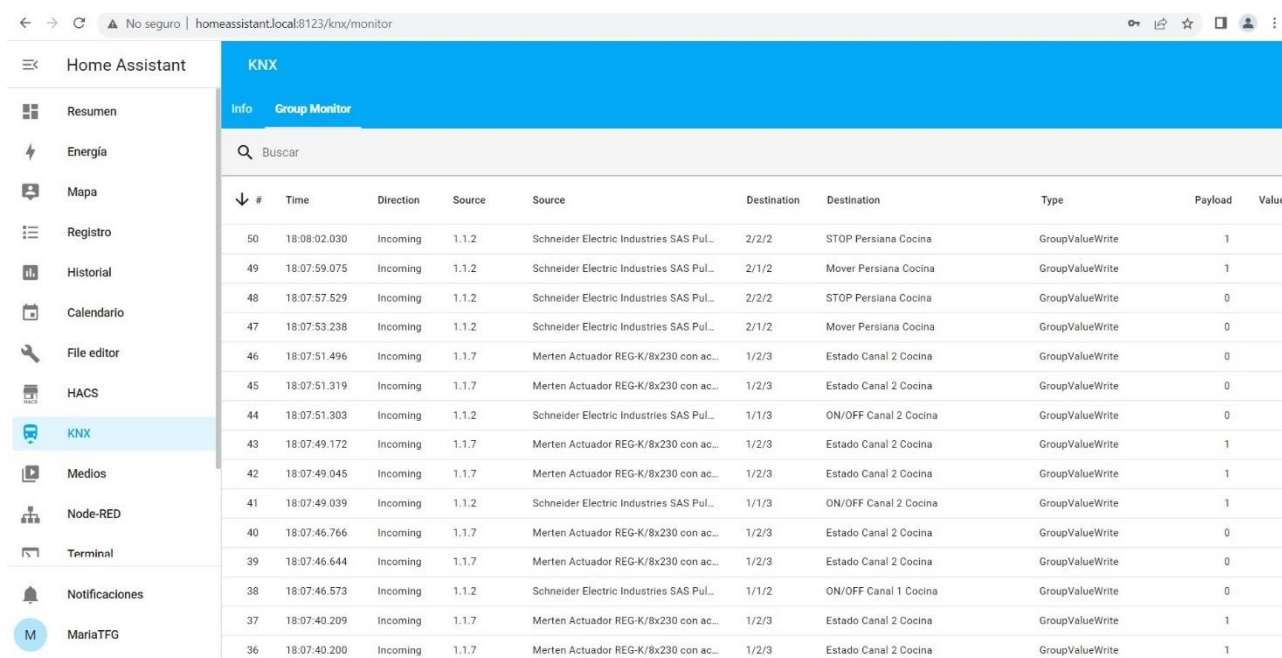


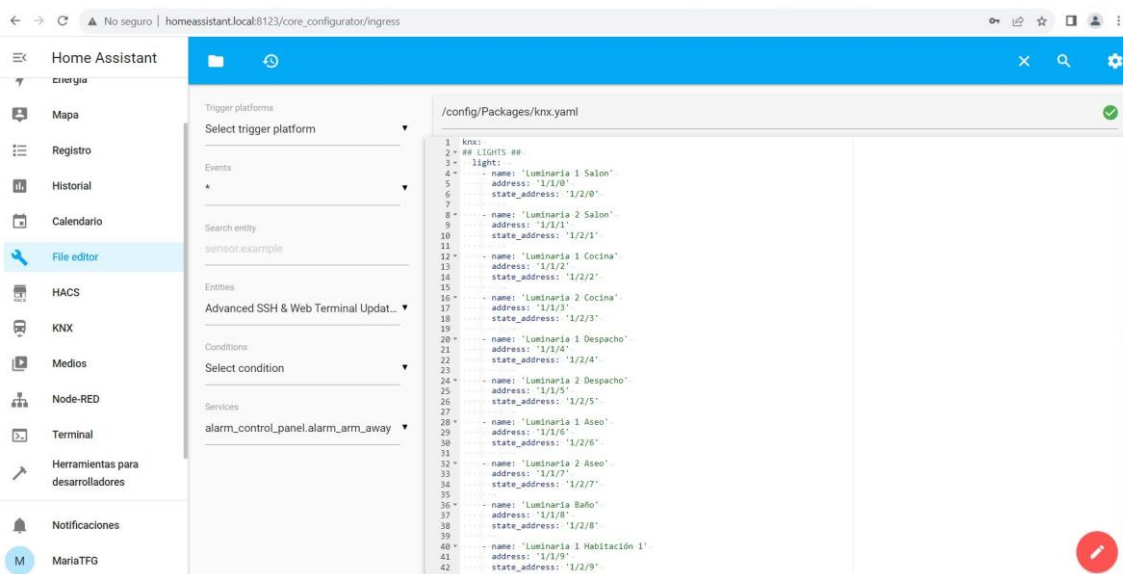
Figura 6.2.6.2: Cohesión KNX



The screenshot shows the Home Assistant web interface with the 'KNX' tab selected. The 'Group Monitor' sub-tab is active, displaying a table of telegrams. The table has columns for ID, Time, Direction, Source, Destination, Type, Payload, and Value. The telegrams are listed in descending order of time.

#	Time	Direction	Source	Destination	Type	Payload	Value
50	18:08:02.030	Incoming	1.1.2	2/2/2	GroupValueWrite	1	
49	18:07:59.075	Incoming	1.1.2	2/1/2	GroupValueWrite	1	
48	18:07:57.529	Incoming	1.1.2	2/2/2	GroupValueWrite	0	
47	18:07:53.238	Incoming	1.1.2	2/1/2	GroupValueWrite	0	
46	18:07:51.496	Incoming	1.1.7	1/2/3	GroupValueWrite	0	
45	18:07:51.319	Incoming	1.1.7	1/2/3	GroupValueWrite	0	
44	18:07:51.303	Incoming	1.1.2	1/1/3	GroupValueWrite	0	
43	18:07:49.172	Incoming	1.1.7	1/2/3	GroupValueWrite	1	
42	18:07:49.045	Incoming	1.1.7	1/2/3	GroupValueWrite	1	
41	18:07:49.039	Incoming	1.1.2	1/1/3	GroupValueWrite	1	
40	18:07:46.766	Incoming	1.1.7	1/2/3	GroupValueWrite	0	
39	18:07:46.644	Incoming	1.1.7	1/2/3	GroupValueWrite	0	
38	18:07:46.573	Incoming	1.1.2	1/1/2	GroupValueWrite	0	
37	18:07:40.209	Incoming	1.1.7	1/2/3	GroupValueWrite	1	
36	18:07:40.200	Incoming	1.1.7	1/2/3	GroupValueWrite	1	

Figura 6.2.6.3: Cohesión KNX. Telegramas



The screenshot shows the Home Assistant web interface with the 'File editor' tab selected. The file being edited is '/config/Packages/knx.yaml'. The file contains a list of light entities configured for the KNX system.

```

1 knx:
2   ## LIGHTS ##
3   light:
4     - name: 'Luminaria 1 Salon'
5       address: '1/1/0'
6       state_address: '1/2/0'
7
8     - name: 'Luminario 2 Salon'
9       address: '1/1/1'
10      state_address: '1/2/1'
11
12    - name: 'Luminaria 1 Cocina'
13      address: '1/1/2'
14      state_address: '1/2/2'
15
16    - name: 'Luminaria 2 Cocina'
17      address: '1/1/3'
18      state_address: '1/2/3'
19
20    - name: 'Luminaria 1 Despacho'
21      address: '1/1/4'
22      state_address: '1/2/4'
23
24    - name: 'Luminaria 2 Despacho'
25      address: '1/1/5'
26      state_address: '1/2/5'
27
28    - name: 'Luminaria 1 Aseo'
29      address: '1/1/6'
30      state_address: '1/2/6'
31
32    - name: 'Luminaria 2 Aseo'
33      address: '1/1/7'
34      state_address: '1/2/7'
35
36    - name: 'Luminaria Baño'
37      address: '1/1/8'
38      state_address: '1/2/8'
39
40    - name: 'Luminaria 1 Habitación 1'
41      address: '1/1/9'
42      state_address: '1/2/9'
43

```

Figura 6.2.6.4: Archivo de configuración YAML

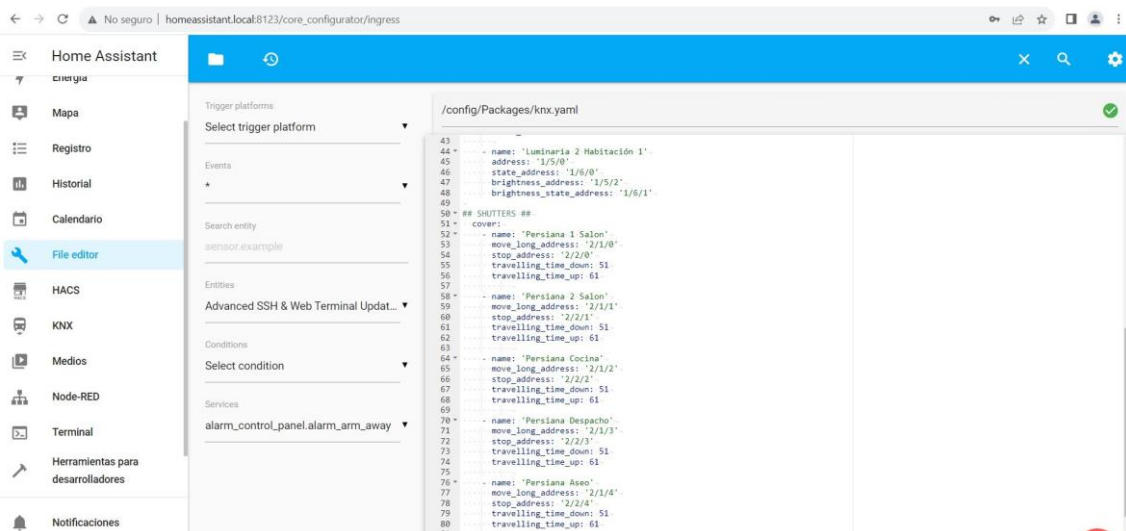


Figura 6.2.6.5: Archivo de configuración YAML



Figura 6.2.6.6: Archivo de configuración YAML

6.2.7. Integración de Arduino

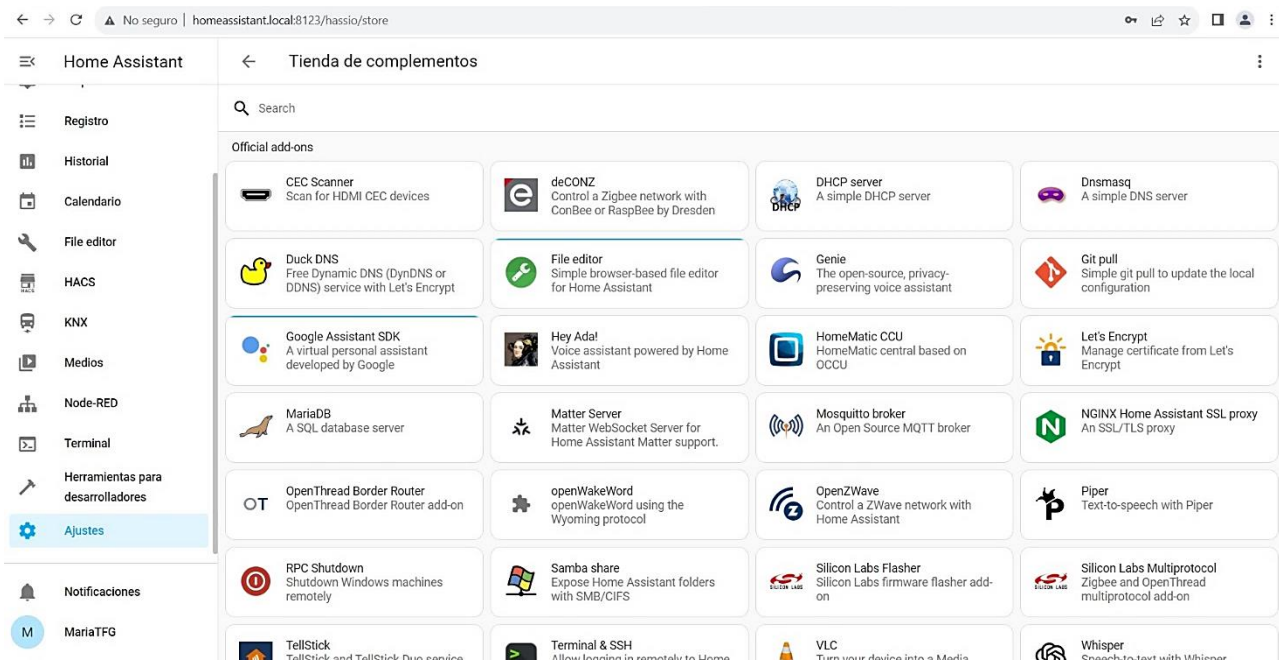


Figura 6.2.7.1: Tienda de complementos

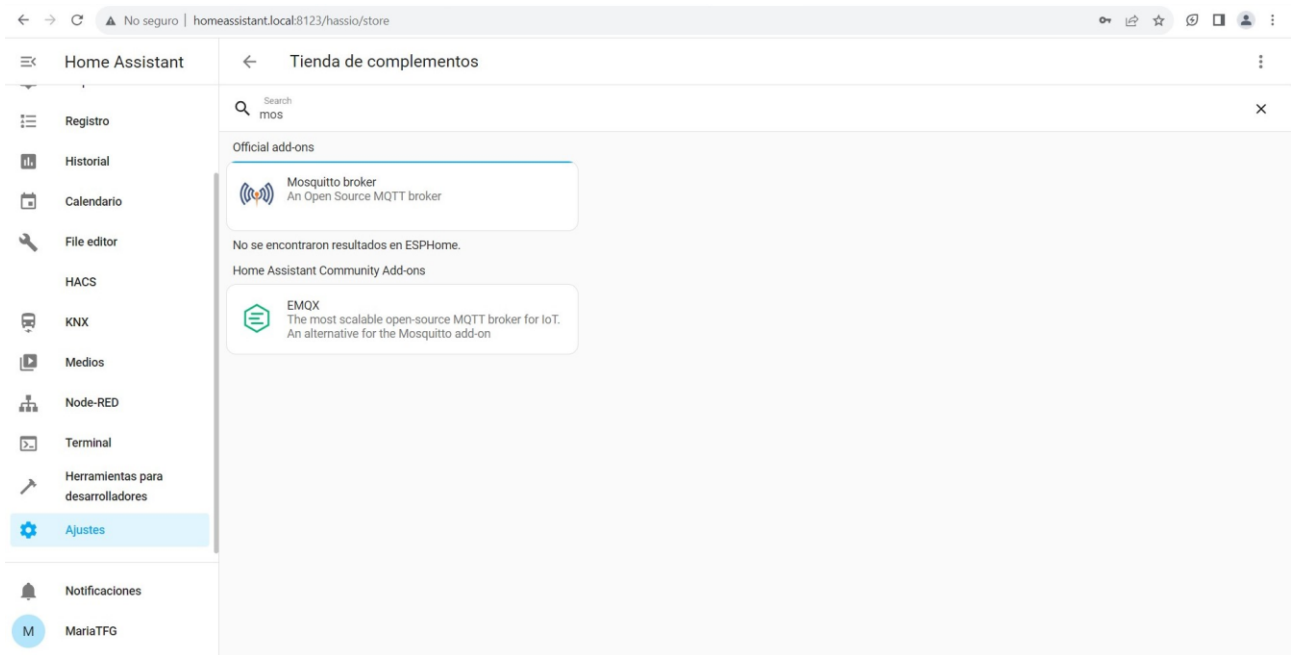


Figura 6.2.7.2: Mosquitto broker

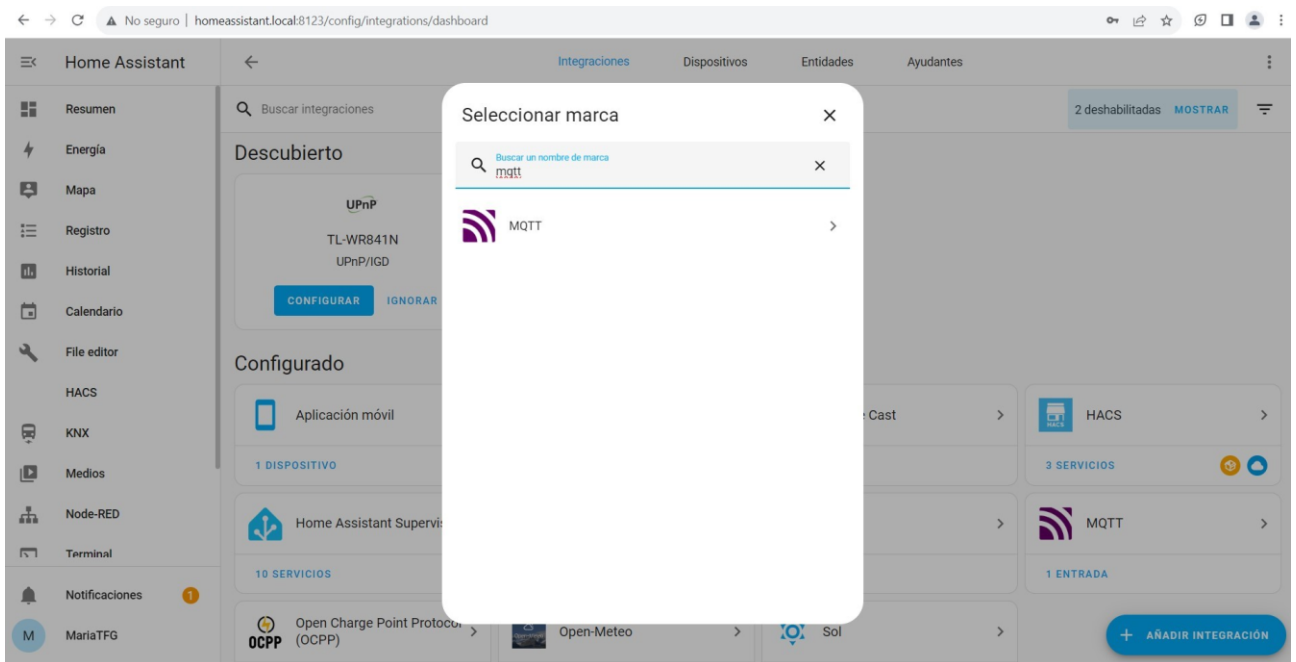


Figura 6.2.7.3: Integración MQTT

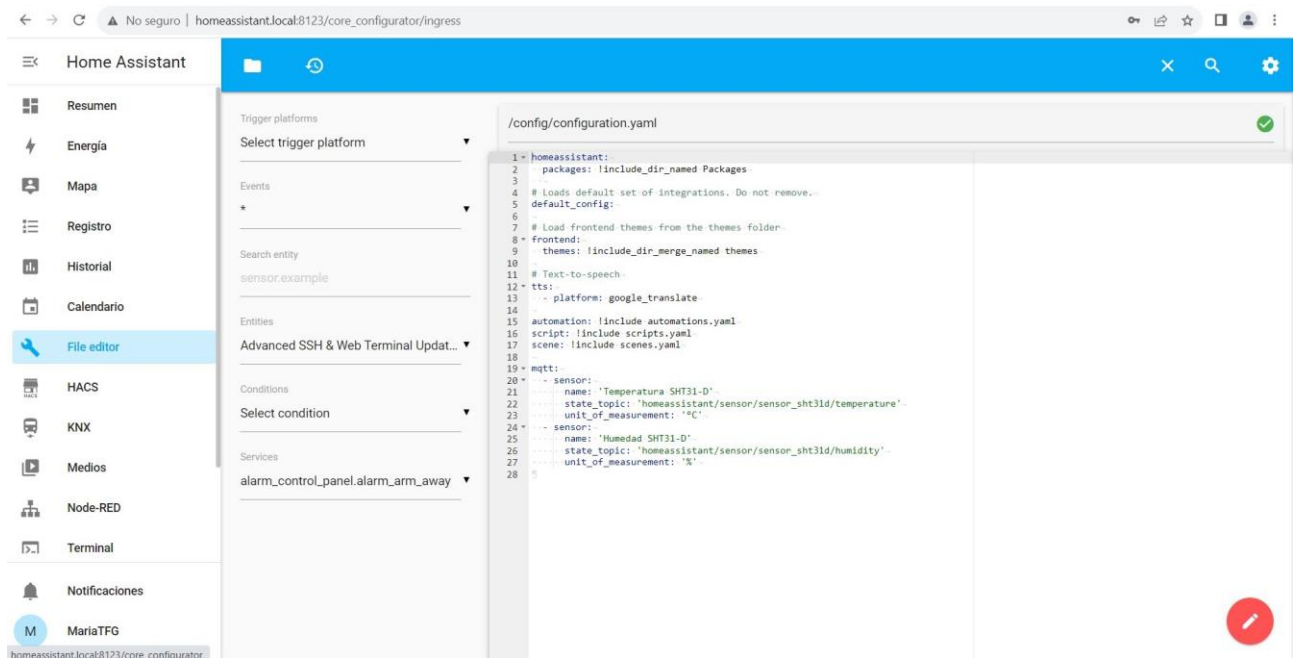


Figura 6.2.7.4: Archivo de configuración YAML

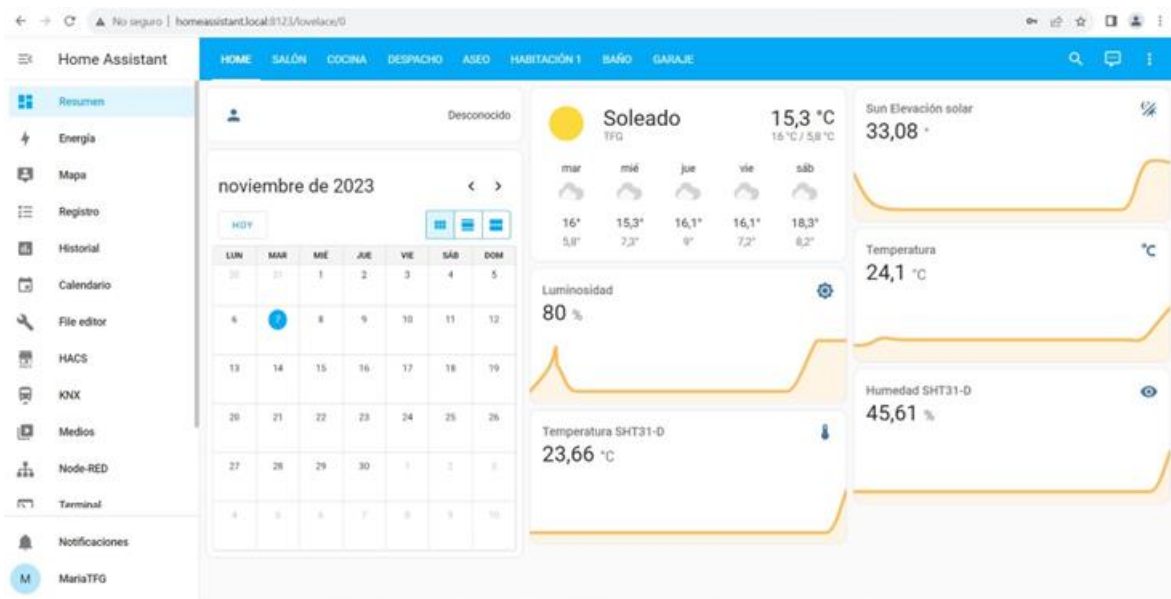


Figura 6.2.7.5: Interfaz de Home Assistant

6.2.8. Configuración de pantalla

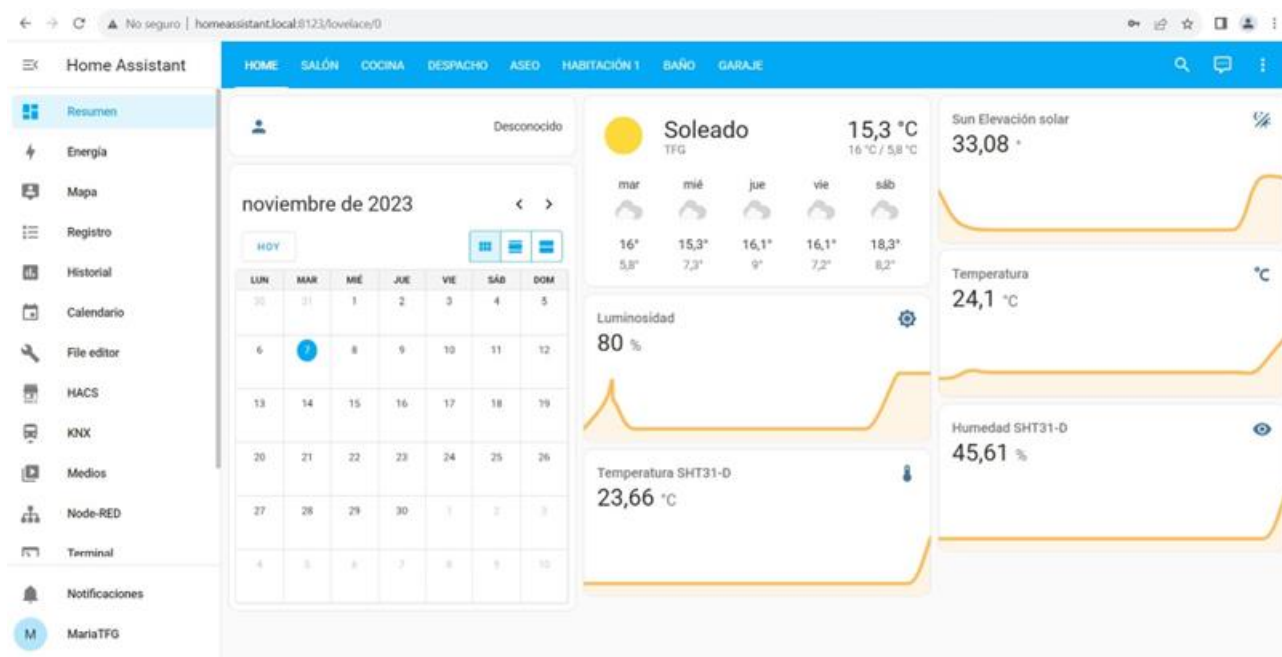


Figura 6.2.8.1: Interfaz de Home Assistant

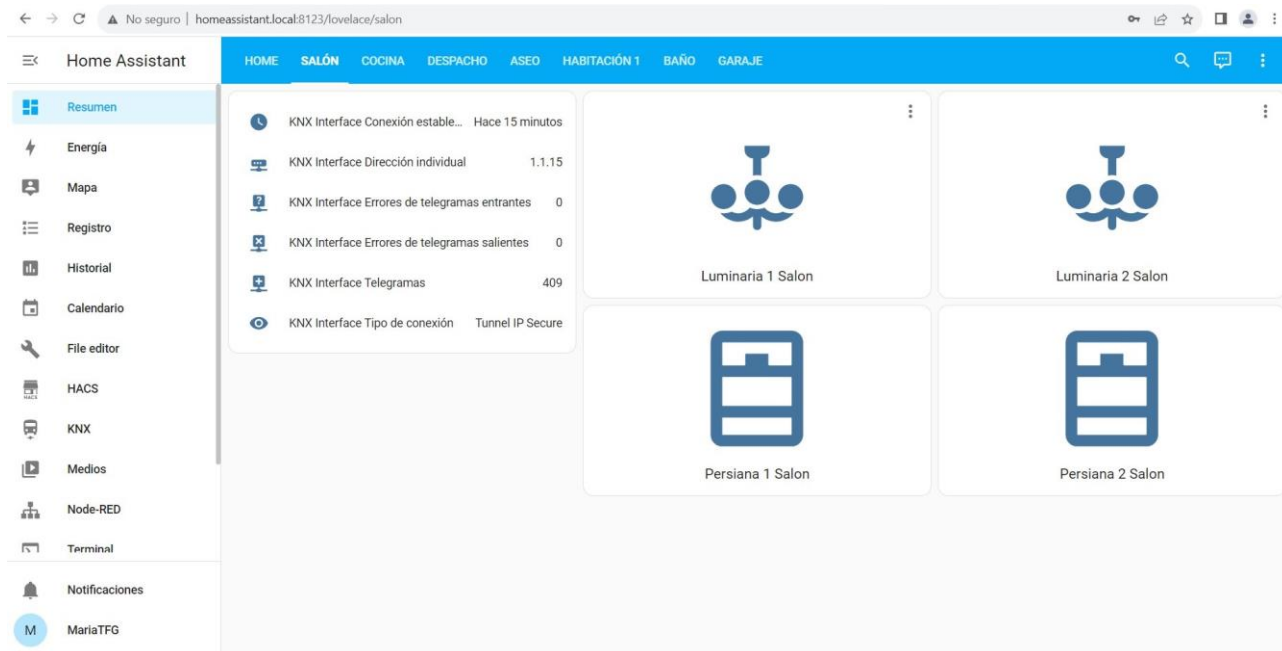


Figura 6.2.8.2: Interfaz de Home Assistant

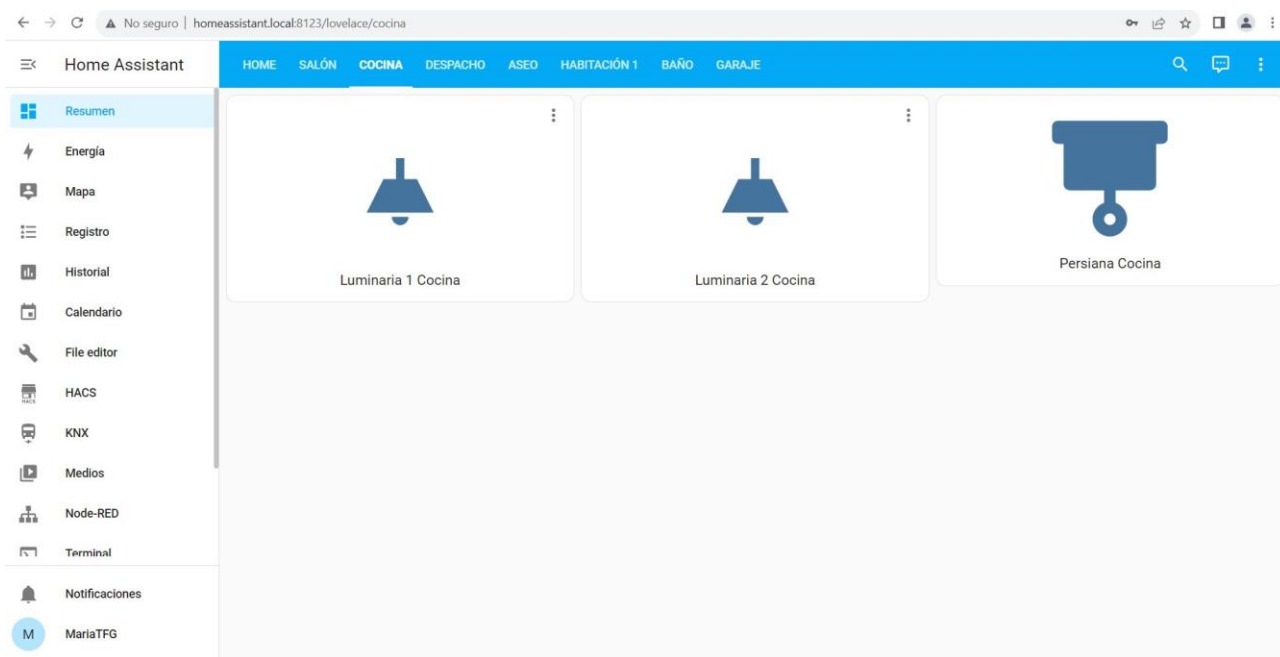


Figura 6.2.8.3: Interfaz de Home Assistant

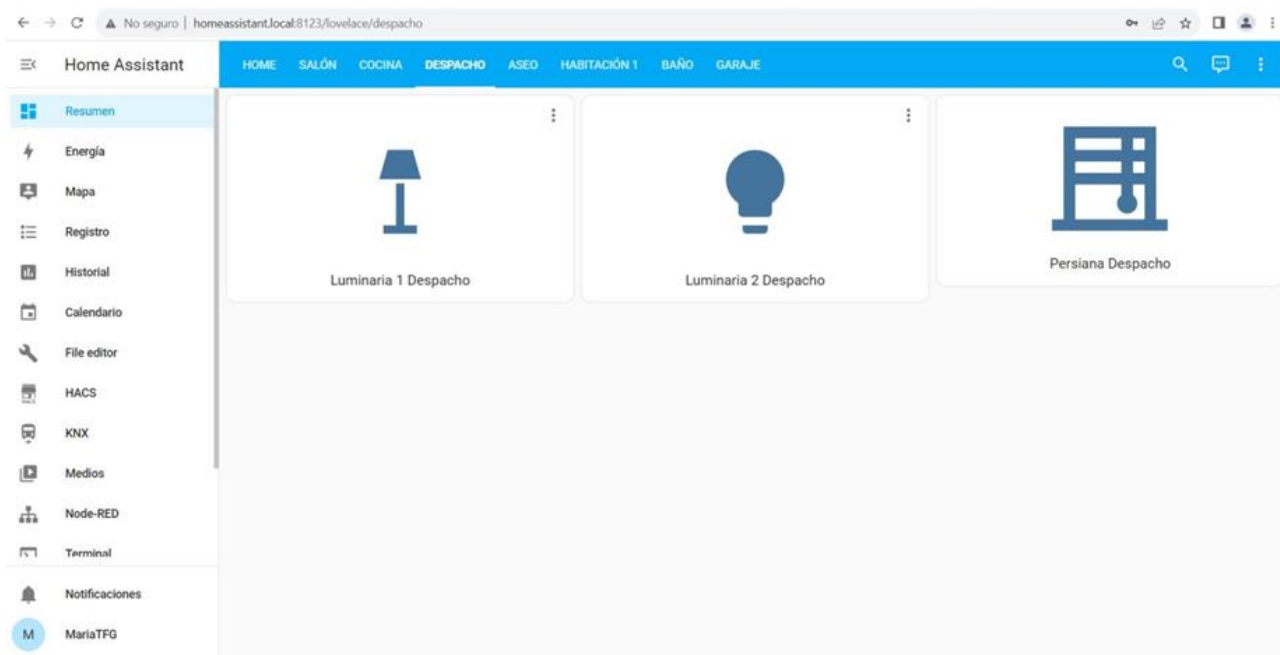


Figura 6.2.8.4: Interfaz de Home Assistant

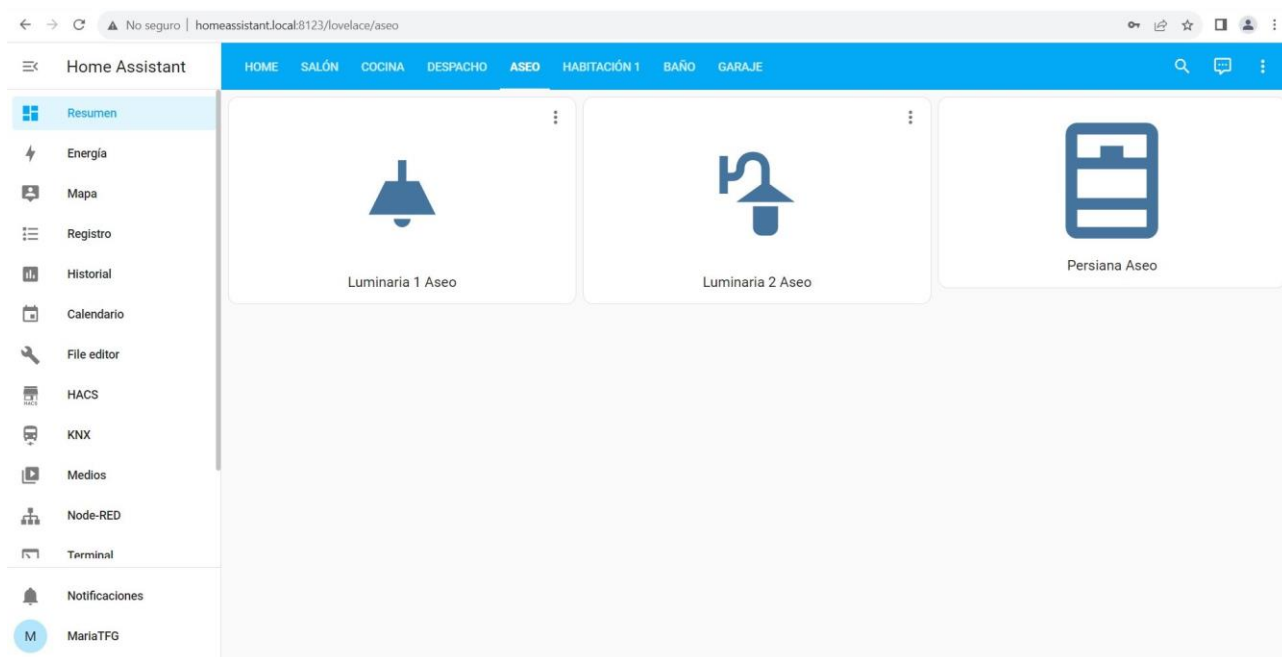


Figura 6.2.8.5: Interfaz de Home Assistant

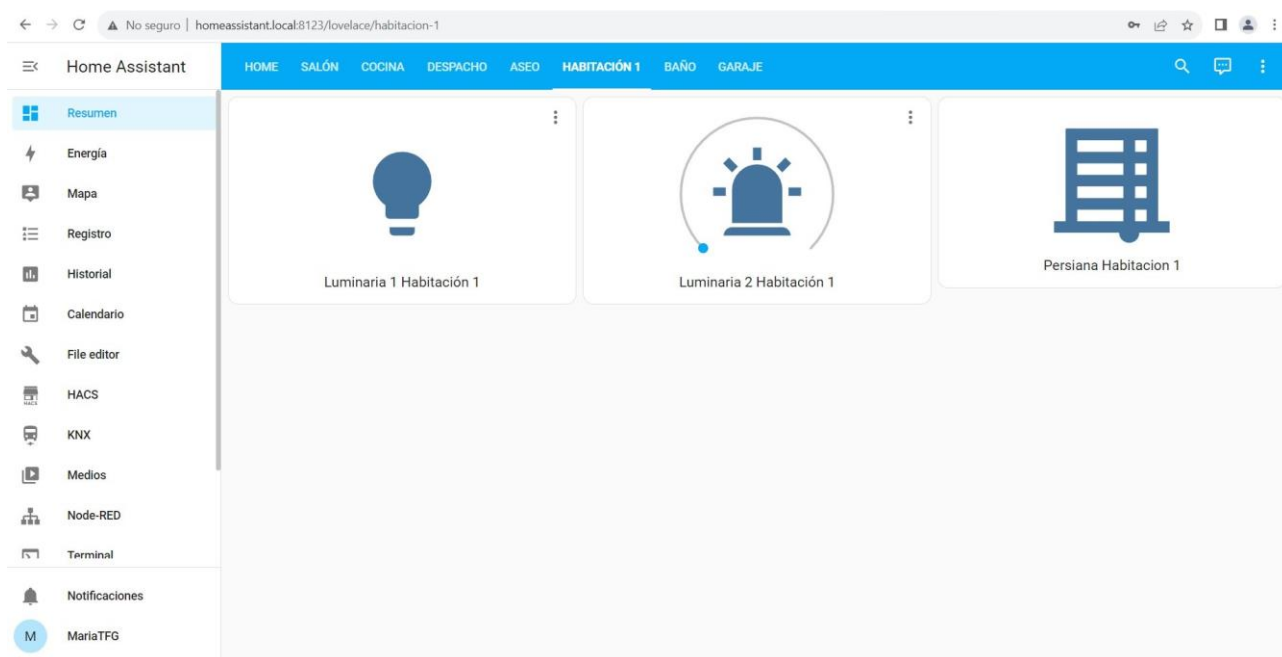


Figura 6.2.8.6: Interfaz de Home Assistant

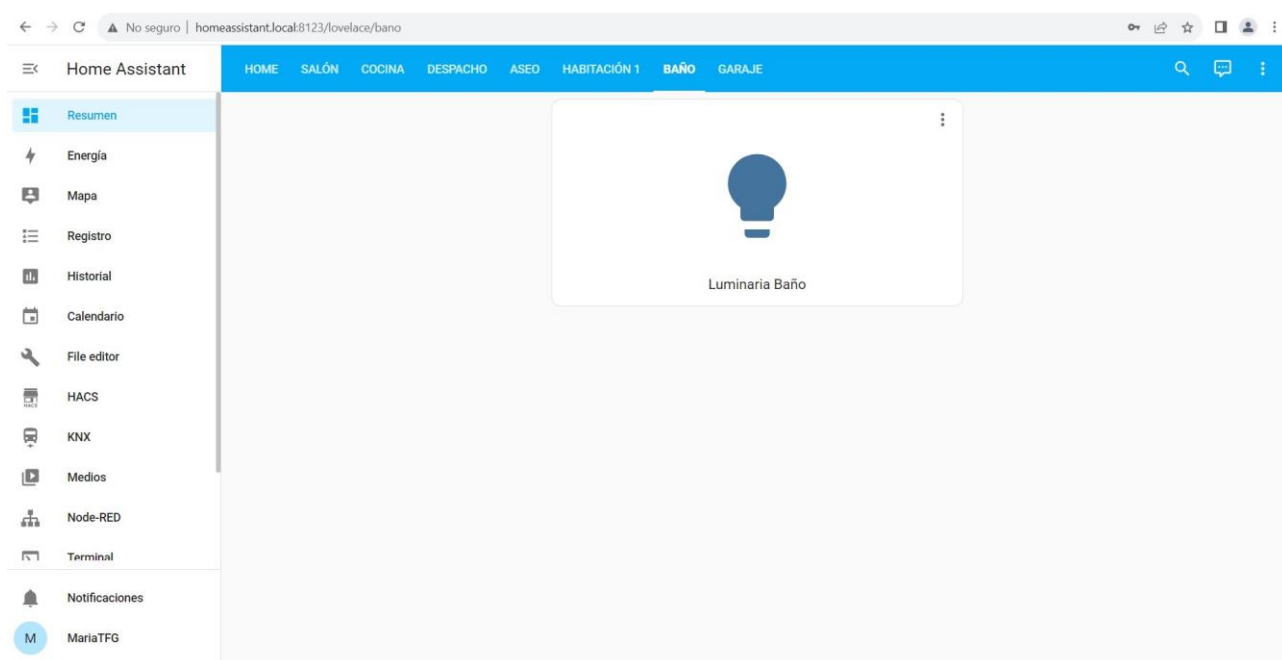


Figura 6.2.8.7: Interfaz de Home Assistant

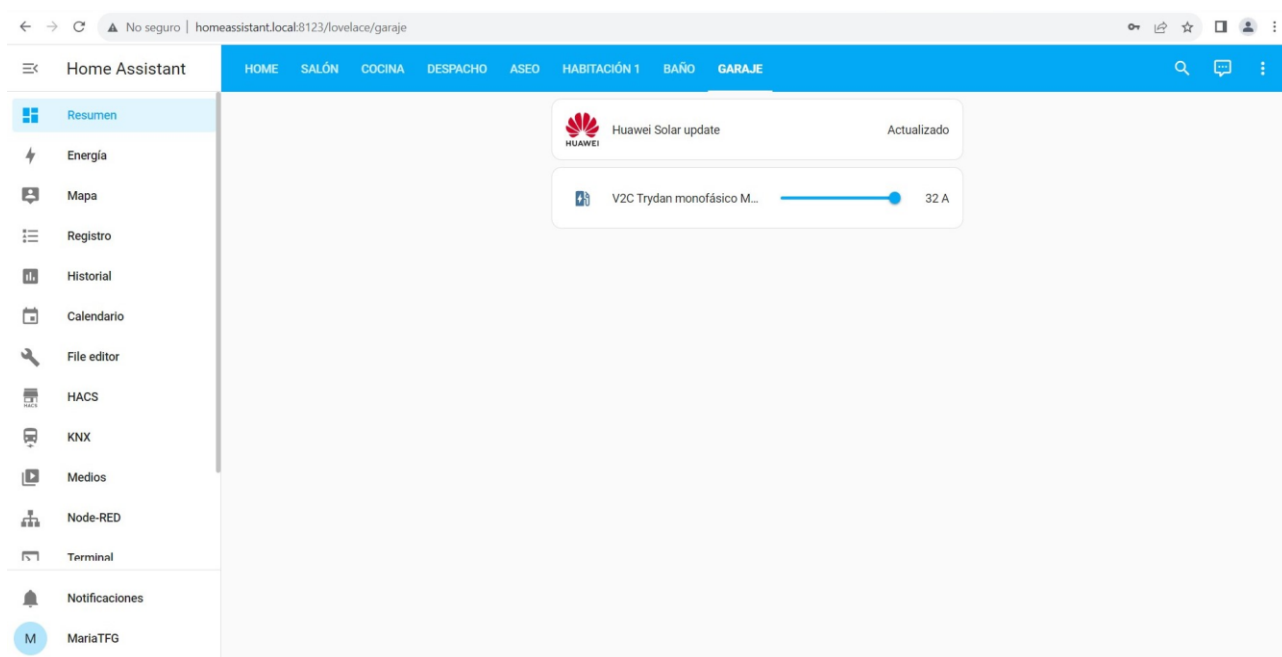


Figura 6.2.8.8: Interfaz de Home Assistant

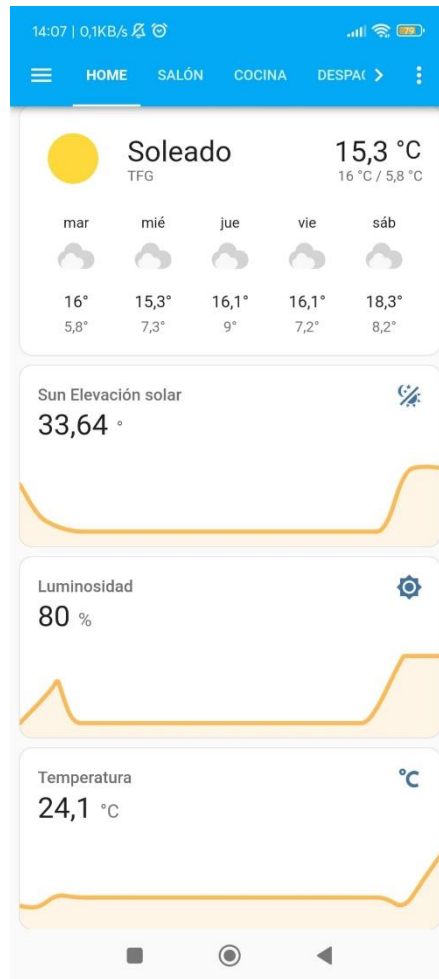


Figura 6.2.8.9: App móvil de Home Assistant

6.2.9. Ejemplos de funcionamiento

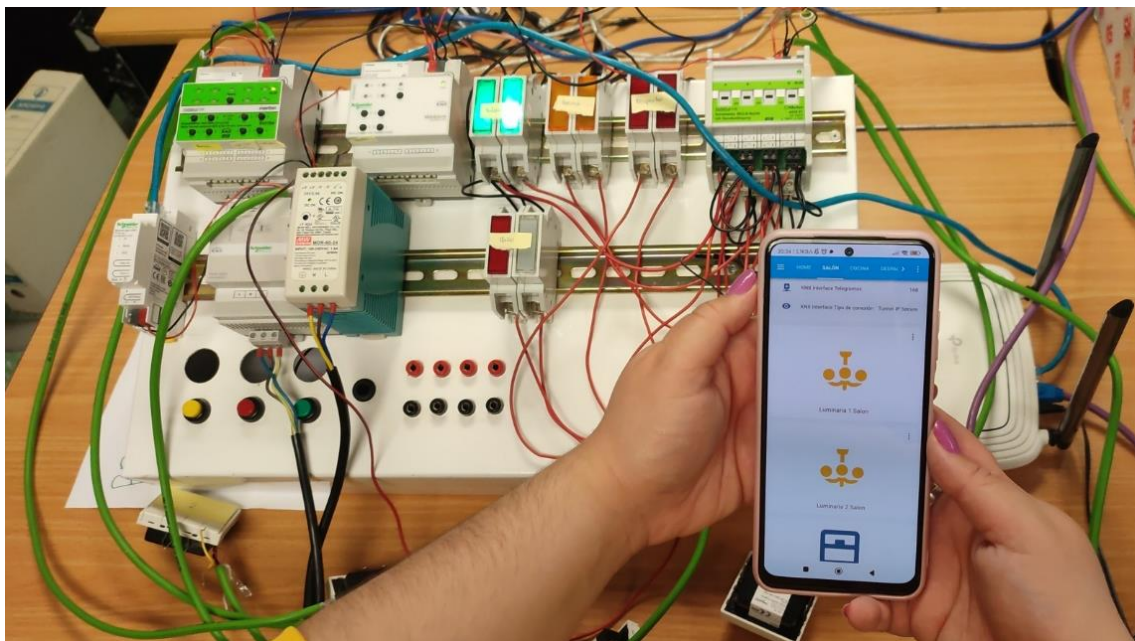


Figura 6.2.9.1: Funcionamiento

6.3. COHESIÓN CON ARDUINO

6.3.1. Código de Arduino

El código de Arduino que permite enviar los valores de temperatura y humedad del sensor al bróker MQTT en Home Assistant a través de conexión Wi-Fi es el siguiente:



```

1 #include <dire.h>
2 #include <Adafruit_SHT31.h>
3 #include <WiFi.h>
4 #include <PubSubClient.h>
5
6 #define BROKER_ADDR           // Dirección IP bróker MQTT
7 #define BROKER_PORT          1883
8 #define BROKER_USERNAME
9 #define BROKER_PASSWORD
10 #define SENSOR_NAME          "sensor_sht31d"
11 #define TEMPERATURE_TOPIC    "homeassistant/sensor/sensor_sht31d/temperature"
12 #define HUMIDITY_TOPIC        "homeassistant/sensor/sensor_sht31d/humidity"
13
14 char ssid[] =                // Nombre de red WiFi
15 char pass[] =                // Contraseña de red WiFi
16
17 byte mac[] = {0x00, 0x10, 0xFA, 0x6E, 0x3B, 0xAA};
18
19 WiFiClient wifiClient;
20 PubSubClient mqttClient(wifiClient);
21
22 Adafruit_SHT31 sht31 = Adafruit_SHT31();
23
24 void setup() {
25   Serial.begin(9600);
26   delay(10);
27
28   // Inicialización del sensor SHT31
29   if (!sht31.begin(0x44)) {
30     Serial.println("No se encontró el sensor SHT31");
31     while (1);
32   }
33

```

Figura 6.3.1.1: Código Arduino



```

33
34   // Conexión a WiFi
35   Serial.println();
36   Serial.println("Conectando a WiFi");
37   WiFi.begin(ssid, pass);
38   while (WiFi.status() != WL_CONNECTED) {
39     delay(500);
40     Serial.print(".");
41   }
42   Serial.println("Conectado a la red WiFi");
43
44   // Configuración MQTT
45   mqttClient.setServer(BROKER_ADDR, BROKER_PORT);
46   mqttClient.setCallback(callback);
47
48   // Conexión al bróker MQTT
49   if (mqttClient.connect(SENSOR_NAME, BROKER_USERNAME, BROKER_PASSWORD)) {
50     Serial.println("Conectado al bróker MQTT");
51   } else {
52     Serial.println("Fallo al conectar al bróker MQTT");
53     while (1);
54   }
55 }
56

```

Figura 6.3.1.2: Código Arduino


```
56
57 void loop() {
58   mqttClient.loop();
59
60   // Leer valor del sensor
61   float tempValue = sht31.readTemperature();
62   float humValue = sht31.readHumidity();
63
64   // Enviar el valor a Home Assistant
65   mqttClient.publish(TEMPERATURE_TOPIC, String(tempValue).c_str());
66   mqttClient.publish(HUMIDITY_TOPIC, String(humValue).c_str());
67
68   Serial.println("Datos enviados a Home Assistant");
69
70   delay(15000); // Esperar 15 segundos antes de enviar otro valor
71 }
72
73 void callback(char* topic, byte* payload, unsigned int length) {
74   // Función de callback, procesar mensajes recibidos desde el broker
75 }
76
```

Figura 6.3.1.3: Código Arduino

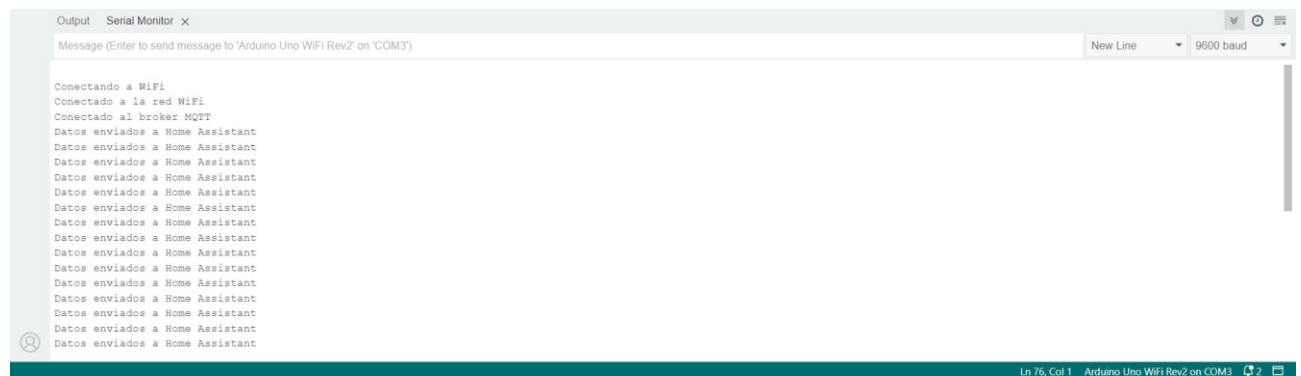


Figura 6.3.1.4: Código Arduino

6.3.2. Ejemplos de funcionamiento

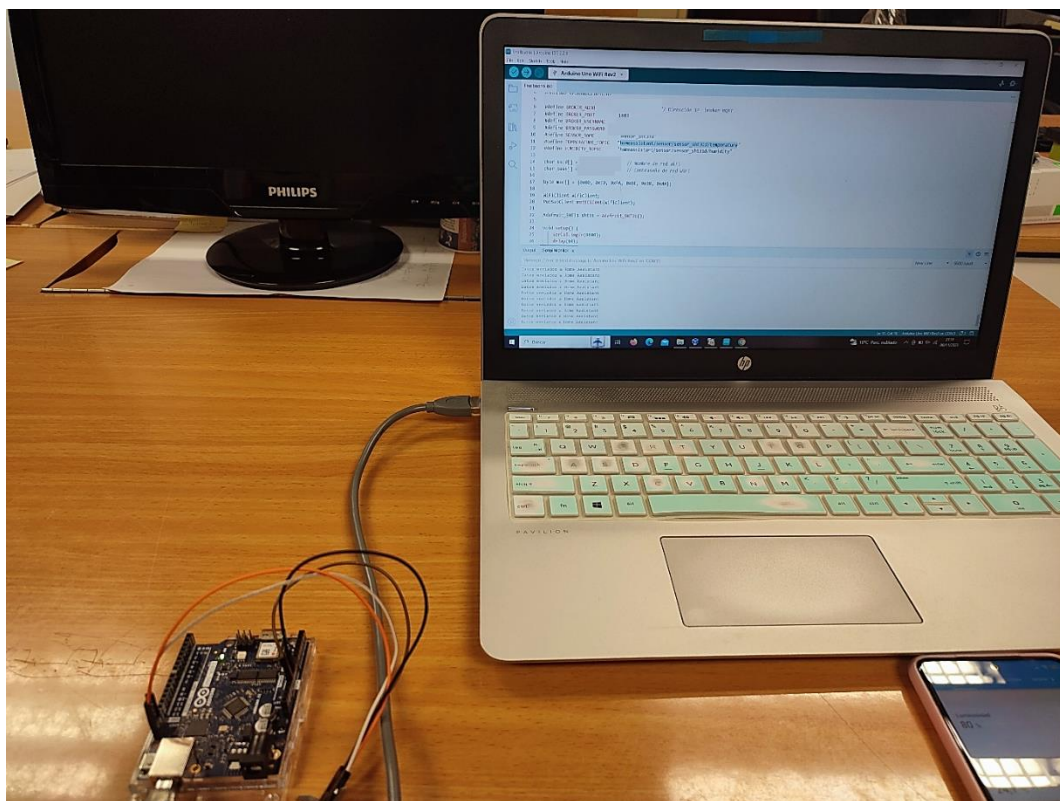


Figura 6.3.2.1: Arduino a través de MQTT

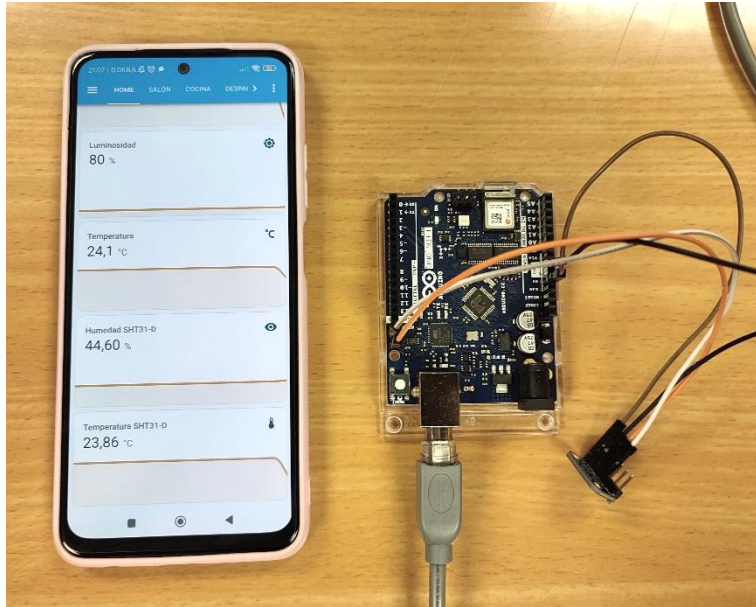


Figura 6.3.2.2: Arduino a través de MQTT

6.4. ENLACE A FOTOS

Todas las imágenes del proyecto se pueden ver a través del siguiente código QR:



Figura 6.3.2.1: Código QR fotos

6.5. ENLACE A VÍDEOS

Todos los vídeos del funcionamiento del proyecto se pueden ver a través del siguiente código QR:



Figura 6.3.2.1: Código QR vídeos