UNIVERSITAT POLITÈCNICA DE VALÈNCIA

_ TELECOM ESCUELA TÉCNICA VLC SUPERIOR DE INGENIERÍA DE TELECOMUNICACIÓN

PLATAFORMA Y APLICACIÓN IOT PARA EVITAR EL AISLAMIENTO SOCIAL DE PERSONAS MAYORES

Dario Alandes Codina

Tutor: Carlos Enrique Palau Salvador

Trabajo Fin de Máster presentado en la Escuela Técnica Superior de Ingeniería de Telecomunicación de la Universitat Politècnica de València, para la obtención del Título de Máster en Ingeniería Telecomunicación

Curso 2020-21

Valencia, 25 de junio de 2021

Escuela Técnica Superior de Ingeniería de Telecomunicación
Universitat Politècnica de València
Edificio 4D. Camino de Vera, s/n, 46022 Valencia
Tel. +34 96 387 71 90, ext. 77190
www.etsit.upv.es

VLC/ CAMPUS
VALENCIA, INTERNATIONAL CAMPUS OF EXCELLENCE

EMAS
Gestión ambiental verificada
REG.NO. ES-CV- 000030

Resumen

Este proyecto final de máster de carácter profesional pretende realizar el desarrollo de una aplicación IoT, basado en una plataforma IoT abierta, para la detección del aislamiento social que pueden llegar a sufrir las personas mayores para poder evitarlo. Para ello se hará uso de las diferentes tecnologías del llamado internet de las cosas para llevar a cabo una monitorización de las actividades de la persona mayor.

Para la realización se desarrollará una aplicación IoT capaz de recopilar los datos de los diferentes sensores y prepararlos para ser analizados por una aplicación Android que muestre las actividades realizadas por la persona mayor a lo largo de una semana.

El objetivo final del proyecto es proporcionar a cuidadores herramientas que permitan tener un conocimiento de las actividades que realiza la persona que está a su cargo, haciendo que su cuidado se pueda enfocar en alguna de las carencias que pueda tener.

Resum

Aquest projecte final de màster de caràcter professional pretén realitzar el desenvolupament d'una aplicació IOT, basat en una plataforma Iot oberta, per a la detecció de l'aïllament social que poden arribar a patir les persones grans per poder evitar-ho. Per a això es farà ús de les diferents tecnologies de l'anomenat internet de les coses per dur a terme un monitoratge de les activitats de la persona gran.

Per a la realització es desenvoluparà una aplicació IOT capaç de recopilar les dades dels diferents sensors i preparar-los per ser analitzats per una aplicació Android que mostri les activitats realitzades per la persona gran al llarg d'una setmana.

L'objectiu final del projecte és proporcionar als cuidadors eines que permetin tenir un coneixement de les activitats que realitza la persona que està al seu càrrec, que fa que la cura es pugi enfocar en alguna de les mancances que pugi tenir.

Abstract

This final master's degree project aims at developing an IoT application, based on an open IoT platform, to detect and avoid the social isolation older people may suffer. To do so, different technologies of the so-called *internet of things* were used for monitoring activities carried out by elderly people.

It was developed an IoT application capable of collecting data from the different sensors and preparing them to be analyzed by an Android application, showing the activities carried out by an elderly person over a week.

The final aim of the project is to provide caregivers with tools that allow them to understand the activities carried out by the person in their charge, so that their care can be focused on any of the deficiencies they may have.

Índice

Capítulo 1. Introducción

La estructura de la población mundial está experimentando cambios importantes. La OCDE (Organización para la Cooperación y el Desarrollo Económicos) pronostica que, a nivel mundial, la proporción de personas mayores de 65 años en la población se duplicará con creces hasta el 16,2 % para 2050, y que el porcentaje de personas de 80 años o más se triplicará. [1]

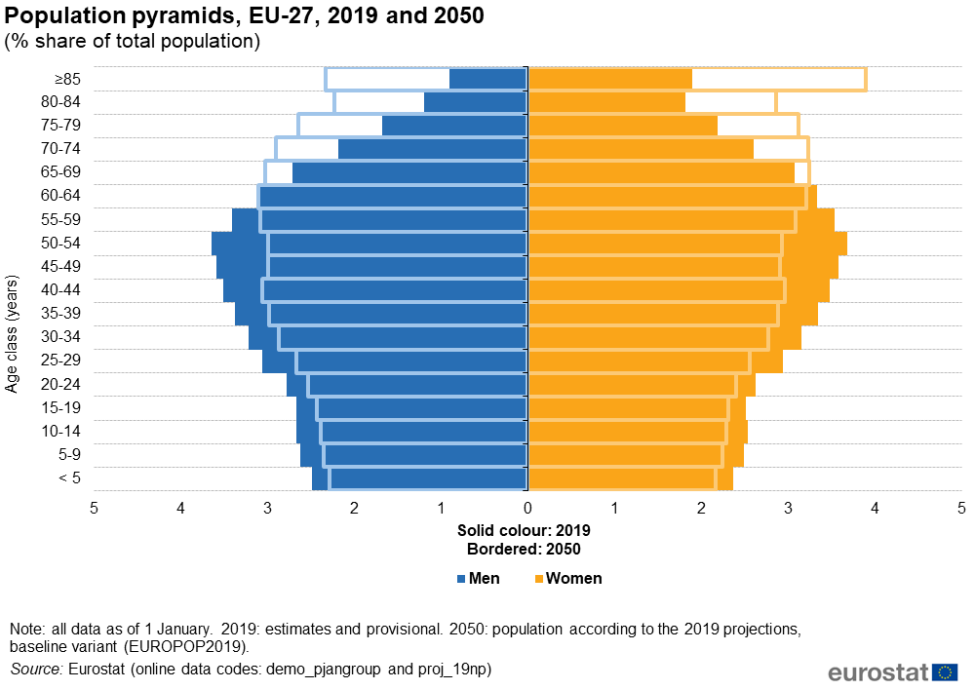

Figura 1 Previsión demográfica para 2050

La disminución de las tasas de fertilidad, junto con el aumento de la esperanza de vida general, está aumentando la proporción de personas mayores en la población. Como resultado, se espera que la demanda de cuidados para personas mayores aumente significativamente. Si bien las naciones emergentes mantienen niveles de fertilidad relativamente altos, las tasas de natalidad en Europa han estado cayendo durante décadas, lo que ha resultado en proporciones significativamente más altas de personas mayores. Casi el 30% de la población europea tendrá más de 65 años en 2050 y más del 11% tendrá más de 80 años. Se espera que Alemania, Italia y España tengan un 13% o más de su población de 80 años o más. El creciente número de personas mayores tendrá un impacto en la demanda futura de servicios de atención a largo plazo (LTC). Sin embargo, aunque se prevé que todos los tipos de servicios LTC aumenten, no lo harán de

manera uniforme. La atención domiciliaria, la línea de servicio preferida para la mayoría de los países europeos en los últimos años, resultará ineficaz dada la creciente proporción de pacientes con necesidades médicas especializadas. Además, los cambios sociales, incluida la disminución del tamaño de la familia, el aumento de la falta de hijos y el aumento de los arreglos de vida no tradicionales, significan que las residencias de ancianos desempeñarán un papel crucial en el cuidado de la creciente población de ancianos. [2]

Los cambios en nuestra estructura social están obligando a muchas personas mayores a vivir en hogares de un solo residente, o en un dormitorio privado en una residencia de ancianos o en apartamentos de vida asistida, pero siempre solos lejos de su familia y sin participar en actividades sociales.

Mantener el sentido de comunidad en una persona es una tarea muy importante, el aislamiento social puede ocasionar graves consecuencias, como: síndrome de desuso, depresión mental, incluso suicidio.

El aislamiento social de las personas mayores es un concepto multidimensional, al que se refieren tanto aspectos de carácter "estructural" (como vivir solo y la escasez de relaciones) como aspectos de carácter "funcional" (como el apoyo material y emocional que transmiten las relaciones existentes).

La principal causa del aislamiento social es la ausencia o relativa escasez de relaciones sociales, que representan un importante factor de riesgo para la salud comparable, si no superior, a factores de riesgo bien conocidos como el tabaquismo, el abuso de alcohol y la obesidad.

En particular, en las personas mayores, también se ha descubierto que el aislamiento social es uno de los desencadenantes más comunes para el deterioro de las habilidades cognitivas y, en general, de un peor estado de salud, tanto psicológico como físico.

Las relaciones sociales pueden influir en el estado de salud a través del intercambio de información, apoyo emocional y ayuda material, que a su vez promueven el comportamiento adaptativo en presencia de fuentes de estrés agudas o crónicas; pero la red de relaciones también puede actuar, directa o indirectamente, en promover la adopción de conductas saludables.

Finalmente, también hay evidencia de una relación directa entre el aislamiento social y la salud. Para las personas mayores solitarias, la pérdida de contacto con un ser querido tiene un impacto negativo que podría superarse manteniéndose en contacto con ex compañeros de trabajo, familiares y amigos. Además, involucrar a las personas mayores, incluso en actividades sencillas, podría reavivar su entusiasmo perdido.

Es común pensar que la mayoría de las personas mayores que viven solas sufren de aislamiento social y que, en cambio, quienes viven en una residencia de ancianos sufren menos porque están en contacto con otras personas casi constantemente. De hecho, incluso dentro de la residencia, es bastante común que los ancianos se sientan solos, pasando gran parte de su tiempo en su habitación o en silencio. Se podría decir, de hecho, que vivir en un lugar ajeno a la vida anterior puede agravar la sensación de soledad y por tanto la aparición de penurias desde el aislamiento social. Las opciones sobre la estructura de la atención son una decisión increíblemente importante que toman en nombre de las personas que a menudo son extremadamente vulnerables. La naturaleza y la calidad de la atención tienen un impacto enorme en la felicidad, la salud y la longevidad de la persona.

La solución para hacer frente al aislamiento social de las personas mayores se centra en dos necesidades principales de las personas mayores:

- Mantener un nivel de interacción, que puede ser con ellos mismos, con el entorno que los rodea y por supuesto con otras personas, especialmente con familiares o amigos cercanos.
- No perder el presente, el anciano trae consigo un montón de vivencias, de momentos de su vida que suelen dar lugar a un sentimiento común, la nostalgia. En el momento en que un anciano siente nostalgia en su mente, el presente convive con el pasado y la persona que está allí se encuentra cara a cara con la persona que fue.

La nostalgia es un elemento clave utilizado para desconectar a las personas mayores del aislamiento. Paradójicamente, de hecho, es precisamente la reconstrucción mental de nuestro pasado lo que nos ayuda a recordar quiénes somos ahora y las relaciones que podemos seguir teniendo. La nostalgia puede llegar repentinamente al escuchar una canción o al escuchar el dialecto del lugar de origen. La consideración más constructiva para el bienestar psicológico de la nostalgia es sin duda la conciencia de que nuestros recuerdos nos pertenecen y de que el pasado vive continuamente dentro de nosotros.

El aislamiento social depende de varias cosas, como el proceso mental y el proceso físico. Entonces, tratamos de medir el aislamiento de las personas mayores en tres ejes: soledad física, soledad mental y soledad social.

El cálculo de la soledad de cada uno de los tres ejes se realiza reconociendo las acciones en la vida diaria. Se consideran las relaciones entre las acciones típicas de la vida diaria y la soledad física, la soledad mental y la soledad social.

Los indicadores que generalmente muestra un modelo de aislamiento incluyen movilidad, frecuencia de visitas al baño, patrones de sueño, inquietud en la cama, nivel de actividades nocturnas, adherencia a la medicación, actividades de preparación de comidas. Es importante monitorear tres actividades diarias:

1. Bañarse y ducharse (sin incluir cepillarse / peinarse / peinarse, ir al baño, limpiarse y volver a levantarse)
2. Tiempo de movilidad funcional, medido por la frecuencia para caminar, entrar y salir acostarse y sentarse y levantarse de una silla; salir de casa.
3. Actividades sociales: tiempo de entretenimiento (ver televisión o escuchar / tocar música), hablar (teléfono), leer o jugar.

Al detectar cada parámetro, necesitamos identificar diferentes niveles de aislamiento.

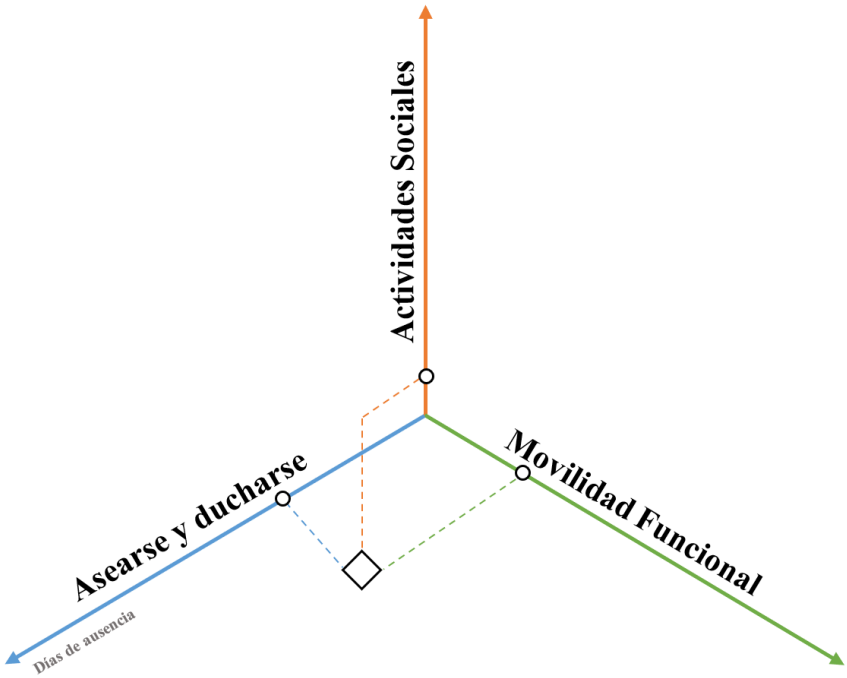

Figura 2. Ejes para la detección de soledad o aislamiento

Analizando la figura tridimensional, es posible captar la tendencia de qué eje el grado de soledad es alto o bajo. Después del mapeo, podemos hacer un plan de cuidados personalizado para reducir la soledad que es diferente persona a persona.

Capítulo 2. Objetivos

El objetivo principal de este proyecto trata sobre la creación de una plataforma para evitar el aislamiento social en las personas mayores, para ello se seguirán una serie de objetivos para conseguir llegar a este objetivo principal:

- Conocer el estado actual de la sociedad, en tanto del envejecimiento de la población., gracias a esto, seremos capaces de identificar los parámetros diferentes parámetros con los cuales podemos detectar el grado de aislamiento de una persona.
- Estudiar las diferentes tecnologías para comunicación entre dispositivos de bajo consumo, entre ellas se encuentran Zigbee o Z-Wave.
- Estudiar las diferentes opciones de sensores comerciales que existen en el mercado, centrándonos en las dos tecnologías mencionadas anteriormente.
- Estudiar los diferentes sensores DIY para poder realizar tus propios sensores permitiéndonos cubrir necesidades más específicas que los sensores comerciales no nos permitirían.
- Estudiar diferentes microcontroladores para diferentes tecnologías, en concreto se estudiará el funcionamiento del módulo ESP8266 y del módulo Z-Uno.
- Estudiar las diferentes plataformas de IoT existentes y seleccionar la más adecuada.
- Estudiar diferentes bases de datos NoSQL, concretamente InfluxDB ya que esta nos permite almacenar y evaluar los datos de los sensores con marcas temporales.
- Desarrollar las actividades con las cuales se detectará el aislamiento.
- Estudiar el lenguaje de programación C para el desarrollo de los sensores.
- Desarrollar e implementar los diferentes sensores que nos permitirán monitorizar las diferentes actividades.
- Investigación formas de conectar la pulsera Miband3 a una Raspberry Pi.
- Identificar las formas de detectar las actividades para la detección del grado de aislamiento.
- Desarrollo e implementacion de la aplicación IoT y base de datos para recopilar toda la información de los sensores.
- Desarrollo de aplicación Android para conocer los valores de las actividades para la detección del grado de aislamiento social.

Capítulo 3. Estado del arte

Este proyecto combinó tecnología con cierta tradición de uso (tales como los sensores), con recientes avances en el campo de las Telecomunicaciones como podrían ser las diferentes plataformas IoT. En este capítulo se exponen aquellas herramientas de la Telecomunicación involucradas en el proyecto, en concreto, se describen dos de las principales tecnologías de comunicación entre sensores existentes en la actualidad (Zigbee y z-wave), así como una variedad de sensores que satisfacen las necesidades del proyecto, y, finalmente, se identificarán diferentes plataformas de domótica donde reunir los datos captados por todos los sensores instalados junto con la bases de datos que pueden ser utilizadas para guardar dicha información.

3.1 Tecnologías de comunicación de sensores

En la actualidad existe una amplia gama de tecnologías que permiten la comunicación entre diferentes sensores. Para la realización de este proyecto nos servimos de una de estas tecnologías para conectar los sensores que se distribuirían por diferentes partes de la casa. Mediante el estudio de las diferencias entre las opciones existentes se llegó a la selección de la tecnología más adecuada para el desarrollo de este proyecto. A continuación, se exponen algunas de las diferentes alternativas barajadas entre las que ofrece el mercado actual

3.1.1 Zigbee

Zigbee es el nombre que de un protocolo inalámbrico habitual en domótica que permite la comunicación de los dispositivos inteligentes sin necesidad de conectarse a la red WiFi, o a internet, ya que se comunican entre ellos a través de ondas de radio de baja energía. [3]

Figura 3. Zigbee

Zigbee funciona a través de una red de malla, una red de malla es un conjugo te dispositivos que emplean un lenguaje común para comunicarse, cada dispositivo actúa como un repetidor en caso necesario para transferir la señal al siguiente dispositivo. Permitiendo así que los dispositivos no tengan conexión directa con la plataforma IoT.

Zigbee se trata de un estándar abierto gestionado por **Zigbee Alliance**, esto permite una gran accesibilidad, pero no asegura la compatibilidad entre todos los dispositivos, como por ejemplo la línea de dispositivos Hue de Philips, el protocolo desde el lanzamiento de Hue ha sido

modificado por Philips para limitar la compatibilidad de su pasarela con el resto de los dispositivos que utilizan el mismo protocolo.

Una de las principales ventajas de Zigbee es el número de saltos que se pueden dar hasta llegar a la pasarela IoT, tantos como sea necesario, sin embargo, el radio de acción de los dispositivos es de 10 a 20 metros, para entornos domésticos es más que suficiente. Se pueden conectar aproximadamente 65.000 dispositivos.

En cuanto a velocidad de transferencia de datos, Zigbee alcanza entre los 40 a 250 kbps, empleando la banda ISM, en Europa concretamente los 868 MHz, también puede usar la banda de 2.4 GHz.

Las marcas como, Amazon, Ikea, LG y Samsung utilizan Zigbee en sus productos.

3.1.2 Z-wave

Z-Wave, al igual que Zigbee es un protocolo de comunicación inalámbrico que emplea ondas de radio de baja energía, tampoco requiere de conexión con la red WiFi para s funcionamiento, lo único necesario sería un controlador para su funcionamiento. Z-wave también funciona a través de una red de malla permitiendo la comunicación entre dispositivos usándose ellos mismos como repetidores, consiguiendo así que todos se comuniquen, aunque no estén dentro del radio de acción. [4]

Figura 4. Z-Wave

Al contrario que Zigbee, z-wave es un estándar cerrado, propiedad de Silicon Labs, la gran ventaja que esto supone es que todos los dispositivos que estén bajo este estándar son compatibles, ya que para poder integrar soporte con z-wave el fabricante debe adquirir una licencia. Además, cada dispositivo viene con un identificador único para que durante una conexión se pueda identificar fácilmente el dispositivo.

Z-wave tiene la desventaja que solo se puede dar 4 saltos para llegar a la pasarela, sin embargo, la distancia que se puede alcanzar son 100 metros con la serie 700. Además, los dispositivos máximos que se pueden conectar a la red son 232 unidades.

La velocidad de transferencia de datos en el caso de Z-wave son de 9.6-100kbps, operando, en Europa, en las bandas 868.40/868.42, pudiendo así minimizar la posibilidad de interferencias.

3.2 Sensores

3.2.1 Sensores de presencia

Un sensor de presencia es un dispositivo que detecta objetos en movimiento, especialmente personas. Un dispositivo de este tipo a menudo se integra como un componente de un sistema que realiza automáticamente una tarea o alerta al usuario del movimiento en un área.

Se pueden colocar sensores de presencia en cada habitación de la casa para poder determinar la ubicación de la persona mayor. Combinando esta información con eventos detectados por los otros sensores colocados en la casa, proporciona inteligencia al sistema para discernir si se está produciendo una acción más compleja.

Fibaro FGMS-001

Figura 5 Fibaro FGMS-001

El sensor de movimiento FIBARO es un sensor universal funcionando con Z-Wave. La característica principal del sensor es la detección de movimiento, sin embargo, también posee otros sensores, como el sensor de temperatura, que nos mide la temperatura ambiental, y un sensor lumínico para saber en qué condiciones lumínicas se encuentra. El sensor dispone de un acelerómetro incorporado con el cual puede llegar a detectar la menor alteración del propio dispositivo. El sensor de movimiento FIBARO funciona con una pilas, está diseñado para ser instalado de una forma fácil y rápida en cualquier tipo de superficie. El indicador LED indica la detección de movimiento, la temperatura ambiental mediante un código de colores, y modo de funcionamiento, además el indicador LED se utiliza para comprobar la conectividad con la red Z-Wave. [5]

Características:

- Compatibilidad con cualquier controlador Z-Wave.
- Admite seguridad con cifrado AES-128.

- Funciona con pilas.
- Infrarrojos pasivo para la detección de movimiento.
- Sensor de temperatura.
- Sensor lumínico.
- Facilidad de instalación en cualquier lugar.
- Detección de manipulación.
- LED integrado que indica movimiento, temperatura.
- Detecta vibraciones.

Centralite 3-Series Micro Motion Sensor

Figura 6 Centralite 3-Series Micro Motion Sensor

El sensor de movimiento Centralite 3-Series se puede utilizar para agregar seguridad y funciones avanzadas de automatización del hogar a un hogar conectado. Las notificaciones se pueden activar cuando hay movimiento en un área determinada y se pueden aplicar para controlar la iluminación y activar alarmas de seguridad basadas en el movimiento. El sensor es bastante estrecho y fino, lo cual permite una instalación discreta. El sensor se puede instalar en las paredes o directamente sobre un mueble. [6]

Características:

- Placa de montaje para sensor fácil de instalar.
- Alcance de detección de 4,5 m.
- Fácil compatibilidad con dispositivos ZigBee HA 1.2 de otros fabricantes.
- Sensor de temperatura incorporado.
- Proceso de unión pull-to-pair.
- Actualizaciones de firmware inalámbricas.

Aeotec multisensor

Figura 7 Multisensor Aeotec

El multisensor Aeotec integra sensores de movimiento, temperatura, luz, humedad, vibración y sensor de luminosidad en el mismo dispositivo. [7]

Características:

- Z-Wave (Gen5)
- Admite cifrado AES-128
- Detección de manipulación (a través del sensor de vibración)
- Alcance de comunicación de 150 m
- Rango de medición de luz: 0-1000 lux
- Rango de detección de movimiento: 5 m, 120 grados

3.2.2 Sensores de proximidad

Un sensor de proximidad es un sensor capaz de detectar la presencia de objetos cercanos sin ningún contacto físico dentro del radio de acción del sensor. La detección se realiza mediante la emisión de un campo electromagnético o un haz de radiación electromagnética (infrarrojos) y detecta cambios en el campo o en la señal de retorno. Los diferentes objetivos de los sensores de proximidad exigen sensores diferentes.

Pueden colocarse junto a un objeto y utilizarse para determinar si el objeto se está utilizando o no. Por ejemplo, colocar un sensor de proximidad junto al sofá nos permitirá saber si en algún momento el sofá está siendo utilizado por una persona. De la misma manera, se pueden utilizar para determinar si se está utilizando el fregadero.

Sensor de distancia IR GP2Y0A21YK0F

Figura 8 Sensor de distancia IR

Este sensor de distancia SHARP rebota con infrarrojos en los objetos para determinar qué tan lejos están. Devuelve un voltaje analógico que se puede usar para determinar qué tan cerca está el objeto más cercano. Viene con un cable de interfaz 3-JST de 12 "de largo. Estos sensores son buenos para la detección de corto alcance (menos de 1 m). [8]

Para usar, conecte el cable negro a tierra, el cable rojo a 5V y el cable blanco a la entrada analógica. El voltaje de salida analógico variará entre 3 V cuando un objeto está a solo 4 "(10 cm) de distancia y 0,4 V cuando el objeto está a 32" (80 cm) de distancia

Características:

- Precisión de detección a 80 cm: ± 10 cm
- Alcance: 20 a 150 cm

Sensor de distancia ultrasónico HC-SR04

Figura 9 Sensor de distancia ultrasónico HC-SR04

El sensor de distancia HC-SR04 es un sensor que mide el tiempo que tarda una señal de sonido alcanzar un obstáculo y volver, permitiendo saber la distancia a la que se encuentra dicho obstáculo. El módulo incluye el transmisor ultrasónico, el receptor y la circuitería básica para el funcionamiento. [9]

Características:

- Funciona a 5 voltios
- Ángulo del sensor menor de 15 grados
- Distancias de detección entre 2 centímetros y los 4 metros con precisiones de hasta 3 milímetros

3.2.3 Sensores de apertura de puerta

Actualmente, existen muchos sensores magnéticos en el mercado que detectan cuando una puerta o ventana se abre. El funcionamiento es sencillo, el sensor consta de dos componentes, un imán y un sensor, el imán crea un campo magnético que es detectado por el sensor cuando la puerta está cerrada, cuando la puerta está abierta este campo no puede ser detectado.

Se puede utilizar un sensor de apertura de puerta para activar un algoritmo de procesamiento capaz de determinar cuándo una persona entra o sale de la casa. De esta forma, el sistema podrá inferir cuando la persona mayor abandona la casa o tiene una visita.

Fibaro FGBHDW-002

Figura 10 Fibaro FGBHDW-002

El sensor de puerta / ventana FIBARO es un sensor de apertura que utiliza tecnología inalámbrica Bluetooth de baja potencia. Además de la detección del estado de apertura o cierre, incluye un accesorio capaz de medir la temperatura ambiental. La detección de apertura se efectúa separando el cuerpo del sensor y el imán. Incluye en sensor que detecta intentos de manipulación del sensor. [10]

Características:

- Detección de apertura / cierre
- Detección de manipulación
- Medición de temperatura
- Medición del nivel de la batería

Centralite 3-Series Micro Door Sensor – 3323

Figura 11 Centralite 3-Series Micro Door Sensor

Al detectar la apertura y el cierre de puertas y ventanas, el sensor puede notificar de forma inalámbrica a los usuarios las llegadas y salidas. El sensor consta de una base, que ha de ser montada en el marco de la puerta y un imán que ha de ser colocado justo al lado de la base pero en la puerta, quedándose estos juntos cuando la puerta este cerrada. El sensor es una forma asequible de agregar automatización, seguridad y detección de ocupación a los sistemas domésticos conectados existentes. [11]

Características:

- Sensor de efecto Hall para mayor confiabilidad, batería de larga duración y mayor sensibilidad.
- Distancia del imán de hasta 1 pulgada.
- Placa de montaje fácil de instalar para sensor e imán.
- Sensor de temperatura incorporado
- Fácil compatibilidad con dispositivos ZigBee HA 1.2 de otros fabricantes.
- Proceso de unión pull-to-pair.
- Actualizaciones de firmware inalámbricas

3.2.4 Sensores lumínicos

Un sensor de luz puede detectar la intensidad de la luz ambiental. Los principales tipos de sensores de luz son los fotodiodos, que son capaces de convertir la luz en corriente o voltaje, fotorresistores, que son resistencias cuya resistencia es proporcional a la intensidad de la luz, y fototransistores, que controlan el flujo de corriente en un circuito en función del nivel de luz.

Los sensores de luz se pueden usar para estimar el nivel de luminancia en ciertas habitaciones, que se pueden combinar con datos de otros sensores para obtener una estimación de la actividad.

Sensor lumínico BH1750

Figura 12 Sensor lumínico BH1750

El BH1750 mide la intensidad del sensor de luz directamente desde su luxómetro, lo que reduce el tiempo y simplifica los cálculos. [12]

Esta es una placa de conexión de sensor de intensidad de luz BH1750 con un convertidor AD de 16 bits incorporado que puede emitir directamente una señal digital. Los datos emitidos por este sensor se emiten directamente en Lux (Lx).

3.2.5 Sensores de detección de sonido

Un detector de sonido combina un micrófono y algunos circuitos de procesamiento. Proporciona una indicación binaria de la presencia de sonido y también una representación analógica de la amplitud del sonido.

Se pueden colocar detectores de sonido en determinadas habitaciones para obtener una estimación de la actividad social (conversaciones, uso de TV y radio, etc.) a través de la intensidad, duración, etc. de los sonidos detectados.

Detector de sonido SparkFun

Figura 13 Detector de sonido SparkFun

El detector de sonido SparkFun es una placa que consta de un pequeño micrófono para la detección de audio, de uso muy sencillo y con tres salidas diferentes. El detector de sonido proporciona una salida de audio, una indicación digital de la presencia o ausencia de sonido y una representación analógica de su amplitud del propio sonido. Todas las salidas son simultáneas y completamente independientes. [13]

Características:

- Fácil de usar.
- Diferentes salidas para diferentes datos

Sensor de Sonido Grove

Figura 14 Sensor de Sonido Grove

El sensor de sonido Grove puede detectar la intensidad del sonido del entorno. El componente principal del módulo es un micrófono simple, que se basa en el amplificador LM386 y un micrófono electret. La salida de este módulo es analógica y un Seeeduino puede muestrear y probar fácilmente. [14]

Características:

- Fácil de usar
- Proporciona una señal de salida analógica.
- Se integra fácilmente con módulos lógicos en el lado de entrada de los circuitos Grove

3.2.6 Sensores de lluvia

Los sensores de lluvia son típicamente placas con circuitos impresos que al ser mojados cambian el valor de sus resistencia haciendo medible la cantidad de agua que cae sobre ellos

El sensor de lluvia se puede colocar en lugares como la ducha para monitorizar el uso de esta.

Sensor de lluvia YL83

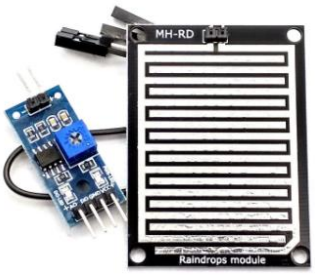

Figura 15 Sensor de lluvia YL83

El sensor de agua YL83 para Arduino es un sensor que se utiliza para detectar cantidad de agua que cae sobre la placa, comúnmente se utiliza para detectar si esta lloviendo. El sensor consta de una placa con un circuito impreso, que al caer agua sobre él se cierran caminos haciendo que su resistencia varíe, y un controlador, estas están separadas para mayor seguridad y comodidad, ya que el controlador es sensible al agua podría estropearse si se moja.

El controlador dispone de un LED que indica si esta encendido y un potenciómetro para poder ajustar el nivel de sensibilidad que nos proporcionará la salida del sensor. [15]

3.2.7 Sensor de Presión

Los sensores de presión permiten saber el peso de un objeto que se les pone encima, suelen constar de una membrana que varía su resistencia en función de la presión ejercida sobre ella.

Pueden ser útiles a la hora de detectar cuando un asiento está siendo utilizado, por ejemplo.

Sensor de presión MD30-60

El sensor de presión MD30-60 detecta cambio en el peso que se le aplica encima la cambiar el valor de la resistencia del propio sensor, permitiendo detectar cambios de presión entre 0-30 kg. [16]

Características:

- Sensor ultrafino e impermeable
- Alta precisión
- Rango: 0 – 30 kg

Figura 16 Sensor de presión MD30-60

3.2.8 Microcontroladores

Muchos de los sensores previamente descritos necesitan de un controlador para poder comunicarse con el resto de los sensores o la misma pasarela o plataforma. Para ello se estudiarán dos de ellos, el microcontrolador que funciona bajo la frecuencia de WiFi, 2,4 GHz, y otro que funciona para los dispositivos z-wave

ESP8266

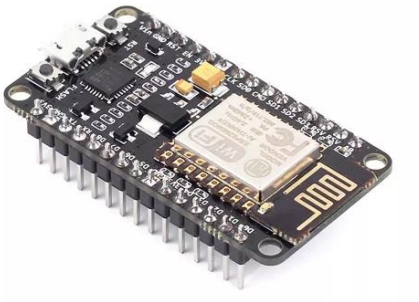

Figura 17 Modulo NodeMCU microcontrolador ESP8266

El microcontrolador ESP8266 es un chip de bajo costo WiFi que implementa una pila TCP/IP completa. Se alimenta con 3.3 voltios y dispone de un procesador de 80Mhz, 64KB de memoria RAM para las instrucciones y 96 KB para datos y 16 pines GPIO. Es compatible con los estándares WiFi IEEE 802.11 b/g/n y además soporta seguridad WEP, WPA y WPA2. [17]

Se suele usar para añadir conectividad WiFi, permitiendo, por ejemplo, domotizar una casa mediante la interconexión de los sensores y actuadores.

Usando el módulo NodeMCU el uso se simplifica mucho, ya que contiene todo lo necesario para funcionar sin añadir ningún modulo, dispone de un puerto microUSB que permite cargar el código. Incluye firmware que permite programar con diferentes lenguajes como por ejemplo Python o JavaScript.

Z-Uno

Figura 18 Z-Uno

Z-Uno es el microcontrolador que nos permite conectar diferentes dispositivos mediante el protocolo Z-Wave sin ser necesario tener un conocimiento avanzado del funcionamiento del protocolo. [18]

Z-Uno se programa en lenguaje C simplificado, y la programación se carga en Z-Uno utilizando el programa IDE de Arduino. Además, es compatible con otros controladores Z-Wave.

3.3 Plataformas IoT

3.3.1 OpenHAB

El **Open Home Automation Bus** (openHAB) se trata de una plataforma domótica de código abierto que funciona como elemento central de un hogar inteligente. Como principales ventajas podríamos destacar: [19]

- Capacidad de integración de una gran variedad de dispositivos. En openHAB se integran tanto sistemas de automatización, dispositivos inteligentes y otras tecnologías en una solución única.
- Interfaz de usuario clara y sencilla que permite que las reglas de automatización se uniformes en todo el sistema, sin importar la variedad de fabricantes y subsistemas que se vean involucrados.
- Herramienta flexible que permite implementar con facilidad cualquier solución domótica.

openHAB se comunica electrónicamente con dispositivos inteligentes, y dispositivos que no lo son tantos, es capaz de realizar acciones predefinidas y proporcionar diferentes páginas web

con contenido previamente definidos por el usuario, así como herramientas para interactuar con todos los dispositivos. Para llevar esto a cabo, openHAB segmenta y compartimenta algunas funciones y operaciones. La siguiente tabla ofrece una descripción de nivel superior de los conceptos más importantes, así como un enlace a más información:

CONCEPTO	SIGNIFICADO
BINDINGS (ENLACES)	Son el componente openHAB que proporciona las herramientas necesarias para interactuar con los dispositivos.
THINGS (OBJETOS)	Son las entidades físicas que se pueden añadir a openHAB.
CHANNELS (CANALES)	Son la conexión openHAB entre "Things" y "Items".
ITEMS (ITEMS)	Son la representación generada por openHAB de la información sobre los dispositivos
RULES (REGLAS)	Son necesidades que se tiene que cumplir para que se realicen las acciones automáticas definidas.
SITEMAP	Es la interfaz de usuario (sitio web) generada por openHAB que presenta información y permite interacciones

Tabla 1 Conceptos básicos openHAB

3.3.1.1 Enlaces

Los enlaces (“*bindings*”) son paquetes de software que instala el usuario en openHAB. El propósito principal de los enlaces es establecer la conexión entre el dispositivo y el objeto. Los enlaces se comunican con el dispositivo y traducen todos los comandos hacia y desde openHAB entre el dispositivo y el objeto.

Los enlaces se proporcionan en la sección Add-on del sitio web. Donde se encuentra una lista de búsqueda de varios cientos de enlaces para admitir tantos dispositivos como sea posible. Regularmente se agregan nuevos enlaces a medida que los desarrolladores integran más dispositivos en openHAB.

Para cada enlace, se proporcionan instrucciones detalladas y ejemplos que incluyen orientación sobre la configuración (si existe) del enlace en sí, la definición de los elementos admitidos por este enlace y los canales que estos elementos proporcionan. En gran parte de los casos, la descripción también contiene un ejemplo completamente elaborado que incluye una definición de objeto y sus canales, elementos vinculados a esos canales y el uso de estos elementos en un Sitemap.

3.3.1.2 Canales

Los canales ("*channels*") son el vínculo lógico entre un objeto y un ítem. Los canales se originan a partir de la definición de objeto y definen cómo su cosa puede comunicarse con el ítem (y viceversa). Se crearán canales al definir el objeto.

Durante la definición del objeto, se identificará el canal al que se vinculará el ítem. Estos dos pasos garantizan que openHAB pueda transmitir la información del objeto al ítem (y viceversa).

3.3.1.3 Objetos

Los objetos son fuentes de datos que se agregan al sistema de openHAB y que pueden proporcionar una amplia gama de funcionalidades. Es importante tener en cuenta que los objetos no necesariamente son dispositivos, sino también servicios web o cualquier otra fuente manejable de información. Desde la perspectiva del usuario, son relevantes durante el proceso de instalación y configuración, pero no durante la operación.

Los objetos pueden tener propiedades de configuración, opcionales u obligatorias. Dichas propiedades pueden ser información básica o una configuración específica del dispositivo que altera su funcionamiento.

3.3.1.4 Ítems

OpenHAB separa de una forma muy estricta el mundo físico de la capa virtual, los ítems son la representación virtual, dentro de openHAB, de un objeto. Los ítems tienen un estado y se utilizan a través de eventos.

3.3.1.5 Reglas

Las reglas se utilizan para automatizar procesos, cuando una regla se activa, esta a su vez ejecuta scripts para la realización de cualquier tipo de tarea, por ejemplo, encienda las luces modificando sus elementos, haga cálculos matemáticos, inicie temporizadores, etc.

OpenHAB tiene un motor de reglas altamente integrado, liviano pero poderoso incluido. En esta página, aprenderá cómo aprovechar su funcionalidad para hacer una verdadera automatización del hogar.

3.3.2 HomeAssistant

Home Assistant es una plataforma domótica para el hogar con capacidad para integrar una gran diversidad de dispositivos y servicios, ya sean estos comerciales, de terceros, o desarrollados por uno mismo. Se trata de software de código abierto, con una base de usuarios enorme, además cuenta con una curva de aprendizaje muy asequible. [20]

Es un software desarrollado en Python, compatible con una gran variedad de sistemas operativos y de dispositivos. Esta distribuido de muchas formas, lo cual lo hace perfecto para los usuarios nuevos sin dejar de lado a los usuarios más avanzados y con experiencia.

Existen varios tipos de instalación de Home Assistant:

3.3.2.1 Hass.io

Hass.io es un todo en uno que incluye una distribución de Linux optimizada para la ejecución de Home Assistant sobre contenedores Docker y un supervisor de estos. Esta instalación está hecha a medida para que sea instalar y funcionar, acompañado de un sistema de Add-on muy completo, sin embargo, no sería posible acceder al propio sistema operativo.

3.3.2.2 Home Assistant Core

Home Assistant Core consta del software en sí, pudiéndose instalar en cualquier máquina, ya sea sobre Windows o sobre una distribución de Linux. Esta opción se instalaría como aplicación y no sería posible acceder al sistema de Add-on que nos permite el primer método, pero a cambio tenemos control total sobre el sistema operativo.

3.3.2.3 Home Assistant Supervised

Home Assistant Supervised sería una opción mixta entre las dos anteriores permitiendo tener un control total del sistema operativo y teniendo la aplicación funcionando sobre contenedores Docker, con el añadido de los Add-on. El uso de este método requiere de conocimientos avanzados para evitar problemas.

3.3.3 FIWARE

FIWARE es una plataforma abierta que define un conjunto universal de estándares para la gestión de datos de contexto que facilitan el desarrollo de soluciones inteligentes para diferentes

dominios como Smart Cities, Smart Industry, Smart Agrifood y Smart Energy. La plataforma proporciona especificaciones de API públicas y protocolos interoperables para la creación de nuevos servicios y aplicaciones de internet.

FIWARE permite portar los dates para diferentes aplicaciones, también aporta un conjunto de habilitadores genéricos que permiten acceder a información relevante mediante la API NGSI, lo que viene a significar que habilita una forma rápida y eficiente de recolección, publicación, intercambio y proceso de grandes volúmenes de datos.

Además, FIWARE, también proporciona soporte para una infraestructura de hosting en la nube mediante habilitadores genéricos, accesibles mediante REST APIs. Esta arquitectura en la nube incluye servicios como IaaS (Infracstructure as a Service), servicios de almacenamiento de datos y metadatos en la nube, y servicios de gestión y monitorización de aplicaciones y dispositivos.

El habilitador genérico de procesado de datos se divide en dos bloques funcionales. Como primera parte está el bloque de procesado por lotes, este aporta infraestructuras bajo demanda que soportan herramientas de análisis de Big Data. Y en la otra parte está el bloque de procesado de flujos, este proporciona una interfaz modular para realizar el procesado de los datos en tiempo real, utilizando Apache Storm.

En cualquier solución inteligente existe la necesidad de recopilar y gestionar la información del contexto, procesando esa información e informando a los actores externos, permitiéndoles actuar y por lo tanto alterar o enriquecer el contexto actual. El componente FIWARE Orion Context Broker es el componente central de cualquier plataforma "Powered by FIWARE". Permite al sistema realizar actualizaciones y acceder al estado actual del contexto.

3.3.4 FIWARE Orion Context Broker

El Context Broker es el que se encarga de la gestión de la comunicación entre las aplicaciones y los dispositivos con el fin de separar las aplicaciones IoT de las instalaciones de los dispositivos. Para la comunicación entre los dispositivos y las aplicaciones, el Context Broker utiliza el protocolo NGSIv2, lo cual permite generar y recibir notificaciones de eventos contextuales. [21]

Las entidades (*entities*) son las unidades destinadas a almacenar los datos, cada una se identifica mediante un identificador único (*id*) y pertenece a un tipo (*type*), varias entidades pueden formar parte a un mismo tipo. Las entidades, además, pueden tener atributos (atributes), estos atributos describen la información que contiene la entidad mediante un valor (*value*) y un

tipo de datos (*type*), opcionalmente pueden contener metadatos (*metadata*) introducidos por el usuario.

Cabe destacar que valor que se guarda es el último valor de cada atributo de las entidades. Los datos se almacenan en un base de datos NoSQL, como podría ser MongoDB.

Estas entidades pueden ser clasificados según un sistema jerárquico, siendo el nivel principal un *Fiware-Service*, pudiendo contener niveles inferiores, formando un árbol, el camino hacia los cuales se llama *Fiware-ServicePath*.

Se pueden diferenciar dos roles para Orion, el rol de productor, los cuales se encargan de la creación y actualización de los datos de las entidades, y el rol de los consumidores, los cuales se suscriben a Orion para recibir notificaciones de los datos de las entidades a los que, previamente, se han suscrito.

3.4 Base de datos

3.4.1 MongoDB

MongoDB es una base de datos gratuita y de código abierto de tipo NoSQL, orientada a documentos. El almacenamiento de datos en MongoDB es mediante el uso de documentos dinámicos en formato BSON, gracias a esto la estructura de datos puede variar con el tiempo y entre diferentes documentos. [22]

Aunque la consola de MongoDB este escrita en JavaScript, esto permite al usuario una fácil gestión de la base de datos mediante funciones en dicho lenguaje, existen múltiples librerías de clientes en muchos lenguajes de programación como Python, Java, C++, etc.

3.4.2 InfluxDB

InfluxDB es una base de datos gratuita y de código abierto enfocada a series temporales, esto permite que se pueda usar de manera totalmente gratuita. InfluxData.Inc, la empresa desarrolladora, también pone a nuestro alcance una versión comercial, esta nos ofrece mantenimiento y controles especiales para los que adquieran la versión comercial, esta versión se instalaría en los servidores de la empresa. [23]

Flux es el lenguaje usado en InfluxDB, se trata de un lenguaje de scripts y consultas para bases de datos de series temporales, su sintaxis está basada en el lenguaje de programación JavaScript aportando facilidades a la hora de su aprendizaje además de mantener la misma flexibilidad. Como característica principal a la hora de la utilización de Flux esta la compatibilidad entre fuentes de datos diversas.

InfluxDB está pensada para ser usado como bases de datos de series temporales, estas bases de datos se usan, principalmente, para almacenar la información que proviene de sensores ya que estos disponen de una marca temporal junto con el valor medido.

Al estar las bases de datos pensadas para series temporales estas bases suelen ser muy compactas, y bastaría con tablas de tres columnas, estas columnas serian: nombre del sensor, valor y la marca temporal.

Principales ventajas:

- Bases de datos muy rápidas, mucho más que las relacionales, ya que almacenan y procesan los datos con marcas temporales con mayor agilidad.
- Toda la gestión de la base de datos se concentra en un archivo del programa, agilizando el proceso de gestión.

Capítulo 4. Desarrollo

Como objetivo fundamental de este proyecto está la monitorización de las actividades que la persona hace a lo largo del día, para ello, se despliegan sensores en los lugares estratégicos de la residencia, una vez recolectados los datos y mediante la definición de unos umbrales de frecuencia se intentará que la sensación de soledad o aislamiento sea la menor posible. Para ello, y teniendo en cuanta los tres ejes de los que se hablan en la introducción de este documente podemos llegar a definir actividades y umbrales para la detección de aislamiento o soledad de una persona.

Para detectar el aislamiento de las personas mayores, EBASI monitoreará básicamente cuatro comportamientos principales de las personas mayores, que pueden mostrar un estado emotivo:

- Bañarse y ducharse
- Tiempo de movilidad funcional, medido por la frecuencia para caminar, acostarse y levantarse de la cama y sentarse y levantarse de una silla; salir de casa. etc.
- Silencio y habla, incluido el tiempo de silencio durante las actividades diarias, combinado con la posición.
- Actividades sociales: tiempo de entretenimiento hablar, leer, recibir visitas de otras personas en casa, tiempo de iluminación.

Este comportamiento se monitorizará en la propia vivienda, concretamente en tres espacios: el baño, la sala de estar y el dormitorio.

En la siguiente tabla se muestra unos umbrales de ejemplo para la monitorización de actividades:

Bañarse			
Frecuencia		Nivel	Respuesta
>= 3 veces por semana		0	Ninguna
2 veces por semana		1	Interacción
< 2 veces por semana		2	Alerta
Movilidad Funcional			
Actividad	Medidas	Nivel	Respuesta
Dormir	Hasta 6 – 9 horas al día	0	Ninguna

	Entre 5 – 6 / 9 – 10 horas al día	1	Interacción
	< 5 / >10 horas al día	2	Alerta
Estar sentado sin hacer nada	< 2 horas al día	0	Ninguna
	2 – 3 horas al día	1	Interacción
	> 3 horas al día	2	Alerta
Salir	< 1 hora cada día	0	Ninguna
	< 1 hora cada dos días	1	Interacción
	< 1 hora cada cuatro días	2	Alerta
Actividades Sociales			
Actividad	Medidas	Nivel	Respuesta
Jugar o leer	Al menos una vez cada dos días	0	Ninguna
	Al menos una vez a la semana	1	Interacción
	Al menos una vez cada dos semanas	2	Alerta
Ver la tele, escuchar música …	< 1 hora al día	0	Ninguna
	< 1 hora cada tres días	1	Interacción
	< 1 hora a la semana	2	Alerta
Hablar con alguien	> 3 veces a la semana	0	Ninguna
	1 – 3 veces por semana	1	Interacción
	< 1 vez por semana	2	Alerta

Tabla 2. Umbrales y actividades por defecto

Por ejemplo, el acto de hablar con los demás a la hora de comer, esa acción se considera beneficiosa para reducir la soledad física y el aislamiento social, por lo que se les dan los buenos puntos a estos dos ejes (Figura 1). Cuanto mayor sea la puntuación de cada eje, menor será la sensación de soledad que puedan sentir las personas mayores. Definir una tabla que combine tales acciones y puntos para actividades típicas de la vida diaria.

Los tipos de interacción serán principalmente 3:

1. Seguimiento y Detección: posición, comportamiento y emociones, actitudes, detección de aislamiento. La función más básica es monitorear los diversos estados de los ancianos, como la posición y la actividad.
2. Pronóstico: También es deseable pronosticar el cambio de estado. Para la vida diaria, la predicción del comportamiento y las actividades normales es importante para un cuidador.
3. Interacción y alerta: si la persona se encuentra en un estado crítico, el sistema debe reconocer el estado y poner la alerta dentro de un tiempo especificado. El sistema creará una alerta para los cuidadores. Es necesario estimular la implicación activa del usuario, animándolo a estar activo, pero también a realizar actividades físicas si, por ejemplo, el sistema percibe un período prolongado frente al sofá.

4.1 Arquitectura general

Para la monitorización se colocarán diferentes tipos de sensores inalámbricos en las tres áreas de la casa: (baño, salón y dormitorio). Los sensores se conectarán a un dispositivo Raspberry Pi que gestionará la interfaz del sensor y la gestión de datos. Si uno de los parámetros de actividad social o movilidad funcional supera el umbral, el sistema alertará al cuidador mediante una notificación para que realice las acciones pertinentes.

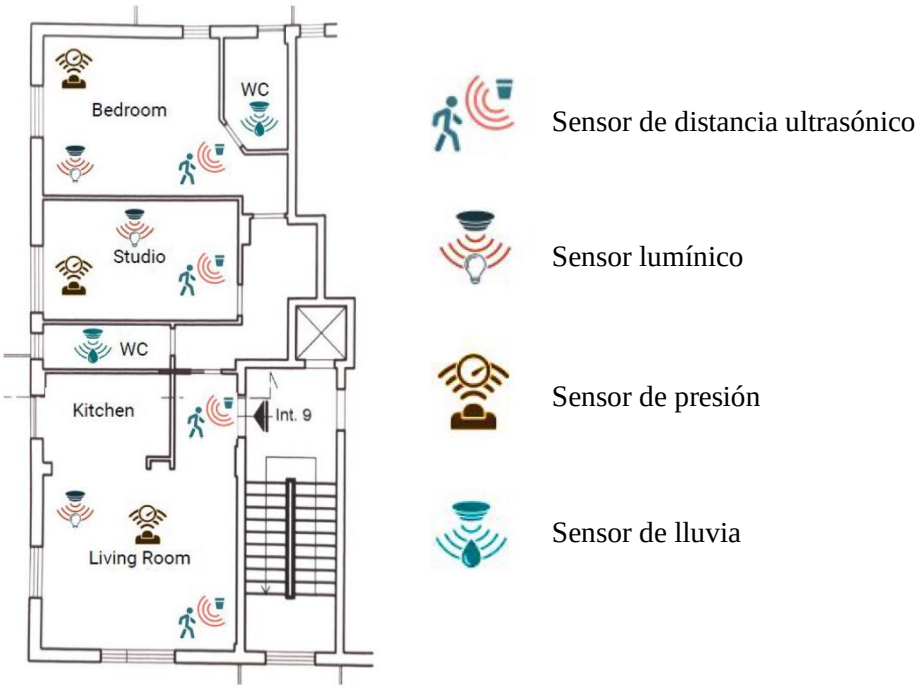

Figura 19. Ejemplo de disposición de sensores

4.2 Sensores

En cuanto a los sensores utilizados para detectar las actividades de la persona se ha decidió por el uso de 4 sensores en concreto más el uso de una pulsera inteligente, miband3 de Xiaomi, para medir actividades como los paseos o el pulso cardiaco.

Para conectar los sensores se requiere de un controlador, en este caso al estar utilizando el protocolo z-wave, se utilizará el microcontrolador z-uno, como se ha descrito en el capítulo tres, este nos permite conectar de forma fácil con el resto de los dispositivos z-wave, ocupándonos de tan solo la parte de adquisición de datos de los sensores, ya que de la parte de conexión entre de dispositivos es completamente transparente en cuanto a la programación del controlador.

La arquitectura de los sensores seria idéntica en los 4 tipos, siendo esta la que se puede observar en la siguiente figura, el sensor conectado al microcontrolador Z-Uno y este siendo alimentado por una batería portátil de las que sirven para cargar por ejemplo los móviles.

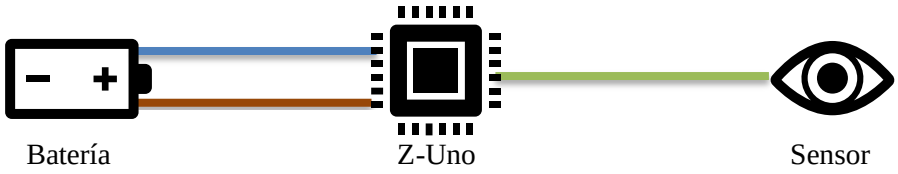

Figura 20 Esquema conexión de sensores

4.2.1 Sensor de distancia

Para medir distancias se utilizará el sensor ultrasónico **HC-SR04**, este sensor se utiliza principalmente para detectar si la persona está haciendo como: ir al baño, salir de casa…, en cuanto a la conexión del sensor con el microcontrolador, el sensor dispone de 4 pines, uno para la fuente, dos para los sensores, el emisor y el receptor, y finalmente la masa. La fuente se conecta a alimentación de 5 voltios del microcontrolador, los sensores se conectan a dos pines de salida digital del microcontrolador y la masa a la masa del microcontrolador. En la siguiente figura se puede observar la conexión el sensor de distancia con un microcontrolador NodeMCU ESP8266, para la conexión con el Z-Uno seria idéntico.

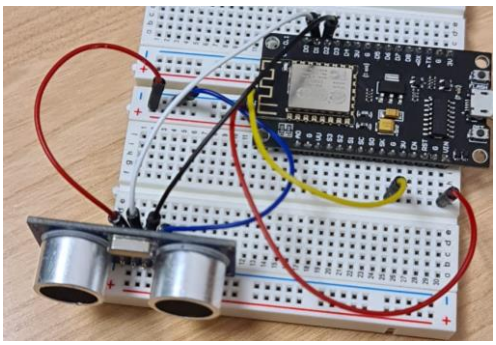

Figura 21 HC-SR04 conectado a microcontrolador

```
//////////////////distance sensor//////////////////
int tmp;

// trigger measurement
digitalWrite(triggerPin, LOW);
delayMicroseconds(10);
digitalWrite(triggerPin, HIGH);
delayMicroseconds(10);
digitalWrite(triggerPin, LOW);

// read pulse width
tmp = pulseIn(readPin, HIGH, 100000);
if (tmp != 0) {
  lastValue = tmp / 58; // convert to cm, see datasheet
  Serial.println(lastValue);

  // send On/Off to control group
  if (lastValue < turn_on_distance_cm && controlState == 0) {
    zunoSendToGroupSetValueCommand(CTRL_GROUP_1, 255);
    controlState = 255;
  } else if (lastValue >= turn_on_distance_cm && controlState == 255) {
    zunoSendToGroupSetValueCommand(CTRL_GROUP_1, 0);
    controlState = 0;
  }

  // send report to controller (Life Line group)
  zunoSendReport(1);
}
```

Figura 22 Código Sensor distancia

4.2.2 Sensor lumínico

En cuanto al sensor lumínico se usará el sensor **BH1750**, para detectar la ausencia o presencia de luz, indicando cuantas horas, incluidas las horas diurnas se mantiene la habitación en estado de penumbra. La conexión con el microcontrolador seria casi idéntica a la del sensor de distancia, solo que los sensores irían conectados a los pines analógicos del microcontrolador. En la siguiente figura se puede ver un ejemplo de código para la recepción de nivel de luz.

```
void setup(){
  Serial.begin(9600);
  Serial.println("Inicializando sensor...");
  Luxometro.begin(BH1750_CONTINUOUS_HIGH_RES_MODE); //inicializamos el sensor
}


void loop() {
  uint16_t lux = Luxometro.readLightLevel();//Realizamos una lectura del sensor
  Serial.print("Luz(iluminancia):  ");
  Serial.print(lux);
  Serial.println(" lx");
  delay(500);
}
```

Figura 23 Ejemplo código sensor lumínico

4.2.3 Sensor de presión

Para la detección de uso de la cama o asientos tipo silla, sofá o sillón se utilizará el sensor de presión **MD30-60**, como su funcionamiento es como una resistencia simplemente se conecta a 5 o 3.3 voltios, funciona perfectamente en ambos voltajes, y otro pin a masa, añadiendo una resistencia de 10kΩ entre la conexión entre un pin analógico y masa. En la siguiente figura se puede observar un esquema de conexión del sensor con un Arduino, para hacerlo con Z-Uno es idéntico.

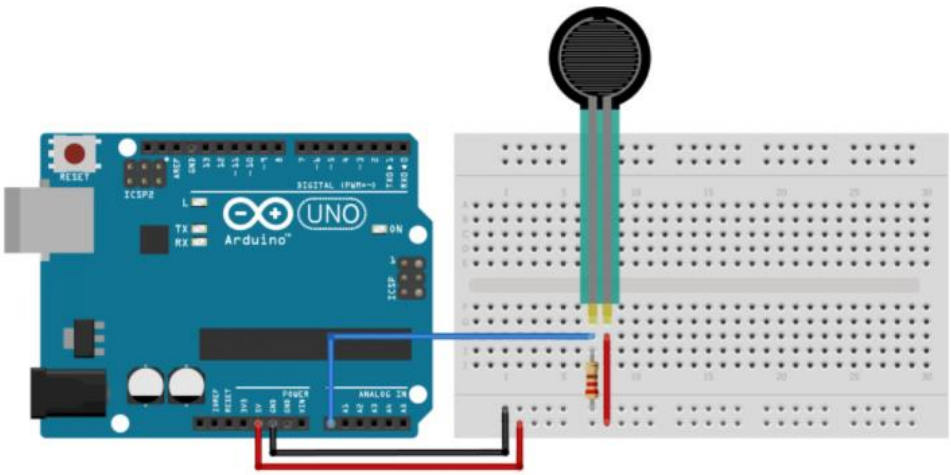

Figura 24 Esquema de conexión sensor de presión con Arduino

4.2.4 Sensor de lluvia

El sensor de lluvia elegido es el descrito anteriormente, YL83. El agua creará nuevos caminos en el circuito impreso del sensor haciendo que la resistencia varíe en función de cantidad de agua, la placa del sensor estará actuando como una resistencia variable que variando de los 100k ohmios cuando está totalmente empapada en agua a 2M ohmios cuando no hay agua en la superficie del sensor. En resumen, cuanto más húmedo esté el tablero, más corriente se conducirá.

```
const int digitalPin =5;
const int analogPin =0

void setup(){
    Serial.begin(9600);
    pinMode(digitalPin, OUTPUT);
}
void loop(){
    if(digitalRead(digitalPin) == HIGH) Serial.println("No se detecta agua");
    else Serial.println("agua detectada");
}
```

Figura 25 Ejemplo de código para detección agua

4.2.5 Sensor de sonido

El sensor elegido es el sensor de sonido **Sparkfun**, el cual nos permite saber si el ruido ambiental supera cierto umbral, lo cual nos permite saber junto con los demás sensores la actividad en los alrededores del sensor de sonido.

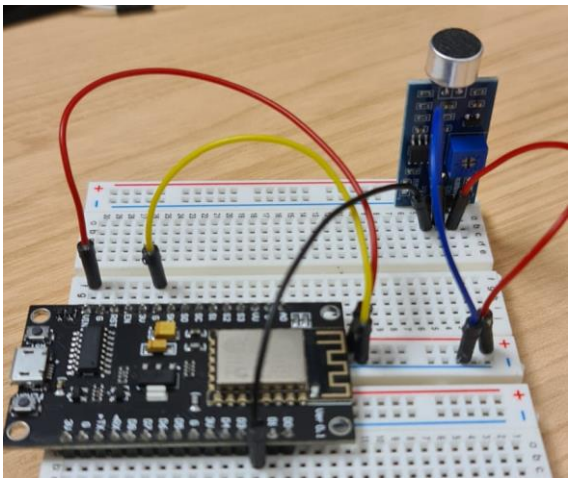

Figura 26 Sensor de sonido conectado a microcontrolador

4.2.6 Miband3

La Miband3 de Xiaomi se usará para recopilar los datos de ritmo cardiaco y pasos, permitiendo saber en mayor o menor medida el ejercicio realizado al finalizar el día. En este punto es donde existe una complicación, la pulsera inteligente se conecta un smartphone por vía de la aplicación oficial de Xiaomi, Mi Fit, pero, muchas de las personas mayores a las cuales está enfocado este proyecto no poseen smartphone y tampoco queremos que como requisito para el funcionamiento sea la posesión de un smartphone, por tanto, lo ideal es que la pulsera se conectará directamente a la Raspberry Pi, cosa que en un principio, no es posible ya que Xiaomi no presenta esa posibilidad.

Sin embargo, existen estudios anteriores que demuestran y consiguen conectar una pulsera MiBand2 a una Raspberry Pi, haciendo uso de Wireshark capturando los paquetes intercambiados por la aplicación y la pulsera inteligente.

Para conectar la Raspberry Pi y la pulsera inteligente hay que seguir los siguientes pasos:

Primero que nada, hay que saber la dirección de MAC de la pulsera inteligente, esto lo podemos saber emparejando normalmente la pulsera con la aplicación, en la información de la pulsera dentro de la aplicación aparece indicada la dirección MAC.

Una vez sabemos la dirección MAC de la pulsera lo que necesitamos es saber cómo se emparejan el smartphone y la pulsera, para ella debemos habilitar las opciones de desarrollador en el smartphone, para hacer esto basta con ir a “ajustes>sistema>acerca del teléfono” y hacer clic 7 veces sobre “Número de compilación”, una vez hecho esto entramos en las opciones de desarrollador y activamos la opción “Habilitar registro de Bluetooth HCI”. Esta opción lo que hace es analizar todas las comunicaciones realizadas por el smartphone con el resto de los dispositivos bluetooth.

Tras activar la anterior opción pasaremos a realizar el emparejamiento de la pulsera inteligente, hay que desemparejarla si lo estaba, para poder observar los UUID de los servicios que nuestro smartphone envía para conectarse con la pulsera.

Una vez realizado el emparejamiento desconectamos el bluetooth del smartphone y extraemos el registro creado, este se encuentra en /mtklog/blog/btsnoop_hci.log, y lo abrimos con el Wireshark para ver todos los paquetes intercambiados.

En la siguiente figura se observa el código, realizado en Python, mediante el cual podemos realizar la autenticación. El código, lo que hace, es notificar a la pulsera que se va a realizar la autenticación, tras esto se envía una clave de 16 bytes, tras esto se pide una clave aleatoria y tras

la recepción se encripta con nuestra clave de 16 bytes y se envía a la pulsera, si todo se ha realizado correctamente la Raspberry Pi y la pulsera estarán conectadas.

```
class AuthenticationDelegate(DefaultDelegate):

    def __init__(self, device):
        DefaultDelegate.__init__(self)
        self.device = device

    def handleNotification(self, hnd, data):
        if hnd == self.device._char_auth.getHandle():
            if data[:3] == b'\x10\x01\x01':
                self.device._req_rdn()
            elif data[:3] == b'\x10\x01\x04':
                self.device.state = AUTH_STATES.KEY_SENDING_FAILED
            elif data[:3] == b'\x10\x02\x01':
                # 16 bytes
                random_nr = data[3:]
                self.device._send_enc_rdn(random_nr)
            elif data[:3] == b'\x10\x02\x04':
                self.device.state = AUTH_STATES.REQUEST_RN_ERROR
            elif data[:3] == b'\x10\x03\x01':
                self.device.state = AUTH_STATES.AUTH_OK
            elif data[:3] == b'\x10\x03\x04':
                self.device.status = AUTH_STATES.ENCRIPTION_KEY_FAILED
                self.device._send_key()
            else:
                self.device.state = AUTH_STATES.AUTH_FAILED
        elif hnd == self.device._char_heart_measure.getHandle():
            self.device.queue.put((QUEUE_TYPES.HEART, data))
        elif hnd == 53:
            if len(data) == 20 and struct.unpack('b', data[0])[0] == 1:
                self.device.queue.put((QUEUE_TYPES.RAW_ACCEL, data))

            # Not sure about this, need test
            elif len(data) == 16:
                self.device.queue.put((QUEUE_TYPES.RAW_HEART, data))
        else:
            self.device._log.error("Unhandled Response " + hex(hnd) + ": " +
                                   str(data.encode("hex")) + " len:" + str(len(data)))
```

Figura 27 Código para la autenticación

Para poder recibir los diferentes tipos de datos hay que tener en cuenta el UUID de los diferentes servicios de la pulsera, algunos de estos se pueden observar en la siguiente figura:

```
CHARACTERISTIC_HZ = "00000002-0000-3512-2118-0009af100700"
CHARACTERISTIC_SENSOR = "00000001-0000-3512-2118-0009af100700"
CHARACTERISTIC_AUTH = "00000009-0000-3512-2118-0009af100700"
CHARACTERISTIC_HEART_RATE_MEASURE = "00002a37-0000-1000-8000-00805f9b34fb"
CHARACTERISTIC_HEART_RATE_CONTROL = "00002a39-0000-1000-8000-00805f9b34fb"
CHARACTERISTIC_ALERT = "00002a06-0000-1000-8000-00805f9b34fb"
CHARACTERISTIC_CUSTOM_ALERT = "00002a46-0000-1000-8000-00805f9b34fb"
CHARACTERISTIC_BATTERY = "00000006-0000-3512-2118-0009af100700"
CHARACTERISTIC_STEPS = "00000007-0000-3512-2118-0009af100700"
```

Figura 28 UUID de los servicios de la MiBand3

Para la recepción de los datos habría que decodificar los datos, en la siguiente figura se puede observar el código para la obtención de la información relativa a los pasos:

```
def get_steps(self):
    char = self.svc_1.getCharacteristics(UUIDS.CHARACTERISTIC_STEPS)[0]
    a = char.read()
    steps = struct.unpack('h', a[1:3])[0] if len(a) >= 3 else None
    meters = struct.unpack('h', a[5:7])[0] if len(a) >= 7 else None
    fat_gramms = struct.unpack('h', a[2:4])[0] if len(a) >= 4 else None
    # why only 1 byte??
    callories = struct.unpack('b', a[9])[0] if len(a) >= 10 else None
    print ("steps: ", steps)
    return {
        "steps": steps,
        "meters": meters,
        "fat_gramms": fat_gramms,
        "calories": callories
    }
```

Figura 29 Código para obtener pasos de la MiBand3

La MiBand3 también contempla modos de funcionamiento continuos, es decir, están midiendo continuamente los diferentes sensores, el cardiaco y el acelerómetro, para ello solo hay que mandar el comando con el UUID del servicio del que queremos tener información y se nos devuelve, la siguiente figura se trata del código donde se observa los comandos para activar la medición continua:

```
    char_sensor = self.svc_1.getCharacteristics(UUIDS.CHARACTERISTIC_SENSOR)[0]
    char_sensor_d = char_sensor.getDescriptors(forUUID=UUIDS.NOTIFICATION_DESCRIPTOR)[0]
    char_sensor1 = self.svc_1.getCharacteristics(UUIDS.CHARACTERISTIC_HZ)[0]
    char_sensor1_d = char_sensor1.getDescriptors(forUUID=UUIDS.NOTIFICATION_DESCRIPTOR)[0]


    ## stop heart monitor continues & manual
    char_ctrl.write(b'\x15\x02\x00', True)
    char_ctrl.write(b'\x15\x01\x00', True)

    ## IMO: enablee heart monitor notifications
    char_d.write(b'\x13\x01', True)

    ## start hear monitor continues
    char_ctrl.write(b'\x15\x01\x01', True)

    # enecabling accelerometer raw data notifications
    char_sensor.write(b'\x01\x01\x19')
    char_sensor.write(b'\x02')
    char_sensor1_d.write(b'\x01\x00', True)

    t = time.time()
    loop = 0
    while loop<2:
        self.waitForNotifications(0.5)
        self._parse_queue()
        # send ping request every 12 sec
        if (time.time() - t) >= 12:
        print(loop)
        loop+=1
            char_ctrl.write(b'\x16', True)
            t = time.time()
```

Figura 30 Código para la medición continua de acelerómetro y sensor cardiaco

- Para habilitar el sensor cardiaco hay que mandar el comando **\x15\x01\x01**
- Para el acelerómetro hay que mandar varios comandos: **\x01\x01\x19** y **\x02**

El sensor cardiaco nos devuelve el número de pulsaciones y el acelerómetro nos devuelve en cada paquete 3 medidas de los diferentes ejes (x, y, z).

Toda esta información estará siendo guardará en la base de datos, de la que hablaremos posteriormente.

4.3 Detección de actividades

Ahora que ya tenemos todos los sensores y sabemos cómo funcionan podemos describir cómo se van a identificar las diferentes actividades usando solo estos sensores, toda esta información se puede observar en la siguiente tabla:

Actividades Diarias

Actividad	Sensores implicados	Medidas
Ducharse	Sensor de lluvia	El sensor de lluvia tiene un rango de 0-1024 valores en los cuales podemos definir diferentes casos, para nosotros existirán dos: - El primer rango entre 0 -256 es cuando el sensor está completamente seco, o, puede que caigan unas gotas de agua ya sea por limpiar la ducha o cualquier otra razón que no sea darse una ducha, y por tanto no reportará nada. - El segundo caso, entre los valores 257-1024, es sensor reportará que alguien se está duchando.
Ir al baño	Sensor de distancia	Se pone el sensor detrás del inodoro, cuando detecta hay un obstáculo muy cercano se puede decir que se está usando.
Dormir	Sensor de presión + Miband3	Se pone un sensor de presión debajo de las sabanas para ver cuando está siendo usada y si a eso le añadimos el sensor cardiaco y el acelerómetro de la MiBand3 podemos llegar a saber si está en la cama durmiendo.

Estar sentado sin hacer nada	Sensor de presión + Sensor de sonido + Miband3	Este es un caso similar al de dormir, salvo que las pulsaciones al estar despierto son mayores, además, si no se detecta ruido se puede decir que está sentado sin hacer nada.
Salir de casa	Sensor de distancia + Miband3	Para saber si es la persona monitorizada la que sale de casa se ha optado por combinar el sensor de distancia junto con la Miband3. Para saber si ha salido de casa y coloca el sensor en la puerta de casa y si detecta un cambio en la distancia, junto a la desconexión de la pulsera se puede saber cuándo sale de casa, ya que la conexión bluetooth no tiene mucho alcance, además podemos saber a la vuelta de la persona, los pasos que ha realizado, añadiendo un extra de seguridad a la afirmación de que ha salido realmente de casa.

Actividades sociales diarias

Ver la televisión/ escuchar música/ leer un libro	Sensor de sonido + Sensor de presión + Miband3	Si está viendo la televisión el sensor de presión estará activado, junto con el sonido. Para escuchar música puede ser lo mismo, o está sentado escuchando o se está moviendo, por tanto, en el acelerómetro y las pulsaciones se podría saber. En cambio, si está haciendo crucigramas, está sentado, el acelerómetro, más o menos con movimiento, no hay sonido, pero se estima que al estar realizando alguna actividad de este tipo las pulsaciones aumentan ligeramente con respecto a las pulsaciones en reposo.

Hablar con alguien	Sensor de sonido + Sensor de presión + Miband3	Es un caso similar al anterior, salvo que se estima que las personas gesticulan al hablar, por tanto, el acelerómetro estaría en constante movimiento.
Estar sentado sin hacer nada en un lugar oscuro	Sensor de sonido + Sensor de presión + Miband3 + Sensor lumínico	Uno de los mayores síntomas de aislamiento social es estar viviendo en silencio sin hacer nada en lugares oscuros. Para ello se monitorizará con los sensores lumínicos las condiciones de luz en todas las estancias llegando a detectar este tipo de conductas.

Tabla 3 Detección de actividades

4.4 Servidor

Llegados a este punto, tenemos los 5 tipos de sensores y la Miband3, el siguiente paso es configurar el servidor, la máquina que haremos como servidor es una Raspberry Pi 3 modelo B. Características:

- CPU 1,2 GHz Broadcom BCM2837 de 64 bits
- 1 GB de RAM
- Bluetooth de baja energía (BLE).
- Puerto Ethernet de 100Mb.
- GPIO extendido de 40 pines.
- 4 puertos USB 2.0.
- Salida estéreo de 4 polos y puerto de video compuesto.
- HDMI.
- Puerto CSI para la conexión de una cámara Raspberry Pi.
- Puerto DSI para la conexión de una pantalla táctil Raspberry Pi.
- Puerto microSD.
- Fuente de alimentación micro USB de hasta 2,5 A

Los componentes extra necesarios para el correcto funcionamiento serian una microSD de 32Gb y el adaptador USB Aeotec Z-Stick Gen5 para poder comunicarse con los sensores, ya que estos al estar funcionando mediante el protocolo Z-Wave la Raspberry Pi no dispone de sistemas capaces de comunicarse con ellos, además el adaptador USB funcionará como maestro del resto de los sensores.

Figura 31 adaptador USB Aeotec Z-Stick Gen5

El sistema operativo que instalaremos es el oficial disponible en la página web de Raspberry Pi, https://www.raspberrypi.org/software/operating-systems/, permitiéndonos instalar y configurar cualquier aplicación y/o componentes que necesitásemos.

Para poder manejar toda la información relativa a los sensores haremos uso de la plataforma IoT openHAB, esta nos permitirá conectar todos los señores que hemos desarrollado anteriormente con facilidad además de permitir guardar todos datos en una base de datos, base de datos que en este caso será InfluxDB, que como se ha visto en el apartado 3.4.2, esta base de datos guarda los datos de forma muy compacta, tan solo necesitamos tres columnas, en la siguiente figura se puede observar cómo sería un ejemplo de tabla de datos de los sensores:

Sensor	Valor	Hora
Sensor 1	140,50	23/04/2020 @ 10:00
Sensor 2	110,02	23/04/2020 @ 10:00
Sensor 1	142,32	23/04/2020 @ 10:05
Sensor 2	110,50	23/04/2020 @ 10:05

Figura 32 Ejemplo de tabla en InfluxDB

4.4.1 Instalación de openHAB

Una vez conocidos todos los requisitos pasamos a la instalación de la plataforma, para ello, seguimos los siguientes comandos:

>> sudo apt-get upgrade

>> sudo raspi-config advanced options/ Memory Split ->16

>> wget -qO - 'https://bintray.com/user/downloadSubjectPublicKey?username=openhab' | sudo apt-key add -

>> sudo apt-get install apt-transport-https

>> echo 'deb https://openhab.jfrog.io/openhab/openhab-linuxpkg unstable main' | sudo tee /etc/apt/sources.list.d/openhab2.list

>> sudo apt-get update

>> sudo apt-get install openhab2

>> sudo apt-get install openhab2-addons

Tras ejecutar los comandos tenemos instalado openHAB, pero aparte, se recomienda tener instalado la versión java8, para ello con los siguientes comandos instalaremos los requisitos adicionales:

>> sudo su

>> apt-key adv --keyserver hkp://keyserver.ubuntu.com:80 --recv-keys 0x219BD9C9

>> echo 'deb http://repos.azulsystems.com/debian stable main' > /etc/apt/sources.list.d/zulu.list

>> apt-get update -qq

>> apt-get install zulu-embedded-8

Para iniciar openHAB:

>> sudo systemctl start openhab2.service

>> sudo systemctl status openhab2.service

>> sudo systemctl daemon-reload

>> sudo systemctl enable openhab2.service

Finalmente, es necesario añadir permisos:

>> sudo adduser openhab dialout

>> sudo adduser openhab tty

>> sudo adduser openhab Bluetooth

>> sudo adduser openhab audio

4.4.2 Configuración de openHAB

Para acceder basta con hacerlo a través del navegador web, en puerto 8080 estará activa la plataforma, la figura 34 es lo que se debería observar al acceder por primera vez OpenHAB.

Ahora, tenemos que añadir los dispositivos, openHAB cuenta con un repositorio enorme de prácticamente todos los sensores comerciales que hay en el mercado de las diferentes tecnologías y fabricantes, para conectar un dispositivo que este en ese repositorio sería muy sencillo, bastaría con instalar el Binding correspondiente y el dispositivo se podría añadir muy fácilmente.

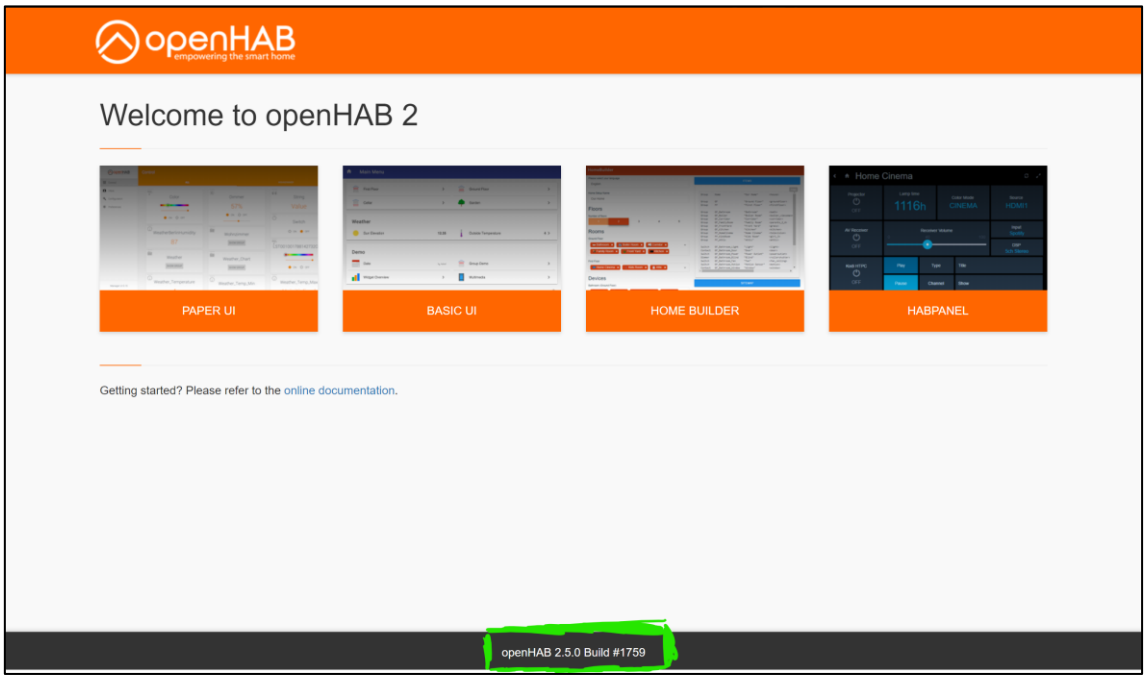

Figura 33 Panel principal OpenHAB

Pero, en nuestro caso, nuestros sensores los hemos desarrollado nosotros desde cero, por tanto, estos no están en el repositorio, hemos de modificar dicho repositorio para que estos aparezcan cuando busquemos nuevos dispositivos dentro de la plataforma.

La manera de añadir nuestros dispositivos no es muy complicada, basta con descargarse la versión del repositorio, se trata de un archivo .jar, para la versión que estemos usando de la plataforma, en este caso se trataría de la versión snapshot 2.5.0.

Lo que hay que añadir son un archivo XML por cada sensor que queramos poner nuevo, en la siguiente figura podemos ver un ejemplo de cómo quedaría el archivo XML que hay que añadir para poder detectar el sensor.

Estos archivos hay que añadirlos a la carpeta, dentro del archivo .jar, \ESH-INF\things, en la siguiente figura se vería nuestro dispositivo dentro del archivo:

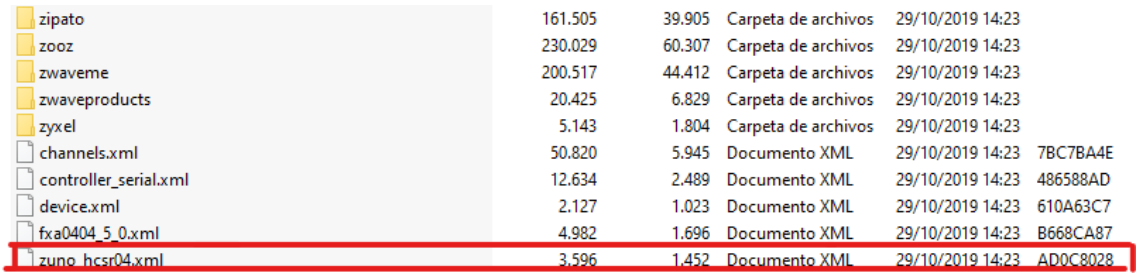

Figura 34 Archivo jar

```xml
<?xml version="1.0" encoding="UTF-8"?>
<thing:thing-descriptions bindingId="zwave"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:thing="https://openhab.org/schemas/thing-description/v1.0.0"
  xsi:schemaLocation="https://openhab.org/schemas/thing-description/v1.0.0
                      https://openhab.org/schemas/thing-description/v1.0.0">

  <thing-type id="zwaveme_hcsr04_00_005" listed="false">
    <label>Z-Uno HC-SR04 Ultrasonic distance sensor based on HC-SR04</label>
    <description></description>
    <category>Sensor</category>

    <!-- CHANNEL DEFINITIONS -->
    <channels>
      <channel id="sensor_temperature" typeId="sensor_temperature">
        <label>Sensor (distance)</label>
        <properties>
          <property name="binding:*:QuantityType">COMMAND_CLASS_SENSOR_MULTILEVEL;type=TEMPERATURE</property>
        </properties>
      </channel>
    </channels>

    <!-- DEVICE PROPERTY DEFINITIONS -->
    <properties>
      <property name="vendor">Z-Wave.Me</property>
      <property name="modelId">Z-Uno HC-SR04</property>
      <property name="manufacturerId">0115</property>
      <property name="manufacturerRef">0110:0001</property>
      <property name="versionMin">0.5</property>
      <property name="versionMax">10</property>
      <property name="dbReference">1046</property>
    </properties>

    <!-- CONFIGURATION DESCRIPTIONS -->
    <config-description>

      <!-- STATIC DEFINITIONS -->
      <parameter name="node_id" type="integer" min="1" max="232" readOnly="true" required="true">
        <label>Node ID</label>
        <advanced>true</advanced>
      </parameter>

    </config-description>

  </thing-type>

</thing:thing-descriptions>
```

Figura 35 XML del sensor de distancia

Una vez hecho esto pasamos a habilitar dicho paquete en la plataforma, para ello basta con acceder a la consola de openHAB, y ejecutar los siguientes comandos:

>> openhab-cli console

>> openhab-trasnport-serial

Finalmente reiniciamos la Raspberry Pi y ya estaría nuestro paquete disponible para instalar. Para realizar la instalación, vamos a Configuración>Bindings y seleccionamos Z-Wave Binding.

El siguiente paso es conectar todos los sensores al Adaptador USB Aeotec para que estos sepan como comunicarse entre ellos, el emparejamiento es muy fácil, tan solo hay que mantener pulsado el botón del USB hasta que este se quede parpadeando con luz azul y rápidamente mientras esta en este modo mantener pulsado el botón “pair” del microcontrolador Z-Uno hasta que este esté parpadeando con luz azul, tras unos segundos ambos emitirán una luz azul

parpadeante pero con una frecuencia mayor y se apagaran las luces, tras esto ambos dispositivos están correctamente emparejados. Este proceso hay que repetirlo para cada sensor, en nuestro caso hay que hacer esto con los 5 sensores que hemos desarrollado anteriormente.

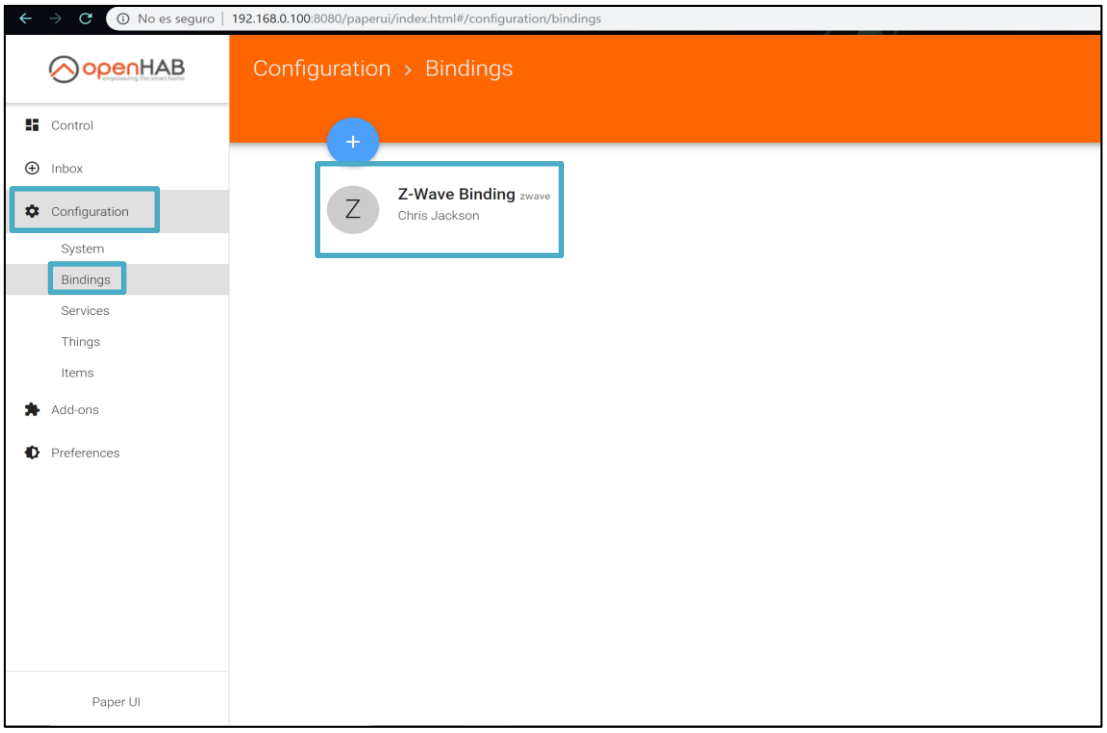

Figura 36 Instalación paquete Z-Wave

Tras emparejar todos los sensores, introducimos el USB en la Raspberry Pi y accedemos al menú Inbox de la plataforma openHAB, le damos al botón "+", como se puede observar en a la siguiente figura, y nos debería aparecer el controlador que acabamos de instalar.

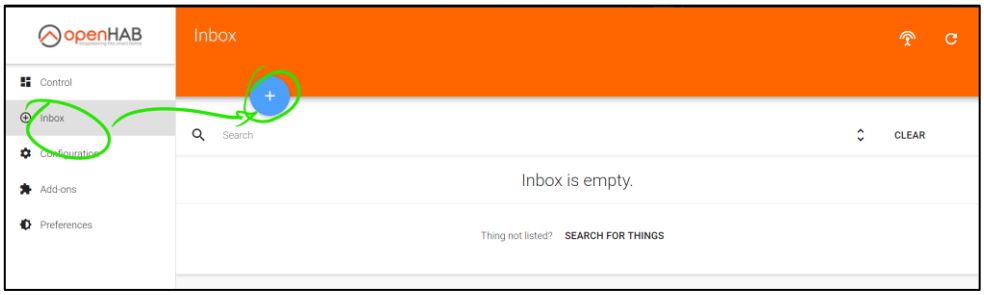

Figura 37 Menú Inbox

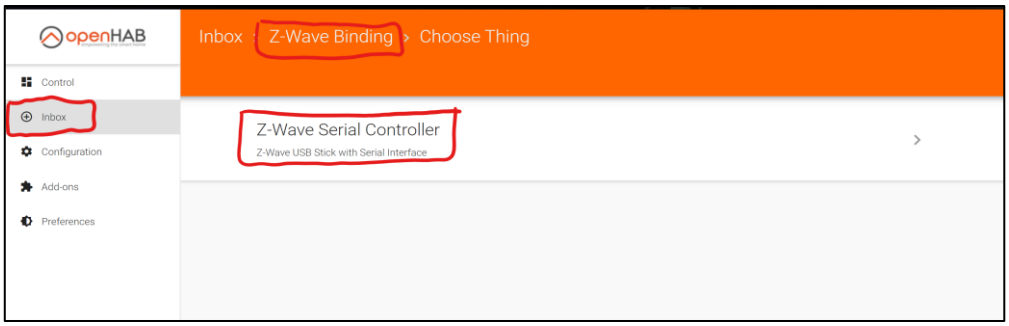

Figura 38 Controlador Z-Wave instalado

Finalmente, al seleccionar el controlador nos deberían ir apareciendo los diferentes sensores que están conectados a dicho controlador, en la siguiente imagen se puede observar un ejemplo de cómo aparecerían dichos sensores. Para añadir los sensores a la plataforma, bastaría con darles al botón azul y listo.

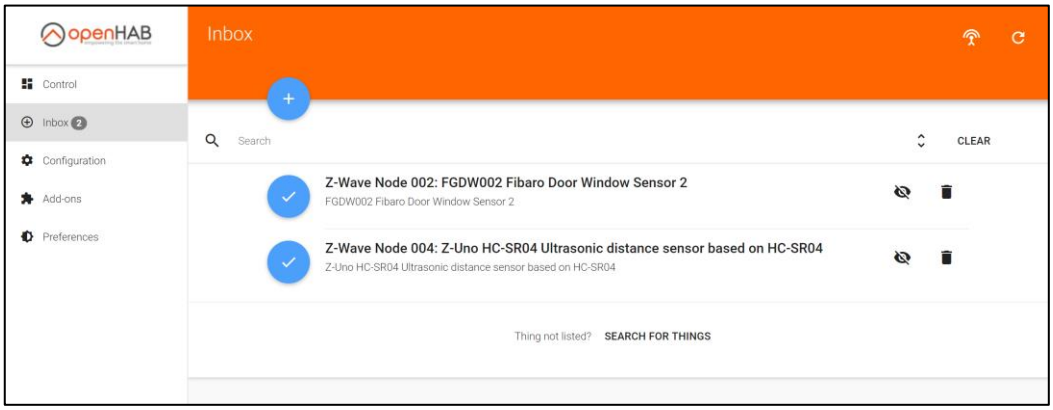

Figura 39 Sensores disponibles para añadir

El siguiente paso, una vez, añadidos todos los sensores, es crear los ítems dentro de la plataforma, uno para cada sensor, para ello vamos a Configuración>Things y le damos a la opción crear un ítem por cada canal, también podemos seleccionar los que queramos, pero les daremos a todos por facilidad y porque solo están los canales que nosotros hemos creado en el XML. En la siguiente figura se puede ver un ejemplo de los diferentes canales que tiene un sensor comercial.

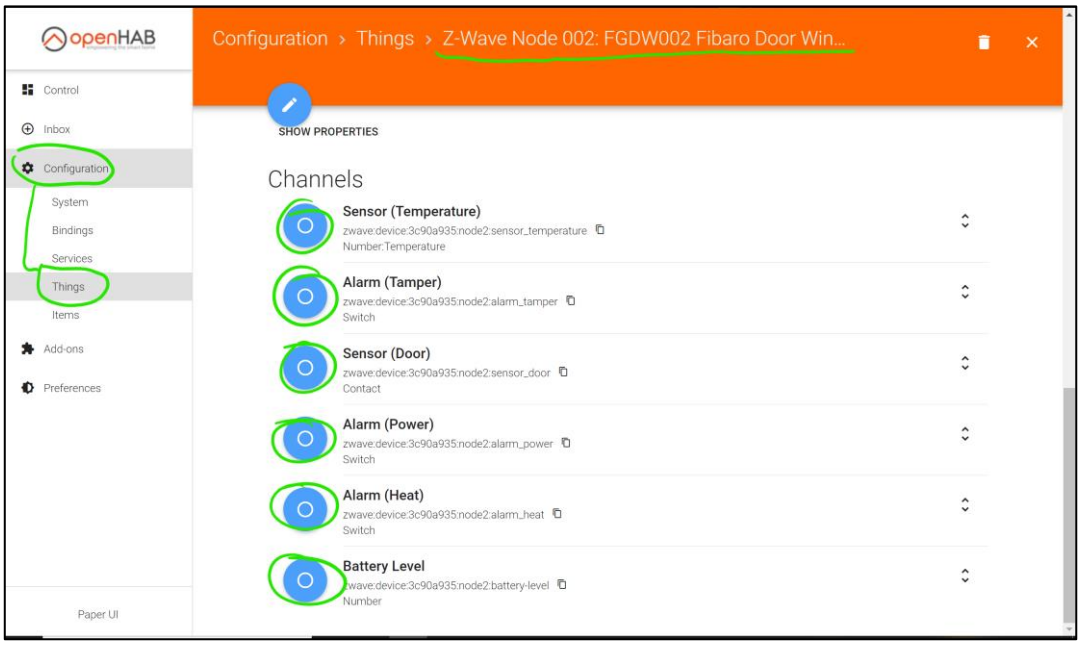

Figura 40 Canales de un sensor comercial

Desde el panel de control podremos ver los diferentes sensores con sus valores, en la siguiente imagen se puede observar cómo sería con un sensor comercial, este no tiene ningún valor puesto que no le hemos dado tiempo a reportar los datos.

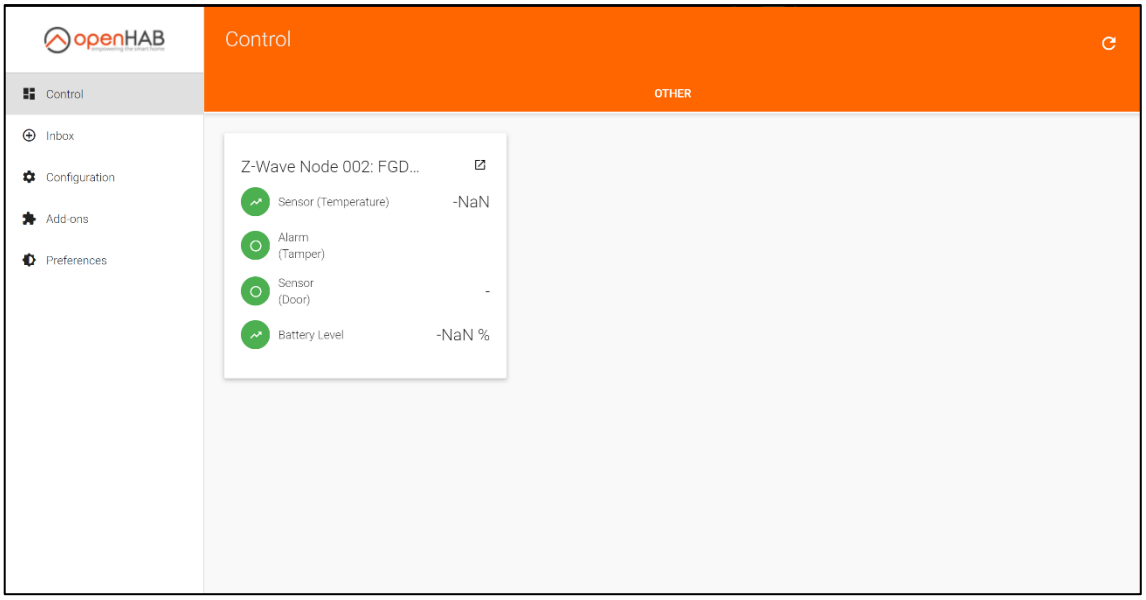

Figura 41 Panel Control de openHAB

El siguiente paso en la configuración de openHAB es la instalación de la base de datos y la habilitación de persistencia en los sensores. Para la instalación de la base de datos hay que seguir los siguientes comandos:

>> wget -qO- https://repos.influxdata.com/influxdb.key | sudo apt-key add –

>> echo "deb https://repos.influxdata.com/debian stretch stable" | sudo tee /etc/apt/sources.list.d/influxdb.list

>> sudo apt update

>> sudo apt install influxdb

>> sudo systemctl unmask influxdb

>> sudo systemctl enable influxdb

>> sudo systemctl start influxdb

Para habilitar la persistencia en openHab basta con ir a la pestaña Add-ons>Persistence donde buscamos, seleccionamos e instalamos la opción InfluxDB Persistence, en la siguiente figura observamos como quedaría:

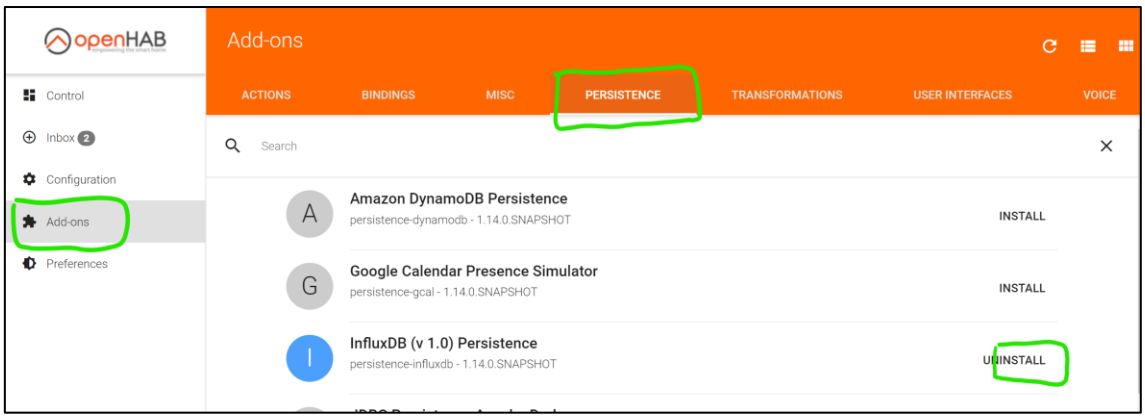

Figura 42 Instalación de persistencia InfluxDB

Para configurar la persistencia, para decirle donde está la base de datos a la plataforma y cada cuanto tiene que guardar los valores, editamos dos archivos de texto que podemos encontrar en:

- **/etc/openhab2/services/influxdb.cfg**, en este archivo pondremos cada cuanto queremos que se guarden los valores, para este caso haremos que se actualicen forzosamente cada hora, aunque no haya cambiado o cada vez que se realice un cambio, se puede observar en la siguiente figura como quedaría.

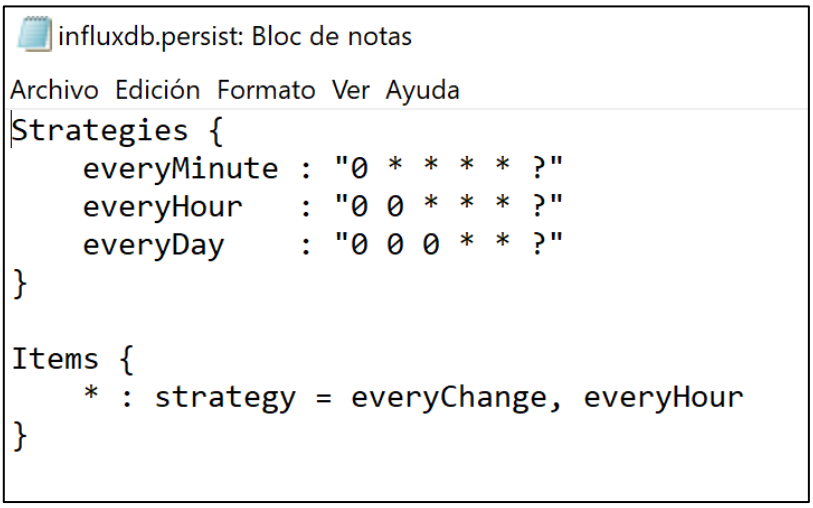

Figura 43 Archivo influxdb.cfg

- **/etc/openhab2/persistence/influxdb.persist**, en este archivo lo que se configura es la dirección de la base de datos, así como credenciales de acceso a la misma, en la siguiente figura se observa como quedaría:

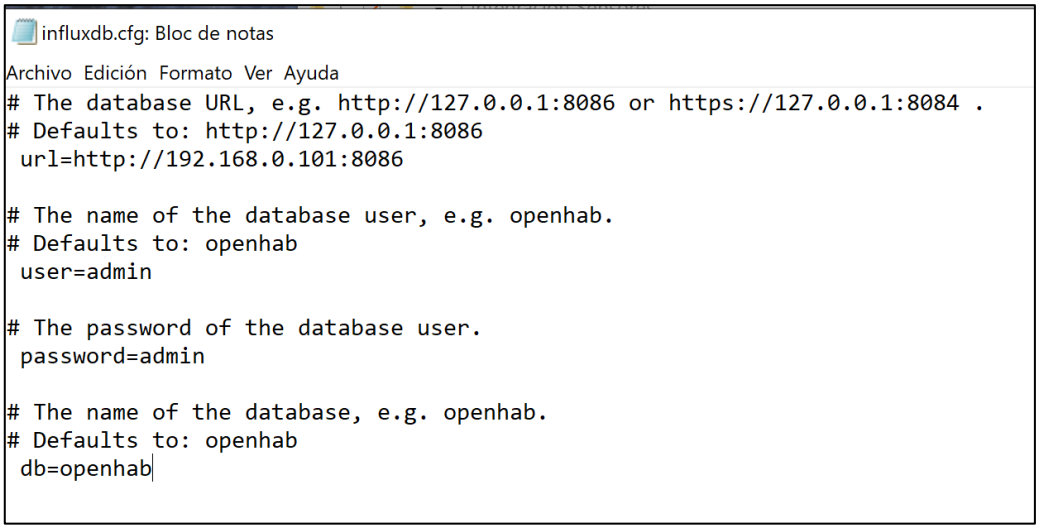

Figura 44 Archivo influxdb.persist

Tras realizar todos estos pasos tendremos completamente integrados todos los sensores junto con la pulsera inteligente Miband3, enviando todos los datos recopilados a la base de datos Influxdb instalada en la Raspberry Pi, lo que falta ahora es poder acceder a todos esos datos y juntarlos para verlos como se ha descrito en la tabla 2, para ello se desarrollara una aplicación Android para que el cuidador o familiar pueda ver de forma fácil y rápida.

4.5 Aplicación Android

La aplicación tiene que servir principalmente para poder observar las actividades que nos permite saber el grado de aislamiento de la persona. Para ello, la aplicación se conectará a la base de datos mediante WiFi y se descargará todos los datos de la base de datos filtrando por los datos de la última semana, lo cual nos permitirá saber el grado de aislamiento. En la siguiente figura se observa como quedaría la aplicación con el resultado de la última semana de actividades de una persona monitorizada.

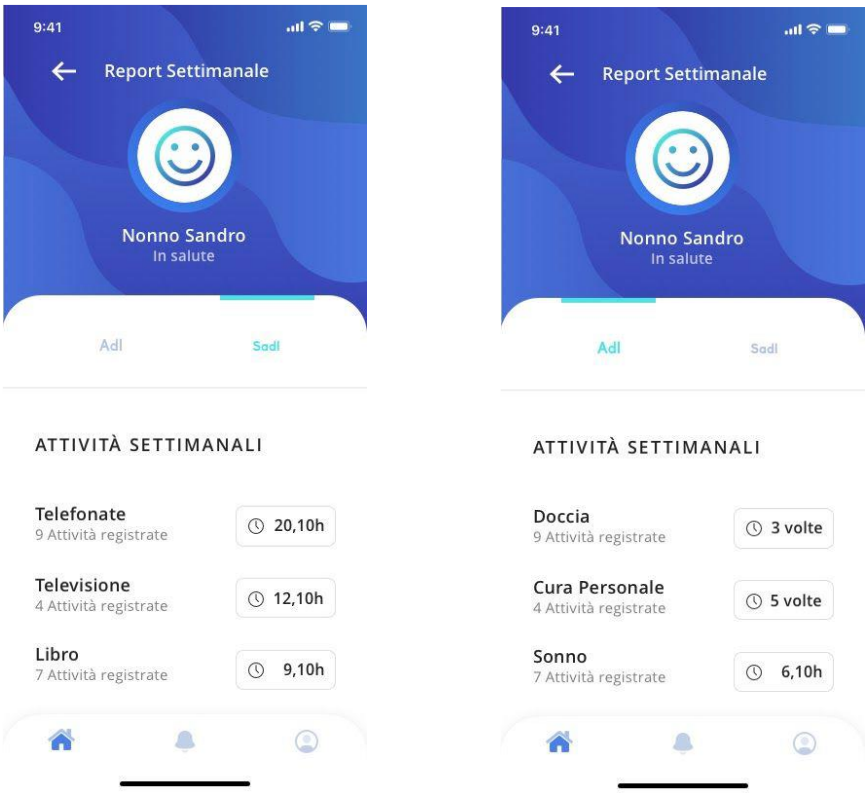

Figura 45 Aplicación para visualizar actividades

Teniendo el resumen de actividades dado por la aplicación y los umbrales que se han tenido que definir previamente, ya que no todas las personas son iguales ni tienen las mismas rutinas, el cuidador puede saber en qué ámbito del cuidado tiene que hacer énfasis.

Capítulo 5. Conclusiones y líneas futuras

5.1 Conclusiones

En el trabajo final de máster realizado, de carácter profesional, se ha desarrollado con éxito una aplicación IoT, basada en una plataforma IoT abierta, capaz de recibir información de diversos sensores, permitiendo la monitorización de actividades cuotidianas que una persona realiza a lo largo del día en su casa, permitiendo saber si la persona en cuestión puede llegar a padecer de aislamiento social.

Para realizar este proyecto se han tenido que realizar un estudio de las diferentes tecnologías actuales en cuanto a sensores y domotización de una casa, concretamente se han visto las diferencias entre tecnologías como Zigbee y z-wave, que sirven para el mismo propósito para cada una enfocada en un aspecto diferente, Zigbee se centra en masividad de dispositivos conectados con baja distancia entre ellos mientras que Z-wave se centra en pocos dispositivos pero con mayor distancia entre ellos, para monitorizar una casa cualquiera valdría, se decidió por Z-Wave ya que nos permite interconectar los dispositivos de una forma transparente, solo nos hemos de preocupar por recopilar la información y enviarla a través de las herramientas que nos proporciona el propio microcontrolador.

Para las plataformas IoT se profundizo en una plataforma para uso domótico, es decir, esa principalmente pensada para recoger datos de sensores y actuar en los diferentes actuadores, por ejemplo, las luces. También existe la posibilidad de hacerlo mediante FIWARE, pero se desestimó ya que la facilidad de uso que proporciona la plataforma de domótica era muy superior ya que permite configurar en entorno grafico muchas cosas.

Todo esto se implementó usando una Raspberry Pi 3 como servidor, aunque es un dispositivo de bajo coste y bajo rendimiento es más que capaz de asumir el rol que se le exige, además al estar su sistema operativo basado en Linux se ha podido profundizar en el proceso de creación de servicios en entornos Linux.

En cuanto a la aplicación Android una mejora sería tener algún tipo de interactividad entere cuidador y persona cuidada, pero el tiempo que llevo desarrollar otras partes de este proyecto solo permitió tener listo agregado de actividades para la monitorización más clara de las mismas.

Para finalizar, personalmente, el desarrollo de este proyecto ha resultado de lo más interesante, ya que se han tenido que ver diferentes aproximaciones en cuanto a la hora de

monitorizar actividades con diferentes sensores. Además, la realización de este proyecto también ha supuesto un reto ya que de un supuesto teórico se ha llegado a implementar un sistema completo y relativamente barato que es capaz de ayudar a muchas personas si se usa adecuadamente.

5.2 Líneas futuras

Una línea futura de mejora que se podrían hacer es el desarrollo de una inteligencia artificial que en caso de que el cuidador no esté disponible en ese momento, anime a las personas cuidadas a realizar las actividades para evitar el riesgo de aislamiento social, por ejemplo, un asistente de voz que recomiende la realización de actividades tales como ducharse, hacer un crucigrama, etc.

Algo con lo que habría que trabajar seria la compactación de los sensores, ya que ahora mismo se está usando una batería portátil bastante abultada junto con el microcontrolador y el sensor, todo a la vista, sin empaquetado, no queda muy estético y puede dar problemas con las conexiones, se podría diseñar como primer paso, una caja para que quede todo más compacto.

También se podría implementar algún sistema para que los familiares pudieran interactuar de forma indirecta, por ejemplo, cargando fotos o videos de un evento importante y hacer que una inteligencia artificial sea capaz de recomendar su visualización a la persona cuidada y mostrar el contenido a través de un televisor, por ejemplo.

Finalmente, como aspecto fundamental, la seguridad de los datos, en principio, estos no tienen acceso a internet, aunque es necesario usar un router para acceder que la aplicación acceda a los datos de las Raspberry Pi, no es necesario que este router tenga acceso a internet, pero, se podría llegar a estos datos de forma relativamente sencilla, ya que la única seguridad es la contraseña del WiFi del router y la contraseña para acceder a la base de datos, habría que protegerlos bastante mejor mediante diferentes mecanismos, sobre todo si se quiere dar acceso a los datos desde fuera del hogar.

Bibliografía

[1] OECD (2017), Health at a Glance 2017: OECD Indicators, OECD Publishing, Paris, https://doi.org/10.1787/health_glance-2017-73-en.

[2] https://data.oecd.org/pop/elderly-population.htm

[3] https://zigbeealliance.org/es/solution/Zigbee/

[4] https://www.domoticalia.es/es/content/14-z-wave

[5] https://manuals.fibaro.com/es/motion-sensor/

[6] https://zigbeealliance.org/es/zigbee_products/3-series-micro-motion-sensor/

[7] https://aeotec.com/z-wave-sensor/

[8] https://www.electronicaembajadores.com/es/Productos/Detalle/SSPX001/sensores-y-componentes-arduino/sensores-de-proximidad/sharp-gp2y0a21yk0f-sensor-de-proximidad-por-infrarrojos

[9] https://naylampmechatronics.com/sensores-proximidad/10-sensor-ultrasonido-hc-sr04.html

[10] https://manuals.fibaro.com/es/hk-door-window-sensor/

[11] https://www.centralite.com/downloads/DataSheet-3SeriesMicroDoorSensor.pdf

[12] https://naylampmechatronics.com/sensores-luz-y-sonido/76-modulo-sensor-de-luz-digital-bh1750.html

[13] https://learn.sparkfun.com/tutorials/sik-experiment-guide-for-the-arduino-101genuino-101-board-spanish/experimento-15-usar-la-placa-de-detector-de-sonido

[14] https://www.330ohms.com/products/sensor-de-ruido-grove

[15] https://tostatronic.com/store/sensores/152-sensor-de-lluvia-yl-83.html

[16] https://tecnoidea.es/producto/sensor-de-fuerza-md30-60-30kg/

[17] https://programarfacil.com/podcast/esp8266-wifi-coste-arduino/

[18] https://zwave.es/index.php?_route_=ZMEEZUNO

[19] Documentación oficial openHab https://www.openhab.org/docs/

[20] Documentation official home assistant https://home-assistant.io/docs/

[21] Documentación oficial de FIWARE Orion: https://fiware-orion.readthedocs.io

[22] MongoDB Inc, What is MongoDB? https://www.mongodb.com/what-is-mongodb

[23] https://docs.influxdata.com/influxdb/v2.0/