



*ugr*

Universidad  
de **Granada**

TRABAJO FIN DE MÁSTER  
INGENIERÍA INFORMÁTICA

# **Uso de un entorno de hogar digital para la monitorización y control de personas.**

---

**Autor**

Arturo Cortés Sánchez

**Directores**

Francisco Luis Gutiérrez Vela



Escuela Técnica Superior de Ingenierías Informática y de  
Telecomunicación

—  
Granada, Septiembre de 2021



# **Uso de un entorno de hogar digital para la monitorización y control de personas**

Arturo Cortés Sánchez

**Palabras clave:** Domótica, Internet de las cosas, Monitorización, Home Assistant, Personas mayores, Cuidados, Android, MQTT, Zigbee, Sensor IoT, Flutter, Tercera edad.

## **Resumen**

Durante este trabajo se ha desarrollado un sistema de monitorización de personas mayores cuya principal cualidad es ser capaz de aprovechar un entorno domótico previamente instalado en el hogar.

El sistema consiste en una Raspberry Pi 3 con una antena Zigbee y una instalación de Home Assistant adecuadamente configurada, un add-on para Home Assistant, dos aplicaciones móviles para android (fácilmente portables a iOS) y un conjunto de sensores IoT. El desarrollo se ha realizado con sensores de Xiaomi y de Ikea, pero gracias a home assistant no debería costar demasiado trabajo incorporar soporte para nuevos sensores.

Desde su correspondiente aplicación la persona encargada del cuidado, puede definir una serie actividades cuya realización va enlazada a lo que detecten los sensores, por ejemplo se puede colocar un sensor en un pastillero para detectar si la persona dependiente ha hecho uso de él, y en qué momento lo ha hecho.

Cuando una actividad no ha sido realizada en la hora que el cuidador seleccionó al definirla, tanto el cuidador como la persona mayor recibirán una notificación alertando de ello.

Finalmente la aplicación de la persona mayor es capaz de detectar caídas de la persona dependiente y avisar de ello.

# **Índice**

- 1. Introducción.**
- 2. Estado del arte**
- 3. Análisis inicial del problema**
- 4. Tecnología a usar**
- 5. Metodologías a usar en el proyecto.**
- 6. Plan de entregas**
- 7. Desarrollo. Entregas e iteraciones.**
- 8. Conclusiones y Trabajo futuro.**
- 9. Bibliografía.**

## **1. Introducción:**

Vivimos en una sociedad cada vez más envejecida, según el informe de la Organización Mundial de la Salud, para 2050 se espera que la población mundial de 60 años o más ascienda a 2.000 millones, frente a los 900 millones de 2015. Actualmente, 125 millones de personas tienen 80 años o más. [1]

El envejecimiento supone una gran presión y muchos ancianos viven solos sin que nadie les陪伴 porque sus hijos están ocupados con el trabajo. Para los ancianos que viven de forma independiente en sus propios hogares, existe un riesgo creciente de caídas y accidentes cerebrovasculares que podrían amenazar sus vidas.

En Estados Unidos, millones de personas mayores de 65 años sufren accidentes por caídas. De hecho, se calcula que uno de cada cuatro ancianos se cae cada año y, dentro de ellos, 2,8 millones de personas mayores son atendidas en los servicios de urgencias por lesiones por caídas. Los accidentes por caídas se han convertido en algo muy frecuente entre las personas mayores. [2]

El New England Journal of Medicine llegó a la conclusión de que las posibilidades de sobrevivir a una caída, un infarto o un ictus son seis veces mayores si los ancianos reciben ayuda en menos de una hora. Por tanto, los ancianos que no están acompañados necesitan una vigilancia en tiempo real para reducir la ansiedad de los mismos y el riesgo de accidentes [3]

No obstante y gracias al desarrollo de tecnologías como la informática móvil, la informática distribuida y las redes de sensores inalámbricos, es posible ofrecer a los ancianos servicios de asistencia sanitaria que puedan monitorizarlos en cualquier momento y lugar.

Por ejemplo, en Australia, Pro Medicus es un éxito de la sanidad informática. Desarrolla un sistema inteligente seguro para transmitir los resultados médicos a los médicos. La red de sensores inalámbricos proporciona un método útil para adquirir y monitorizar a distancia las señales fisiológicas sin necesidad de interrumpir la vida normal del paciente. [4]

La ajetreada vida actual ha causado que la gente que tiene a personas mayores a su cargo tengan dificultades para estar pendientes de ellas, es por ello que existe la necesidad de una herramienta que permita realizar un seguimiento a distancia de las acciones de una persona mayor que viva sola.

Este proyecto pretende ayudar a los cuidadores a suplir las necesidades de las personas mayores. Para ello se desarrollará una herramienta que en conjunto con un hogar inteligente permitirá el monitorizar las actividades de la vida diaria de las personas mayores.

El objetivo principal del proyecto es diseñar e implementar un sistema de domótica en una casa típica, que pueda ser utilizado para monitorear las actividades de las

personas que viven en ella. El segundo objetivo es el diseño de un sistema móvil que facilite la configuración y monitorización visual tanto de la domótica (controladores, sensores y actuadores) como de las actividades monitorizadas. Se prestará especial atención a la notificación en tiempo real de cualquier problema detectado en el sistema, la casa y las personas que la habitan.

El resultado consistirá en un prototipo funcional de un entorno doméstico inteligente, adaptado a las necesidades de las personas mayores, que permitirá a los cuidadores y familiares estar informados de cualquier anomalía en la casa y en las personas que la habitan.

Capítulos:

- En el estado del arte se explican conceptos como la monitorización, domótica, sensores, actuadores, plataformas de control y se finaliza realizando una exploración del mercado
- En el análisis inicial del problema se describe una aproximación inicial al problema se expone una primera solución al problema, se estudia la “monitorizabilidad” de distintas actividades de la vida diaria, y se realiza un modelo conceptual.
- En el apartado de tecnologías a usar se explican en profundidad las distintas tecnologías usadas en este proyecto: Zigbee, MQTT, Home Assistant, Raspberry Pi, Mi Smart Sensor set e Ikea TRÅDFRI.
- En el apartado de metodologías se describe la metodología ágil que se ha aplicado durante el desarrollo del proyecto.
- El apartado del plan de entregas, además de incluir un plan de entregas, contiene las historias de usuario y un presupuesto del proyecto.
- Desarrollo. Entregas e iteraciones
- Conclusiones y trabajo futuro

## 2. Estado del arte

### 2.1 Introducción

Las personas mayores son especialmente vulnerables a la soledad y el aislamiento social, y esto puede tener un grave efecto sobre la salud. Pero hay formas de superar la soledad, incluso si vive solo y le resulta difícil salir.

Cientos de miles de personas mayores se sienten solas y aisladas de la sociedad en este país, especialmente los mayores de 75 años.

Según Age UK, más de dos millones de personas mayores de 75 años en Inglaterra viven solas, y más de un millón de personas mayores dicen pasar más de un mes sin hablar con un amigo, vecino o familiar.

Las personas pueden quedar aisladas socialmente por diversas razones, como envejecer o debilitarse, dejar de ser el centro de su familia, dejar el lugar de trabajo, la muerte de cónyuges y amigos, o por discapacidad o enfermedad.

Cualquiera que sea la causa, es sorprendentemente fácil que las personas mayores se sientan solas y vulnerables, lo que puede conducir a la depresión y a un grave deterioro de la salud física y el bienestar. [5]

Un sistema domótico puede ayudar a paliar la soledad por ejemplo con dispositivos de comunicación distribuidos por el hogar, además, con un buen sistema de monitorización, recordatorios y alertas se pueden solventar algunos de los problemas asociados a la soledad, como por ejemplo el deterioro de la salud y reducción en el bienestar.

## 2.2 Domótica

Antes de seguir es conveniente conocer exactamente en qué consisten la domótica y la monitorización.

La palabra "domótica" es una contracción de la palabra latina para casa (domus) y la palabra robótica. La palabra "inteligente" en "casa inteligente" se refiere a que el sistema es consciente del estado de sus dispositivos, lo que se hace a través del protocolo de las tecnologías de la información y la comunicación (TIC) y del Internet de las cosas (IoT).

La automatización del hogar o domótica es la construcción de la automatización de un hogar, llamada hogar inteligente o casa inteligente. Un sistema de domótica supervisa y/o controla atributos del hogar como la iluminación, la climatización, los sistemas de entretenimiento y los electrodomésticos. También puede incluir la seguridad del hogar, como el control de acceso y los sistemas de alarma.

Un sistema de automatización del hogar suele conectar los dispositivos controlados a un hub o "gateway". Para la interfaz de usuario y el control del sistema normalmente se utilizan apps de smartphone o una interfaz web que también puede ser accesible fuera de las instalaciones a través de Internet.

Aunque hay muchos proveedores que compiten entre sí, cada vez se hacen más esfuerzos en favor de los sistemas de código abierto. Sin embargo, el estado actual de la automatización del hogar tiene ciertos problemas, como la falta de medidas de seguridad estandarizadas y la falta de soporte de dispositivos antiguos.

La primera automatización del hogar comenzó con los electrodomésticos, eléctricos o de gas, empezaron a ser viables en la década de 1900 con la introducción de la distribución de energía eléctrica y condujeron a la introducción de lavadoras (1904), calentadores de agua (1889), frigoríficos, máquinas de coser, lavavajillas y secadoras de ropa.

En 1975 se desarrolló la primera tecnología de red domótica de uso general, el X10. Se trata de un protocolo de comunicación para dispositivos electrónicos. Utiliza principalmente el cableado de transmisión de energía eléctrica para la señalización y el control, donde las señales implican breves ráfagas de radiofrecuencia de datos digitales, y sigue siendo el más extendido. En 1978, los productos X10 incluían una consola de comandos de 16 canales, un módulo de lámparas y un módulo de electrodomésticos. [6]

## 2.3 Actividades de la vida diaria

Por otro lado la monitorización consiste en utilizar dispositivos especiales para detectar anomalías en el comportamiento de lo que se desea monitorizar, en este caso una persona.

En el caso de una persona mayor es interesante monitorizar las actividades de la vida diaria.

Las actividades de la vida diaria (AVD) son un término utilizado en el ámbito sanitario para referirse a las actividades de autocuidado diario de las personas. Los profesionales sanitarios suelen utilizar la capacidad o incapacidad de una persona para realizar AVD como medida de su estado funcional, especialmente en lo que respecta a las personas después de una lesión, con discapacidades y a los ancianos. [7]

Entre las AVD más comunes se encuentran alimentarse, bañarse, vestirse, asearse, trabajar, hacer las tareas del hogar, limpiarse después de defecar y el ocio.

Las AVD básicas consisten en tareas de autocuidado que incluyen:

- Bañarse y ducharse
- Higiene y aseo personal (incluido el cepillado/peinado/peinado del cabello)
- Vestirse
- Higiene en el baño (ir al baño, limpiarse y volver a levantarse)
- Movilidad funcional, a menudo denominada "traslado", medida por la capacidad de caminar, entrar y salir de la cama, y subirse y bajarse de una silla.
- Alimentación (no incluye cocinar ni masticar y tragar)

Las AVD básicas incluyen las cosas que muchas personas hacen cuando se levantan por la mañana y se preparan para salir de casa: levantarse de la cama, ir al baño, bañarse, vestirse, asearse y comer.

Las actividades instrumentales de la vida diaria (AIVD) no son necesarias para el funcionamiento fundamental, pero permiten a una persona vivir de forma independiente en una comunidad:

- Limpieza y mantenimiento de la casa
- Gestionar el dinero

- Moverse en la comunidad
- Preparar las comidas
- Comprar alimentos y artículos de primera necesidad
- Tomar los medicamentos prescritos
- Utilizar el teléfono u otra forma de comunicación

Los terapeutas ocupacionales suelen evaluar las AIVD al completar las evaluaciones de los pacientes. La Asociación Americana de Terapia Ocupacional identifica 12 tipos de AIVD que pueden ser realizadas como una co-ocupación con otros:

- Cuidado de otras personas (incluyendo la selección y supervisión de cuidadores)
- Cuidado de animales domésticos
- Crianza de niños
- Gestión de la comunicación
- Movilidad en la comunidad
- Gestión financiera
- Gestión y mantenimiento de la salud
- Establecimiento y mantenimiento del hogar
- Preparación y limpieza de comidas
- Observancias religiosas
- Procedimientos de seguridad y respuestas de emergencia
- Compras

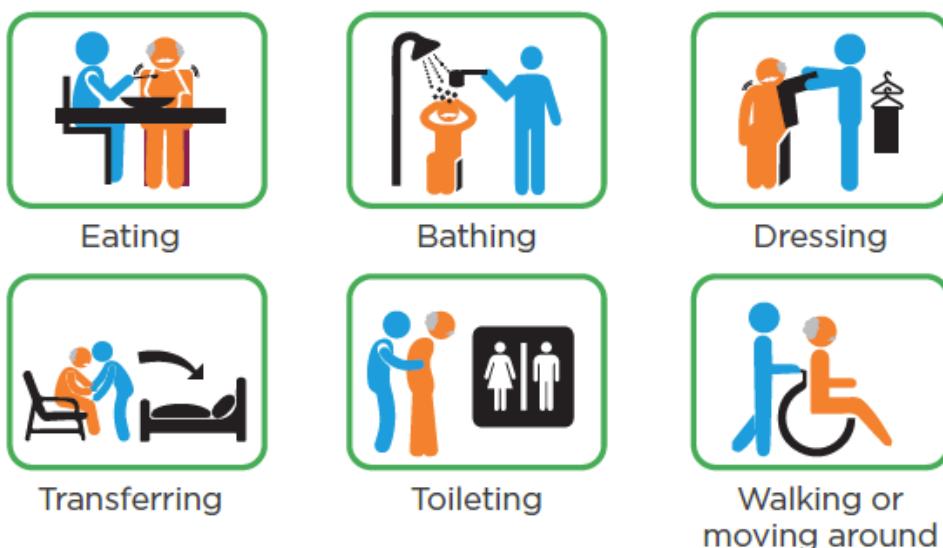


Figura 1. Algunas actividades de la vida diaria

## 2.4 Sensores y Actuadores

Un sensor, es un dispositivo cuya tarea es detectar eventos o cambios en su entorno inmediato y convertir estos fenómenos físicos (como la temperatura, la luz, la humedad del aire, el movimiento, la presencia de sustancias químicas y muchos otros) en impulsos eléctricos que luego pueden interpretarse de forma significativa. Un actuador, en cambio, puede considerarse una herramienta que funciona de forma inversa al sensor. Al interpretar los impulsos eléctricos enviados desde el sistema de control y

convertirlos en movimiento mecánico, introduce realmente cambios en su entorno físico mediante una serie de acciones sencillas, como abrir y cerrar válvulas, cambiar la posición o el ángulo de otros dispositivos, activarlos o emitir sonidos o luz.

#### 2.4.1 Tipos de sensores IoT [8]

##### Sensores de temperatura

Este tipo de sensor más básico encuentra su aplicación en todo tipo de casos de uso del IoT en los que es vital hacer un seguimiento de las condiciones térmicas del aire, el entorno de trabajo, las máquinas u otros objetos. Los sensores de temperatura son especialmente útiles en las plantas de fabricación, almacenes, sistemas de información meteorológica y la agricultura, donde se controla la temperatura del suelo para proporcionar un crecimiento equilibrado y maximizado.

- Termistor: es un tipo de resistor, cuya resistencia depende significativamente de la temperatura. Los termistores se utilizan ampliamente como sensores en la electrónica, por ejemplo, como sensores de temperatura en termómetros electrónicos o en sistemas que evitan el aumento excesivo de la corriente.
- Termopares: elementos de un circuito eléctrico formado por dos conductores diferentes. Aprovechando el hecho de que la fuerza electromotriz que surge entre los conectores de un termopar es proporcional a la diferencia de temperatura, los termopares pueden utilizarse como sensores de temperatura o, a veces, incluso como fuente de energía con una tensión muy baja y una corriente relativamente alta.

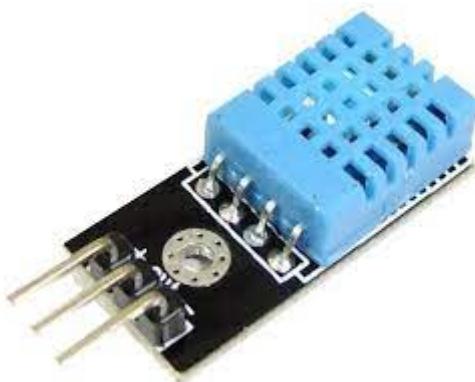


Figura 2. Sensor de temperatura

##### Sensores de humedad

Aunque su uso más obvio y extendido es el de las estaciones meteorológicas para informar y pronosticar el tiempo, sorprendentemente, los sensores de humedad también se emplean ampliamente en la agricultura, la vigilancia del medio ambiente, la cadena de suministro de alimentos, la climatización y la vigilancia de la salud.

- Sensor de humedad de tensión capilar: es el tipo tradicional y más antiguo de sensor de humedad. El diseño del dispositivo se basa en las propiedades específicas del pelo humano o de caballo, aunque actualmente también se utilizan fibras sintéticas o de algodón. Estas fibras cambian su longitud al entrar en contacto con la humedad. El puntero, que muestra la lectura en la balanza,

está conectado al pelo (o a la fibra) y reacciona a un cambio en su longitud. La ventaja de este tipo de sensor de humedad es una construcción sencilla y barata, que además es resistente a los daños.

- Psicrómetro: la construcción de este tipo de dispositivo se basa en dos termómetros (los llamados "seco" y "húmedo"). El termómetro húmedo tiene un recipiente de mercurio revestido de material que absorbe la humedad. Utiliza el fenómeno de la amortiguación de la evaporación por el aire húmedo y la aceleración por el aire seco. Cuanto mayor sea la humedad, menor será la temperatura del termómetro húmedo. El termómetro seco muestra la temperatura ambiente y la humedad se puede calcular a partir de la diferencia de temperatura.

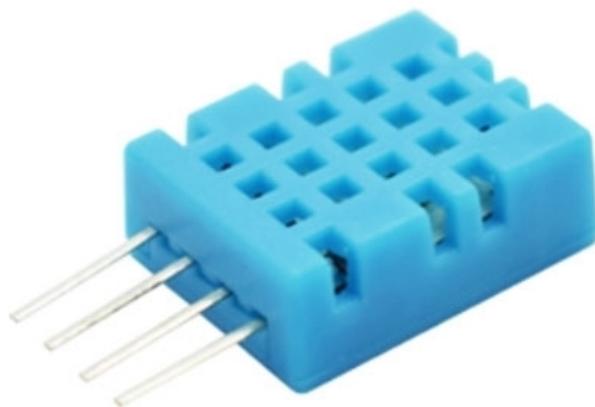


Figura 3. Sensor de humedad

### Sensores de luz

En función de la intensidad de la luz ambiental, los televisores inteligentes, los teléfonos móviles o las pantallas de ordenador pueden ajustar su brillo gracias a los sensores de luz, pero los sensores para detectar la luz ambiental no sólo son habituales en la electrónica de consumo, sino también en las aplicaciones para ciudades inteligentes. Cada vez se utilizan más para adaptar las luces de las calles o los niveles de iluminación urbana con el fin de aumentar la economía.

- Fotoresistencia: es un elemento fotosensible, cuya resistencia cambia por la radiación. Se puede conectar fácilmente, por ejemplo, a Arduino como sensor de luz analógico. Gracias a ello es posible construir, por ejemplo, lámparas que se enciendan automáticamente al anochecer. Los fotorresistores también pueden utilizarse como sensores de proximidad muy sencillos o incluso como sensores de temperatura.
- Fotodiodo: es un diodo que funciona basado en el efecto fotoeléctrico. Cuando los fotones llegan a la unión de un fotodiodo son absorbidos, lo que hace que el electrón se transfiera a la banda de conductividad para crear un par electrón-hueco. Los fotodiodos se utilizan ampliamente en la automatización industrial (sistemas de señalización y control), en las telecomunicaciones (optoacopladores, enlaces optoelectrónicos) y en muchos otros sectores.

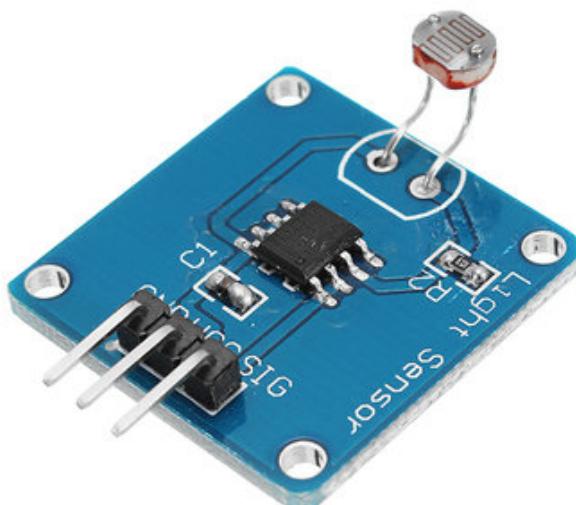


Figura 4. Sensor de luz

#### Sensores acústicos

Los sensores acústicos inteligentes permiten controlar el nivel de ruido en un entorno determinado. Al ser capaces de medir y proporcionar datos para ayudar a la prevención de la contaminación acústica.

- Hidrófono: es un micrófono utilizado para captar los sonidos que se propagan en el agua u otros líquidos. Los hidrófonos son un elemento estructural básico de los sonares pasivos utilizados, por ejemplo, para detectar peces en diversos entornos acuáticos.
- Geófono: es un sensor que convierte las vibraciones del suelo (frecuencia y amplitud) en tensión eléctrica. Se puede decir que es un tipo de sismómetro y se ha utilizado como tal en la luna en el Experimento Sísmico Activo del Apolo 16.

#### Sensores de nivel de agua

Para prevenir las catástrofes naturales, los datos recogidos por los sensores de control del nivel del agua pueden utilizarse en sistemas de alerta de inundaciones para su análisis y predicción. Además de la protección del medio ambiente, este sensor se utiliza en diversas aplicaciones industriales para controlar y optimizar los procesos de fabricación.

- Sensor de presión hidrostática: se utiliza para medir el nivel de llenado de líquidos. Estos sensores funcionan a partir de mediciones de nivel utilizando la llamada paradoja hidrostática: la presión hidrostática medida en el punto de medición del tanque es proporcional sólo a la altura del llenado de líquido, independientemente de la forma y el volumen del tanque.
- Sensor óptico: es un sensor que detecta el nivel de agua implícito en la refracción de la luz en el prisma tras el contacto con el líquido. Los sensores ópticos tienen una cierta ventaja sobre los típicos sensores de nivel de agua debido a que están limitados por piezas mecánicas y móviles que pueden romperse o simplemente desgastarse.

#### Sensores de presencia y proximidad

Al emitir un haz de radiación electromagnética, este tipo de sensor es capaz de detectar la presencia de su objeto objetivo y determinar la distancia que separa a ambos. Con su alta fiabilidad y larga vida útil, no es de extrañar que se hayan introducido rápidamente en tantos sectores del IoT, como los coches inteligentes, la robótica, la fabricación, las máquinas, la aviación e incluso las soluciones de aparcamiento inteligente.

- Radar Doppler: radar que utiliza el efecto Doppler, es decir, la diferencia entre la frecuencia de la onda enviada por su fuente y la frecuencia de la onda registrada por el observador que se mueve con respecto a esta fuente. El radar se utiliza no sólo para detectar objetos y determinar su ubicación, sino también para conocer su dirección y velocidad.
- Sensor de ocupación: es un tipo de sensor de presencia que utiliza luz infrarroja o altas frecuencias para detectar el movimiento en instalaciones de oficinas y sanitarias, pasillos, pasadizos, almacenes, etc.

### Sensores de movimiento

Suelen usarse para ayudar a vigilar los espacios privados o públicos de la intrusión y el robo, aunque el uso de los sensores de movimiento se está extendiendo a las soluciones de gestión de la energía, las cámaras inteligentes, los dispositivos automatizados y muchos otros.

- Sensor de movimiento activo y ultrasónico: envía y recibe ondas pasivas ultrasónicas
- Sensor de movimiento infrarrojo pasivo: detecta cambios en la radiación infrarroja
- Sensor de radar activo: emite y recibe ondas electromagnéticas

También existen versiones de sensores que combinan todos los principios anteriores relacionados con el movimiento. Un ejemplo es el sensor de infrarrojos pasivo (PIR): sensor electrónico de detección de movimiento, que se utiliza habitualmente en sistemas de alarma, iluminación y ventilación automáticas, etc.



Figura 5. Sensor de movimiento

### Giroscopios

La tarea de este tipo de sensor es detectar la rotación y medir la velocidad angular, lo que lo hace perfecto para los sistemas de navegación, la robótica, la electrónica de consumo y los procesos de fabricación que implican rotación. Para una aplicación más

cotidiana del IoT, los sensores giroscópicos se instalan cada vez más en los dispositivos del IoT utilizados por los atletas para medir con precisión los movimientos del cuerpo y analizar y mejorar su rendimiento deportivo.

- Acelerómetro: este sensor no mantiene una dirección constante, sino que indica la velocidad angular del objeto en el que se encuentra. En este grupo se incluyen los giroscopios mecánicos, que tienen una libertad de rotación limitada (normalmente en uno de los ejes del sistema de coordenadas cartesianas), los giroscopios ópticos (láser y fibra óptica) y, por último, los giroscopios que utilizan el efecto Coriolis, que afecta al elemento vibratorio.
- Indicador de rumbo: un giroscopio direccional o indicador de rumbo permite observar la rotación del cuerpo al que está unido. El giroscopio suele ser un objeto rígido de rotación rápida (normalmente un disco) suspendido en una estructura adecuada que permite su rotación libre con respecto al sistema de referencia (por ejemplo, el cuerpo al que está unido).

### Sensores químicos

Los sensores capaces de detectar compuestos químicos (sólidos, líquidos y gases) son elementos indispensables en los sistemas de seguridad industrial, las soluciones de protección del medio ambiente y, obviamente, la investigación científica.

- Alcoholímetro electroquímico: un sencillo sensor utilizado para determinar el contenido de alcohol en sangre. El sensor electroquímico es resistente a efectos secundarios como el mentol, el humo de los cigarrillos o el ácido cítrico y puede indicar la concentración de alcohol en el aire exhalado con una precisión de 0,000 %.
- Nariz electrónica: un detector químico, o más bien un conjunto de detectores, que reacciona a los diferentes tipos de partículas contenidas en el ambiente o a sus diferentes características (como la presencia de enlaces químicos específicos, la acidez, la alcalinidad, la capacidad de estabilizar los dipolos vecinos, etc.). La información obtenida mediante el uso de este conjunto de detectores permite conocer la composición química del entorno.

### Sensores de imagen

Al convertir los datos ópticos en impulsos eléctricos, un sensor de imagen permite al objeto conectado ver el entorno que le rodea y actuar sobre él utilizando la inteligencia obtenida del análisis de los datos proporcionados. Los sensores de imagen se utilizan siempre que es necesario que el dispositivo inteligente "vea" su entorno inmediato, lo que incluye vehículos inteligentes, sistemas de seguridad, equipos militares como radares y sonares, dispositivos de imagen médica y, por supuesto, cámaras digitales.

- Sensor de píxeles activos: disposición de muchos elementos fotosensibles realizada en tecnología CMOS. Los sensores CMOS se encuentran en muchos dispositivos, como cámaras web, cámaras digitales compactas, cámaras DSLR (Digital Single Lens Reflex), elementos de control en la automatización de la producción, cámaras digitales de rayos X, etc.
- Dispositivo de carga acoplada: sistema de muchos elementos fotosensibles que registran y permiten leer una señal eléctrica proporcional a la cantidad de luz que incide sobre él. Los filtros de color utilizados en los sensores CCD se

encuentran muy a menudo en las cámaras digitales, lo que les permite registrar la intensidad de una anchura específica del espectro luminoso en un punto determinado de la matriz.

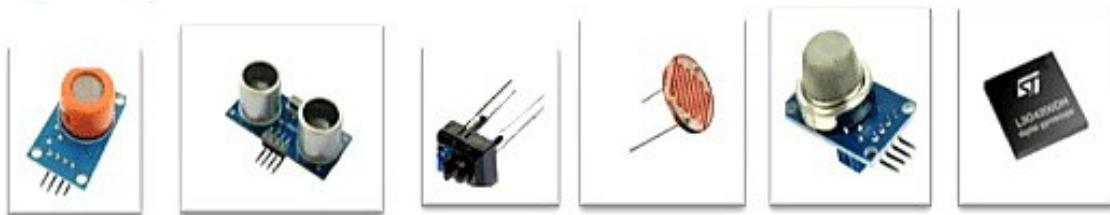


Figura 6. Más modelos de sensores

#### 2.4.2 Tipos de actuadores IoT [9]

Los actuadores, como su propio nombre indica, pueden actuar sobre su entorno inmediato para permitir el correcto funcionamiento de las máquinas o dispositivos en los que están integrados. Por su pequeño tamaño, rara vez son visibles durante su funcionamiento, pero los efectos de su trabajo pueden sentirse en vehículos, máquinas industriales o cualquier otro equipo electrónico que implique tecnologías de automatización. Pueden separarse en cuatro categorías principales en función de su patrón de construcción y del papel que desempeñan en un entorno específico de IoT:

- Actuadores lineales: Un actuador lineal es un actuador que crea un movimiento en línea recta, en contraste con el movimiento circular de un motor eléctrico convencional. Los actuadores lineales se utilizan en máquinas-herramienta y maquinaria industrial, en periféricos informáticos como unidades de disco e impresoras, en válvulas y amortiguadores, y en muchos otros lugares donde se requiere un movimiento lineal.
- Motores: permiten movimientos rotativos precisos de componentes de dispositivos u objetos enteros.
  - Servomotores: Un Servo es un pequeño dispositivo que incorpora un motor de corriente continua de dos hilos, un tren de engranajes, un potenciómetro, un circuito integrado y un eje (columna de salida). El eje puede posicionarse en posiciones angulares específicas enviando al servo una señal codificada. De los tres cables que sobresalen de la carcasa del servo, uno es para la alimentación, otro para la tierra y otro es una línea de entrada de control. Cuando se aplica una señal de control a un servo que representa una posición de salida deseada del eje del servo, el servo aplica energía a su motor de CC hasta que su eje gira a esa posición. Utiliza el dispositivo de detección de posición para determinar la posición de rotación del eje, por lo que sabe hacia dónde debe girar el motor para mover el eje a la posición ordenada.
  - Motores paso a paso: Los motores paso a paso son motores de corriente continua que se mueven en pasos discretos. Tienen múltiples bobinas que se organizan en grupos llamados "fases". Al cargar cada fase en secuencia, el motor girará, un paso a la vez. Con un paso controlado por ordenador, se puede conseguir un posicionamiento y/o control de velocidad muy preciso. Un servomotor consume energía

mientras gira hasta la posición ordenada, pero luego el servomotor descansa. Los motores paso a paso siguen consumiendo energía para bloquear y mantener la posición ordenada.

- Motores de corriente continua: El motor de corriente continua es el actuador más común utilizado en proyectos de electrónica. Son simples, baratos y fáciles de usar. Además, vienen en una gran variedad de tamaños, para adaptarse a diferentes tareas. Los motores de CC convierten la energía eléctrica en mecánica. Constan de imanes permanentes y bucles de alambre en su interior. Cuando se aplica corriente, los bucles de alambre generan un campo magnético, que reacciona contra el campo exterior de los imanes estáticos.



Figura 7. Servo motor

- Relés: Un relé es un interruptor accionado eléctricamente. Muchos relés utilizan un electroimán para accionar mecánicamente un interruptor, pero también se utilizan otros principios de funcionamiento, como los relés de estado sólido. La ventaja de los relés es que se necesita una cantidad de energía relativamente pequeña para hacer funcionar la bobina del relé, pero el propio relé puede utilizarse para controlar motores, calentadores, lámparas o circuitos de corriente alterna que pueden consumir mucha más energía eléctrica.



Figura 8. Relé

- Solenoides: Un solenoide es simplemente un electroimán especialmente diseñado. Los solenoides son baratos y su uso se limita principalmente a aplicaciones de encendido y apagado, como el enclavamiento, el bloqueo y la activación. Se utilizan con frecuencia en electrodomésticos (por ejemplo, válvulas de lavadoras), equipos de oficina (por ejemplo, fotocopiadoras), automóviles (por ejemplo, cierres de puertas y el solenoide de arranque), máquinas de pinball (por ejemplo, émbolos y parachoques) y automatización de fábricas.



Figura 9. Solenoide

## 2.5 Plataformas de control

Las plataformas de control de IoT se consideran el componente más importante del ecosistema de IoT. Cualquier dispositivo de IoT permite conectarse a otros dispositivos y aplicaciones de IoT para transmitir información utilizando protocolos estándar de Internet. Las plataformas de IoT llenan el vacío entre los sensores del dispositivo y las redes de datos. Conecta los datos al sistema de sensores y brinda información utilizando aplicaciones de back-end para crear una idea de la gran cantidad de datos desarrollados por los muchos sensores. [10]

### **2.5.1 openHAB:**

open Home Automation Bus es un software de domótica de código abierto escrito en Java. Se despliega y se conecta a dispositivos y servicios de diferentes proveedores. En 2019, cerca de 300 integraciones están disponibles como módulos OSGi. Las actividades, como el encendido de las luces, se activan mediante reglas, comandos de

voz o controles en la interfaz de usuario de openHAB. El proyecto openHAB comenzó en 2010. En 2013, la funcionalidad principal se convirtió en un proyecto oficial de la Fundación Eclipse bajo el nombre de Eclipse SmartHome. Según [www.openhab.net](http://www.openhab.net), openHAB está desarrollado por uno de los mayores equipos de código abierto del mundo y cuenta con una comunidad de usuarios muy activa.

openHAB funciona sobre la máquina virtual de Java y puede desplegarse en servidores que ejecuten varios sistemas operativos, una instancia dedicada de Raspberry Pi o algunos sistemas de almacenamiento conectados a la red. Las integraciones necesarias pueden añadirse en tiempo de ejecución a través de OSGi. openHAB admite una serie de backends de persistencia para almacenar y consultar los datos del hogar inteligente, incluyendo bases de datos relacionales.

Desde la versión principal 2 de OpenHAB las conexiones a los dispositivos físicos se dividen en 2 niveles. Las cosas son los elementos de interfaz a un dispositivo físico específico (por ejemplo, una interfaz a una red de automatización del hogar como KNX, Z-Wave o ZigBee). Dentro de estas cosas, se pueden definir o descubrir uno o más elementos. Estos elementos corresponden a un componente específico, como un relé que controla una luz, la temperatura deseada de un sistema de calefacción o un porcentaje de atenuación. [11] [12]

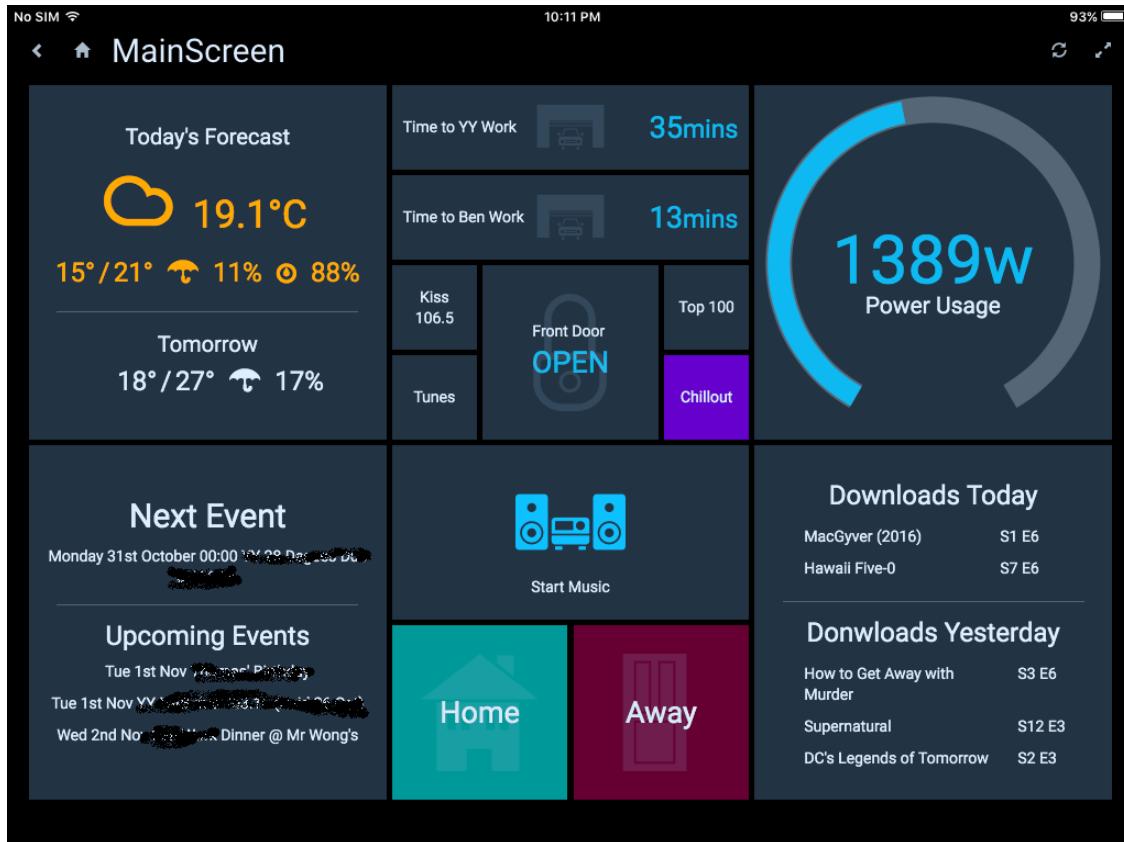


Figura 10. Panel de openHab

## 2.5.2 Home Assistant

Home Assistant es un software de automatización del hogar gratuito y de código abierto diseñado para ser el sistema de control central de los dispositivos domésticos inteligentes, centrándose en el control local y la privacidad. Se puede acceder a él a través de una interfaz de usuario basada en la web, a través de aplicaciones complementarias para Android e iOS, o mediante comandos de voz a través de un asistente virtual compatible como Google Assistant o Amazon Alexa.

Los dispositivos, los servicios y las tecnologías IoT se apoyan en componentes de integración modulares, como protocolos como Bluetooth, MQTT, Zigbee y Z-Wave. La información procedente de estas entidades puede utilizarse dentro de scripts o desencadenar automatizaciones, por ejemplo, para controlar la iluminación, la climatización, los sistemas de entretenimiento y los electrodomésticos.

Home Assistant puede instalarse en múltiples plataformas, entre las que se incluyen los SBC de ODROID, Raspberry Pi, Asus Tinkerboard, Intel NUC, sistemas operativos como Windows, macOS, Linux, así como máquinas virtuales y sistemas NAS. Es posible utilizar Home Assistant como puerta de enlace o puente para dispositivos que utilicen diferentes tecnologías IoT como Zigbee o Z-Wave, el hardware necesario puede montarse en pines de E/S de propósito general o utilizando puertos USB. Además, puede conectarse directa o indirectamente a dispositivos IoT locales, concentradores/pasarelas/puentes de control o servicios en la nube de muchos proveedores diferentes, incluyendo otros ecosistemas domésticos inteligentes abiertos y cerrados. [13][14]

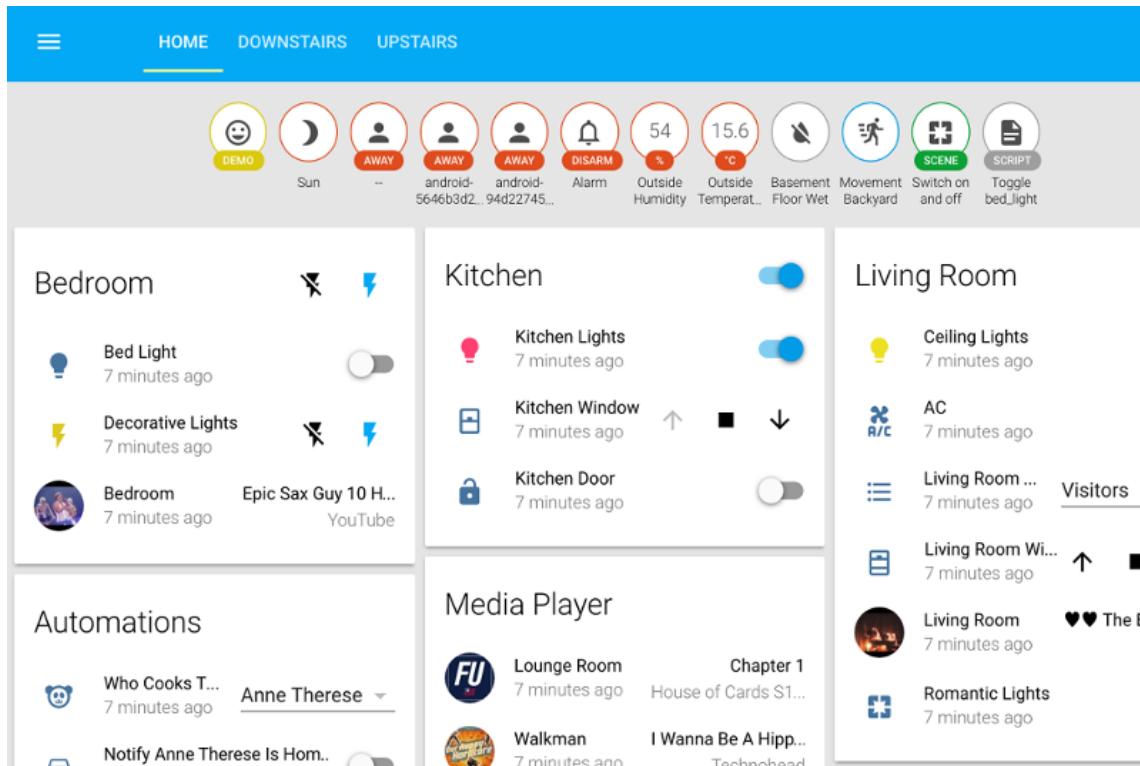


Figura 11. Panel de Home Assistant

## 2.5.4 Domoticz

Domoticz es un sistema de automatización del hogar que le permite controlar y configurar varios dispositivos como: Luces, Interruptores, varios sensores/medidores como temperatura, lluvia, viento, UV, electricidad, gas, agua y mucho más. Se pueden enviar notificaciones/alertas a cualquier dispositivo. Este sistema está diseñado para funcionar en varios sistemas operativos. La interfaz de usuario es un frontend web HTML5 escalable, y se adapta automáticamente a los dispositivos de escritorio y móviles.

La primera versión de Domoticz fue en diciembre de 2012 y está programado en C++. Utiliza su propio servidor web embebido, escrito en C++, para una ejecución eficiente y para evitar dependencias.

La lógica del manejo de eventos es programable por el usuario, utilizando Blockly (para la programación visual) o Lua. [15][16][17]



Figura 12. Panel de Domoticz

|                   | Home Assistant   | openHab   | Domoticz   |
|-------------------|--|---|--|
| Programado en     |  |   |  |
| Disponible en     | <a href="#">Raspberry Pi</a> ,<br><a href="#">ODROID</a> , <a href="#">ASUS</a><br><a href="#">Tinkerboard</a> ,<br><a href="#">Generic x86-64</a> ,<br><a href="#">Windows</a> , <a href="#">macOS</a> ,<br><a href="#">Linux</a> | <a href="#">Raspberry Pi</a> ,<br><a href="#">Linux</a> , <a href="#">Windows</a> ,<br><a href="#">macOS</a> , <a href="#">Docker</a> | <a href="#">Raspberry Pi</a> ,<br><a href="#">Linux</a> , <a href="#">Windows</a> ,<br><a href="#">macOS</a> , <a href="#">Docker</a> ,<br><a href="#">Synology NAS</a> ,<br><a href="#">FreeNAS</a> |
| Número de add-ons | <a href="#">1835</a>   | <a href="#">352</a>   | <a href="#">136</a>  |

|                  |  |   |   |
|------------------|--|---|---|
| Asistente de voz | <u>Almond &amp; Ada</u> ,<br><u>Home assistant</u> | <u>Mycroft AI</u> , <u>Google assistant</u> | <u>Google assistant</u> ,<br><u>Alexa</u> ( |
| Scripting        | Yaml   | Javascript                                  | Visual / Lua                                |
| App para móvil   | Android & iOS                                      | Android & iOS                               | Android & iOS                               |

Tabla 1. Comparativa entre plataformas domóticas

## 2.6 Monitorización de mayores: Sistemas similares

Dado que el objetivo de este proyecto es diseñar un sistema de monitorización de personas mayores, previamente se ha realizado un trabajo de investigación para encontrar productos que cubran un nicho de mercado similar. Los productos encontrados han sido los siguientes:

- TruSense Passive Monitoring es un conjunto de sensores domésticos conectados que actúan como una red de seguridad para facilitar a las personas mayores notificar emergencias. Consta de una Web App en la que el cuidador puede ver entre otras cosas si la persona anciana ha pasado demasiado tiempo en una habitación o si la temperatura ambiental está fuera de los límites seguros.[18]



Figura 13. Dispositivos de TruSense

- Kwido Home es un sistema de monitorización en domicilio que ayuda a empresas de teleasistencia a cuidar de las personas ancianas. Tiene un conjunto de sensores autónomos que monitoriza y alerta sobre comportamientos anómalos de las personas mayores. Además la empresa ofrece servicios extra como uno de estimulación cognitiva u otro de telemedicina. [19]



Figura 14. Dispositivos de Kwido Home

- Rest Assured ofrece un conjunto completo de servicios y soluciones de supervisión remota para ayudar a las personas mayores a mantener una vida sana y segura. Incluye sensores electrónicos, pantallas de comunicación de audio/vídeo bidireccionales, cámaras (en zonas comunes de la vivienda), detectores de humo, detectores de temperatura, sistemas de respuesta a emergencias y mucho más. Aunque a diferencia del resto de servicios que he encontrado, este no incluye imágenes de sus dispositivos.[20]
- AI Senior Safety Monitoring System: Proporciona una red de sensores que junto a un sistema de inteligencia artificial, es capaz de monitorizar a los ancianos de forma similar a una persona. También cuenta con una App que mantiene un historial de comportamiento, proporcionando registros y predicciones que advierten de posibles peligros con antelación. En comparación con los servicios anteriormente mencionados, la web de este último prácticamente no explica nada sobre el funcionamiento del producto. [21]



Figura 14. Dispositivo de AI Senior Safety Monitoring System

## 2.7 Conclusión

A día de hoy los sistemas domóticos están cada vez más extendidos gracias a su bajo precio y a la gran variedad de marcas. Su capacidad de facilitarnos tareas cotidianas mediante una aplicación en el móvil un asistente de voz no para de llamar la atención de un público creciente.

Por otro lado, la población mundial está cada vez más envejecida, y el sector de los cuidados para personas en la tercera edad está sufriendo cada vez más demanda. Es por ello que se puede facilitar esta labor aprovechando un sistema domótico previamente instalado en el hogar. Utilizando un software programado para ello, los sensores de dicho sistema pueden ser utilizados para monitorizar a la persona mayor y sus actividades.

### 3. Análisis inicial del problema

El proyecto consiste en desarrollar un sistema capaz de aprovechar un hogar domotizado para monitorizar personas mayores. Los usuarios de esta aplicación se dividen en dos roles:

- Cuidador: Un cuidador es una persona que se encarga del cuidado de una o varias personas dependientes. Mediante el sistema es capaz de supervisar qué actividades realiza la persona mayor así como su localización aproximada dentro del hogar.
- Persona dependiente: Persona mayor que conserva cierta independencia pero necesita supervisión. El sistema notificará a esta persona cuando se le olvide realizar una actividad. Una persona dependiente puede tener varios cuidadores.

Para cada uno de estos dos roles el sistema contará con una aplicación móvil:

- Aplicación para el cuidador: Esta aplicación se conecta a la nube y recibe información de los sensores distribuidos por todo el hogar. El usuario puede asociar dichos sensores a distintas actividades para recibir notificaciones de lo que está realizando la persona dependiente en cada momento. También podrá ver la posición aproximada dentro del hogar y controlar algunos electrodomésticos.
- Aplicación para la persona dependiente: Esta aplicación monitoriza las posibles caídas o movimientos bruscos utilizando el acelerómetro, además se leerán tarjetas NFC asociadas a actividades. También se recibirán recordatorios en caso de que una actividad no haya sido realizada.

Además de usar tarjetas NFC, la persona dependiente podrá usar botones distribuidos por la casa para notificar acciones rutinarias difíciles de detectar de otra forma, por ejemplo ducharse.

#### 3.1 Actividades monitoreables

| Actividad de la vida diaria | ¿Cómo | podría | ¿Monitorizable? |
|-----------------------------|-------|--------|-----------------|
|-----------------------------|-------|--------|-----------------|

|  |  |    |
|--|--|----|
|  | monitorizarse?   |    |
| Bañarse y ducharse   | Podrían ponerse sensores de movimiento en el baño y un sensor de humedad en la bañera o plato de ducha. Pero dado que no tenemos sensor de humedad sería imposible distinguir si la persona está en el baño por otras razones. | No |
| Higiene y aseo personal (incluido el cepillado /peinado/peinado del cabello) | Igual que en el caso anterior es imposible distinguir por qué está la persona en el baño o la habitación que use para asearse.   | No |
| Vestirse   | Colocando un sensor de apertura y cierre en un armario podemos ver cuando se hace uso de él. Lo que nos da una idea bastante buena de cuando la persona se viste.  | Si |
| Higiene en el baño   | Igual que en casos anteriores, no es posible distinguir las razones por las que la persona está en el baño.  | No |
| Movilidad funcional  | Utilizando sensores de movimiento es posible monitorizar cuanto se mueve por su casa la persona.   | Si |
| Alimentación   | Colocando un sensor de apertura/cierre en el frigorífico o muebles de la cocina es posible monitorizar si la persona hace uso de ellos. Lo cual nos da una idea de si la persona come y de cuando lo hace.                     | Si |
| Limpieza y mantenimiento de la casa  | No hay ningún sensor que pueda medir esto.   | No |
| Gestionar el dinero  | No hay ningún sensor que pueda medir esto.   | No |
| Moverse en la comunidad  | No hay ningún sensor que pueda medir esto.   | No |

|  |   |         |
|--|---|---------|
| Preparar las comidas                               | No sería fácilmente distinguible del caso de la alimentación.   | ¿Puede? |
| Comprar alimentos y artículos de primera necesidad | No hay ningún sensor que pueda medir esto.  | No      |
| Tomar los medicamentos prescritos                  | Si la persona utiliza un pastillero y este es lo suficientemente grande, se podría colocar un sensor de apertura y cierre en él | Si      |
| Utilizar el teléfono u otra forma de comunicación  | Con la aplicación en modo persona dependiente se podría monitorizar el uso del teléfono   | Si      |

Tabla 2: Evaluación de las actividades

Además con el modo persona dependiente se podrían monitorizar las caídas.

En el caso de algunas de las actividades que hemos concluido que no se pueden monitorizar, quedaría la opción de asignarles un botón inalámbrico, de forma que la persona dependiente lo pulse cada vez que vaya a realizar la actividad. No obstante esto puede resultar problemático ya que para una persona mayor sería una acción fácil de olvidar.

A continuación en la figura 15 se muestra un modelo conceptual de las entidades del sistema y cómo interactúan entre ellas.

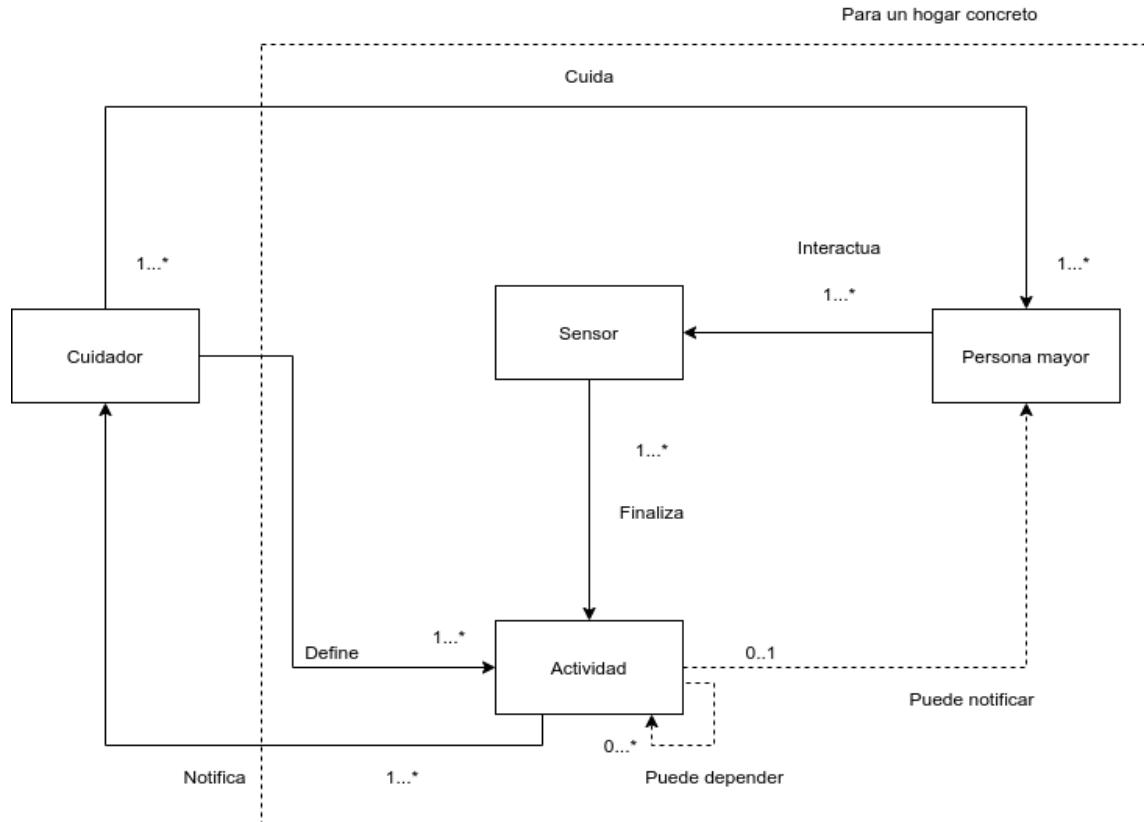


Figura 15: Modelo conceptual del sistema

El sistema está pensado de forma que una persona mayor pueda estar al cuidado de varios cuidadores y un cuidador pueda tener a su cargo a varias personas mayores. Para interactuar con el sistema el cuidador deberá seleccionar primero un hogar, el cual contiene sensores, actuadores y un solo habitante. Se ha decidido que solo habrá un habitante por hogar porque monitorizar a más de una persona con un mismo conjunto de sensores es demasiado complicado, además una persona que no vive sola no tiene tanta necesidad de un sistema como este. El cuidador puede definir varias actividades para dentro de un mismo hogar

Una parte vital a la hora de diseñar el funcionamiento de la aplicación es definir la entidad Actividad, pues todo el sistema gira en torno a ella. Ya sea como una tabla en una base de datos, o una clase en código, una actividad debe de tener:

- Nombre: El nombre de la actividad, será utilizado principalmente para labores de visualización.
- Hogar: Hogar en el que vive la persona que va a realizar la actividad, no es necesario que sea seleccionado en la pantalla de creación de actividad, pues habrá sido seleccionado previamente. Pero si es necesario definir internamente en qué hogar se va a realizar cada actividad para poder relacionarla con los sensores correspondientes.
- Hora: Hora o rango horario en el que se quiere que se realice la actividad
- Margen de notificación: Tiempo máximo que puede pasar tras la hora de realización de la actividad para que se notifique a las personas correspondientes.

- Personas a notificar: Lista de personas que se van a notificar cuando la actividad no haya sido realizada dentro de su horario correspondiente más el margen. Puede incluir al cuidador que define la actividad, otros cuidadores a cargo de este hogar y la persona de la que se está cuidando.
- Descripción: Sección donde el cuidador puede añadir cualquier texto extra para que en caso de necesidad pueda ser visualizado por la persona mayor, por ejemplo podría incluir recordatorios de donde se encuentran algunos objetos necesarios para la tarea en cuestión o los pasos a realizar para finalizarla.
- Lista de sensores: Lista de sensores cuya activación determina que la actividad ha sido realizada.
- Actividades previas: Lista de actividades que deben de haber sido realizadas previamente para que esta actividad pueda darse por finalizada.

## 4. Tecnología a usar

### 4.1 Plataforma de control

Tras probar tanto OpenHAB como Home Assistant hemos optado por el segundo debido a su mejor interfaz, gran capacidad de integración con servicios de notificación externos (actualmente más de 60), su extensa y activa comunidad. Aunque la característica más determinante ha sido la existencia del plugin zigbee2mqtt, que facilita mucho la comunicación con dispositivos zigbee.

Home Assistant está desarrollado utilizando Python 3 para el backend y Polymer (componentes web) para los componentes del frontend. Como producto de código abierto, está licenciado bajo Apache 2.0. Desde el punto de vista del proceso de desarrollo, el proyecto ha adoptado recientemente un ciclo de publicación de dos semanas, con una semana más para las pruebas beta, lo que ha permitido mejorar la calidad, aunque recientemente ha habido publicaciones con errores.

Home Assistant, anteriormente llamado Hass.io, es uno de los dos sabores de Home Assistant y es un sistema operativo que se encargará de instalar y actualizar Home Assistant. Es fácil de gestionar desde la interfaz de usuario de Home Assistant, lo que permite crear/restaurar instantáneas de su configuración y puede ampliarse fácilmente mediante complementos de Hass.io, como Google Assistant y Let's Encrypt.

Como se ha explicado anteriormente Home Assistant, soporta instalaciones sobre un gran número de plataformas, entre ellas ODROID, Raspberry Pi, Asus Tinkerboard etc. [22]



Figura 16: Logotipo de Home Assistant

## 4.2 Plataforma hardware

La plataforma elegida para instalar Home Assistant ha sido la Raspberry Pi 3. Se eligió esta plataforma debido a que su popularidad, comunidad y bajo precio garantizan que sea una de las mejores opciones para implementar un sistema domótico a nivel no comercial.

La Raspberry Pi es una serie de “single board computers” desarrollados en el Reino Unido por la Fundación Raspberry Pi en asociación con Broadcom. El proyecto Raspberry Pi se orientó originalmente a la promoción de la enseñanza de las ciencias informáticas básicas en las escuelas y en los países en desarrollo. El modelo original se hizo más popular de lo previsto, vendiéndose fuera de su mercado objetivo para usos como la robótica. Se utiliza ampliamente en muchos ámbitos, como la vigilancia meteorológica, por su bajo coste, su modularidad y su diseño abierto. Suele ser utilizado por aficionados a la informática y la electrónica, debido a la adopción de dispositivos HDMI y USB. [23]



Figura 17: Raspberry Pi 3

## 4.3 Dispositivos IoT

### 4.3.1 Mi Smart Sensor Set

Se trata de uno de los varios kits de sensores IoT fabricado por Xiaomi, además de los sensores mostrados a continuación , el kit incluye un gateway que debe ser

configurado con una app móvil. Desgraciadamente por razones que se explicarán más adelante, no ha sido posible integrar dicho gateway en home assistant.

El proyecto inicialmente iba a trabajar únicamente con estos sensores, pero al ver que había problemas de integración se optó por comprar unos de otra marca

|   |  |
|---|--|
| Botón inalámbrico Xiaomi WXKG01LM                       |    |
| Detector de movimiento Xiaomi RTCGQ01LM                 |   |
| Sensor de apertura y cierre de puertas Xiaomi MCCGQ01LM |  |

Tabla 3: Dispositivos de Xiaomi

#### 4.3.2 IKEA TRÅDFRI

Conjunto de sensores de la marca de dispositivos IoT de IKEA. A diferencia de los sensores de Xiaomi, estos solo se pueden comprar por separado. Oficialmente requieren un gateway para su funcionamiento, pero igualmente que en el caso anterior no ha sido posible integrar dicho gateway en Home Assistant.

Estos sensores fueron adquiridos con la intención de sustituir a los sensores Xiaomi que habían demostrado ser considerablemente difíciles de integrar en Home Assistant, Desgraciadamente tras probar las integraciones disponibles de IKEA TRÅDFRI vimos que tenían un problema bastante similar. La integración permite interactuar con el hub, pero no con los sensores enlazados a este.

|  |  |
|--|--|
| Enchufe inteligente IKEA E1603/E1702                             |    |
| Control remoto / Botón inalámbrico multifunción IKEA E1524/E1810 |    |
| Botón inalámbrico On/Off IKEA E1743                              |  |
| Botón de acción inalámbrico IKEA E1812                           |  |
| Sensor de movimiento IKEA E1525/E1745                            |  |

Tabla 3: Dispositivos de Ikea

### 4.3.3 Comunicación con los dispositivos IoT

#### **4.3.3.1 Antena Zigbee**

La solución a los problemas de integración de los sensores vino de la mano de un dongle USB que permite a la Raspberry Pi interceptar conexiones Zigbee. El dongle posee entre otras cosas una antena SMA desmontable y un chip CC2531. Este chip es un SOC diseñado para dispositivos USB y con capacidades IEEE 802.15.4, ZigBee y RF4CE. Además viene previamente flasheado por el fabricante con un firmware que lo dota de compatibilidad con Zigbee2mqtt.



Figura 18: Antena Zigbee

#### **4.3.3.2 Zigbee**

Zigbee es una especificación basada en IEEE 802.15.4 para un conjunto de protocolos de comunicación de alto nivel utilizados para crear redes de área personal con radios digitales pequeñas y de baja potencia, como por ejemplo para automatización del hogar, recopilación de datos de dispositivos médicos y otras necesidades de baja potencia y bajo ancho de banda.

La tecnología definida por la especificación Zigbee pretende ser más sencilla y menos costosa que otras redes inalámbricas de área personal (WPAN), como Bluetooth o redes inalámbricas más generales como Wi-Fi. Entre sus aplicaciones están los interruptores de luz inalámbricos, los monitores de energía domésticos, los sistemas de gestión del tráfico y otros equipos industriales y de consumo que requieren una transferencia de datos inalámbrica de corto alcance y baja velocidad.

Su bajo consumo de energía limita las distancias de transmisión a 10-100 metros en línea de visión, dependiendo de la potencia de salida y de las características ambientales los dispositivos Zigbee pueden transmitir datos a largas distancias pasando los datos a través de una red de malla de dispositivos intermedios para llegar a otros más distantes. Zigbee se utiliza normalmente en aplicaciones de baja velocidad de datos que requieren una larga duración de la batería y una red segura. (Las redes Zigbee se aseguran con claves de encriptación simétricas de 128 bits.) Zigbee tiene una tasa definida de 250 kbit/s, más adecuada para transmisiones intermitentes de datos desde un sensor o dispositivo de entrada.

En Zigbee, hay tres tipos diferentes de dispositivos: dispositivo final, router y coordinador. Una red Zigbee siempre tiene un solo coordinador, y puede tener múltiples routers y dispositivos finales. [24]

#### **Dispositivo final :**

Los dispositivos finales no enrutan el tráfico. También pueden dormir, lo que hace que los dispositivos finales sean una opción adecuada para los dispositivos que funcionan con baterías. Un dispositivo final sólo tiene un parent, ya sea el coordinador o un router, generalmente el dispositivo más cercano cuando fue emparejado. Todas las comunicaciones hacia y desde el dispositivo final se realizan a través de su parent. Si un router parent se desconecta, todo el tráfico hacia sus hijos cesará hasta que esos dispositivos finales agoten el tiempo de espera e intenten encontrar un nuevo parent.

#### **Router :**

Los routers son responsables de enrutar el tráfico entre los diferentes nodos. Los routers no pueden dormir. Por ello, los routers no son una opción adecuada para los dispositivos que funcionan con baterías. Los routers también son responsables de recibir y almacenar los mensajes destinados a sus hijos. Además, los routers son los guardianes de la red. Son los responsables de permitir que nuevos nodos se unan a la red.

#### **Coordinador**

Un coordinador es un router especial. Además de todas las capacidades del router, el coordinador es responsable de formar la red. Para ello, debe seleccionar el canal apropiado, el ID de PAN y la dirección de red extendida. También es responsable de seleccionar el modo de seguridad de la red.[25]

### **4.3.3.3 MQTT**

Message Queuing Telemetry Transport, abreviado como MQTT, es un protocolo de red ligero de publicación-suscripción que transporta mensajes entre dispositivos. El protocolo suele ejecutarse sobre TCP/IP; sin embargo, cualquier protocolo de red que proporcione conexiones ordenadas, sin pérdidas y bidireccionales puede soportar MQTT. Está diseñado para conexiones con ubicaciones remotas en las que se requiere una "pequeña huella de código" o el ancho de banda de la red es limitado.

El protocolo MQTT define dos tipos de entidades de red: un broker de mensajes y un número de clientes. Un broker MQTT es un servidor que recibe todos los mensajes de los clientes y luego los enruta a los clientes de destino apropiados. Un cliente MQTT es cualquier dispositivo (desde un microcontrolador hasta un servidor completo) que ejecuta una biblioteca MQTT y se conecta a un broker MQTT a través de una red.

La información se organiza en una jerarquía de topics. Cuando un editor tiene un nuevo dato que distribuir, envía un mensaje de control con los datos al broker conectado. A continuación, el broker distribuye la información a los clientes que se hayan suscrito a ese topic. El editor no necesita tener ningún dato sobre el número o la

ubicación de los suscriptores, y los suscriptores, a su vez, no tienen que estar configurados con ningún dato sobre los editores.

Si un broker recibe un mensaje sobre un tema para el que no hay suscriptores actuales, el broker descarta el mensaje a menos que el publicador del mensaje lo haya designado como mensaje retenido. Un mensaje retenido es un mensaje MQTT normal con la bandera retenida establecida como verdadera. El broker almacena el último mensaje retenido y la correspondiente QoS para el tema seleccionado. Cada cliente que se suscribe a un patrón de tema que coincide con el tema del mensaje retenido recibe el mensaje retenido inmediatamente después de suscribirse. El broker sólo almacena un mensaje retenido por tema, lo que permite a los nuevos suscriptores de un tema recibir el valor más actual en lugar de esperar a la siguiente actualización de un editor.

Cuando un cliente publicador se conecta por primera vez al broker, puede configurar un mensaje por defecto que se enviará a los suscriptores si el broker detecta que el cliente publicador se ha desconectado inesperadamente del broker.

Los clientes sólo interactúan con un broker, pero un sistema puede contener varios servidores broker que intercambian datos en función de los temas de sus suscriptores actuales.

Un mensaje de control MQTT mínimo puede tener tan sólo dos bytes de datos. Un mensaje de control puede transportar casi 256 megabytes de datos si es necesario. Hay catorce tipos de mensajes definidos que se utilizan para conectar y desconectar un cliente de un broker, publicar datos, acusar recibo de datos y supervisar la conexión entre cliente y servidor. [26]

#### **4.3.3.4 Zigbee2mqtt**

Es un software que permite utilizar dispositivos Zigbee sin el puente o gateway de los proveedores.

Hace de puente para los eventos y permite controlar dispositivos Zigbee a través de MQTT. De este modo, los dispositivos Zigbee se pueden integrar con cualquier infraestructura doméstica inteligente de la que se haga uso.

Zigbee ya tiene de por sí una cierta capacidad de automatización del hogar ya definida. Pero si se requiere más flexibilidad de la que viene en los grupos predefinidos de Zigbee, por ejemplo "Encender el ventilador cuando la humedad relativa es superior al 70%", entonces se necesita un control más inteligente. Es aquí donde entra Zigbee2MQTT. Traduce entre Zigbee y MQTT.

Cuando se añade un dispositivo a la red con un controlador Zigbee-Herdsman, el controlador utiliza los clústeres de configuración de bajo nivel para entrevistar al dispositivo y averiguar qué es, qué puntos finales tiene y qué clústeres implementa cada uno de esos puntos finales. Los convertidores Zigbee-Herdsman registran entonces, para cada modelo de dispositivo, qué clusters debe enlazar el controlador, y cómo debe manejarse la conversión a MQTT. La mayoría de los dispositivos en Zigbee-Herdsman-Converters utilizan convertidores genéricos que se unen a cada clúster Zigbee y proporcionan una interfaz MQTT estándar para ese clúster.

Con esta configuración, cuando un interruptor se activa, envía un mensaje al controlador Zigbee2MQTT. El controlador envía entonces un mensaje MQTT. El controlador MQTT decide entonces qué hacer basándose en ese mensaje. Podría, por ejemplo, decidir encender una luz en particular, por lo que enviaría un mensaje MQTT solicitando que la luz se encienda. Zigbee2MQTT recibiría ese mensaje, y luego enviaría un mensaje zigbee al punto final de la luz utilizando el clúster zigbee apropiado.

Este sistema es significativamente más flexible que el sistema Zigbee base. Pero también tiene una mayor latencia (el sistema tarda más en reaccionar a la activación de un interruptor) y tiene más puntos de fallo. Con la configuración del Zigbee base, ni siquiera el controlador interviene una vez que se ha completado la configuración. Con la configuración Zigbee2MQTT hay dos mensajes zigbee, dos mensajes MQTT, y tres pasos de procesamiento adicionales (el controlador principal decidiendo qué hacer, y MQTT procesando los mensajes en cada dirección). [27][28]

## 5. Metodologías a usar en el proyecto

Desde el inicio del proyecto se ha decidido que se va a trabajar con una metodología ágil debido a la naturaleza del proyecto, que aunque está planteado como un producto, a la hora de la verdad es un ejercicio educativo que requiere supervisión continuada por parte del tutor. Es por esta razón que se ha optado por una metodología similar a scrum en la que se han realizado reuniones bisemanales en las que entre otras cosas se ha discutido el avance de cada entrega y las dudas del alumno.

En el desarrollo de software, las metodologías ágiles (a veces escritas Agile) implican el descubrimiento de requisitos y el desarrollo de soluciones a través del esfuerzo de colaboración de equipos autoorganizados y multifuncionales y sus clientes usuarios finales. Aboga por la planificación adaptativa, el desarrollo evolutivo, la entrega temprana y la mejora continua, y fomenta las respuestas flexibles al cambio.

Se popularizó con el Manifiesto para el Desarrollo Ágil de Software de 2001. Los valores y principios que se recogen en este manifiesto se derivan de una amplia gama de marcos de desarrollo de software, como Scrum y Kanban, y los sustentan.

La mayoría de los métodos de desarrollo ágil dividen el trabajo de desarrollo de productos en pequeños incrementos que minimizan la cantidad de planificación y diseño por adelantado. Las iteraciones, o sprints, son plazos cortos (cajas de tiempo) que suelen durar de una a cuatro semanas. En cada iteración participa un equipo multifuncional que trabaja en todas las funciones: planificación, análisis, diseño, codificación, pruebas unitarias y pruebas de aceptación. Al final de la iteración se demuestra a los interesados un producto que funciona. Esto minimiza el riesgo general y permite que el producto se adapte rápidamente a los cambios. Puede que una iteración no añada suficiente funcionalidad como para justificar un lanzamiento al mercado, pero el objetivo es tener una versión disponible (con un mínimo de errores) al final de cada iteración. A través del desarrollo incremental, los productos tienen espacio para "fallar a menudo y pronto" a lo largo de cada fase iterativa en lugar de

hacerlo drásticamente en una fecha de lanzamiento final. El software que funciona es la principal medida de progreso.[29]

El principio de la co-localización es que los compañeros del mismo equipo deben estar situados juntos para establecer mejor la identidad como equipo y mejorar la comunicación. Esto permite la interacción cara a cara, idealmente frente a una pizarra, que reduce el tiempo de ciclo que se suele emplear cuando las preguntas y respuestas están mediadas por el teléfono, el chat persistente, la wiki o el correo electrónico.

Independientemente del método de desarrollo que se siga, todo equipo debe incluir un representante del cliente ("Product Owner" en Scrum). Esta persona es acordada por las partes interesadas para actuar en su nombre y se compromete personalmente a estar disponible para los desarrolladores para responder a las preguntas a lo largo de la iteración. Al final de cada iteración, las partes interesadas y el representante del cliente revisan el progreso y reevalúan las prioridades con el fin de optimizar el retorno de la inversión (ROI) y garantizar la alineación con las necesidades del cliente y los objetivos de la empresa. La importancia de la satisfacción de las partes interesadas, que se detalla mediante la interacción y la revisión frecuentes al final de cada fase, es la razón por la que la metodología suele denominarse "metodología centrada en el cliente".

Una característica común en el desarrollo ágil de software es el stand-up diario (un scrum diario en el marco de Scrum). En una breve sesión, los miembros del equipo se informan mutuamente de lo que han hecho el día anterior para alcanzar el objetivo de la iteración de su equipo, lo que pretenden hacer hoy para alcanzar el objetivo y cualquier obstáculo o impedimento que puedan ver para alcanzarlo.

Scrum es un marco de trabajo que utiliza una mentalidad ágil para desarrollar, entregar y mantener productos complejos, con un énfasis inicial en el desarrollo de software, aunque se ha utilizado en otros campos como la investigación, las ventas, el marketing y las tecnologías avanzadas. Está diseñado para equipos de diez o menos miembros, que dividen su trabajo en objetivos que pueden ser completados dentro de iteraciones de tiempo, llamadas sprints, de no más de un mes y más comúnmente dos semanas. El equipo scrum evalúa el progreso en reuniones diarias de 15 minutos o menos, llamadas scrums diarios. Al final del sprint, el equipo celebra otras dos reuniones: la revisión del sprint, en la que se muestra el trabajo realizado a las partes interesadas para obtener comentarios, y la retrospectiva del sprint, que permite al equipo reflexionar y mejorar.[30]

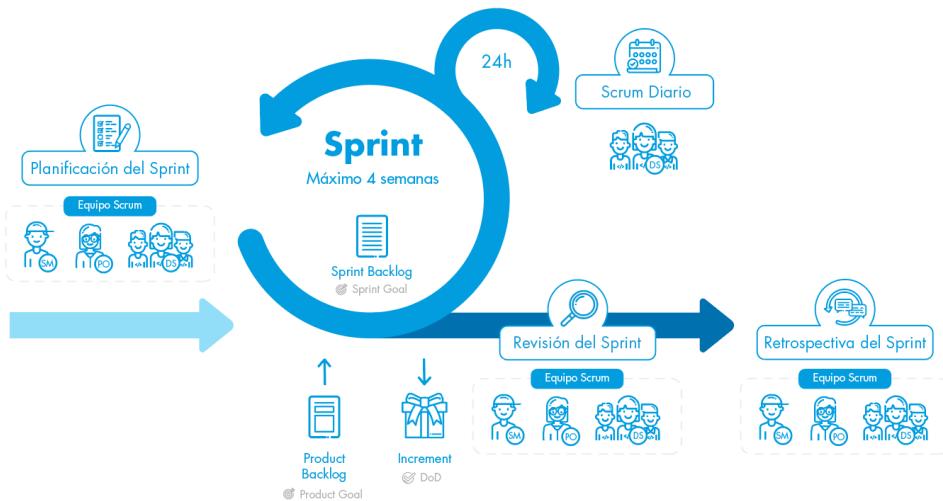


Figura 19: Diagrama explicativo de scrum

## 6. Plan de entregas

### 6.1 Historias de usuario

Tras analizar el problema se ha obtenido la siguiente lista de historias de usuario. Si bien este listado puede cambiar durante el desarrollo, es una buena indicación de todos los puntos que el proyecto debería cubrir

|       |  |
|-------|--|
| HU.1  | Como usuario de la aplicación quiero poder iniciar sesión como usuario cuidador o como usuario dependiente                     |
| HU.2  | Como usuario cuidador quiero poder definir un hogar que incluya una persona de la que cuidar y una serie de sensores.          |
| HU.3  | Como usuario cuidador quiero poder recibir notificaciones de las actividades llevadas a cabo por la persona que cuido.         |
| HU.4  | Como usuario cuidador quiero saber la localización aproximada en la casa de la persona que cuido.                              |
| HU.5  | Como usuario cuidador quiero poder ver un listado de los sensores disponibles así como su localización aproximada en el hogar. |
| HU.6  | Como usuario cuidador quiero poder añadir y configurar sensores a la App   |
| HU.7  | Como usuario cuidador quiero poder definir una nueva actividad   |
| HU.8  | Como usuario cuidador quiero poder asignar sensores a nuevas actividades.  |
| HU.9  | Como usuario cuidador quiero poder eliminar una actividad  |
| HU.10 | Como usuario cuidador quiero poder desactivar los electrodomésticos que puedan ser peligrosos.                                 |
| HU.11 | Como usuario dependiente quiero poder recibir notificaciones cada vez que se me olvide una actividad.                          |

|       |  |
|-------|--|
| HU.12 | Como usuario dependiente quiero que se notifique al usuario cuidador cada vez que sufra una caída.             |
| HU.13 | Como usuario cuidador quiero poder cuidar de más de una persona  |
| HU.14 | Como usuario cuidador quiero poder seleccionar el hogar que voy a visualizar en la App                         |
| HU.15 | Como usuario dependiente quiero que las notificaciones relacionadas conmigo puedan llegar a más de un cuidador |
| HU.16 | Como usuario dependiente quiero poder visualizar los detalles de una tarea                                     |
| HU.17 | Como persona dependiente quiero poder notificar que he realizado una actividad mediante una tarjeta NFC        |

Tabla 5: Historias de usuario

## 6.2 Plan de entregas

| Entrega   | Objetivo  |           |                 |   |                                    |   |  |   |   |
|-----------|---|-----------|-----------------|---|------------------------------------|---|--|---|---|
| <b>1</b>  | Instalación y prueba del entorno domótico   |           |                 |   |                                    |   |  |   |   |
|           | <table border="1"> <thead> <tr> <th>Iteración</th><th>Objetivo/Tarea.</th></tr> </thead> <tbody> <tr> <td>1</td><td>Instalación de Home assistant</td></tr> <tr> <td>2</td><td>Añadir dispositivos mediante zigbee2mqtt</td></tr> </tbody> </table>   | Iteración | Objetivo/Tarea. | 1 | Instalación de Home assistant      | 2 | Añadir dispositivos mediante zigbee2mqtt |   |   |
| Iteración | Objetivo/Tarea.   |           |                 |   |                                    |   |  |   |   |
| 1         | Instalación de Home assistant   |           |                 |   |                                    |   |  |   |   |
| 2         | Añadir dispositivos mediante zigbee2mqtt  |           |                 |   |                                    |   |  |   |   |
| <b>2</b>  | Creación de un prototipo inicial de un sistema de monitorización  |           |                 |   |                                    |   |  |   |   |
|           | <table border="1"> <thead> <tr> <th>Iteración</th><th>Objetivo/Tarea.</th></tr> </thead> <tbody> <tr> <td>1</td><td>Configuración del entorno domótico</td></tr> <tr> <td>2</td><td>Diseño de la aplicación de control</td></tr> <tr> <td>3</td><td>Implementación y pruebas de la aplicación</td></tr> </tbody> </table> | Iteración | Objetivo/Tarea. | 1 | Configuración del entorno domótico | 2 | Diseño de la aplicación de control       | 3 | Implementación y pruebas de la aplicación |
| Iteración | Objetivo/Tarea.   |           |                 |   |                                    |   |  |   |   |
| 1         | Configuración del entorno domótico  |           |                 |   |                                    |   |  |   |   |
| 2         | Diseño de la aplicación de control  |           |                 |   |                                    |   |  |   |   |
| 3         | Implementación y pruebas de la aplicación   |           |                 |   |                                    |   |  |   |   |
| <b>3</b>  | Ampliación del prototipo con un sistema de configuración de actividades   |           |                 |   |                                    |   |  |   |   |
|           | <table border="1"> <thead> <tr> <th>Iteración</th><th>Objetivo/Tarea.</th></tr> </thead> </table>   | Iteración | Objetivo/Tarea. |   |                                    |   |  |   |   |
| Iteración | Objetivo/Tarea.   |           |                 |   |                                    |   |  |   |   |

|          |                                       |  |  |
|----------|---------------------------------------|--|--|
|          |                                       | <b>1</b>   | Diseño e implementación de la base de datos        |
|          |                                       | <b>2</b>   | Incluir un sistema de configuración de actividades |
|          |                                       | <b>3</b>   | Incluir un sistema de notificación de actividades  |
| <b>4</b> | Aplicación para personas dependientes |  |  |
|          | <b>Iteración</b>                      | <b>Objetivo/Tarea.</b>                                   |  |
|          | <b>1</b>                              | Recepción de notificaciones                              |  |
|          | <b>2</b>                              | Detección de caídas                                      |  |
| <b>5</b> | Pruebas de evaluación de usabilidad   |  |  |
|          | <b>Iteración</b>                      | <b>Objetivo/Tarea.</b>                                   |  |
|          | <b>1</b>                              | Creación de un cuestionario de usabilidad                |  |
|          | <b>2</b>                              | Prueba de la aplicación con usuarios reales              |  |
| <b>6</b> | Despliegue del sistema                |  |  |
|          | <b>Iteración</b>                      | <b>Objetivo/Tarea.</b>                                   |  |
|          | <b>1</b>                              | Descripción detallada de como poner en marcha el sistema |  |

Tabla 6: Plan de entregas

### 6.3 Presupuesto

Los componentes hardware para desarrollar este proyecto han sido los que se listan a continuación. Los costes de la electricidad y la conexión a internet no han sido incluidos. Es teóricamente posible cambiar la Raspberry Pi 3 por otro sistema capaz de ejecutar Home Assistant, pero debido a la popularidad y bajo precio de la Raspberry Pi 3 nos hemos decantado por ella.

| <b>Componente</b>                              | <b>Precio</b>  | <b>Enlace de compra</b>   |
|--|----------------|---|
| Kit Raspberry Pi 3                             | 69,95€         | <a href="https://www.amazon.es/Raspberry-Pi-Official-Desktop-Starter/dp/B01CI58722">https://www.amazon.es/Raspberry-Pi-Official-Desktop-Starter/dp/B01CI58722</a>                                 |
| Xiaomi Mi Smart Sensor Set                     | 72€            | <a href="https://www.amazon.es/Xiaomi-Smart-Sensor-Set-Interruptor/dp/B07KXN6S74">https://www.amazon.es/Xiaomi-Smart-Sensor-Set-Interruptor/dp/B07KXN6S74</a>                                     |
| Ikea TRÅDFRI Mando a distancia                 | 15€            | <a href="https://www.ikea.com/es/es/p/tradfri-mando-distancia-30443124/">https://www.ikea.com/es/es/p/tradfri-mando-distancia-30443124/</a>   |
| Ikea TRÅDFRI Wireless motion sensor            | 10€            | <a href="https://www.ikea.com/es/en/p/tradfri-wireless-motion-sensor-white-70429913/">https://www.ikea.com/es/en/p/tradfri-wireless-motion-sensor-white-70429913/</a>                             |
| Botón de atajo                                 | 6€             | <a href="https://www.ikea.com/es/es/p/tradfri-boton-atajo-blanco-40356381/">https://www.ikea.com/es/es/p/tradfri-boton-atajo-blanco-40356381/</a>   |
| Ikea TRÅDFRI Regulador iluminación inalámbrico | 6€             | <a href="https://www.ikea.com/es/es/p/tradfri-regulador-iluminacion-inalambrico-blanco-70408595/">https://www.ikea.com/es/es/p/tradfri-regulador-iluminacion-inalambrico-blanco-70408595/</a>     |
| Ikea TRÅDFRI Enchufe inteligente inalámbrico   | 10€            | <a href="https://www.ikea.com/es/es/p/tradfri-enchufe-inteligente-inalambrico-90356166/">https://www.ikea.com/es/es/p/tradfri-enchufe-inteligente-inalambrico-90356166/</a>                       |
| Antena Zigbee USB CC2531                       | 23,87€         | <a href="https://www.amazon.es/C2531-Zigbee2MQTT-Firmware-Assistente-dom%C3%A9stico/dp/B085MFRK9W/">https://www.amazon.es/C2531-Zigbee2MQTT-Firmware-Assistente-dom%C3%A9stico/dp/B085MFRK9W/</a> |
| <b>Total</b>                                   | <b>212,82€</b> |   |

Tabla 7: Presupuesto

## 7. Desarrollo, entregas e iteraciones

### 7.1 Primera entrega: Toma de contacto con los sensores y Home Assistant

El principal objetivo de esta entrega es familiarizarse con las tecnologías elegidas, así como realizar la configuración inicial de Home Assistant. No obstante también se ha decidido empezar a diseñar la arquitectura del sistema. En la figura 20 puede verse como queda esta primera revisión de la arquitectura.

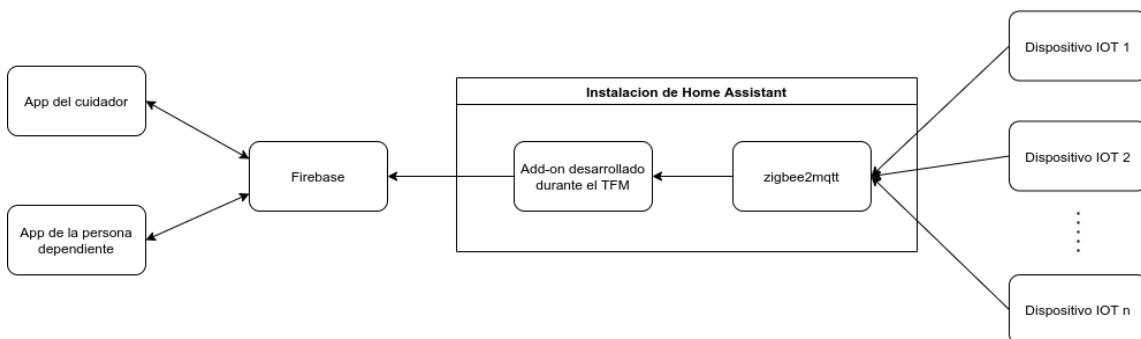


Figura 20: Diagrama explicativo de la arquitectura del sistema

### Instalación y prueba del entorno domótico

Para instalar home assistant son necesarios los siguientes materiales:

- Raspberry Pi 3
- Fuente de alimentación para la Raspberry Pi
- Tarjeta Micro SD.
- Lector de tarjetas Micro SD.
- Cable Ethernet. Home Assistant puede funcionar con Wi-Fi, pero una conexión Ethernet es más fiable y muy recomendable.
- Imagen de Home Assistant

Comenzamos descargando la imagen desde el enlace oficial y cuando la descarga haya terminado la descomprimimos:  
[https://github.com/home-assistant/operating-system/releases/download/6.1/haos\\_rpi3-6-4-6.1.img.xz](https://github.com/home-assistant/operating-system/releases/download/6.1/haos_rpi3-6-4-6.1.img.xz)

Desde la documentación oficial de Home Assistant recomiendan el uso de una interfaz gráfica llamada Balena Etcher [31] para flashear la imagen, pero las herramientas de Unix pueden realizar esta operación fácilmente. En este caso basta con usar lsblk para identificar cómo se llama la micro SD y dd para flashear la imagen sobre dicho disco

```

Archivo Editar Ver Marcadores Preferencias Ayuda
Nueva pestaña Separar vista izquierda/derecha Separar vista arriba/abajo Copiar Pegar Buscar
[arturo@arturo-ms7845 Escritorio]$ wget https://github.com/home-assistant/operating-system/releases/download/6.1/haos_rp13-64-6.1.img.xz
--2021-06-26 14:52:00-- https://github.com/home-assistant/operating-system/releases/download/6.1/haos_rp13-64-6.1.img.xz
Cargado certificado CA '/etc/ssl/certs/ca-certificates.crt'
Resolviendo github.com (github.com)... 140.82.121.3
Conectando con github.com [140.82.121.3], puerto 443... conectado.
Petición HTTP enviada, esperando respuesta... 200 OK
Localización: https://github-releases.githubusercontent.com/11592009/f2291d80-d28b-11eb-8b61-e931bb7e03487x-Amz-Algorithm=AWS4-HMAC-SHA256X-Amz-Credential=AKIAIWUJYAX4CSVEH53%2F202106262fus-east-1%2Fs3%2Faws4-requestx-Amz-Date=20210626T125029Zx-Amz-Expires=3008x-Amz-Signature=e6a0425642365865d7cc6ca04b4878730317f76a3c1f42eb254c45d879118x-Amz-SignedHeaders=host&actor_id=0&key_id=0&repo_id=11592009&response-content-disposition=attachment%3Bfilename%3Dhaos_rp13-64-6.1.img.xz&response-content-type=application%2Foctet-stream [siguiendo]
--2021-06-26 14:52:00-- https://github-releases.githubusercontent.com/11592009/f2291d80-d28b-11eb-8b61-e931bb7e03487x-Amz-Algorithm=AWS4-HMAC-SHA256X-Amz-Credential=AKIAIWUJYAX4CSVEH53%2F202106262fus-east-1%2Fs3%2Faws4-requestx-Amz-Date=20210626T125029Zx-Amz-Expires=3008x-Amz-Signature=e6a20425642365865d7cc6ca04b4878730317f76a3c1f42eb254c45d879118x-Amz-SignedHeaders=host&actor_id=0&key_id=0&repo_id=11592009&response-content-disposition=attachment%3Bfilename%3Dhaos_rp13-64-6.1.img.xz&response-content-type=application%2Foctet-stream
Resolviendo github-releases.githubusercontent.com (github-releases.githubusercontent.com)... 185.199.110.154, 185.199.108.154, 185.199.109.154, ...
Conectando con github-releases.githubusercontent.com (github-releases.githubusercontent.com)[185.199.110.154]:443... conectado.
Petición HTTP enviada, esperando respuesta... 200 OK
Longitud: 239839632 (229M) [application/octet-stream]
Grabando a: «haos_rp13-64-6.1.img.xz»
haos_rp13-64-6.1.img.xz 100%[=====] 228,73M 66,7MB/s en 4,7s
2021-06-26 14:52:05 (48,4 MB/s) - «haos_rp13-64-6.1.img.xz» guardado [239839632/239839632]

[arturo@arturo-ms7845 Escritorio]$ unxz haos_rp13-64-6.1.img.xz
[arturo@arturo-ms7845 Escritorio]$ lsblk
NAME MAJ:MIN RM SIZE RO TYPE MOUNTPOINTS
sda   8:0    0 931.5G  0 disk 
└─sda1 8:1    0 931.5G  0 part /
sdb   8:16   0 489G  0 disk 
└─sdb1 8:17   0 243.3G 0 part 
└─sdb2 8:18   0 1.5G  0 part 
└─sdb3 8:19   0 244.2G 0 part /
sdc   8:32   1 14.6G 0 disk 
└─sdc1 8:33   1 32M  0 part 
└─sdc2 8:34   1 24M  0 part 
└─sdc3 8:35   1 256M 0 part 
└─sdc4 8:36   1 24M  0 part 
└─sdc5 8:37   1 256M 0 part 
└─sdc6 8:38   1 6M  0 part 
└─sdc7 8:39   1 96M 0 part 
└─sdc8 8:40   1 1G  0 part 
sr0   11:0   1 1024M 0 rom 
zram0 254:0   0 3.1G 0 disk [SWAP]
[arturo@arturo-ms7845 Escritorio]$ sudo dd if=/dev/sdc status=progress
[sudo] password for arturo:
2138124800 bytes (2,1 GB, 2,0 GiB) copied, 483 s, 4,4 MB/s
4194304+0 registros leidos
4194304+0 registros escritos
2147483648 bytes (2,1 GB, 2,0 GiB) copied, 564,635 s, 3,8 MB/s
[arturo@arturo-ms7845 Escritorio]$ 

```

Figura 21: Instalación de Home Assistant

Una vez ha terminado el proceso, insertamos la micro SD en la Raspberry Pi, la conectamos al router con el cable ethernet y la enchufamos a la corriente

Home Assistant tardará unos minutos en estar listo, cuando lo esté se podrá acceder a [homeassistant.local:8123](http://homeassistant.local:8123) desde el navegador web. En esta dirección se encuentra la interfaz de configuración del primer inicio.

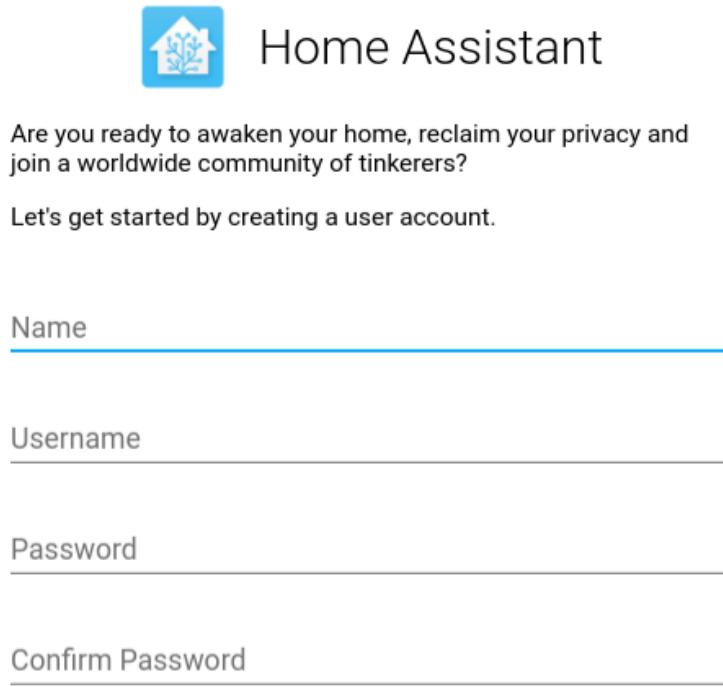


Figura 21: Primer inicio de Home Assistant

Tras llenar el formulario para crear el primer usuario, home assistant nos mostrará mapa para seleccionar nuestra localización y unidades de medida

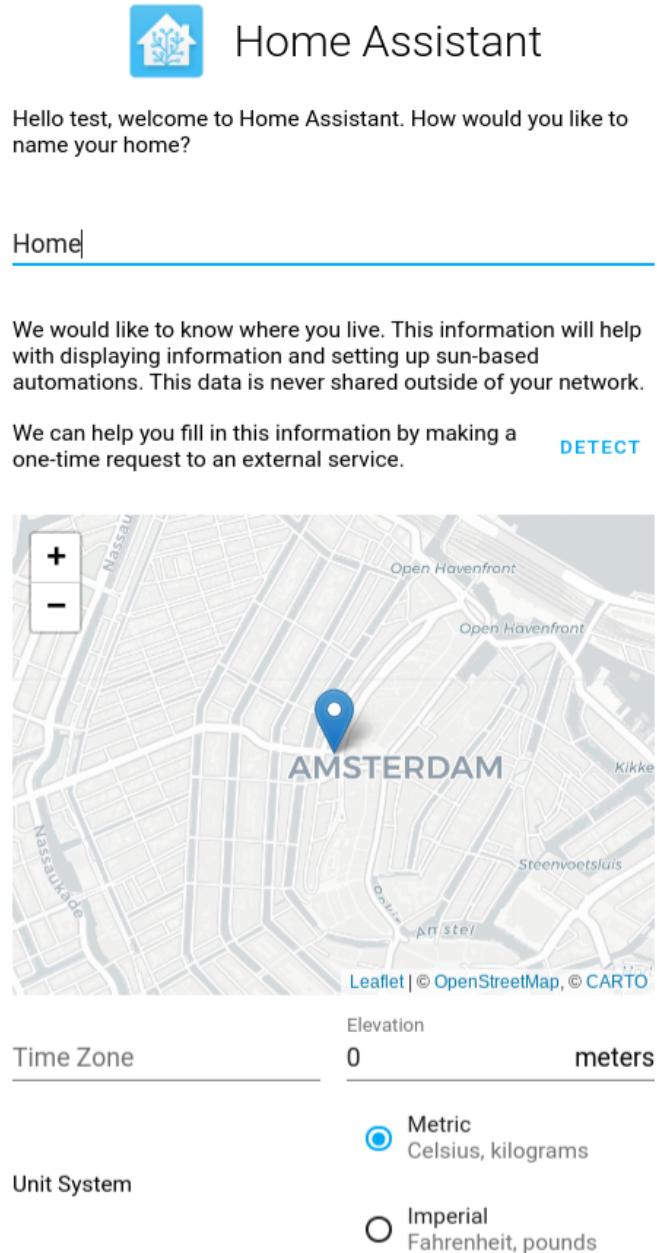


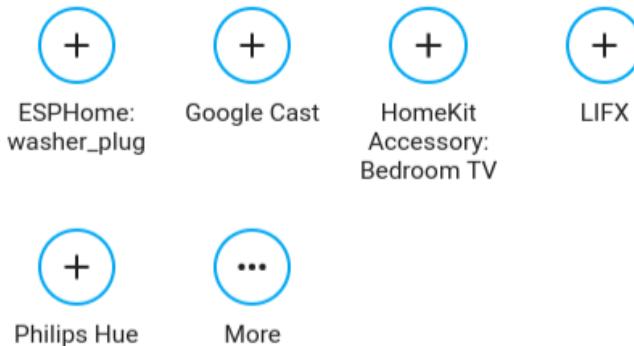
Figura 22: Primer inicio de Home Assistant

En la siguiente pantalla Home Assistant nos muestra los dispositivos compatibles que ha encontrado en la red local. No obstante no nos interesa seleccionar ninguno, pues primero tenemos que instalar un plugin



# Home Assistant

Devices and services are represented in Home Assistant as integrations. You can set them up now, or do it later from the configuration screen.



[FINISH](#)

Figura 23: Primer inicio de Home Assistant

Una vez finalizada la configuración veremos el dashboard de Home Assistant

The screenshot shows the Home Assistant dashboard. On the left is a sidebar with the following items:

- Overview (selected)
- Map
- Logbook
- History
- Media Browser
- Developer Tools
- Supervisor
- Configuration
- Notifications (with 1 notification)
- Awesome User

The main area is titled "Home". It displays a weather forecast for "Home":

**Partly cloudy**  
17.6 °C  
Precipitation 0.7 mm

| Day | Temp (°C) | Temp (°F) |
|-----|-----------|-----------|
| Wed | 18.4°     | 11.8°     |
| Thu | 16.3°     | 9.5°      |
| Fri | 13.5°     | 7.9°      |
| Sat | 12.9°     | 10.5°     |
| Sun | 13.1°     | 10°       |

Below this is another weather section:

**Partly cloudy**  
17.6 °C  
17.7 °C / 13.7 °C

| Hour | Temp (°C) | Temp (°F) |
|------|-----------|-----------|
| 3 PM | 17.7°     | 17.6°     |
| 4 PM |           |           |
| 5 PM | 17.3°     | 17.3°     |
| 6 PM | 16.8°     | 16.8°     |
| 7 PM | 15.9°     | 15.9°     |

At the top right, there are three circular icons: "Updater" (blue), "Awesome User" (red), and "Sun" (orange).

Figura 24: Pantalla principal de Home Assistant

Ahora vamos a instalar el Add-on Zigbee2mqtt, pero para ello primero tenemos que instalar un broker MQTT.

En el menú lateral seleccionamos Supervisor, luego Add-on store, buscamos Mosquito broker y lo instalamos.

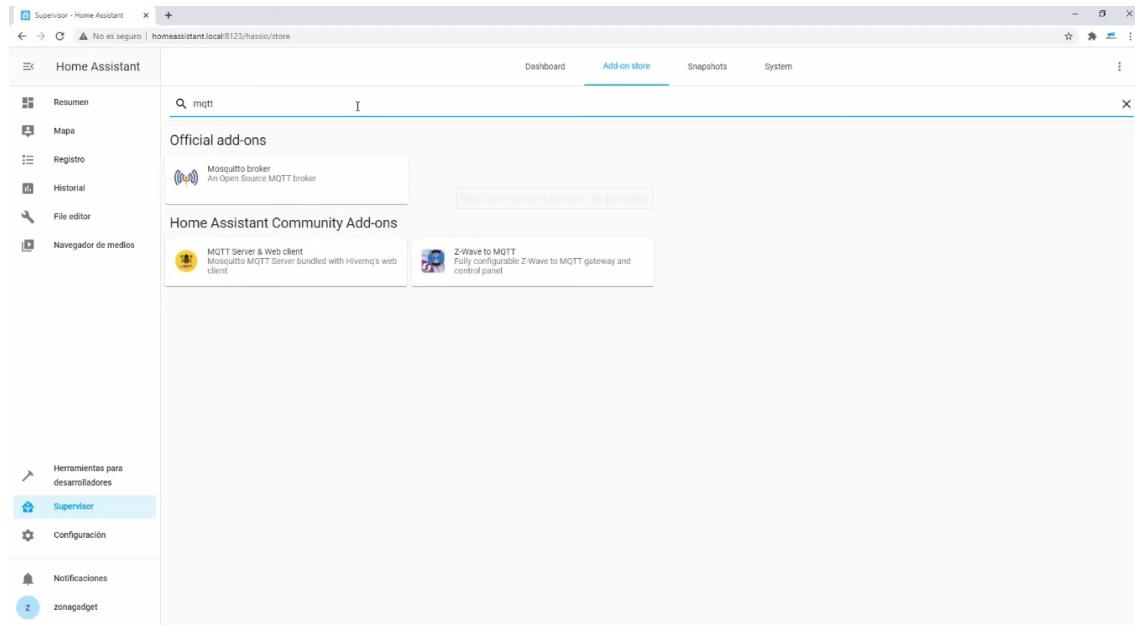


Figura 25: Búsqueda en la Add-on store

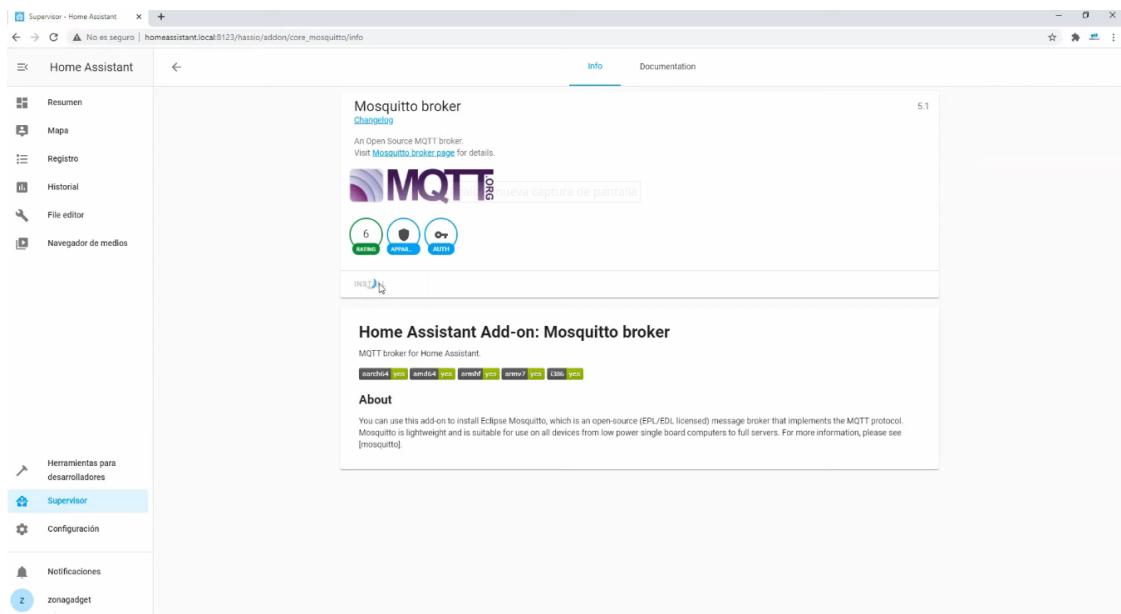


Figura 26: Instalación del add-on

Una vez tengamos instalado Mosquito broker debemos crear un nuevo usuario, para ello vamos a Configuración -> Usuarios, le damos al botón flotante con un “más” en la esquina inferior derecha y nos aparecerá un formulario para introducir los datos del nuevo usuario. Estos datos son los que luego se van a usar para interactuar con el servicio MQTT

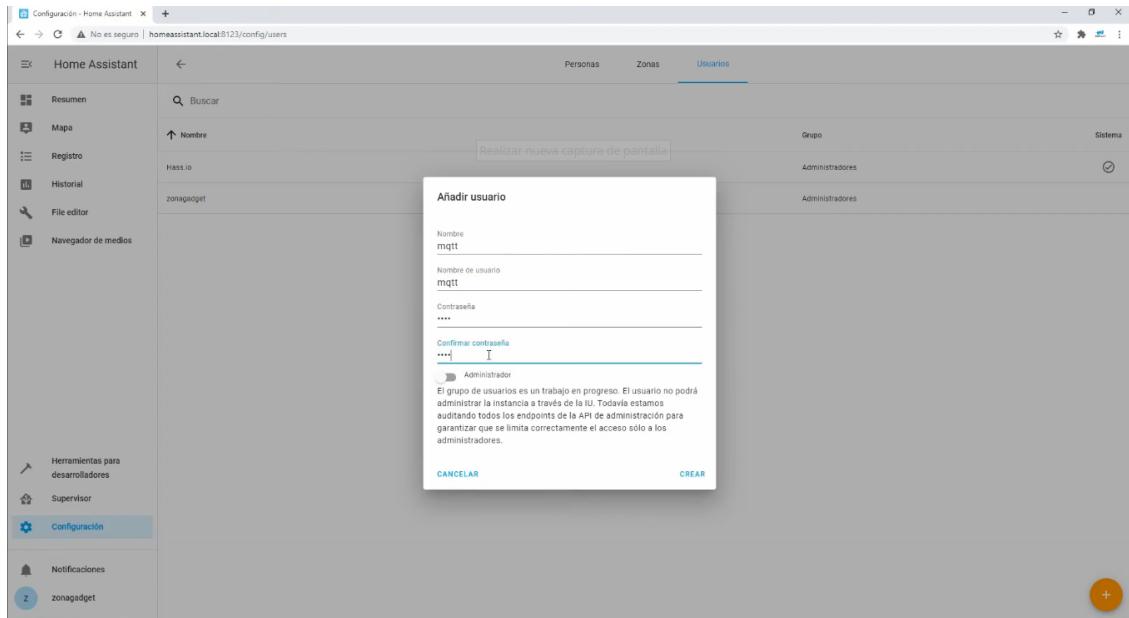


Figura 27: Configuración de mosquito-mqtt

Tras haber creado el usuario volvemos al supervisor, seleccionamos Mosquito Broker y le damos a start. Cuando el Add-on haya terminado de arrancar nos vamos al menú de configuración, seleccionamos integraciones y veremos que MQTT aparece entre ellas. Le damos al botón de configurar, seleccionamos habilitar descubrimiento y le damos a enviar.

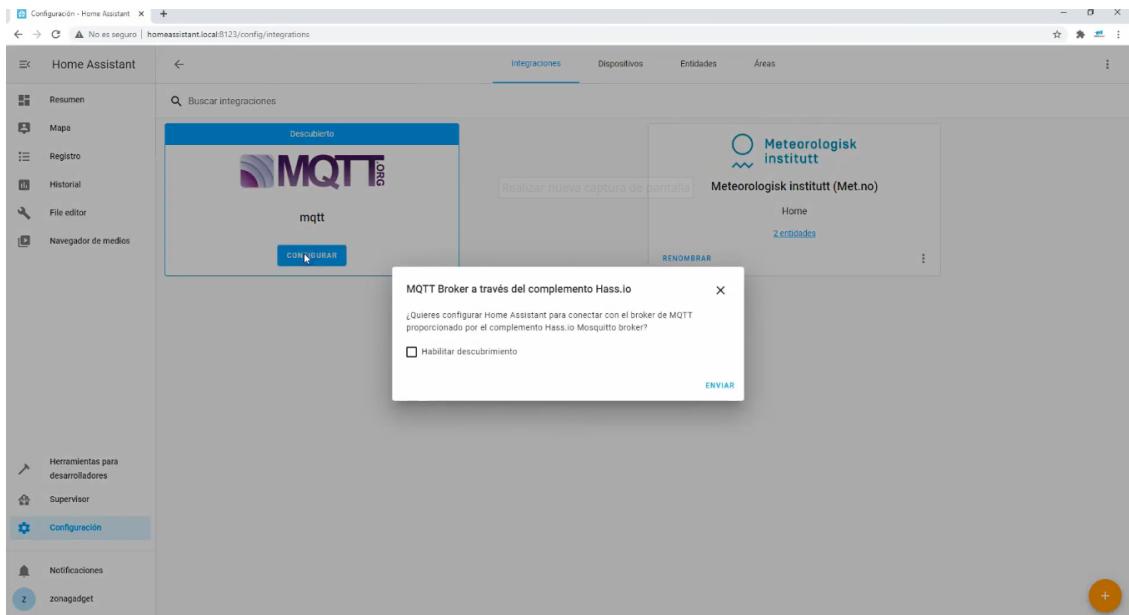


Figura 28: Configuración de mosquito-mqtt

Con esto habremos configurado el servicio MQTT en Home Assistant. Ahora ya podemos instalar Zigbee2mqtt. Vamos a la zona de Addons y pulsamos en el menú superior izquierdo (los tres puntos). Pegamos el enlace del repositorio y le damos a añadir:

<https://github.com/zigbee2mqtt/hassio-zigbee2mqtt>

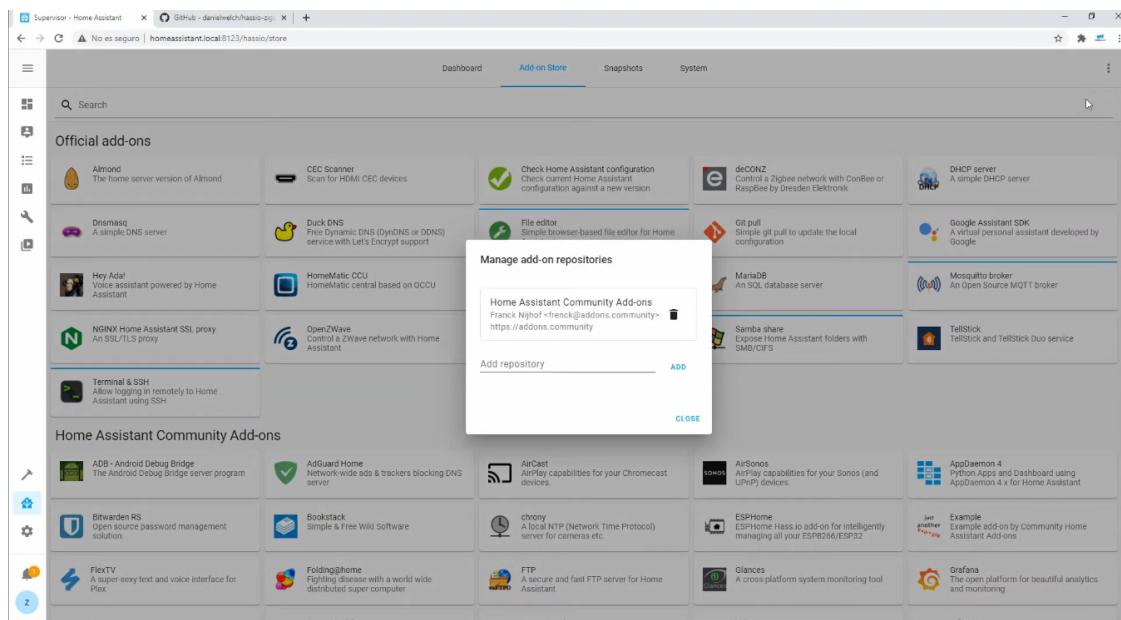


Figura 29: Añadiendo el repositorio de Zigbee2mqtt

Al haber añadido el repositorio ahora aparece el Add-on en la lista de Add-ons disponibles

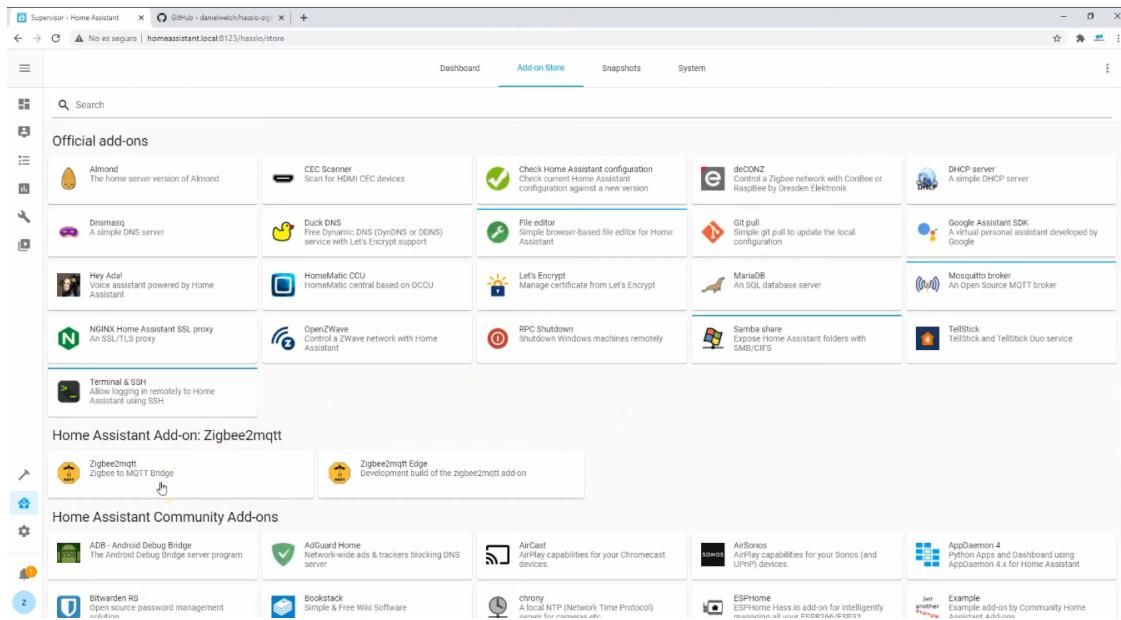


Figura 30: Añadiendo el repositorio de Zigbee2mqtt

Seleccionamos Zigbee2mqtt y lo instalamos. Una vez esté instalado tenemos que activar la opción de show in sidebar para poder acceder con más facilidad a los menús del Add-on.

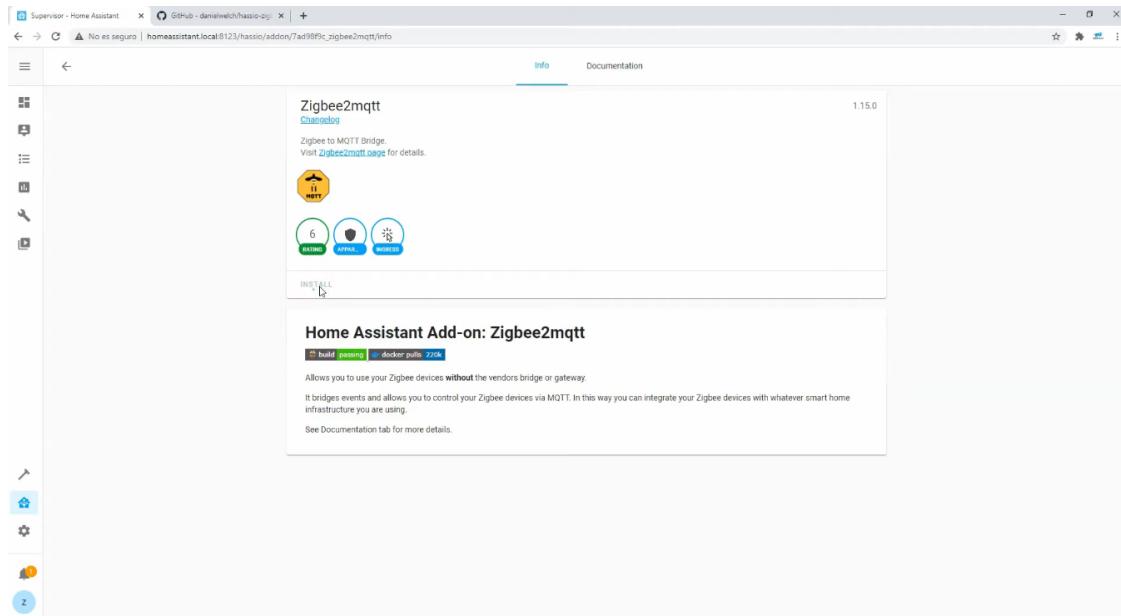


Figura 30: Instalación de Zigbee2mqtt

Cuando se haya instalado zigbee2mqtt nos dirigimos a la pestaña de configuración, tenemos que cambiar el usuario y la contraseña (línea 10 y 11 respectivamente) y poner las credenciales del usuario que hemos creado durante la instalación de MQTT. Además es conveniente cambiar la network\_key ya que por defecto siempre es la misma. Si no la cambiamos nos exponemos a ataques de tipo spoofing. Para finalizar guardamos la nueva configuración. [32]

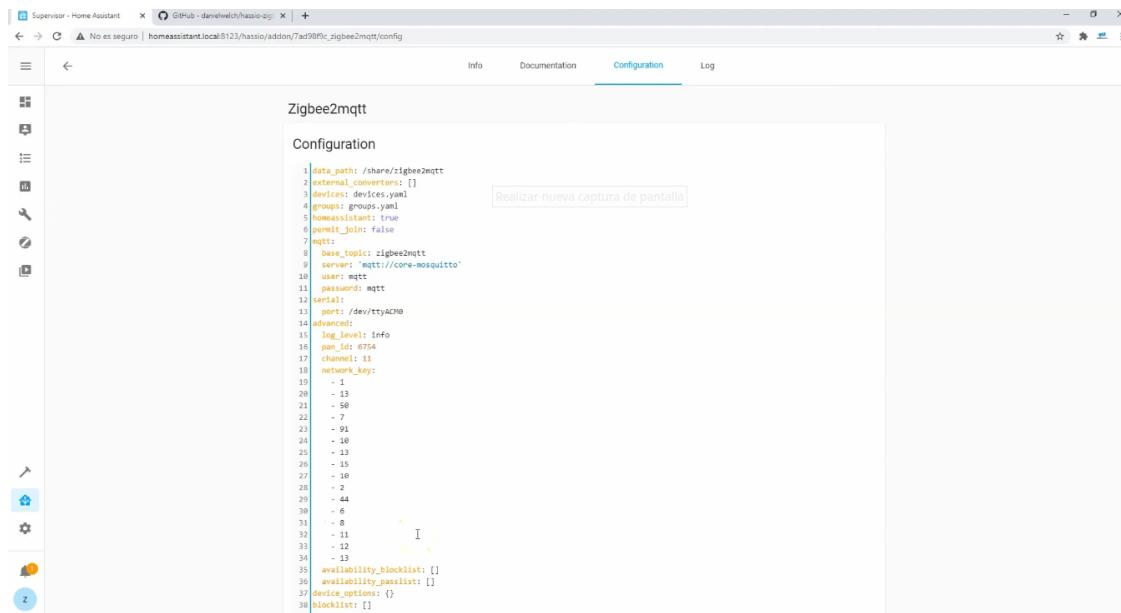


Figura 31: Configuración de Zigbee2mqtt

## Comunicarse con los sistemas mediante MQTT

Una vez que hemos terminado de instalar y configurar Zigbee2mqtt, podemos pasar a sincronizar los distintos dispositivos con Home Assistant.

Vamos al menú de Zigbee2mqtt (ahora podemos acceder rápidamente desde el menú lateral izquierdo). En este menú es donde aparecerán los dispositivos conectados directamente por zigbee a Home Assistant. Para añadir un nuevo dispositivo le damos al botón de arriba a la derecha que pone Permit join(all). Al pulsarlo se inicia una cuenta atrás durante la cual podremos añadir los dispositivos. Si queremos añadir uno basta con pulsar el escondido botón de sincronización que todos los dispositivos tienen. En el caso de los de Xiaomi hace falta un alfiler para pulsar dicho botón, en los de IKEA hay que desatornillar la tapadera de las pilas para poder acceder al botón en cuestión.

Cuando se añade un dispositivo nuevo es conveniente cambiarle el friendly name ya que por defecto es el mismo valor que IEEE Address.

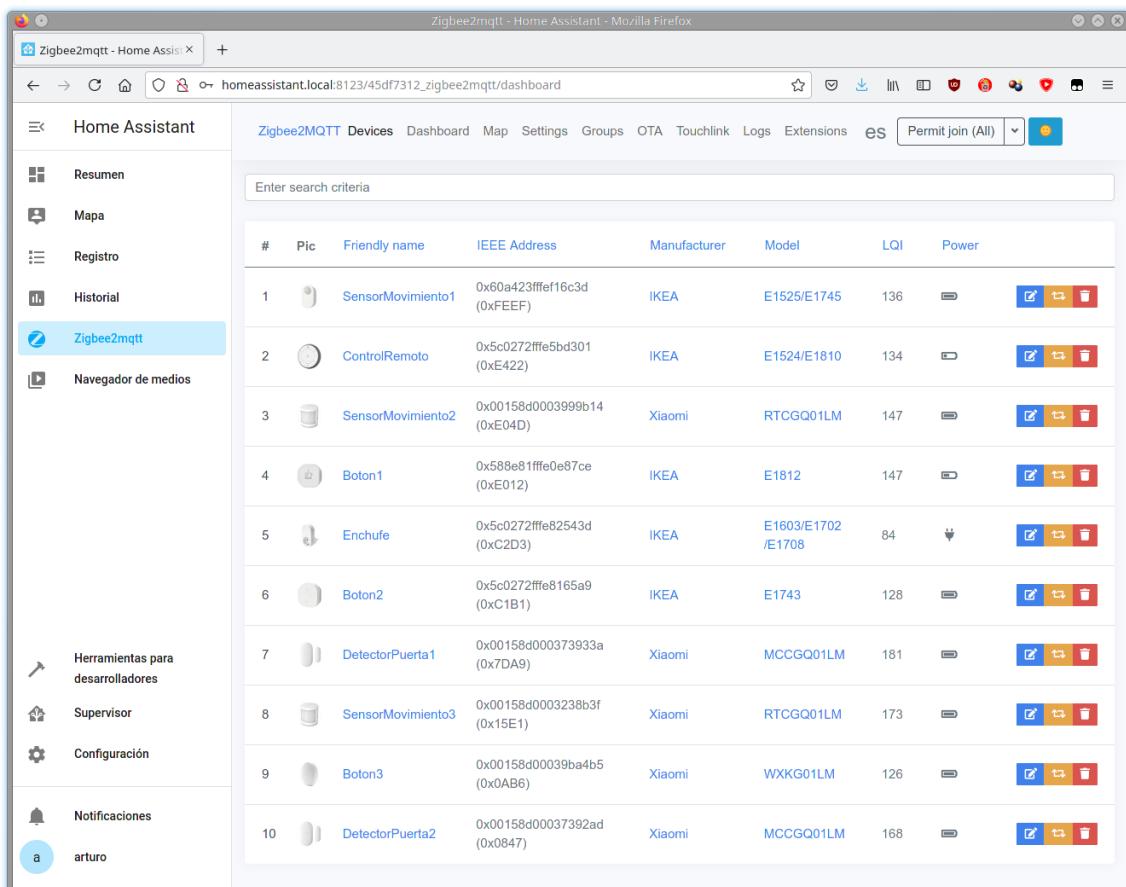


Figura 32: Pantalla principal de Zigbee2mqtt

Para comunicarse con los dispositivos es necesario una librería MQTT, he elegido la librería paho-mqtt. Paho es un proyecto de la fundación eclipse que está desarrollando bibliotecas de código abierto para MQTT; hay disponibles bibliotecas MQTT C y Java con Lua, Python, C++ y JavaScript en varias etapas de desarrollo. Esto me ha permitido realizar pruebas rápidamente con la librería de Python y luego pasar a la versión de Java para la aplicación Android.

El script realizado para las pruebas ha sido el siguiente:

```
1  #!/usr/bin/env python
2
3  import paho.mqtt.client as mqtt
4  import json
5
6  names = {}
7
8  def on_connect(client, userdata, flags, rc):
9      print("Connected with result code "+str(rc))
10     client.subscribe("zigbee2mqtt/bridge/devices/#")
11
12 def on_message(client, userdata, msg):
13     if msg.topic == "zigbee2mqtt/bridge/devices":
14         devices = json.loads(msg.payload.decode('unicode-escape'))
15         for d in devices:
16             if d["definition"] != None:
17
18                 client.subscribe("zigbee2mqtt/"+d["friendly_name"])
19                 print(d["definition"]["description"], " ", 
d["friendly_name"])
20                 names[d["friendly_name"]] =
d["definition"]["description"]
21             else:
22                 friendly_name = msg.topic.split("/")[1]
23                 print(names[friendly_name], " ", friendly_name, " ",
json.loads(msg.payload.decode('unicode-escape'))))
24
25
26
27 client = mqtt.Client()
28 client.on_connect = on_connect
29 client.on_message = on_message
30 client.username_pw_set(username="mqtt", password="mqtt")
31 client.connect("192.168.1.129", 1883, 60)
32 client.loop_forever()
```

Este script se conecta al broker mqtt que hemos instalado en Home Assistant. Nada mas conectarse se suscribe a zigbee2mqtt/bridge/devices/#, este topic devuelve un mensaje con todos los dispositivos disponibles en la red. Cuando dicha información es recibida es procesada y utilizada para suscribirse al topic de cada uno de los dispositivos. A partir de ahí simplemente muestra los mensajes recibido de cada dispositivo

```

[arturo@arturo-ms7845 TFM]$ ./mqtt.py
Connected with result code 0
TRADFRI shortcut button Boton1
Mijia wireless switch Boton3
TRADFRI ON/OFF switch Boton2
TRADFRI motion sensor SensorMovimiento1
TRADFRI remote control ControlRemoto
TRADFRI control outlet Enchufe
Mijia human body movement sensor SensorMovimiento3
Mijia door & window contact sensor DetectorPuerta2
Mijia door & window contact sensor DetectorPuerta1
Mijia human body movement sensor SensorMovimiento2
TRADFRI shortcut button Boton1 {'action': 'on', 'battery': 74, 'linkquality': 99, 'update': {'state': 'idle'}, 'update_available': False}
TRADFRI shortcut button Boton1 {'action': '', 'battery': 74, 'linkquality': 99, 'update': {'state': 'idle'}, 'update_available': False}
Mijia door & window contact sensor DetectorPuerta2 {'battery': 100, 'contact': False, 'linkquality': 99, 'voltage': 3025}
Mijia door & window contact sensor DetectorPuerta2 {'battery': 100, 'contact': True, 'linkquality': 99, 'voltage': 3025}
Mijia door & window contact sensor DetectorPuerta2 {'battery': 100, 'contact': False, 'linkquality': 102, 'voltage': 3025}
Mijia door & window contact sensor DetectorPuerta2 {'battery': 100, 'contact': True, 'linkquality': 105, 'voltage': 3025}
Mijia wireless switch Boton3 {'action': 'single', 'battery': 100, 'linkquality': 162, 'voltage': 3072}
Mijia wireless switch Boton3 {'action': None, 'battery': 100, 'click': 'single', 'linkquality': 162, 'voltage': 3072}
Mijia wireless switch Boton3 {'action': '', 'battery': 100, 'linkquality': 162, 'voltage': 3072}
Mijia wireless switch Boton3 {'action': None, 'battery': 100, 'click': '', 'linkquality': 162, 'voltage': 3072}
Mijia wireless switch Boton3 {'action': None, 'battery': 100, 'linkquality': 162, 'voltage': 3062}

```

Figura 33: Script en funcionamiento

Añadir los dispositivos al sistema domótico resulta bastante sencillo. No obstante, estos tienden a desconectarse pasado un tiempo, causando errores en Home Assistant. La solución a este problema pasa por forzar la parada del plugin Zigbee2mqtt, borrar los dispositivos conectados y añadirlos de nuevo.

Por otro lado la comunicación con los dispositivos, aunque inicialmente es sencilla, produce JSONs arbitrarios, lo cual dificulta notablemente la escalabilidad de este proyecto, ya que habría que añadir soporte manualmente para cada dispositivo que se quiera usar.

Como hemos podido ver, la puesta en marcha de home assistant es un proceso bastante sencillo, no obstante la instalación y configuración de los complementos requeridos, así como el enlace con los dispositivos IOT puede resultar más tediosa.

## 7.2 Segunda entrega: Creación de un prototipo inicial de un sistema de monitorización

En esta entrega comienza el desarrollo del sistema de monitorización, compuesto por dos apps de android y un complemento para Home Assistant

### Configuración del entorno domótico

Para poner en marcha el entorno domótico previamente es necesario distribuir adecuadamente los sensores por el hogar.

- Sensores de movimiento: Uno en cada habitación para que el sistema pueda monitorizar la posición aproximada de la persona
- Botones: Uno cerca de cada localización donde se quiera controlar una actividad complicada de monitorizar. Por ejemplo un botón en la cocina para avisar de que se ha empezado a cocinar.

- Sensores de apertura y cierre de puertas: Uno en cada puerta para ayudar a los sensores de movimiento en la tarea de monitorizar la posición aproximada de la persona mayor. Además, dado su pequeño tamaño también se pueden colocar en cualquier otro tipo de aparato que se abra y cierre, por ejemplo un pastillero. Esto ayudaría bastante a monitorizar ciertos tipos de actividades que no son sencillas de controlar.

## Diseño de la aplicación de control

Los wireflows son un formato de especificación del diseño que combina diseños de páginas al estilo de los wireframes con una forma simplificada de representar las interacciones de una interfaz.

A continuación se muestra un diagrama de navegación de las distintas pantallas de la aplicación:

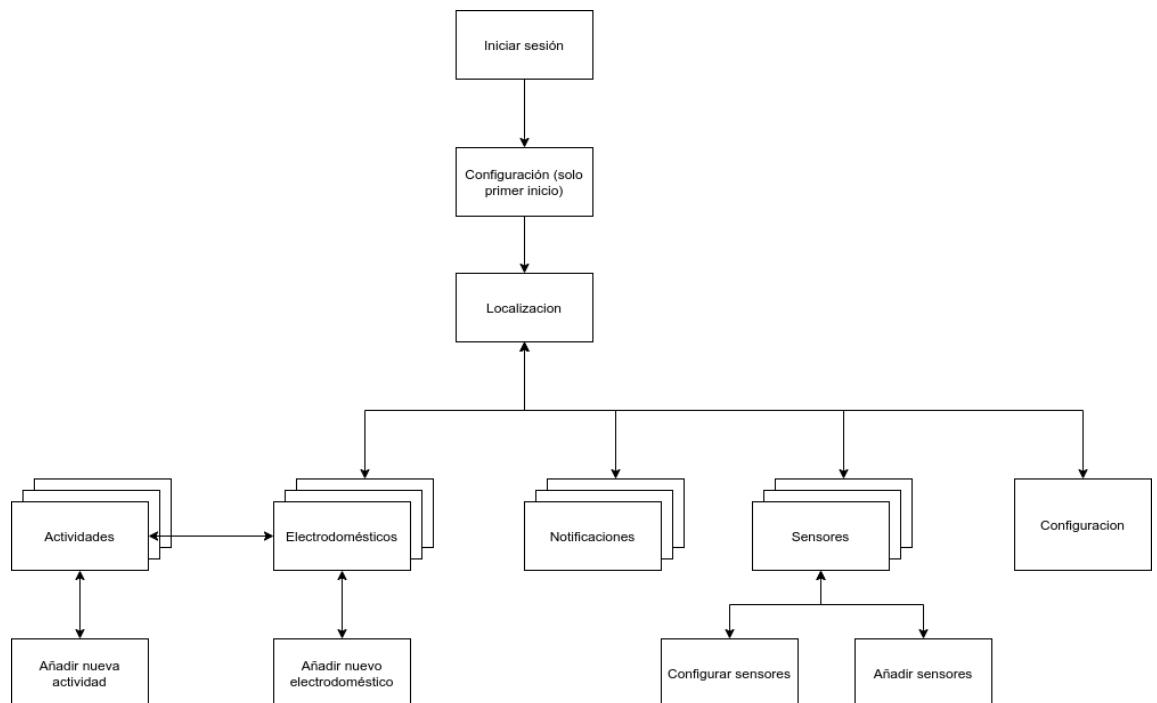


Figura 33: Wireflow de la aplicación del cuidador

En el caso de la aplicación para la persona mayor solo permitirá visualizar las notificaciones relacionadas con las actividades a realizar, además utilizará el giroscopio para detectar posibles caídas.

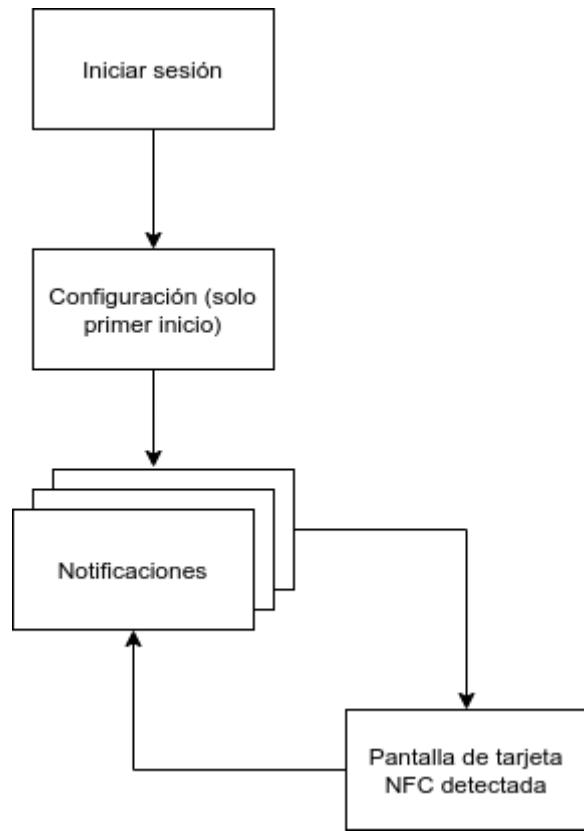
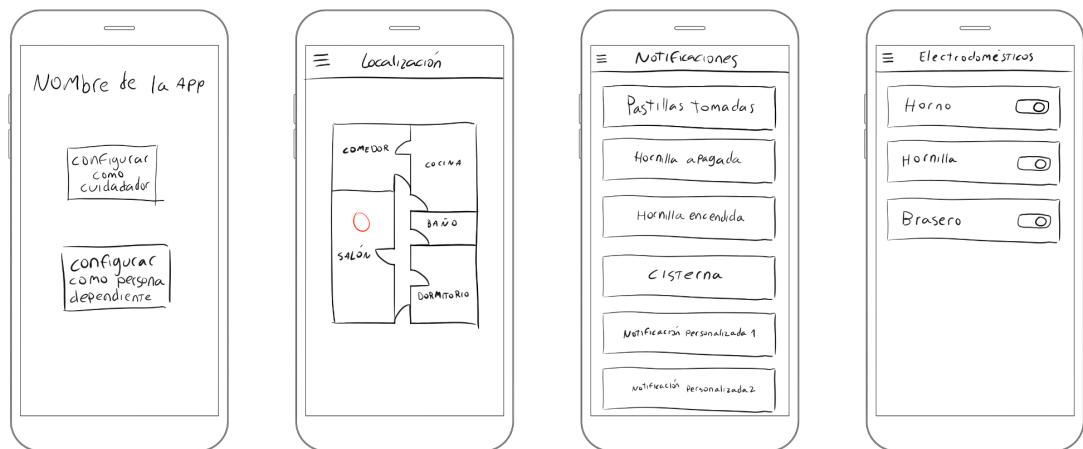


Figura 34: Wireflow de la aplicación de la persona dependiente

### Crear bocetos de cada actividad (escena)

Los bocetos realizados muestran de forma esquemática cómo van a ser las distintas actividades de la aplicación. Se trata de bocetos iniciales, por lo que pueden diferir ligeramente de la interfaz en la aplicación final.



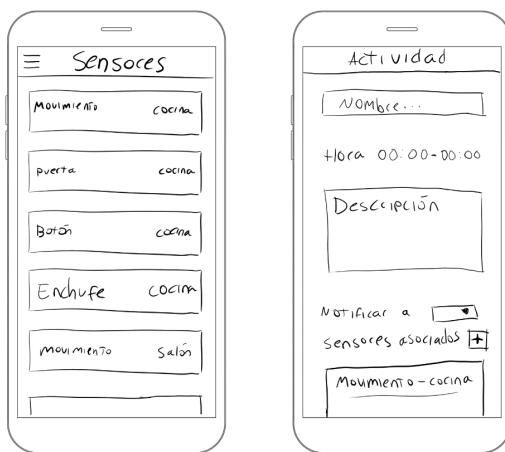
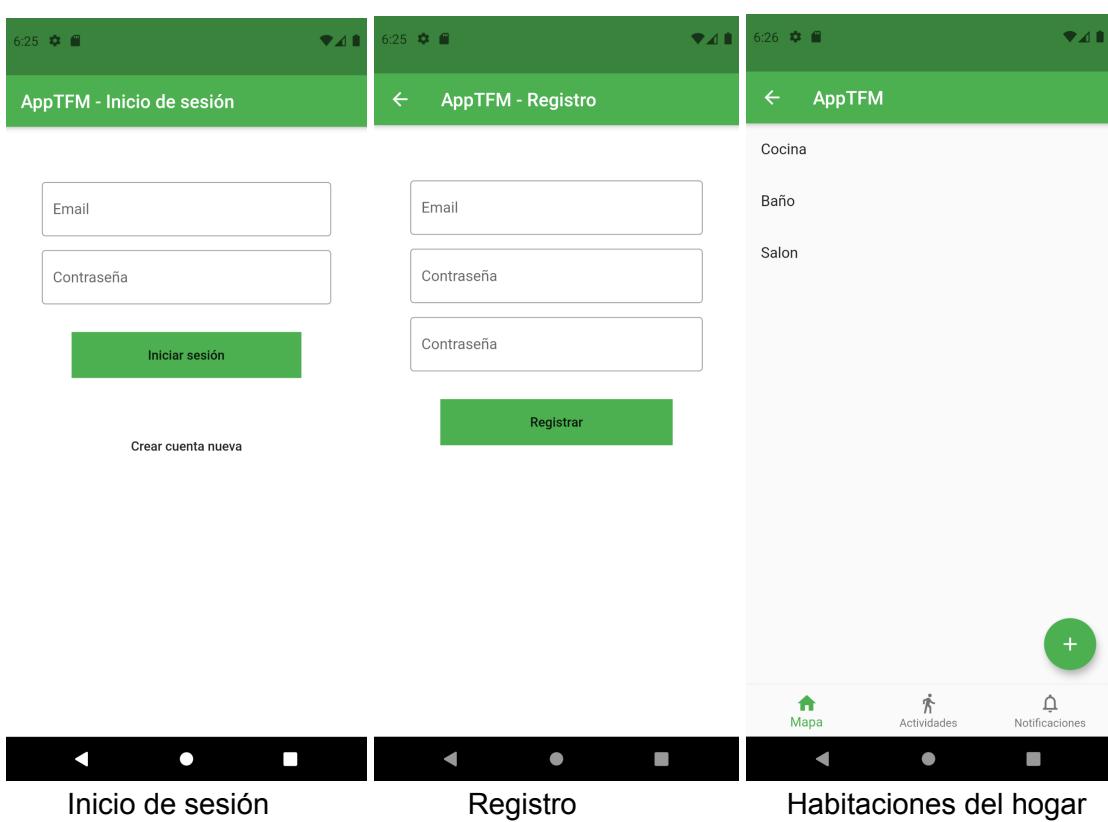


Figura 35: Bocetos de la interfaz

Tras enseñarle al tutor estos diagramas se concluyó que un navigation drawer de android no es la opción de navegación más óptima para una app con tan pocas pantallas. En su lugar se decidió que la navegación se llevaría a cabo con una bottom navigation bar.

## Implementación inicial de la aplicación

Inicialmente la aplicación comenzó a implementarse en Kotlin, pero tras diversos problemas con las bibliotecas disponibles, se decidió migrar el desarrollo a Dart y Flutter, ya que este conjunto de tecnologías permiten un desarrollo más rápido. Flutter es un framework de código fuente abierto de desarrollo de aplicaciones móviles creado por Google. Suele usarse para desarrollar interfaces de usuario para aplicaciones en Android, iOS y Web. La interfaz ha quedado de la siguiente manera:



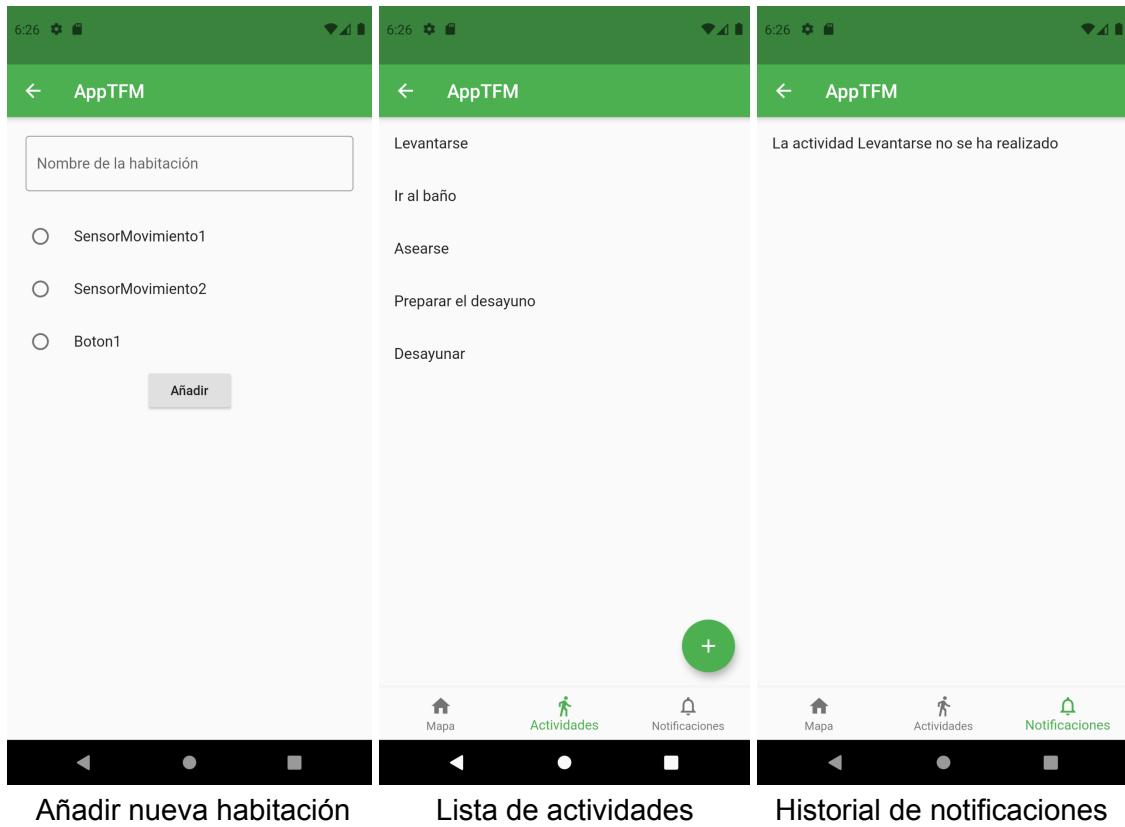


Figura 36: Diseño de la aplicación del cuidador

Para enviar información a la aplicación se ha creado un add-on de home assistant que mediante mqtt recibe los eventos de los sensores y los envía a firebase, desde donde los recibe la aplicación.

Inicialmente se pretendía que la aplicación se conecta únicamente a home assistant utilizando mqtt, pero el soporte de mqtt para Flutter no es lo suficientemente maduro, así que se optó por la solución anteriormente descrita, utilizar un add-on que haga de intermediario.

Los complementos de home assistant son bastante cómodos de crear, ya que utilizan docker, por tanto es fácil incluir cualquier código en un complemento. Un complemento de home assistant consiste en una carpeta con al menos tres archivos:

- Un archivo json de configuración
- Un script o archivo de código fuente que contenga la lógica del add-on
- Un archivo Dockerfile que lance dicho script o compile el código y luego lo ejecute.

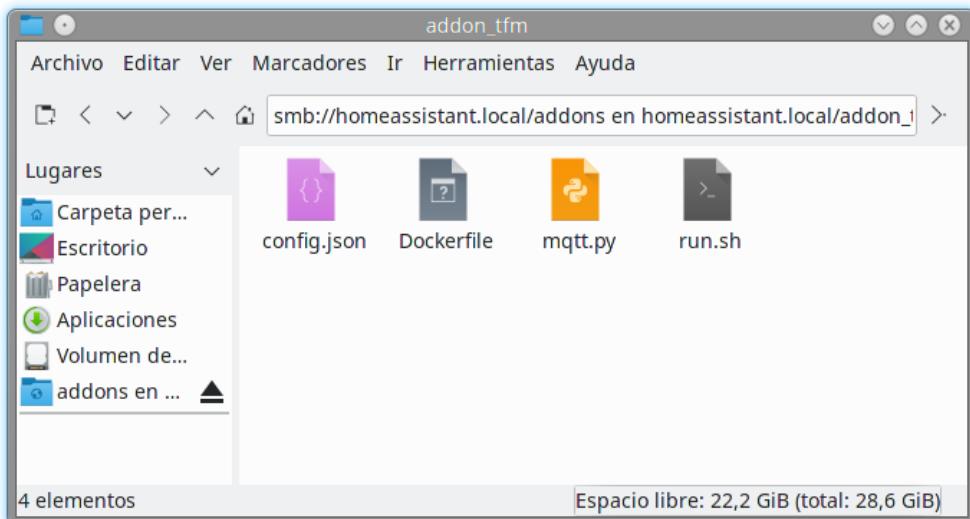


Figura 37: Add-on instalado

A la hora de crear un complemento local para home assistant, es necesario poder acceder a su sistema de archivos, una forma sencilla de hacerlo es instalando el add-on oficial de samba

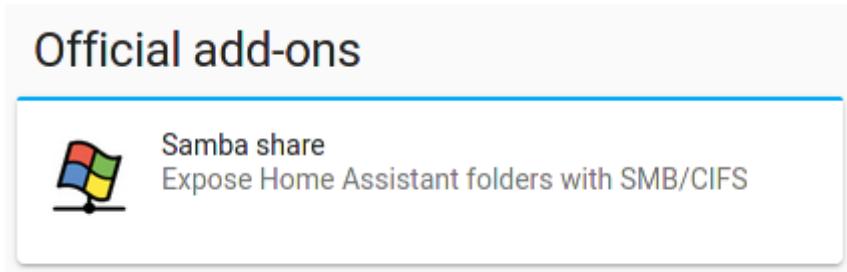


Figura 38: Add-on samba share

Una vez hemos accedido se nos mostrarán una serie de carpetas, dentro de la llamada addons es donde colocaremos nuestras extensiones. Tras hacer esto aparecerá un add-on local en la Add-on store. Lo instalamos y ya estará listo para funcionar.

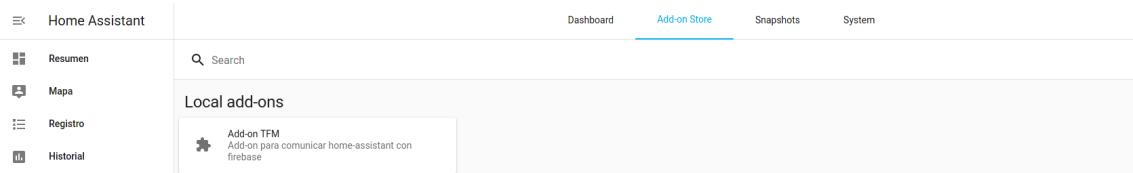


Figura 39: Vista del add-on desarrollado

A pesar de ser mi primer contacto con flutter, el desarrollo de la aplicación ha sido bastante cómodo, existen una gran cantidad de tutoriales por internet y está muy bien integrado con android studio. En mi opinión simplifica bastante el desarrollo de aplicaciones móviles en comparación con Java o Kotlin, además, las aplicaciones realizadas en el son multiplataforma. Por otro lado desarrollar complementos para home assistant también ha resultado cómodo gracias a su uso de Docker.

## 7.3 Tercera entrega: Ampliación del prototipo con un sistema de configuración de actividades

En esta entrega se expande la el prototipo desarrollado en la anterior, ademas se empieza a trabajar con los servicios de Firebase.

### **Creación de una base de datos en Firebase.**

Para la base de datos se ha utilizado el servicio Cloud Firestore de firebase. Cloud Firestore es una base de datos NoSQL flexible y escalable para el desarrollo en servidores, dispositivos móviles y la Web desde Firebase y Google Cloud. Es capaz de mantener datos sincronizados entre apps cliente a través de objetos de escucha en tiempo real y ofrece soporte sin conexión para dispositivos móviles y la Web, por lo que se pueden compilar apps con capacidad de respuesta que funcionan sin importar la latencia de la red ni la conectividad a Internet. [33]

A continuación se presenta un modelo conceptual de la base de datos, en él se pueden ver las relaciones entre las distintas entidades del sistema, así como los parámetros que hará falta almacenar.

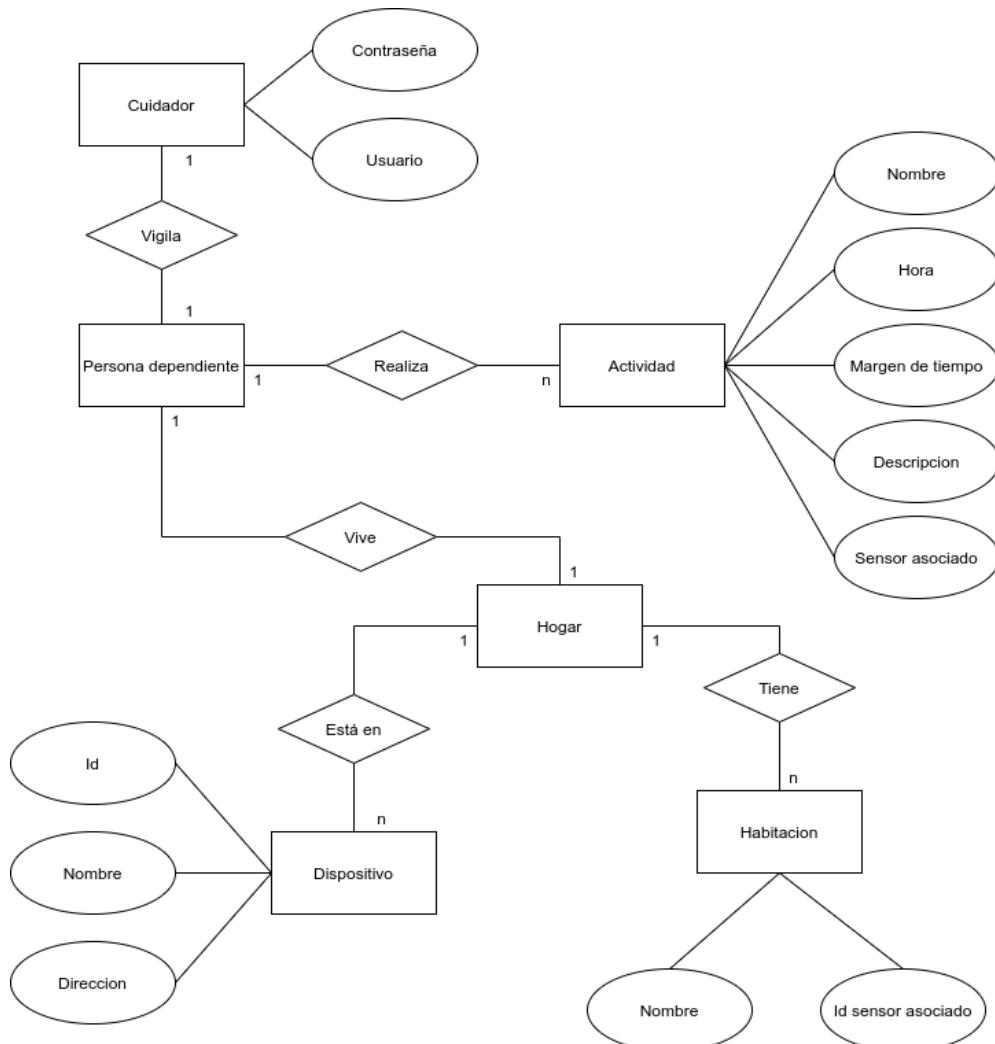


Figura 40. Diagrama entidad relación de la base de datos

Dado que Firestore es una base de datos NoSQL, no los datos no pueden ser almacenados fácilmente en forma de tablas, por ello en lugar de generar una serie de tablas a partir del diagrama anterior, se ha usado una descripción similar al lenguaje YAML para describir las colecciones.

En la base de datos de Firestore ha creado una única colección llamada hogar. Esta colección, como su nombre indica representa un hogar en el que el sistema está instalado.

Cada entrada de la colección tiene la siguiente forma:

Hogar:

- id del hogar
- id del usuario asociado
- Lista de sensores:
  - Sensor 1
    - Id
    - Nombre
    - Dirección
  - Sensor 2
    - Id
    - Nombre
    - Dirección
  - ...
- Lista de habitaciones:
  - Habitación 1
    - Nombre
    - Id del sensor asociado
  - Habitación 2
    - Nombre
    - Id del sensor asociado
  - ...
- Lista de actividades
  - Actividad 1
    - Nombre
    - Hora de realización
    - Margen de tiempo
    - Descripción
    - Sensor asociado
  - Actividad 2
    - Nombre
    - Hora de realización
    - Margen de tiempo
    - Descripción
    - Sensor asociado
  - ...
- Eventos de los dispositivos IOT
  - Evento 1
    - Timestamp
    - Json del evento
  - Evento 2
    - Timestamp
    - Json del evento
  - ...
- Notificaciones
  - Notificación 1
    - Timestamp
    - Cuerpo de la notificación

- Notificacion 2
  - Timestamp
  - Cuerpo de la notificación
- ...

Los usuarios no son almacenados en esta colección, si no que se almacenan utilizando un servicio distinto, Firebase Authentication. Firebase Authentication proporciona servicios de backend, SDK fáciles de usar y bibliotecas de IU ya elaboradas para autenticar a los usuarios. Admite la autenticación mediante contraseñas, números de teléfono, proveedores de identidad federada populares, como Google, Facebook y Twitter, y mucho más. [34]

Para hacer uso de estos servicios en flutter hay que instalar las siguientes bibliotecas cloud\_firestore [35] y firebase\_auth [36]

El proceso de registro e inicio de sesión es relativamente simple, ya que la biblioteca firebase\_auth provee los siguientes métodos:

- createUserWithEmailAndPassword
- signInWithEmailAndPassword

Estos métodos junto a un poco de código boilerplate gestionan el registro y el inicio de sesión de los usuarios

## Sistema de actividades

Un cuidador necesita poder definir las actividades de la vida diaria de la persona que cuida.

Una actividad se compone de un nombre, una hora para su realización, uno sensor que verifica que se ha realizado y una descripción detallada de esta.

En la lista de actividades se encuentra un botón flotante que permite acceder a la pantalla de añadir nuevas actividades.

Una vez introducidos los datos de la actividad, ésta se guarda en firebase haciendo uso de cloud\_firestore.

En la figura 41 se puede ver como quedaría la pantalla para añadir actividades. En la figura 42 se puede ver el diagrama de navegación final.

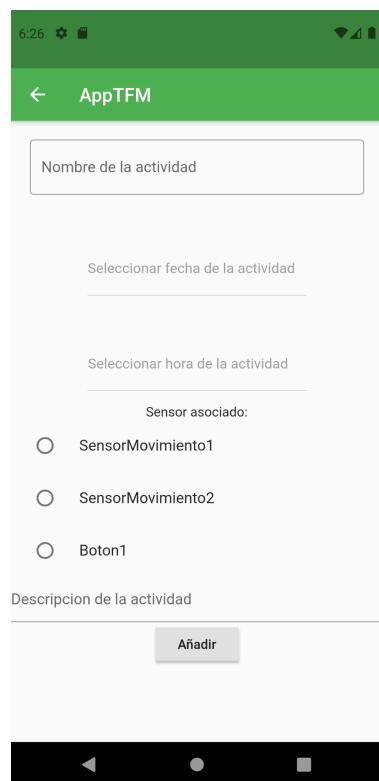


Figura 41. Añadir nueva actividad

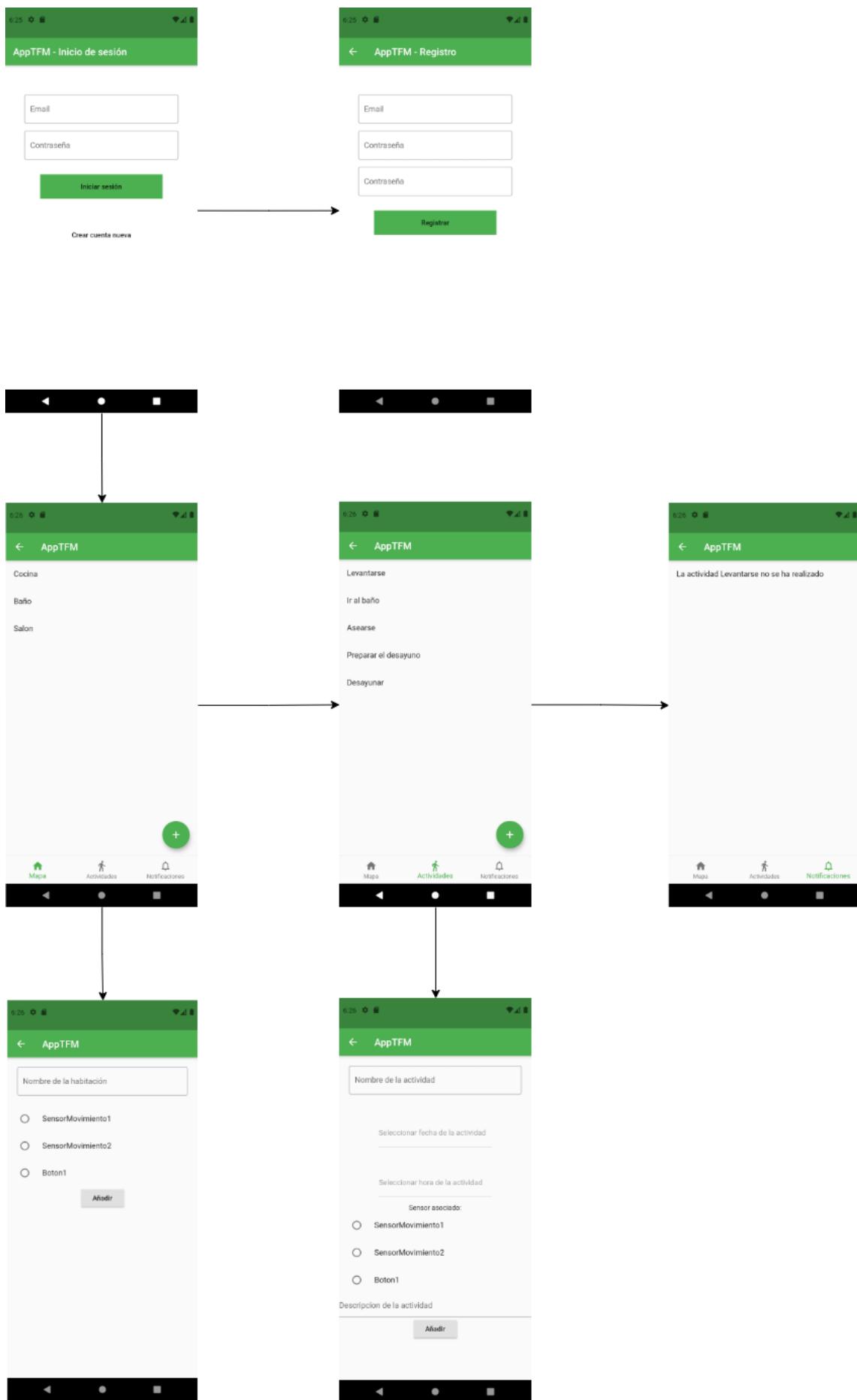


Figura 42: Diagrama de navegación

# Sistema de notificaciones

Las notificaciones se envían desde el add-on de home assistant haciendo uso de Firebase cloud messaging.

El addon se encarga de comparar las notificaciones de MQTT recibidas con las actividades en la base de datos. Para ello recorrerá las actividades y las ordenará temporalmente y esperará a la hora de cada actividad, si en un margen temporal de 10 minutos el sensor relacionado con dicha actividad no ha sido accionado, se enviará una notificación tanto al cuidador como a la persona mayor.

Para enviar dicha notificación se mandará una petición POST a la siguiente URL <https://fcm.googleapis.com/fcm/send> y con el siguiente contenido:

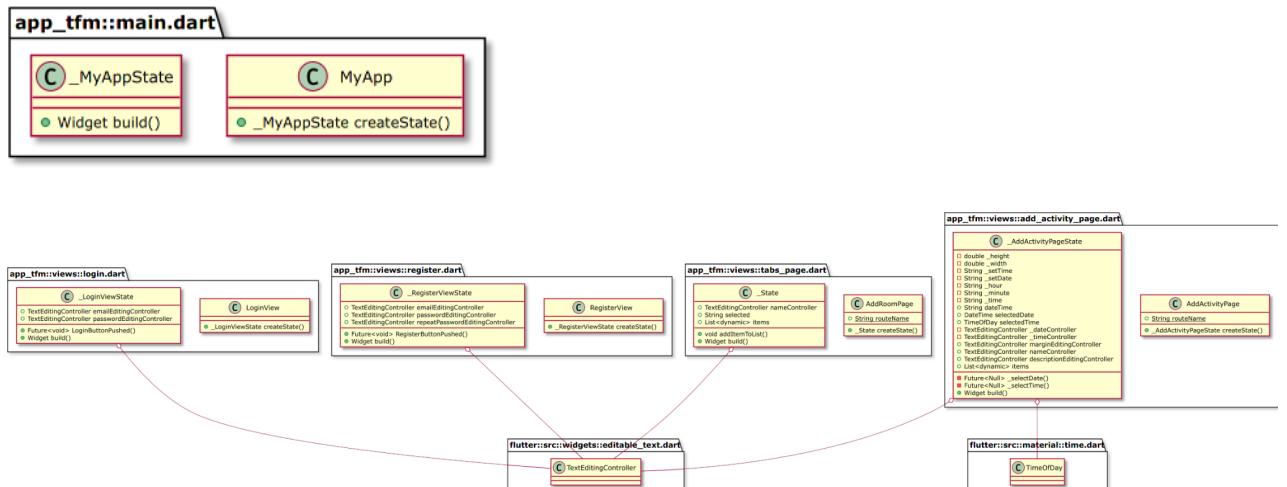
Authorization key="key"

```
{
  "to": "/topics/id_hogar",
  "notification": {
    "title": "Actividad {actividad} no realizada",
    "body": "{descripción de la actividad}"
  },
  "data": {
    "msgId": "msg_12342"
  }
}
```

Con esto se consigue que los dispositivos suscritos al topic "/topics/id\_hogar" reciban una notificación de que dicha actividad no ha sido realizada.

La librería firebase\_messaging [37] proporciona las herramientas necesarias para poder recibir este tipo de notificaciones en flutter.

Tras añadir todo esto el diagrama de clases queda de la siguiente forma:



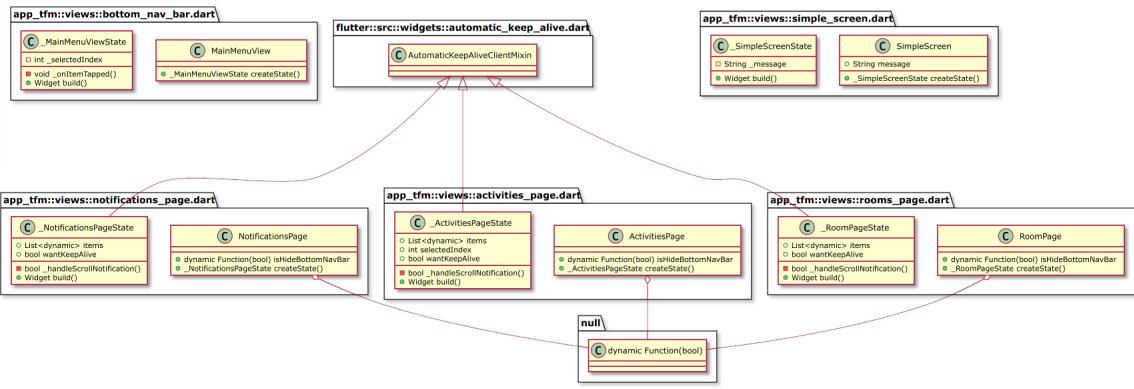


Figura 43: Diagrama de clases de la aplicación del cuidador

Al contrario que con la entrega anterior, esta ha sido más complicada, trabajar con firebase es algo complicado, además, ha sido ahora cuando se ha empezado a manifestar una propiedad de flutter que no me había supuesto un problema hasta el momento: Los mensajes de error. Son poco explicativos, y más aún si tienen que ver con firebase.

## 7.4 Cuarta entrega: Aplicación para personas dependientes

Como se ha mencionado en puntos anteriores, el sistema consta de dos aplicaciones. Uno para cuidadores y otro para personas dependientes. En esta entrega se va a desarrollar la segunda aplicación.

La aplicación para personas dependientes es bastante más sencilla que la del cuidador, ya que solo consta de dos pantallas. Una de inicio de sesión, y otra de notificaciones:

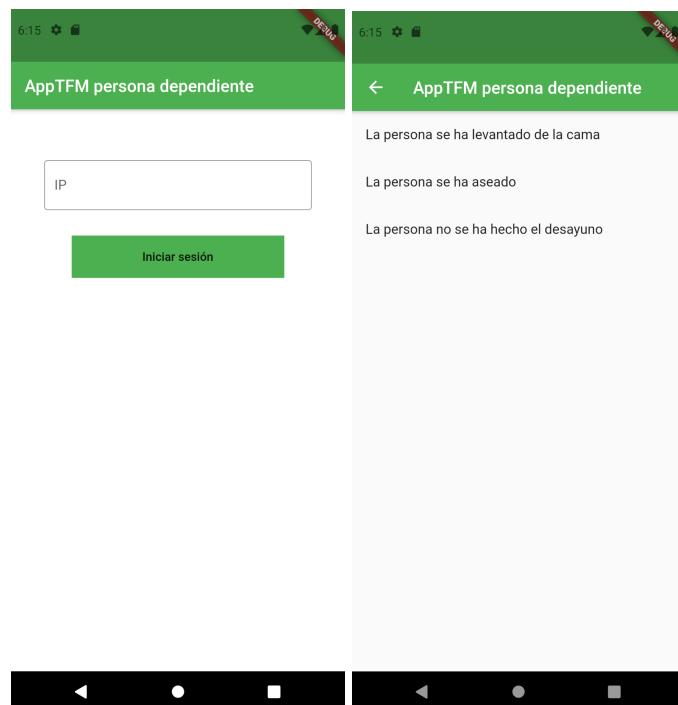


Figura 44: Aplicación de la persona dependiente

Con únicamente dos pantallas, el diagrama de navegación queda bastante sencillo:

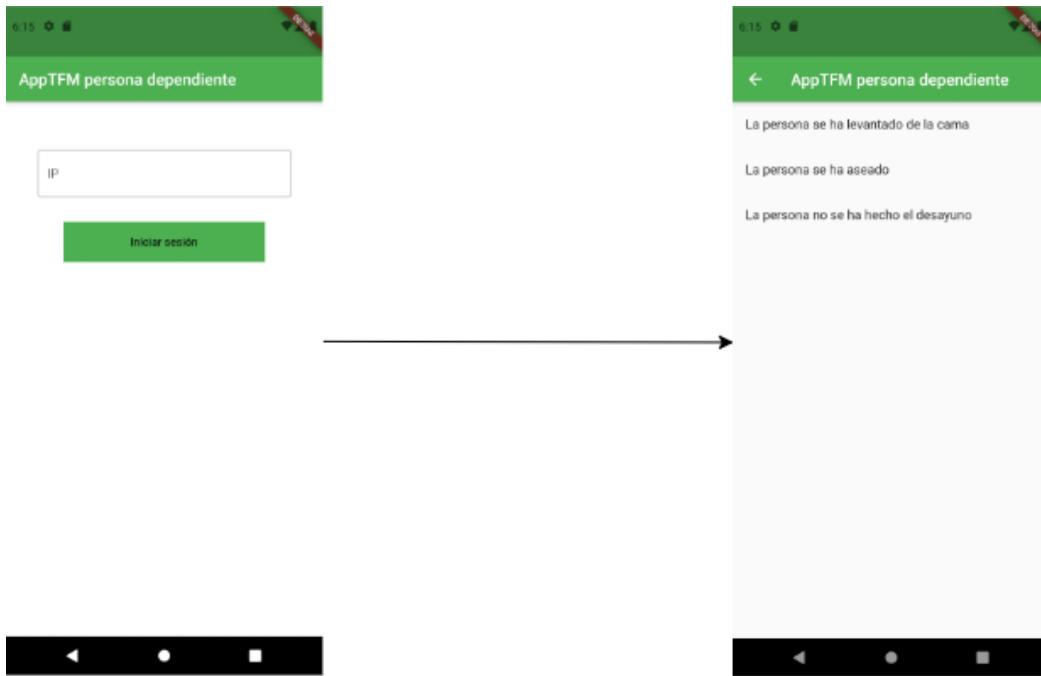


Figura 45: Diagrama de navegación de la aplicación de la persona dependiente

En este caso, la pantalla de inicio de sesión es bastante distinta. Ya que solo requiere la IP local del servidor de Home assistant. ¿Por qué requiere únicamente este dato? Porque el proceso de inicio de sesión en esta aplicación consiste en realizar una petición GET [http://homeassistant.local:3030/id\\_hogar](http://homeassistant.local:3030/id_hogar) y que este le devuelva la ID del hogar. Con esta ID la aplicación se puede suscribir al topic del hogar en cuestión para recibir las notificaciones pertinentes.

Una vez se ha realizado este proceso no es necesario repetirlo ya que la id del hogar no cambia nunca a diferencia de la IP local del servidor.

Una vez la aplicación se ha suscrito al topic, el sistema de recepción de notificaciones es igual que el de la aplicación para cuidadores.

## Detección de caídas

El acelerómetro y el giroscopio de un móvil son los únicos sensores que tenemos disponibles capaces de detectar caídas. Idealmente se utilizará otro tipo de sensor, ya que la monitorización continua de estos sensores es probable que tenga un impacto considerable en la duración de la batería.

Para poder interactuar con estos dos sensores desde flutter es necesario instalar la siguiente biblioteca: `sensors_plus`. Su uso es extremadamente sencillo, permite colocar un listener en cada uno de los sensores. Utilizando estos listeners se puede crear un callback que compruebe la velocidad medida, si la velocidad supera un cierto límite, por ejemplo 10 radianes por segundo, se enviará una notificación al topic `/topic/id_hogar` avisando de una posible caída.

El diagrama de clases de esta aplicación sería el siguiente:

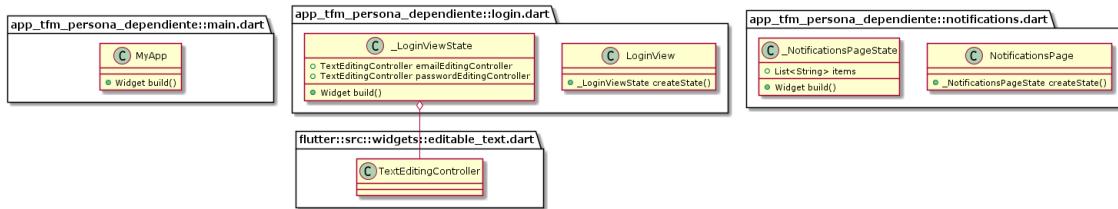


Figura 45: Diagrama de clases de la aplicación de la persona dependiente

Este desarrollo ha sido considerablemente más corto que el resto que el resto de las entregas, ya que la funcionalidad necesaria en esta aplicación es más bien escasa. De hecho inicialmente el sistema se planteó como una única aplicación (cosa que se puede ver en los bocetos iniciales) pero se decidió separarlo en dos aplicaciones porque si sobraba tiempo existía la posibilidad de crear un front end web en Flutter para el cuidador.

## 7.5 Quinta entrega: Pruebas de evaluación de usabilidad

En esta entrega se va a testear la aplicación con usuarios reales. Para ello se va a realizar primero un test de recorridos cognitivos y luego un test de usabilidad.

### Test de recorridos cognitivos

Con el fin de observar cómo los usuarios manejan la aplicación se ha realizado un test de recorridos cognitivos sobre dos usuarios, el primero se trata de un usuario experimentado con las tecnologías, mientras que el segundo es un usuario no demasiado hábil con los ordenadores.

| Tarea                        | Observación  | Categoría                           | Severidad<br>(1-5) | Importancia<br>(1-5) |
|------------------------------|--|-------------------------------------|--------------------|----------------------|
| Registrarse como cuidador    | El usuario se ha registrado sin problemas                      | Éxito                               | 1                  | 2                    |
| Iniciar sesión como cuidador | El usuario ha iniciado sesión sin problemas                    | Éxito                               | 1                  | 3                    |
| Añadir habitaciones          | El usuario ha sabido añadir la habitación , pero le ha costado | Éxito con pequeña aclaración previa | 2                  | 3                    |

|  |   |                                     |   |   |
|--|---|-------------------------------------|---|---|
|  | identificar la lista de sensores asociados  |                                     |   |   |
| Añadir actividades                                 | El usuario ha tenido problemas para distinguir entre el botón de añadir actividad y el de añadir habitaciones, también ha tenido problemas comprendiendo la idea de sensor asociado | Éxito, pero con varias aclaraciones | 3 | 4 |
| Borrar una actividad                               | El usuario ha tardado en encontrar la opción de borrado   | Éxito                               | 2 | 3 |
| Borrar una habitación                              | El usuario ha borrado la aplicación sin problemas   | Éxito                               | 1 | 1 |
| Iniciar sesión como persona dependiente            | El usuario no ha tenido problemas para iniciar sesión, pero ha manifestado su descontento por el método de inicio de sesión   | Éxito                               | 4 | 3 |
| Comprendión general del funcionamiento del sistema | Requiere una explicación previa   | Éxito                               | 3 | 5 |

Tabla 8: Recorrido cognitivo del primer usuario

| Tarea  | Observación  | Categoría             | Severidad<br>(1-5) | Importancia<br>(1-5) |
|--|--|-----------------------|--------------------|----------------------|
| Registrarse como cuidador                          | Al usuario le ha costado un poco encontrar el botón de registro                  | Éxito                 | 2                  | 2                    |
| Iniciar sesión como cuidador                       | El usuario ha realizado la tarea sin problemas                                   | Éxito                 | 1                  | 3                    |
| Añadir habitaciones                                | El usuario ha requerido una explicación al respecto de los sensores asociados    | Éxito, pero con ayuda | 3                  | 3                    |
| Añadir actividades                                 | El usuario ha requerido una explicación respecto al concepto de actividad        | Éxito                 | 3                  | 4                    |
| Borrar una actividad                               | El usuario no ha encontrado el botón de borrado                                  | Error                 | 3                  | 3                    |
| Borrar una habitación                              | El usuario no ha encontrado el botón de borrado                                  | Error                 | 3                  | 1                    |
| Iniciar sesión como persona dependiente            | El usuario no ha sabido iniciar sesión porque no conocía la IP de Home Assistant | Error                 | 4                  | 3                    |
| Comprensión general del funcionamiento del sistema | El usuario ha tenido ciertos problemas para entender el funcionamiento de la APP | ~                     | 3                  | 5                    |

Tabla 9: Recorrido cognitivo del segundo usuario

## Cuestionario de usabilidad

Con el fin de identificar posibles errores en el diseño de la interfaz, se va a realizar un test de usabilidad sobre unos usuarios.

| Pregunta  | Usuario 1  | Usuario 2   |
|---|--|---|
| ¿Qué te ha parecido la forma de recorrer los menús?                         | Poco habitual, no estoy acostumbrado a esa forma de recorrer los menús | Bien, la barra de abajo aclara cómo moverse por la aplicación             |
| ¿Cómo de difíciles de realizar te han parecido las distintas acciones?      | No demasiado tras recibir una explicación                              | No está clara la forma de realizar algunas acciones                       |
| ¿Cuál es la tarea más complicada de realizar?                               | Iniciar sesión como persona dependiente                                | Iniciar sesión como persona dependiente, requiere saber la IP previamente |
| ¿Qué opinas de la estética de la aplicación?                                | Demasiado espacio vacío  | Está bien   |
| ¿Te parece adecuada la localización de las opciones?                        | Si   | Si  |
| ¿Te parece adecuado como está distribuida la información en la aplicación ? | Si   | Si  |
| ¿Encuentras intuitiva la navegación por la aplicación?                      | Si, aunque requiere una pequeña familiarización                        | Si, la barra de abajo ayuda   |
| ¿El uso de los iconos te parece adecuado?                                   | Si   | Si, aunque estaría bien que las actividades tuvieran iconos               |
| ¿Cómo de difícil has encontrado añadir una actividad?                       | Fácil  | Medio complicado  |
| ¿Cómo de difícil has encontrado añadir una habitación?                      | Fácil  | Más fácil que añadir una actividad  |
| ¿Cómo de difícil has encontrado iniciar sesión?                             | Fácil  | Fácil   |
| ¿Cómo de difícil has encontrado registrarte?                                | Fácil  | El botón de registro no es muy visible                                    |
| ¿Ves viable el uso diario de esta aplicación?                               | Si   | Si, aunque con algunos ajustes  |

Tabla 10. Test de usabilidad

Tras realizar estos tests, se puede ver que algunos usuarios, especialmente los menos experimentados en el ámbito de las nuevas tecnologías han tenido dificultades para realizar ciertas actividades. Ambos usuarios han visto problemática la forma de iniciar sesión en la aplicación de la persona dependiente. No obstante dado que esta actividad solo tiene que ser realizada una única vez por el usuario que ha desplegado el sistema, no creo que haya demasiados problemas.

## 7.6 Sexta entrega: Despliegue del sistema

### **Instrucciones de instalación del software:**

#### Método lento:

1. Descargar e instalar Home Assistant así como el add-on de Zigbee2mqtt en una Raspberry pi 3 con una antena zigbee compatible (El software solo se ha probado en una Raspberry Pi 3, pero es probable que funcione en una Raspberry Pi 4).  
El proceso detallado está descrito en el punto 7.1
2. Descargar el repositorio del addon de home assistant que se ha desarrollado durante este proyecto. Instalar dicho add-on de forma manual  
Para una descripción más en profundidad del proceso, mirar el punto 7.2
3. Descargar e instalar Flutter
  - a. En un sistema Ubuntu 18.04 o superior se puede instalar desde la snap store utilizando el siguiente comando: “sudo snap install flutter --classic”
4. Descargar el repositorio que contiene las aplicaciones de este proyecto y ejecutar en la carpeta raíz de cada aplicación: “flutter build apk”. Cuando termine el proceso de compilación se habrá generado un archivo apk en la siguiente ruta: “./app\_tfm/build/app/outputs/apk/release/app-release.apk”

## **Despliegue**

Una vez se ha descargado e instalado el software se puede proceder a instalar y configurar los dispositivos IoT.

Previamente a la instalación el usuario cuidador debe pensar que actividades de la vida diaria desea monitorizar. Después el usuario deberá distribuir los sensores por el hogar, de forma que la realización de una actividad active un sensor. Los sensores de apertura y cierre de puerta son bastante versátiles y tienen más usos de los que de primeras se puede pensar. Por ejemplo se pueden colocar en un pastillero para comprobar cuando la persona mayor lo ha abierto para tomarse las pastillas.

Si la actividad es demasiado compleja o el cuidador no tiene un sensor capaz de detectar dicha actividad, siempre se puede colocar un botón para que la persona dependiente lo pulse cuando vaya a realizar la actividad.

Tras distribuir de forma adecuada los sensores, se procederá a configurar las aplicaciones móviles y el sistema. Pasos para realizar dicha configuración:

1. Para empezar el cuidador necesita registrar un hogar utilizando la aplicación del usuario cuidador.
2. A continuación el cuidador deberá iniciar sesión en su instalación local de Home Assistant, dirigirse al menú de Zigbee2mqtt y enlazar los distintos dispositivos IoT. Este proceso está descrito con más detalle en el punto 7.1.
3. Aun en la interfaz de Home Assistant, dirigirse a la configuración del add-on desarrollado en este proyecto e introducir las credenciales del hogar registrado en el punto 1.
4. A continuación el cuidador podrá iniciar sesión en su aplicación y podrá empezar a añadir las distintas habitaciones de la casa, así como las actividades de la vida diaria.

## 8. Conclusiones y trabajo futuro

Una herramienta como la desarrollada en este proyecto puede ser de gran utilidad para personas mayores y sus cuidadores. No obstante los dispositivos IoT con los que se han trabajado no son los más aptos para la monitorización. Tener que andar pulsando un botón cada vez que se realiza una actividad es una tarea fácil de olvidar para una persona mayor, seguramente habría sido interesante tratar de adaptar alguno de estos botones para que se activen cuando por ejemplo se tire de la cisterna. Ahora bien, el objetivo de este proyecto era aprovechar un sistema domótico ya instalado para monitorizar personas mayores, las limitaciones impuestas por este tipo de sensores forman parte de la investigación.

Home Assistant ha resultado ser una herramienta muy interesante, potente y cómoda de utilizar. Tiene gran cantidad de herramientas de automatización y su interfaz es muy intuitiva. Si algún día domotizo mi casa seguramente mantenga una instancia local de Home Assistant.

En lo que respecta al desarrollo, flutter ha resultado ser una agradable sorpresa. Su modelo de programación es bastante distinto al de android tradicional, pero es fácil de comprender y permite un ciclo de desarrollo muy ágil gracias a características como el hot reloading. Eso sí, la interacción con algunas de sus librerías se ha vuelto un tanto engorrosa

Posibles mejoras en el futuro:

- Añadir interfaz de escritorio y web: Ya que flutter es un framework multiplataforma, desarrollar una versión de escritorio es bastante sencillo. Eso sí, requeriría un rediseño de la interfaz. Este rediseño podría ser utilizado también para generar una versión web.
- Versión de IOS: En la línea de la mejora anterior, flutter también puede ser compilado para IOS, aunque requiere de un ordenador con macOS. Si se compilase la versión actual para IOS, el resultado sería una aplicación con interfaz material design, cosa que no es habitual en el sistema operativo de Apple, por lo que esta versión también requeriría cierto rediseño de la interfaz.
- Mejorar la interfaz de la aplicación de la persona dependiente: Si bien esta aplicación es muy sencilla, para el primer inicio es necesario conocer la IP del servidor local de Home Assistant. Esto puede ser un problema para los usuarios menos hábiles en lo que a las nuevas tecnologías respecta. Una posible alternativa consistiría en que la aplicación del cuidador generase un código QR. Para iniciar sesión como persona dependiente bastaría con leer dicho QR.

## 9. Bibliografía

- [1] S, A. R. y N, V. K. M. (2020, 23 de enero). (PDF) Smart Companion Robot for Elders Health Care with Emergency Rescue System. ResearchGate.  
[https://www.researchgate.net/publication/344386027\\_Smart\\_Companion\\_Robot\\_for\\_Elders\\_Health\\_Care\\_with\\_Emergency\\_Rescue\\_System](https://www.researchgate.net/publication/344386027_Smart_Companion_Robot_for_Elders_Health_Care_with_Emergency_Rescue_System) Último acceso 7 Septiembre 2021
- [2] Hu, B. D. C., Fahmi, H., Yuhao, L. y Harun, A. (2018, 1 de agosto). (PDF) Internet of Things (IOT) Monitoring System for Elderly. ResearchGate.  
[https://www.researchgate.net/publication/329491801\\_Internet\\_of\\_Things\\_IOT\\_Monitoring\\_System\\_for\\_Elderly](https://www.researchgate.net/publication/329491801_Internet_of_Things_IOT_Monitoring_System_for_Elderly) Último acceso 7 Septiembre 2021
- [3] Park, S. J., Subramaniyam, M., Kim, S. E. y Seo, Y. (2017, 1 de enero). (PDF) Development of the Elderly Healthcare Monitoring System with IoT. ResearchGate.  
[https://www.researchgate.net/publication/305876749\\_Development\\_of\\_the\\_Elderly\\_Healthcare\\_Monitoring\\_System\\_with\\_IoT](https://www.researchgate.net/publication/305876749_Development_of_the_Elderly_Healthcare_Monitoring_System_with_IoT) Último acceso 7 Septiembre 2021
- [4] Ziyu Lv, Feng Xia, Guowei Wu, Lin Yao, Zhikui Chen arXiv.org e-Print archive.  
<https://arxiv.org/ftp/arxiv/papers/1011/1011.3852.pdf> Último acceso 7 Septiembre 2021
- [5] NHS website. (s. f.). Loneliness in older people. nhs.uk.  
<https://www.nhs.uk/mental-health/feelings-symptoms-behaviours/feelings-and-symptoms/loneliness-in-older-people/> Último acceso 7 Septiembre 2021
- [6] Contributors to Wikimedia projects. (2003, 18 de febrero). Home automation - Wikipedia. Wikipedia, the free encyclopedia.  
[https://en.wikipedia.org/wiki/Home\\_automation](https://en.wikipedia.org/wiki/Home_automation) Último acceso 7 Septiembre 2021

- [7] Contributors to Wikimedia projects. (2003b, 29 de mayo). Activities of daily living - Wikipedia. Wikipedia, the free encyclopedia.  
[https://en.wikipedia.org/wiki/Activities\\_of\\_daily\\_living](https://en.wikipedia.org/wiki/Activities_of_daily_living) Último acceso 7 Septiembre 2021
- [8] AVSystem. (2020, 4 de noviembre). Top sensor types used in IoT. AVSystem – Shaping The World of Connected Devices.  
<https://www.avsystem.com/blog/iot-sensors-iot-actuators/> Último acceso 7 Septiembre 2021
- [9] Basic IoT Actuators. (s. f.). IoT Bytes.  
<https://iotbytes.wordpress.com/basic-iot-actuators/> Último acceso 7 Septiembre 2021
- [10] 12 plataformas y herramientas de Internet de las cosas (IoT) de código abierto. (s. f.). Geekflare. <https://geekflare.com/es/iot-platform-tools/> Último acceso 7 Septiembre 2021
- [11] openHAB. (s. f.). openHAB. <https://www.openhab.org/> Último acceso 7 Septiembre 2021
- [12] Contributors to Wikimedia projects. (2019, 26 de noviembre). OpenHAB - Wikipedia. Wikipedia, the free encyclopedia. <https://en.wikipedia.org/wiki/OpenHAB> Último acceso 7 Septiembre 2021
- [13] Home Assistant. (s. f.). Home Assistant. <https://www.home-assistant.io/> Último acceso 7 Septiembre 2021
- [14] Contributors to Wikimedia projects. (2020, 27 de mayo). Home Assistant - Wikipedia. Wikipedia, the free encyclopedia.  
[https://en.wikipedia.org/wiki/Home\\_Assistant](https://en.wikipedia.org/wiki/Home_Assistant) Último acceso 7 Septiembre 2021
- [15] Domoticz. (s. f.). Domoticz. <https://www.domoticz.com/> Último acceso 7 Septiembre 2021
- [16] GitHub - domoticz/domoticz: Open source Home Automation System. (s. f.). GitHub. <https://github.com/domoticz/domoticz> Último acceso 7 Septiembre 2021
- [17] About Domoticz - Domoticz. (s. f.). Domoticz.  
[https://www.domoticz.com/wiki/About\\_Domoticz](https://www.domoticz.com/wiki/About_Domoticz) Último acceso 7 Septiembre 2021
- [18] Passive Monitoring | TruSense. (s. f.). TruSense. <https://www.mytrusense.com/> Último acceso 7 Septiembre 2021
- [19] Kwido Home | Sistema de Monitorización de Mayores en Domicilio. (s. f.). Kwido. <https://kwido.com/es/monitorizacion-mayores-domicilio/> Último acceso 7 Septiembre 2021
- [20] Helping Seniors Live Safely | Rest Assured. (s. f.). Rest Assured.  
<https://restassured.com/seniors/> Último acceso 7 Septiembre 2021

[21] AI Senior Safety Monitoring System – We make home safer for seniors. (s. f.). AI Senior Safety Monitoring System – We make home safer for seniors.  
<https://aeyesafe.com/> Último acceso 7 Septiembre 2021

[22] Best of open source smart home: Home Assistant vs OpenHAB. (s. f.). Smart Home University.  
<https://smarthome.university/your-smart-home-platform-home-assistant-vs-openhab/>  
Último acceso 7 Septiembre 2021

[23] Contributors to Wikimedia projects. (2011, 6 de mayo). Raspberry Pi - Wikipedia. Wikipedia, the free encyclopedia. [https://en.wikipedia.org/wiki/Raspberry\\_Pi](https://en.wikipedia.org/wiki/Raspberry_Pi) Último acceso 7 Septiembre 2021

[24] Zigbee network. (s. f.). [zigbee2mqtt.io](https://www.zigbee2mqtt.io/information/zigbee_network.html).  
[https://www.zigbee2mqtt.io/information/zigbee\\_network.html](https://www.zigbee2mqtt.io/information/zigbee_network.html) Último acceso 7 Septiembre 2021

[25] Contributors to Wikimedia projects. (2003, 12 de marzo). Zigbee - Wikipedia. Wikipedia, the free encyclopedia. <https://en.wikipedia.org/wiki/Zigbee> Último acceso 7 Septiembre 2021

[26] Contributors to Wikimedia projects. (2011b, 9 de agosto). MQTT - Wikipedia. Wikipedia, the free encyclopedia. <https://en.wikipedia.org/wiki/MQTT> Último acceso 7 Septiembre 2021

[27] GitHub - Koenkk/zigbee2mqtt: Zigbee  to MQTT bridge , get rid of your proprietary Zigbee bridges . (s. f.). GitHub. <https://github.com/koenkk/zigbee2mqtt>  
Último acceso 7 Septiembre 2021

[28] Zigbee network. (s. f.-b.). [zigbee2mqtt.io](https://www.zigbee2mqtt.io/information/zigbee_network.html).  
[https://www.zigbee2mqtt.io/information/zigbee\\_network.html](https://www.zigbee2mqtt.io/information/zigbee_network.html) Último acceso 7 Septiembre 2021

[29] Contributors to Wikimedia projects. (2004, 6 de mayo). Agile software development - Wikipedia. Wikipedia, the free encyclopedia.  
[https://en.wikipedia.org/wiki/Agile\\_software\\_development](https://en.wikipedia.org/wiki/Agile_software_development) Último acceso 7 Septiembre 2021

[30] Contributors to Wikimedia projects. (2006, 14 de abril). Scrum (software development) - Wikipedia. Wikipedia, the free encyclopedia.  
[https://en.wikipedia.org/wiki/Scrum\\_\(software\\_development\)](https://en.wikipedia.org/wiki/Scrum_(software_development)) Último acceso 7 Septiembre 2021

[31] balenaEtcher - Flash OS images to SD cards & USB drives. (s. f.). balenaEtcher.  
<https://www.balena.io/etcher/> Último acceso 7 Septiembre 2021

[32] ZonaGadget. (2020, 24 de octubre). Home Assistant: Instalación y configuración Addon Zigbee2mqtt | Home Assistant Addons [Video]. YouTube.  
<https://www.youtube.com/watch?v=ouTLTMuySTM> Último acceso 7 Septiembre 2021

[33] Cloud Firestore | Firebase Documentation. (s. f.). Firebase.  
<https://firebase.google.com/docs/firestore> Último acceso 7 Septiembre 2021

[34] Firebase Authentication. (s. f.). Firebase. <https://firebase.google.com/docs/auth>  
Último acceso 7 Septiembre 2021

[35] cloud\_firestore | Flutter Package. (s. f.). Dart packages.  
[https://pub.dev/packages/cloud\\_firestore](https://pub.dev/packages/cloud_firestore) Último acceso 7 Septiembre 2021

[36] firebase\_auth | Flutter Package. (s. f.). Dart packages.  
[https://pub.dev/packages/firebase\\_auth](https://pub.dev/packages/firebase_auth) Último acceso 7 Septiembre 2021

[37] firebase\_messaging | Flutter Package. (s. f.). Dart packages.  
[https://pub.dev/packages/firebase\\_messaging](https://pub.dev/packages/firebase_messaging) Último acceso 7 Septiembre 2021

[38] sensors\_plus | Flutter Package. (s. f.). Dart packages.  
[https://pub.dev/packages/sensors\\_plus](https://pub.dev/packages/sensors_plus) Último acceso 7 Septiembre 2021