



industriales
etsii

**Escuela Técnica
Superior
de Ingeniería
Industrial**

UNIVERSIDAD POLITÉCNICA DE CARTAGENA

**Escuela Técnica Superior de Ingeniería
Industrial**

**Diseño, prototipado e integración de sensores
de bajo coste en un sistema domótico
heterogéneo orientado al cuidado de personas
mayores**

TRABAJO FIN DE GRADO

**GRADO EN INGENIERÍA ELECTRÓNICA INDUSTRIAL Y
AUTOMÁTICA**

Autor: Natalia Sáez Pérez
Director: Dr. José Alfonso Vera Repullo
Codirector: Dr. Manuel Jiménez Buendía

Cartagena, 6 de septiembre de 2022



**Universidad
Politécnica
de Cartagena**

AGRADECIMIENTOS:

Me gustaría agradecer a los profesores José Alfonso Vera Repullo y Manuel Jiménez Buendía por darme la oportunidad de trabajar desarrollando este proyecto.

Por otra parte, les doy las gracias a mi familia y amigos más cercanos por haber estado conmigo durante esta etapa de la vida.

Gracias.

RESUMEN

En los últimos años, el número de personas pertenecientes al grupo de edad de mayores de 65 años ha incrementado y con ello la necesidad de estudiar métodos de mejora de la calidad de vida con la tecnología desarrollada para que este grupo mantenga la independencia del hábito cotidiano pero estudiado por los cuidadores y familia que puedan estar pendientes de él. Por otra parte, a raíz de la aparición de la enfermedad COVID-19 la tendencia a la soledad de este grupo de mayores aumentó y con ello una caída del bienestar mental por lo que estos métodos empezaron a abarcar también el estudio del estado de ánimo de las personas.

Este proyecto propone un sistema domótico basado en sensores de bajo coste que permite monitorizar la vida cotidiana de las personas mayores para posteriores estudios psicológicos y brindando así la independencia que quieren mantener.

Este sistema domótico está integrado en un microordenador o SBC (Single Board Computer) que posee una plataforma de automatización del hogar. Por otro lado, los sensores de bajo coste irán conectados a un System of a Chip (SoC) que hará de puente entre la plataforma y los sensores para su lectura y, posterior, visualización de los parámetros. La selección de los sensores de bajo se encarga de monitorizar parámetros como la temperatura y uso de la vitrocerámica o la presencia y uso del sofá.

Incluye el diseño y prototipado de una placa de circuito impreso para implementar todos estos sensores directamente conectador al SoC así como del uso de gestores de bases de datos para visualizar los parámetros obtenidos.

ABSTRACT

In recent years, the number of people belonging to the over 65 age group has increased and with it the need to study methods of improving the quality of life with the developed technology so that this group maintains the independence of the daily habit but studied by the caregivers and family who can take care of them. On the other hand, following the appearance of the COVID-19 disease, the tendency towards loneliness in this group of elderly people increased and with it a drop in mental wellbeing, so these methods also began to include the study of people's state of mind.

This project proposes a low-cost sensor-based home automation system to monitor the daily life of the elderly people for further psychological studies, thus providing the independence they want to maintain.

This home automation system is integrated into a microcomputer that has a home automation platform. On the other hand, the low-cost sensors will be connected to a System of a Chip (SoC) that will act as a bridge between the platform and the sensors for reading and, subsequently, displaying the parameters.

The selection of low-cost sensors is responsible for monitoring parameters such as the temperature and use of the glass ceramic hob or the presence and use of the sofa.

It includes the design and prototyping of a printed circuit board to implement all these sensors directly connected to the SoC as well as the use of database managers to visualise the parameters obtained.

Contenido:

Índice de figuras	11
Índice de tablas	15
1. Introducción	17
1.1. Situación	17
1.2. Objetivos	18
2. Estado del Arte	19
2.1. Desarrollo hasta la actualidad de Smart Home y AAL	19
2.2. Inmótica	24
2.3. Domótica	24
2.4. Estándares	26
2.4.1. X-10	26
2.4.2. KNX	27
2.4.3. LonWorks	28
2.4.4. Z-Wave	29
2.4.5. ZigBee	29
2.4.6. Wi-Fi	31
2.5. Smart-Home	32
3. Materiales	34
3.1. Raspberry Pi	34
3.2. ESP32	35
3.3. Sensores de bajo coste	38
3.3.1. Termopar tipo K	38
3.3.2. MAX6675	39
3.3.3. Células de carga	39
3.3.4. HX711	41
3.3.5. EFS-04P	42
3.3.6. Sensor de presión de asiento	42
3.4. Home-Assistant	43
3.5. ESPHome	45
3.6. Diptrace	45
4. Métodos	47
4.1. Instalación y configuración Home Assistant	47
4.2. Instalación y configuración de la herramienta ESPHome y ESP32	49
4.3. Conexión de sensores	58
4.3.1. Células de carga y HX711	58
4.3.2. Termopar tipo K y MAX6675	59
4.3.3. Sensores binarios	60
4.4. Configuración y programación de sensores	61
4.4.1. HX711	61
4.4.2. MAX6675	63
4.4.3. Sensores binarios	65
4.5. Interfaz Home Assistant	69
4.6. Interfaz Grafana e InfluxDB	72

4.7. Desarrollo de PCB	75
4.7.1. PCB con módulos	78
4.7.2. PCB con integrados	80
4.7.3. PCB con ADS1115 integrado	82
4.8. Definición de experimentos	83
5. Resultados	84
6. PRESUPUESTO Y TRABAJOS FUTUROS.	87
6.1. Presupuesto	87
6.1.1. Presupuesto con módulos	87
6.1.2. Presupuesto con integrados	88
6.2. Trabajos futuros	88
7. Bibliografía	89
ANEXO: Programación ESPHome	92

Índice de figuras

Figura 1. Porcentaje de personas que viven solas según sexo y edad en el año 2020. [3]	17
Figura 2. Jim Sutherland sentado enfrente del ECHO IV junto a su familia. (1966).....	19
Figura 3. Evolución de la población hasta 2019 en España. [INE:1900-2011: Censos de población y vivienda. 2019]	20
Figura 4. Logo del proyecto HOPE. [9]	20
Figura 5. Arquitectura de EMUTEM [10].....	21
Figura 6. Sistema completo SIRMAVED. [11].....	21
Figura 7. Componentes de un sistema SMARTA. [13].....	22
Figura 8. Esquema Hardware del proyecto HIMTAE.....	23
Figura 9. Esquema de transmisión del protocolo X-10.	27
Figura 10. Logo KNX.....	27
Figura 11. Ejemplo de simbología KNX.	28
Figura 12. Topologías de red ZigBee.	30
Figura 13. Raspberry Pi 3 Model B.....	34
Figura 14. ESP32DevKit	36
Figura 15. ESP32-WROOM-32	36
Figura 16. Esquemático ESP32-WROOM-32.....	37
Figura 17. Esquema de periféricos ESP32-WROOM-32.....	37
Figura 18. Termopar tipo K.....	39
Figura 19. Módulo MAX6675	39
Figura 20. Células de carga	40
Figura 21. Montaje en cuarto de puente.	40
Figura 22. Montaje en medio puente.	40
Figura 23. Montaje en puente completo.	40
Figura 24. Módulo HX711	41
Figura 25. Diagrama de aplicación típica para el HX711	41
Figura 26. EFS-04P	42
Figura 27. Sensor de presión de asiento.	42
Figura 28. Dispositivos disponibles para la instalación de Home Assistant.[47].....	43
Figura 29. Comparación entre los distintos métodos de instalación de Home Assistant. [47]	44
Figura 30. Página principal ESPHome.	45
Figura 31. Menú principal DipTrace.	46
Figura 32. Dirección IP Home Assistant	47
Figura 33. Creación de cuenta de usuario en Home Assistant.	47
Figura 34. Instalación de servicios en Home Assistant,	48
Figura 35. Página principal Home Assistant	48
Figura 36. ESPHome obsoleto en máquina virtual con Home Assistant 7.6	49
Figura 37. Tienda de complementos Home Assistant 8.2	50
Figura 38. URL ESPHome	50
Figura 39. Tienda de complementos con ESPHome	51
Figura 40. Página de información ESPHome	51
Figura 41. Página de información ESPHome	52
Figura 42. Interfaz ESPHome.....	52
Figura 43. Elección nombre y modelo ESP32.....	53

Figura 44. Interfaz ESPHome.....	53
Figura 45. Código base ESPHome	53
Figura 46. Elección del método de instalación del programa en ESP32	54
Figura 47. Descarga del proyecto en formato .bin	54
Figura 48. Interfaz web para la programación.....	55
Figura 49. Selección puerto COM3 ESP32	55
Figura 50. Página Web de instalación	56
Figura 51. Instalación del programa en ESP32	56
Figura 52. Dirección IP ESP32.....	57
Figura 53. ESP32 de toda la vivienda integrados en ESPHome	57
Figura 54. Célula de carga.....	58
Figura 55. Conexión células de carga.....	58
Figura 56. Conexión HX711 a células de carga y ESP32	59
Figura 57. Conector termopar tipo K.	59
Figura 58. Conexión MAX6675 a ESP32	59
Figura 59. Conexión de sensores al ESP32	60
Figura 60. Conexión de sensores en PCB_integrados y PCB_módulos a ESP32.....	60
Figura 61. Programación básica en ESPHome de HX711	61
Figura 62. Programación HX711 en el dormitorio en ESPHome	62
Figura 63. Programación HX711.....	62
Figura 64. Programa de ocupación de HX711	62
Figura 65. Interfaz Home Assistant para HX711	62
Figura 66. Programación base para MAX6675	63
Figura 67. Programación MAX6675 en ESPHome	63
Figura 68. Interfaz en Home Assistant del módulo MAX6675.....	63
Figura 69. Programación uso vitrocerámica.....	64
Figura 70. Tarjeta de uso de vitrocerámica	64
Figura 71. Programación básica sensores binarios.....	65
Figura 72. Programación sensores binarios en salón-cocina en ESPHome	65
Figura 73. Menú File editor	66
Figura 74. Contador para frecuencia de uso del sillón	66
Figura 75. Cronómetro para tiempo de uso del sillón	67
Figura 76. Carpeta config/configuration.yaml	67
Figura 77. Interfaz en Home Assistant de los sensores binarios	67
Figura 78. Interfaz Salón Cocina con formato en horas del tiempo de uso.....	68
Figura 79. Interfaz Home Assistant área salón-cocina.	69
Figura 80. Configuración entidad Wi-Fi	69
Figura 81. Interfaz Home Assistant área dormitorio	70
Figura 82. Interfaz Home Assistant área aseo	70
Figura 83. Interfaz Home Assistant Historial de uso.....	71
Figura 84. Programación de presencia de la cama	71
Figura 85. Programación de encendido vitrocerámica	71
Figura 86. Herramienta InfluxDB en Home Assistant	72
Figura 87. Herramienta Grafana en Home Assistant.....	72
Figura 88. Creación usuario InfluxDB	73
Figura 89. Configuración InfluxDB en Home Assistant	73
Figura 90. Ejemplo configuración gráfica señal Wi-Fi	74
Figura 91. Ejemplo de Dashboard creada en Grafana	74
Figura 92. Batería recargable de Polímero de Litio de 3.7V	75
Figura 93. Módulo ADS1115	75

Figura 94. Esquemático general ESP32-WROOM-32	76
Figura 95. Esquemático general conexión batería.....	76
Figura 96. Valor Dropout Voltage en hoja de datos de CE6260.....	76
Figura 97. Esquemático general cargador de batería.....	77
Figura 98. Esquemático PCB con módulos	78
Figura 99. Diseño PCB con módulos TOP.....	79
Figura 100. Esquemático PCB integrados	80
Figura 101. Circuito de aplicación de la hoja de datos del HX711	81
Figura 102. Circuito de aplicación de la hoja de datos de MAX6675.....	81
Figura 103. Circuito de aplicación de la hoja de datos de ADS1115.....	81
Figura 104. Diseño PCB con integrados	81
Figura 105. Esquemático PCB con ADS1115 integrado.....	82
Figura 106. PCB ADS1115 integrado TOP	82
Figura 107. Sensor de presión de asiento situado en sofá	84
Figura 108. Detección de presencia sofá	84
Figura 109. Tiempo de uso del sillón en una sola tarjeta	84
Figura 110. Tiempo de uso del sillón en tarjetas diferentes	85
Figura 111. Cambio de la temperatura en la prueba de agua en ebullición.....	85
Figura 112. Medida de peso con células de carga	85
Figura 113. PCB con módulos.....	86
Figura 114. PCB con integrados.....	86

Índice de tablas

Tabla 1. Sensores asignados para este proyecto	18
Tabla 2. Características sistemas KNX [21].....	28
Tabla 3. Características protocolo Z-Wave. [21]	29
Tabla 4. Características protocolo ZigBee. [28].....	31
Tabla 5. Comparación de tecnologías inalámbricas bajo el estándar IEEE 802.11[30].	31
Tabla 6. Ventajas e inconvenientes de la red Wi-Fi [31]	32
Tabla 7. Definiciones destacadas del término Smart Home, de 1992 a 2019. [32].....	32
Tabla 8. Características Raspberry Pi 3 Model B. [43]	34
Tabla 9. Características ESP32 y ESP8266.....	35
Tabla 10. Características ESP32. [44]	36
Tabla 11. Tipos de termopares y su composición. [45].....	38
Tabla 12. Especificaciones técnicas HX711[46].....	41
Tabla 13. Características Diptrace [50]	45
Tabla 14. Presupuesto utilizando módulos	87
Tabla 15. Presupuesto con integrados	88

1. Introducción

1.1. Situación

En los últimos años el número de personas mayores ha ido en aumento. Esto se debe al cambio producido en los hábitos y la forma de vivir con los avances de la medicina, cuidados y tecnología. Según un estudio de la Organización de las Naciones Unidas [1] el envejecimiento de las personas mayores puede llegar a ser una de las transformaciones sociales más significativas del siglo XXI. De esta manera, en 2019 el 9% de la población mundial era mayor de 65 años y se estima que para el año 2050 este porcentaje se incremente al 16%.

El crecimiento de la población anciana ha despertado gran interés entre los investigadores a utilizar los recursos y avances en la tecnología e inteligencia artificial para crear diferentes sistemas y métodos que aporten una mejora de la calidad de vida en el envejecimiento.

Un ejemplo de este nicho de investigación se encuentra en Europa donde se ha creado el programa AAL (Active Assisted Living). Este programa financia el desarrollo de servicios que supongan una ayuda a las personas para enfrentarse al envejecimiento, así como a las personas encargadas de cuidar a las personas. [2]

Por otra parte, los acontecimientos sucedidos a partir de 2019 con la crisis sanitaria producida por la aparición del COVID-19 ha dado lugar a comprobar el porcentaje de personas que viven en soledad.

Según las estadísticas del Instituto Nacional de Estadística en España, en 2020, los grupos de personas mayores de 65 años son aquellos que más tienden a vivir en soledad como se puede observar en la Figura 1.

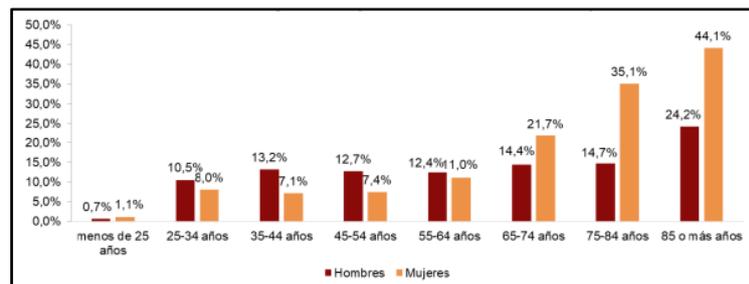


Figura 1. Porcentaje de personas que viven solas según sexo y edad en el año 2020. [3]

El conjunto de estos acontecimientos y resultados ha dado lugar a que el campo de investigación de la tecnología orientada al cuidado de las personas mayores esté en desarrollo y haya aumentado tanto en los últimos años.

1.2. Objetivos

El proyecto que se va a detallar en este documento se enmarca dentro del proyecto Robwell (Robotic-based Well-Being Monitoring and Coaching for Elderly People during Daily Life Activities). Este proyecto tiene como objetivo principal combinar el uso de la domótica, robótica e inteligencia artificial con el uso de un entorno domótico y un robot móvil en una vivienda para capturar información sobre los parámetros fisiológicos y rutinas del usuario para que, en función de las respuestas obtenidas, el robot móvil responda de forma adecuada para mejorar ese estado del usuario. [4]

Este trabajo fin de grado forma parte del entorno domótico del proyecto Robwell. Esta parte incluye el diseño, prototipado e integración de un sistema de ese entorno basado en la recogida y captura de información de parámetros a través de sensores de bajo coste controlados por un SoC y donde la información se visualiza en una plataforma de automatización del hogar instalado en un microordenador.

Por tanto, el objetivo principal de este proyecto es:

- Uso de sensores de bajo coste de manera no intrusiva para captar y recoger la información de los hábitos del usuario. Estos hábitos pueden ser:
 - Obtener datos del pesaje de la cama mediante células de carga colocadas en las patas.
 - Extraer datos de temperatura de la vitrocerámica a través de un sensor de temperatura para su posterior interpretación.
 - Uso de sensores de presión de asiento situado en los sillones o sofá del salón.
 - Uso de interruptores de caudal o caudalímetros situado en los grifos tanto del aseo como de la cocina.

Tabla 1. Sensores asignados para este proyecto

	Dormitorio	Cocina	Salón	Baño
Sensor de presión de asiento			✓	
Sensor caudal		✓		✓
Células de carga	✓			
Sensor temperatura		✓		

Para poder cumplir con el objetivo principal es necesario proceder con unos objetivos previos.

- Documentación acerca del entorno de Home Assistant
- Documentación acerca de la utilización de la herramienta ESPHome.
- Instalación y puesta en marcha de Home Assistant.
- Programación de los diferentes sensores en ESPHome.
- Estudio de la herramienta InfluxDB y Grafana para la gestión y visualización de los datos de Home Assistant.
- Diseño de tarjeta de circuito impreso (PCB).

2. Estado del Arte

2.1. Desarrollo hasta la actualidad de Smart Home y AAL

El inicio de la domótica en la historia es impreciso, ya que se trata de un conjunto de procesos que fueron evolucionando en paralelo a los sistemas electrónicos y de computación.

El concepto de domótica se aplicó por primera vez en la década de los 70 con la invención del ECHO IV (Electronic Computing Home Operator 4), creado por Jim Sutherland y era capaz de almacenar recetas, hacer una lista de la compra y apagar electrodomésticos, entre otras muchas de las tareas que podía realizar. Al tratarse de un aparato de gran tamaño y una tecnología muy avanzada, no captó la atención de muchos investigadores de la época, pero supuso la base de lo que se conoce actualmente como domótica.



Figura 2. Jim Sutherland sentado enfrente del ECHO IV junto a su familia. (1966)

No fue hasta la década de los 80 cuando se empezaron a crear las primeras experiencias de domótica. La primera experiencia conocida fue en 1982 por la sociedad OSAKA GAS, en Japón, que realizó el prototipo “CASA JEXT”. En el cual albergaba una familia representativa de los hogares japoneses. Se hacía uso del gas económico como fuente principal de energía e incluía la gestión energética a partir de baterías solares.

Por otra parte, se creó el concepto de domótica y casa inteligente desde un enfoque americano durante la década de los 80, que se acerca a lo que conocemos hoy en día como domótica. Este enfoque se basa en la existencia de una red unificada que posee una inteligencia distribuida para corroborar y asegurar la distribución eléctrica y el control de las comunicaciones y datos. Define así que los dispositivos domóticos deben estar conformados con protocolos de comunicación para poder comunicarse con los sistemas central y de control, así como con los sensores. [5]

Fue a partir de la década de los 90 con el *boom* de internet que estas tecnologías se empezaron a desarrollar a pasos agigantados. Con el desarrollo avanzado de las tecnologías y de lo que conocemos como Internet of Things (IoT) ha supuesto un gran avance en las aplicaciones que nos permiten el control de dispositivos o de sistemas electrónicos y eléctricos.

Por otra parte, a lo largo de los años, con las mejoras médicas y de calidad de vida, la esperanza de vida media de las personas es mayor mientras que la tasa de natalidad está disminuyendo. Esto da a lugar a que el número de personas mayores haya aumentado.

En 2019, el porcentaje de población mundial según los datos de las Naciones Unidas era del 9% y según los datos del informe “Perspectivas de la población mundial 2019” se estimaba que en el año 2050 el 16% de la población mundial tendrá más de 65 años. Esto supone uno de los cambios sociales y demográficos más importantes del siglo XXI. [6]

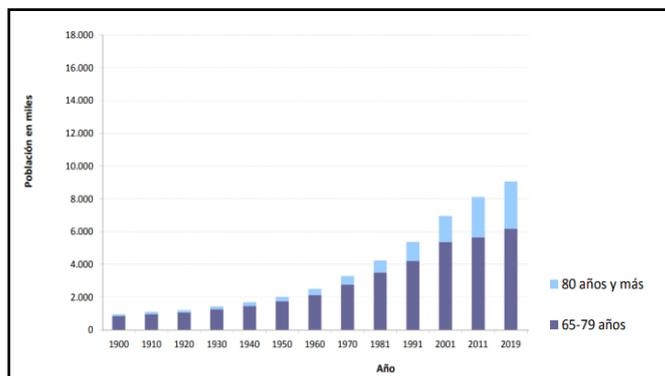


Figura 3. Evolución de la población hasta 2019 en España. [INE:1900-2011: Censos de población y vivienda. 2019]

Este grupo de personas (en comparación con los grupos de personas más jóvenes) tienden a sufrir múltiples problemas de salud, enfermedades físicas crónicas, limitación del movimiento y demencia que pueden requerir de atención constante y, a pesar de que la atención a mayores es un campo muy desarrollado y que facilita la calidad de vida de las personas mayores, este grupo tiende a tener preferencias por seguir con su independencia para el día a día. [7]

Este cambio demográfico y el desarrollo de la tecnología IoT en los últimos años ha provocado un gran interés en los investigadores de todo el mundo por desarrollar proyectos para ayudar a mejorar la calidad de vida de las personas mayores, así como, darles acceso a la independencia que ellos buscan. [8]

A continuación, se presentarán algunos proyectos realizados en la última década.

El proyecto HOPE (Smart Home for the Elderly People) en 2010. Se trata de un proyecto cofinanciado por la Comunidad Europea cuyo objetivo es mejorar la calidad de vida de las personas mayores, específicamente a aquellos que padecen Alzheimer. Está enfocado, por tanto, a la prolongación de la independencia de los enfermos con Alzheimer mediante el uso de las tecnologías de la Información y la Comunicación (TIC). Además, les permite realizar por sí mismos actividades que antes no podían hacer y que son importantes para su vida personal cotidiana. HOPE consiste en una innovadora caja de control universal (UCB) inteligente basada en IP que utiliza la inteligencia para gestionar los diversos subsistemas y dispositivos conectados dentro de una residencia de personas mayores. [9]



Figura 4. Logo del proyecto HOPE. [9]

En 2011, la Agencia Nacional de Investigación francesa (ANR) fundó el proyecto AuoVADIs (Ayuda Distancia a la Vida Cotidiana para personas mayores con deterioro cognitivo) en el cual se propone una plataforma multimodal llamada EMUTEM que está compuesta por un conjunto de micrófonos repartidos por la vivienda para monitorizar el ambiente acústico de la persona, un dispositivo portátil que mide las señales fisiológicas y detecta la caída y un conjunto de sensores infrarrojos capaces de detectar tanto la presencia como la posición en la que se encuentra la persona. Esta plataforma utiliza la lógica difusa para su funcionamiento. [10]

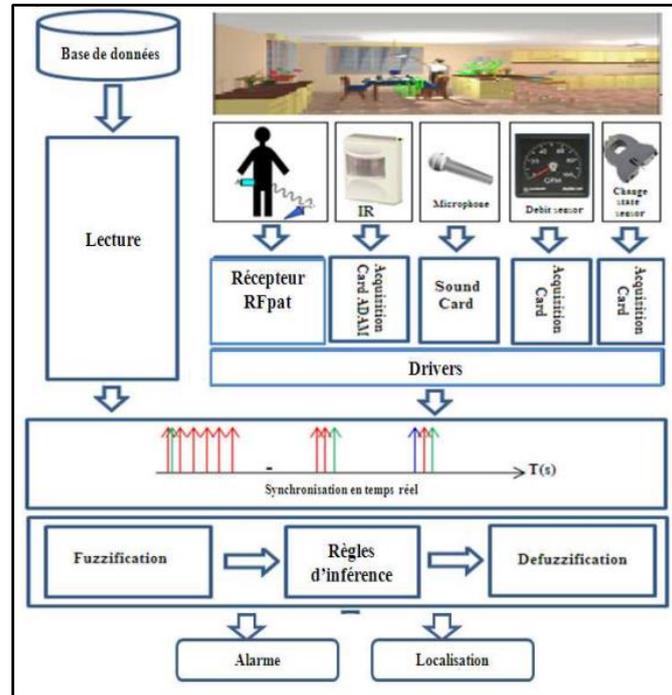


Figura 5. Arquitectura de EMUTEM [10]

En 2015, Un grupo de universidades españolas ideó el proyecto SIRMAVED. Se trata del desarrollo de un sistema integral robótico de monitorización e interacción para personas con daño cerebral adquirido y dependientes. La base de este sistema es crear una terapia basada en el uso de un ambiente inteligente, así como de un robot social autónomo de estimulación activa para el hogar. [11] El ambiente inteligente está basado en varios sensores RGB-D que se encuentran distribuidos por la vivienda con un sistema de segmentación de personas y un algoritmo de seguimiento para proporcionar de forma estimada la posición a la que se encuentra la persona. Por otro lado, se incluye en este ambiente la instalación de alarman de caída o de estancias prolongadas en el baño que alertan de forma directa a los familiares y cuidadores. [12]

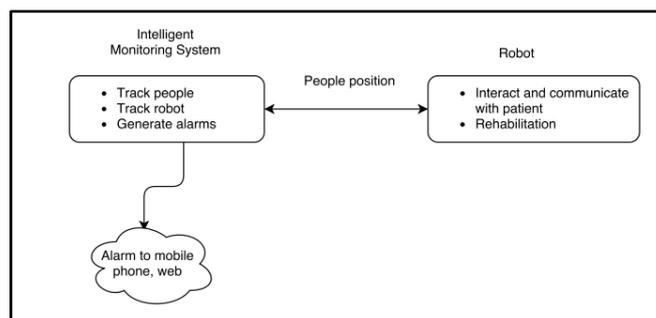


Figura 6. Sistema completo SIRMAVED. [11]

En 2017, la universidad Politécnica de Milán ideó un sistema de televigilancia basado en la instalación de sensores para la seguridad de las personas mayores llamado SMARTA. Este proyecto consta de cuatro partes; un subsistema de monitorización biomédica, un subsistema de entorno basado en acelerómetros, una red de sensores/actuados de automatización doméstica y un centro/servidor para recoger, procesar y visualizar los datos. Unos de los subsistemas de monitorización biomédica es la creación de una camiseta sensorizada a medida para la detección de caídas y seguimiento de vidas de esta manera la persona mayor está siempre monitorizada, tanto en el interior como en el exterior.

El proyecto SMARTA está desarrollado para ser integrado en una vivienda ya construida incluyendo así piezas de domótica y sensores. Este proyecto está desarrollado en el departamento DAT de la Fondazione don Gnocchi (Milán, Italia). [13]

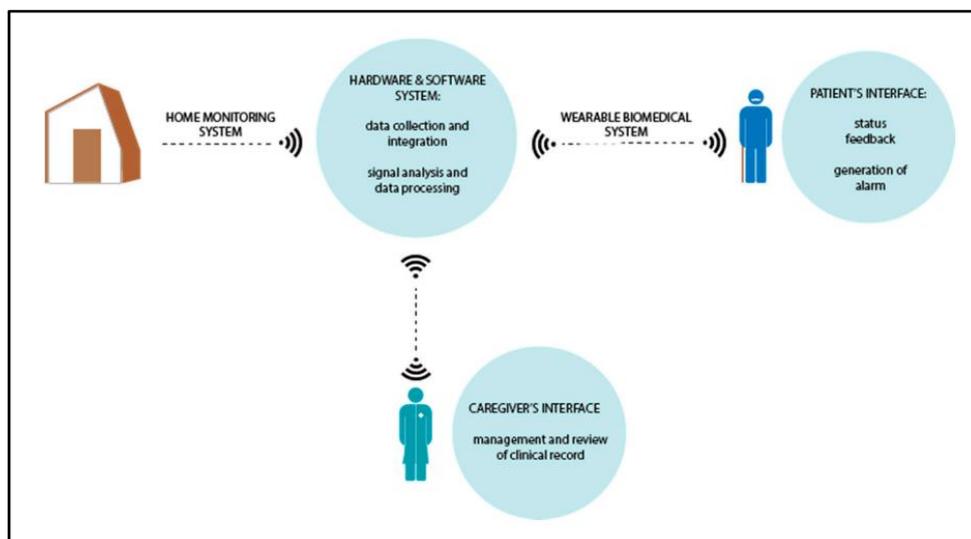


Figura 7. Componentes de un sistema SMARTA. [13]

En 2019, la Universidad Politécnica de Cartagena (Murcia, España), la Universidad de Murcia (España), La Universidad de Orebro (Suecia) y la Universidad de Kyushu (Japón) proponen el proyecto Robwell (Robotic-based Well-Being Monitoring and Coaching for Elderly People during Daily Life Activities).

Este proyecto se basa en la propuesta de una plataforma robótica asistencial que debe circular de manera autónoma en el entorno real de una vivienda y que, adicionalmente, debe realizar actividades de coaching emocional mejorar el estado anímico del usuario. En segundo lugar, se propone la realización de un ecosistema domótico en la vivienda donde se encuentre integrado el robot basado en dispositivos de bajo coste (en su mayoría con protocolo ZigBee) para monitorizar el estilo de vida del usuario. Por otro lado, se puede realizar un estudio psicológico mediante el uso de tecnología de predicción del estado anímico basado en inteligencia artificial con el uso de una aplicación móvil y de dispositivos como es la Empática 4. El proyecto Robwell es un subproyecto de otro mayor llamado HIMTAE (Heterogeneous Intelligent Multi-Robot Team for Assistance of Elderly People) en el cual participa también la Universidad Carlos III de Madrid. [14]

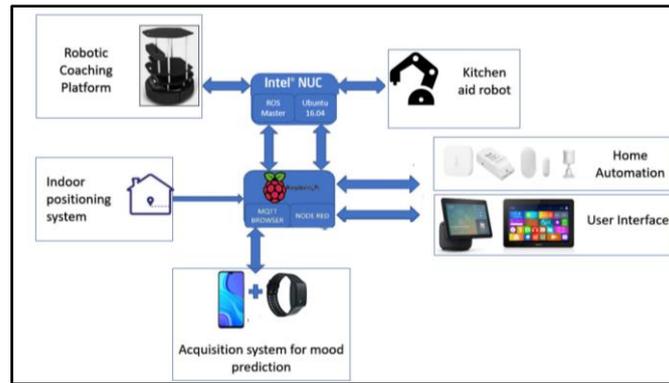


Figura 8. Esquema Hardware del proyecto HIMTAE.

Con este repaso en el tiempo de los diferentes proyectos de domótica y Smart Home aplicados a la mejora de la calidad de vida de las personas mayores se puede ver que hay numerosas formas y métodos de aplicación. Aunque el desarrollo de esta tecnología está en un punto importante y avanza a pasos agigantados todavía queda mucho por hacer y mejorar. Es por ello que este proyecto se basa en la monitorización de los aspectos básicos de la vida cotidiana de una persona mayor orientado al uso de un sistema de bajo coste y de fácil adquisición para mejorar la accesibilidad y que esté al alcance de todos.

2.2. Inmótica

Previo a introducir el término Domótica se debe conocer y diferenciar entre el término Domótica e Inmótica.

La Inmótica hace referencia a aquellas tecnologías aplicadas al control y la automatización que no están destinadas a viviendas, es decir que están destinados a edificios como pueden ser empresas, hospitales u hoteles.

Esto hace que su desarrollo e instalación sea más complejo y se utilicen métodos de instalación orientados a sistemas interconectados por cable complementado con sistemas secundarios inalámbricos.

A diferencia de la domótica, la inmótica tiene el objetivo principal de buscar resolver funciones más complejas como pueden ser los riesgos, las bases de datos o la refrigeración del edificio, pero sin dejar de lado la función de confort que puede brindar al usuario.

A pesar de las diferencias que se encuentran entre domótica e inmótica, no se tratan de tecnologías incompatibles. Ambos sistemas pueden complementarse y los elementos de cada uno se pueden utilizar en los dos sistemas.

2.3. Domótica

Se habla del término “Domótica” como una relación directa con la palabra “Informática”, sustituyendo el prefijo que significa información por otro derivado de la palabra latina “*Domus*” que su significado es casa. [15] De este modo el término domótica proviene de la unión de las palabras “*domus*” y “*tica*” que significan respectivamente casa y automática en griego, vocablo griego para decir “casa que funciona por sí sola”. [16]

Distintos organismos franceses acordaron una definición técnica para el término domótica:

“Conjunto de servicios del hábitat suministrados por sistemas que realizan varias funciones y que pueden conectarse entre sí y a redes internas y externas de comunicación. Entre estas funciones, se encuentran principalmente el ahorro de energía, y la gestión técnica, la información y la comunicación, el dominio del confort, la seguridad y la asistencia”. [17]

La domótica agrupa un conjunto de técnicas que emplean la electrónica, la informática y los automatismos industriales. Su funcionamiento se basa en la identificación de señales de entrada que son captadas por sensores y, posteriormente, el procesamiento de la información para ejecutar un actuador. Sus fines son ofrecer al usuario confort, ahorro energético, seguridad, comunicación y accesibilidad como se explica a continuación [15]:

- **Ahorro energético:** Es un concepto que se puede interpretar de varias formas como puede ser con una gestión eléctrica a la hora de racionalizar cargas eléctricas desconectando equipos de uso no prioritario en función del consumo eléctrico que tengas en un momento dado.
- **Confort:** Este objetivo conlleva todas aquellas funciones que permitan una mejora del confort de una vivienda. Se puede diferenciar entre confort ambiental (creación de un medio ambiente agradable como puede ser el confort visual o

acústico) y el confort de actividad (este objetivo viene del deseo y necesidad de facilitar las actividades cotidianas como puede ser dormir y alimentarse).

- **Seguridad:** Se basa en una red de seguridad que sirve para proteger tanto los Bienes Patrimoniales como la seguridad personal como puede ser con la simulación de presencia, la instalación de alarmas de detección de incendios y gas o la alerta médica y teleasistencia.
- **Comunicación:** Son los sistemas o infraestructuras de comunicaciones que posee el sistema. Esto incluye el control remoto desde Internet y mando inalámbricos y las intercomunicaciones.
- **Accesibilidad:** Este fin se propone como que la domótica es un diseño para todos. Es accesible para la diversidad humana, la inclusión social y la igualdad. De esta manera personas con discapacidad reducida puedan tener acceso a estas tecnologías sin temor a la inaccesibilidad que puedan creer tener.

La domótica tiene como objetivo ofrecer una mejor calidad de vida en la vivienda dando respuesta a diferentes necesidades. Por ello, la domótica puede cubrir las distintas necesidades con ciertas funciones que dependen de la información captada y transmitida por los sistemas de medida como son las funciones de confort, gestión y comunicación.

- **Función de control:** Esta función aporta al usuario información sobre el estado de funcionamiento del sistema y de las instalaciones que lo integran, así como, la creación de un registro de los distintos parámetros obtenidos. Dentro de la función de control se encuentra el control técnico, que tiene el objetivo de hacer más veraz el uso del equipo, los dispositivos e instalaciones para poder lograr así un mantenimiento predictivo como puede ser la recepción de notificaciones por un mal funcionamiento de las instalaciones. Por otro lado, se encuentra la asistencia a la salud que consiste en la conexión del usuario con centros asistenciales para asegurar el control y seguimiento de ciertos parámetros que pueden ser claves dependiendo de cada paciente.
- **Función de gestión:** Tiene como objetivo automatizar diferentes acciones que principalmente están relacionadas con el confort. Se puede destacar dentro de esta función elementos de control como puede ser la iluminación, que ofrece al usuario una gestión de los elementos principales que están ligados al confort, la calefacción, ventilación e incluso la calidad del aire. Todas estas acciones pueden ser realizadas desde el interior por el usuario mediante una interfaz o ser conectadas desde el exterior. Del mismo modo todas estas funciones de gestión pueden ejecutarse de forma automática y ser cancelada y/o modificada desde el interior y/o desde el exterior.
- **Función de comunicación:** Los servicios y aplicaciones contemplan el intercambio de mensajes tanto entre personas, como entre personas y equipos, dentro de la propia estructura y desde el exterior. El objetivo de esta función es, mediante técnicas de control, lograr el control del usuario con el sistema de forma remota, como es el caso de la utilización de comandos a distancia. Otro objetivo es que no solo se realice una conexión entre el usuario y los dispositivos, sino que exista un intercambio de información entre los diferentes dispositivos como pueden ser señales de video y audio.

2.4. Estándares

Para lograr la comunicación entre los dispositivos, así como la comunicación entre los dispositivos y el usuario, es necesario una red de comunicación que conecte el sistema. Es por ello por lo que es necesaria la estandarización. Un estándar de comunicación establece las normas para el intercambio de información.

La estandarización del sistema tanto en el medio físico como de los protocolos de acceso y comunicación es un punto importante para tener en cuenta para crear de forma independiente un cableado interior del tipo de dispositivos que se utilizarán, así como, asegurar la interconectividad de distintos dispositivos independientemente del tipo o del proveedor que sea. [5]

Huidobro J.M. y Millán R. (2004) recogen que el origen de la domótica se remonta a los años 70, es ahí cuando comenzaron las primeras estandarizaciones. Este origen es dado cuando aparecieron los primeros dispositivos de automatización de edificios basados en la tecnología X-10. [16]

El estándar X-10 es de los protocolos más antiguos para la domótica, pero hasta la actualidad se han desarrollado diferentes protocolos. Los protocolos desarrollados junto con el X-10 son:

2.4.1. X-10

El sistema X-10 fue diseñado por la empresa escocesa Pico Electronics Ltd entre los años 1.976 y 1.978 con el objetivo de transmitir datos por la red eléctrica de baja tensión (en el caso de Estados Unidos de 115V a 60Hz y de Europa de 230V a 50 Hz) a muy baja velocidad (en el caso de Estados Unidos a 60 bps y 50bps en el caso de Europa) y con un coste muy bajo [16].

Está compuesto por dos módulos; un controlador que actúa como transmisor y un interruptor remoto que actúa como receptor. De esta manera el segundo módulo, es decir, el receptor puede ser conectado a cualquier toma eléctrica, de esta forma, al conectar un electrodoméstico, el receptor recibe, comprueba e interpreta la llegada de señales a través de la línea eléctrica.

Las señales que representan la información se transmiten en ráfagas de pulsos de 120 KHz. Esto hace que el estándar X-10 se base en la detección de los pasos por cero determinado por las señales codificadas. Dentro de las señales que puede recibir existen dos tipos: el código de dirección y el código de función. Puede asignar hasta 16 grupos de direcciones denominados “Códigos de casa” (que va desde la A hasta la P) y hasta otras 16 direcciones denominadas “Códigos de Unidad” (Que va desde 1 hasta 16). Es por esto por lo que con este protocolo pueden ser localizados hasta 256 dispositivos en la red.

Un 1 lógico representa un pulso de 120KHz durante 1 milisegundo del semiciclo positivo de la señal eléctrica y la ausencia de la ráfaga en el semiciclo negativo y un 0 lógico representa la ausencia de este pulso. Para realizar una transmisión completa en un código X-10 es necesario realizar 11 ciclos de corriente. [18]

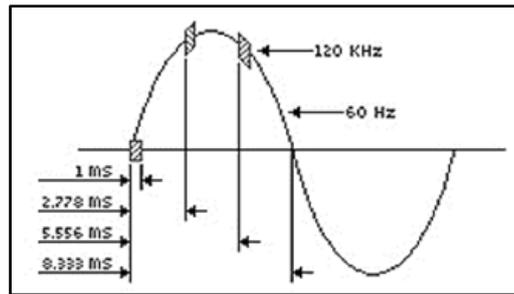


Figura 9. Esquema de transmisión del protocolo X-10.

El sistema no tiene inteligencia para ejecutar tareas de modo automático y en su primera versión tan solo contemplaba las operaciones de encender, apagar, aumentar, disminuir, todo apagado y todo encendido.

Las principales características de este estándar son: Posee una instalación sencilla de conectar y empezar a funcionar, es de fácil manejo por el usuario, es flexible y ampliable y es un sistema descentralizado y no programable. [5]

2.4.2. KNX

Es un estándar común europeo ideado en 1999 por la Asociación EIB (EIBA), el Batibus Club Internacional (BCI, Francia) y la European Home Systems Association (EHSA, Holanda). Llamado KNX, en la actualidad este estándar está certificado como; estándar europeo (CENELEC EN 50090 y CEN EN 13321-1 y 13321-2), estándar internacional (ISO/IEC 14543-3), estándar chino (GB/Z 20965) y estándar norteamericano (ANSI/ASHRAE 135).



Figura 10. Logo KNX.

Tal y como define la página oficial de KNX, El sistema KNX es un sistema de bus desarrollado para el control y la automatización de viviendas y edificios. Todos los dispositivos usan el mismo medio de comunicación a través del bus común. [19]

Un sistema KNX está formado principalmente por cuatro elementos; actuadores, sensores, pasarelas y acopladores. [20]

- **Actuadores:** Son aquellos elementos que se encuentran conectados físicamente sobre los elementos que se van a controlar del espacio. Por ejemplo: luces y motores.
- **Sensores:** Son a aquellos elementos encargados de interpretar órdenes del usuario, o recopilan datos. Por ejemplo: pulsador, detector de movimiento y termostato.
- **Pasarelas:** También llamadas Gateway o routers. Son aquellos elementos que se encargan de enlazar sistemas que poseen otros protocolos de comunicación con KNX. Por ejemplo: DALI, IP y RS485.

- **Acopladores:** Estos elementos hacen una separación dentro del bus, lo que hace que se agrupen los dispositivos en un segmento de características que está determinado para la cantidad de equipos o ubicaciones físicas que haya y conectarlo con otro segmento para una mayor eficiencia en el envío de datos a través del bus, así como, alcanzar mayores distancias.

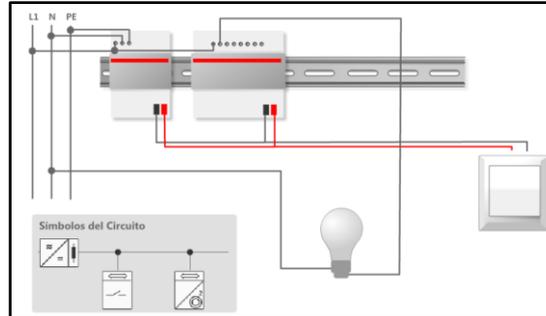


Figura 11. Ejemplo de simbología KNX.

El diseño, la implementación y la configuración de toda instalación KNX se realiza mediante una única herramienta, denominada ETS (Engineering Tool Software), que es independiente de la aplicación y del fabricante.

Las principales características de KNX se muestran en la Tabla 2 :

Tabla 2. Características sistemas KNX [21]

Características KNX:	Es un estándar internacional.
	Es un sistema abierto, independiente de cualquier fabricante de producto.
	Es libre de cualquier tipo de canon.
	Cualquier instalación es escalable. Permite ampliaciones futuras de forma sencilla.
	Es de alta calidad. ISO 9001 es aplicado en todos los fabricantes KNX.
	Soporta varios medios de transmisión: par trenzado (TP), corrientes portadoras (PL), Ethernet (IP) y radiofrecuencia (RF).
	Puede ser acoplado a otros sistemas: DALI, Modbus, BACnet...

2.4.3. LonWorks

Fue creado por la empresa Echelon Corporation en 1988 y se trata de un sistema de automatización enfocado a edificios (por su gran complejidad) en el que los dispositivos (nodos) se comunican a través de un bus utilizando el protocolo LonTalk.

Las redes LonWorks se basan en tres conceptos muy sencillos. Estos conceptos son: 1) Sistemas de control fundamentalmente idénticos, 2) Un sistema de control distribuido presenta ventajas en potencia y flexibilidad que un sistema centralizado, 3) La reducción económica a largo plazo que tienen las empresas con el uso de sistemas distribuidos. [22]

De esta manera, los nodos que conforman la red poseen su propia inteligencia para poder ejecutar de forma casi independiente diferentes programas entre sí a la misma que proporciona información a los dispositivos que se encuentran en otras áreas.

La empresa Echelon define la interoperabilidad como la capacidad de integrar productos de distintos fabricantes en sistemas que sean flexibles y funcionales sin tener la necesidad de incluir un desarrollo hardware o software a medida. Es por ello que en la red LonWork todas las tareas que se configuran son gestionadas por dispositivos interoperables.[23]

Como se ha mencionado antes, la comunicación en las redes LonWorks se realizan mediante un protocolo estándar de comunicación llamado LonTalk. Este protocolo proporciona servicios de envío y recepción de datos sea cual sea la topología o funciones de los nodos.[24]

2.4.4. Z-Wave

Según explican en la página oficial de Z-Wave, se trata de una tecnología inalámbrica que no interfiere con la señal Wi-Fi y funciona con baja potencia. Cuando la tecnología Z-Wave se utiliza dentro de productos cotidianos como cerraduras y luces, estos productos se convierten en "inteligentes", dándoles la capacidad de hablar entre sí y permitiéndole controlar los dispositivos y, por lo tanto, su hogar, desde cualquier lugar [25]. Está certificado como estándar en más de 30 países y el estándar europeo es el EN 300 220 donde los productos Z-Wave trabajan entre 868.40 y 869.85 MHz.

Su funcionamiento se basa en una tecnología de radiofrecuencia inalámbrica que permite a los elementos inteligentes de un sistema domótico comunicarse entre sí. [26]

Las características de Z-Wave se muestran en la Tabla 3 :

Tabla 3. Características protocolo Z-Wave. [21]

Características Z-Wave:	Opera en la banda sub-1 GHz.
	La tecnología de comunicaciones por radiofrecuencia de baja potencia que soporta redes de malla completa sin la necesidad de un nodo coordinador.
	Impermeable a la interferencia de Wi-Fi y otras tecnologías inalámbricas en la banda de 2,4 GHz (Bluetooth, ZigBee, etc.).
	Diseñado específicamente para aplicaciones de control y estado
	Soporta velocidades de datos de hasta 100 kbps.
	Puenteador con éxito y ensayado con OpenADR, 1 SEP-SEP-1.1 y otros protocolos de uso inteligente de energía.

2.4.5. ZigBee

Se trata de un protocolo de comunicación inalámbrico de alto nivel ideado por el grupo ZigBee Alliance. Este protocolo está basado en el estándar IEEE 802.15.4 de redes inalámbricas personales (tiene similitud al estándar Wi-Fi 802.11). Este protocolo es capaz de gestionar de forma automática la red, creando y manteniendo así las rutas de una forma dinámica, gracias a esto la red puede corregir automáticamente situaciones de desconexión tanto parcial como total de los nodos que lo forman, de esta manera se evita el acceso físico a los equipos para resolver este tipo de problemas.

Posee un alcance de entre los 10 y los 20 metros de longitud. Los dispositivos ZigBee son capaces de comunicarse enviando y recibiendo datos entre sí, esto da a lugar a que se puede encadenar varios dispositivos y vincularlos a un *hub* sin que, necesariamente, todos los dispositivos estén dentro del alcance del *hub*.

Se conoce como *hub* como un dispositivo que actúa como puente entre diferentes dispositivos utilizando una única conexión. Este dispositivo puede poseer varias entradas y salidas o varias salidas y una única entrada por las cuales se conectan y utilizan, ya sea entre varios o uno a la vez, los dispositivos.

Dentro de los dispositivos ZigBee encontramos tres tipos; Coordinador, Router y End-Device. [27]

- **Coordinador:** Es el dispositivo más importante y se encarga de crear la red, organizar las rutas y permitir las conexiones que provienen de otros nodos de la red. Por cada red sólo existe un coordinador. Se puede asociar a lo que antes se ha explicado como *hub*.
- **Router:** Posee una función similar al controlador, con la diferencia que es capaz de enrutar y enlazar diferentes dispositivos, pero no es capaz de permitir aceptar conexiones. Se utiliza, sobre todo, para “repetir” la señal ZigBee entre dispositivos.
- **End-Device (Dispositivos):** Estos nodos representan los dispositivos más sencillos de la red. Son capaces de enviar y recibir información, pero únicamente a controladores y routers. Estos dispositivos no se pueden comunicar entre sí. Corresponden con el final de la red, es decir, las bombillas o interruptores de tu entorno domótico.

En cuanto al funcionamiento existen diferentes topologías de red posibles; Estrella, árbol y malla como se muestra en la Figura 12.

- **Estrella:** Todos los dispositivos en la red se comunican únicamente con el coordinador.
- **Árbol:** El coordinador es aquel que establece la red inicial y, haciendo analogía con un árbol, los routers representan las ramas que permiten la comunicación a los dispositivos que actúan como hojas en esta topología de árbol.
- **Malla:** Se conoce como topología de punto a punto en el que los dispositivos pueden comunicarse entre sí sin restricciones.

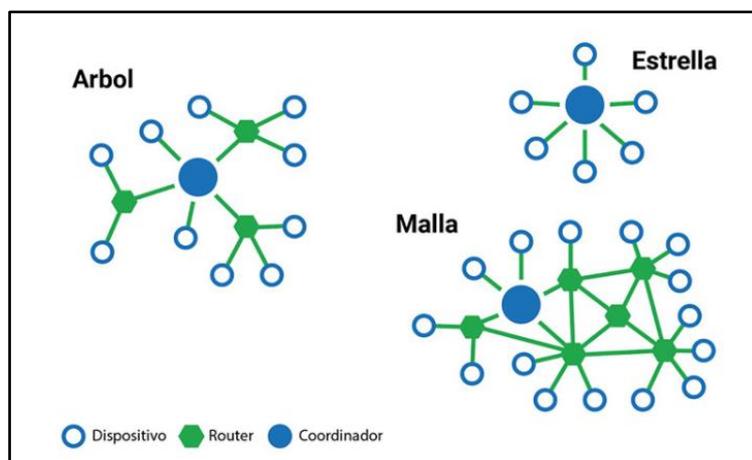


Figura 12. Topologías de red ZigBee.

Las características del protocolo ZigBee se describen en la Tabla 4 :

Tabla 4. Características protocolo ZigBee. [28]

Características ZigBee:	Puede trabajar tanto en las bandas de 2.4GHz como en la de 868/915MHz.
	Tasa de transmisión máxima de: 250 kbps @2,4 GHz, 40 kbps a 915 MHz y 20 kbps a 868 MHz.
	64 bits de direccionamiento determinan una cantidad máxima de $1.8.10^{19}$ dispositivos.
	16 bits para identificar redes que determina un total de 65536 redes.
	Alto rendimiento y baja latencia para aplicaciones de bajo ciclo de trabajo (<0,1%).
	Protocolo con handshake (diálogo) para mejorar la seguridad en las transferencias.
	Rango: Hasta 50 m (Valor típico).

2.4.6. Wi-Fi

Wi-Fi (*Wireless Fidelity*) es un conjunto de estándares para redes inalámbricas (*WLAN*) que pertenece a la familia del estándar IEEE 802.11 que trabaja en una onda de frecuencia de 2.4GHz. Bajo la marca Wi-Fi se estableció el certificado para la interoperabilidad entre dispositivos en abril del 2000 por la asociación WECA (*Wireless Ethernet Compatibility Alliance*). Esta asociación fue creada en 1999 por las principales empresas de soluciones inalámbricas como son 3Com, Aironet y Nokia y tiene el objetivo principal de diferenciar y hacer comprensible la tecnología Wi-Fi.[29]

El uso de la red inalámbrica Wi-Fi permite que distintos tipos de dispositivos puedan estar interconectados sin necesidad de cableado.

Como podemos observar en la Tabla 5 se presenta una comparación entre las distintas variantes que existen dentro del estándar IEEE 802.11.

Tabla 5. Comparación de tecnologías inalámbricas bajo el estándar IEEE 802.11[30]

	802.11a	802.11b	802.11g
Año de publicación	1999	1999	2003
Banda de frecuencia	5 GHz	2.4 GHz	2.4 GHz
Modulación	OFDM	DSSS	DSSS y OFDM
Velocidad máxima	54 Mbps	11Mbps	54 Mbps
Velocidad típica	18 Mbps	6 Mbps	12 Mbps
Rango de velocidades (Mbps)	54, 48, 36, 24, 18, 12 9 y 6	11, 5.5, 2 y 1	54, 48, 36, 24, 18, 12, 11, 9, 6, 5.5, 2 y 1
Número de canales sin sobreposición	8	3	3
Ancho de banda en un área	432 Mbps (8 x 54)	33 Mbps (3 x 11)	162 Mbps (3 x 54)
Usuarios en un área	512	192	192

Para la conexión de los sensores y captación de la información proporcionada en este proyecto se ha usado el protocolo Wi-Fi desde un módulo ESP32 conectado a una Raspberry Pi.

En la Tabla 6 se presentan las ventajas e inconvenientes del protocolo Wi-Fi.

Tabla 6. Ventajas e inconvenientes de la red Wi-Fi [31]

Ventajas	Inconvenientes
Implementación fácil y rápida	Ancho de banda menor que de una red cableada
Se puede trabajar en cualquier lugar dentro del alcance del punto de servicio	Es vulnerable a interferencias tanto externas como internas
Bajo coste económico	No son redes totalmente seguras.

2.5. Smart-Home

El término *Smart Home* o casa inteligente ha estado presente en la mente de los inventores desde que la domótica se empezó a desarrollar. *Smart Home* hace referencia a un conjunto de sistemas encargado de gestionar y regular elementos y electrodomésticos que están instalados en una vivienda.

Se trata de una incorporación de soluciones para hogar que se compone de varios productos que proporcionan al usuario una mejora del confort y la seguridad de su vivienda en el día a día.

El término *Smart House* es algo que ha estado durante décadas sobre las mentes de los diferentes inventores desde los primeros pasos con la domótica en los años 70, pero no fue hasta 1992 que se empezó a cuestionar una definición propia para el concepto *Smart Home*. Desde entonces se han ideado numerosas definiciones para este término como se muestra en la Tabla 7 :

Tabla 7. Definiciones destacadas del término *Smart Home*, de 1992 a 2019. [32]

Fuente	Fecha	Definición:
Lutolf [33]	1992	La integración de diferentes servicios dentro de un hogar mediante el empleo de un sistema de comunicación común. Asegura un funcionamiento económico, seguro y confortable de la vivienda e incluye un alto grado de funcionalidad inteligente y flexibilidad.
Aldrich [34]	2003	Una residencia equipada con tecnología informática y de la información, que anticipa y responde a las necesidades de los ocupantes, trabajando para promover su comodidad, conveniencia, seguridad y entretenimiento a través de la gestión de la tecnología dentro del hogar y las conexiones con el mundo más allá.
De Silva et al. [35]	2012	Un entorno similar a un hogar que posee inteligencia ambiental y control automático, lo que le permite responder al comportamiento de los residentes y ofrecerles diversas facilidades.
Balta-Ozkan et al. [36]	2014	Una residencia equipada con una red de comunicaciones, que enlaza sensores, aparatos domésticos y dispositivos, que pueden ser supervisados, accedidos o controlados a distancia y que proporcionan servicios que responden a las necesidades de sus habitantes.
Buildings Performance Institute Europe [37]	2017	Un edificio inteligente es altamente eficiente desde el punto de vista energético y cubre su bajísima demanda de energía en gran medida con fuentes de energía renovable in situ o a través de sistemas de distrito. Un edificio inteligente (i) estabiliza e impulsa

		una descarbonización más rápida del sistema energético mediante el almacenamiento de energía y la flexibilidad del lado de la demanda; (ii) otorga a sus usuarios y ocupantes el control de los flujos de energía; (iii) reconoce y reacciona a las necesidades de los usuarios y ocupantes en términos de confort, salud, calidad del aire interior, seguridad y requisitos operativos.
Hargreaves and Wilson [38]	2017	Un Smart Home recoge y analiza datos del entorno doméstico, transmite información a los usuarios (y a los proveedores de servicios) y mejora el potencial de gestión de los diferentes sistemas domésticos (por ejemplo, calefacción, iluminación, entretenimiento)
Strengers and Nicholls [39]	2017	El Smart Home engloba las TIC del hogar, los dispositivos y aparatos conectados y automatizados, y el Internet de las Cosas (IoT).
Shin et al. [40]	2018	Un entorno inteligente capaz de adquirir y aplicar conocimientos sobre sus habitantes y su entorno para adaptarse y cumplir los objetivos de confort y eficiencia.
Gram-Hanssen and Darby [41]	2018	Uno en el que una red de comunicaciones enlaza sensores, aparatos, controles y otros dispositivos para permitir la supervisión y el control a distancia.
Marikyan et al. [42]	2019	Una residencia equipada con tecnologías inteligentes destinadas a ofrecer servicios a medida de los usuarios.

3. Materiales

3.1. Raspberry Pi

Para la implementación de este sistema domótico en la vivienda se decidió por el uso de un microordenador o SBC (Single Board Computer) por lo que se ha realizado un estudio de los diferentes SBC del mercado finalmente llegando a la opción del uso de una Raspberry Pi.

Para este proyecto se utiliza el modelo Raspberry Pi 3 Model B que se muestra en Figura 13 cuyas características están mostradas en la Tabla 8. Se ha utilizado este modelo y no el modelo posterior (Raspberry Pi 4) debido a que, por la escasez de microcontroladores, su precio ha aumentado en gran medida, por ello la opción más económica y de igual aplicación es la Raspberry Pi 3 Model B.

Se trata de un ordenador de bajo coste constituido por una placa única y que está basado en Linux y ARM. Raspberry Pi ha sido creado por la fundación Raspberry Pi en asociación con la empresa Broadcom y su función inicial era como herramienta de aprendizaje para el fomento del interés por la informática entre los niños de edad escolar pero poco a poco su función se ha ampliado considerablemente, tanto es que actualmente sus aplicaciones más comunes son: servidor web, controlador de robots, sistemas de seguridad, servidor de juegos y muchos más.



Figura 13. Raspberry Pi 3 Model B

Tabla 8. Características Raspberry Pi 3 Model B. [43]

Procesador:	Quad Core 1.2GHz Broadcom BCM2837 64bit CPU
Memoria RAM:	1GB RAM
Puertos:	4 x USB 2.0 HDMI Salida de vídeo tipo Jack de 3,5 mm
Red:	Ethernet RJ45 10/100 BaseT
GPIO:	40-pin extended GPIO
Alimentación:	+5V a 3A vía conector hembra MicroUSB
Bluetooth:	BCM 43438 LAN inalámbrico y Bluetooth Low Energy (BLE)
Almacenamiento:	Puerto Micro SD para cargar el sistema operativo y almacenar datos

3.2. ESP32

Para comunicar a los sensores con la Raspberry Pi de una forma sencilla y directa se optó por hacerlo por Wi-Fi.

Para realizar esta comunicación se valoró la posibilidad de utilizar tanto un ESP8266 como un ESP32 debido a las altas prestaciones y bajo coste que poseen, además de que se encuentran disponibles en el mercado. En la Tabla 9 se muestra las diferentes características de ambos.

Tabla 9. Características ESP32 y ESP8266

	ESP32	ESP8266
MCU	Xtensa Dual-Core 32-bit LX6	Xtensa Single-Core 32-bit L106
Bits	32 bits	32 bits
Velocidad	160 MHz (Hasta 240 MHz)	80 MHz (Hasta 160MHz)
SRAM	512 kB	160 kB
SPI FLASH	Hasta 16 MiB	Hasta 16 MiB
Alimentación	2.2 V a 3.6 V	3.0 V a 3.6 V
Rango de temperatura	-40°C a 125°C	-40°C a 125°C
Consumo DeepSleep	2.5 uA	20 uA
Consumo de corriente	80 mA	80 mA
Wi-Fi	802.11 b/g/n HT40	802.11 b/g/n HT20
GPIO	34	17
UART	3	2
I2C	2	1
SPI	4	2
CAN-Bus	Sí	No
Precio	3€ - 6€	4€ - 6€

Como podemos observar el ESP32 posee un procesador de doble núcleo y, por tanto, más potente además de que posee más memoria, Bluetooth BLE y reloj de tiempo real.

En cuanto a las entradas y salidas se puede ver que el ESP32 posee 36 pines GPIO y el ESP8266 17 además de que el ESP32 está más avanzado en cuestión de cantidad de buses de comunicación ya que posee 4 UART, 2 I2C, 4 SPI y 1 CAN BUS 2.0.

Por otra parte, en este proyecto se busca una aplicación óptima y si se alimenta el microcontrolador de forma externa con una batería se debe tener en cuenta todas las acciones que promuevan el ahorro de batería, por ello podemos comprobar que, aunque el consumo de corriente es el mismo en ambos, hay que tener en cuenta el consumo en DeepSleep. Como se puede observar el ESP32 tiene un consumo de 2.5uA frente a los 20 uA del ESP8266.

Analizando la tabla de características y las ventajas que presenta el ESP32 frente al ESP8266 finalmente se decidió trabajar con el ESP32.

El ESP32 es un microcontrolador, para ser más concretos un SoC (System of Chip), como el ESP8266, diseñado por la compañía china Espressif y fabricado por TSMC.

La característica principal de este SoC es que dispone de conectividad Bluetooth y Wi-Fi integrada lo que hace que su campo de aplicación sea muy amplio. En este proyecto se programa este dispositivo directamente desde la herramienta ESPHome en Home Assistant pero es posible emplear el IDE de Arduino para su programación.

La Figura 14 muestra el kit de desarrollo de ESP32 creado por la empresa Espressif. Aunque ya se han explicado en la tabla anterior las características principales que existen en el ESP32 y ESP8266 en la Tabla 10 se exponen en más detalle las características del ESP32. Este kit es el más habitual en el desarrollo de pequeños proyectos.

Todas las pruebas iniciales de funcionamiento y montaje para el posterior desarrollo se han realizado a través de una protoboard y con el módulo de desarrollo de la Figura 14.



Figura 14. ESP32DevKit

Tabla 10. Características ESP32. [44]

CPU:	Xtensa® Dual-Core LX6 de 32 bits
Memoria:	ROM: 448 KBytes
	RAM: 520 Kbytes
	Flash: Externa hasta 16MiB
GPIO:	32 pins GPIO
Conversores:	Convertor analógico digital (ADC) de 12bits y 18 canales
	2 conversores digital analógico (DAC) de 8bits
Bluetooth:	Bluetooth Low Energy (BLE) 4.2
Wi-Fi:	Wi-Fi Direct (P2P), P2P Discovery, modo P2P Group Owner y P2P Power Management

Una vez comprobado y testeado el funcionamiento del sistema desde la protoboard se pensó en un diseño final más práctico y profesional a través de la creación de una placa PCB como se explicará a lo largo de este documento.

Para este diseño final de la placa que contiene todos los adaptadores de los sensores que se quieran utilizar finalmente se ha usado el controlador ESP32-WROOM-32 (Figura 15) debido a que gracias a que puede ser programado de forma externa y alimentado a través de un regulador lineal por una batería se reduce notablemente el consumo de la batería.



Figura 15. ESP32-WROOM-32

La Figura 16 y la Figura 17 son los esquemas del ESP32-WROOM-32 que podemos encontrar en la página web de Espressif.

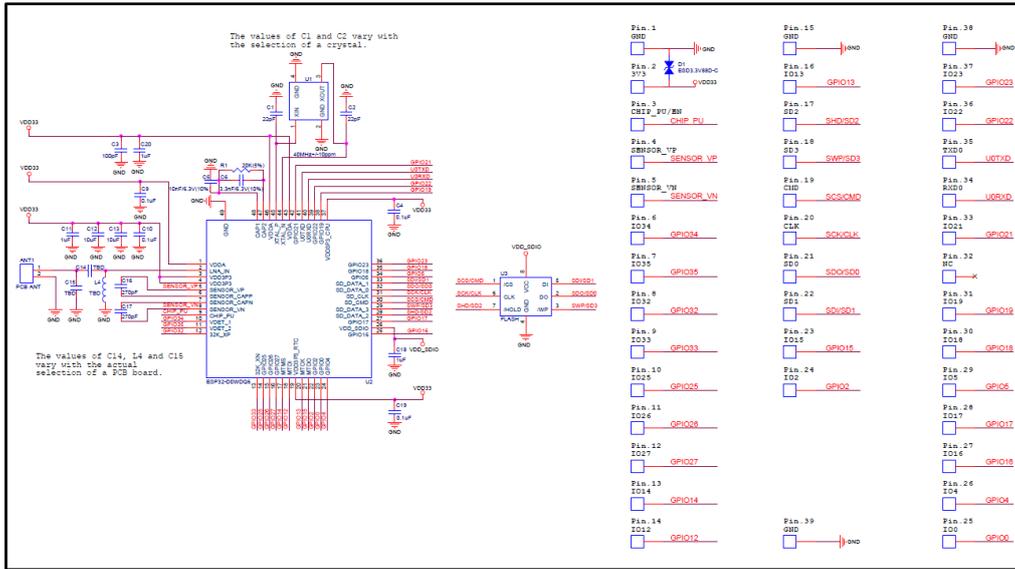


Figura 16. Esquemático ESP32-WROOM-32

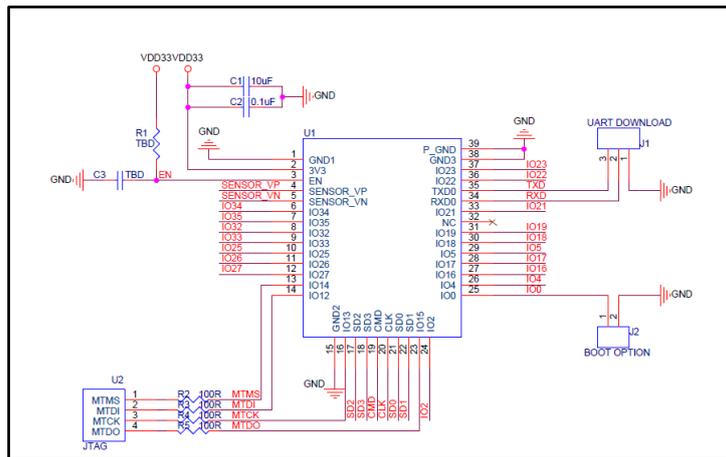


Figura 17. Esquema de periféricos ESP32-WROOM-32.

3.3. Sensores de bajo coste

3.3.1. Termopar tipo K

Para la medición de la temperatura se ha optado por el uso de termopares debido a su bajo coste y a sus amplios rangos de temperatura y durabilidad.

Debemos saber que un termopar es un sensor diseñado para la medición de la temperatura y está basado en los efectos termoeléctricos, esto da a lugar a que pueda medir temperaturas superiores a los 2000°C e inferiores a los -250°C dependiendo del material que esté fabricado.

Este sensor está constituido por un circuito formado por dos conductores de metales o aleaciones de metales diferentes que se encuentran unidos en uno de los extremos, estos cables están contenidos en un tubo de metal y aislados eléctricamente del tubo exterior para evitar que cualquier corriente altere la lectura de la temperatura.

Cuando en la unión formada existe una diferencia de temperatura lo que ocurre es que se origina una fuerza electromotriz, a esta fuerza se le conoce como el efecto Seebeck, que se presenta en forma de tensión eléctrica de un valor muy pequeño y que la mayoría de los dispositivos no son capaces de leer, es por ello por lo que necesitan un controlador.

El uso de termopares tiene una aplicación bastante extensa y existen para ello diferentes tipos de termopares como podemos ver en la Tabla 11, que varían en función de la composición de sus dos cables.

Tabla 11. Tipos de termopares y su composición. [45]

Termopar tipo J	Elemento positivo: Hierro Elemento negativo: Cobre 55% y Níquel 45%
Termopar tipo K	Elemento positivo: Níquel 90% y Cromo 10% Elemento negativo: Níquel 95%, Manganeso 2%, Aluminio 2% y Silicio 1%
Termopar tipo T	Elemento positivo: Cobre 100% Elemento negativo: Cobre 55% y Níquel 45%
Termopar tipo R	Elemento positivo: Platino 87% y Rodio 13% Elemento negativo: Platino 100%
Termopar tipo S	Elemento positivo: Platino 90% y Rodio 10% Elemento negativo: Platino 100%

Dado que la aplicación de este sensor en este proyecto es la monitorización de la temperatura de la vitrocerámica, se van a trabajar con temperaturas altas durante un tiempo prolongado. Es por ello que se debe de tener en cuenta el rango de temperaturas, composición de los materiales e impacto económico debido a que se está trabajando con sensores de bajo coste.

De esta manera, el termopar que más se adecúa para esta aplicación es el Termopar tipo K como el que se puede ver en la Figura 18.

Esto es así debido a que está compuesto por Níquel-Cromo y Níquel-Alumel, gracias a esta composición de Níquel posee una alta resistencia a la corrosión. Por otro lado, posee el mayor rango de operación de temperatura en lo que podemos denominar termopares económicos y su uso es apropiado para atmósferas oxidantes e inertes. Se trata de un

sensor de temperatura base metal que se utiliza frecuentemente para las altas temperaturas.



Figura 18. Termopar tipo K

3.3.2. MAX6675

El dispositivo MAX6675, mostrado en Figura 19, se trata de un controlador que sirve para digitalizar la señal de un termopar tipo K y que para ello utiliza un bus SPI para las comunicaciones.

Es capaz de medir temperaturas de entre 0°C y +1024°C utilizando 12 bits con una resolución de 0.25°C y posee un margen de error en el entorno de $\pm 3^\circ\text{C}$.

Posee un tiempo de conversión de entre 170 ms y 220 ms con el inconveniente de que no es posible saber cuándo ha finalizado la conversión, es por ello por lo que se suele dejar un tiempo superior al máximo, aproximadamente 250 ms, que es un valor seguro para trabajar.

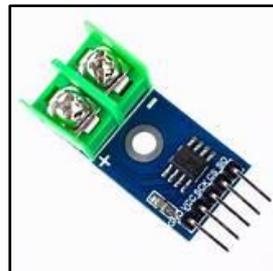


Figura 19. Módulo MAX6675

3.3.3. Células de carga

Una célula de carga es un transductor que es capaz de convertir la fuerza aplicada sobre ella en una señal eléctrica medible. Las células de carga consisten en una pieza de metal a la que se incorporan unos medidores (galgas extensiométricas).

Las células de carga (Figura 20) son sensores cuya resistencia eléctrica interna varía en función de la fuerza o presión aplicada sobre el mismo. Su funcionamiento se basa en el efecto piezorresistivo. Se trata de la variación de la resistencia eléctrica de un material, sobre todo en semiconductores, en función de la tensión mecánica que se le aplica.

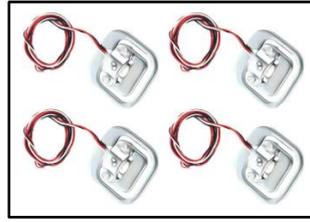


Figura 20. Células de carga

La variación de resistencia eléctrica que se produce es medible pero no de forma directa, por lo que se debe de conectar a un circuito eléctrico que permita medir esta variación en forma de tensión.

La conexión a un circuito eléctrico se debe a que es necesario conocer los parámetros eléctricos a la salida de los transductores y para ello es necesario conectar un circuito de acondicionamiento de la señal de salida.

Existen varios circuitos de acondicionamiento de la señal, en este caso a estar trabajando con variaciones muy pequeñas de resistencia el circuito más adecuado es un puente de medida o un puente de Wheatstone ya que permite aumentar la sensibilidad empleando sensores múltiples, así como, compensar determinadas interferencias.

Existen diversos tipos de montaje del circuito en puente Wheatstone en función del número de galgas que se vayan a conectar.

Podemos encontrar el montaje en cuarto de puente (Figura 21), montaje en medio puente (Figura 22) y montaje en puente completo (Figura 23).

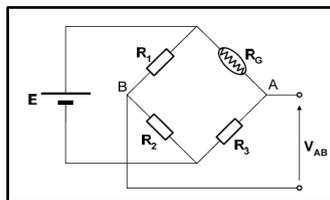


Figura 21. Montaje en cuarto de puente.

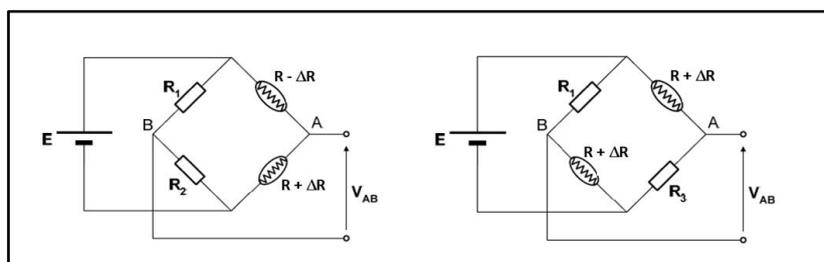


Figura 22. Montaje en medio puente.

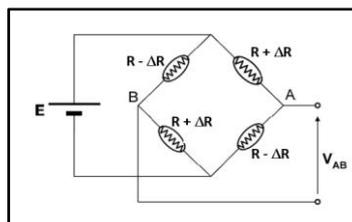


Figura 23. Montaje en puente completo.

Para este proyecto se utilizará el montaje en puente completo ya que es el más utilizado y adecuado para aplicaciones de pesaje.

3.3.4. HX711

El HX711 (Figura 24) se trata de un convertidor analógico-digital de 24 bits de alta precisión que está diseñado para la realización de básculas de pesaje y aplicaciones de control industrial, posee internamente la electrónica para la lectura del puente de Wheatstone que se forma con las células de carga.

Actúa como transmisor entre las células de carga y un microcontrolador como puede ser Arduino o un ESP32, permitiendo así realizar una lectura de manera sencilla del peso de la celda y con las especificaciones técnicas que se indican en la Tabla 12.

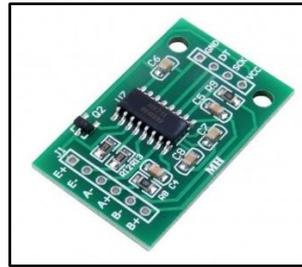


Figura 24. Módulo HX711

Tabla 12. Especificaciones técnicas HX711[46]

Voltaje de operación:	5V DC
Consumo de corriente:	10mA
Voltaje de entrada diferencial:	±40mV
Resolución de la conversión A/D:	24 Bits
Frecuencia de lectura:	80Hz

Posee un multiplexor a la entrada que permite realizar una selección entre los canales A o B de entrada diferencial que luego se amplifica y pasa por el convertor A/D como se puede observar en la Figura 25. Las diferencias entre la utilización de estos dos canales está en la ganancia que posee cada uno, el canal A posee unas ganancias de 64 y 128 mientras que el canal B posee una ganancia de 32.

La información se comunica a través de 2 pines DOUT y PD_SCK al controlador que se vaya a utilizar.

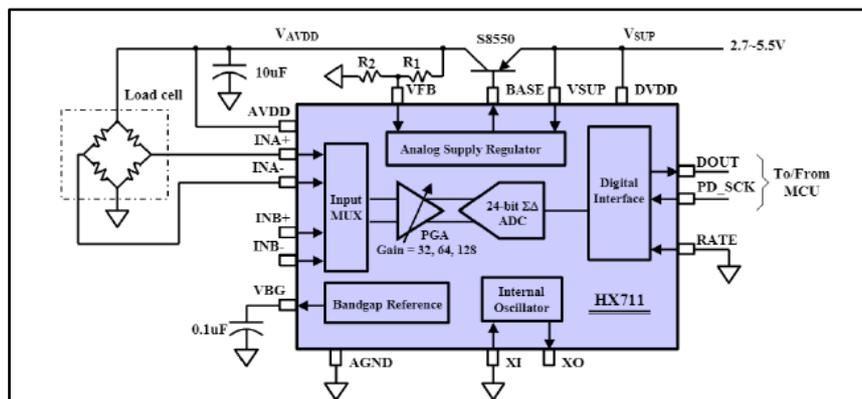


Figura 25. Diagrama de aplicación típica para el HX711

3.3.5. EFS-04P

El EFS-04P, Figura 26, es un sensor de flujo o interruptor de caudal, son equipos de monitoreo de fluidos en tuberías que funcionan con el desplazamiento de un pistón magnético que indica el paso o no de flujo de líquido.



Figura 26. EFS-04P

3.3.6. Sensor de presión de asiento

Un sensor de presión es aquel que es capaz de detectar una fuerza que se ejerce sobre un punto determinado de una superficie, o lo que es lo mismo, detectar si se ha ejercido presión sobre la superficie. Cuando se ejerce una fuerza, se produce un cambio de dimensión de la superficie que converge en una salida eléctrica que es la que indica que se ha ejercido una fuerza.

En este proyecto se utilizará un sensor de presión de asiento universal (Figura 27) que es muy utilizado en el mundo de la automoción para la detección de ocupación en el asiento del vehículo para activar el aviso de los cinturones de seguridad.

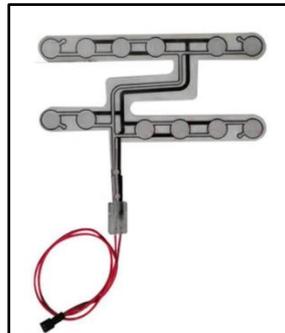


Figura 27. Sensor de presión de asiento.

3.4. Home-Assistant

Todos los dispositivos seleccionados anteriormente deben de estar integrados en una plataforma de automatización del hogar. Por ello se ha realizado un estudio de las diferentes plataformas como son:

- **OpenHAB:** Es un software de código abierto para la automatización del hogar muy conocido ya que su principal ventaja es independiente de la nube si se desea trabajar con mayor privacidad.
- **Domoticz:** Este sistema es ligero y sencillo, permitiendo el control de luces, interruptores y sensores. Posee una interfaz bastante simple para la integración de los dispositivos
- **Home Assistant:** Este es un software de código abierto que se ejecuta en Python 3. Se trata de un software relativamente actual pero que se encuentra en constante desarrollo por su comunidad. Trabaja de manera local ya que fue desarrollado para la seguridad y el confort.

Finalmente se ha optado por el uso de la plataforma Home Assistant, esto es así por la principal ventaja de que solo trabaja en local, posee numerosos servicios y prestaciones además de que posee una interfaz gráfica bastante intuitiva y desde la que se puede tener acceso desde diferentes dispositivos con acceso a internet.

El montaje de Home Assistant se puede hacer en numerosos dispositivos como se puede ver en la Figura 28, aunque, como iniciación y como se va a realizar en este proyecto, la instalación más recomendada y utilizada es en una Raspberry Pi.

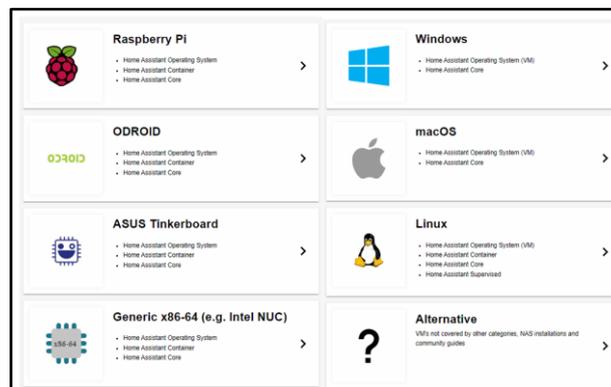


Figura 28. Dispositivos disponibles para la instalación de Home Assistant.[47]

Para la instalación, Home Assistant ofrece diferentes métodos. Los métodos que recomienda para su instalación son:

- **Home Assistant Operating System:** Sistema operativo mínimo optimizado para potenciar el Asistente Doméstico. Viene con el Supervisor para gestionar el núcleo del Asistente Doméstico y los complementos.
- **Home Assistant Container:** Instalación independiente basada en contenedores del núcleo del asistente doméstico (por ejemplo, Docker).

A parte de estos métodos de instalación también están disponibles otros métodos recomendados para usuarios con más experiencia como son:

- **Home Assistant Supervised:** Instalación manual del Supervisor

- **Home Assistant Core:** Instalación manual utilizando el entorno virtual de Python.

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

Figura 29. Comparación entre los distintos métodos de instalación de Home Assistant. [47]

3.5. ESPHome

ESPHome es una herramienta de Home Assistant cuyo objetivo es facilitar la gestión de las placas ESP. Su funcionamiento se basa en leer un fichero de configuración YAML y crear un *firmware* personalizado que se puede subir al módulo ESP32 o ESP8266 con los sensores que se quiera conectar, de esta manera los dispositivos y sensores añadidos podrán mostrarse y monitorizarse desde la interfaz de usuario de Home Assistant. [48]



Figura 30. Página principal ESPHome.

En este proyecto se utilizarán sensores y módulos cuya programación básica para su lectura está integrada en esta herramienta y que se puede encontrar de manera accesible desde la página oficial de ESPHome.[49]

En apartados posteriores se explicará la programación de cada sensor y en cómo ha sido basada en esta programación básica proporcionada.

3.6. Diptrace

Diptrace es una plataforma software de captura esquemática y de diseño PCB que ofrece todas las herramientas necesarias para crear placas simples o complejas de múltiples capas, desde los esquemas a los archivos de fabricación. [50]

Las características de este software se presentan en la Tabla 13 :

Tabla 13. Características Diptrace [50]

Características Diptrace.	Interfaz de usuario sencilla.
	Esquemas de múltiples jerarquías.
	Enrutador automático de alta eficiencia.
	Herramientas avanzadas de trazado manual.
	Señales de alta velocidad.
	Verificación avanzada en tiempo real (DRC).
	Previsualización de PCB en 3D y exportación 3D.

Dentro de Diptrace existen cuatro subpartados de trabajo que son [50] :

- **Captura esquemática:** Se trata de una herramienta para crear esquemáticos con un diseño jerárquico de multi-hoja y multi-nivel que permite la conexión de pines visualmente, es decir, con puertos de red, buses o lógicamente. Durante todo el proceso de creación del esquemático se realiza la comprobación de reglas eléctricas y la verificación jerárquica.
- **Diseño de PCB:** Esta herramienta está pensada para el diseño de placas de circuitos impresos, el PCB Layout permite realizar un enrutamiento y colocación inteligente, así como, un enrutamiento inteligente, copia de bloques jerárquicos y características de verificación para lograr una buena precisión. Incluye DCR en tiempo real que permite corregir errores sobre la marcha y así lograr una mayor calidad del producto.
- **Creación de bibliotecas:** Se trata de un sistema de gestión de bibliotecas con editores de componentes e importación directa desde el exterior del entorno DipTrace. Se incluye en esta herramienta la creación de plantillas integradas para la creación rápida de componentes y patrones.
- **Modelos 3D:** Funcionan como parte del PCB Layout. Esta herramienta permite al diseñador previsualizar la placa con los componentes ya instalados en cualquier etapa del proyecto. Estos modelos 3D pueden ser creados desde la plataforma DipTrace o importados por archivos 3DS, VRML, STEP e IGES.

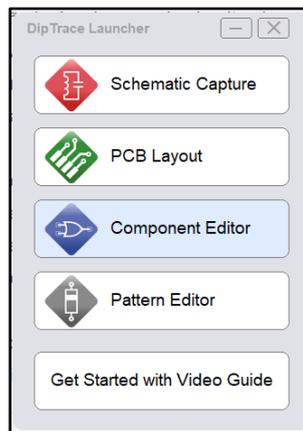


Figura 31. Menú principal DipTrace.

Se valoró la utilización de Altium Designer para el diseño con la licencia de estudiante de la que se tiene acceso desde la cuenta universitaria pero esta licencia solo permite trabajar con las librerías que te proporciona por defecto sin tener la opción de incluir nuevos componentes o una librería propia. Por ello, e ha utilizado este programa para el diseño del esquemático y PCB debido a que es un programa trabajado en una asignatura y el que se ha ido utilizando para los diferentes proyectos avenidos durante el grado además de que se trata de un programa muy intuitivo y de fácil manejo.

4. Métodos

4.1. Instalación y configuración Home Assistant

En primer lugar, se realizará la instalación de Home Assistant que, como se ha comentado anteriormente, es en una Raspberry Pi 3 Model B.

Para esta instalación es necesario grabar la imagen del sistema operativo en una tarjeta microSD como se explica en la página oficial de Home Assistant [47]

Una vez grabada la imagen y la tarjeta Micro SD está en el lector de tarjetas de la Raspberry Pi es necesario abrir la interfaz web a través de la dirección IP que tiene asignada. En este caso la dirección es la **192.168.1.224** como se puede ver en la Figura 32.

Estado	Nombre	IP	Fabricante	Dirección MAC	Comentarios
>	router.asus.com	192.168.1.1	ASUSTek COM...	AC:9E:17:45:03:60	
	192.168.1.33	192.168.1.33		E8:9F:80:E5:95:07	
	192.168.1.42	192.168.1.42		3A:FD:F5:32:14:2C	
	tiago-71c.GAIIRiteU	192.168.1.71	Intel Corporate	60:F6:77:B0:A4:6E	
	DESKTOP-30AB64E	192.168.1.79	Intel Corporate	FC:77:74:5F:AC:B2	
	192.168.1.106	192.168.1.106		E8:9F:80:E5:96:8D	
	LAPTOP-VUU9IVK8	192.168.1.131	Liteon Technol...	98:22:EF:CF:11:BB	
	development.utt.fr	192.168.1.139	Intel Corporate	64:5D:86:64:C4:45	
	192.168.1.174	192.168.1.174		32:CC:47:7C:41:A2	
	DESKTOP-M7JS7QL	192.168.1.197		0C:7A:15:19:73:96	
	HUAWEI_P20_lite-3574bcb47.GAIIRiteU	192.168.1.217	HUAWEI TECH...	C0:F4:E6:EA:F3:DC	
	homeassistant	192.168.1.224	Raspberry Pi Fo...	B8:27:EB:3A:F3:4A	
	DESKTOP-M7JS7QL	192.168.56.1		0A:00:27:00:00:04	

Figura 32. Dirección IP Home Assistant

Conocida la dirección IP, se procede a abrir en el navegador **192.168.1.224:8123**.

Aparece entre 2 y 5 minutos la pantalla de preparación para luego acceder a la pantalla de creación del usuario como se puede observar en la Figura 33.

Home Assistant

¿Estás listo para despertar tu casa, reclamar tu privacidad y unirse a una comunidad mundial de pensadores?

Comencemos creando una cuenta de usuario.

Nombre*

Nombre de usuario*

Contraseña*

Confirmar contraseña*

CREATE A ACCOUNT

Figura 33. Creación de cuenta de usuario en Home Assistant.

Una vez creada la cuenta el sistema operativo te da la opción de instalar los servicios que te ofrece Home Assistant antes de acceder a la pantalla principal (Figura 34) aunque en este caso se instala más tarde como se muestra en el apartado 4.2.

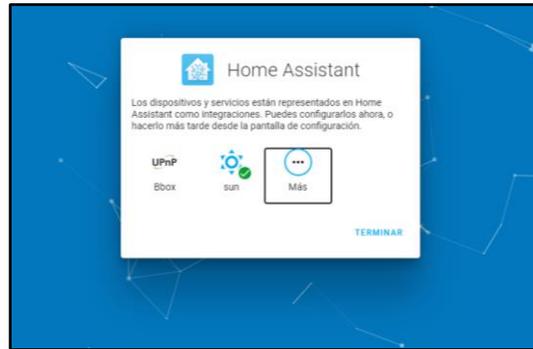


Figura 34. Instalación de servicios en Home Assistant,

Terminado el proceso ya se puede acceder a la página principal que es la que se muestra en la Figura 35.

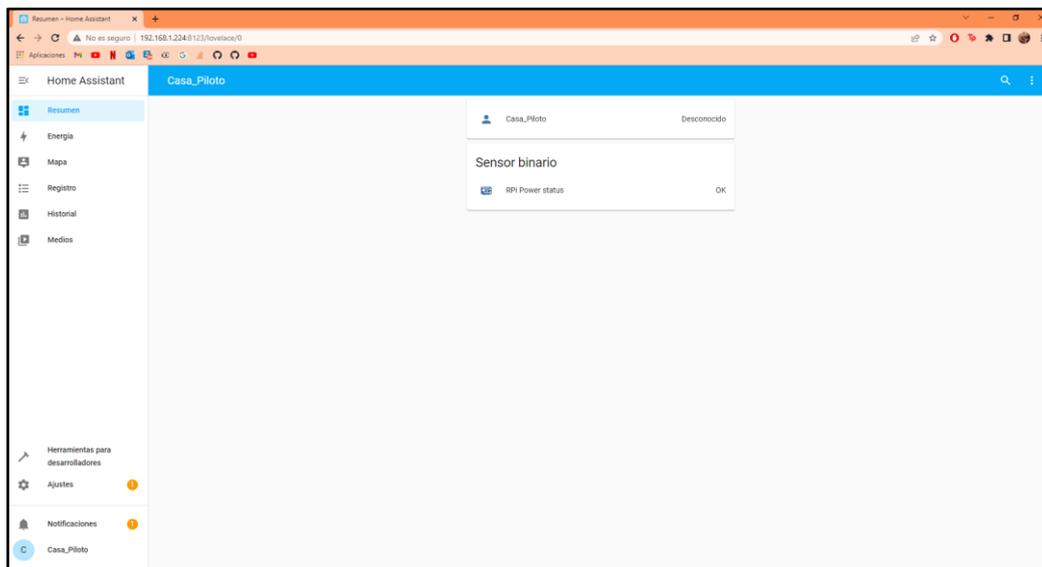


Figura 35. Página principal Home Assistant

4.2. Instalación y configuración de la herramienta ESPHome y ESP32

En este punto se explica la instalación de la herramienta ESPHome, así como la configuración del ESP32.

En versiones anteriores la instalación de ESPHome se hacía directamente accediendo a la tienda de componentes de Home Assistant donde se mostraban visibles los Add-on creados por la comunidad y desde ahí se podía realizar la instalación.

Como se puede observar en la Figura 36 este tipo de instalación está obsoleta (Imagen obtenida de la instalación de ESPHome en una máquina virtual). Esto es debido a que los creadores en la comunidad de este Add-on lo configurado de otra forma y actualmente la instalación debe de hacerse a través de un repositorio.

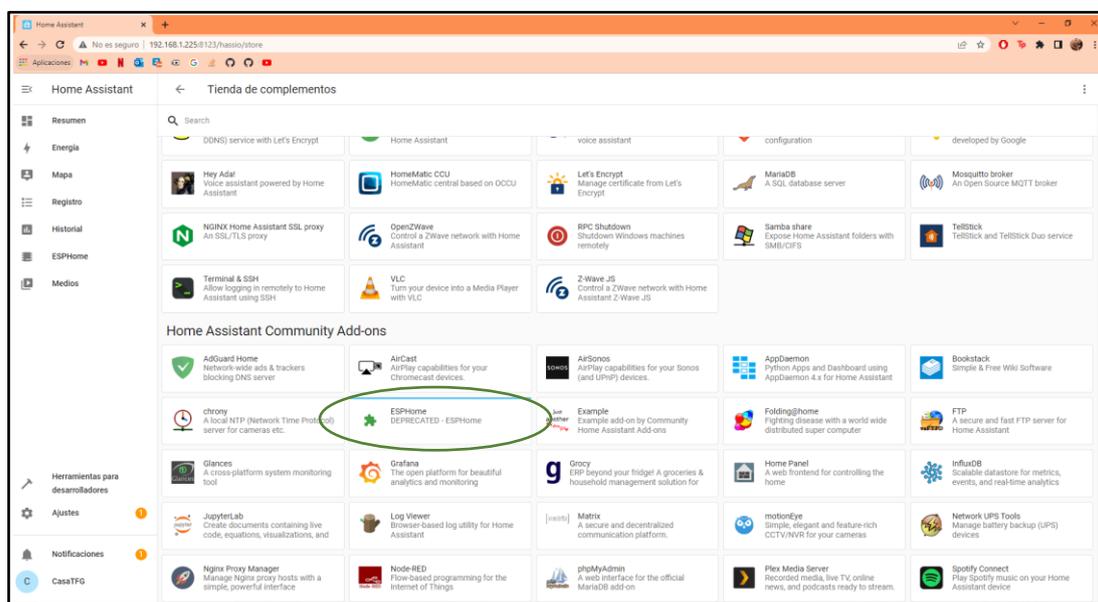


Figura 36. ESPHome obsoleto en máquina virtual con Home Assistant 7.6

Para ello se accede de la misma manera a **Ajustes -> Complementos -> Tienda de complementos**. En la tienda de complementos se ve en la esquina superior derecha tres puntos donde da la opción de instalar por repositorio. (Figura 37)

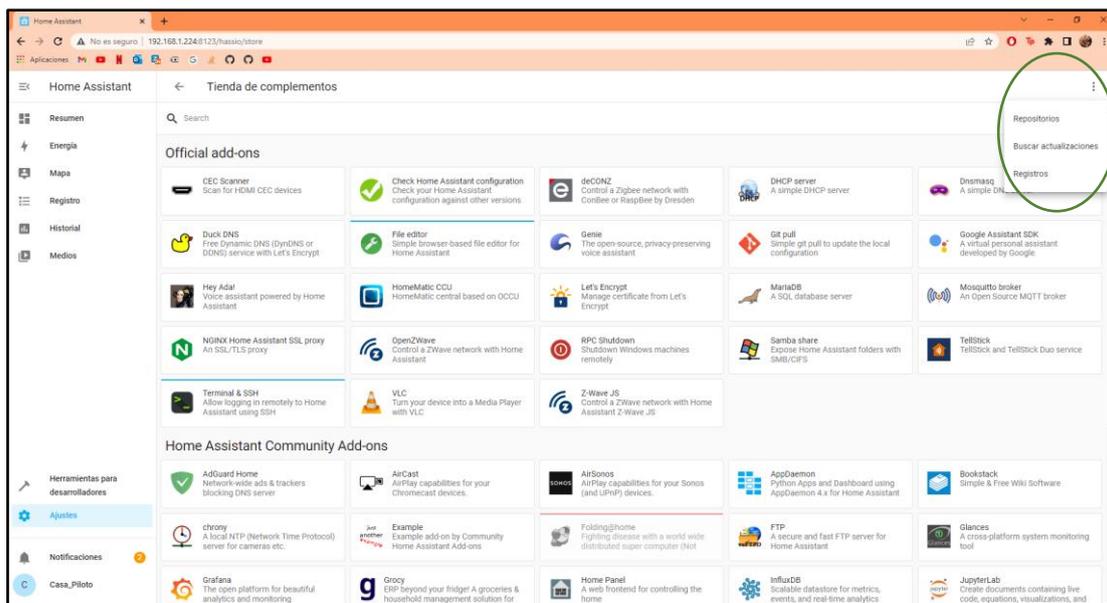


Figura 37. Tienda de complementos Home Assistant 8.2

La URL del repositorio utilizado es el siguiente: <https://github.com/esphome/home-assistant-addon>

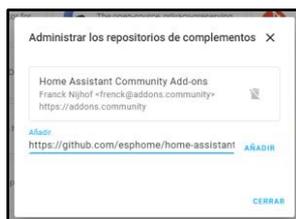


Figura 38. URL ESPHome

Una vez añadido se ve en la tienda de complementos que ya aparece el apartado de ESPHome (Figura 39) donde aparecerán tres apartados llamados ESPHome, ESPHome(beta) y ESPHome(dev). Para acceder a la pantalla de instalación es necesario hacer click en la primera opción.

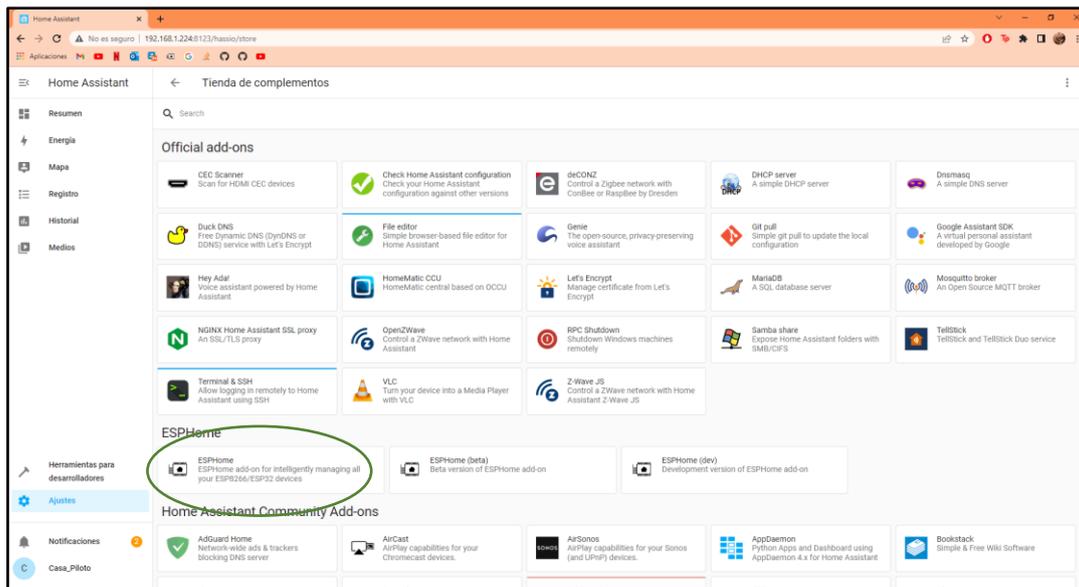


Figura 39. Tienda de complementos con ESPHome

En la Figura 40 se puede observar la página de información e instalación de ESPHome, por ello solo es necesario acceder desde el apartado INSTALAR.

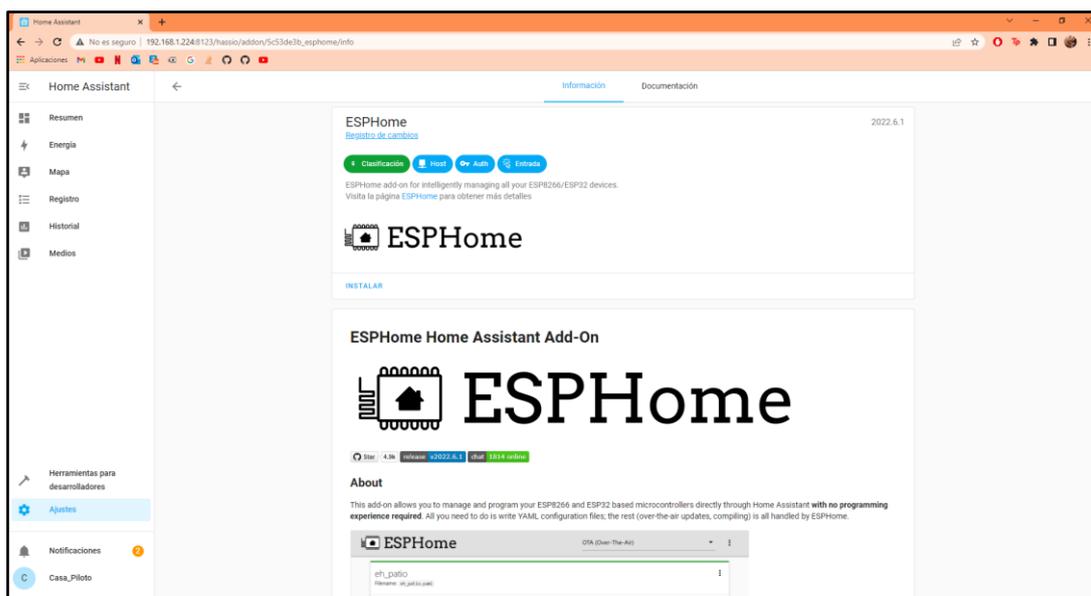


Figura 40. Página de información ESPHome

Finalmente instalado se puede ver en la pantalla de información, como se muestra en la Figura 41, acceso a la documentación, configuración, registro e interfaz web de la herramienta ESPHome.

Para que sea más sencillo y visual a la hora de acceder a la parte de programación de la herramienta se ha activado la opción de Mostrar en la barra lateral. De esta manera se tiene acceso a la interfaz web desde los ajustes de la herramienta o desde el acceso directo creado en el lateral izquierdo de la interfaz de Home Assistant.

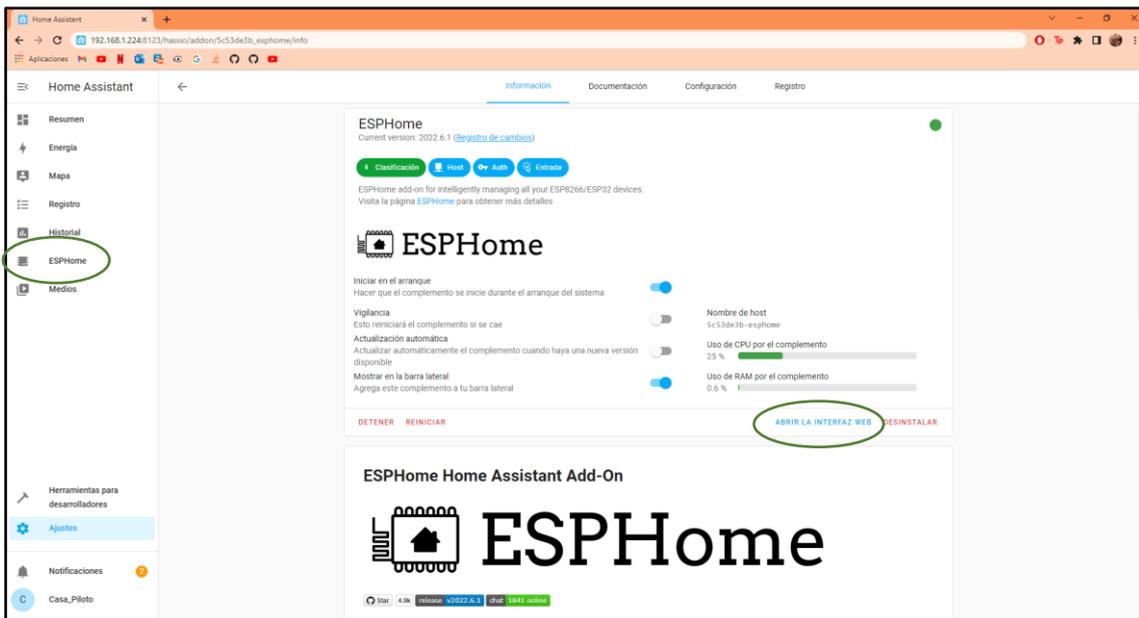


Figura 41. Página de información ESPHome

Completada la instalación de la herramienta es momento de comenzar con la configuración del ESP32 que se vaya a utilizar, para ello es necesario abrir la interfaz web de ESPHome (Figura 42) y hacer click en New Device.

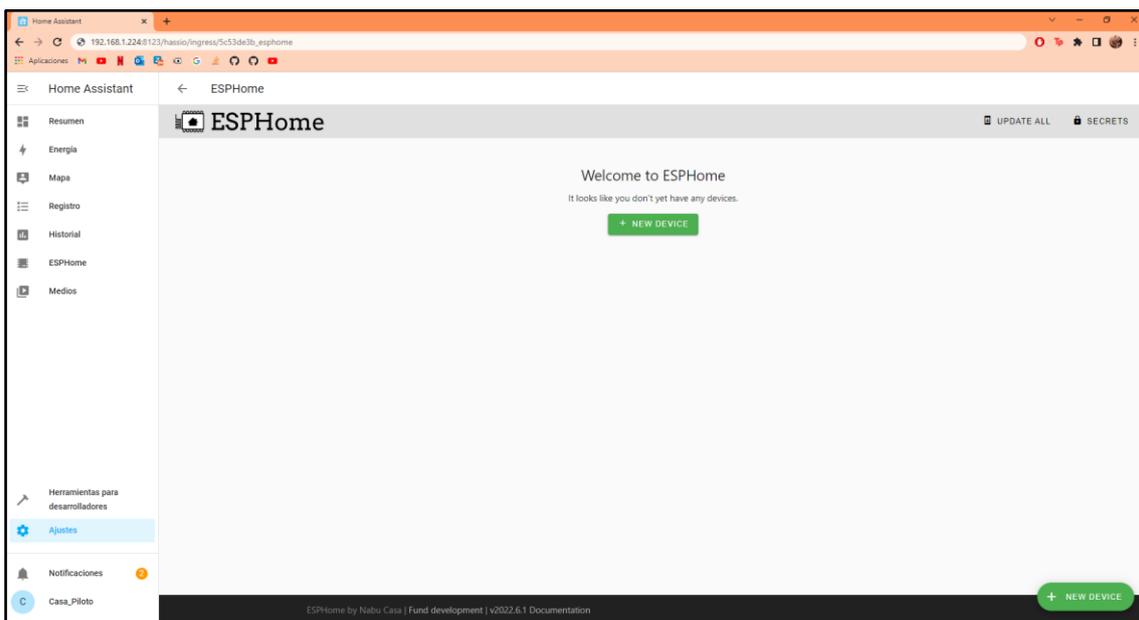


Figura 42. Interfaz ESPHome

Creando la configuración hay que poner asignar un nombre, en este caso al tratarse del ESP32 que estará destinado al dormitorio se llamará esp32-dormitorio y elegiremos el modelo de ESP que se vaya a utilizar que en este caso es el ESP32.

Figura 43. Elección nombre y modelo ESP32

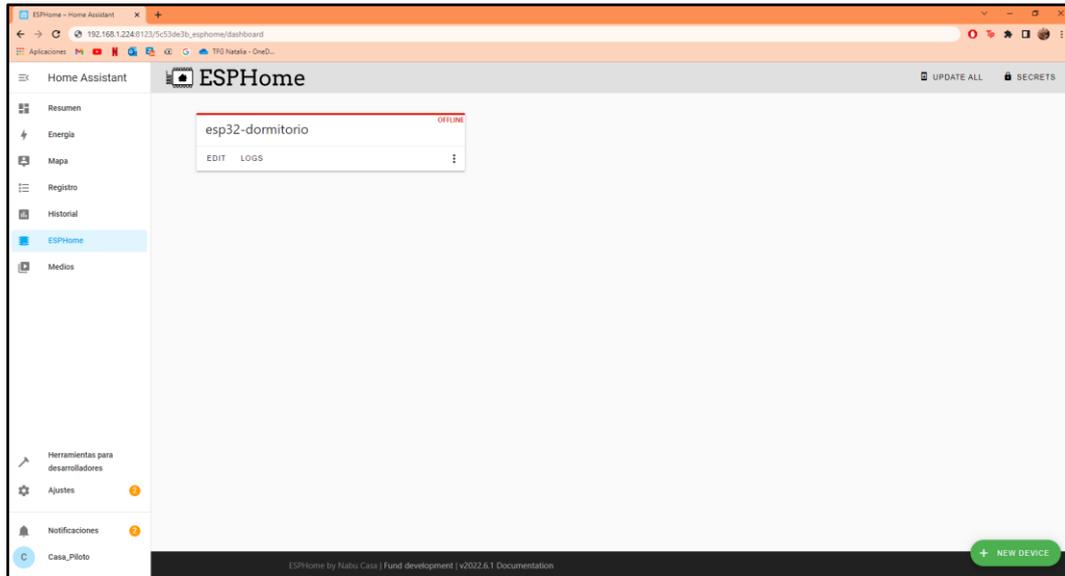


Figura 44. Interfaz ESPHome

Una vez creado el entorno, en este caso del dormitorio, se realiza la primera instalación. En este caso se ha realizado una instalación básica con la red Wi-Fi para conectar el ESP32 ya que la primera instalación hay que realizarla por un servidor web proporcionado por la herramienta ESPHome.

Al abrir el editor de esp32-dormitorio se observa un código base (Figura 45). Para conectar a la red es necesario completar los campos *ssid* con el nombre de la red y *password* con la contraseña correspondiente. Una vez estén los parámetros es momento de realizar la instalación.

```

esp32-dormitorio.yaml
1 - esphome:
2   name: esp32-dormitorio
3
4 - esp32:
5   board: esp32dev
6   framework:
7     type: arduino
8
9 # Enable logging
10 logger:
11
12 # Enable Home Assistant API
13 - api:
14   encryption:
15     key: "5AKkl+sY3Y88Xgptavp7/GdSP2NpASnY+FuIzoeIFvA="
16
17 - ota:
18   password: "2cf5cdb7e9eb696b0425e9c84b888b6"
19
20 - wifi:
21   ssid: !secret wifi_ssid
22   password: !secret wifi_password
23
24 # Enable fallback hotspot (captive portal) in case wifi connection fails
25 - ap:
26   ssid: "Esp32-Dormitorio"
27   password: "TprkH36ainzik"
28
29 captive_portal:
30

```

Figura 45. Código base ESPHome

Existen diferentes métodos de instalación en este caso se utiliza la opción *Plug into this computer* (Conectar a este equipo) debido a que es la primera instalación y más adelante cuando ya esté conectado el ESP32 a la red inalámbrica se hará por vía *Wireless* (Inalámbrica).

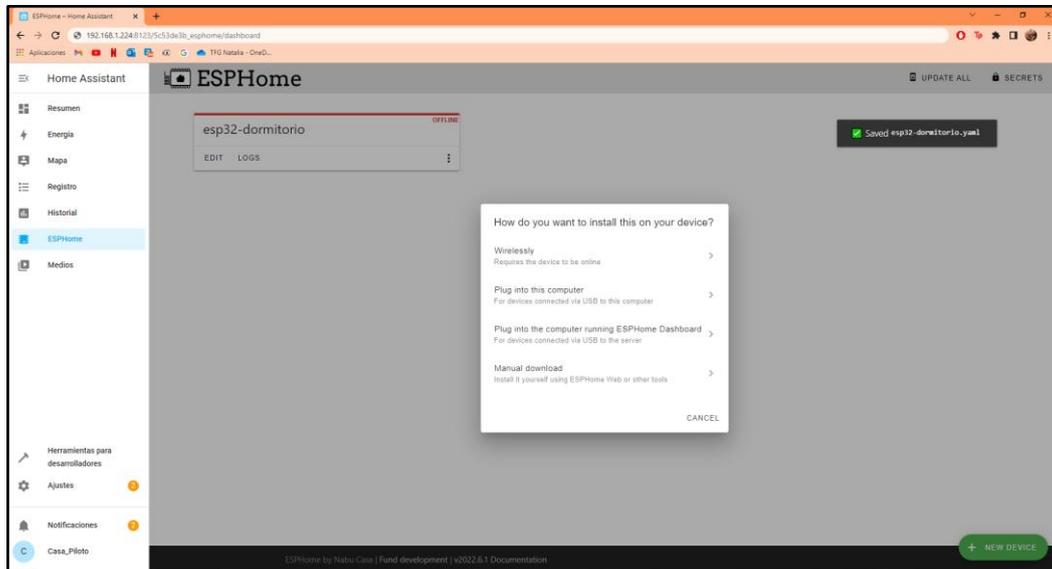


Figura 46. Elección del método de instalación del programa en ESP32

En la opción escogida se realiza la instalación desde el navegador web pero primero hay que descargar el archivo *.bin* que es el programa escrito llamado *esp-dormitorio-factory.bin*.

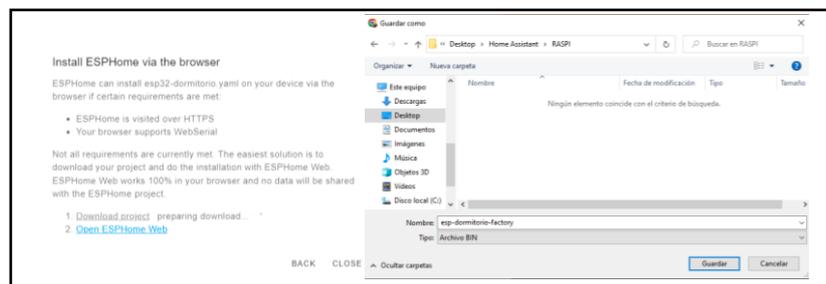


Figura 47. Descarga del proyecto en formato *.bin*

Teniendo los pasos previos completados es momento de realizar la instalación abriendo el apartado *Open ESPHome Web* y haciendo click en el apartado de *Connect* (Figura 48).

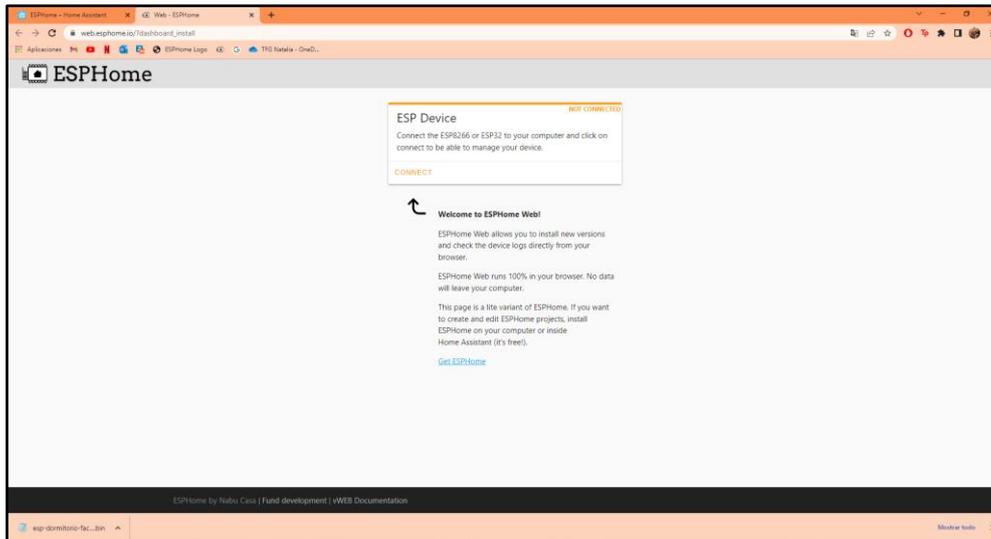


Figura 48. Interfaz web para la programación

Al iniciar la conexión se abre una ventana emergente que indica el puerto COM que se quiera utilizar. En este caso es el CP2102 COM3 que corresponde con el ESP32.

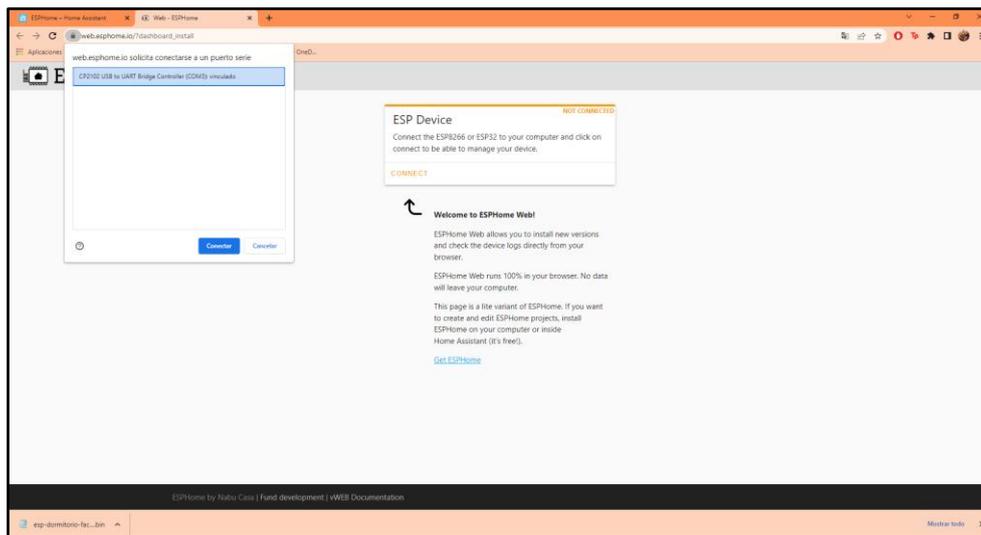


Figura 49. Selección puerto COM3 ESP32

Una vez seleccionado el puerto correspondiente es momento de instalar. Al instalar pide que se adjunte el archivo *.bin* que se va a utilizar. Se carga el archivo que previamente se ha descargado y al instalar hay que tener en cuenta que se debe mantener pulsado el botón BOOT del ESP32 hasta que comience la instalación como se ve en la Figura 51.

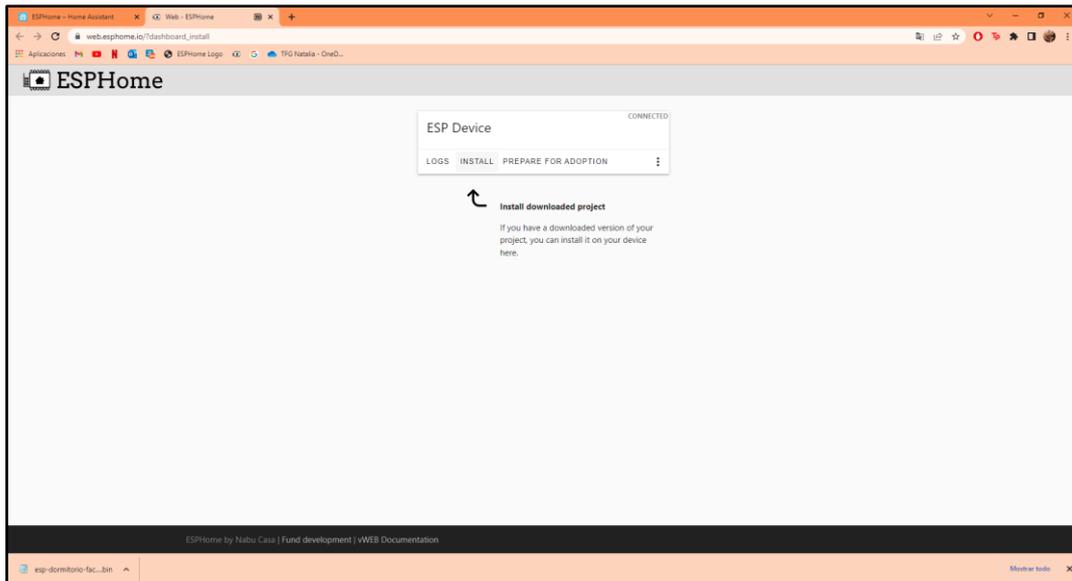


Figura 50. Página Web de instalación

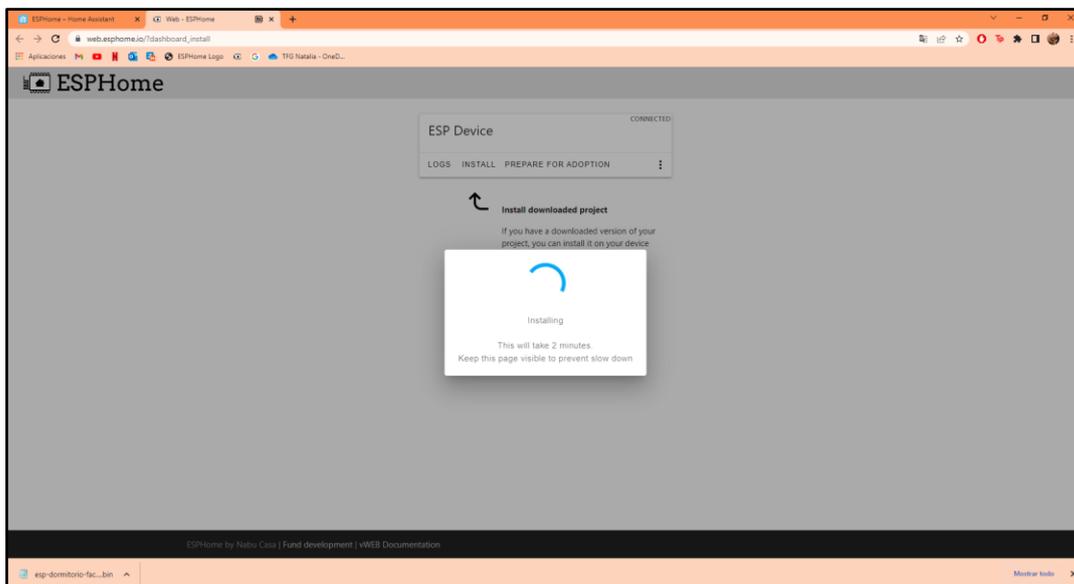


Figura 51. Instalación del programa en ESP32

Una vez realizada la primera instalación el ESP32 ya se encuentra conectado a la red a la que se esté trabajando. Esto hace que para posteriores instalaciones no sea necesario realizar todo este proceso y solo sea necesario seleccionar la opción *Wireless*.

Para poder realizar esta conexión *Wireless* es necesario localizar la dirección IP del dispositivo ESP32 que se esté configurando. El nombre asignado en la herramienta ESPHome corresponde al nombre del ESP32 conectado a la red, esto quiere decir que al buscar se encontrará como esp32-dormitorio y, en este caso, corresponde con la dirección **192.168.1.92**.

Hay que incluir esta dirección IP manualmente en el apartado de conexión como se muestra en la Figura 52.

```

20 ▾ wifi:
21     ssid: ..
22     password:
23
24 ▾ manual_ip:
25     static_ip: 192.168.1.92
26     gateway: 192.168.1.1
27     subnet: 255.255.255.0
  
```

Figura 52. Dirección IP ESP32

Para la conexión inalámbrica es necesario pulsar el botón BOOT del ESP32 la primera vez que se realiza, a partir de las siguientes el código se carga automáticamente en el ESP32.

La explicación anterior está diseñada para un espacio de la vivienda, el dormitorio. Ya que el proyecto está pensado para la colocación de sensores en diferentes áreas de la vivienda es necesario realizar la instalación de un ESP32 por área.

Esto es posible ya que la herramienta permite incluir más de un dispositivo siguiendo los mismos pasos que han sido explicados en este punto.

La implementación real de este sistema se ha hecho en una vivienda que dispone de salón y cocina compartida por lo que el segundo dispositivo añadido es llamado esp32-saloncocina y el tercero corresponde a esp32-aseo, situado en el cuarto de baño de la vivienda como se observa en la Figura 53.

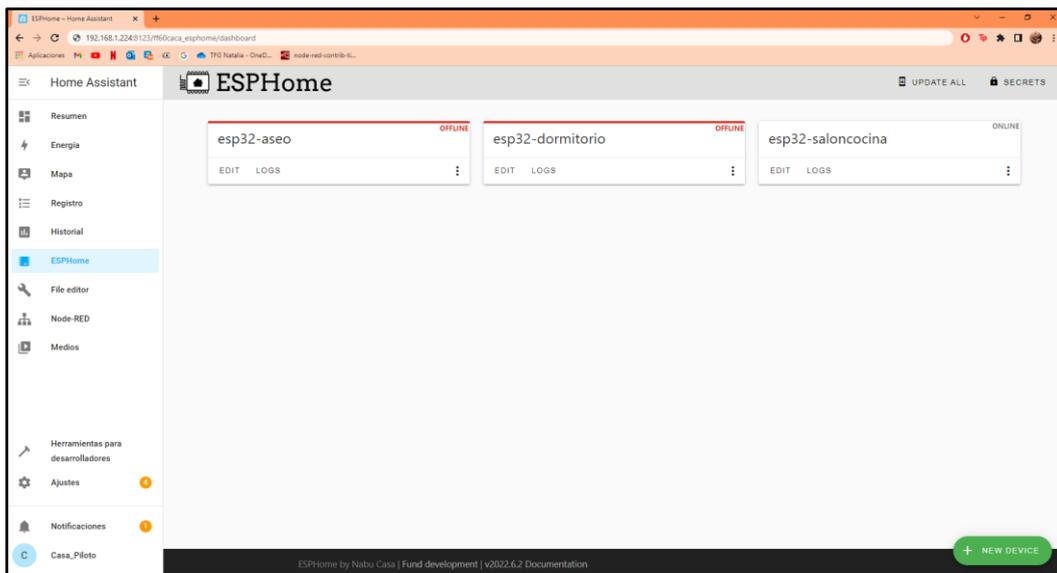


Figura 53. ESP32 de toda la vivienda integrados en ESPHome

4.3. Conexión de sensores

A continuación, se explica la conexión física realizada para cada sensor.

4.3.1. Células de carga y HX711

Como se ha comentado en apartados anteriores, para la conexión de las células de carga se ha utilizado una conexión de puente de Wheatstone en puente completo dado que esta configuración es la utilizada para las aplicaciones de pesaje.

Cada célula de carga está compuesta por tres cables de color rojo, blanco y negro como se ve en la Figura 54.



Figura 54. Célula de carga

La distribución de la conexión se observa en la Figura 55 en donde se han distribuido las células de carga como UL (*Upper left*, Arriba Izquierda), UR (*Upper Right*, Arriba Derecha), LL (*Lower Left*, Abajo Izquierda) y LR (*Lower Right*, Abajo Derecha) que es la distribución que se ve desde el punto de vista del usuario.

Tanto las células de carga UL y UR como LL y LR están conectadas entre sí por el cable blanco y las conexiones entre UL y LL, así como entre UR y LR, se realizan con el cable negro dejando así cada cable rojo para la conexión con el HX711.

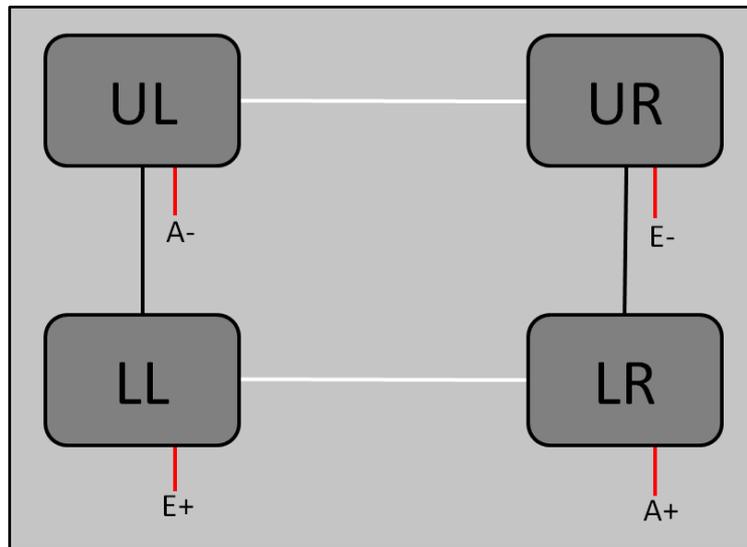


Figura 55. Conexión células de carga

La Figura 56 representa la conexión entre el HX711 y las células de carga y el ESP32. Como se ha hablado anteriormente podemos ver la conexión a las células de carga que coincide con el esquema de la Figura 55.

Los pines E+ y E- corresponden a los pines de excitación conectados a las células de carga. Por otra parte, existen dos canales para la conexión: el canal A (A+, A-) y el canal B (B+, B-).

La diferencia entre estos dos canales es que el canal A trabaja a una ganancia de amplificación de 64 o 128, en cambio, el canal B trabaja a una ganancia de 32. Es por ello que para este proyecto se ha realizado la conexión final con el canal A.

En relación con la conexión entre el HX711 y el ESP32 tenemos el pin GND que está conectado a masa y el pin VCC que es la alimentación y va a 3.3V. Por otro lado, está el pin DT que corresponde a la salida de datos en serie que está conectado al pin GPIO14 y el pin SCK que corresponde al control de apagado y entrada de reloj serial conectado al pin GPIO12.

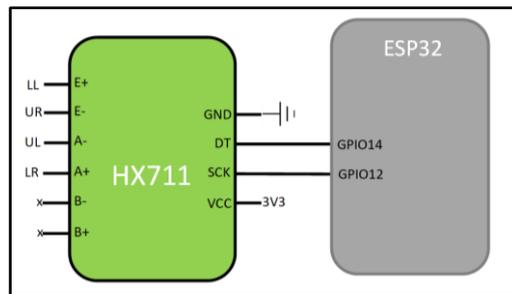


Figura 56. Conexión HX711 a células de carga y ESP32

4.3.2. Termopar tipo K y MAX6675

El termopar tipo K utilizado posee un conector indicativo de los polos del termopar como se observa en la Figura 57.

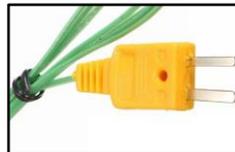


Figura 57. Conector termopar tipo K.

A continuación, en la Figura 58, se muestra la conexión realizada para el MAX6675. Como se ha comentado en puntos anteriores el MAX6675 es compatible con el bus SPI por lo que la conexión está basada en el esquema del ESP32. El pin SCK que corresponde con la entrada de reloj serial está conectado al pin GPIO18, el pin SO que corresponde con la salida de datos está conectado al pin GPIO19 y, finalmente, el pin CS que corresponde con la elección de chips está conectado al pin GPIO2.

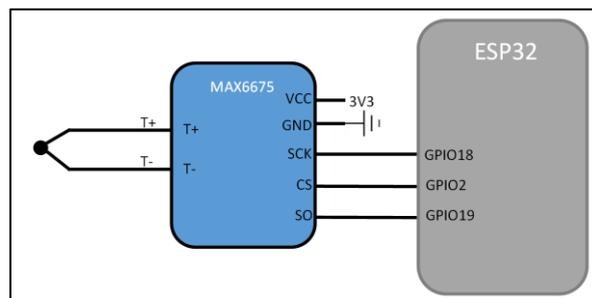


Figura 58. Conexión MAX6675 a ESP32

4.3.3. Sensores binarios

Dentro de este punto se incluyen los sensores de caudal y de presión de asiento que se conectan de la misma manera. Estos sensores que disponen de dos cables solo necesitan conectar uno a la entrada correspondiente en el ESP32 y el otro a GND.

Para elegir los pines correctamente hay que tener en cuenta que uno de los objetivos de diseño es hacer un sistema que dure el máximo tiempo posible con la utilización de una batería ahorrando la mayor cantidad de batería para alargar su ciclo de trabajo. Por eso se ha tenido en cuenta la utilización del *Deep Sleep* en el ESP32 y la elección de pines se ha hecho buscando los pines RTC GPIO ya que se encuentran conectados al subsistema RTC (*Real Time Clock*).

La posibilidad de este proyecto es de conectar hasta 5 sensores (en la PCB diseñada que se explicará en puntos posteriores) como se puede observar en la Figura 59.

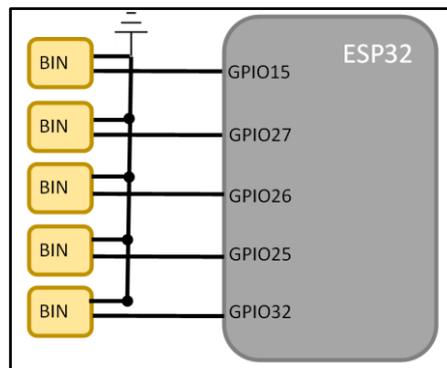


Figura 59. Conexión de sensores al ESP32

A la hora de hacer el diseño de PCB se distinguen tres diseños de los cuales en este proyecto solo se han desarrollado dos de ellos. En puntos posteriores se explicarán en profundidad estos dos diseños, pero en este punto simplemente se mencionará que se tratan de un diseño con módulos ya desarrollados y otro con la inclusión de todos los integrados directamente en la placa.

La conexión de estos sensores en las diferentes placas no es la misma debido a un fallo de diseño en la placa PCB con módulos.

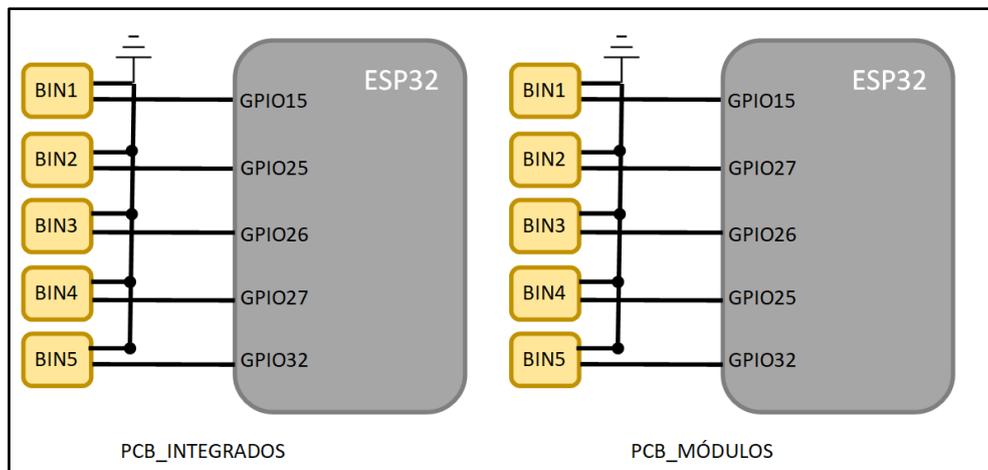


Figura 60. Conexión de sensores en PCB_integrados y PCB_módulos a ESP32

4.4. Configuración y programación de sensores

Con el entorno de trabajo Home Assistant y la herramienta de trabajo ESPHome preparados, instalados y conocidos los esquemas de conexión de los diferentes módulos y sensores, el siguiente paso es realizar la programación de cada uno.

En este punto se explicará la programación básica que se proporciona desde la página oficial de ESPHome a cada módulo, así como la programación final utilizada para la aplicación deseada.

4.4.1. HX711

ESPHome

Quick search

Go

Hide Search Matches

Table of Contents

HX711 Load Cell Amplifier

- Configuration variables:
- Converting Units
- See Also

Join the community

Follow us on Twitter

Source Code

Contact (no support!)

ESPHome is an open source project by Nabu Casa

This site is powered by Netlify

HX711 Load Cell Amplifier

The **hx711** sensor platform allows you to use your **HX711** load cell amplifier ([datasheet](#), [SparkFun](#)) with ESPHome



HX711 Load Cell Amplifier. Image by SparkFun licensed and re-distributed under [CC BY 2.0](#).

Connect GND to GND, VCC to 3.3V and the other two DOUT (or DT for short) and CLK (or SCK) to free GPIO pins.

```
# Example configuration entry
sensor:
  - platform: hx711
    name: "HX711 Value"
    dout_pin: 00
    clk_pin: D1
    gain: 128
    update_interval: 60s
```

Configuration variables:

- name** (Required, string): The name for the load cell sensor.
- dout_pin** (Required, [Pin Schema](#)): The DOUT (or DAT) pin.
- clk_pin** (Required, [Pin Schema](#)): The CLK pin.
- gain** (Optional, enum): The gain. Implicitly selects the channel. Defaults to 128.

Converting Units

As the **HX711** does not have any calibration data, you have to convert the measurement to units yourself. To calibrate the sensor:

- Place a zero load or a known mass on the sensor, for example 0kg
- Wait for the data to arrive in the logs and write down the value. For example 120.
- Place another (different) known mass on the sensor, for example 1kg
- Again wait for the data to arrive and note the value, for example 810.

Once you've done those steps, you can use the `calibrate_linear` filter to map the incoming value to the calibrated one:

```
# Example configuration entry
sensor:
  - platform: hx711
    # ... Other HX711 options
    filters:
      - calibrate_linear:
        - 120 -> 0
        - 810 -> 1
    unit_of_measurement: kg
```

Replace the masses and values 120.0, 810.0, etc with your values.

Figura 61. Programación básica en ESPHome de HX711

En la Figura 61 se puede observar la guía de programación básica que proporciona la web de ESPHome.

Esta programación es la base de lo que es el código final para este módulo como se puede ver en la Figura 62.

```

esp32-dormitorio.yaml
19 password: "f20e6202a5205f6e31350f727ead7"
20
21 wifi:
22 ssid: esp32-dormitorio
23 password: "1234567890"
24
25 manual_ip:
26 static_ip: 192.168.1.92
27 gateway: 192.168.1.1
28 subnet: 255.255.255.0
29
30 # Enable fallback hotspot (captive portal) in case wifi connection fails
31 ap:
32 ssid: "Esp32-Dormitorio"
33 password: "1234567890"
34
35 captive_portal:
36
37 sensor:
38 - platform: hx711
39 name: "Peso Cama"
40 id: pesocama
41 dout_pin: 14
42 clk_pin: 12
43 filters:
44 - calibrate_linear:
45   - 215770 -> 0
46   - 228150 -> 45
47 update_interval: 30s
48 unit_of_measurement: kg
49 accuracy_decimals: 4
50 gain: 128
51
52 - platform: wifi_signal
53 name: "WiFi Signal Sensor"
54 update_interval: 60s
55
56 binary_sensor:
57 - platform: template
58 name: "Persona en cama"
59 device_class: occupancy
60 lambda: |-
61   if (id(pesocama).state > 40) {
62     return true;
63   } else {
64     return false;
65   }

```

Figura 62. Programación HX711 en el dormitorio en ESPHome

El HX711 es el único sensor ubicado en el dormitorio siguiendo la estructura del proyecto, por lo que se muestra directamente la programación completa del ESP32-dormitorio.

```

37 sensor:
38 - platform: hx711
39 name: "Peso Cama"
40 id: pesocama
41 dout_pin: 14
42 clk_pin: 12
43 filters:
44 - calibrate_linear:
45   - 215770 -> 0
46   - 228150 -> 45
47 update_interval: 30s
48 unit_of_measurement: kg
49 accuracy_decimals: 4
50 gain: 128

```

Figura 63. Programación HX711

En este caso las pruebas se han realizado con una silla y con un sujeto cuyo peso es de 50 kg, de ahí a que la calibración se muestre entre 0kg (siendo la silla vacía) a 45 kg (la silla ocupada)

```

56 binary_sensor:
57 - platform: template
58 name: "Persona en cama"
59 device_class: occupancy
60 lambda: |-
61   if (id(pesocama).state > 40) {
62     return true;
63   } else {
64     return false;
65   }

```

Figura 64. Programa de ocupación de HX711

Aunque viendo los valores en kg se puede suponer si existe la presencia de una persona o no (tanto en una silla como en una cama) se ha incluido un pequeño apartado de código como se muestra en la Figura 64 que de lo que se encarga es de mostrar que existe presencia en el momento que la medida supera los 40 kg como se muestra en la Figura 65.



Figura 65. Interfaz Home Assistant para HX711

4.4.2. MAX6675

MAX6675 K-Type Thermocouple Temperature Sensor

The **max6675** temperature sensor allows you to use your **MAX6675** thermocouple temperature sensor ([datasheet](#), [SainSmart](#)) with ESPHome

MAX6675 K-Type Thermocouple Temperature Sensor.

As the communication with the **MAX6675** is done using SPI, you need to have an **SPI** bus in your configuration with the **miso_pin** set (MOSI is not required).

Connect **GND** to **GND**, **VCC** to **3.3V** and the other three **MISO** (or **S0** for short), **CS** and **CLOCK** (or **CLK**) to free GPIO pins.

Living Room Temperature 15.6 °C

```
# Example configuration entry
spi:
  miso_pin: D0
  clk_pin: D1

sensor:
  - platform: max6675
    name: "Living Room Temperature"
    cs_pin: D2
    update_interval: 60s
```

Configuration variables:

- name** (Required, string): The name for the temperature sensor.
- cs_pin** (Required, Pin Schema): The Chip Select pin of the SPI interface.
- update_interval** (Optional, Time): The interval to check the sensor. Defaults to 60s.
- spi_id** (Optional, ID): Manually specify the ID of the **SPI Component** if you want to use multiple SPI buses.
- id** (Optional, ID): Manually specify the ID used for code generation.
- All other options from **Sensor**.

Figura 66. Programación base para MAX6675

De la misma manera se mostrará para el módulo MAX6675.

```
40
41 sensor:
42   - platform: max6675
43     name: "Temperatura vitrocerámica"
44     id: vitrottemp
45     cs_pin: 2
46     update_interval: 30s
47
```

Figura 67. Programación MAX6675 en ESPHome

Dentro de este sensor no se ha realizado una programación muy detallada ya que con el valor de la medida en temperatura es suficiente. Posteriormente se explica cómo se trabaja con este valor. Así, se muestra en pantalla el valor de la temperatura.



Figura 68. Interfaz en Home Assistant del módulo MAX6675

Además, para el sensor de temperatura se ha incluido una entidad que muestre si la vitrocerámica está activada en función del valor de temperatura que tenga. En este caso se ha marcado como temperatura de encendido 50 °C.

```
binary_sensor:  
- platform: template  
  name: "Vitrocerámica en uso"  
  device_class: power  
  lambda: |-  
    if (id(vitrotemp).state > 50) {  
      return true;  
    } else {  
      return false;  
    }  
}
```

Figura 69. Programación uso vitrocerámica



Figura 70. Tarjeta de uso de vitrocerámica

4.4.3. Sensores binarios

Binary Sensor Component

With ESPHome you can use different types of **binary sensors**. They will automatically appear in the Home Assistant front-end and have several configuration options.

Base Binary Sensor Configuration

All **binary sensors** have a platform and an optional device class. By default, the binary will chose the appropriate device class itself, but you can always override it.

```
binary_sensor:
  - platform: ...
    device_class: motion
```

Configuration variables:

- device_class** (Optional, string): The device class for the sensor. See <https://developers.home-assistant.io/docs/core/entity/binary-sensor/#available-device-classes> for a list of available options.
- icon** (Optional, icon): Manually set the icon to use for the **binary** sensor in the frontend.
- filters** (Optional, list): A list of filters to apply on the **binary** sensor values such as inverting signals. See [Binary Sensor Filters](#).

Automations:

- on_press** (Optional, Automation): An automation to perform when the button is pressed. See [on_press](#).
- on_release** (Optional, Automation): An automation to perform when the button is released. See [on_release](#).
- on_state** (Optional, Automation): An automation to perform when a state change is published. See [on_state](#).
- on_click** (Optional, Automation): An automation to perform when the button is held down for a specified period of time. See [on_click](#).
- on_double_click** (Optional, Automation): An automation to perform when the button is pressed twice for specified periods of time. See [on_double_click](#).
- on_multi_click** (Optional, Automation): An automation to perform when the button is pressed in a specific sequence. See [on_multi_click](#).

Figura 71. Programación básica sensores binarios

Dentro del grupo de sensores binarios se encuentra el sensor de caudal, situado en el cuarto de baño y en la cocina (salón-cocina en este caso) y el sensor de presión de asiento situado en el salón.

La programación básica de cada uno de estos sensores es la misma como se puede ver en la Figura 72, donde se le asigna una identidad.

Es necesario para los sensores binarios activar el modo *INPUT_PULLUP* por la resistencia del pin en el ESP32 y para que la detención *ON/OFF* sea correcta hay que invertir en ambos casos la salida.

```
49 - binary_sensor:
50   - platform: gpio
51     id: fregadero
52     device_class: door
53     pin:
54       number: 32
55       mode: INPUT_PULLUP
56     filters:
57       - invert:
58         name: "Fregadero"
59
60   - platform: gpio
61     id: sillón
62     device_class: occupancy
63     pin:
64       number: 27
65       mode: INPUT_PULLUP
66     filters:
67       invert:
68         name: "Presencia en sillón"
```

Figura 72. Programación sensores binarios en salón-cocina en ESPHome

De esta manera se obtiene si el sensor de caudal está activado o no y, además, se ha incluido una plantilla para que muestre en la interfaz si existe presencia donde se encuentre el sensor de presión de asiento.

Esto se ha realizado debido a que los datos que se obtienen de estos sensores son datos de estado, es decir, si el sensor se encuentra en ON o en OFF por lo que interesa que exista, de una forma más visual, una manera de ver que el estado ON representa una persona sentada en el sofá.

Siendo este proyecto parte del proyecto HIMTAE [14] el estudio de la frecuencia de uso y el tiempo de uso de cada uno de estos sensores son unos parámetros interesantes para trabajar.

Por ello se ha desarrollado la forma de monitorizar la frecuencia de uso y el tiempo de uso de cada sensor realizando una programación desde los archivos `.yaml` de Home Assistant.

Para ello es necesario instalar desde la tienda de complementos el *Add-on Field Editor*. Una vez instalado, al abrir la interfaz, aparecerán las carpetas como en la Figura 73.

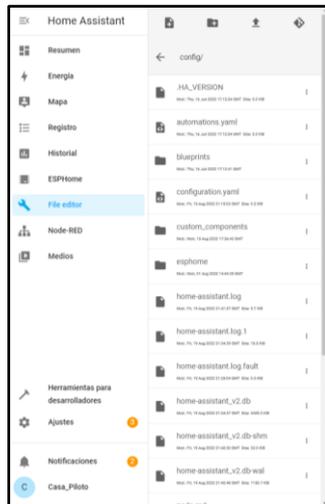


Figura 73. Menú File editor

Para comenzar a programar debemos acceder a la carpeta `sensores.yaml`. En esta carpeta se incluye la programación de un contador que lleva un registro por día, semana y mes de las entidades que se configuran (Figura 74) y un cronómetro para la cuenta del tiempo de uso por día, semana y mes (Figura 75).

```

29 *# CONTADOR PARA SILLON (DIA/SEMANA/MES):
30 |
31 *--- platform: history_stats
32 *--- name: Frecuencia de uso sillón (Día)
33 *--- entity_id: binary_sensor.presencia_en_sillon
34 *--- state: "on"
35 *--- type: count
36 *--- start: '{{ now().replace(hour=0).replace(minute=0).replace(second=0) }}'
37 *--- end: '{{ now() }}'
38 *---
39 *--- platform: history_stats
40 *--- name: Frecuencia de uso sillón (Semana)
41 *--- entity_id: binary_sensor.presencia_en_sillon
42 *--- state: "on"
43 *--- type: count
44 *--- start: '{{ as_timestamp( now().replace(hour=0).replace(minute=0).replace(second=0) ) - now().weekday() * 86400 }}'
45 *--- end: '{{ now() }}'
46 *---
47 *--- platform: history_stats
48 *--- name: Frecuencia de uso sillón (Mes)
49 *--- entity_id: binary_sensor.presencia_en_sillon
50 *--- state: "on"
51 *--- type: count
52 *--- end: '{{ now().replace(hour=0, minute=0, second=0) }}'
53 *--- duration:
54 *--- days: 30

```

Figura 74. Contador para frecuencia de uso del sillón

```

87 # CRONOMETRO PARA SILLON (DIA/SEMANA/MES)
88
89 --- platform: history_stats
90 --- name: Tiempo de uso sillon (Dia)
91 --- entity_id: binary_sensor.presencia_en_sillon
92 --- state: 'on'
93 --- type: time
94 --- start: '{{ now().replace(hour=0, minute=0, second=0) }}'
95 --- end: '{{ now() }}'
96
97 --- platform: history_stats
98 --- name: Tiempo de uso sillon (Semana)
99 --- entity_id: binary_sensor.presencia_en_sillon
100 --- state: 'on'
101 --- type: time
102 --- start: '{{ as_timestamp( now().replace(hour=0, minute=0, second=0) ) - now().weekday() * 86400 }}'
103 --- end: '{{ now() }}'
104
105 --- platform: history_stats
106 --- name: Tiempo de uso sillon (Mes)
107 --- entity_id: binary_sensor.presencia_en_sillon
108 --- state: 'on'
109 --- type: time
110 --- end: '{{ now().replace(hour=0, minute=0, second=0) }}'
111 --- duration:
112 --- days: 30

```

Figura 75. Cronómetro para tiempo de uso del sillón

Una vez programados todos los contadores y cronómetros es importante llamar desde la carpeta `/config/configuration.yaml` a esta carpeta `sensor.yaml` (subrayado de azul en la Figura 76) y luego reiniciar el sistema para que se guarden en Home Assistant como entidades.

```

/config/configuration.yaml
1
2 # Loads default set of integrations. Do not remove.
3 default_config:
4
5 # Text to speech
6 tts:
7   - platform: google_translate
8
9 automation: !include automations.yaml
10 script: !include scripts.yaml
11 sensor: !include sensors.yaml
12

```

Figura 76. Carpeta `config/configuration.yaml`

En la Figura 77 se ve un ejemplo de la configuración para un sensor de presencia de asiento (Presencia en sillón) y para un sensor de caudal (Fregadero) dentro de un mismo espacio como es salón-cocina.

<p>Presencia en sillon</p> <p>Detectado</p>	<p>Fregadero</p> <p>Cerrada</p>
<p>Frecuencia de uso sillón</p> <p>☑ Día 7</p> <p>☑ Semana 24</p> <p>☑ Mes 154</p>	<p>Frecuencia de uso fregadero</p> <p>☑ Día 0</p> <p>☑ Semana 22</p> <p>☑ Mes 98</p>
<p>Tiempo de uso sillón</p> <p>☑ Día 0</p> <p>☑ Semana 1:19:48</p> <p>☑ Mes 3:40:48</p>	<p>Tiempo de uso fregadero</p> <p>☑ Día 0</p> <p>☑ Semana 36</p> <p>☑ Mes 3:00</p>

Figura 77. Interfaz en Home Assistant de los sensores binarios

El formato temporal del tiempo de uso tanto del sillón como del fregadero está representado de tal manera que se va incluyendo el tiempo en segundos, minutos y horas según vaya transcurriendo el tiempo.

Esto quiere decir que al encontrar un valor único ese valor se interpresas como segundos, al haber completado lo 60 segundos se crea un nuevo formato 00:00 que representa

minutos y segundos. De la misma manera ocurre con las horas, al haber completado un ciclo en minutos el formato de visualización del tiempo cambia a 00:00:00 correspondiendo los dos primeros dígitos a la hora, los dos segundos a minutos y los últimos a los segundos.

Este formato está dado cuando en la interfaz de Home Assistant configuramos estos tres valores como un grupo de entidades. Si queremos representar en la interfaz como una única entidad el formato de visualización del tiempo cambia a horas como se observa en la Figura 78.

Tiempo de uso fregadero (Dia) 0,0 h		Tiempo de uso sillón (Dia) 0,72 h	
Tiempo de uso fregadero (Semana) 0,0 h		Tiempo de uso sillón (Semana) 0,72 h	
Tiempo de uso fregadero (Mes) 0,06 h		Tiempo de uso sillón (Mes) 0,01 h	

Figura 78. Interfaz Salón Cocina con formato en horas del tempo de uso

En el caso de la obtención de los valores de tiempo existen otras opciones de configuración como por ejemplo puede ser la creación de un *timer* y, a través de una automatización, activar el *timer* siempre y cuando el estado de la entidad principal del sensor sea ON y desactivarlo cuando se encuentre en OFF.

Esta opción fue valorada y probada pero se ha optado por el método explicado dado que es más sencillo y rápido de programar y poner en marcha además de que proporciona los valores directamente como entidades sin tener la necesidad de crear un ayudante y automatizaciones, de ahí su sencillez.

4.5. Interfaz Home Assistant

El siguiente paso es crear y estructurar la interfaz Home Assistant. En puntos anteriores se han mostrado de forma individual las diferentes entidades, en este punto se mostrará y explicará en conjunto.

Se ha creado un apartado de Casa TFG donde están incluidas todas las áreas de la vivienda, en este caso se han separado por Dormitorio, Salón-Cocina y Aseo.

En primer lugar, se expone la interfaz correspondiente al área salón-cocina de la vivienda (Figura 79).

La primera columna representa el sensor de presencia de sillón donde la primera entidad muestra si detecta o no presencia en el sillón, es decir si el sensor está ON. Por otra parte, en la parte inferior de esta columna están representados las frecuencias de uso en día, semana y mes y el tiempo de uso en día, semana y mes del sillón.

La segunda columna recoge la información sobre la temperatura de la vitrocerámica y además se ha incluido la señal Wi-Fi en dBm (decibel-milivatio). El dBm determina la cantidad de señal que llega al ESP32. Esta señal puede oscilar entre -100 dBm y -20 dBm, representando que no existe señal y que la señal es muy buena, respectivamente. Por ello, se ha creado una entidad por colores que te indica en verde si oscila entre -20 y -40 ya que se trata de una señal buena, amarillo si oscila entre -40 y -60 ya que se trata de una señal con menos calidad y rojo entre -60 y -100 ya que indica que no hay señal (Figura 80).

La tercera columna representa la misma información que se ha explicado para la presencia del sillón, pero para el uso del fregadero por lo que muestra si el grifo del fregadero está abierto o cerrado.

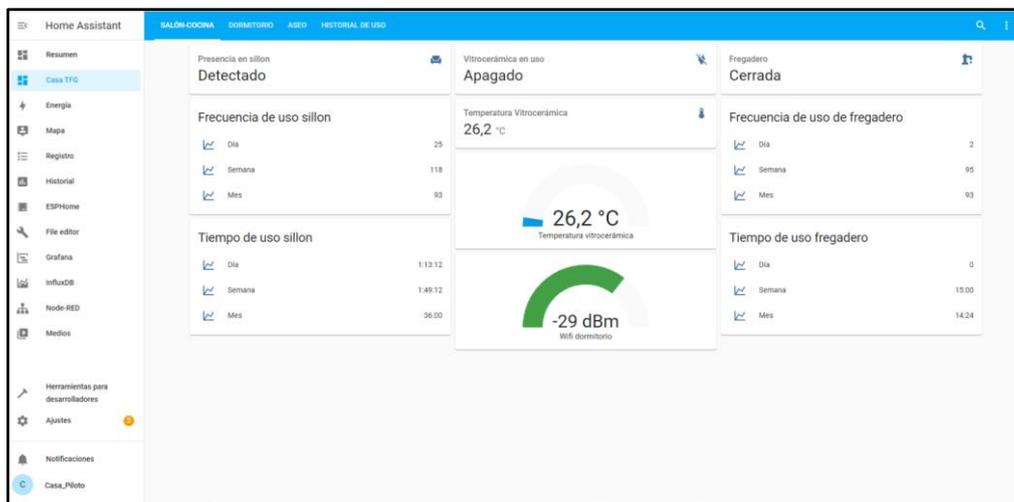


Figura 79. Interfaz Home Assistant área salón-cocina.



Figura 80. Configuración entidad Wi-Fi

En segundo lugar (Figura 81), se expone la interfaz del dormitorio de la vivienda. En esta área solo se muestra si se detecta persona o no como se ha explicado anteriormente y cuanto peso está detectando el sensor, así como la señal Wi-Fi del ESP32 correspondiente.

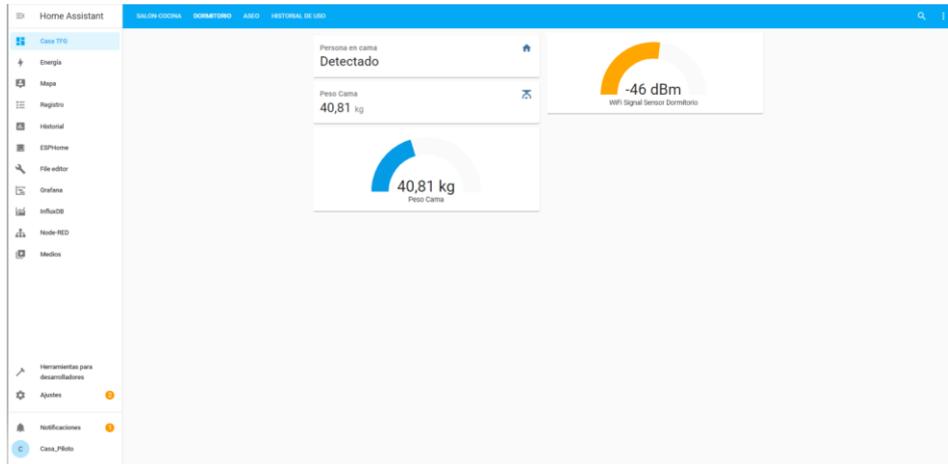


Figura 81. Interfaz Home Assistant área dormitorio

En tercer lugar (Figura 82), se encuentra el apartado de aseo que incluye los sensores de caudal de la ducha, el lavabo de manos y el bidé. La visualización de los datos de estos sensores es exactamente la misma que la anteriormente vista como fregadero. De la misma manera aparece la señal Wi-Fi del ESP32 ubicado en el aseo.

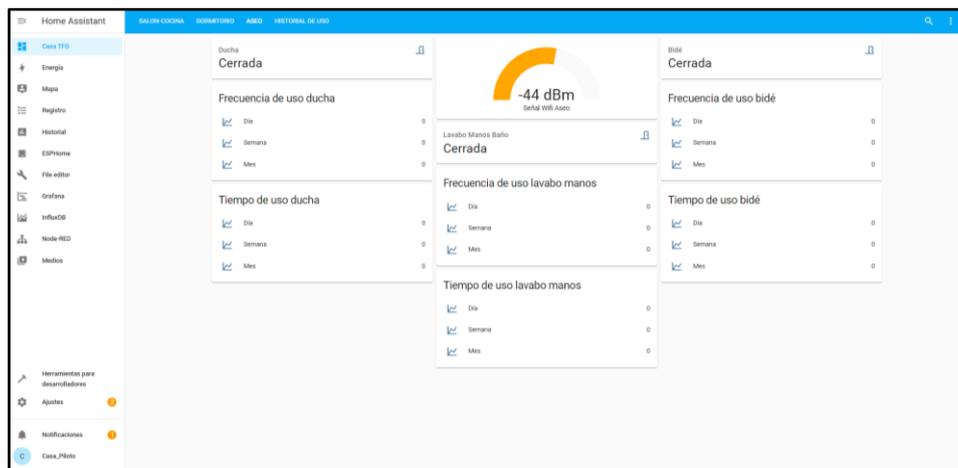


Figura 82. Interfaz Home Assistant área aseo

Por último, se ha creado un área dentro de la interfaz de Home Assistant que abarca el registro de uso de todos los sensores en un tiempo de 24h (Figura 83).

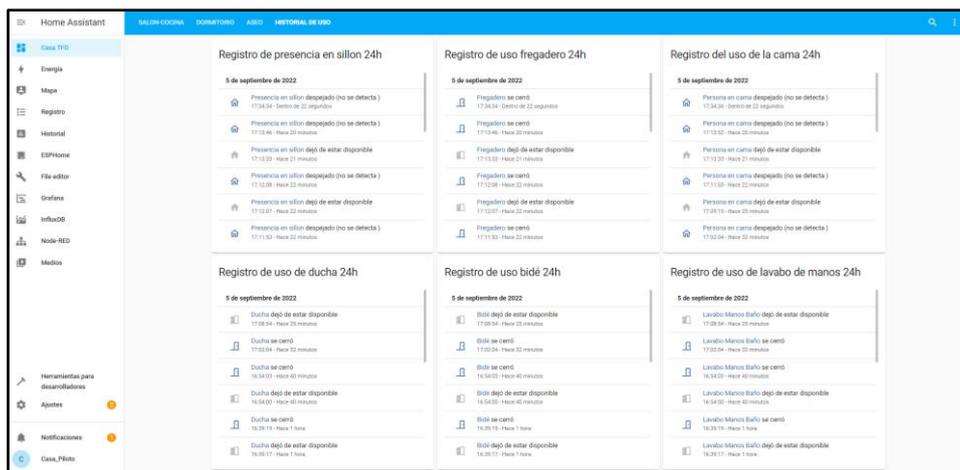


Figura 83. Interfaz Home Assistant Historial de uso

En el caso del registro de uso del uso de la cama y de la vitrocerámica se ha utilizado una plantilla que determina que el sensor está en uso a partir de un valor. En el caso de la cama cuando la medida supera los 40 kg (Figura 84) y en el caso de la temperatura de la vitrocerámica cuando el valor de temperatura supera los 50°C (Figura 85).

```

56 - binary_sensor:
57 -   platform: template
58     name: "Persona en cama"
59     device_class: occupancy
60     lambda: |-
61       if (id(pesocama).state > 40) {
62         return true;
63       } else {
64         return false;
65       }

```

Figura 84. Programación de presencia de la cama

```

74 -   platform: template
75     name: "Vitrocerámica en uso"
76     device_class: power
77     lambda: |-
78       if (id(vitrotemp).state > 50) {
79         return true;
80       } else {
81         return false;
82       }

```

Figura 85. Programación de encendido vitrocerámica

4.6. Interfaz Grafana e InfluxDB

Por otra parte, este proyecto al estar dentro de un proyecto centrado en la monitorización de los hábitos es importante guardar los datos para su posterior análisis, de esta manera se ha incluido la herramienta InfluxDB y Grafana para visualizar esos datos.

InfluxDB es un sistema de gestión de base de datos de series temporales sin esquema y de código abierto desarrollado por InfluxDB. Ese sistema está integrado como herramienta en Home Assistant y por la cual se pueden utilizar sus datos.

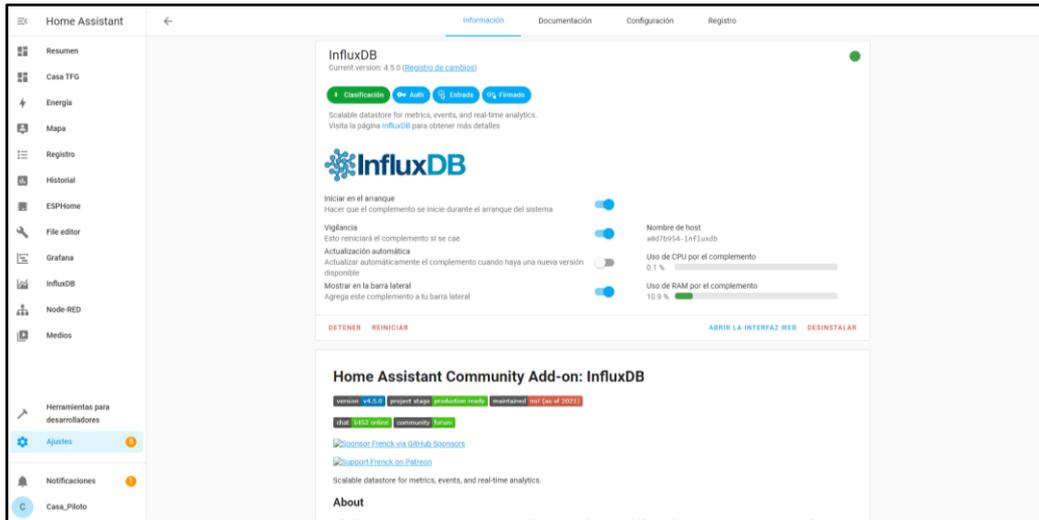


Figura 86. Herramienta InfluxDB en Home Assistant

Para visualizar esos datos se utiliza la herramienta Grafana.

Se trata de una plataforma de código abierto para la supervisión, el análisis y la visualización en tiempo real de los datos del sistema. Tiene como objetivo la presentación de una cantidad grande de datos que proceden de diferentes fuentes de forma fácil e intuitiva.

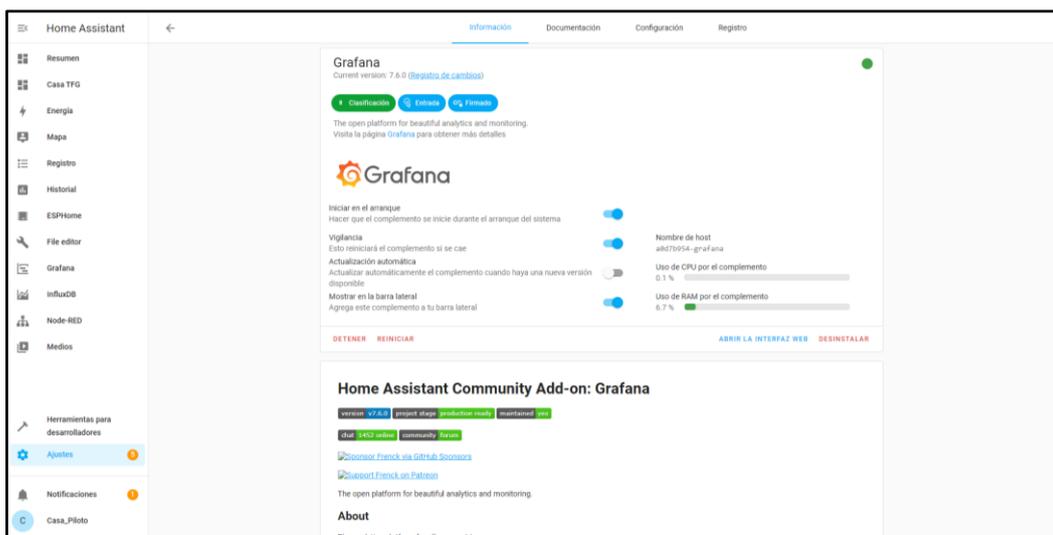


Figura 87. Herramienta Grafana en Home Assistant

La plataforma Grafana permite utilizar InfluxDB como gestor de base de datos de los datos obtenidos de Home Assistant.

El primer paso para lograr utilizar estas dos plataformas conjuntas es realizar la configuración en InfluxDB.

Hay que crear en el apartado de explore un usuario llamado “homeassistant” con una contraseña y seleccionar todos los permisos.

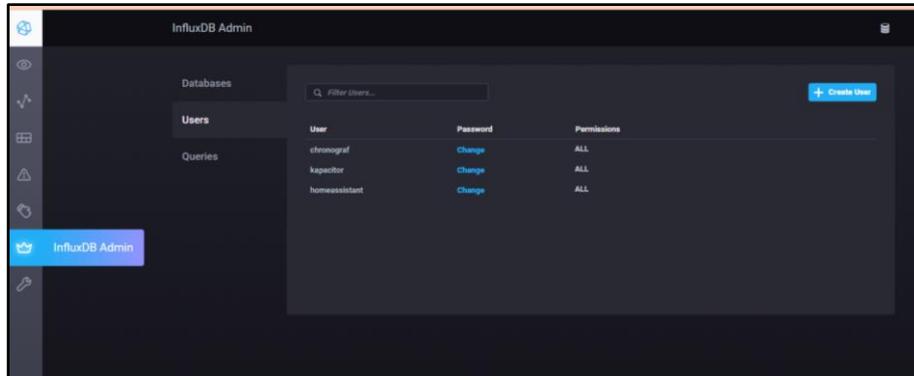


Figura 88. Creación usuario InfluxDB

Para que Home Assistant proporcione sus datos debe de tener registrado el usuario de InfluxDB, esta acción se realiza desde el archivo `config/configuration.yaml`, como se ha trabajado anteriormente para los sensores, incluyendo el apartado de código subrayado de azul de la Figura 89.

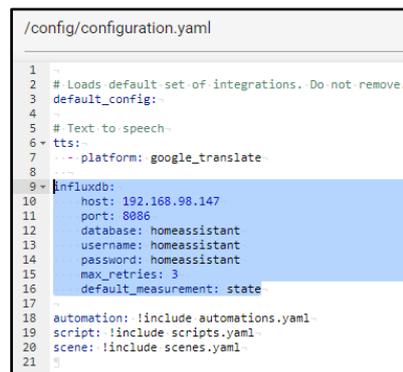


Figura 89. Configuración InfluxDB en Home Assistant

El último paso es seleccionar como gestor de bases de datos en Grafana a InfluxDB en el apartado de `Data Sourcers` incluyendo el host, que en este caso es **192.168.1.224:8086**, y los datos de `database`, `username` y `password` incluidos en el archivo `.yaml`.

A partir de aquí ya se puede crear los diferentes `dashboards` con la visualización de datos.

En este caso no se han separado por áreas, sino que todos los datos se han incluido en la misma pantalla.

Según la forma en la que se quiera ver los valores se configurará un tipo de gráfica u otra. En la Figura 90 se observa la gráfica que determina la velocidad de la señal Wi-Fi separada por colores en función de su valor.

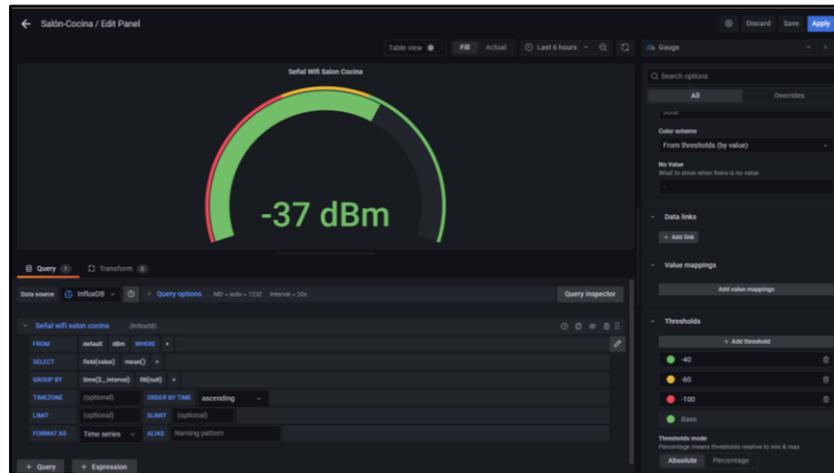


Figura 90. Ejemplo configuración gráfica señal Wi-Fi

Este apartado aún se encuentra en desarrollo por lo que se ha trabajado con valores puntuales como la temperatura, el peso o la señal wifi para estudiar los diferentes tipos de gráficas y formas de visualizar los datos que proporciona la plataforma.



Figura 91. Ejemplo de DASHBOARD creada en Grafana

4.7. Desarrollo de PCB

Este punto trata de la creación de una placa de circuito impreso que permita la conexión de todos los sensores que se plantean en la vivienda, siendo así, una placa multifuncional.

El diseño de esta placa permite la versatilidad de aplicar los sensores que se desean en cada punto de la vivienda.

En todos los diseños se han usado ESP32-WROOM-32 como microcontrolador que va alimentado por una batería recargable de polímero de litio de 3.7V (Figura 92) donde se incluye la opción de cargar la batería al mismo tiempo que está en funcionamiento.

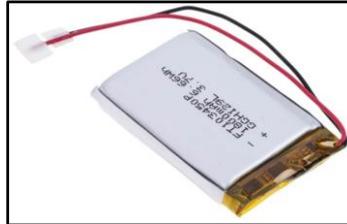


Figura 92. Batería recargable de Polímero de Litio de 3.7V

Por otra parte, se ha incluido un módulo ADS1115 (Figura 93), un convertidor analógico-digital, que añade tres entradas analógicas a la placa para futuras aplicaciones y cuya entrada A0 está destinada a la medida de voltaje de la batería.

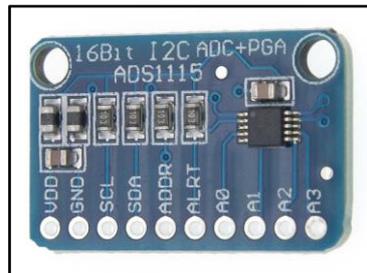


Figura 93. Módulo ADS1115

Teniendo en cuenta lo anterior se han realizado tres diseños diferentes: PCB módulos, PCB integrados y PCB ADS1115 integrado.

Antes de profundizar en los diferentes diseños planteados se va a explicar aquellas partes que son comunes para todas.

En primer lugar, se explica el ESP32-WROOM-32 (Figura 94).

Siguiendo el esquemático proporcionado en el *datasheet* que se muestra en la Figura 17 para los componentes correspondientes a los pines 3V3 y EN se han añadido dos conectores más.

La función de estos conectores es para la programación desde el puerto serie (J2) usando una FTDI (Adaptador de datos de puerto serie a USB) y para conectar el botón de BOOT para poder cargar el programa en el ESP32 (J1).

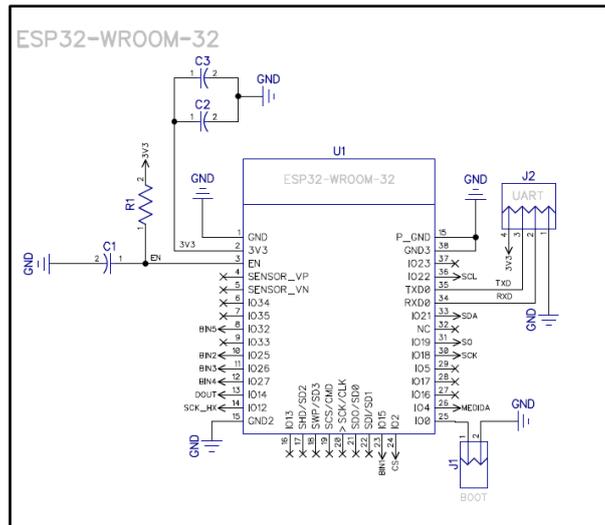


Figura 94. Esquemático general ESP32-WROOM-32

En la Figura 95 se muestra el circuito de la batería. En este circuito podemos ver el conector J11 que corresponde con la batería mencionada anteriormente. Seguido viene el conector J13 que se trata de un conector que sirve para alimentar cualquier cosa que se vaya a necesitar directamente desde la batería.

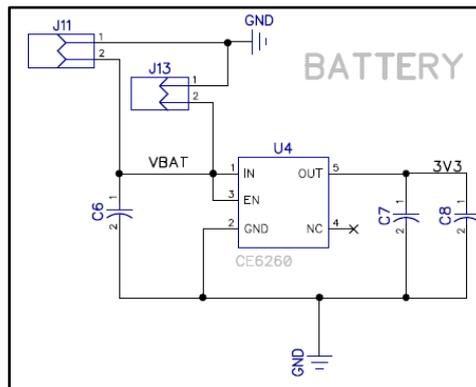


Figura 95. Esquemático general conexión batería

A continuación, se encuentra el integrado CE6260. Se trata de un regulador de tensión con salida a 3V3 para la alimentación del ESP32. La elección de este componente se hizo en base a la tensión de caída que soportaba ya que, como se muestra en la Figura 96 resaltado en amarillo, posee una tensión de caída muy baja lo que nos ahorra un consumo adicional de batería.

■ ELECTRICAL CHARACTERISTICS							
CE6260 Series (V _{IN} =V _{OUT} +1V, C _{IN} = C _{OUT} = 3.3μF, Ta=25°C, unless otherwise specified)							
PARAMETER	SYMBOL	CONDITIONS		MIN	TYP	MAX	UNITS
Output Voltage	V _{OUT(E)} (Note 2)	I _{OUT} =100mA, V _{IN} =V _{OUT} +1V, V _{IN} ≥2V	1.5V<V _{OUT} ≤5.0V	V _{OUT} *0.98	V _{OUT} (Note 1)	V _{OUT} *1.02	V
			0.9V≤V _{OUT} ≤1.5V	V _{OUT} -0.03	V _{OUT}	V _{OUT} +0.03	V
Supply Current	I _{SS}	V _{CE} =V _{IN} =V _{OUT} +1V			6		μA
Shutdown Current	I _{SHDN}	-			0.1	1.0	μA
Output Current	I _{OUT}	V _{IN} ≥2V, V _{IN} =V _{OUT} +1V		500			mA
Dropout Voltage (Note 3)	V _{dir1}	I _{OUT} = 100mA			120		mV
	V _{dir2}	I _{OUT} = 500mA			650		mV

Figura 96. Valor Dropout Voltage en hoja de datos de CE6260

La búsqueda de este componente entre los diferentes fabricantes ha sido muy difícil debido a la escasez de mercado por lo que se ha buscado una alternativa equivalente utilizando los parámetros más aproximados posibles dando con el integrado AP2204K-3V3.

Para el cargador se realizará mediante un conector MICROUSB y con el integrado MCP73831T-2ACI/OT.

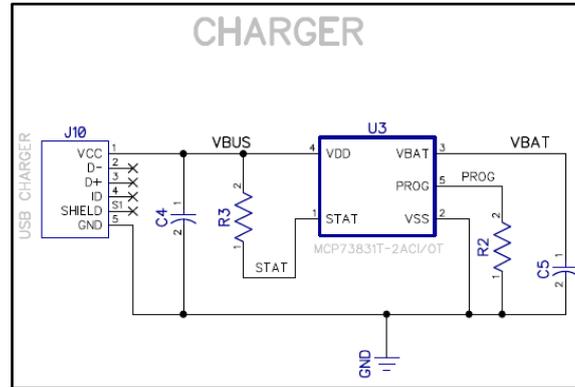


Figura 97. Esquemático general cargador de batería

4.7.1. PCB con módulos

Esta versión es la inicialmente pensada, se trata de una placa en la cual solo se acoplaría mediante pines los módulos que fueran necesarios donde, en este caso, se tiene la opción del MAX6675 (Figura 19) y del HX711 (Figura 24).

El convertidor analógico-digital ADS1115 también se encontraría como módulo, pero es un elemento que siempre será necesario ya que se utiliza para la medida del nivel de batería por lo que se encontraría soldado a la placa.

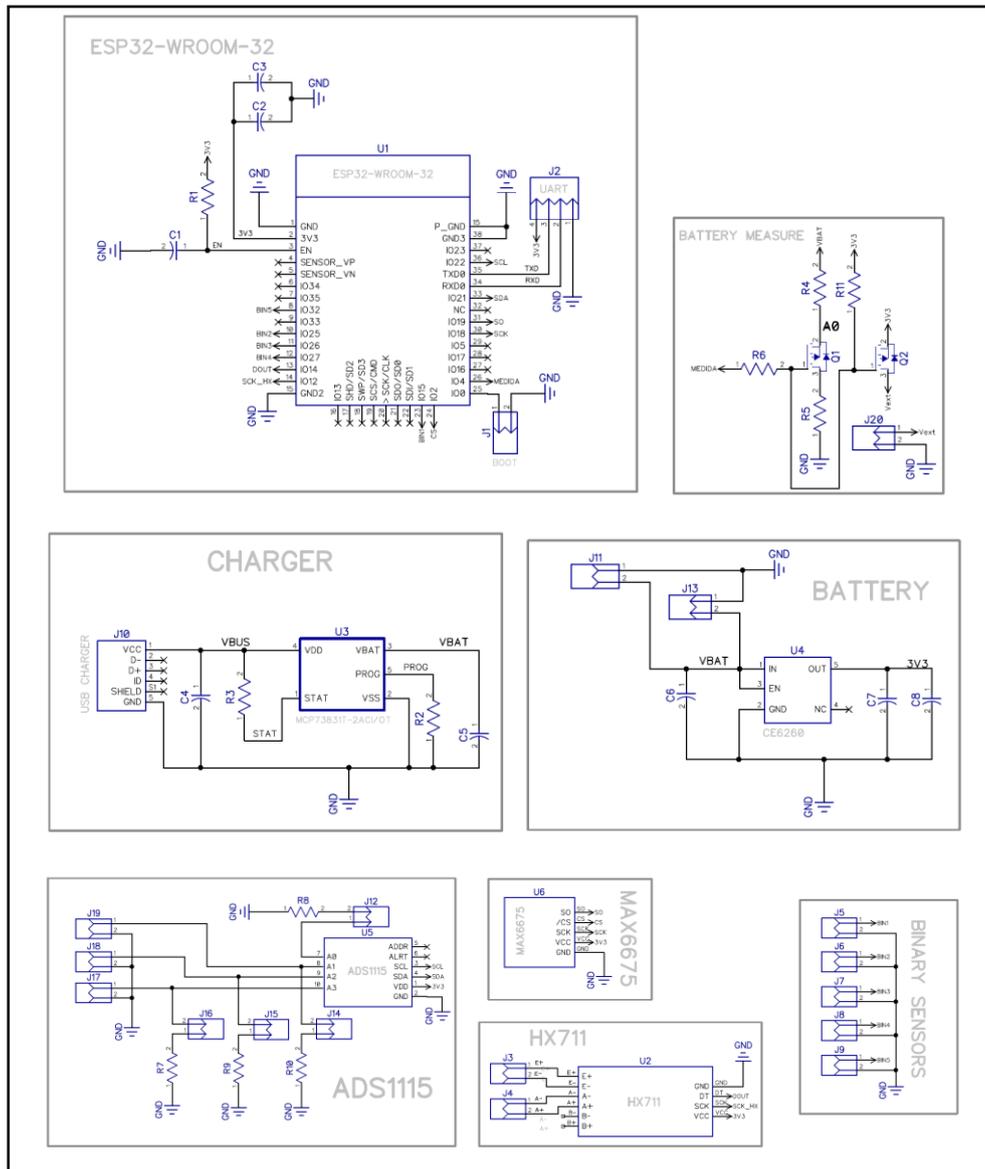


Figura 98. Esquemático PCB con módulos

La idea de esta versión surge de las pruebas físicas inicialmente realizadas en una *protoboard* ya que está pensada de tal manera que si en la vivienda sólo se necesita controlar el peso de la cama sólo sería necesario adquirir un HX711 para acoplarlo.

En comparación con placas que se explicarán posteriormente esta placa es la más grande con un tamaño de 78 x 55 mm.

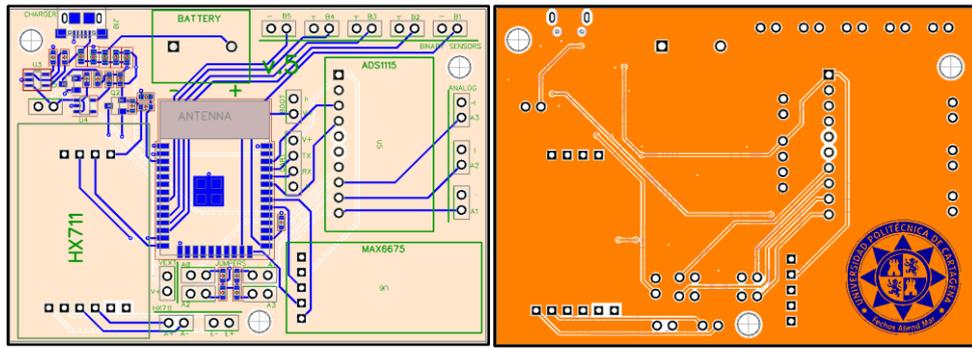


Figura 99. Diseño PCB con módulos TOP

Finalmente, el diseño de esta placa se ha llevado a la fabricación haciendo posible así, para futuros trabajos, el montaje y pruebas de esta.

4.7.2. PCB con integrados

En esta versión se plantea que tanto el HX711 como el MAX6675 y el ADS1115 se encontrarían integrados y soldados en la placa.

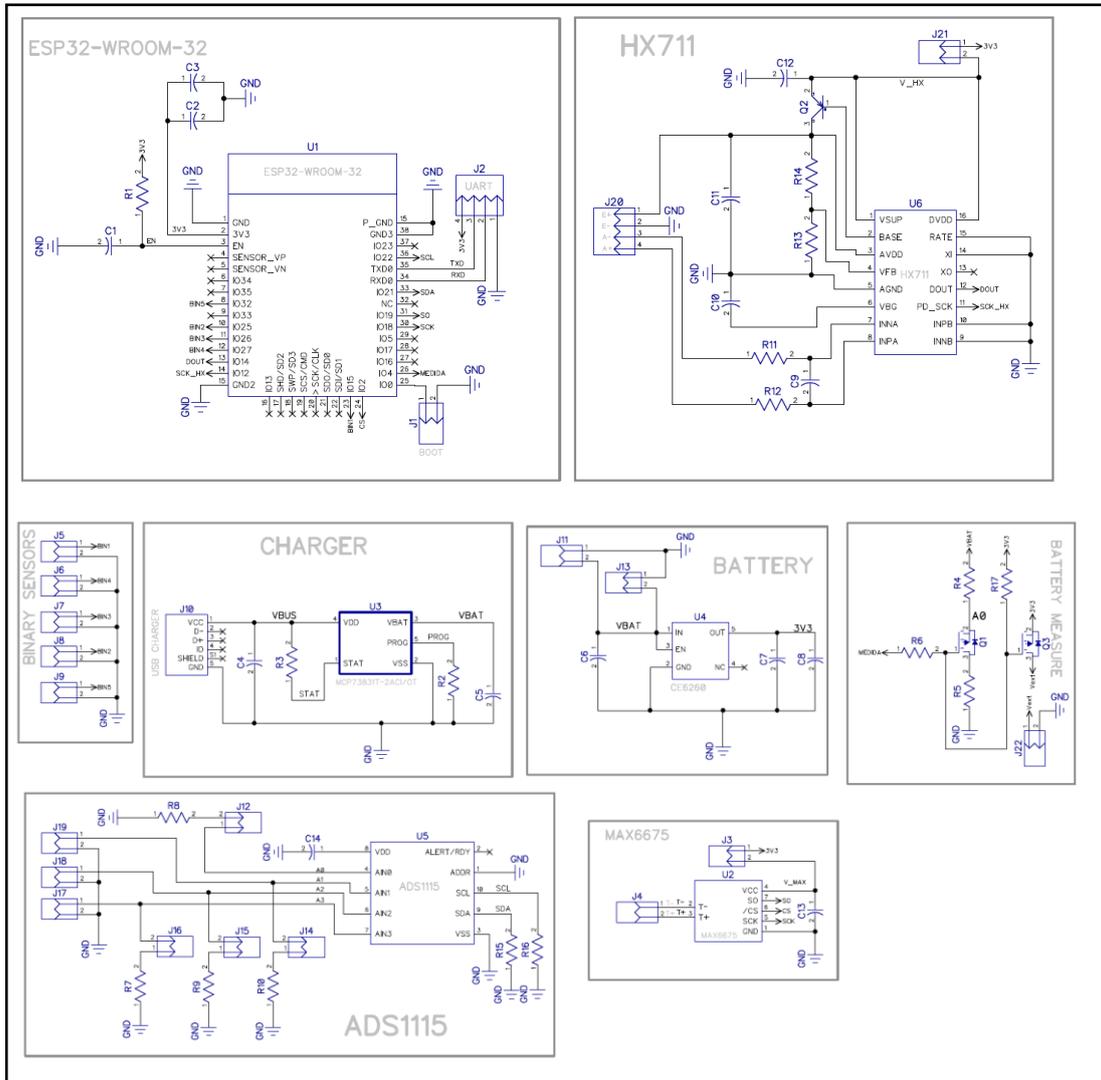


Figura 100. Esquemático PCB integrados

Ya que no en todas las áreas de la vivienda se van a utilizar los mismos sensores el HX711 y el MAX6675, que se encuentren integrados en la placa, disponen de un jumper en la alimentación para que solo dispongan de alimentación cuando la aplicación lo requiera.

El desarrollo de los circuitos del HX711 (Figura 101), MAX6675 (Figura 102) y ADS1115 (Figura 103) se ha hecho recreando el módulo físico utilizado para las pruebas iniciales y cuyos esquemáticos se encuentran en las hojas de datos de los respectivos componentes.

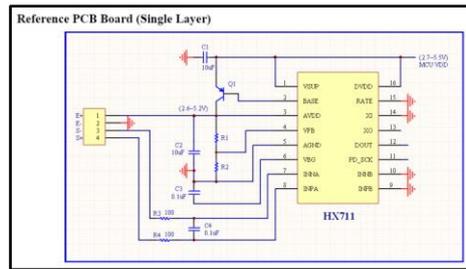


Figura 101. Circuito de aplicación de la hoja de datos del HX711

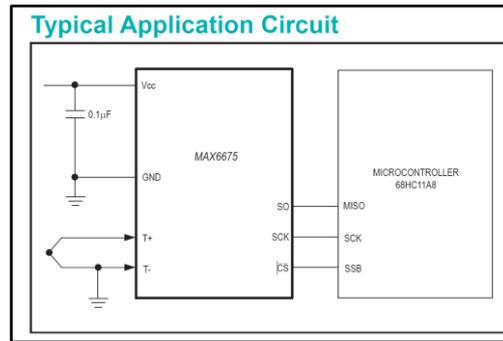


Figura 102. Circuito de aplicación de la hoja de datos de MAX6675

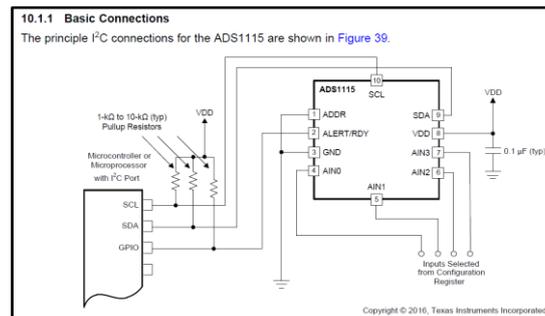


Figura 103. Circuito de aplicación de la hoja de datos de ADS1115

El diseño de esta placa se hizo con la idea de crear una placa más profesional, pequeña y versátil que no implicara la compra aparte de todos los módulos a utilizar.

Finalmente, como se muestra en la Figura 104 se intentado realizar un diseño lo más visual posible para que cualquier persona ajena al proyecto pueda interpretarla y utilizarla. El tamaño de la placa se ha reducido a 45 x 48 mm.

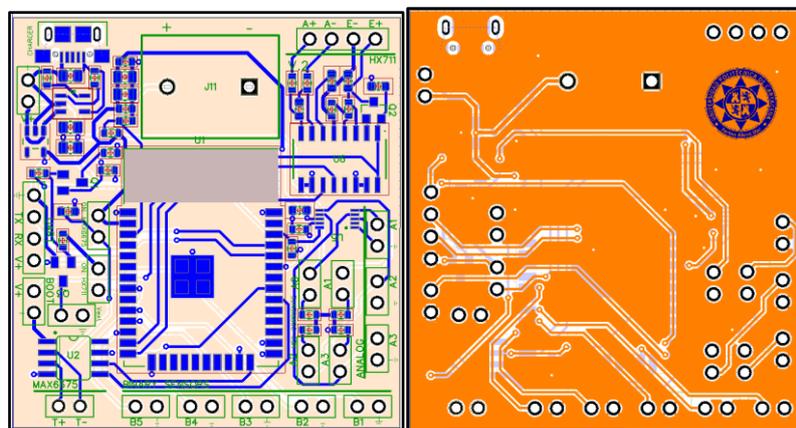


Figura 104. Diseño PCB con integrados

4.7.3. PCB con ADS1115 integrado

Esta versión se presenta como un híbrido entre las dos ideas anteriores de tal manera que los módulos HX711 y MAX6675 están presentes pero el convertidor analógico-digital se encuentra integrado en la placa y no como módulo como se ha visto en el apartado 4.6.1.

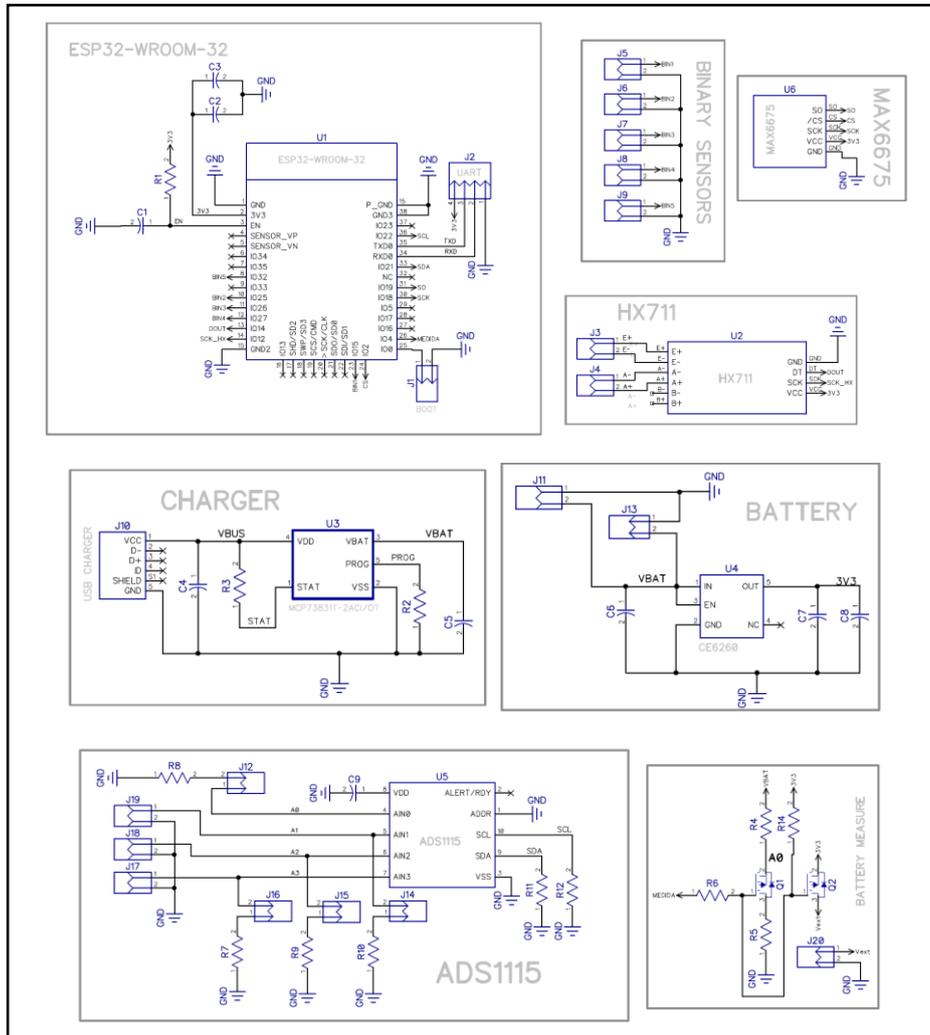


Figura 105. Esquemático PCB con ADS1115 integrado

Quedando así un diseño de placa mostrado en la Figura 106 con un tamaño de 62 x 59 mm.

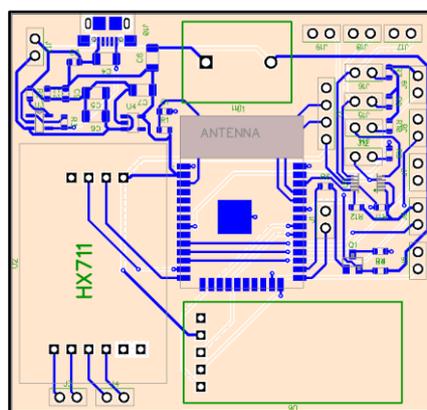


Figura 106. PCB ADS1115 integrado TOP

4.8. Definición de experimentos

Antes de realizar el desarrollo y diseño de la PCB es importante estar seguro de que todos los sensores y módulos funcionan correctamente realizando diferentes pruebas.

Para ello, se han realizado una serie de experimentos de los distintos sensores desde la protoboard para asegurar su correcta conexión y funcionamiento.

Los experimentos han sido los siguientes.

- Colocación del sensor de presión de asiento debajo de las fundas de un sofá para detectar la presencia.
- Colocación de un termopar Tipo K en el grifo del fregadero para comprobar que puede detectar cuando existe paso de agua.
- Colocación de las células de carga en una plantilla realizada en impresión 3D simulando una báscula
- Colocación de las células de carga en las patas de una silla para detectar con más precisión si una persona se sienta.

5. Resultados

Este apartado incluye los resultados ante los experimentos expuestos en puntos anteriores, así como en el resultado final de la realización de este proyecto.

En primer lugar, se exponen los resultados antes los experimentos.

El experimento final con el sensor de presión de asiento trata de colocarlo debajo de las fundas del sofá para asegurar la correcta detección (Figura 107)



Figura 107. Sensor de presión de asiento situado en sofá

Ante estas pruebas realizando diferentes posturas en el sofá como puede ocurrir de que una persona está sentada, acostada o reclinada se corrobora que realiza una correcta detección.



Figura 108. Detección de presencia sofá

Este experimento tiene detrás otros más básicos como es la utilización de este sensor durante las horas de estudio para comprobar, además de la detección, que la programación de la frecuencia y el tiempo de uso es correcta obteniendo así los siguientes datos:

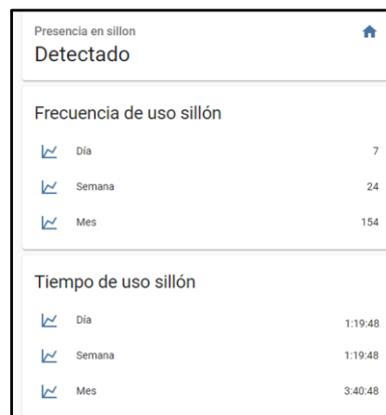


Figura 109. Tiempo de uso del sillón en una sola tarjeta

Como se ha comentado, el formato de hora no es la misma cuando la entidad de tiempo de uso está sola o con un conjunto de entidades.

Se hizo el experimento también con la entidad sola mostrando los siguientes resultados:



Figura 110. Tiempo de uso del sillón en tarjetas diferentes

Por otra parte, se ha probado el sensor de temperatura poniendo una olla de agua a hervir llegando de forma progresiva a unos valores de temperatura próximos a la temperatura de ebullición del agua por lo que se obtiene un resultado positivo.



Figura 111. Cambio de la temperatura en la prueba de agua en ebullición

En el caso de las pruebas con las células de carga se han encontrado más problemas debido a que los valores tomando medidas en diferentes intervalos (desde 10s a 60s) poseen un error muy alto.

Para la obtención del valor final de medida del peso se ha establecido una media aritmética de un número total de 10 medidas por cambio de peso, pero aún así en diferentes pruebas mostraba un valor de 7kg cuando no había peso en las galgas.

Realizando pruebas con diferentes conjuntos de galgas se consiguió obtener valores de pesaje cercanos a los reales, pero también con bastante acumulación de error.



Figura 112. Medida de peso con células de carga

Con respecto al diseño de la placa de circuito impreso se ha logrado mandar a fabricar pero debido a falta de tiempo no ha sido posible realizar el montaje y las pruebas de la misma.

Todas las conexiones de los módulos y sensores diseñados en la placa están basadas en el funcionamiento de la protoboard por lo que no se ha podido testear los circuitos correspondientes de la conexión de la batería y del cargador con salida MICROUSB, así como la medida de la batería a través del módulo ADS1115.

En las siguientes figuras se observa el resultado final de estas placas a falta del montaje.

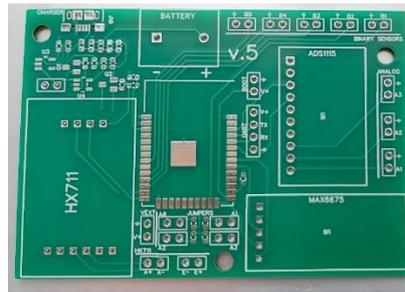


Figura 113. PCB con módulos



Figura 114. PCB con integrados

Con respecto al resultado final del proyecto se ha conseguido instalar todos los sensores, así como realizar las respectivas pruebas para corroborar que funcionan y así crear la interfaz de Home Assistant.

Entre las posibilidades de monitorización se han configurado:

- Visualización del peso de las células de carga (Figura 112).
- Visualización de la temperatura de la vitrocerámica (Figura 111)
- Muestra del estado de los sensores de presión de asiento y caudal, así como la visualización de la frecuencia y el tiempo de uso de estos dispositivos (Figura 109 y Figura 110).
- Creación de la interfaz Home Assistant (Figura 79, Figura 81, Figura 82 y Figura 83).
- Instalación y configuración de las herramientas InfluxDB y Grafana para la visualización de los parámetros.
- Incluir en la interfaz Grafana los gráficos de temperatura, pesaje y señal Wi-Fi (Figura 91).
- Incluir la visualización de la señal Wi-Fi de cada uno de los ESP32 con un baremo por colores según el valor de la señal (Figura 90).
- Diseño y fabricación de dos placas de circuito impreso de diseño diferentes (Figura 113 y Figura 114).

Por otra parte, aunque se han conseguido muchos objetivos en el proyecto hay diferentes puntos que no se han podido desarrollar como son:

- Visualización de los estados ON/OFF de los sensores binarios en la herramienta Grafana.
- Medición óptima del parámetro de pesaje.
- Montaje y pruebas de las placas de circuito impreso.

6. PRESUPUESTO Y TRABAJOS FUTUROS.

Para la realización del proyecto, se hace un estudio técnico y económico. Por ello, en este apartado se van a ver los presupuestos de los modelos propuestos en puntos anteriores.

6.1. Presupuesto

6.1.1. Presupuesto con módulos

En este primer punto podemos observar el presupuesto realizado para el modelo presentado como PCB módulos. En la Tabla 14 se presenta un modelo de presupuesto lo más económico posible y se proporciona el precio por unidad de componente, precio por unidad de proyecto y precio de 20 unidades.

Teniendo en cuenta que se van a usar todos los sensores en una sola parte de la vivienda y utilizando una PCB podemos ver que el precio por unidad sería de 134,24€.

Tabla 14. Presupuesto utilizando módulos

Nombre	Referencia	Tipo	Cantidad	Precio unidad	Precio total 1 und	Precio 20 uds	Fabricante	Distribuidor
Raspberry Pi 3 Model B			1	15,99 €	15,99 €	319,80 €		Amazon
Sensor caudal			3	10,19 €	30,57 €	611,40 €		Amazon
Sensor presión asiento			2	13,09 €	26,18 €	523,60 €		Amazon
Termopar tipo K			1	3,90 €	3,90 €	77,92 €		Amazon
Células de carga			1	11,99 €	11,99 €	239,80 €		Amazon
PCB			1	5,00 €	5,00 €	100,00 €		JLC
Stencil			1	11,06 €	11,06 €	221,20 €		JLC
ESP32-WROOM-32	U1	Integrado	1	4,75 €	4,75 €	95,00 €		AliExpress
HX711	U2	Módulo	1	0,99 €	0,99 €	19,80 €		AliExpress
MAX6675	U6	Módulo	1	2,89 €	2,89 €	57,80 €		AliExpress
AP2204K-3.3	U4	Integrado	1	0,49 €	0,49 €	9,80 €	Diodes Incorporated	Mouser
MCP73831T-2ACI/OT	U3	Integrado	1	0,77 €	0,77 €	15,40 €	Microchip Technology	Mouser
ADS1115	U5	Módulo	1	5,04 €	5,04 €	100,80 €		AliExpress
Conector pin	J2, J1, J2, J3, J4, J5, J6, J7, J8, J9, J10, J11, J12, J13, J14, J15, J10		1	9,79 €	9,79 €	195,80 €		Amazon
Micro_USB	J10		1	0,30 €	0,30 €	6,00 €		Amazon
RRR040P03HZGTL	Q1, Q2	Transistor	2	0,80 €	1,60 €	32,00 €	ROHM Semiconductor	Mouser
R_0603	R1, R2, R3, R4, R5, R6, R7, R8, R9, R10, R11	Resistencia	11	0,09 €	1,02 €	20,46 €	KEMET	Mouser
C_0603	C1, C2	Condensador	2	0,17 €	0,33 €	6,64 €	KEMET	Mouser
C_0603	C3	Condensador	1	0,32 €	0,32 €	6,46 €	KEMET	Mouser
C_0603	C4, C5	Condensador	2	0,49 €	0,98 €	19,52 €	KEMET	Mouser
C_0603	C6, C8	Condensador	2	0,09 €	0,18 €	3,68 €	KEMET	Mouser
C_0603	C7	Condensador	1	0,10 €	0,10 €	1,90 €	KEMET	Mouser

TOTAL: 134,24 € 2.684,78 €

Partiendo de que no se conoce qué sensores se van a conectar se puede estimar que el precio de la placa con los componentes fijos y la Raspberry pi es de 57,72€.

Con esta estimación se puede comprobar que el precio elevado de la Tabla 14 es debido a los sensores y que el precio de este variará en función de la aplicación que se le quiera dar.

6.1.2. Presupuesto con integrados

Como se observa en la Tabla 15 el precio se ha encarecido dejando el precio de una placa por 328,96€ teniendo en cuenta que se conectan todos los sensores.

Tabla 15. Presupuesto con integrados

Nombre	Referencia	Tipo	Cantidad	Precio unidad	Precio total 1 und	Precio 20 uds	Fabricante	Distribuidor
Raspberry Pi 3 Model B			1	15,99 €	15,99 €	319,80 €		Amazon
Sensor caudal		Sensor	3	10,19 €	30,57 €	611,40 €		Amazon
Sensor presión de asiento		Sensor	2	13,09 €	26,18 €	523,60 €		Amazon
Termopar tipo K		Sensor	1	3,09 €	3,09 €	61,80 €		Amazon
Células de carga		Sensor	1	11,99 €	11,99 €	239,80 €		Amazon
PCB			1	5,00 €	5,00 €	100,00 €		JLC
Stencil			1	11,06 €	11,06 €	221,20 €		JLC
ESP32-WROOM-32	U1	Integrado	1	4,75 €	4,75 €	95,00 €		AliExpress
HX711	U6	Integrado	1	0,99 €	0,99 €	19,80 €		Mouser
MAX6675	U2	Integrado	1	14,92 €	14,92 €	298,40 €		Mouser
AP2204K-3.3	U4	Integrado	1	0,49 €	0,49 €	9,80 €	Diodes	Mouser
MCP73831T-2ACI/OT	U3	Integrado	1	0,77 €	0,77 €	15,40 €	Microchip	Mouser
ADS1115	U5	Integrado	1	6,20 €	6,20 €	124,00 €		Mouser
Conector pin	J1, J2, J3, J4, J5, J6, J7, J8, J9, J11, J12, J13, J14, J15, J16, J17, J18, J19, J20, J21, J22	Conector	19	9,79 €	186,01 €	3.720,20 €		Amazon
Micro_USB	J10	Conector	1	0,30 €	0,30 €	6,00 €		Amazon
S8550	Q2	Transistor	1	1,69 €	1,69 €	33,80 €		Mouser
RRR040P03HZGTL	Q1, Q3	Transistor	2	0,80 €	1,60 €	32,00 €	ROHM	Mouser
R_0603	R1, R2, R3, R4, R5, R6, R7, R8, R9, R10, R11, R12, R13, R14, R15, R16, R17	Resistencia	17	0,09 €	1,53 €	30,60 €	KEMET	Mouser
C_0603	C1, C2, C9, C10, C13, C14	Condensador	5	0,33 €	1,65 €	33,00 €	KEMET	Mouser
C_0603	C3	Condensador	1	0,32 €	0,32 €	6,40 €	KEMET	Mouser
C_0603	C11, C12	Condensador	2	0,98 €	1,96 €	39,20 €	KEMET	Mouser
C_0805	C4, C5	Condensador	2	0,18 €	0,36 €	7,20 €	KEMET	Mouser
C_0805	C7	Condensador	1	0,48 €	0,48 €	9,60 €	KEMET	Mouser
C_0805	C6, C8	Condensador	2	0,53 €	1,06 €	21,20 €	KEMET	Mouser

TOTAL	328,96 €	6.579,20 €
-------	----------	------------

6.2. Trabajos futuros

Este trabajo se ha centrado en la obtención y monitorización de los valores proporcionados por los sensores, pero también introduce otros enfoques como la creación de las diferentes placas y el uso del gestor de base de datos.

Se trata de un proyecto que abarca una gran cantidad de objetivos que no se han podido realizar en el tiempo de desarrollo de este trabajo por lo que existen mejoras y trabajos que se pueden implementar en el futuro para dicha optimización como pueden ser:

- Montaje y prueba de las diferentes placas utilizando la alimentación por batería o USB.
- Estudio del consumo de batería para las placas.
- Búsqueda y estudio de otros tipos de células de carga para el pesaje de la cama.
- Mejora de la visualización de datos desde la herramienta Grafana.
- Diseño e impresión de un modelo 3D para colocar las células de carga usadas en el proyecto.
- Diseño e impresión de un modelo 3D que contenga la placa desarrollada para una aplicación física.

7. Bibliografía

- [1] United Nations Department of Public Information, “Comunicado de Prensa, Perspectivas de la Población Mundial. 2019,” pp. 2–2, 2019.
- [2] “<https://www.aal-europe.eu/>.”
- [3] Instituto Nacional de Estadística, “INE: Encuesta Continua de Hogares.” 2020.
- [4] F. M. Calatrava Nicolás, F. J. Ortiz Zaragoza, J. A. Vera Repullo, J. Roca González, M. Jiménez Buendía, and Ó. Martínez Mozos, “Sistema heterogéneo para la monitorización de la actividad diaria en el hogar y el bienestar de personas mayores,” in *XLII JORNADAS DE AUTOMÁTICA : LIBRO DE ACTAS*, 2021. doi: 10.17979/spudc.9788497498043.632.
- [5] A. Recuero, “Estado actual y perspectivas de la domótica,” *Informes de la Construcción*, vol. 50, no. 459, 1999, doi: 10.3989/ic.1999.v50.i459.827.
- [6] “<https://www.un.org/es/global-issues/ageing>.”
- [7] M. N. Kamel Boulos *et al.*, “Connectivity for healthcare and well-being management: Examples from six European projects,” *Int J Environ Res Public Health*, vol. 6, no. 7, 2009, doi: 10.3390/ijerph6071947.
- [8] S. Chernbumroong, A. S. Atkins, and H. Yu, “Perception of smart home technologies to assist elderly people,” 2010.
- [9] “<http://www.hope-project.eu/>.”
- [10] H. Medjahed, “Environnement multimodal pour la télévigilance médicale à domicile EMUTEM.,” *SFTAG 2009: 1er Congrès de la Société Française des Technologies pour l’Autonomie et de Gérontechnologie.*, 2009.
- [11] M. Cazorla, “SIRMAVED: Development of a comprehensive monitoring and interactive robotic system for people with acquired brain damage and dependent people,” *Conferencia CAEPIA, Albacete, Spain.* , 2015.
- [12] A. Garcia-Garcia *et al.*, “Multi-sensor 3D object dataset for object recognition with full pose estimation,” *Neural Comput Appl*, vol. 28, no. 5, 2017, doi: 10.1007/s00521-016-2224-9.
- [13] L. Pigni *et al.*, “Pilot Test of a New Personal Health System Integrating Environmental and Wearable Sensors for Telemonitoring and Care of Elderly People at Home (SMARTA Project),” *Gerontology*, vol. 63, no. 3, 2017, doi: 10.1159/000455168.
- [14] F. M. Calatrava-Nicolás *et al.*, “Robotic-based well-being monitoring and coaching system for the elderly in their daily activities,” *Sensors*, vol. 21, no. 20, 2021, doi: 10.3390/s21206865.
- [15] G. Morales, “La domótica como herramienta para un mejor confort, seguridad y ahorro energético.,” *Ciencia e Ingeniería*, vol. 32, no. 1, 2011.
- [16] J. HUIDOBRO and E. MILLAN, “MANUAL DE DOMOTICA ,” *Madrid, España: Creaciones Copyright*, 2010.
- [17] “EDF, FIEE, FNB, FNEE, IFB. «Domotique».”

- [18] H. Paz Penagos, “Consideraciones técnicas sobre transmisión de información a través de la red eléctrica domiciliaria.,” *Tecnura 7.13* , pp. 57–70, 2003.
- [19] “<https://www.knx.es/es>.”
- [20] M. I. Pizarro Robles, “Inmótica y gestión energética bajo el estándar KNX.,” *Universidad Andrés Bello*, 2018.
- [21] “https://domoticasistemas.com/tienda/tutoriales/1_sistemas-existentes-tipos-y-estandares.html.”
- [22] C. P. Alvarez Barreno, D. O. Holguin Cabezas, E. Serrano Guevara, and E. Bolivar Leyton, “Diseño de una instalación domótica en un condominio para el control de seguridad e iluminación mediante la tecnología lonworks.,” *dspace.espol.edu.ec*, 2009.
- [23] C. CALAFAT, “Introducción a la Tecnología Lonworks.,” *Asociación de Usuarios LonUsers España, Barcelona–España.*
- [24] D. Gualpa Yumi and I. Maza Alcañiz, “Diseño e implementación de un sistema basado en equipos de tecnología LonWorks para la docencia de la Domótica.,” *Universidad de Sevilla. Departamento de Ingeniería de Sistemas y Automática*, 2021.
- [25] “<https://www.z-wave.com/faq>.”
- [26] C. Paetz, “Z-Wave Essentials. ,” *eBook Partnership*, 2017.
- [27] S. Safaric and M. Kresimir, “ZigBee wireless standard,” *Proceedings ELMAR 2006 IEEE*, 2006.
- [28] S. S. R. Ahamed, “The role of zigbee technology in future data communication system,” *J Theor Appl Inf Technol*, vol. 5, 2009.
- [29] F. Meléndez and M. Cano Edgar, “Metodologías de la Investigación PATHFI: Sistema portátil de monitoreo de temperatura, humedad relativa, presión atmosférica y altitud utilizando el protocolo Wi-Fi,” 2014.
- [30] J. A. Carballar, *Wi-Fi. Lo que necesitas conocer.*, RC Libros. 2010.
- [31] J. E. Vega Pérez and J. C. Guerri Cebollada, “Proyecto domótico basado en plataformas abiertas y medio inalámbrico: DomOpenHW,” *Universidad Politécnica de Valencia, Valencia*, 2016.
- [32] B. K. Sovacool and D. D. Furszyfer Del Rio, “Smart home technologies in Europe: A critical review of concepts, benefits, risks and policies,” *Renewable and Sustainable Energy Reviews*, vol. 120. 2020. doi: 10.1016/j.rser.2019.109663.
- [33] R. Lutolf, “Smart Home Concept and the Integration of Energy Meters Into a Home Based System,” *Metering Apparatus and Tariffs for Electricity Supply. Seventh International Conference*, 1992.
- [34] F. K. Aldrich, “Smart Homes: Past, Present and Future,” in *Inside the Smart Home*, 2006. doi: 10.1007/1-85233-854-7_2.
- [35] L. C. de Silva, C. Morikawa, and I. M. Petra, “State of the art of smart homes,” *Eng Appl Artif Intell*, vol. 25, no. 7, 2012, doi: 10.1016/j.engappai.2012.05.002.

- [36] N. Balta-Ozkan, B. Boteler, and O. Amerighi, “European smart home market development: Public views on technical and economic aspects across the United Kingdom, Germany and Italy,” *Energy Res Soc Sci*, vol. 3, no. C, 2014, doi: 10.1016/j.erss.2014.07.007.
- [37] BPIE, “Smart buildings decoded: the concept beyond the buzzword. ,” *Brussels*. 2017.
- [38] C. Wilson, T. Hargreaves, and R. Hauxwell-Baldwin, “Smart homes and their users: a systematic analysis and key challenges,” *Pers Ubiquitous Comput*, vol. 19, no. 2, 2015, doi: 10.1007/s00779-014-0813-0.
- [39] Y. Strengers and L. Nicholls, “Convenience and energy consumption in the smart home of the future: Industry visions from Australia and beyond,” *Energy Res Soc Sci*, vol. 32, 2017, doi: 10.1016/j.erss.2017.02.008.
- [40] J. Shin, Y. Park, and D. Lee, “Who will be smart home users? An analysis of adoption and diffusion of smart homes,” *Technol Forecast Soc Change*, vol. 134, 2018, doi: 10.1016/j.techfore.2018.06.029.
- [41] K. Gram-Hanssen and S. J. Darby, “‘Home is where the smart is’? Evaluating smart home research and approaches against the concept of home,” *Energy Res Soc Sci*, vol. 37, 2018, doi: 10.1016/j.erss.2017.09.037.
- [42] D. Marikyan, S. Papagiannidis, and E. Alamanos, “A systematic review of the smart home literature: A user perspective,” *Technol Forecast Soc Change*, vol. 138, 2019, doi: 10.1016/j.techfore.2018.08.015.
- [43] “<https://www.raspberrypi.com/products/raspberry-pi-3-model-b/>.”
- [44] “<https://www.az-delivery.de/en/products/esp32-developmentboard>.”
- [45] “<https://www.unirioja.es>.”
- [46] “<https://www.digikey.es>.”
- [47] “<https://www.home-assistant.io/installation/>.”
- [48] “https://esphome.io/guides/getting_started_hassio.html.”
- [49] “<https://esphome.io/index.html>.”
- [50] “<https://diptrace.com/es/diptrace-software/>.”

ANEXO: Programación ESPHome

esp32-dormitorio.yaml

```

1 esphome:
2   name: esp32-dormitorio
3
4 esp32:
5   board: esp32dev
6   framework:
7     | type: arduino
8
9 # Enable logging
10 logger:
11
12 # Enable Home Assistant API
13 api:
14   encryption:
15     | key: "o4QZA570PUPqWtMPLH/gH9MTtvQTWkQgFx42ancKWBY="
16
17
18 ota:
19   password: "f28e86202a520b5fed311550f727ead7"
20
21 wifi:
22   ssid: SFR_CD48
23   password: 7as125jnmT8wiw96hbka
24
25   manual_ip:
26     static_ip: 192.168.1.92
27     gateway: 192.168.1.1
28     subnet: 255.255.255.0
29
30 # Enable fallback hotspot (captive portal) in case wifi connection fails
31 ap:
32   ssid: "Esp32-Dormitorio"
33   password: "s1AE5hw0VR1l"
34
35 captive_portal:
36
37 sensor:
38   - platform: hx711
39     name: "Peso Cama"
40     id: pesocama
41     dout_pin: 14
42     clk_pin: 12
43     filters:
44       - calibrate_linear:
45         - 215770 -> 0
46         - 228150 -> 45
47     update_interval: 30s
48     unit_of_measurement: kg
49     accuracy_decimals: 4
50     gain: 128
51
52   - platform: wifi signal
53     name: "WiFi Signal Sensor Dormitorio"
54     update_interval: 60s
55
56 binary_sensor:
57   - platform: template
58     name: "Persona en cama"
59     device_class: occupancy
60     lambda: |-
61       if (id(pesocama).state > 40) {
62         return true;
63       } else {
64         return false;
65       }
  
```

esp32-saloncocina.yaml

```

1  esphome:
2    name: esp32-saloncocina
3
4  esp32:
5    board: esp32dev
6    framework:
7      type: arduino
8
9  # Enable logging
10 logger:
11
12 # Enable Home Assistant API
13 api:
14 encryption:
15   key: "TlwFxe/53oCFd+9Z9r+07Qy6jXQvH45Re6H3lozxpFg="
16
17 ota:
18   password: "1c52315dfeba5457e0e0bdb2f9f9a6e0"
19
20 wifi:
21   ssid: SFR_CD48
22   password: 7asi25jnmT8wiw96hbka
23
24   manual_ip:
25     static_ip: 192.168.1.81
26     gateway: 192.168.1.1
27     subnet: 255.255.255.0
28
29 # Enable fallback hotspot (captive portal) in case wifi connection fails
30 ap:
31   ssid: "Esp32-Saloncocina"
32   password: "Zc8xxOCGbaKO"
33
34 captive_portal:
35
36
37 spi:
38   miso_pin: 19
39   clk_pin: 18
40
41 sensor:
42 - platform: max6675
43   name: "Temperatura vitrocerámica"
44   id: vitrottemp
45   cs_pin: 2
46   update_interval: 30s
47
48 - platform: wifi_signal
49   name: "WiFi Signal Sensor Salon-Cocina"
50   update_interval: 60s
51
52
53 binary_sensor:
54 - platform: gpio
55   id: fregadero
56   device_class: door
57   pin:
58     number: 32
59     mode: INPUT_PULLUP
60   filters:
61     - invert:
62       name: "Fregadero"
63
64 - platform: gpio
65   id: sillon
66   device_class: occupancy
67   pin:
68     number: 27
69     mode: INPUT_PULLUP
70   filters:
71     invert:
72       name: "Presencia en sillon"
73
74 - platform: template
75   name: "Vitroceramica en uso"
76   device_class: power
77   lambda: |-
78     if (id(vitrottemp).state > 50) {
79       return true;
80     } else {
81       return false;
82     }
  
```

esp32-aseo.yaml

```
20 wifi:
21   ssid: SFR_CD48
22   password: 7asi25jnm8wiw96hbka
23
24 manual_ip:
25   static_ip: 192.168.1.72
26   gateway: 192.168.1.1
27   subnet: 255.255.255.0
28
29 # Enable fallback hotspot (captive portal) in case wifi connection fails
30 ap:
31   ssid: "Esp32-Aseo Fallback Hotspot"
32   password: "r8pBVYS8ESMD"
33
34 captive_portal:
35
36 sensor:
37 - platform: wifi_signal
38   name: "Señal Wifi Aseo"
39   update_interval: 60s
40
41 binary_sensor:
42 - platform: gpio
43   id: lavabobano
44   device_class: door
45   pin:
46     number: 26
47     mode: INPUT_PULLUP
48   filters:
49     - invert:
50       name: "Lavabo Manos Baño"
51
52 - platform: gpio
53   id: bide
54   device_class: door
55   pin:
56     number: 25
57     mode: INPUT_PULLUP
58   filters:
59     - invert:
60       name: "Bidé"
61
62 - platform: gpio
63   id: ducha
64   device_class: door
65   pin:
66     number: 15
67     mode: INPUT_PULLUP
68   filters:
69     invert:
70       name: "Ducha"
71
```