



UNIVERSIDAD DE JAÉN
Escuela Politécnica Superior (Jaén)

TRABAJO FIN DE MÁSTER

HIGIA - SISTEMA DE RECONOCIMIENTO DE ACTIVIDADES HIGIÉNICAS CON SENSORES MULTIMODALES

ALUMNA:

Aurora Polo Rodríguez

TUTOR:

Dr. D. Javier Medina Quero

(Departamento de Informática)

Enero, 2022



Universidad de Jaén

Departamento de Informática

Dr. D. **Javier Medina Quero** tutor del Proyecto Fin de Máster titulado: **HIGIA - Sistema de reconocimiento de actividades higiénicas con sensores multimodales**, que presenta **Aurora Polo Rodríguez**, autoriza su presentación para defensa y evaluación en la Escuela Politécnica Superior de Jaén.

Jaén, enero de 2022

La alumna:

Aurora Polo Rodríguez

El tutor:

Javier Medina Quero

*Quisiera expresar mi agradecimiento a aquellos
que me han apoyado de una forma u otra
en este camino aún en construcción.*

*Especialmente, agradecerle a mi tutor ser la guía,
la motivación, la sabiduría y el cariño que cualquier
estudiante anhela descubrir en esta etapa.*

*Lo que me hace ser afortunada
es contar con cada uno de vosotros.*

Índice

1. Introducción	11
1.2. Objetivos	14
1.3. Propuesta	15
1.4. Planificación temporal	16
1.4.1. Estimación de tiempos	16
1.4.2. Diagrama de Gantt	18
1.5. Estimación de recursos y costes	18
1.6. Estructura de la memoria	21
2. Estado del arte	23
2.1. Wearables	24
2.2. Visión	25
2.3. Audio	26
2.4. Arquitecturas y Machine Learning en IoT	27
2.5. Propuestas IoT relacionadas con hábitos de higiene	29
3. Análisis y diseño del sistema	32
3.1. Definición de requisitos	32
3.1.1. Funcionales	32
3.1.2. No funcionales	34
3.2. Casos de uso	36
3.3. Selección de dispositivos	40
3.3.1. Sensores de gas para la detección de deposiciones	40
3.3.2. Sensor de humedad para detectar duchas	41
3.3.3. Sensores de calidad del agua para detección de orina y/o deposiciones	42
3.3.4. Wearable para detectar movimientos, orientación y proximidad al baño	44
3.3.5. Sensores de audio para detectar eventos higiénicos	45
3.3.6. Cámaras de visión termal y espectro visible	46
3.4. Conectividad de dispositivos	47
3.5. Middleware de comunicación, persistencia y arquitectura	51
3.5.1. MQTT	51

3.5.2. HTTP	53
3.5.3. MongoDB	55
3.6. Diseño del modelo de representación y persistencia de datos	57
3.7. Arquitectura general	59
4. Implementación	61
4.1. Integración de sensores heterogéneos y multimodales	61
4.1.1. Sensores de gas y humedad	61
4.1.2. Sensores de calidad del agua	65
4.1.3. Wearable (sensor inercial + RSSI)	68
4.1.4. Sensores multimedia	72
4.2. Reconocimiento de actividades con modelos de Machine Learning	73
4.2.1. Procesamiento de sensores ligeros y wearables	73
4.2.3. Clasificación de eventos de audio	78
4.2.4. Estimación de poses mediante sensores termal	83
5. Experimentación y resultados	90
5.1. Evaluación de sensores ambientales	90
5.2. Evaluación de procesamiento inercial del wearable	92
5.3. Evaluación de sensores de calidad de agua	93
5.4. Evaluación de reconocimiento de audio	97
5.5. Evaluación de estimación de landmarks desde cámaras termal	98
6. Conclusiones y trabajos futuros	100
7. Referencias	103

Índice de tablas

Tabla 1. <i>Estimación temporal</i>	17
Tabla 2. <i>Estimación de recursos y costes</i>	21
Tabla 3. <i>Requisitos funcionales definidos</i>	34
Tabla 4. <i>Requisitos no funcionales definidos</i>	36
Tabla 5. <i>Comparativa de especificaciones de los protocolos Wi-Fi y Bluetooth Low Energy</i>	50
Tabla 6. <i>Arquitectura de los modelos CNN + MFCC y CNN + LM</i>	82
Tabla 7. <i>Parámetros entrenables, tiempo de aprendizaje, millones de instrucciones (MI) y tiempo de evaluación</i>	97

Índice de figuras

Figura 1. <i>Propuesta del sistema y componentes desarrollados en el trabajo</i>	15
Figura 2. <i>Diagrama de Gantt</i>	18
Figura 3. <i>Casos de uso del sistema</i>	36
Figura 4. <i>Caso de uso “detectar excreciones humanas y duchas”</i>	37
Figura 5. <i>Caso de uso “detectar proximidad al baño y movimientos”</i>	37
Figura 6. <i>Caso de uso “reconocer sonido asociado a eventos higiénicos”</i>	38
Figura 7. <i>Caso de uso “recabar información de dispositivos ambientales”</i>	38
Figura 8. <i>Caso de uso “reconocer poses”</i>	39
Figura 9. <i>Caso de uso “persistir datos”</i>	39
Figura 10. <i>Sensor MQ-8</i>	41
Figura 11. <i>Sensor MQ-2</i>	41
Figura 12. <i>Sensor DHT11</i>	42
Figura 13. <i>Wi-Fi Pool Kit Quality Monitoring. En círculos rojos se detallan los principales componentes: Placa ESP8266, sondas de conexión y sensores RTD, pH y ORP</i>	43
Figura 14. <i>Xiaomi Mi Band 3, tamaño del dispositivo wearable y correa para la muñeca</i>	44
Figura 15. <i>Raspberry Pi Modelo 3B con micrófono USB y aplicación móvil para etiquetado de datos</i>	45
Figura 16. <i>Raspberry Pi NoIR Camera Module V2 + PureThermal 2 Smart I/O Module + FLIR Lepton 3.5 + Raspberry Pi Model 3B</i>	46
Figura 17. <i>Esquema de funcionamiento del protocolo MQTT</i>	52
Figura 18. <i>Ejemplo de estructura de documentos en MongoDB</i>	56
Figura 19. <i>Modelo de representación de datos</i>	58
Figura 20. <i>Arquitectura detallada del sistema propuesto</i>	59
Figura 21. <i>Placa NodeMCU ESP8266 con sensores integrados (sensor de gas MQ8 y sensor de temperatura y humedad DHT11)</i>	62
Figura 22. <i>Constantes definidas para los sensores MQ2 y MQ8, respectivamente</i> ...	63
Figura 23. <i>Sensor MQ integrado en placa NodeMCU ESP8266</i>	63
Figura 24. <i>Sensor DHT11 integrado en placa NodeMCU ESP8266</i>	64
Figura 25. <i>Lectura y publicación de datos por MQTT en placa NodeMCU ESP8266 con sensores integrados</i>	65
Figura 26. <i>Detalles de la placa Atlas Scientific Pool Kit</i>	66
Figura 27. <i>Fragmento de muestras tomadas por los sensores de calidad del agua</i> ..	66
Figura 28. <i>Calibración de los sensores de calidad del agua</i>	67
Figura 29. <i>Hebra para activar la conexión de giroscopio de la Xiaomi Band 3 en streaming</i>	68
Figura 30. <i>Envío de comandos de cabecera para iniciar y mantener el streaming de datos en tiempo real de la Xiaomi Band 3</i>	69
Figura 31. <i>Extracción de valores X, Y, Z del buffer de datos de Xiaomi Band 3</i>	69

Figura 32. Extracción de pasos, intensidad y HR de Xiaomi Band 3	70
Figura 33. Sincronización de datos del dispositivo wearable mediante BLE y MQTT71	
Figura 34. Captura de muestras de sonido desde Python.....	72
Figura 35. Captura de muestras de imágenes de las cámaras PureThermal2 y Raspberry Pi NoIR V2 en Python	73
Figura 36. Normalización de datos de RSSI.....	74
Figura 37. Evolución de cercanía al cuarto de baño basado en RSSI.....	74
Figura 38. Arriba) Cantidad de movimiento basada en la norma del giroscopio, centro) valor de los componentes de los ejes X,Y,Z y abajo) detección de la elevación basada en el eje $X > 180^\circ$	75
Figura 39. Arquitectura de componentes para el reconocimiento de sonido ambiente	79
Figura 40. Descripción del modelo de la CNN.....	80
Figura 41. Ejemplo de señales de audio sin procesar a 44,1 kHz, espectrograma log-Mel (LM) y coeficiente cepstral de frecuencia Mel (MFCC) de los eventos de higiene ..	81
Figura 42. Imagen de espectro visible correlacionada con la imagen de espectro termal y los puntos de landmark de estimación de OpenPose en la termal en escala de calor.....	84
Figura 43. Imagen original e imagen sintética generada por la aumentación de datos junto a los landmarks relacionados	85
Figura 44. Bloque residual con conexiones de enlace en Keras (incluye normalización, filtrado RELU y dropout)	86
Figura 45. Creación del modelo de landmarks en Keras. Integración de las imágenes de 120,x160, red residual y capas finales de relación de filtros con las salidas de los landmarks	87
Figura 46. Fragmento de las primeras capas generadas por los bloques residuales y últimas capas del modelo residual	89
Figura 47. Cuatro evaluaciones de la detección de las deposiciones con el sensor MQ-2 y MQ-8 (hidrógeno en azul y metano en rojo respectivamente).....	91
Figura 48. Concentración de humedad antes, mientras y durante la ducha	92
Figura 49. Cantidad de movimiento basada en la norma del giroscopio y detección de la elevación basada en la aceleración del eje X.....	92
Figura 50. Izquierda) Sensores del kit de monitorización de la calidad del agua acoplados al W.C. Derecha) Sondas conectadas desde los sensores a la placa ESP8266.....	93
Figura 51. Detección de eventos de cambios de caída de ORP, caída de PH, subida de ORP y subida de RTD durante los días del experimento.....	95
Figura 52. Sumatoria de los cambios de eventos en sensores detectados (caída de ORP, caída de Ph, aumento de ORP y RTD) con respecto a la etiqueta de verdad (ground truth) en círculos azules.....	96
Figura 53. Arriba) Reconocimiento de audio a partir del modelo entrenado y, abajo) la verdad del caso de estudio en tiempo real	97

Figura 54. <i>Evolución del RMSE en el conjunto de entrenamiento y test durante las 50 épocas de aprendizaje</i>	98
Figura 55. <i>Estimación de la pose de un usuario: los círculos azules corresponden a la estimación de la pose realizada por OpenPose desde el espectro visible (trasladado al termal), y los círculos rojos la realizada por la red residual desarrollada en este trabajo</i>	99

Capítulo 1

1. Introducción

En este capítulo se presenta el trabajo Fin de Máster “HIGIA - Sistema de reconocimiento de actividades higiénicas con sensores multimodales”, donde se introduce el problema a resolver y la motivación para desarrollar un sistema de sensorización y reconocimiento de actividades centrado en los hábitos de higiene de usuarios en espacios inteligentes.

1.1. Motivación

La supervisión de los hábitos higiénicos se perfila como una investigación prometedora donde se relacionan el correcto comportamiento y el estado funcional de las personas. Estas rutinas son especialmente sensibles en el caso de los pacientes dependientes con trastornos mentales, como los enfermos de Alzheimer o las personas de tercera edad con demencia que viven en residencias o en su domicilio.

Por ejemplo, la eliminación de la orina y las heces es una función corporal esencial que solemos independizar en los primeros años de vida, siendo el control de la continencia un hito importante del desarrollo y sociabilidad humana (Downey, 2010). La eliminación de residuos corporales suele ser una función privada, pero la discapacidad o los problemas de salud física o mental afectan a la independencia de esta, sobre todo cuando las afecciones se van agravando de forma progresiva (Van Kerrebroeck, 2002). Por otra parte, la higiene corporal se define como fundamental para garantizar el bienestar del individuo tanto físico como mental, es por ello que en la literatura científica y en el contexto sanitario se destaca la importancia de tomar

una ducha de forma periódica, hasta el punto de llevar un control semanal de estos hábitos como predictor clave del estado de salud del paciente. De hecho, en este contexto, las enfermeras desempeñan un papel importante en la prevención de problemas y el mantenimiento de la higiene (Baillie, 2014).

Siempre que sea posible, el objetivo de la supervisión o monitorización en este ámbito es que se pueda realizar de forma independiente, promoviendo la dignidad y la intimidad, algo fundamental para satisfacer las necesidades de eliminación de las personas implicadas. Esto requiere una especial habilidad y sensibilidad para llevar a cabo estos cuidados y, al mismo tiempo, preservar la autoestima de los pacientes, que en muchos casos están afectados por diversos motivos de salud (Escalada-Hernández, 2015).

Por otra parte, en la última década se ha propuesto una amplia gama de sensores heterogéneos en el campo del Internet de las Cosas (IoT) (Al-Sarawi, 2017) para recoger las actividades de los usuarios, aumentando así las capacidades de percepción de los entornos inteligentes. Con los desarrollos más recientes, la investigación en el reconocimiento de actividades ha ampliado la gama de sensores ambientales y vestibles que pueden utilizarse para detectar el comportamiento humano tanto en espacios interiores (Medina-Quero, 2018) como exteriores (Cruciani, 2019). En particular, el uso de sensores sintéticos y multimodales (Laput, 2017) ha permitido el uso de diferentes tecnologías de propósito general para la supervisión de la actividad del usuario mediante la fusión de sensores. La nueva tendencia de la multimodalidad integra normalmente sensores de vídeo, audio y wearables, así como otros dispositivos IoT con capacidades informáticas cada vez más elevadas.

En nuestro caso, los enfoques en hábitos de higiene deben tener en cuenta la privacidad y la no invasión como requisitos fundamentales para cualquier solución asistida por la tecnología (Ienca, 2018), por lo que, por ejemplo, debe evitarse el uso de sensores de espectro visible.

Así, en este trabajo, proponemos la siguiente sensorización de espacios inteligentes de acuerdo a los requisitos de no invasividad:

- Evaluación del nivel de humedad ambiente mediante sensores de temperatura y humedad situados en el cuarto de baño, con el objetivo de detectar duchas.
- Evaluación del movimiento inercial de la muñeca y proximidad al baño mediante sensores vestibles, en este caso una pulsera de actividad, con el objetivo de detectar diferentes eventos relacionados con la higiene (lavarse los dientes, lavarse las manos, etc).
- Evaluación de la calidad del aire mediante sensores de gas en el aseo, con el objetivo de detectar las deposiciones.
- Evaluación de la calidad del agua mediante sensores de pH, ORP y temperatura en los aseos, con el objetivo de detectar las excreciones humanas en condiciones naturalistas.
- Evaluación de los diferentes sonidos producidos en el cuarto de baño mediante sensores de audio, con el objetivo de predecir comportamientos o acciones asociadas a ellos.
- Estimación de la pose de una persona mediante sensores de visión termal, con el objetivo de detectar la presencia y pose en la estancia.

La integración de información obtenida a través de diversas fuentes es un punto diferenciador respecto a otras propuestas, ya que otorga al sistema una mayor precisión y robustez, minimizando los falsos positivos.

1.2. Objetivos

Los objetivos **principales** de este proyecto son los siguientes:

- Selección de dispositivos IoT para la sensorización de actividades higiénicas y eventos en el contexto del cuarto de baño.
- Integración de sensores heterogéneos y multimodales en una plataforma IoT que sincronice y envíe los datos en tiempo real de forma unificada.
- Recolección de datos y etiquetado de casos de uso en el contexto de eventos de higiene.
- Procesamiento de datos, modelado de datos y reconocimiento de eventos de higiene basado en modelos de Machine Learning para cada sensor seleccionado.

Los objetivos secundarios que se plantean son:

- Desarrollar modelos ligeros (de bajo coste computacional) para el tratamiento de sensores de baja frecuencia (binarios, ambientales) que puedan ser integrados en placas IoT en la detección de eventos de higiene.
- Evaluar sistemas de reconocimiento de audio en tiempo real basados en Deep Learning en el contexto de espacios de higiene.
- Crear modelos de regresión de Deep Learning para la estimación de poses humanas a partir de imágenes de espectro termal.

1.3. Propuesta

El propósito de este Trabajo de Fin de Máster es la creación de un sistema de sensorización y reconocimiento de actividades centrado en los hábitos de higiene que integre dispositivos multimodales de diferente naturaleza, tales como sensores ambientales de presencia, wearable, humedad, gas, medición de agua, audio y cámaras de visión termal. Dicho sistema permitirá la monitorización de actividades que son clave en las necesidades básicas humanas: correcta eliminación (orina y heces) y hábitos de higiene (ducha, aseo).

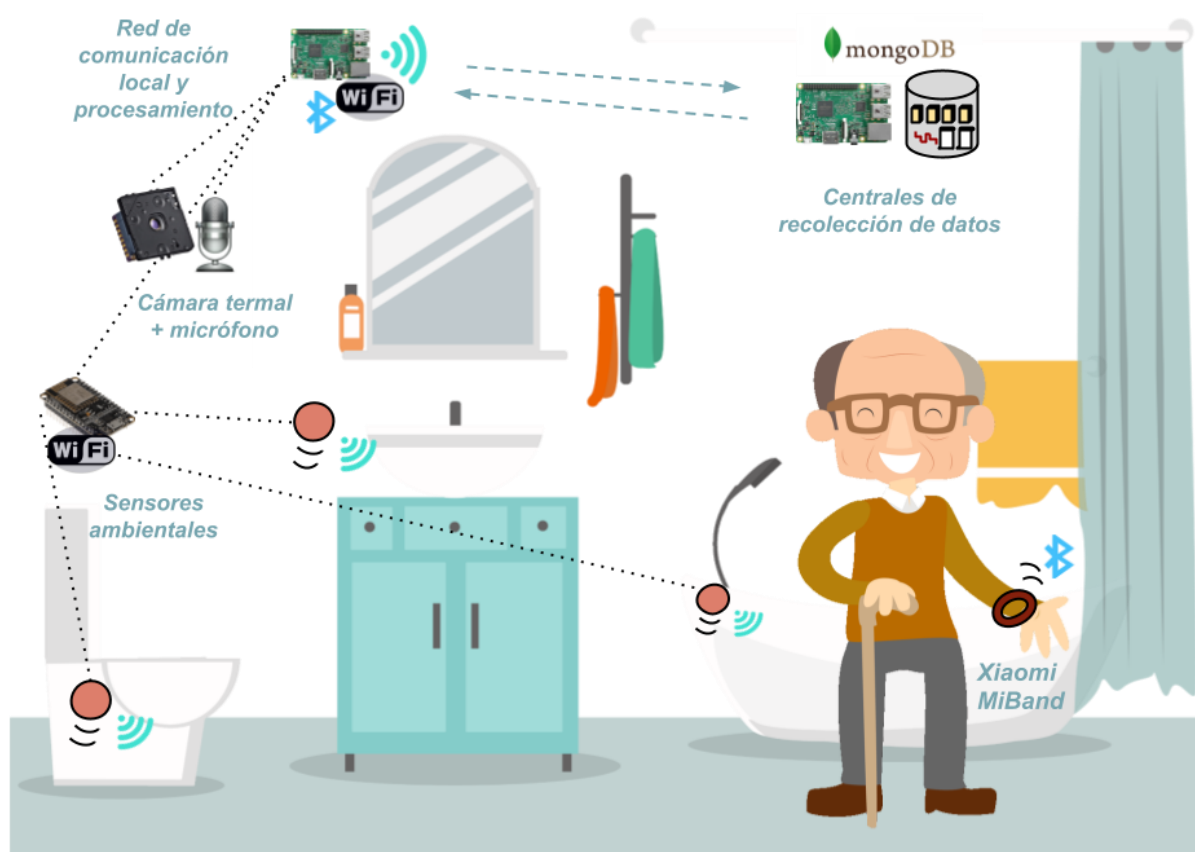


Figura 1. Propuesta del sistema y componentes desarrollados en el trabajo

1.4. Planificación temporal

Para lograr una correcta comprensión de la planificación temporal de este proyecto, se ha realizado una tabla con la estimación de tiempos de cada una de las tareas, acompañada de un diagrama de Gantt que permita comprobar visualmente el desarrollo a lo largo del tiempo.

1.4.1. Estimación de tiempos

Gracias a la siguiente lista, acompañada de una tabla, es posible contemplar de un vistazo las tareas en las que está dividido este proyecto, acompañado de la estimación inicial de la duración (en días y horas) de las tareas a desarrollar.

- **A0.** Análisis del estado del arte y selección de dispositivos
- **A1.** Pruebas de conexión, configuración y programación de los dispositivos y sensores de forma independiente
 - **A1.1** Sincronización de dispositivos binarios con la central de datos usando protocolos inalámbricos, generando un evento asíncrono ante la activación de los mismos
 - **A1.2** Recolección de datos de dispositivos multimodales (visión termal y audio) en tiempo real
 - **A1.3** Despliegue de un middleware de publicación-suscripción en tiempo real para recolección homogénea de los dispositivos
- **A3.** Recolección de datos en casos de uso dentro de entorno higiénicos
- **A4.** Procesamiento de Machine Learning
 - **A4.1** Procesamiento de sensores ligeros/wearable
 - **A4.2** Clasificación de eventos de audio
 - **A4.3** Estimación de poses mediante espectro termal
- **A5.** Desarrollo de memoria
- **A6.** Revisión y confección de la versión final

Tarea	Estimación de tiempo en días	Estimación de tiempo en horas (7 horas/día)
A0	10	70
A1	30	210
A1.1	18	126
A1.2	12	84
A1.3	3	21
A2	10	70
A2.1	7	49
A2.2	3	21
A3	15	105
A4	20	140
A4.1	7	49
A4.2	6	42
A4.3	7	49
A5	20	140
A6	5	35
TOTAL 110 días y (7h/día) 770 horas		

Tabla 1. Estimación temporal

1.4.2. Diagrama de Gantt

Para plasmar la planificación y programación de las tareas a lo largo del tiempo, se ha confeccionado un diagrama de Gantt, reflejando la dedicación establecida para cada una de las tareas:

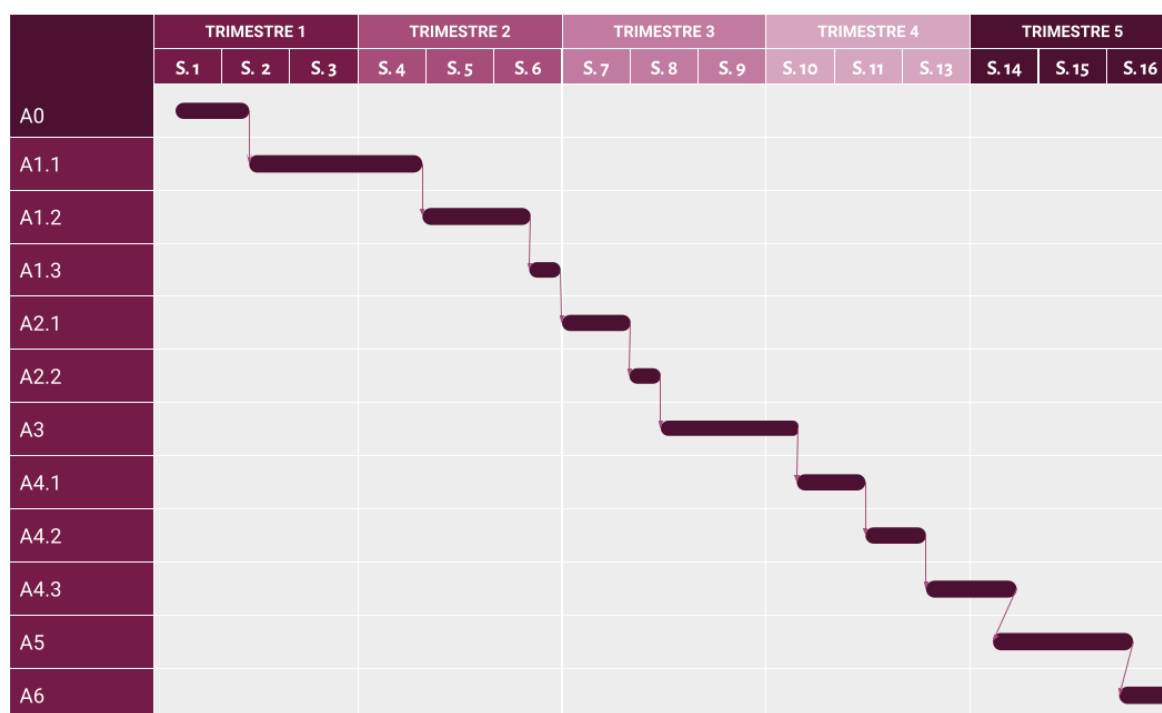


Figura 2. Diagrama de Gantt

1.5. Estimación de recursos y costes

El presupuesto es un punto muy importante para el desarrollo del proyecto, ya que es nuestra primera toma de contacto con el cliente. Para un presupuesto se realiza una estimación del esfuerzo, recursos y tiempo que conlleva desarrollar el proyecto. Estos recursos se pueden estructurar y definir de diversas formas, aunque en nuestro caso hemos optado por una división clara y concisa: costes hardware, software, de personal e indirectos. Hemos de destacar que no se han incluido en el

coste aquellos dispositivos de los que ya se disponía. A continuación, en la siguiente tabla se detalla el presupuesto para este Trabajo de Fin de Máster:

Recurso	Precio unidad + IVA	Unidades	Coste (€)
Costes hardware			
Raspberry Pi 3B	37,5	2	75
Xiaomi Mi Band 3	30	1	30
NodeMCU ESP8266	10	2	20
Sensor de gas MQ-2	5,29	1	5,29
Sensor de gas MQ-8	5	1	5
Sensor de temperatura y humedad DHT11	5,49	1	5,49
Micrófono USB	18,9	1	18,9
Cámara térmica FLIR Lepton 3.5	191	1	191
Cámara espectro visible Pi NoIR v2.1 8 MP 1080p	29,30	1	29,30
Módulo de conexión cámara	99	1	99

termal			
Wi-Fi Pool Kit Quality Monitoring	350	1	350
Servidor/ Computador		1	
Costes totales hardware: 828,98 € (No se incluye el computador, ya que se disponía de él)			
Costes software			
PlatformIO+Atom	0	1	0
Netbeans	0	1	0
Raspbian SO	0	1	0
Geany	0	1	0
MongoDB	0	1	0
Jupyter	0	1	0
Arduino IDE	0	1	0
Costes totales software: 0 €			
Costes de personal			

23 €/hora x 770 horas = 17.710 €
Costes indirectos
10% de 17.710€ = 1.771 €
TOTAL: 20.309,98 €

Tabla 2. Estimación de recursos y costes

1.6. Estructura de la memoria

La estructura de la presente memoria se ha dividido en las siguientes secciones:

- I. En el **capítulo 2** mostramos los trabajos relacionados sobre diferentes tipos de sensores, tecnologías y sistemas que utilizan un enfoque parecido al de nuestra propuesta.
- II. En el **capítulo 3** se describe el análisis y diseño del sistema con los requerimientos, casos de uso, dispositivos seleccionados, middleware de conectividad, arquitectura y diseño del modelo de datos.
- III. En el **capítulo 4** se detalla la implementación de sensores, la creación de los servicios remotos, el procesamiento y reconocimiento con Machine Learning y la implementación del modelo de persistencia; finalmente.
- IV. En el **capítulo 5** se muestran varios casos de estudio en un hogar real con condiciones naturalistas

- V. En el **capítulo 6** se incluyen las conclusiones y trabajos futuros derivados de este trabajo.
- VI. Por último, en el **capítulo 7** recogemos todas las referencias del presente documento.

Capítulo 2

2. Estado del arte

Los entornos inteligentes han surgido como una tecnología prometedora y disruptiva para reconocer el comportamiento de los habitantes en su vida diaria (Ali Hamad, 2018). Estos ecosistemas de sensores aportan soluciones alentadoras a los desafíos relacionados con el envejecimiento de la población (Bouchard, 2018) y las aplicaciones de eHealth (Bravo, 2018), mejorando la calidad de los servicios de atención y permitiendo que las personas permanezcan independientes en sus hogares el mayor tiempo posible (Rashidi, 2012). La integración de la tecnología en nuestra vida diaria de forma inmersiva y transparente fue introducida por el término computación ubicua, cuando la tecnología pasa a un segundo plano (López, 2020). Desde esta perspectiva visionaria de principios de los años 90 hasta nuestro concepto actual de IoT, se han explotado dos características clave en los últimos 30 años:

1. Baja invasividad de los dispositivos integrados (tanto en nuestro cuerpo como en nuestro entorno).
2. Dispositivos inteligentes conectados que proporcionan resultados interpretables procesando los datos recogidos, es decir, transformación de información en conocimiento.

Se han propuesto una amplia gama de sensores heterogéneos en el campo relacionado del Internet de las cosas (IoT) (Al-Sarawi, 2017) para identificar las actividades de los usuarios. Al principio, se desplegaron sensores binarios como dispositivos de bajo coste para recoger las interacciones con el entorno y con los objetos (Medina-Quero, 2018a). A continuación, los trabajos empezaron a incorporar dispositivos wearables, que proporcionaban una perspectiva en primera persona de

la actividad del usuario, permitiendo el reconocimiento de actividades y gestos (Ordóñez, 2016) utilizando principalmente sensores inerciales. Enfoques recientes han incorporado sensores de visión que se han propuesto, junto con modelos de Deep Learning, para analizar las observaciones visuales para el reconocimiento de patrones en los habitantes (Zhang et al., 2017b). De hecho, en las últimas propuestas científicas, el floreciente crecimiento de los dispositivos está promoviendo el uso de sensores multimodales que suelen integrar vídeo, audio y sensores wearables (Laput, 2017), así como otros dispositivos del IoT con una capacidad de computación cada vez mayor. Las nuevas tendencias convergen hacia los sensores sintéticos que se despliegan para percibir todo lo que hay en una habitación determinada, lo que permite el reconocimiento de actividades mediante la fusión de sensores. En este contexto, el procesamiento de audio mediante micrófonos para el etiquetado de eventos audibles está abriendo un prometedor campo de investigación dentro del reconocimiento de actividades (Cruz-Sandoval, 2019).

En este proyecto tomamos como referencia estas nuevas líneas de investigación, donde se combina el uso de sensores de diversa índole para el reconocimiento de eventos en un contexto determinado, en nuestro caso, el del cuarto de baño.

2.1. Wearables

El uso de los wearables de actividad ha crecido de forma espectacular en la población mundial (Shih, 2015). El impacto de los dispositivos wearables ha crecido en los últimos años, ya que son descriptores de la actividad física de los usuarios (Patel, 2015) y el uso de estos dispositivos ha demostrado aumentar la actividad física entre jóvenes (Heale, 2018), adultos y mayores (Mercer, 2016) promoviendo un cambio positivo en el comportamiento de salud de las personas que los visten (Mercer, 2016b). En consecuencia, los wearables de actividad son clave en los nuevos modelos socio sanitarios de promoción y rehabilitación que permiten evitar la

inactividad física, que se estima que genera alrededor de 3 millones de muertes cada año (Lim, 2012).

Entre los datos más relevantes registrados por estos dispositivos están los datos inerciales, la frecuencia cardíaca, el tiempo, intensidad y duración de las etapas del sueño. Además, el uso de wearable permite la diferenciación de los usuarios en entornos de multiocupación, resolviendo las limitaciones de los sensores binarios ambientales mediante las señales de localización Bluetooth (Espinilla, 2018).

Entre los dispositivos wearables, existe una gama más avanzada de los mismos que ha acercado fronteras entre dispositivos móviles y smart watch. Estos dispositivos integran un Sistema Operativo programable (Android Wear, iOS, Tizen) que permite desarrollar aplicaciones empotradas y adaptadas a estos dispositivos mediante plataformas de desarrollo abiertas a los programadores. El uso de estas aplicaciones personalizadas ha tenido un impacto profundo en el sector de aplicaciones de salud (King, 2018) como herramienta poco invasiva para obtener datos de los usuarios en su contexto de salud, destacando los campos de: autogestión de enfermedades crónicas, reconocimiento de actividad y velocidad de paso del usuario, y enfermería o atención domiciliaria (Zheng, 2014). Por ejemplo, en el contexto del espectro autista, los relojes inteligentes se han integrado como una herramienta adecuada para la autorregulación emocional con fines de asistencia (Torrado, 2017) incluyendo ubicuidad, detección y posibilidades de interacción.

2.2. Visión

El uso de sensores de visión en entornos inteligentes se ha visto impulsado por el auge del aprendizaje profundo (Mo, 2016), que ha mostrado un rendimiento alentador en el análisis de datos multimedia utilizando IoT (Hiriyannaiah, 2020). Las cámaras de espectro visible se utilizan para realizar seguimiento y localización (Brauer, 2014), detección de caídas (Waheed, 2017) y autenticación biométrica (Macek, 2016). Sin embargo, esta tecnología plantea preocupaciones en cuanto a la

invasividad (Sixsmith, 2004) en los despliegues en interiores y la privacidad del usuario (Medina-Quero, 2018b), que es un requisito esencial en términos de seguridad (Kong et al., 2018) y aceptación de tales tecnologías.

Con la aparición de los sensores de visión térmica, estas desventajas pueden mitigarse en gran medida, siendo una excelente alternativa para encontrar soluciones que permitan monitorizar a las personas en sus viviendas minimizando las preocupaciones éticas y de privacidad (Griffiths, 2018; Mohammadmoradi, 2017). Además, presentan una solución robusta a los problemas relacionados con la luz y las sombras (Han, 2005). De hecho, empresas científicas y comerciales han diseñado sensores de visión térmica con diferentes capacidades de resolución que han destacado en la clasificación de actividades y poses (López-Medina, 2020); siendo una excelente combinación con redes neuronales convolucionales (CNN), por ejemplo, para detectar caídas (Medina-Quero, 2018). En este contexto, el aprendizaje profundo (Deep Learning) también se ha mostrado como un enfoque adecuado para extraer características (Wang, 2019), ya que está relacionado con el reconocimiento de sensores multimodales como la visión y el audio, donde la obtención de características jerárquicas para reducir la complejidad es clave.

2.3. Audio

En el campo del reconocimiento de audio, la combinación de CNNs (Albawi, 2017) con el uso del espectrograma para la representación del sonido (Wyse, 2017) ha demostrado generar resultados alentadores que pueden ser utilizados para la clasificación de sonidos ambientales (Salamon, 2017; Kim, 2020; Lasseck, 2018). En concreto, se ha propuesto el uso del espectrograma log-Mel (LM) y del coeficiente cepstral de frecuencias Mel (MFCC) para una representación robusta en la clasificación de sonidos (Su, 2018).

En el ámbito del reconocimiento del sonido ambiental en espacios interiores, destacamos varias aproximaciones:

- En (Beltrán, 2015), el reconocimiento de eventos se llevó a cabo mediante la representación espectral del sonido que se aprendió utilizando modelos de Markov.
- En (Beltrán, 2019), se reconocieron dos clases de sonidos (por ejemplo, golpear y lavarse las manos) utilizando el espectro y el histograma de los sonidos mediante Support Vector Machine en condiciones naturalistas dentro de una residencia geriátrica.
- En parte del estudio de (Cruz-Sandoval, 2019), los micrófonos direccionales espaciales 3D permitieron capturar audio multidireccional de alta calidad para detectar eventos y la ubicación de los sonidos en el entorno. Para ello, calcularon los coeficientes cepstrales de frecuencia Mel como características espaciales que se relacionan con los eventos mediante modelos gaussianos.
- En (Laput, 2018), se recogieron 30 eventos para reconocer las 7 habitaciones o espacios donde el habitante realizaba actividades (baño, dormitorio, entrada de la casa, cocina, oficina, exterior y taller). En este trabajo también se evaluaron los espectrogramas log-Mel para la clasificación de eventos sonoros, junto con un modelo DL (VGG-16) pre-entrenado con audios de YouTube. Los resultados demostraron que la precisión difiere notablemente entre las condiciones controladas y los contextos de la vida real.

2.4. Arquitecturas y Machine Learning en IoT

Gracias a la irrupción del IoT en el campo de Ciencias de la Computación, han aparecido nuevos enfoques donde los objetos inteligentes interactúan entre sí y cooperan con los de su alrededor para alcanzar objetivos comunes (Atzori, 2010). De este modo, se traslada la computación en la nube con procesamiento centralizado (Atzori, 2010) a redes colaborativas de sensores, generando un gran impacto en modelos de Computación Ubicua y **Fog Computing** con la integración de dispositivos ambientales (Chen, 2015) y wearable (Lara, 2013).

En cuanto a la arquitectura de los componentes para el aprendizaje y la comunicación de los dispositivos, los paradigmas de la computación Edge (Shi, 2016) o la computación Fog (Bonomi, 2012) han localizado los datos y los servicios dentro de los dispositivos en los que se integran los sensores, proporcionando recursos virtualizados y servicios comprometidos. El éxito en el rendimiento en las transferencias de datos de comunicación y la distribución de procesos de aprendizaje de este tipo de arquitecturas, nos ha llevado a decantarnos por ellas para este proyecto.

Por otra parte, como ya hemos mencionado con anterioridad, el reconocimiento de actividades se ha convertido en un tema de investigación activo centrado en la detección de comportamientos humanos en entornos inteligentes (Espinilla, 2018). Como fuentes de información para realizar el reconocimiento, destacan los siguientes tipos de sensores:

1. Dispositivos vestibles o wearables, que se han utilizado para analizar actividades y gestos (Ordóñez, 2016).
2. Dispositivos de localización, que en la actualidad alcanzan una precisión extremadamente alta en contextos de interior (Ruiz, 2017).
3. Sensores de visión (sensores de luz visible o térmicos-infrarrojos) en secuencias de vídeo e imagen (Xu, 2013).
4. Sensores de audio (Cruz-Sandoval, 2019) que reconocen eventos basándose en los sonidos del entorno. A esto le ha seguido una nueva tendencia de sensores multimodales que ha permitido el uso de tecnologías de detección de propósito general para monitorizar diferentes actividades.

Los enfoques en el reconocimiento de actividades se agrupan principalmente en dos categorías: enfoques basados en conocimiento (López-Medina, 2020) y enfoques basados en datos (Medina-Quero, 2018). Varios estudios anteriores en este campo se han centrado en la clasificación de actividades en las que se conoce de antemano el comienzo y el final de las mismas y, por tanto, las características clave, lo que se denomina segmentación explícita (Krishnan, 2014) o evaluación

fuera de línea, ya que no trabajan en tiempo real. Sin embargo, la inclusión del tiempo real es un requisito clave para proporcionar respuestas a las condiciones naturalistas del mundo real (Radu, 2016), permitiendo servicios de asistencia adecuados. En el reconocimiento de actividades en tiempo real, donde se desconoce el inicio y el final de los eventos, se requieren enfoques basados en ventanas deslizantes para segmentar el flujo de datos (Krishnan, 2014). Además, en el contexto de los sensores multimodales, el uso de modelos de aprendizaje profundo ha mostrado un rendimiento prometedor en el procesamiento de datos multimedia, el cual abordaremos en las siguientes secciones.

2.5. Propuestas IoT relacionadas con hábitos de higiene

Como hemos podido comprobar en secciones previas, existen diferentes dispositivos con diversas capacidades que proporcionan una gran cantidad de información relevante, pudiendo adaptarse al contexto. De esta forma, la fusión de sensores se ha hecho cada vez más popular y necesaria. Además, la producción de conjuntos de datos disponibles públicamente para la comunidad científica se ha visto incrementada precisamente por la detección multimodal (Cruciani, 2019).

En el contexto de este Trabajo de Fin de Máster, la creación de un espacio inteligente en el cuarto de baño representa un contexto concreto, donde una amplia gama de dispositivos convive para capturar la información sobre los hábitos de higiene claves para el desarrollo que, a su vez, sirven como métrica de la calidad de vida de las personas. Esto representa un reto, ya que los eventos y actividades a reconocer representan un desafío para realizar una sensorización que no tiene todavía un extenso desarrollo en la literatura (centrada principalmente en actividades cotidianas generales).

Cabe destacar la importancia de la sensorización ligada a contextos concretos como es el de las actividades higiénicas. Por ejemplo, es importante saber cuándo se produce la defecación o la micción para las personas dependientes, ya

que es una variable que refleja su estado de salud. Una estrategia para detectar estas actividades es medir los gases producidos:

- Los autores de (Ye, 2017) diseñaron un sensor de red inalámbrica corporal para medir los gases producidos durante la micción (NH_4) y la defecación (H_2S), y la temperatura del paciente para detectar la fiebre.
- Los autores de (Zhu, 2019) utilizando una estrategia similar, desarrollaron un sistema inteligente colocado en una cuna que incluye dos sensores de gas y un sensor de temperatura por infrarrojos. Los sensores se instalan a la altura de la posición de la cadera del bebé acostado en la cuna. Midiendo el nivel de hidrógeno y metano en la zona y el cambio de temperatura del pañal del bebé, el sistema sabe cuándo el bebé ha defecado. Cuando se produce, la alarma se enciende y las enfermeras pueden cambiar el pañal, evitando diferentes problemas de salud. De esta forma se optimiza también el tiempo del personal de enfermería.

En el caso de los pacientes que no necesitan ayuda y no están postrados en cama, los sensores pueden instalarse en el inodoro para controlar su actividad:

- Los autores de (Inoue, 2019) desarrollaron un sistema basado en un sensor de gas que detecta los olores producidos por excreción, distinguiendo si se trata de una micción o una defecación.

Las soluciones descritas con anterioridad no sólo se centran en mejorar el bienestar del usuario, sino también en ahorrar tiempo en la atención y cuidados. De hecho, incluso para las personas sanas es muy interesante conocer el número de deposiciones y analizar la orina y las heces al ser descriptores del estado de salud:

- Los autores de (Bae, 2018) diseñaron un inodoro inteligente que permite la evaluación de la orina en función de diferentes parámetros, pudiendo utilizarse para diagnosticar algunos trastornos. Además, este inodoro, que separa las heces y la orina, ahorra enormes cantidades de agua y transforma la excreción en energía renovable.

Por último, no sólo hemos de tener en cuenta las necesidades de eliminación, las propias necesidades de higiene corporal resultan fundamentales:

- Los autores de (Chengzhang, 2019) crearon un sistema de dos etapas para localizar y reconocer todas las acciones de higiene de manos en cada vídeo no recortado que analizaron con modelos de Deep Learning.
- Los autores de (Kutafina, 2015) hicieron uso de un wearable que era capaz de reconocer la actividad del antebrazo, palma y dedos de la mano, para estimar la calidad del proceso del lavado de manos con modelos de aprendizaje automático y procesamiento de señales.
- Los autores de (Jeroen, 2021) diseñaron un sistema para la detección de actividades relacionadas con la ducha y el grado en el que se realizan. Para ello, hicieron uso principalmente de la presión del agua y transmisión de señales, realizando un posterior filtrado de las ondas y una red neuronal convolucional para la clasificación.

Esta minuciosa revisión de la literatura científica actual en el contexto del proyecto genera una intrínseca motivación a la hora de desarrollar este sistema, ya que existen diversas soluciones que trabajan de forma general y/o aislada para el reconocimiento de actividades cotidianas, pero no adaptadas a un contexto concreto y extremadamente importante como son los hábitos de higiene.

Capítulo 3

3. Análisis y diseño del sistema

En este capítulo se recoge el proceso de análisis que se ha llevado a cabo para desarrollar el sistema, siendo fundamental detectar una colección de requerimientos funcionales y no funcionales para comprender al completo la naturaleza de los módulos que deben construirse y desarrollar un sistema final que sea adecuado a los criterios iniciales. De igual modo, se presenta de forma detallada el diseño del sistema, incluyendo la selección de dispositivos, el middleware de comunicación, el modelo de datos y la arquitectura.

3.1. Definición de requisitos

Como requisito software, podríamos incluir toda aquella condición que el desarrollador debe abarcar para llevar a cabo los objetivos.

3.1.1. Funcionales

Los requisitos funcionales se encargan de definir el comportamiento esperado del sistema. En nuestro caso, al tratarse de un sistema multiplataforma que se compone de diferentes módulos, indicaremos por cada uno de los requisitos funcionales para qué módulo se han definido.

ID	Nombre	Descripción
0	Detección de excreciones humanas	Debe ser capaz de detectar cuándo se produce una deposición.

	(deposiciones/orina)	
1	Detección de duchas	Debe reconocer cuándo se ha producido una ducha.
2	Detección de proximidad al baño	Debe realizar una estimación de cuándo el usuario ha estado o no en el baño.
3	Detección de movimientos concretos y orientación	Debe detectar movimientos asociados a eventos de higiene tales como lavarse la cara o los dientes.
4	Detección de eventos a través del sonido	Debe reconocer el sonido asociado a diferentes acciones que se realizan en el cuarto de baño como tirar de la cisterna, utilizar el secador, ducharse, etc.
5	Estimación de la pose de una persona	Debe detectar a la persona en la estancia así como la estimación de la pose con landmarks.
6	Recogida de datos	Debe ser capaz de realizar una recogida de datos estable y fiable en la medida de lo posible con dispositivos no invasivos. Este requerimiento engloba los anteriores (del requerimiento con ID 0 al 5).
7	Codificación de datos	Debe realizar tareas de procesamiento de los datos y codificarlos de modo que el contenido sea legible por el servidor.
8	Publicación de datos	Debe publicar todos los valores de los sensores por MQTT de acuerdo a la codificación definida legible por el servidor.
9	Persistencia de datos	Debe almacenar todos los nuevos valores que reciba por MQTT de los sensores en la base de datos.
10	Recuperación de datos	Debe recuperar los valores de la base de datos atendiendo a una consulta determinada, haciendo distintas agregaciones y

		proyecciones.
--	--	---------------

Tabla 3. Requisitos funcionales definidos

3.1.2. No funcionales

Los requisitos no funcionales imponen restricciones en diseño e implementación, definiendo aquellas cualidades que debe poseer el sistema, proporcionándole calidad.

ID	Nombre	Descripción	Requisito
0	Disponibilidad	Tiempo que el sistema ha de estar en funcionamiento estable y continuo	El servidor fog debe estar operativo a excepción de actualizaciones o depuración de errores para persistir los datos.
1	Usabilidad	Facilidad al usuario para la implantación del sistema e interpretación de sus herramientas	- Manuales de usuario concretos y adecuados a un lenguaje interpretable por cualquier usuario sin conocimientos previos de informática. - Dispositivos IoT transparentes al usuario, integrables en cualquier objeto cotidiano sin entorpecer la interacción.
2	Robustez	Adaptación a situaciones	En el caso de que alguna parte del sistema falle, no ha de entorpecer

		anómalas	al funcionamiento del resto sistema que permanezca operativo.
3	Autonomía	Independencia que posee el sistema sin requerir de la atención del usuario	• Los dispositivos IoT escogidos han de poseer una batería de larga duración, evitando continuas interrupciones al usuario. • Funcionamiento estable que no requiera continuas y molestas comprobaciones.
4	Software	Software necesario para el funcionamiento del sistema	• MongoDB 4.2 o superior • Java 11 o superior • Terminal ssh/sftp • Mosquitto • Python 3 • Apache
5	Hardware	Hardware necesario para el funcionamiento del sistema	• Centralitas de integración de sensores y envío de datos por MQTT (incluye server Apache): ◦ Raspberry PI modelo 3B o superior ◦ Placas ESP8266 • Sensores IoT ◦ Gas (MQ-8 y MQ-2) ◦ Calidad del agua (Wi-Fi Pool Kit Atlas Scientific) ◦ Wearable Xiaomi Mi Band 3 ◦ Cámara termal (Flip Lepton 3.5)

			○ Cámara espectro visible (Pi NoIR) ○ Humedad (DTH11) ○ Micrófono USB ● Servidor Fog (r. recomendados) ○ 12 GB RAM ○ 200 GB Disco Duro ○ 4 Cores
--	--	--	--

Tabla 4. Requisitos no funcionales definidos

3.2. Casos de uso

Estos diagramas identifican el comportamiento del sistema desde el punto de vista de cada uno de los actores implicados, dejando expuestas las precondiciones que deben cumplirse en cada interacción con el sistema, así como el flujo de eventos asociado a una acción u objetivo.

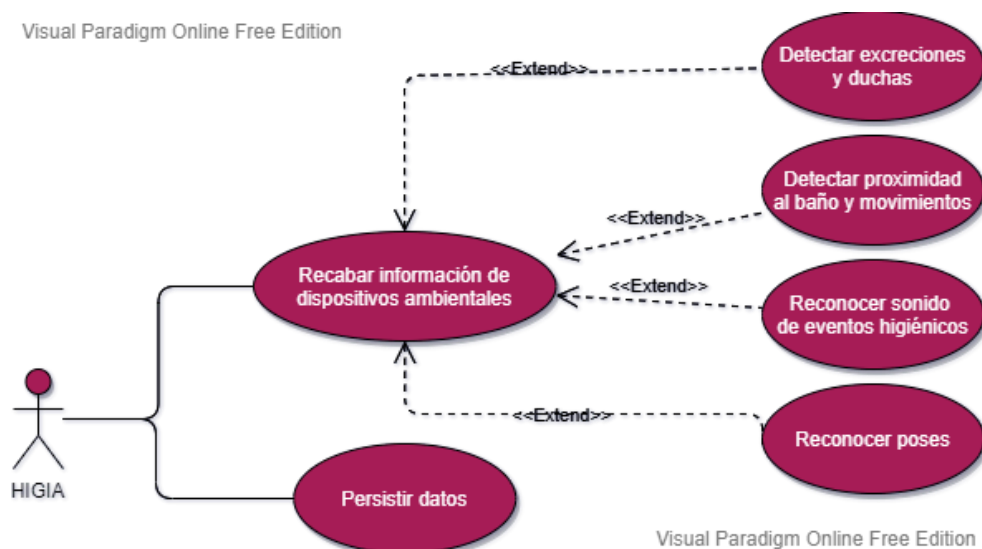


Figura 3. Casos de uso del sistema

- Detectar excreciones humanas y duchas

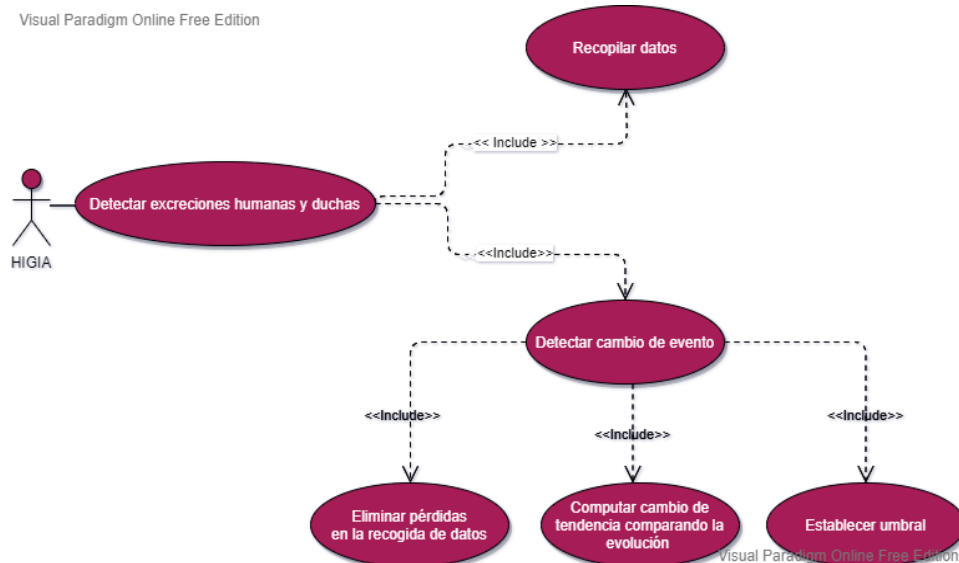


Figura 4. Caso de uso “detectar excreciones humanas y duchas”

- Detectar proximidad al baño y movimientos

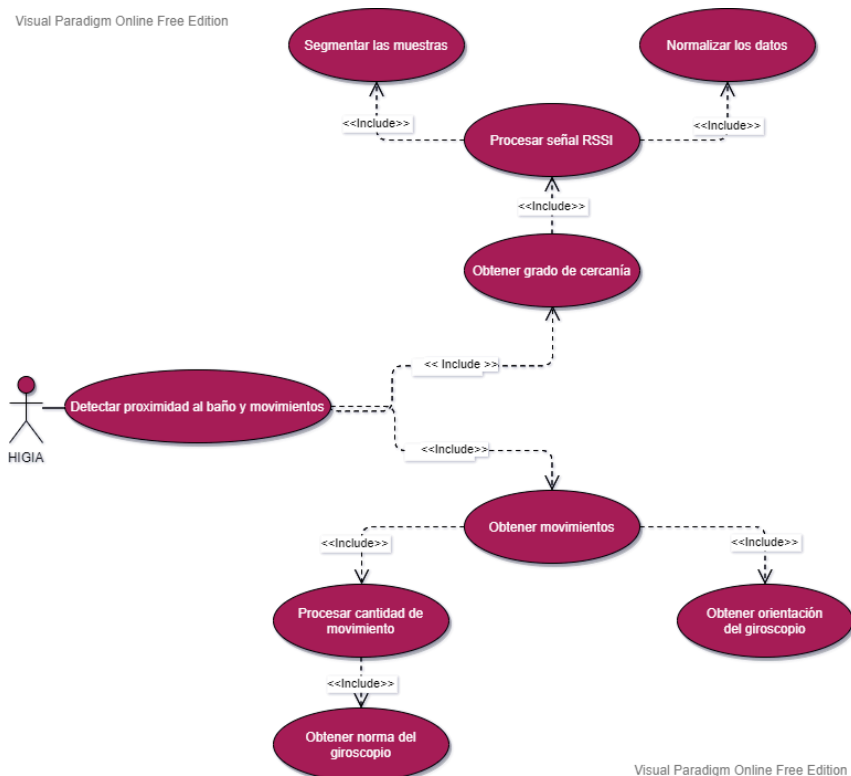


Figura 5. Caso de uso “detectar proximidad al baño y movimientos”

- Reconocer sonido asociado a eventos higiénicos

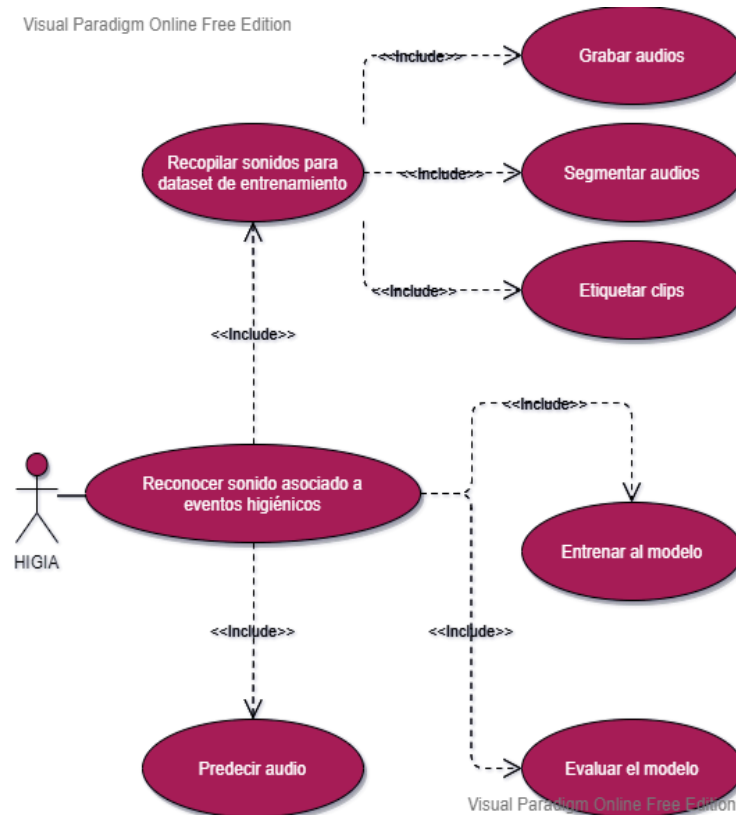


Figura 6. Caso de uso “reconocer sonido asociado a eventos higiénicos”

- Recabar información de dispositivos ambientales

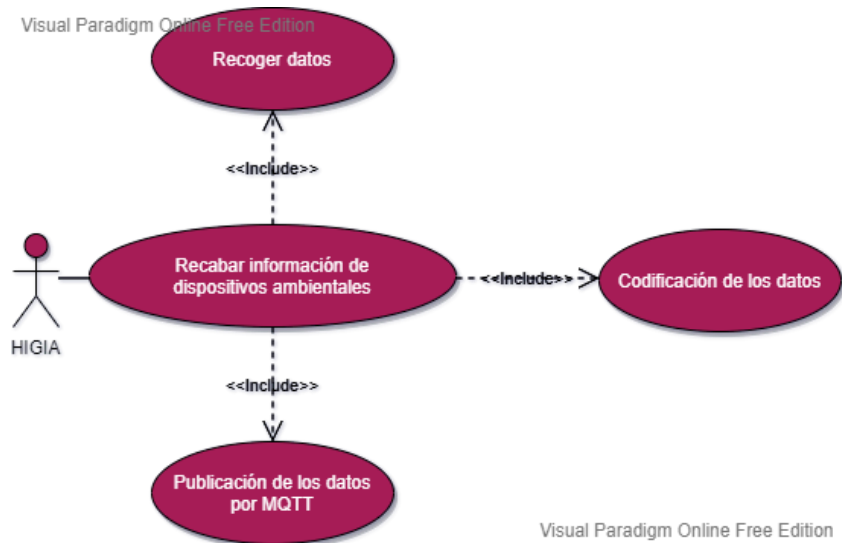
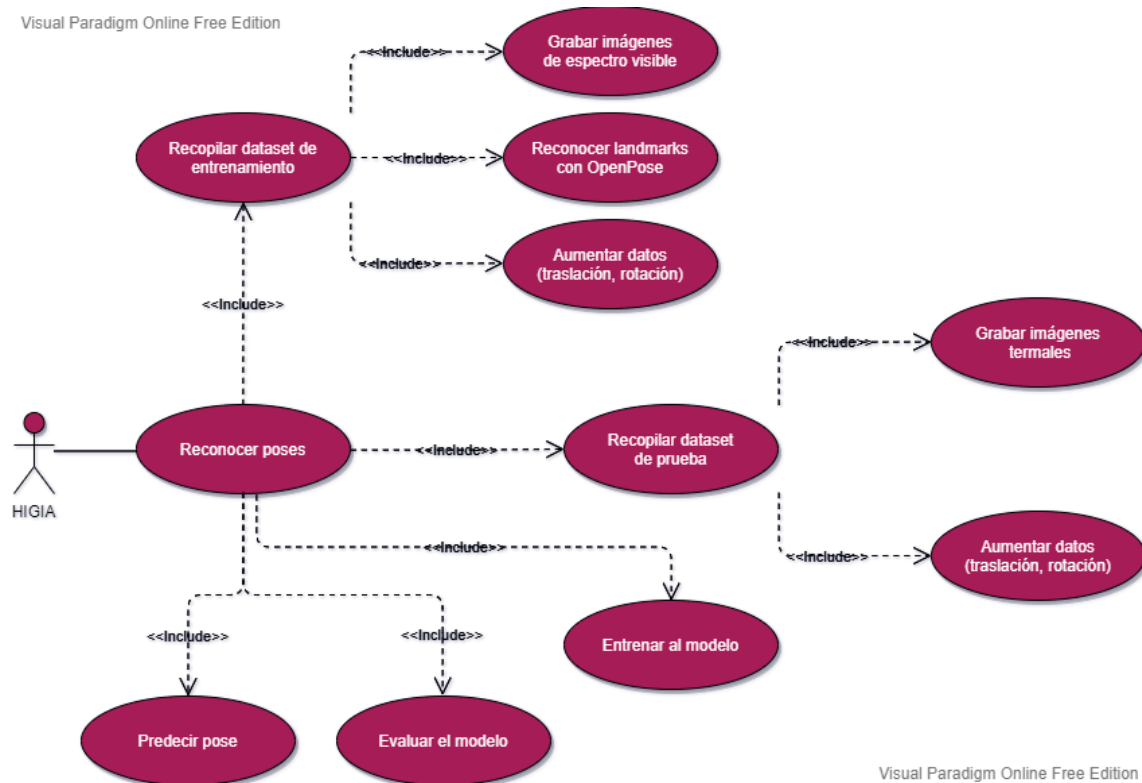


Figura 7. Caso de uso “recabar información de dispositivos ambientales”

• Reconocer poses

Visual Paradigm Online Free Edition

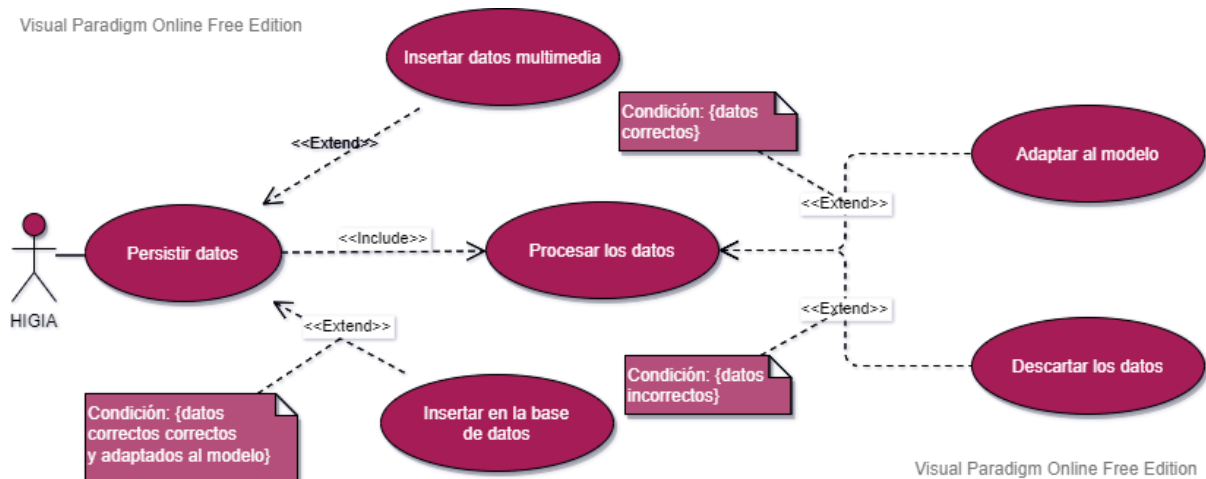


Visual Paradigm Online Free Edition

Figura 8. Caso de uso “reconocer poses”

• Persistir datos

Visual Paradigm Online Free Edition



Visual Paradigm Online Free Edition

Figura 9. Caso de uso “persistir datos”

3.3. Selección de dispositivos

En esta sección describimos de manera detallada el conjunto de dispositivos multimodales seleccionados para el desarrollo de este Trabajo de Fin de Máster. Con motivo de cubrir los requisitos definidos para nuestro sistema, hacemos uso de sensores no invasivos y de modelos de procesamiento de datos fácilmente desplegables como elementos clave para garantizar la privacidad de los habitantes, en particular con respecto al entorno de monitorización de espacios sensibles que concierne al presente proyecto.

El conjunto de sensores desplegados incluye algunos sensores ambientales (sensores de audio, gas, humedad y calidad del agua), sensores inerciales vestibles y sensores multimedia de visión y audio que permiten reconocer las actividades y eventos relacionadas con la necesidad de eliminación de residuos corporales, los hábitos de higiene y la detección de presencia preservando la privacidad.

3.3.1. Sensores de gas para la detección de deposiciones

En los gases intestinales humanos, se ha constatado que la concentración de hidrógeno (H_2) ronda un 20,9%, mientras que el contenido de metano (CH_4) suele estar en torno al 7,2%. El contenido de estos gases en el aire suele estar presente en niveles de concentración inferiores al 1%, sin embargo, estudios desarrollados con anterioridad han demostrado la viabilidad de utilizar la concentración de estos dos compuestos para detectar la defecación (Sahakian, 2010). Tomando como referencia estos datos, en este trabajo hemos optado por realizar nuestro caso de estudio con los siguientes sensores:

- **MQ-8:** Sensor adecuado para detectar concentraciones de hidrógeno en el aire. El sensor puede detectar concentraciones de gas hidrógeno en cualquier lugar de 100 - 10000 partículas por millón (MQ8, 2021). El dispositivo está disponible comercialmente por aproximadamente 4,50 euros.



Figura 10. Sensor MQ-8

- **MQ-2:** Se trata de un sensor de gas combustible. Permite detectar el gas metano en un rango de 300 - 10000 partículas por millón (MQ2, 2021). Su precio ronda los 2,90 euros.

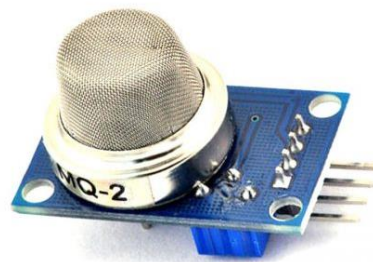


Figura 11. Sensor MQ-2

El material de sensorización de este par de sensores es el SnO_2 , para el que la conductividad es menor en un ambiente limpio y mayor cuando hay presencia de estos gases. Los sensores seleccionados proporcionan una alta sensibilidad y un rápido tiempo de respuesta.

3.3.2. Sensor de humedad para detectar duchas

Teniendo en cuenta los factores ambientales que se ven afectados de forma inherente, el aumento de la humedad en un baño se produce por una alta concentración de vapor de agua caliente respecto a las condiciones estándar. Este hecho puede ser asociado con alta probabilidad a la actividad de la ducha, ya que

provoca la condensación de una gran cantidad de vapor cálido, que se impregna en paredes y suelos.

Para reconocer este evento, hemos seleccionado el sensor DHT11 de temperatura y humedad, que tiene una salida de señal digital calibrada. Al utilizar la técnica de adquisición de señales digitales y la tecnología de detección de temperatura y humedad, garantiza una alta fiabilidad y una excelente estabilidad a largo plazo (Srivastava, 2018).

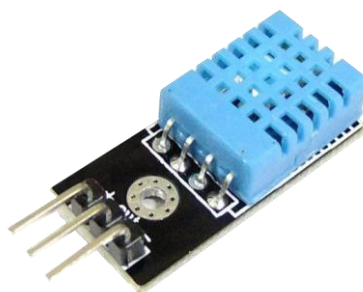


Figura 12. Sensor DHT11

3.3.3. Sensores de calidad del agua para detección de orina y/o deposiciones

En el ámbito estudiado y puesto a prueba en este proyecto, la integración de sensores de diferente índole es fundamental para detectar eventos de higiene en los diferentes espacios del cuarto de baño. Es por ello que, además del uso de sensores de gas, se ha integrado un conjunto de sensores comerciales para medir la calidad del agua de la empresa Atlas Scientific. El sensor se denomina "Wi-Fi Pool Kit Quality Monitoring" con un precio que ronda los 350€ (<https://atlas-scientific.com/kits/wi-fi-pool-kit/>).

El kit incluye sensores de pH, ORP (Oxidation Reduction Potential asociada a la suciedad del agua) y temperatura. La tensión de funcionamiento del dispositivo es de 5V y el consumo de corriente es de 280 mA a 5V. El dispositivo se define como IP64, que proporciona un rendimiento robusto contra el polvo y sumergible por ser

resistente al agua. Integra una placa ESP8266 como controlador de los sensores, lo que permite codificar soluciones embebidas para la monitorización del agua. La placa incluye conexión WiFi, lo cual ofrece la posibilidad de transferir datos mediante TCP, HTTP o MQTT en tiempo real. Atlas Scientific incluye el código y las instrucciones para iniciar el proyecto desde cero. A continuación, se listan los componentes específicos del Wi-Fi Pool Kit:

- Circuito de pH. El sensor de pH determina la acidez/alcalinidad del agua con una resolución de 0,01 de acidez.
- Circuito de Potencial de Oxidación-Reducción (ORP). El rango de ORP del sensor es de +/-1100 mV.
- Circuito de temperatura de resistencia (RTD).
- Sonda de pH, sonda de ORP y sonda de temperatura PT-1000.
- 2 bolsas de 20 ml de solución de calibración de ORP 225 mV.
- 6 bolsas de solución de calibración de 20 ml (2 de pH 4, 2 de pH 7 y 2 de pH 10).

En la figura 13, describimos el dispositivo Wi-Fi Pool Quality Monitoring y detallamos los componentes que lo integran:

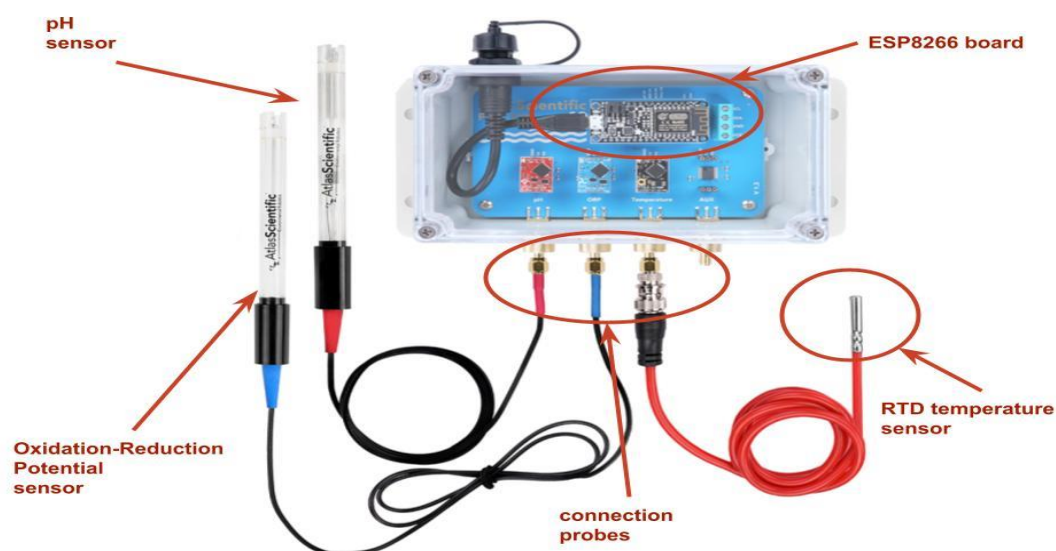


Figura 13. Wi-Fi Pool Kit Quality Monitoring. En círculos rojos se detallan los principales componentes: Placa ESP8266, sondas de conexión y sensores RTD, pH y ORP

La tasa de recogida está configurada por defecto, siendo el tiempo de retardo para actualizar las medidas de 1 segundo. Sin embargo, de forma práctica para la lectura, cómputo y estimación de los valores post calibración de referencia se obtiene una tasa real de muestras por cada 3 segundos, obteniendo una medida de los 3 sensores.

3.3.4. Wearable para detectar movimientos, orientación y proximidad al baño

Se ha demostrado que los sensores inerciales incluidos en pulseras de actividad resultan muy útiles para los movimientos de la mano (López, 2019). El procesamiento de esta información inercial por medio de reglas de conocimiento (Medina, 2017) puede actuar como descriptor de actividades tales como lavarse los dientes, manos, etc. directamente relacionadas con la higiene. Además, es posible realizar una estimación de la proximidad a una estancia determinada si hacemos uso del RSSI (Received Signal Strength Indicator). Entre la amplia gama de dispositivos comerciales, cuyas capacidades y sensorización crece cada año, para este Trabajo de Fin de Máster hemos escogido el dispositivo **Xiaomi Mi Band 3**, por su precio mínimo (20-25 €), dimensiones reducidas (17,9 x 46,9 x 12 mm), bajo peso (20 g incluida la pulsera) y batería de larga duración (aprox. 20 días) que incluye características interesantes: acelerómetro, giroscopio, intensidad, etc. Además, la comunicación basada en BLE (Bluetooth Low Energy) hace posible recopilar datos periódicamente en un entorno ambiental inteligente y no invasivo. En la figura 14, se muestran el pequeño tamaño y diseño de la pulsera de actividad Mi Band 3.



Figura 14. *Xiaomi Mi Band 3, tamaño del dispositivo wearable y correa para la muñeca*

3.3.5. Sensores de audio para detectar eventos higiénicos

La detección de diferentes eventos mediante sensores de gas, humedad, calidad del agua y movimiento (pulseras de actividad) tiene un enorme potencial para detectar eventos cotidianos relacionados con la higiene. Sin embargo, fusionando estas modalidades de sensores con el reconocimiento de sonido ambiental es posible aumentar el rendimiento del sistema y, por tanto, aumentar la precisión y la detección de eventos. En este trabajo, nos hemos centrado en los métodos de reconocimiento de audio para la detección de eventos de agua (aseo, ducha, secador y grifo), así como de ocupación humana (hablar o mover/abrir objetos) y de silencio.

Para conseguir nuestro objetivo hemos hecho uso de un sensor de audio conectado a una placa IoT para recoger y reconocer los diferentes eventos sonoros. La placa IoT es una Raspberry Pi Modelo 3B (Upton, 2014), que por un precio reducido tiene capacidad de computación para aprendizaje automático, incluyendo modelos de aprendizaje profundo (Monteiro, 2018). Los sensores de audio integrados son micrófonos de bajo coste con un conector USB, que proporcionan conectividad plug-and-play con la placa mencionada con anterioridad. En la siguiente figura, se muestran ambos dispositivos desplegados en un baño. A la izquierda, podemos ver una Raspberry Pi Modelo 3B con micrófono USB integrado que aporta al dispositivo la capacidad de recoger y reconocer sonidos ambientales. A la derecha, una aplicación móvil para etiquetar eventos con NFC con motivo de facilitar la recogida y el etiquetado de datos:

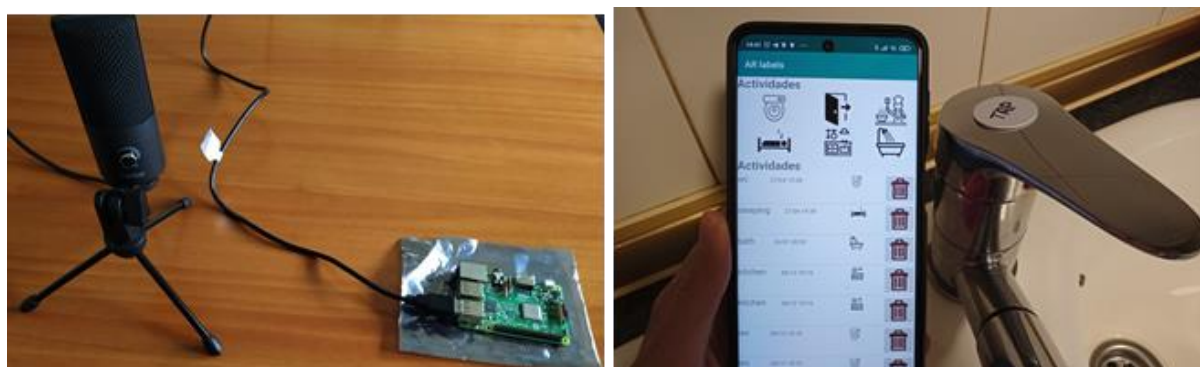


Figura 15. Raspberry Pi Modelo 3B con micrófono USB y aplicación móvil para etiquetado de datos

3.3.6. Cámaras de visión termal y espectro visible

La detección de humanos y su pose es un reto que se lleva abordando años en la comunidad científica por su utilidad para reconocimiento de actividades. Sin embargo, la privacidad sigue siendo primordial en el caso de la monitorización de usuarios, por lo que las soluciones actuales en el mercado que hacen uso de cámaras de espectro visible no resultan viables en espacios como el cuarto de baño. En el caso de estudio que hemos llevado a cabo, se ha optado por un dispositivo IoT que integra un sensor de doble visión térmica que garantiza, de forma parcial y en función del contexto y sensibilidad, la privacidad del usuario.

Para poder crear modelos de reconocimiento adaptados al sensor termal, se ha hecho uso en la etapa de entrenamiento y aprendizaje de un sensor de espectro visible, que no estará desplegado en entornos de despliegue finales. Los dispositivos de visión se han integrado en una placa Raspberry Pi, cuyas configuraciones similares han proporcionado resultados alentadores en contextos sanitarios (Cooney y Bigun, 2017). Los sensores de visión seleccionados son NoIR Camera Module V2 y PureThermal 2 Smart I/O Module + FLIR Lepton 3.5. En la siguiente figura, mostramos la configuración final de los dispositivos:

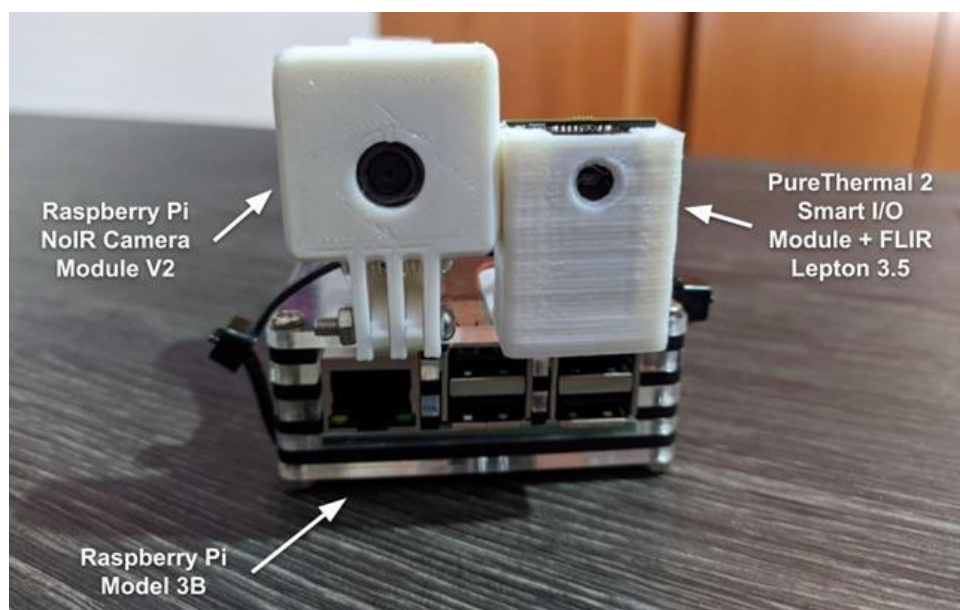


Figura 16. Raspberry Pi NoIR Camera Module V2 + PureThermal 2 Smart I/O Module + FLIR Lepton 3.5 + Raspberry Pi Model 3B

- **Módulo de E/S inteligente PureThermal 2 + FLIR Lepton 3.5.** FLIR Lepton es un dispositivo de cámara LWIR (Long Wavelength Infra-Red) con capacidad radiométrica, más pequeño que una moneda de un euro. Su precio es de unos 350 euros. Con conjuntos de planos focales de 160 x 120 píxeles activos, Lepton se integra fácilmente y captura datos de temperatura precisos, calibrados y sin contacto en cada píxel de cada imagen.
- **Módulo de cámara Raspberry Pi NoIR V2.** Este módulo de cámara se diferencia del módulo de cámara Raspberry Pi normal en que no emplea un filtro de infrarrojos (NoIr = No InfraRed), lo que permite la capacidad de recoger correctamente las imágenes en la oscuridad con iluminación infrarroja. Su precio es de unos 30 euros.

3.4. Conectividad de dispositivos

Uno de los elementos claves en Internet de las Cosas y Entornos Inteligentes es la conectividad de los dispositivos que permite recolectar de forma inalámbrica los datos de los sensores. Existe una gran variedad de protocolos, que han sido previamente analizados, para desarrollar un sistema multiplataforma y multidispositivo. A continuación, se describen los protocolos utilizados para conectar los dispositivos en este Trabajo de Fin de Máster:

WiFi

Wi-Fi es una tecnología de comunicación que utiliza ondas de radio como medio para la conexión. Mediante una antena, las señales se transmiten a un dispositivo inalámbrico. El dispositivo envía sus solicitudes al enrutador que es un decodificador, donde los datos se envían / reciben desde Internet. Este protocolo es el más utilizado hasta la fecha, sin embargo, la investigación afirma con seguridad que Wi-Fi ya no es factible como protocolo para domotización del hogar.

Una ventaja que tiene Wi-Fi es su alta velocidad de transmisión de datos, además de que configurar una topología es relativamente económico en comparación con los demás protocolos. Se ha de tener en cuenta de que el uso de este protocolo puede parecer ideal, ya que está universalmente aceptado. La cantidad de dispositivos compatibles también es una ventaja, aunque debemos tener en cuenta que, teóricamente, se podrían conectar 255 dispositivos de forma simultánea; sin embargo, a partir de 5 dispositivos se produce una lentitud significativa. Si nos centramos en dispositivos IoT de bajo consumo y envío de datos, se estima que podrían alcanzarse unos 45 dispositivos trabajando en consonancia de manera fluida.

Como herramienta para la automatización del hogar, el mayor defecto de Wi-Fi es que consume demasiada energía. Otro punto negativo de este protocolo es que posee topología de estrella, por lo que si falla nuestro nodo central, provocaría el colapso de toda la red, haciendo que el rendimiento de los demás dispositivos se viese afectado. Respecto a seguridad, el hecho de poseer acceso a uno de los nodos nos daría acceso a toda la red, por lo que no resulta muy recomendable.

Bluetooth Low Energy

Bluetooth es una tecnología inalámbrica para intercambiar datos a corta distancia. Aunque este protocolo se creó inicialmente para dispositivos pequeños, no es factible usarlo para domotización; ya que, aunque consume una cantidad notablemente menor de energía que Wi-Fi, todavía consume demasiado. Aquí es donde entra Bluetooth Low Energy.

BLE se implementa en una red de malla similar a la de los dos protocolos mencionados anteriormente. Aparentemente, Bluetooth corrige las debilidades de Wi-Fi en el consumo de energía y la implementación topológica de la red. De hecho, Bluetooth es uno de los protocolos para automatización del hogar más utilizados junto a Z-Wave y ZigBee. Su frecuencia le permite alcanzar una velocidad de datos

de 1 Mbps. El ahorro de energía se consigue mayormente al dormir al dispositivo entre cada transmisión de datos.

Su mayor desventaja es que solo admite hasta siete dispositivos por conexión. Teniendo en cuenta que los hogares inteligentes tienen al menos 15 dispositivos de media, aumenta el coste de integración de nuevos dispositivos, además de ser ínfimamente práctico. En comparación con los demás protocolos, BLE es el que menor alcance posee. Si bien esto no es un problema en el entorno doméstico, agrava el problema al ser altamente susceptible a interferencias, especialmente en áreas donde hay gran densidad de población interconectada. Respecto a la seguridad, tampoco es un protocolo libre de ciberataques.

Bluetooth Low Energy se ha integrado como protocolo de comunicación con la pulsera Xiaomi Band. En nuestro caso, hemos optado en este Trabajo de Fin de Máster por la sincronización de datos por BLE a una placa IoT que retransmite los datos por WiFi. De este modo, pretendemos visibilizar la posibilidad de crear un sistema en el que todos los dispositivos sean tratados de forma homogénea, incluso si la manera en la que recogen y envían la información del entorno no es la misma.

A modo de resumen, se han recogido en una tabla todas las características principales de Wi-Fi y BLE. Este modo de presentar la información nos permite comparar las especificaciones entre los protocolos que se comentaron con anterioridad.

Especificaciones	Wi-Fi	Bluetooth Low Energy
Diseño centrado en	Conexión a WLAN para dispositivos	Intercambio de datos en distancias cortas
Estándar IEEE	802.11	802.15.1

Tipo de red	Estrella	Malla
Red	Wi-fi	Bluetooth
Distancia	20 metros	10-100 metros
Número máximo de nodos conectados	Depende del router	7
Banda operativa	2.4 y 5 GHz	2.4 a 2.485 GHz
Espectro ensanchado	Espectro ensanchado de secuencia directa (DSSS)	Espectro adaptativo de salto de frecuencia (AFH)
Rendimiento	900 Mbps	24 Mbit/s
Datos	Transporte	Intercambio
Seguridad	Cifrado de acceso protegido Wi-Fi (WPA2)	Confidencialidad, autenticación y derivación de claves.
Consumo de energía	Alto	Bajo
Modulación	Modulación por desplazamiento de fase en cuadratura (QPSK)	Modulación de desplazamiento de frecuencia gaussiana (GFSK)

Tabla 5. Comparativa de especificaciones de los protocolos Wi-Fi y Bluetooth Low Energy

Las placas de IoT ligeras (ESP8266) conectadas a los sensores ligeros (gas, humedad, etc), así como las placas de IoT de mayor capacidad (Raspberry Pi), transmiten los datos en tiempo real por WiFi, mientras que el wearable los transmite a través de BLE.

3.5. Middleware de comunicación, persistencia y arquitectura

La arquitectura de este trabajo Fin de Máster se inspira en el paradigma de la computación **edge/fog**, en el que los modelos de aprendizaje automático se incorporan a las placas de IoT para reconocer patrones en tiempo real, muy cerca del sensor donde se recogen los datos, **evitando así la transmisión de datos sensibles y de elevado tamaño y frecuencia como pueden ser las imágenes y el sonido**. Para ello, es necesario establecer unos protocolos de comunicación donde los datos y eventos se envíen en tiempo real de forma apropiada. A continuación, mostramos los protocolos y componentes apropiados para la integración de estas fuentes heterogéneas ambientales, wearables y multimedia.

3.5.1. MQTT

En IoT, la transmisión de mensajes entre dispositivos es esencial. En el caso de MQTT, se ha pensado para operar en máquinas de bajo consumo como un protocolo ligero basado en TCP/IP. Su diseño ofrece un protocolo de mensajería de publicación-suscripción con la mayoría de los requisitos mínimos de ancho de banda, proporcionando flexibilidad en los patrones de comunicación. En el protocolo MQTT, el publicador publica mensajes, y los usuarios que desean conocer la información que envía el publicador, se suscriben. Los mensajes se publican sobre *topics* que pueden considerarse el tema del mensaje. Los *topics* pueden tener diferentes niveles, de este modo, los suscriptores pueden suscribirse solo a aquellos temas particulares que están relacionados con ellos; permitiéndoles descartar la

información que es irrelevante en su ámbito (Soni, 2017), como muestra la siguiente figura:

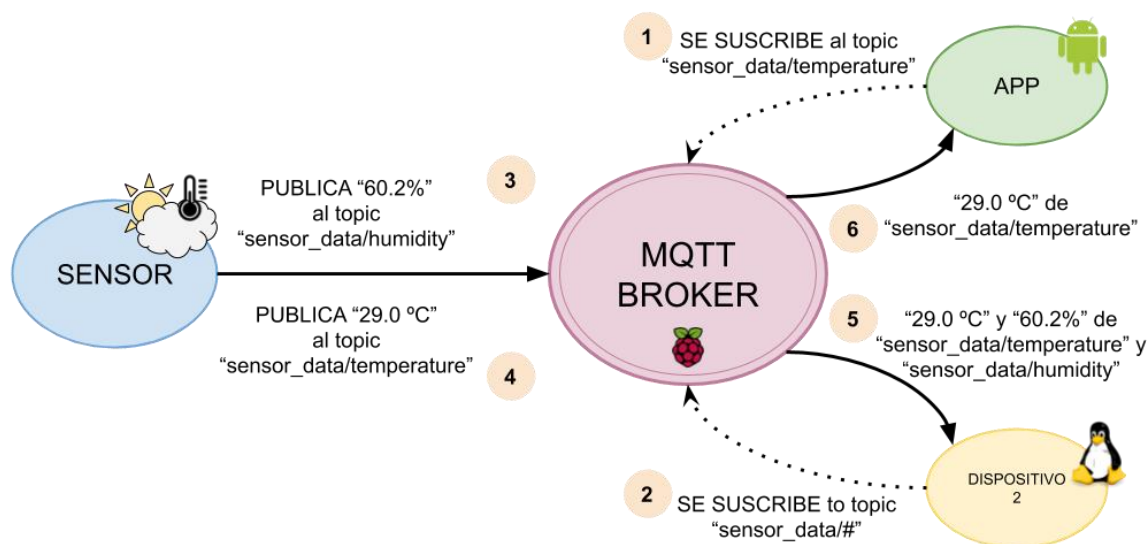


Figura 17. Esquema de funcionamiento del protocolo MQTT

La arquitectura típica de MQTT se puede dividir en dos componentes principales:

- **Ciente:** un publicador o suscriptor. Siempre establece la conexión de red al servidor (bróker).
- **Bróker:** controla la distribución de información y es el principal responsable de recibir todos los mensajes de los publicadores, filtrarlos, decidir quién está interesado en ellos y luego enviar los mensajes a todos los clientes suscritos.

A continuación, listamos las principales ventajas que presenta el protocolo MQTT, siendo una estupenda elección para IoT:

- Es un protocolo ligero.
- Seguridad de los datos (la autenticación se puede implementar en el intermediario).
- El cifrado SSL se puede utilizar junto con este protocolo, ayudando en el cifrado de extremo a extremo.

- Eficiente en la multidifusión de mensajes.
- La función QoS ayuda a especificar garantías de acuse de recibo y envío de mensajes en base a cada contexto de repetición.

El uso de MQTT es clave en la integración de sensores heterogéneos bajo una capa de fusión de datos que distribuye datos en tiempo real en entornos inteligentes. En el caso de este trabajo Fin de Máster, **se publican mediante el protocolo MQTT a cualquier receptor que quiera procesarlos (proceso de machine learning o proceso de persistencia)** la siguiente información:

- **Datos de sensores**
 - Datos brutos de los sensores ambientales (gas, humedad, calidad del aire)
 - Datos de sincronización periódica de la pulsera de actividad (pasos, datos inerciales)
 - Eventos (no datos) de las imágenes y audio de fuentes multimedia
- **Eventos derivados del procesamiento de sensores**
 - Cálculo de intensidad de movimiento del sensor inercial de la pulsera
 - Detección de fuente de origen del sonido
 - Cálculo de la pose del usuario de la cámara termal
 - Valor anormal de un sensor ambiental
 - Cambio de tendencia o anomalía temporal de un sensor de calidad de agua

3.5.2. HTTP

HTTP es un protocolo basado en TCP/IP que permite las transferencias de información a través de archivos (XHML, HTML...) en la World Wide Web para para OBTENER, EDITAR, PUBLICAR y ELIMINAR datos. Cada URL se denomina **solicitud**, mientras que los datos que se le devuelven se denominan **respuesta**.

Estas solicitudes se componen de 4 elementos básicos que le aportan una estructura bien definida para asegurar su funcionamiento:

1. **Ruta:** es la url que solicita. Determina el recurso que está solicitando.
2. **Método:** El método es el tipo de solicitud que se envía al servidor, proporcionando significado a la solicitud que está realizando. Los principales métodos se listan a continuación:
 - a. **GET:** Obtiene un recurso de un servidor. Si se realiza este tipo de petición, el servidor busca los datos que solicitó y se los devuelve. En otras palabras, se realiza una “lectura”.
 - b. **POST:** Crea un nuevo recurso en un servidor. Si se realiza este tipo de petición, el servidor crea una nueva entrada en la base de datos y notifica si la creación se realizó correctamente. En otras palabras, se realiza una “inserción”.
 - c. **PUT/PATCH:** Actualizan un recurso en un servidor. Si realiza este tipo de petición, el servidor actualiza una entrada en la base de datos y notifica si la actualización se realizó correctamente.
 - d. **DELETE:** Elimina un recurso de un servidor. Si se realiza este tipo de petición, el servidor elimina una entrada en la base de datos y notifica si la eliminación se realizó correctamente.
 - e. **HEAD:** Es idéntico a GET, pero el servidor no devuelve el contenido en el HTTP response. Cuando se envía un HEAD request, significa que sólo se está interesado en el código de respuesta y los headers HTTP, no en el propio documento.
3. **Encabezados:** Son pares de propiedades y valores que están separados por dos puntos. Por ejemplo, un encabezado indicaría al servidor que espere contenido JSON/MULTIMEDIA.
4. **Datos:** Contienen la información que desea enviar al servidor. Esta opción sólo se utiliza con peticiones POST, PUT/PATCH o DELETE.

En este proyecto se incluye un **Servidor Apache de HTTP para transmitir y almacenar los datos de fuentes multimedia**. El caso de estos datos multimedia (audio o imágenes), debido a su elevado peso y frecuencia de transmisión **no deben enviar datos brutos por MQTT** realizando un doble proceso de transmisión cuando se generan las muestras:

- **Envío de datos por protocolo HTTP de los datos en formato raw del contenido de la imagen o sonido.** La persistencia de los mismos en el servidor Apache devolverá una URL con el contenido de destino del evento.
- **Publicación del evento por MQTT de la fuente de imagen o sonido** que incluirá la URL por HTTP con los datos raw de la misma.

3.5.3. MongoDB

Cuando hablamos de bases de datos tendemos a pensar en SQL y el modelo de bases de datos relacional. Sin embargo, las bases de datos NoSQL son una alternativa potente. MongoDB es la base de datos no relacional más popular con diferencia. ¿Qué es NoSQL? El término NoSQL se refiere a Not Only SQL. Plantea modelos de datos específicos y esquemas flexibles. Los esquemas se adaptan a los requisitos de las aplicaciones más modernas.

MongoDB es un sistema de base de datos NoSQL orientado a documentos de código abierto y escrito en C++. En lugar de guardar los datos en tablas, lo hace en estructuras de datos BSON (similar a JSON) con un esquema dinámico. Al estar escrito en C++, tiene una gran capacidad para aprovechar los recursos de la máquina y adaptarse a nuestras necesidades. Como características principales, destacamos:

- Cada base de datos contiene colecciones que a su vez contienen documentos. Cada documento puede ser diferente el uno del otro.

- La estructura del documento es similar a la manera en que los desarrolladores construyen sus objetos en sus respectivos lenguajes de programación.
- Las filas (o documentos como se llama en MongoDB) no necesitan tener un esquema definido previamente.
- El modelo de datos disponible en MongoDB le permite representar estructuras complejas más fácilmente.
- MongoDB es muy escalable. Muchas empresas alrededor de todo el mundo definen clusters con más de 100 nodos. Cada nodo podría contener millones de documentos.

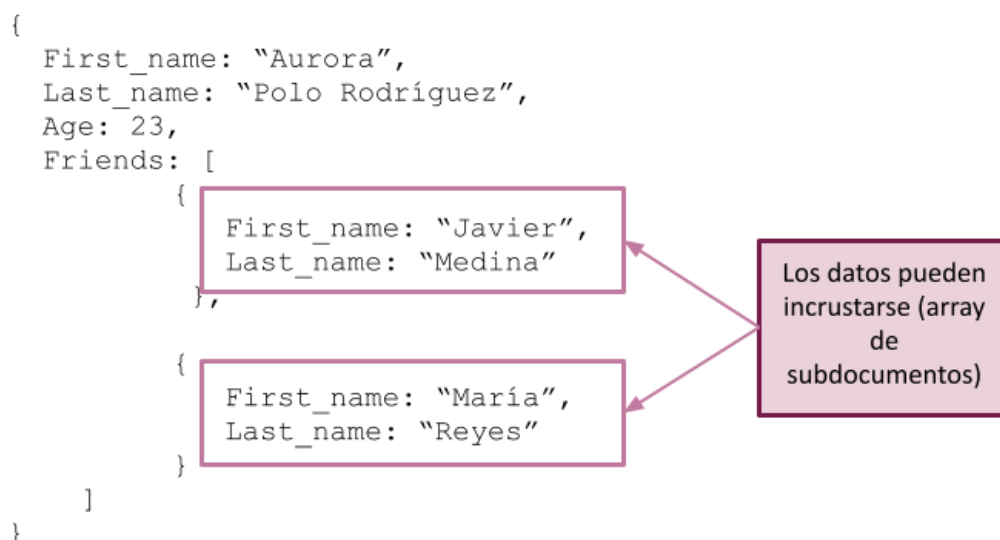


Figura 18. Ejemplo de estructura de documentos en MongoDB

Para la inserción y consulta de datos en MongoDB existen varias herramientas que facilitan la tarea de gestión de datos. Entre ellas, destacamos la herramienta Morphia 2.0 que ofrece un *Object Document Mapper* para MongoDB y que permite realizar el modelado, inserción, consulta, mapeo y agrupación de datos mediante una perspectiva de Orientación a Objetos. Gracias a la integración de este componente, las tareas de gestión de la Base de Datos se reducen y se realizan de forma más escalable y ágil.

En el caso de este proyecto, los datos pertenecientes a todos los sensores se persisten en MongoDB. En el caso de los ligeros, el valor de los mismos se incluye en este campo, y en los multimedia (audio e imágenes) incluyen la URL al servidor HTTP donde se encuentran los frames y sonidos en bruto. Esta división, a su vez, tiene su punto fuerte en la privacidad, ya que datos extremadamente sensibles y deben persistirse en entornos separados por seguridad.

3.6. Diseño del modelo de representación y persistencia de datos

En esta subsección, se describe un modelo de representación que se ha diseñado para realizar el almacenamiento de los datos de los sensores. Este modelo tiene como objetivo mostrar un esquema general que pueda ser utilizado para representar sus datos con independencia de la naturaleza heterogénea de los sensores. Además, gracias a su versatilidad, el procesamiento de los sensores realizados por los modelos de Machine Learning también permite ser representado bajo este modelo. Su estructura es sencilla:

- **Cada sensor estará compuesto por una o más propiedades.** Por ejemplo, para nuestro sensor “WifiPoolKit” se definen las siguientes propiedades: pH, ORP y temperatura. La propiedad puede hacer referencia a un procesamiento de datos del sensor, como “descenso de PH”, “puntos de landmark”.
- **Marca de tiempo en formato UTC (hora universal coordinada).** UTC es el estándar de tiempo primario por el cual el mundo regula los relojes y la hora. Está dentro de aproximadamente 1 segundo del tiempo solar medio a 0° de longitud, y no está ajustado para el horario de verano. En algunos países, se utiliza el término Greenwich Mean Time. Esta marca de tiempo está asociada con el momento en que el sensor recolectó un valor.
- **Cada propiedad tendrá un valor asociado.**
 - En los dispositivos ligeros, contendrá el valor de medida del sensor (intensidad, datos inerciales, pH, humedad, partículas por millón de un gas, etc...).

- En el caso de dispositivos multimedia, contendrá la URL del servidor HTTP donde se almacena los datos raw de sonido o imagen. Por ejemplo, `http://localhost:80/thermal/image1.jpg`
- En el caso de un procesamiento de datos de sensores, el valor o valores resultantes del procesamiento: `{"pH agua":7.9}`, `{"sonido secador de pelo":0.8}`, `{"posición x,y":145,13,312, ....}`
- De forma optativa, se puede incluir la localización del sensor que permita geolocalizar dicha información. En el caso que concierne a este proyecto, hemos prescindido de esta información, puesto que no ofrecía un aporte significativo, pero puede ser muy útil para representaciones, por ejemplo, de sensores en smart cities.

En la siguiente figura, se representa el modelo de datos propuesto junto a un ejemplo de éste:

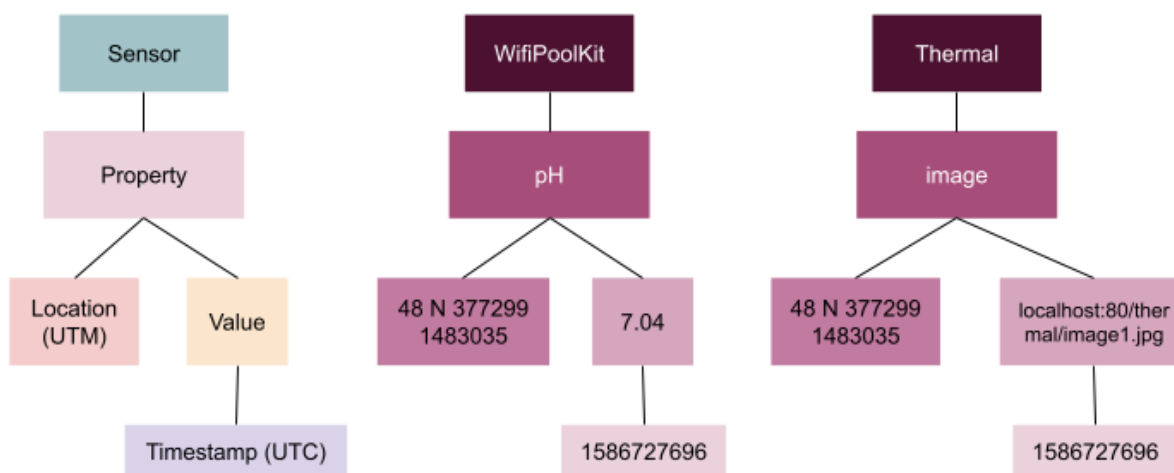


Figura 19. Modelo de representación de datos

Este modelo tiene un doble propósito dentro del sistema propuesto:

- **Servir como esquema de representación de datos de comunicación en el middleware**, cuando se envían datos por MQTT. Tanto la codificación como la decodificación de datos basada en

mensajes se realiza mediante este esquema que se traduce a formato JSON.

- **Servir como modelo de persistencia.** Un suscriptor por MQTT a todos los topics del entorno inteligente, realiza una persistencia en MongoDB de los datos bajo este esquema unificador de sensores heterogéneos. De esta forma, esta Base de Datos contiene todos los eventos que se han generado en el entorno, sirviendo para propósitos de generación de datasets y visualización de datos/dashboards de otros componentes de explotación.

3.7. Arquitectura general

Una vez conocidos los protocolos de conexión y de comunicación, así como los elementos que integran nuestra propuesta, presentamos un esquema de la arquitectura que se implementará en los dispositivos y componentes de este trabajo.

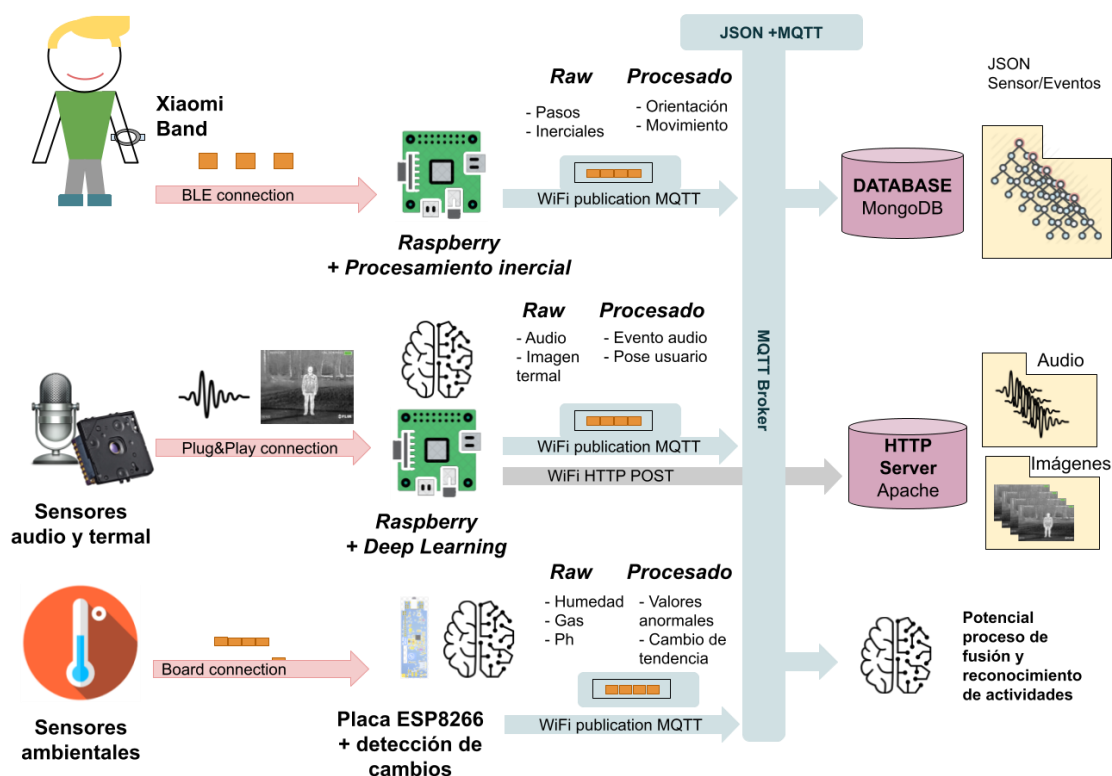


Figura 20. Arquitectura detallada del sistema propuesto

La figura anterior refleja la esencia de esta propuesta, un sistema multiplataforma que configura una arquitectura de diversos componentes que se despliegan en un entorno inteligente. La arquitectura se inspira en el paradigma de la computación fog, en el que los modelos de aprendizaje automático se incorporan en las placas de IoT para reconocer patrones en tiempo real, muy cerca del sensor donde se recogen los datos al que están conectados de forma cableada o inalámbrica. Esta propuesta de arquitectura ha sido publicada en el siguiente trabajo científico:

Polo-Rodriguez, A., Cruciani, F., Nugent, C., & Medina-Quero, J. (2021, April). Recognition of Hygiene Activities by Means of Multimodal Sensors. In The International Research & Innovation Forum (pp. 89-98). Springer, Cham.

Los sensores ambientales, en concreto los de gas, humedad y calidad del agua, se conectan a las placas IoT ESP8266 para recoger y detectar los eventos utilizando un umbral sencillo aplicado a la señal observada procedente de las lecturas de los sensores. En el caso de los sensores de audio e imagen, actúan del mismo modo que los anteriores, pero en este caso se encuentran conectados a una placa IoT Raspberry Pi Modelo 3B donde los datos se transmiten a un servidor HTTP previo envío del evento de MQTT.

Los sensores inerciales vestibles se utilizan para detectar los gestos de las manos, en particular para detectar las manos levantadas y los movimientos que pueden estar relacionados con las actividades de lavado. Estos datos inerciales se transmiten a través de Bluetooth Low Energy.

Por último, se propone el reconocimiento de audio para detectar eventos de agua y ocupación, así como la estimación de la pose del usuario mediante reconocimiento de imágenes termales. Este procesamiento se realiza en placas IoT como Raspberry Pi, procesando así todas las señales de audio e imagen de forma local y enviando los resultados del procesamiento por MQTT sin necesidad de transmitir los datos brutos mediante el canal de publicación-suscripción.

Capítulo 4

4. Implementación

En esta sección se describe la implementación del sistema sobre los componentes y diseño descrito en las secciones anteriores. En primer lugar, se detalla el proceso de integración de cada conjunto de sensores dentro de la arquitectura y middleware propuesto. Posteriormente, se define el procesamiento al que se someten los datos multimedia e inerciales. Por último, se explica el despliegue de los diferentes servidores y la persistencia en éstos.

4.1. Integración de sensores heterogéneos y multimodales

4.1.1. Sensores de gas y humedad

Posibilitando responder a las demandas de la sensorización específica en el contexto del cuarto de baño, se han integrado una serie de dispositivos ad-hoc con placas IoT WiFi. Estos dispositivos no son comerciales y precisan de un esfuerzo adicional: la conexión de los sensores a la placa mediante pin digital analógico y la programación de ésta para recoger los datos obtenidos por los sensores y enviarlos mediante MQTT.

La placa IoT seleccionada fue **NodeMCU**, una pequeña placa WiFi compatible con Arduino. Está basada en **ESP8266**, con sus pines en los laterales. Como ventajas, presenta un regulador de tensión integrado, LED y botón de reset integrados y programación a través del Micro-USB.

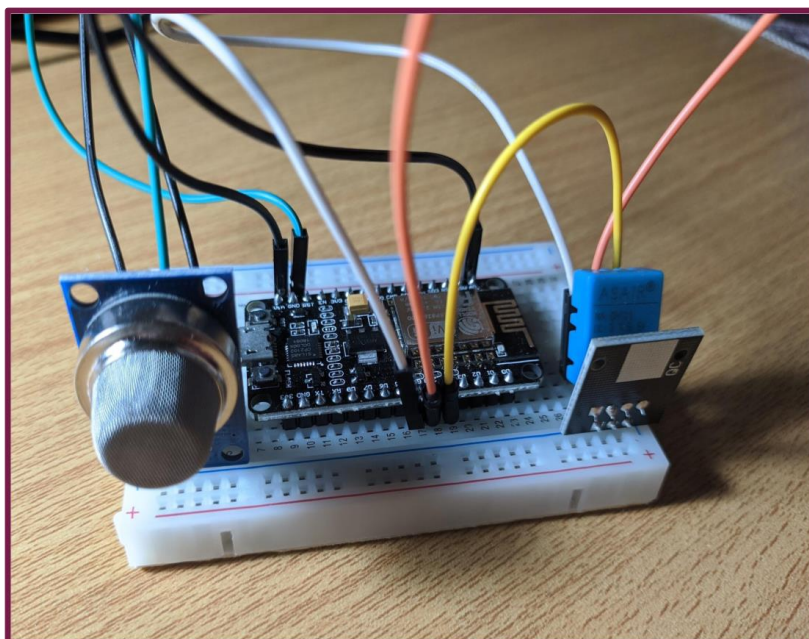


Figura 21. Placa NodeMCU ESP8266 con sensores integrados (sensor de gas MQ8 y sensor de temperatura y humedad DHT11)

La salida proporcionada por estos sensores es una resistencia analógica, que es leída por la placa ESP8266 donde se conecta el sensor. En nuestro caso, utilizamos una regresión exponencial para calcular el valor de la concentración de PPM. Las constantes para configurar los valores de la ecuación y obtener la concentración de partículas por millón (PPM) de cada gas vienen definidas según la documentación de datos técnicos correspondiente a cada sensor. En la siguiente figura, se muestra un fragmento del código fuente donde se puede visualizar dicha información:

```
//Set math model to calculate the PPM concentration and the value of constants
MQ2.setRegressionMethod(1); //_PPM = a*ratio^b
MQ2.setA(987.99); MQ2.setB(-2.162); // Configurate the ecuation values to get LPG concentration
/*
Exponential regression:
Gas    | a      | b
H2     | 987.99 | -2.162
LPG    | 574.25 | -2.222
CO     | 36974  | -3.109
Alcohol| 3616.1 | -2.675
Propane| 658.71 | -2.168
*/
```

```
//Set math model to calculate the PPM concentration and the value of constants
MQ8.setRegressionMethod(1); //_PPM = a*ratio^b
MQ8.setA(8000000000000); MQ8.setB(-6.666); // Configure the equation values to get CH4 concentration

/*
  Exponential regression:
  GAS   | a       | b
  H2    | 976.97  | -0.688
  LPG   | 10000000 | -3.123
  CH4   | 8000000000000 | -6.666
  CO    | 2000000000000000 | -8.074
  Alcohol | 76101  | -1.86
*/
```

Figura 22. Constantes definidas para los sensores MQ2 y MQ8, respectivamente

Como se comentó anteriormente, estas placas incluyen conexión Wi-Fi, lo que nos será útil para publicar los datos brutos y la detección de eventos en tiempo real bajo el protocolo MQTT (Soni, 2017). Se presenta la conexión de los sensores MQ a la placa en la siguiente figura:

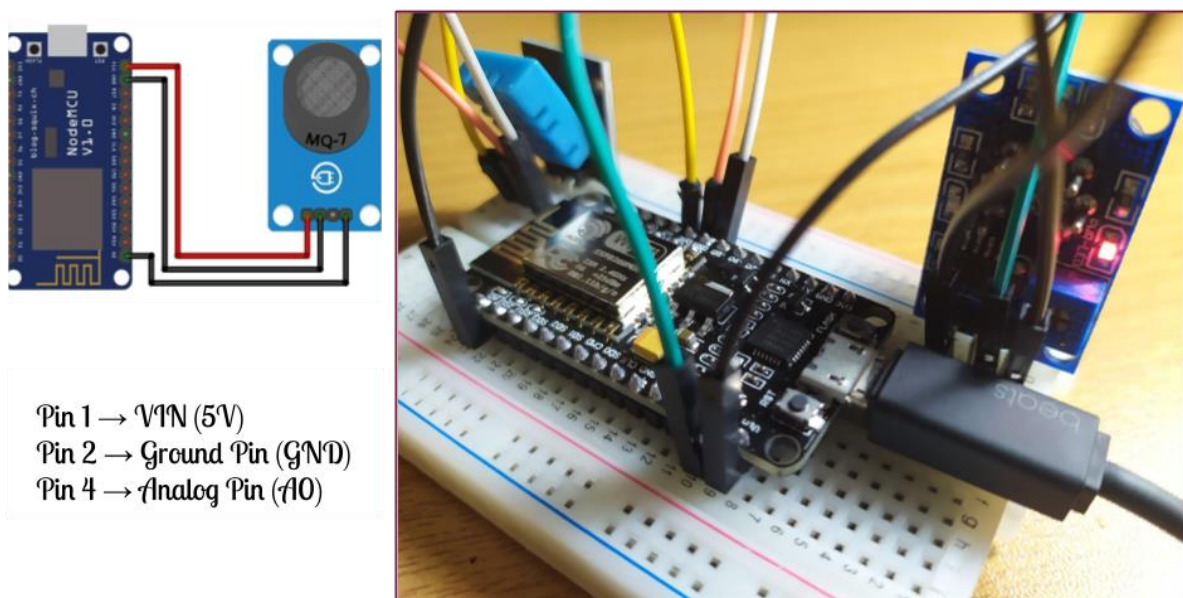


Figura 23. Sensor MQ integrado en placa NodeMCU ESP8266

De manera análoga al anterior, se presenta la conexión del sensor DHT11 a la placa:

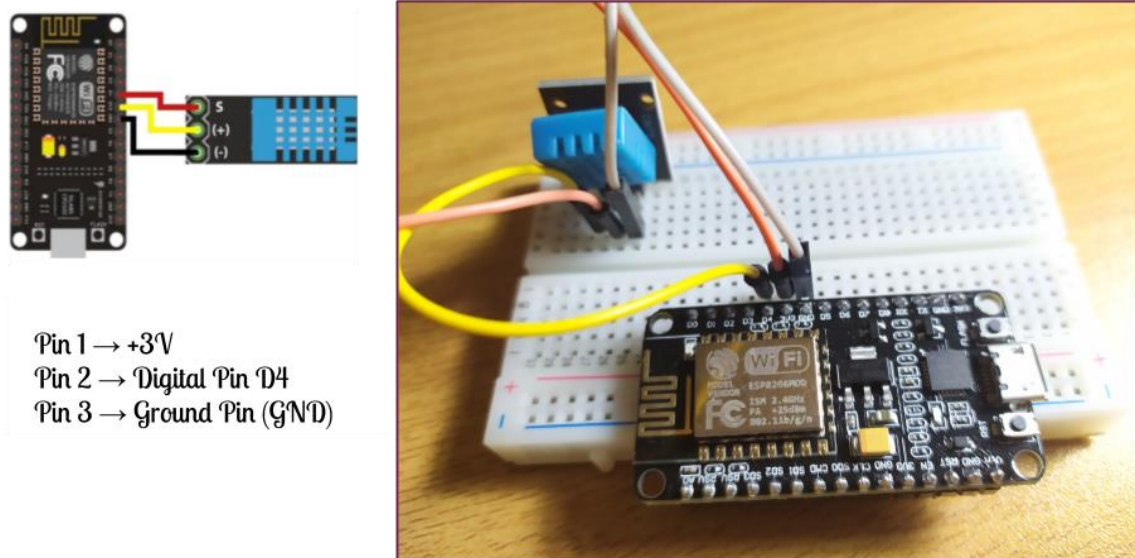


Figura 24. Sensor DTH11 integrado en placa NodeMCU ESP8266

Para la publicación de los datos de los sensores por MQTT, se hizo uso de la **librería PubSubClient**. Asimismo, para evitar aumentar la complejidad del sistema, se unificó la lectura de datos de los sensores y el envío de estos en un mismo archivo. De este modo, fueron necesarios los siguientes puntos:

1. Definición de la IP del servidor y el *topic* en el que se publican los datos.
2. Conexión con el servidor.
3. Lectura periódica de los datos, tratamiento de éstos y publicación del mensaje.



Figura 25. Lectura y publicación de datos por MQTT en placa NodeMCU ESP8266 con sensores integrados

4.1.2. Sensores de calidad del agua

El Wi-Fi Pool Kit está diseñado para supervisar y controlar a distancia la calidad y composición química de algunos componentes del agua. Estos sensores funcionan utilizando un Feather HUZZAH de Adafruit como su CPU, programado con el IDE de Arduino y con una placa ESP8266 como su transmisor Wi-Fi. La principal diferencia con los sensores de gas y humedad respecto a su puesta en marcha versa sobre la calibración posterior a la implementación.

El diseño general de la placa de circuito impreso es bastante sencillo. La CPU se alimenta y se programa a través del conector USB montado en el panel. El

regulador de voltaje de la CPU suministra al bus de alimentación de la placa con 3,3V a 500 mA de pico. Todos los sensores conectados funcionan a 3,3V.

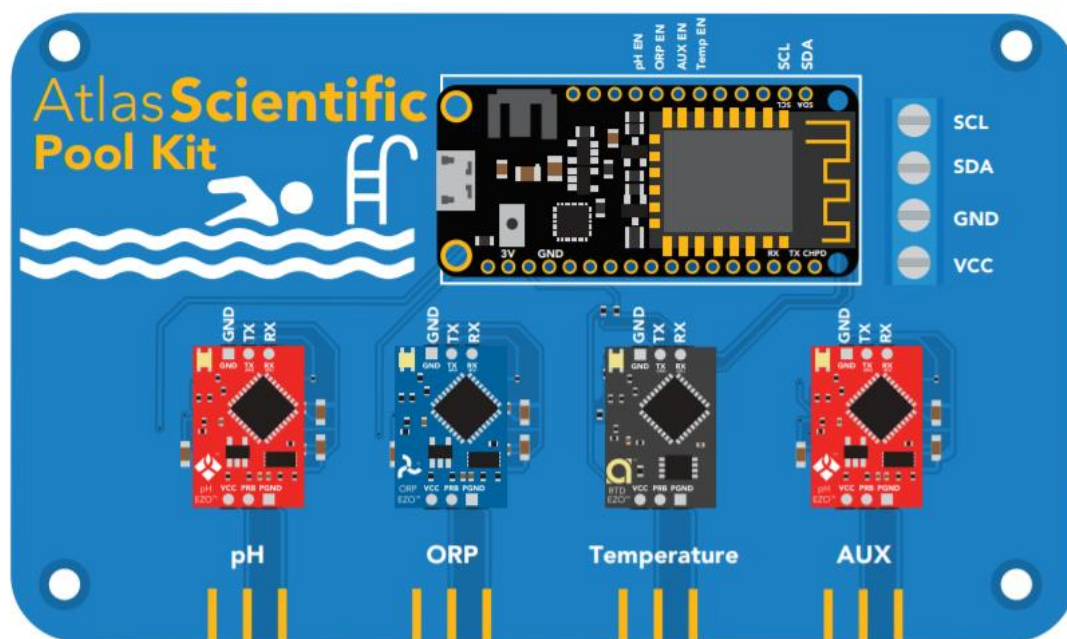


Figura 26. Detalles de la placa Atlas Scientific Pool Kit

Por defecto, los datos de los sensores se cargan a ThingSpeak™, una plataforma gratuita de adquisición y visualización de datos basada en la nube. Sin embargo, este kit de sensores ha sido diseñado también para ser fácilmente modificado, por lo que su código fuente está disponible en [Github](#). Una vez importamos esta librería, podemos ajustar el código libremente. En nuestro caso, **enviamos los datos a través de MQTT cada tres segundos** utilizando la misma librería indicada en la sección anterior (PubSubClient). El formato de los valores tomados por los sensores se muestra en la siguiente figura:

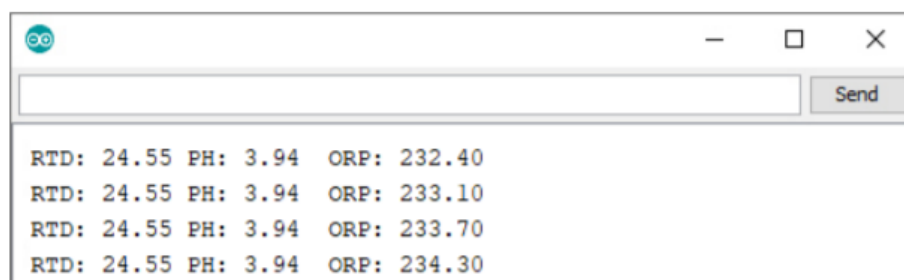


Figura 27. Fragmento de muestras tomadas por los sensores de calidad del agua

La calibración posterior a la configuración de los sensores es fundamental para que las lecturas sean precisas. Como indicamos en la sección 3.3.3, este kit incluye soluciones para realizarla. Se han de seguir las instrucciones del fabricante, mostradas en la siguiente figura:

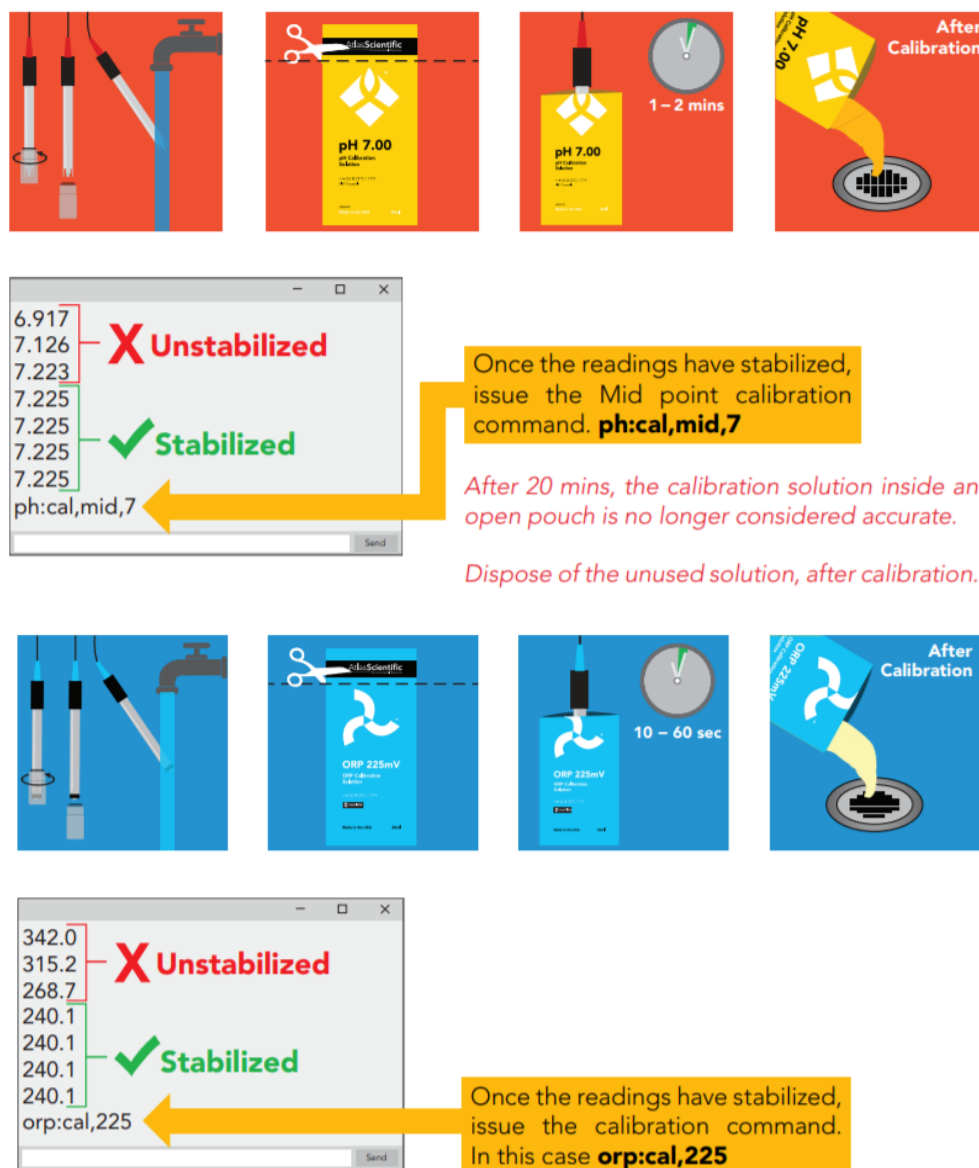


Figura 28. Calibración de los sensores de calidad del agua

Es importante tener en cuenta que en el caso del pH, se debe repetir esta calibración con pH 4 y pH 10, no solo 7. En el caso del sensor de temperatura no es

necesario, pero puede introducirse en agua hirviendo para garantizar mayor precisión.

4.1.3. Wearable (sensor inercial + RSSI)

El wearable de Xiaomi Mi Band 3 permite la transmisión de los datos mediante BLE. La recolección de estos datos se ha realizado mediante una centralita Raspberry Pi y Python de forma periódica, utilizando como base una librería disponible en [GitHub](#). Se han integrado las siguientes métricas de recolección de datos del dispositivo wearable:

- **Giroscopio**

Para manejar enviar los datos en tiempo real del giroscopio es necesario crear una conexión con el dispositivo que recibe datos cada 0.5 segundos

```
def start_heart_and_gyro(self, sensitivity, callback):
    self.heart_measure_callback = callback
    self.gyro_raw_callback = callback

    self.send_gyro_start(sensitivity)
    self.send_heart_measure_start()

    heartbeat_time = time.time()
    while True:
        self.wait_for_notifications_with_queued_writes(0.5)
        self._parse_queue()
        if (time.time() - heartbeat_time) >= 12:
            heartbeat_time = time.time()
            self.send_heart_measure_keepalive()
            self.send_gyro_start(sensitivity)
```

Figura 29. Hebra para activar la conexión de giroscopio de la Xiaomi Band 3 en streaming

La conexión con la pulsera se mantiene activa enviando los siguientes comandos de reconexión:

```
def send_gyro_start(self, sensitivity):
    if not self.gyro_started_flag:
        self._log.info("Starting gyro...")
        self.write_req(self._sensor_handle, BYTEPATTERNS.start)
        self.write_req(self._steps_handle, BYTEPATTERNS.start)
        self.write_req(self._hz_handle, BYTEPATTERNS.start)
        self.gyro_started_flag = True
    self.write_cmd(self._char_sensor, BYTEPATTERNS.gyro_start(sensitivity))
    self.write_req(self._sensor_handle, BYTEPATTERNS.stop)
    self.write_cmd(self._char_sensor, b'\x02')
```

Figura 30. Envío de comandos de cabecera para iniciar y mantener el streaming d datos en tiempo real de la Xiaomi Band 3

De forma que los datos se reciben por bluetooth y procesan obteniendo la componente x, y, z de giroscopio:

```
def _parse_raw_gyro(self, bytes):
    gyro_raw_data_list = []
    for i in range(2, 20, 6):
        gyro_raw_data = struct.unpack("3h", bytes[i:(i+6)])
        gyro_dict = {
            'gyro_raw_x': gyro_raw_data[0],
            'gyro_raw_y': gyro_raw_data[1],
            'gyro_raw_z': gyro_raw_data[2]
        }
        gyro_raw_data_list.append(gyro_dict)
    return_tuple = ["GYRO_RAW", gyro_raw_data_list]
    return return_tuple
```

Figura 31. Extracción de valores X, Y, Z del buffer de datos de Xiaomi Band 3

- **Intensidad, pasos**

La pulsera de actividad Xiaomi Band 3 incluye un parámetro de intensidad de actividad que también resulta útil para conocer la cantidad de movimiento. Estos parámetros (steps, pasos, frecuencia cardiaca si está activada) no se recuperan en

streaming (tiempo real), si no con consultas de sincronización que pueden realizarse desde la Raspberry Pi cada determinado tiempo.

```
i = 1
while i < len(data):
    index = int(pkg) * 4 + (i - 1) / 4
    timestamp = self.device.first_timestamp + timedelta(minutes=index)
    self.device.last_timestamp = timestamp
    # category = int.from_bytes(data[i:i + 1], byteorder='little')
    category = struct.unpack("<B", data[i:i + 1])
    intensity = struct.unpack("B", data[i + 1:i + 2])[0]
    steps = struct.unpack("B", data[i + 2:i + 3])[0]
    heart_rate = struct.unpack("B", data[i + 3:i + 4])[0]

    print("{}: category: {}; acceleration {}; steps {}; heart rate {}:".format(
        timestamp.strftime('%d.%m - %H:%M'),
        category,
        intensity,
        steps,
        heart_rate)
    )

    i += 4
```

Figura 32. Extracción de pasos, intensidad y HR de Xiaomi Band 3

- **RSSI**

Otro parámetro interesante es la intensidad de señal Bluetooth de la pulsera con una estancia determinada. Para ello hemos usado una centralita Raspberry Pi que realiza un escaneo periódico que recupera los dispositivos conectados y la intensidad de señal RSSI de los mismos.

El RSSI es una escala para medir el nivel de potencia de las señales recibidas por un dispositivo en las redes inalámbricas. Gracias a esta escala podemos saber dentro de un entorno controlado si una persona se encuentra cerca de un emisor, por lo que, si este emisor se encuentra en una habitación, sería factible controlar si se encuentra dentro de ella o no. Para llevar a cabo este fin hemos desarrollado dos procesos Python los cuales han sido desplegados en una

Raspberry Pi y quedan a la escucha de obtener la escala de potencia de un emisor embebido en una pulsera de un habitante.

El procesamiento de la intensidad de señal (RSSI) de la pulsera mediante BLE permite reconocer la cercanía a habitaciones y espacios de la vivienda. El escaneo se realiza mediante dispositivos Raspberry Pi localizados en las habitaciones de interés que realizan un barrido cada 15 segundos y envían la intensidad de la señal (RSSI) en tiempo real. Así los valores recogidos por la pulsera llegan mediante BLE a la centralita Raspberry Pi, los datos se envían por MQTT en tiempo real para su posterior procesamiento.

```

connected = False
MAC_ADDR = "E6:FD:82:DB:AC:EB"
atHome = 0

client = mqtt.Client()
client.connect("192.168.1.2",1883)

while connected == False:
    try:
        print ('Attempting to connect to ', MAC_ADDR)
        band = MiBand3(MAC_ADDR, debug=True)
        band.setSecurityLevel(level = "medium")
        band.authenticate()
        connected = True
    except (IOError,BTLEException) as e:
        print ('Impossible to connect')
    time.sleep(5)

client.publish("sensor_data/atHome", str({'atHome': 1}))

while True:
    if connected == True:
        try:
            print(band.get_steps())
            client.publish("sensor_data/steps", str(band.get_steps()))
            print(band.get_heart_rate())
            client.publish("sensor_data/hr", str(band.get_heart_rate()))
            time.sleep(5)
        except (IOError,BTLEException) as e:
            connected = False
            client.connect("192.168.1.2",1883)
    if connected == False:
        time.sleep(5)
        client.publish("sensor_data/atHome", str({'atHome': 0}))
        try:
            band = MiBand3(MAC_ADDR, debug=True)
            band.setSecurityLevel(level = "medium")
            band.authenticate()
            connected = True
            atHome = 1
            client.connect("192.168.1.2",1883)
            client.publish("sensor_data/atHome", str({'atHome': 1}))
        except (IOError,BTLEException) as e:
            print 'Not home yet'

```

Conexión MQTT al servidor

Conexión BLE a Xiaomi Mi Band 3

Publicación de mensajes por MQTT

Figura 33. Sincronización de datos del dispositivo wearable mediante BLE y MQTT

4.1.4. Sensores multimedia

La integración de sensores binarios y wearables es muy útil para el reconocimiento de actividades y eventos. Sin embargo, aumentar la sensorización de los entornos inteligentes con dispositivos multimedia, tales como sensores de audio o de vídeo, permiten incrementar las prestaciones de estos sentidos artificiales que integramos en el espacio. En esta subsección describimos la integración de los mismos de forma conjunta por su sencillez, ya que estos dispositivos incluyen drivers plug and play que permiten interactuar con ellos de forma inmediata desde computadores, incluyendo Raspberry Pi. En concreto, ambos dispositivos se conectan por USB a la placa IoT permitiendo recuperar los datos de forma directa usando python. A continuación, mostramos el sencillo código para capturar las muestras de 3 segundos desde un dispositivo de sonido.

```
import sounddevice as sd
from scipy.io.wavfile import write
from IPython.display import clear_output
fs = 44100 # Sample rate
seconds = 3 # Duration of recording

myrecording = sd.rec(int(seconds * fs), samplerate=fs, channels=2)
sd.wait() # Wait until recording is finished
write('temp.wav', fs, myrecording) # Save as WAV file

sig_temp , sr = librosa.load('temp.wav')
```

Figura 34. Captura de muestras de sonido desde Python

En el caso específico de los sensores de visión, se ha implementado una aplicación python para recoger los datos de entrenamiento en ambos dispositivos en tiempo real, proporcionando una alineación automática entre las imágenes recogidas.

```

with Lepton() as thermal:
    camera=cv2.VideoCapture(0)
    while True:
        img = thermal.grab().astype(np.float32)
        ret,imgCam=camera.read()
        # Rescale to 8 bit
        img=255*(img - img.min()/(img.max()-img.min()))
        if i>0: cv2.imwrite(os.path.join(path , str(i-1)+".jpg"), img.astype(np.uint8))
        if i>0: cv2.imwrite(os.path.join(path2 , str(i-1)+".jpg"), cv2.applyColorMap(img.astype(np.uint8))
        #cv2.imshow('lepton', img.astype(np.uint8))
        cv2.imwrite(os.path.join(path3 , str(i)+".jpg"), cv2.resize(imgCam, dsize))
        time.sleep(1)
        print(str(i) + "\n")
        i+=1

```

Figura 35. Captura de muestras de imágenes de las cámaras PureThermal2 y Raspberry Pi NoIR V2 en Python

4.2. Reconocimiento de actividades con modelos de Machine Learning

4.2.1. Procesamiento de sensores ligeros y wearables

4.2.1.1. Procesamiento de RSSI y datos inerciales del wearable

El procesamiento de la intensidad de señal (RSSI) de la pulsera mediante BLE permite reconocer la cercanía a habitaciones y espacios de la vivienda. El escaneo se realiza mediante dispositivos Raspberry Pi localizados en las habitaciones de interés que realizan un barrido cada 15 segundos y envían la intensidad de la señal (RSSI) en tiempo real.

- La segmentación en 1 minuto de todas las muestras recogidas en ese intervalo se realiza usando el operador de media para todos los valores que se encuentren recogidos en ese minuto. En caso de no obtener ningún valor RSSI se devuelve el valor- ∞ .
- Para normalizar los datos, se ha modelado un término que describe la cercanía del usuario a la baliza en un grado entre [0,1] que otorga la interpretabilidad al término de cercanía.

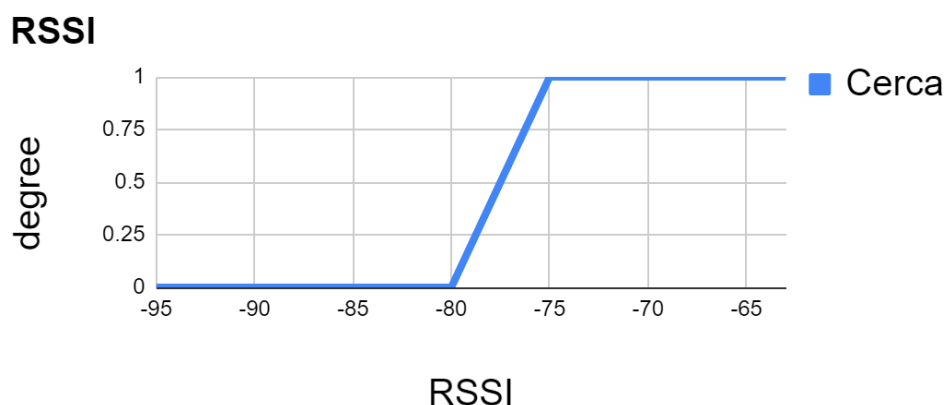


Figura 36. Normalización de datos de RSSI

En la siguiente gráfica se observa la evolución de la cercanía al cuarto de baño a lo largo de un día de recogida de datos mostrando la estancia de un usuario en el mismo:



Figura 37. Evolución de cercanía al cuarto de baño basado en RSSI

En el caso de los datos inerciales del wearable, hemos procesado varias métricas derivadas del giroscopio. Los datos del giroscopio se describen con un componente tridimensional $v_i=(x,y,z)$ que se computan en un tiempo determinado tiempo t_j :

- **Cantidad de movimiento derivado de la normal del giroscopio**
 $\sqrt{(x^2+y^2+z^2)}$
- **Orientación del giroscopio en determinados gestos.** En el ejemplo del cuarto de baño, determina cuando el usuario ha subido la mano (la componente x es mayor de 180°).

Estas métricas tienen como característica relevante que pueden ser computadas en placas de IoT de forma sencilla, pudiendo realizarse procesamiento en los nodos Edge que reciben los datos y enviando las mismas como información de alto nivel. En la Figura 38, mostramos un ejemplo del procesamiento en un caso real.



Figura 38. Arriba) Cantidad de movimiento basada en la norma del giroscopio), centro) valor de los componentes de los ejes X,Y,Z y abajo) detección de la elevación basada en el eje X>180°

4.2.1.2. Procesamiento de sensores de calidad del agua, gas y humedad

En esta subsección, detallamos el procesamiento del flujo de datos de los sensores ligeros para calcular los cambios de nivel tanto en agua (PH, ORP, temperatura) como en la medición de sensores de gas (variación de partículas por millón) y en la concentración de un sensor de humedad. Entre el amplio abanico de métodos de Machine Learning, proponemos una sencilla detección de cambios de

eventos (Khan, 2016) basada en ventana móvil deslizante (Min, 2001) de baja complejidad computacional que evita los tediosos y pesados modelos de clasificación de eventos.

En primer lugar, esta simplicidad se apoya en la estabilidad del flujo de datos de los sensores, que son estables (a corto y medio plazo). Además, el sencillo modelado de los flujos de datos permite computar el cambio de eventos dentro de placas IoT, como la placa ESP8266 que hemos utilizado para la integración de estos sensores. De este modo, este método de procesamiento de datos es compatible con los enfoques de Edge Computing, permitiendo distribuir en tiempo real sólo la información relevante, en este caso el cambio de evento de la calidad del agua, gas o humedad, y así evitar compartir datos brutos innecesarios.

Cada flujo de sensores (s) puede representarse como un conjunto de medidas unidimensionales $s = \{m_i\}$, donde cada medida contiene un valor v_i , que representa una medición, y una marca de tiempo t_i que determina el punto en el tiempo de la medida $m_i = \{v_i, t_i\}$.

La principal operación básica en el flujo de datos s es el cálculo de una media de las medidas dentro de una ventana temporal definida por dos intervalos de tiempo $[-\Delta, +\Delta]$ que se define en segundos. A partir de un momento actual dado t^* podemos calcular la media de los valores dentro de esa ventana deslizante en ese tiempo determinado:

$$\bar{v}_{t^*} [-\Delta, +\Delta] = 1/n \sum_{k=t^*-\Delta}^{t^*+\Delta} v_k$$

donde n es el número de muestras dentro de la ventana temporal.

El método para detectar los cambios de eventos es común para todos los flujos de datos:

1. Eliminamos las pérdidas en la recogida de datos o valores anómalos.

Los datos de los sensores suelen ser estables, pero existen algunas pérdidas o errores en las mediciones debido a la imprecisión en la lectura de los sensores. Estas lecturas erróneas se etiquetan como "no datos" y se traducen en un valor de -1 (fuera del rango de valores posibles). Para sustituir el valor de pérdidas esporádicas $v_i = -1$, se computa un valor sintético establecido por la media de los datos dentro de una ventana temporal en el intervalo $[-\Delta, +\Delta]$, que se define en segundos desde la marca de tiempo de la pérdida t_i :

$$v_i = \bar{v}_i [-\Delta, +\Delta]$$

- ### 2. Computamos el cambio de tendencia comparando la evolución de valores a medio plazo respecto a los de corto plazo.
- La detección de cambios de eventos en un flujo de datos se calcula (incremento o decremento) como la diferencia de medias móviles deslizantes entre el término medio plazo y el de corto plazo. Para ello, se definen dos ventanas temporales $[-\Delta, +\Delta]$ $[-\Delta, +\Delta]$ para definir el intervalo medio y el corto, respectivamente, siendo el intervalo de tiempo del término medio inferior al corto: $-\Delta < -\Delta, +\Delta > +\Delta$. Así, para una determinada marca de tiempo t_i , somos capaces de calcular la evolución de los cambios en los datos de los sensores s_i obteniendo la variación del flujo de datos $\Delta s = \{\Delta m_i\}$, donde $\Delta m_i = \{\Delta v_i, t_i\}$.

$$\Delta v_i = \bar{v}_i [-\Delta, +\Delta] - \bar{v}_i [-\Delta, +\Delta]$$

cuyos valores Δv_i representan el cambio relativo entre el medio y el corto plazo.

- ### 3. Establecemos un umbral α para reconocer los cambios de los valores de medio-corto plazo.
- Este paso es necesario para decidir si el cambio es

suficientemente relevante o no, por lo que para cada flujo de datos de los sensores definimos un umbral α que determina si se genera el evento para cada evolución Δv_i en un timestamp t_i :

$$\alpha_i = \Delta v_i < \alpha$$

Finalmente, obtenemos para cada sensor e instante de tiempo un valor (verdadero o falso) que representa si la medida del sensor ha cambiado (en la evolución de medio-corto plazo), y que está relacionado con la detección de excreciones en el caso de los sensores de gas y calidad del agua, y de duchas en el caso de la humedad. Como hemos insistido numerosas veces a lo largo de la presente memoria, es necesaria una evaluación entre diferentes sensores para proporcionar una detección robusta debido a la imprecisión de los datos que se recogen en condiciones naturalistas.

4.2.3. Clasificación de eventos de audio

En esta sección describimos la metodología seguida en este proyecto para la clasificación de eventos de audio mediante sensores IoT de sonido. El trabajo se ha publicado en el siguiente trabajo científico:

Polo-Rodriguez, A., Vilchez Chiachio, J. M., Paggetti, C., & Medina-Quero, J. (2021). Ambient Sound Recognition of Daily Events by Means of Convolutional Neural Networks and Fuzzy Temporal Restrictions. Applied Sciences, 11(15), 6978.

donde hemos colaborado varios autores. Parte del desarrollo de esta sección se incluye en el Trabajo Fin de Grado de Jose Manuel Vilchez Chiachío titulado *Recolección y reconocimiento de eventos domésticos mediante sensores de sonido en dispositivos IoT*, el cual he tutorizado junto a Javier Medina Quero.

El objetivo de la integración de los sensores de audio en los entornos inteligentes para el reconocimiento de eventos cotidianos es: (i) recoger muestras de

sonido para su entrenamiento, (ii) entrenar modelos de aprendizaje profundo a partir de muestras de sonido etiquetadas, y (iii) reconocer eventos de audio para evaluar los modelos entrenados en un contexto de tiempo real. El lenguaje de programación utilizado para codificar el script integrado en la Raspberry Pi fue Python (Monk, 2016), y los modelos de aprendizaje profundo se implementaron en Python con Keras, una biblioteca de código abierto para redes neuronales (Gulli, 2017). Los servicios remotos para el etiquetado de los datos y la difusión de la salida reconocida de los eventos de audio en tiempo real se desarrollaron bajo MQTT. Este enfoque se inspiró en los paradigmas de Fog/Edge computing, mencionados con asiduidad en secciones anteriores.

La metodología seguida fue la siguiente: en primer lugar, recogimos con Raspberry Pi muestras de sonido de una duración determinada en tiempo real. De forma paralela, la Raspberry Pi estaba suscrita a un topic MQTT, donde se publicaba el inicio y el final de cada evento para etiquetarlo de manera concreta desde una aplicación móvil. Entre el inicio y el final del intervalo de tiempo, la placa almacenaba las muestras de sonido, asociando cada instancia con una etiqueta.

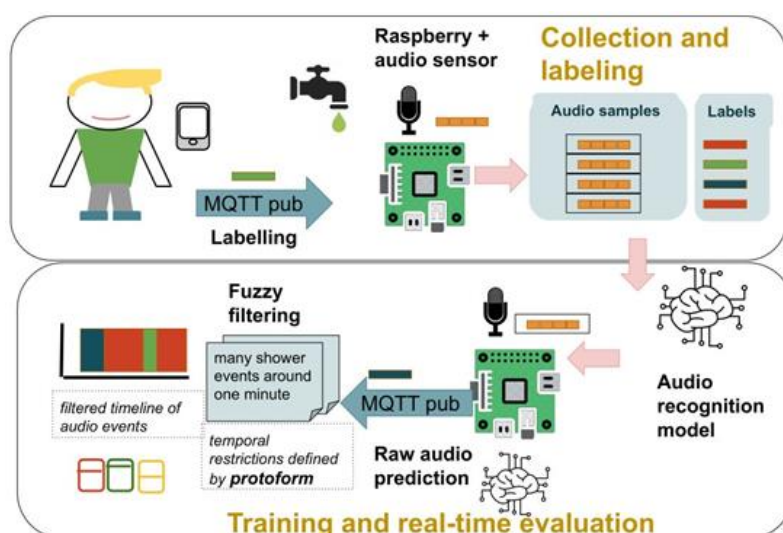


Figura 39. Arquitectura de componentes para el reconocimiento de sonido ambiente

A continuación, describimos de manera exhaustiva el procedimiento:

1. **Grabación de pistas de audio de 3 segundos de duración para cada categoría a clasificar: wc, grifo, ducha, secador y silencio/otro.** Los archivos de audio se etiquetan según la categoría correspondiente.
2. **Computamos el espectrograma espectral y de mel (donde las frecuencias se convierten a la escala mel) para traducir los registros de sonido en imágenes,** de este modo, realizamos una correcta representación del sonido que puede ser procesada por un modelo de Machine Learning. La traslación de muestras de audio digital unidimensional a una representación espacial bidimensional basada en las características del espectrograma (una imagen del sonido) proporciona resultados alentadores en la clasificación del audio ambiental (Wise, 2017).
3. **Clasificamos las imágenes de los espectrogramas Mel con las correspondientes etiquetas de sonido utilizando una red neuronal convolucional (CNN)** como en (Ordóñez, 2013). La arquitectura de la CNN propuesta consta de 6 capas, como se muestra en la siguiente figura:

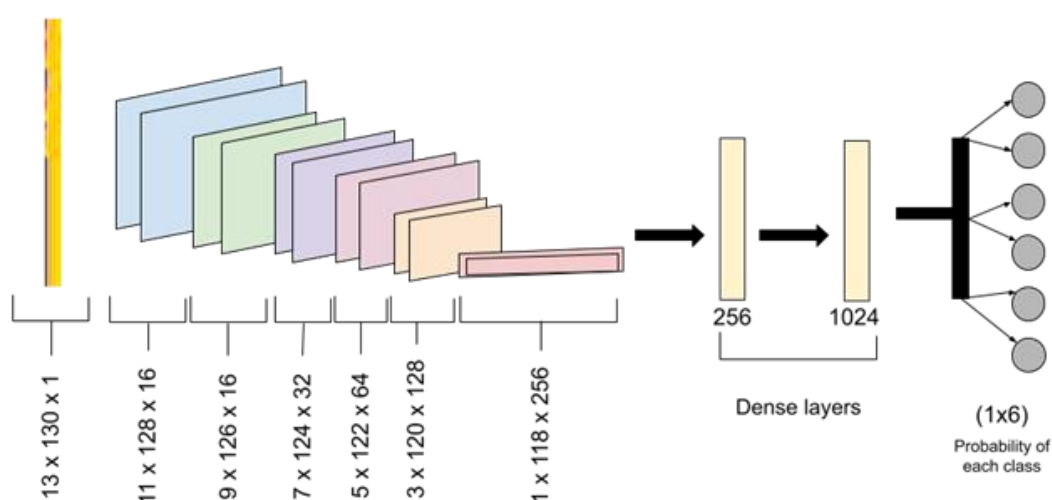


Figura 40. Descripción del modelo de la CNN

En nuestro caso, definimos un tamaño de ventana de 3 segundos para segmentar y recoger las muestras de audio ambiental, ya que proporciona un intervalo de tiempo adecuado para el reconocimiento de audio (Salamon, 2017). La frecuencia de recogida se fijó en 44,1 kHz. Posteriormente, extrajimos dos representaciones del espectro de cada sonido, que fueron evaluadas como entrada por diferentes CNNs:

- El espectrograma log-mel (LM) se calculó para representar en tiempo-frecuencia las señales de audio. Al definir la longitud de la ventana de la transformada rápida de Fourier a 2048, se obtienen imágenes de tamaño 128×130 .
- Coeficientes cepstrales de frecuencia Mel escalados (MFCC) con 13 componentes a partir de las señales de audio sin procesar, que calculan el espectro del sonido utilizando una transformada lineal del coseno. Propusimos 13 características para proporcionar la información más representativa de las muestras de audio según la literatura. Sobre la base de esta configuración, el espectrograma MFCC resultante de las frecuencias positivas obtuvo imágenes de tamaño 13×130 .

En la siguiente figura mostramos un ejemplo de los MFCC de las muestras de audio recogidas de eventos de higiene, que se utilizaron como entradas para ser clasificadas posteriormente con las correspondientes etiquetas de sonido utilizando una CNN.

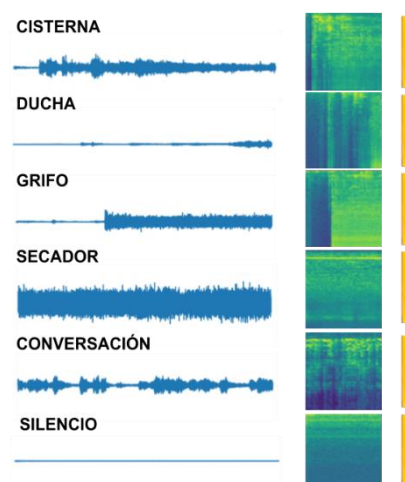


Figura 41. Ejemplo de señales de audio sin procesar a 44,1 kHz, espectrograma log-Mel (LM) y coeficiente cepstral de frecuencia Mel (MFCC) de los eventos de higiene

Las CNN se describen como extractores de características y clasificadores con resultados excelentes en el reconocimiento de imágenes (Ciresan, 2011). En nuestro caso, evaluamos dos modelos de CNN:

- Un modelo con cinco capas convolucionales para el procesamiento de MFCC, en el que se incluye un pooling promedio único después de las convoluciones debido a la reducción del espacio de entrada.
- Un modelo CNN con 5 capas convolucionales y una reducción del pooling máximo cuyas configuraciones se muestran en la siguiente tabla:

Network Architecture from Model CNN + MFCC	
Input	13 × 130 × 1
Conv(3 × 3)	11 × 128 × 16
Conv(3 × 3)	9 × 126 × 16
Conv(3 × 3)	7 × 124 × 32
Conv(3 × 3)	5 × 122 × 64
Conv(3 × 3)	3 × 120 × 128
Conv(3 × 3)	1 × 118 × 256
GlobalAvgPool2D	256
Dense	1024
Dense	15

+	Network Architecture from Model CNN + LM	
	Input	128 × 130 × 1
	Conv(2 × 2)	127 × 129 × 16
	Max-Pool(2 × 2)	63 × 64 × 16
	Conv(2 × 2)	62 × 63 × 32
	Max-Pool(2 × 2)	31 × 31 × 32
	Conv(2 × 2)	30 × 30 × 64
	Max-Pool(2 × 2)	15 × 15 × 64
	Conv(2 × 2)	14 × 14 × 128
	Conv(2 × 2)	13 × 13 × 128
	Flatten	21,632
	Dense	1024
	Dense	1024
	Dense	15

Tabla 6. Arquitectura de los modelos CNN + MFCC y CNN + LM

Los modelos fueron implementados con Keras en Python permitiendo la integración con Raspberry Pi en tiempo real. El modelo entrenado previamente se almacena en la placa IoT, recibiendo como entrada los segmentos de muestras de

audio procedentes del sensor de audio y clasificándolos en tiempo real. La clasificación se publica por MQTT en tiempo real para que pueda ser recuperada por otros dispositivos inteligentes o modelos de reconocimiento de actividades.

4.2.4. Estimación de poses mediante sensores termal

En esta sección se presenta un novedoso método para estimar los landmarks del cuerpo de una persona mediante una cámara termal. La estimación de landmarks o puntos de interés se ha realizado de forma extensa en la literatura científica en sensores de visión de espectro visible. El motivo para incluir un método de estimación de landmarks con cámara termal, es la privacidad. En entornos de higiene, y en general en entornos domésticos, los sensores de visión no deben capturar tanta nitidez ni resolución. Sin embargo, debido a las diferencias de patrones recogidos, no se pueden usar los modelos de espectro visible en termal con grandes resultados.

El uso de modelos de Deep Learning para realizar esta estimación de landmarks, es el método con mejores prestaciones (Cao, 2017) . Un problema que arrastran los modelos de Deep Learning es que necesitan de una gran cantidad de datos para desarrollar el aprendizaje. Esto requería un enorme conjunto de datos que debería además etiquetar cada imagen termal por un observador humano. Para solucionar dicha problemática, proponemos un novedoso sistema de autoetiquetado en la fase de aprendizaje (no en la de despliegue final):

- 1) Se incluye un sistema bicameral donde coexisten un sensor termal y de espectro visible con un enfoque similar.
- 2) Mediante una homografía H se transforman los puntos de las imágenes de espectro visible a termal que rectifica de forma precisa ambas imágenes.
- 3) Se incluye un etiquetado de la imagen de espectro visible usando modelos de Deep Learning estables: OpenPose (Cao, 2019).
- 4) Se relaciona el etiquetado del espectro visible con la cámara termal.

- 5) El proceso se repite recogiendo muestras en entorno doméstico de forma automática en tiempo real (con varias poses del usuario).



Figura 42. Imagen de espectro visible correlacionada con la imagen de espectro termal y los puntos de landmark de estimación de OpenPose en la termal en escala de calor

Este ingenioso método permite resolver la problemática del aprendizaje con grandes volúmenes de datos etiquetados, pero existe un mecanismo que permite multiplicarlos de forma efectiva: *aumentación de datos*. En este caso, hemos realizado un proceso de *aumentación de datos* basado en la transformación de imágenes originales basadas en un método de cambio de escala, *flip* horizontal y vertical, rotación y traslación de puntos de una imagen original que permite generar varias imágenes sintéticas. Estos procesos de rotación, traslación y escala se realizan en parámetros aleatorios y deben cambiar tanto las imágenes X_i como los puntos Y_i para que el etiquetado siga estando correlacionado. En la Figura Y, se muestra un ejemplo de *aumentación de datos* de una misma imagen:

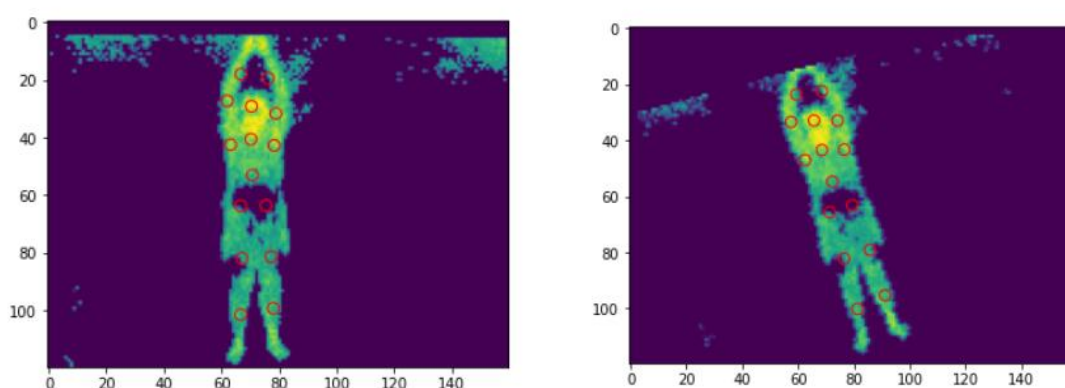


Figura 43. Imagen original e imagen sintética generada por la *aumentación de datos* junto a los *landmarks* relacionados

La estimación de los puntos de landmark se conoce como una regresión (15 puntos de coordenadas x,y) de una salida de 30 valores. Determinamos que es una regresión porque debemos conocer el valor del componente (x p y) para cada valor de salida en lugar de una clasificación. Para ello, se ha propuesto el modelo de red residual (He, 2016). Una red neuronal residual (ResNet) es una red neuronal artificial que se basa en construcciones de células piramidales similares a la de la corteza cerebral. Las redes neuronales residuales hacen esto mediante el uso de conexiones de salto o atajos para saltar sobre algunas capas. Los modelos típicos de ResNet se implementan con saltos de capa que contienen una normalización y

filtrado de valores (ReLU) y reducción de parámetros (Dropout) para reducir el overfitting.

La creación de las redes residuales se realiza mediante bloques que encapsulan los filtros y reducciones de filtros clásicos. Los bloques se construyen con 2-3 capas donde la final establece enlaces (o escapes) entre las entradas o las salidas. Estas redes han mostrado el mejor rendimiento en reconocimiento de patrones. A continuación, se muestra una implementación en Keras, donde el valor `downsample` establece si se realiza una conexión de enlace entre la entrada y la salida.

```
def relu_bn(inputs: Tensor) -> Tensor:
    relu = ReLU()(inputs)
    bn = BatchNormalization()(relu)
    return bn

def residual_block(x: Tensor, downsample: bool, filters: int, kernel_size: int = 3) -> Tensor:
    y = Conv2D(kernel_size=kernel_size,
               strides=(1 if not downsample else 2),
               filters=filters,
               padding="same")(x)
    y = relu_bn(y)
    y = Dropout(0.5)(y)
    y = Conv2D(kernel_size=kernel_size,
               strides=1,
               filters=filters,
               padding="same")(y)

    if downsample:
        x = Conv2D(kernel_size=1,
                   strides=2,
                   filters=filters,
                   padding="same")(x)
    out = Add()([x, y])
    out = relu_bn(out)
    return out
```

Figura 44. Bloque residual con conexiones de enlace en Keras (incluye normalización, filtrado RELU y dropout)

A partir de estos bloques básicos, se puede construir un modelo de red profunda de varios bloques. En nuestro caso hemos definido 8 bloques donde se establecen enlaces entre las entradas y salidas en cada par de capas definidas. En Keras la implementación final se ha codificado así:

```

inputs = Input(shape=(120, 160, 1))
num_filters = 32

t = BatchNormalization()(inputs)
t = Conv2D(kernel_size=3,
           strides=1,
           filters=num_filters,
           padding="same")(t)
t = relu_bn(t)

num_blocks_list = [2,2,2,2]
for i in range(len(num_blocks_list)):
    num_blocks = num_blocks_list[i]
    for j in range(num_blocks):
        t = residual_block(t, downsample=(j==0 and i!=0), filters=num_filters)
        num_filters *= 2

t = AveragePooling2D(2)(t)
t = Flatten()(t)

t = Dense(2048, activation='relu', kernel_regularizer='l2')(t)
t = Dense(512, activation='relu', kernel_regularizer='l2')(t)
t = Dense(128, activation='relu', kernel_regularizer='l2')(t)

outputs = Dense(30)(t)

model = Model(inputs, outputs)

model.compile(
    optimizer='adam',
    loss='mean_squared_error',
    metrics=[tf.keras.metrics.RootMeanSquaredError()]
)

return model

```

Figura 45. Creación del modelo de landmarks en Keras. Integración de las imágenes de 120,x160, red residual y capas finales de relación de filtros con las salidas de los landmarks

En este fragmento destacamos como entradas a las imágenes de 120x160, la creación de la red residual de 8 capas con los valores de bloques alineados y de escape intercalados [2,2,2,2]. Los filtros o kernels de las últimas capas se integran finalmente en una capa de normalización por media (en lugar de aplicar el máximo a diferencia de las redes no profundas residuales). El desenrollado de los bloques genera 17 capas finales para la creación de filtros.

Para relacionar los filtros o patrones abstractos de la última capa con la salida, se ha creado una red neuronal básica (capa densa) de 4 capas que van reduciendo la parametrización de patrones con una red de 2048 neuronas a 30 neuronas de salida, que configuran finalmente los valores de las posiciones de los landmarks (x o y). El modelo total requiere de más de 40 millones de parámetros a estimar. El modelo realiza el aprendizaje de forma iterativa usando como rectificación el error cuadrático medio y como optimizador de pesos de la red el método Adam, que es ofrece resultados óptimos (junto SGD).

Layer (type)	Output Shape	Param #	Connected to
input_1 (InputLayer)	[(None, 120, 160, 1) 0		
batch_normalization (BatchNorma	(None, 120, 160, 1) 4		input_1[0][0]
conv2d (Conv2D)	(None, 120, 160, 32) 320		batch_normalization[0][0]
re_lu (ReLU)	(None, 120, 160, 32) 0		conv2d[0][0]
batch_normalization_1 (BatchNor	(None, 120, 160, 32) 128		re_lu[0][0]
conv2d_1 (Conv2D)	(None, 120, 160, 32) 9248		batch_normalization_1[0][0]
re_lu_1 (ReLU)	(None, 120, 160, 32) 0		conv2d_1[0][0]
batch_normalization_2 (BatchNor	(None, 120, 160, 32) 128		re_lu_1[0][0]
conv2d_2 (Conv2D)	(None, 120, 160, 32) 9248		dropout[0][0]
add (Add)	(None, 120, 160, 32) 0		batch_normalization_1[0][0] conv2d_2[0][0]
re_lu_2 (ReLU)	(None, 120, 160, 32) 0		add[0][0]
batch_normalization_3 (BatchNor	(None, 120, 160, 32) 128		re_lu_2[0][0]
conv2d_3 (Conv2D)	(None, 120, 160, 32) 9248		batch_normalization_3[0][0]
re_lu_3 (ReLU)	(None, 120, 160, 32) 0		conv2d_3[0][0]
batch_normalization_4 (BatchNor	(None, 120, 160, 32) 128		re_lu_3[0][0]
dropout_1 (Dropout)	(None, 120, 160, 32) 0		batch_normalization_4[0][0]
conv2d_4 (Conv2D)	(None, 120, 160, 32) 9248		dropout_1[0][0]
add_1 (Add)	(None, 120, 160, 32) 0		batch_normalization_3[0][0] conv2d_4[0][0]
re_lu_4 (ReLU)	(None, 120, 160, 32) 0		add_1[0][0]
batch_normalization_5 (BatchNor	(None, 120, 160, 32) 128		re_lu_4[0][0]
conv2d_5 (Conv2D)	(None, 60, 80, 64) 18496		batch_normalization_5[0][0]
re_lu_5 (ReLU)	(None, 60, 80, 64) 0		conv2d_5[0][0]
batch_normalization_6 (BatchNor	(None, 60, 80, 64) 256		re_lu_5[0][0]
dropout_2 (Dropout)	(None, 60, 80, 64) 0		batch_normalization_6[0][0]
conv2d_7 (Conv2D)	(None, 60, 80, 64) 2112		batch_normalization_5[0][0]
conv2d_6 (Conv2D)	(None, 60, 80, 64) 36928		dropout_2[0][0]

....

average_pooling2d (AveragePooli	(None, 7, 10, 256)	0	batch_normalization_17[0][0]
flatten (Flatten)	(None, 17920)	0	average_pooling2d[0][0]
dense (Dense)	(None, 2048)	36702208	flatten[0][0]
dense_1 (Dense)	(None, 512)	1049088	dense[0][0]
dense_2 (Dense)	(None, 128)	65664	dense_1[0][0]
dense_3 (Dense)	(None, 30)	3870	dense_2[0][0]
=====			
Total params: 40,620,706			
Trainable params: 40,616,800			
Non-trainable params: 3,906			

Figura 46. Fragmento de las primeras capas generadas por los bloques residuales y últimas capas del modelo residual

En la sección de experimentación, se muestran los resultados del modelo con las imágenes recuperadas de un pequeño dataset.

Capítulo 5

5. Experimentación y resultados

En este capítulo, se detalla la evaluación llevada a cabo a lo largo del presente proyecto, especificando sensor a sensor el procedimiento según el propósito para el que estaba destinado y siempre en condiciones naturalistas. En el desarrollo del capítulo se desvela per se la complejidad y utilidad de sensorizar espacios de higiene con distintas fuentes. Cada sensor se ha usado para detectar un evento o actividad de higiene determinada. La integración de sensores de visión, audio, gas, ambiental y wearables resulta un punto clave y diferenciador para dotar al espacio del cuarto de baño de una percepción sensorial sobresaliente.

5.1. Evaluación de sensores ambientales

En el caso de estudio desarrollado para los **sensores de gas**, se realizaron **cuatro pruebas de cinco minutos** para comprobar la precisión de los mismos al detectar el evento de deposición. Los siguientes gráficos reflejan los datos recogidos por los sensores, obteniendo resultados concluyentes. Estos resultados muestran que la concentración de partículas por millón (PPM) de metano e hidrógeno en el momento de la defecación aumenta significativamente, lo que refleja la robustez para detectar el evento en el inodoro. En la siguiente figura, mostramos las PPM en cuatro experimentos para detectar las deposiciones con MQ-2 y MQ-8.

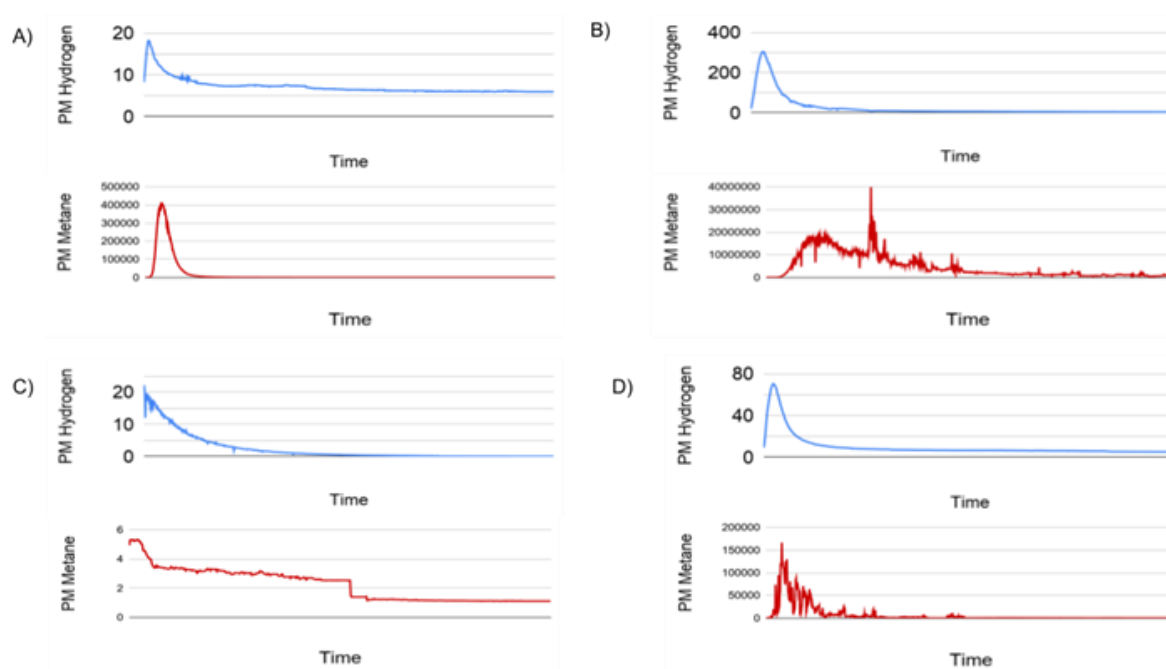


Figura 47. Cuatro evaluaciones de la detección de las deposiciones con el sensor MQ-2 y MQ-8 (hidrógeno en azul y metano en rojo respectivamente)

Continuamos con el **sensor de humedad**, con el que se ha realizado un caso de estudio de **15 minutos almacenando los datos obtenidos por el sensor antes, durante y después de la ducha**. La humedad ambiental varía muy poco cuando no se produce ningún evento, pero cuando comienza la ducha la humedad aumenta incluso hasta el 95-100%. Por lo tanto, la detección de un umbral de referencia de la humedad es una variable clave para una correcta detección de este tipo de actividad. La identificación de la actividad se realizó considerando la concentración al principio y al final de la ducha, detectándose el evento con una concentración que superaba el umbral del 85%.

Como es de esperar, la disminución de la humedad se observa cuando la actividad de la ducha finaliza y se deja la puerta abierta. La siguiente figura muestra la concentración de humedad antes, mientras y después de la ducha en este caso de estudio concreto:

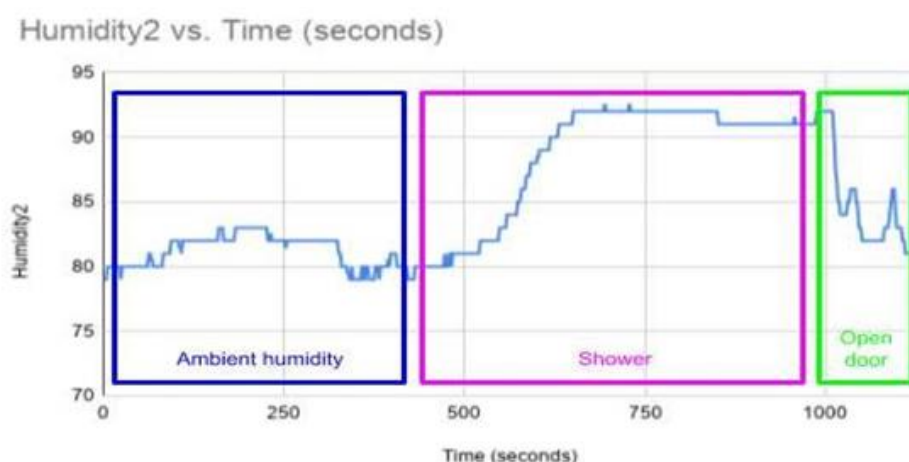


Figura 48. Concentración de humedad antes, mientras y durante la ducha

5.2. Evaluación de procesamiento inercial del wearable

En el caso de la pulsera de actividad, se recogieron 8 minutos de datos en los que una persona que llevaba el wearable (Xiaomi Mi Band 3) realizaba tres tipos de eventos diferentes: cepillarse los dientes, lavarse las manos y lavarse la cara. Los datos inerciales (giroscopio), que proporcionan información sobre la orientación y la cantidad de movimientos, son útiles para reconocer los eventos característicos de los gestos de la mano. La figura 49, ilustra la correlación de levantar la mano utilizando el eje X de la aceleración y la cantidad de movimiento (norma del giroscopio) para las actividades de lavado.



Figura 49. Cantidad de movimiento basada en la norma del giroscopio y detección de la elevación basada en la aceleración del eje X

5.3. Evaluación de sensores de calidad de agua

Para los sensores de calidad del agua, dos habitantes de 70 y 69 años (hombre y mujer, respectivamente) probaron el sistema propuesto en su casa durante dos días y medio, etiquetando cuando se desarrollaba una excreción (orina o heces). En la siguiente figura, mostramos el despliegue de estos sensores en el inodoro. Como podemos ver en la imagen, las sondas se fijaron al inodoro de forma aislada, evitando contactos o movimientos entre ellas que pudieran desarrollar malas medidas mientras se excretaba o se tiraba de la cisterna. Para ello, utilizamos cinta impermeable de doble cara. Además, las sondas se fijaron al W.C. para evitar interferir con las actividades diarias, y la placa se situó cercana a ellas para recoger los datos en tiempo real. Como indicamos en la sección de integración de los sensores, éstos se calibraron con soluciones proporcionadas por la empresa Atlas Scientific.

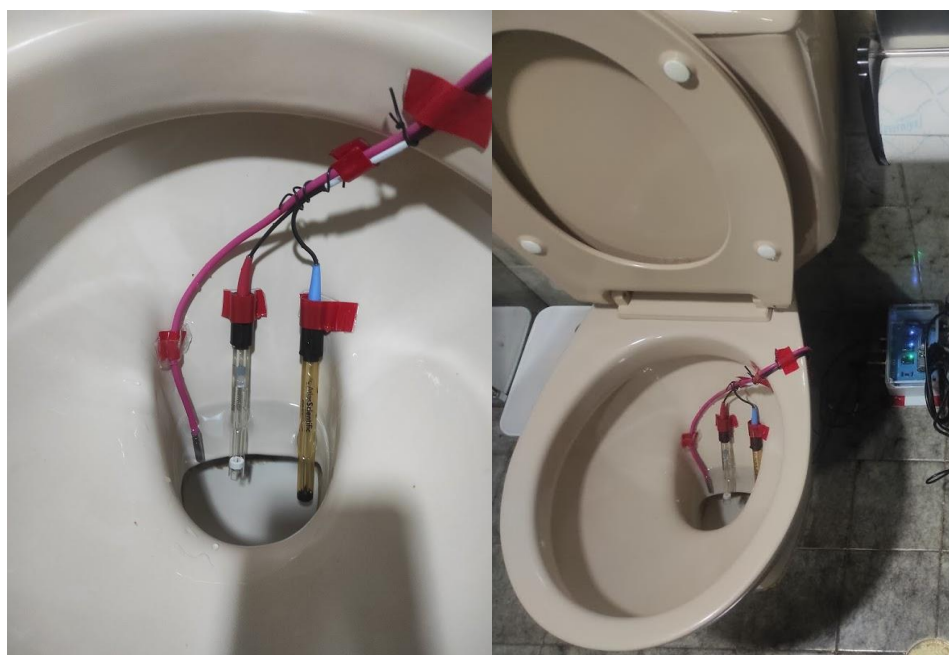


Figura 50. *Izquierda) Sensores del kit de monitorización de la calidad del agua acoplados al W.C. Derecha) Sondas conectadas desde los sensores a la placa ESP8266.*

Durante los dos días se almacenaron los datos, enviando por MQTT la marca de tiempo y las mediciones. Posteriormente, se aplicó el método presentado en la

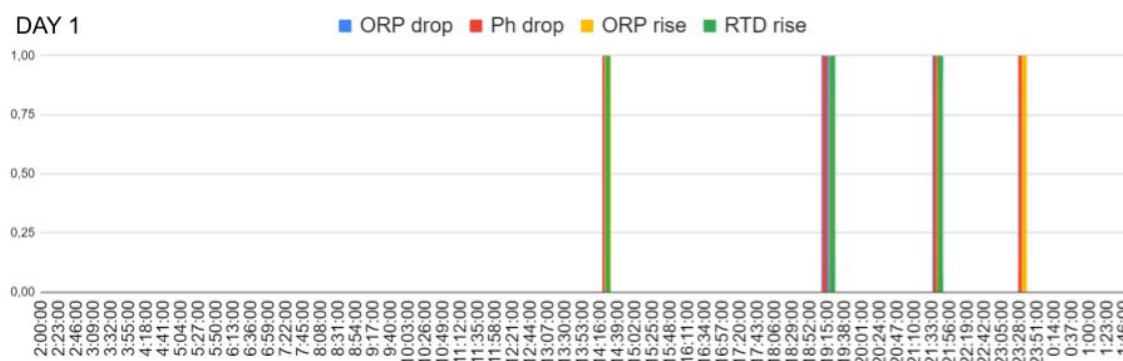
sección 4.2.2.2 para la detección en tiempo real de las excreciones humanas mediante la computación de cambios de eventos con ventana móvil deslizante.

A partir de los flujos de datos obtenidos de los sensores de pH, ORP y temperatura, calculamos los siguientes cambios de eventos que estaban relacionados con las excreciones humanas: caída de ORP, caída de pH, aumento de ORP y aumento de RTD. Además, observamos que definiendo una ventana temporal retardada de 5 segundos mejoraba la toma de decisiones en el reconocimiento de la actividad. Se utilizaron los siguientes umbrales para calcular el cambio de eventos:

- Subida de ORP con $\alpha < -7$
- Bajada de Ph con $\alpha > 0,1$
- Bajada de ORP $\alpha > 5$
- Subida de RTD $\alpha < 0,5$.

El código y los datos del procesamiento de los flujos de sensores basados en el método propuesto se incluyen en el zip adjunto a esta memoria así como en mi [repositorio de GitHub](#).

En la evaluación del caso de estudio, analizamos la detección de eventos proporcionada por el sistema en relación a los datos proporcionados por los habitantes. En la siguiente figura, describimos la detección de eventos para cada uno de los flujos de sensores durante los tres días (el primero sólo fue medio día):



....

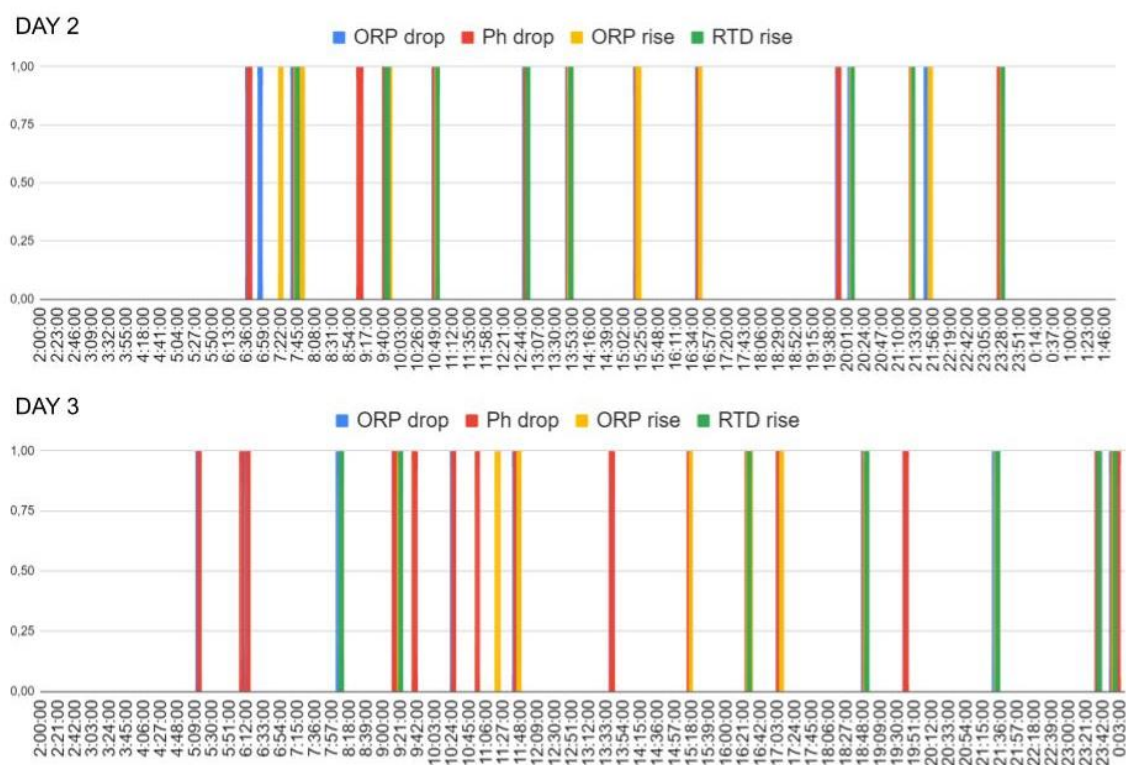


Figura 51. Detección de eventos de cambios de caída de ORP, caída de PH, subida de ORP y subida de RTD durante los días del experimento

A la hora de comparar el número de cambios de eventos detectados con las etiquetas proporcionadas por los habitantes sometidos al estudio, la detección se agregó por minutos y se incluyó una fuzzificación de la respuesta en el último y los dos siguientes minutos. Esta fuzzificación se desarrolló para permitir la comparación entre flujos de sensores que pueden estar atrasados uno o dos minutos respecto a la detección de otro sensor.

Por ejemplo, el pH y el ORP presentan segundos de diferencia en la detección en tiempo real (el pH es más rápido en la detección de cambios que el ORP). En la siguiente figura, mostramos la sumatoria de eventos detectados de cambio de eventos de sensores con respecto a los eventos de verdad etiquetados (ground truth):

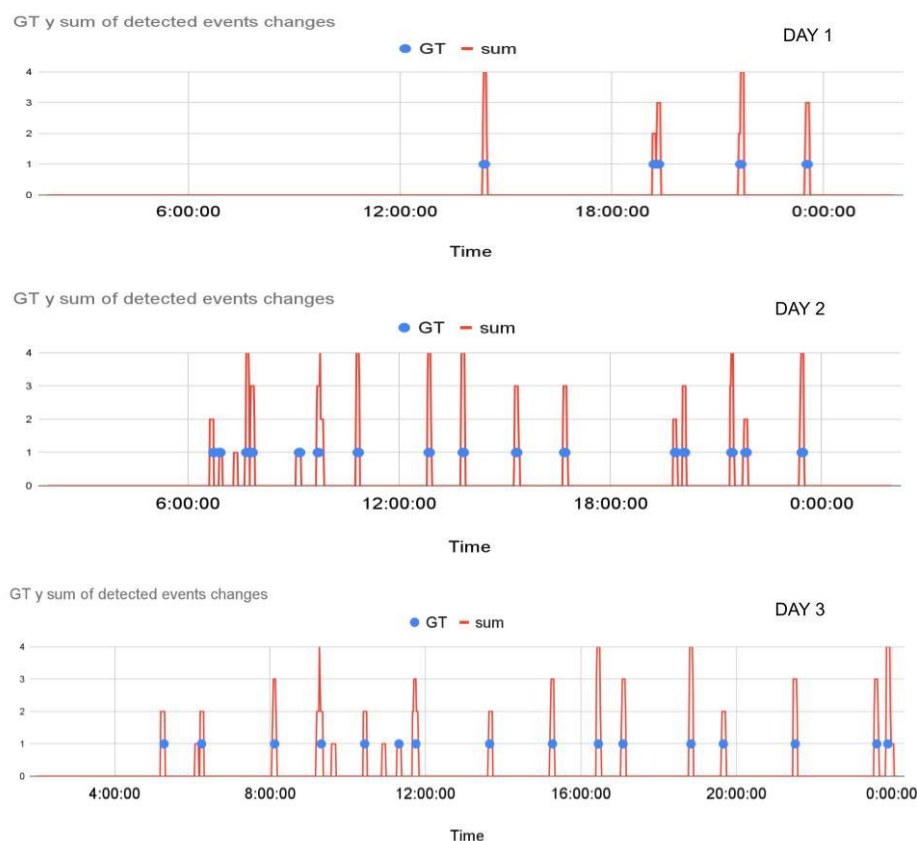


Figura 52. Sumatoria de los cambios de eventos en sensores detectados (caída de ORP, caída de Ph, aumento de ORP y RTD) con respecto a la etiqueta de verdad (ground truth) en círculos azules

Como podemos observar de la colección de datos y eventos computados, existe una correlación robusta, de modo que, con dos o más cambios de evento detectados simultáneamente, podemos asegurar que ha habido excreción. Sólo el segundo día muestra un único cambio significativo coincidente con una excreción; en los demás casos, 2 o más cambios representan esto sin errores de falsos positivos. De esta manera, observamos que no todos los sensores y lecturas describen un cambio significativo en condiciones reales, por lo que es necesario incluir varios sensores y fusionar el reconocimiento de estas fuentes para obtener un sistema robusto como el que describimos en este Trabajo de Fin de Máster.

5.4. Evaluación de reconocimiento de audio

Para el reconocimiento de audio, realizamos un caso de estudio de **12 minutos** mientras un habitante realizaba tareas como ducharse, secarse el pelo o lavarse las manos. Los datos fueron etiquetados por el usuario, comparando posteriormente la predicción del modelo respecto a la observación y obteniendo una **precisión de 0,975 para el modelo CNN+MFCC, mientras que para el modelo CNN+LM una precisión de 0,945**. En la siguiente tabla hemos recopilado los detalles en la evaluación de cada uno de los modelos (cantidad de parámetros entrenables, tiempo de aprendizaje, millones de instrucciones (hasta 40 epochs) y tiempo de evaluación) en una Raspberry Pi 3B cuya frecuencia es de 400 MHz.

	Trainable Parameters	Learning Time	Millions of Instructions (MI)	Evaluation Time
Model CNN + MFCC	1.7 M	96 min	230.4×10^3 MI	2.53 s
Model CNN + LM	23.3 M	207 min	496.8×10^3 MI	2.81 s

Tabla 7. Parámetros entrenables, tiempo de aprendizaje, millones de instrucciones (MI) y tiempo de evaluación

En la siguiente figura, detallamos el rendimiento del reconocedor de sonido del primer modelo, comparando la verdad contra el sonido reconocido.

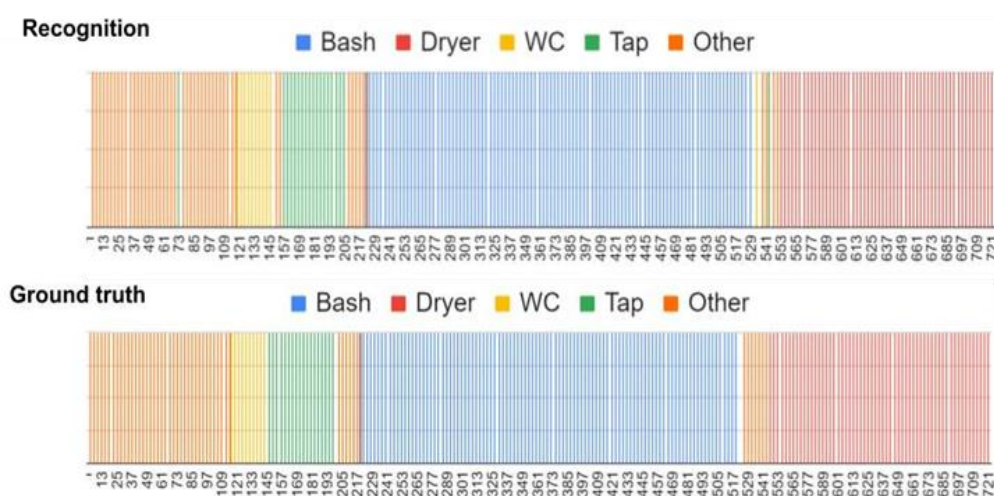


Figura 53. Arriba) Reconocimiento de audio a partir del modelo entrenado y, abajo) la verdad del caso de estudio en tiempo real

5.5. Evaluación de estimación de landmarks desde cámaras termales

En esta sección, presentamos los resultados obtenidos en la evaluación de landmarks corporales de los usuarios a partir de imágenes termales. En primer lugar, tenemos que destacar que se ha realizado una **recolección de datos muy rápida** como se requiere en la integración y optimización de entornos inteligentes y personas. Se recogieron **500 segundos de muestras de un usuario** realizando diferentes poses que fueron recogidas con las cámaras de visión termal y espectro visible. La homografía entre ambas cámaras se realizó con un calibrado manual. Estos datos fueron **aumentados (K=10 veces)** y procesado con **OpenPose** para realizar un etiquetado automático de las imágenes termales a partir de las poses de las visibles. Se ha realizado una **división del conjunto de test y training (20% y 80% respectivamente)** para evaluar con datos no observados en el conjunto de entrenamiento.

Finalmente, el modelo de red residual propuesto para la regresión de los puntos se ha compilado para el aprendizaje en un batch de 64 imágenes (se cargan estas en bloque y se optimizan pesos al final del conjunto) **con 50 épocas**. El **aprendizaje** en un computador con procesador Intel i7 de séptima generación y 16 GB de RAM ha tardado **12 horas y media**.

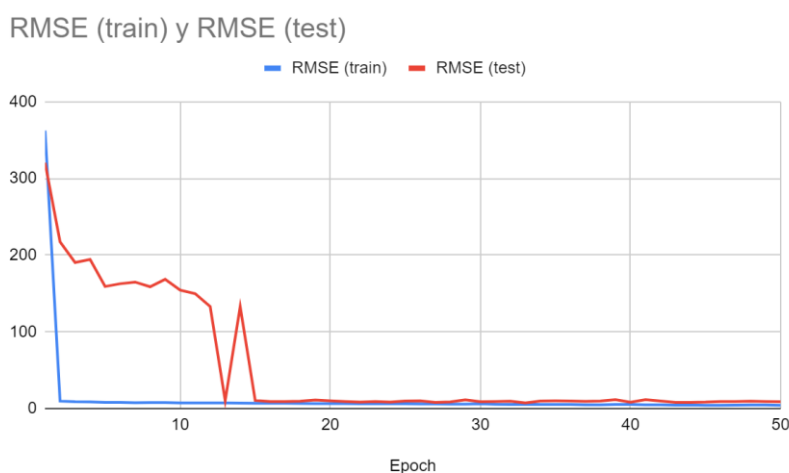


Figura 54. Evolución del RMSE en el conjunto de entrenamiento y test durante las 50 épocas de aprendizaje

La evolución del error cuadrático medio (RMSE) entre el conjunto de entrenamiento y test puede verse en la figura anterior. Podemos comprobar cómo el error de evaluación del test es sensiblemente mayor que el de entrenamiento, siendo estimado en $RMSE(train)=4.5$ y $RMSE(test)=8.8$ en la última época.

Sin embargo, la evolución del RMSE no es muy interpretable respecto a los resultados visuales, ya que, aunque no varíe en las primeras epochs significativamente del final, los resultados visuales sí son muy relevantes. Por ello, en la siguiente figura, incluimos unas imágenes del modelo generado y la validación de 9 imágenes termales que ha realizado el modelo para estimar los 15 puntos de landmark. Recordamos que estas imágenes no fueron integradas en el aprendizaje y los resultados son muy alentadores destacando además las escasas imágenes utilizadas en la recolección de datos. En trabajos futuros una recolección de varios usuarios permitirá establecer un modelo más preciso para el reconocimiento de poses.

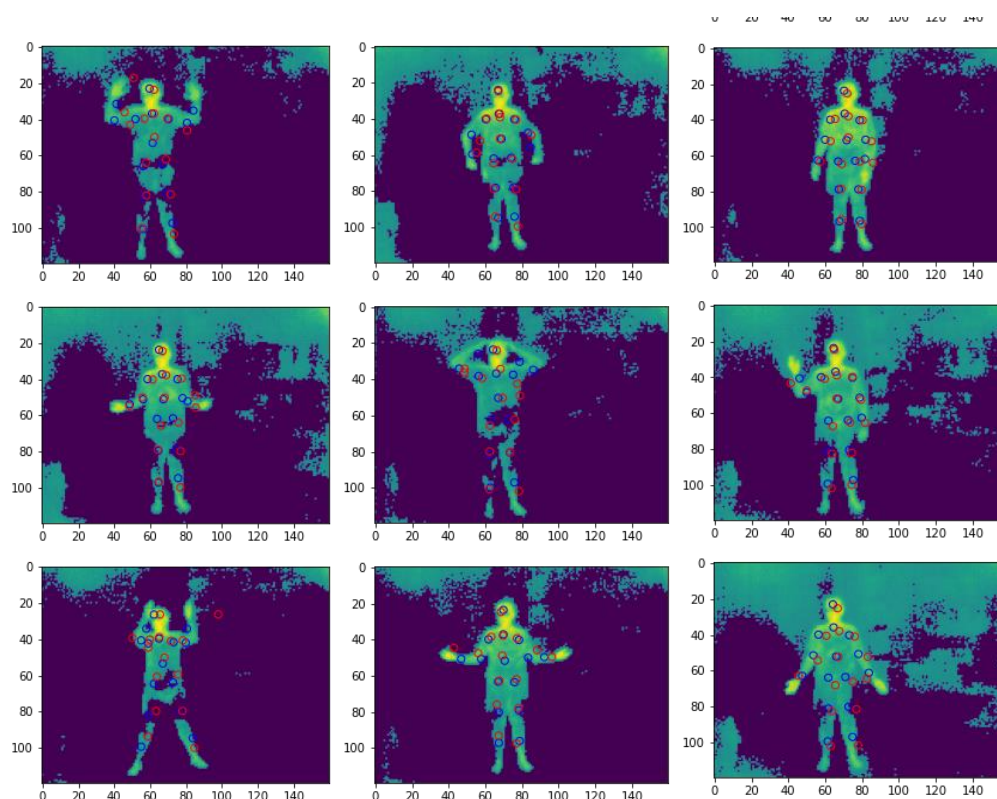


Figura 55. Estimación de la pose de un usuario: los círculos azules corresponden a la estimación de la pose realizada por OpenPose desde el espectro visible (trasladado al termal), y los círculos rojos la realizada por la red residual desarrollada en este trabajo

Capítulo 6

6. Conclusiones y trabajos futuros

En este Trabajo de Fin de Máster hemos propuesto una selección de sensores para reconocer actividades y eventos relacionados con la higiene, que están estrechamente unidos al correcto estado funcional de las personas. Esta selección de sensores está compuesta por sensores multimodales heterogéneos: sensores de gas (metano e hidrógeno), sensores inerciales de pulseras de actividad, sensores de humedad, pH, calidad del agua, temperatura, micrófonos y cámaras termales. De forma adicional a esta selección multimodal de sensores, se han propuesto diferentes modelos de Machine Learning para procesar la información de cada uno de estos sensores heterogéneos. Esto incluye desde clasificadores de eventos de audio basados en información espectral, aprendizaje profundo y detección de cambio de eventos mediante ventana móvil deslizante en flujos de eventos unidimensionales, hasta la estimación de poses mediante landmarks. Además, la integración de estos dispositivos se ha realizado siguiendo un enfoque de edge-fog computing con placas IoT, permitiendo transmitir tanto los datos brutos y la información procesada de forma eficaz evaluando la actividad y eventos generados por el usuario, pero preservando siempre su privacidad.

Los resultados obtenidos en la experimentación son alentadores: el sensor de gas detecta con precisión aceptable el evento para el que ha sido seleccionado, la detección de la defecación. Respecto al reconocimiento de audio para detectar otros eventos, como, por ejemplo, secarse el pelo, ir al baño, ducharse o abrir el grifo, se ha demostrado que tiene una tasa de error muy baja, adaptándose perfectamente al caso de estudio presentado; demostrando ser crucial en escenarios en tiempo real para evitar falsos positivos y proporcionar un reconocimiento coherente de los eventos de higiene. La detección de las duchas se ha relacionado directamente con

el uso del sensor de humedad, del que también se ha obtenido una correlación directa. Además, la inclusión de sensores inerciales en la pulsera del usuario ha servido para la detección de eventos como el cepillado de dientes o el lavado de cara si se procesan dichos datos, lo cual ayuda a completar el amplio abanico de actividades casi al completo. Relativo a los sensores de pH, ORP y temperatura, los resultados muestran un alto rendimiento al fusionar el reconocimiento de cambios de tendencias de mediciones de varias fuentes de sensores. Por último, la estimación de landmarks con cámaras termales nos ofrece unas aproximaciones bastante acertadas para la escasez de datos de entrenamiento, lo que augura unos resultados exitosos en próximos trabajos.

Como conclusión, debemos tener en cuenta que, aunque los resultados son alentadores y parece ser un sistema robusto, es necesario realizar más evaluaciones para investigar los modelos de adaptación al entorno en el que se despliega. En el caso de este trabajo, por ejemplo, el sistema de clasificación de eventos de sonido se desplegó en un baño en el que se obtuvieron las pistas de audio con las que se entrenó el modelo, por lo que no podríamos trasladarlo en el mismo estado a otro baño y esperar los mismos resultados.

De cara a la Tesis Doctoral, se pretende abordar la fusión de la información procesada de todos estos sensores heterogéneos para aumentar las capacidades de detección de actividades ligadas a un contexto dado. De esta forma incluiríamos una segunda capa inteligente de reconocimiento de actividades que fusionaría, no los datos brutos de los sensores, si no el procesamiento de los diferentes procesos de Machine Learning relacionados con cada sensor.

Cabe destacar que la fusión de sensores en condiciones naturalistas es un trabajo harto y tedioso y que requiere de un esfuerzo adicional en la parte de recogida de datos, procesamiento y análisis posterior, tal y como indican los últimos trabajos en despliegues de entornos reales (Lupión, 2021). En el contexto del presente proyecto, se pretende recoger más datos durante periodos de tiempo más largos, procesando la información obtenida de los sensores para crear un sistema de reconocimiento robusto a los falsos positivos que cubra la mayoría de los eventos

relacionados con la higiene. Sería interesante realizar casos de estudio en diferentes hogares, ya que pueden proporcionar datos relevantes sobre las diferencias según el entorno, por ejemplo, con el pH, ORP y temperatura para la detección de excreciones humanas y su clasificación en tiempo real.

Será esencial unificar todo el conocimiento obtenido de esta selección de sensores en un nodo Edge/Fog para su posterior almacenamiento y procesamiento, preservando así la privacidad y el bienestar de las personas dentro del entorno inteligente. Además, el uso de reglas difusas podría mejorar la definición basada en el conocimiento de las actividades con un procesamiento constante a partir de los datos brutos, integrando los datos recogidos de diferentes sensores. Por último, la adición de una interfaz visual para la monitorización y visualización de los datos en tiempo real podría ser un punto a favor, ya que existen herramientas como Kibana que se adaptan a las necesidades del usuario en el contexto de IoT (Bajer, 2017) de manera sencilla y escalable.

Capítulo 7

7. Referencias

1. Downey, L. C., Smith, P. B., & Benjamin Jr, D. K. (2010). Risk factors and prevention of late-onset sepsis in premature infants. *Early human development*, 86(1), 7-12.
2. van Kerrebroeck, P., Victor, A., & Wein, A. (2002). The standardisation of terminology of lower urinary tract function: report from the Standardisation Sub-committee of the International Continence Society. *Neurourology and urodynamics*, 21, 167-178.
3. Baillie, L., & Arrowsmith, V. (2014). Meeting elimination needs. *Developing practical Nursing Skill*.
4. Al-Sarawi, S., Anbar, M., Alieyan, K., & Alzubaidi, M. (2017, May). Internet of Things (IoT) communication protocols. In *2017 8th International conference on information technology (ICIT)* (pp. 685-690). IEEE.
5. Medina-Quero, J., Zhang, S., Nugent, C., & Espinilla, M. (2018). Ensemble classifier of long short-term memory with fuzzy temporal windows on binary sensors for activity recognition. *Expert Systems with Applications*, 114, 441-453..
6. Cruciani, F., Nugent, C. D., Quero, J. M., Cleland, I., McCullagh, P., Synnes, K., & Hallberg, J. (2020). Personalizing Activity Recognition With a Clustering Based Semi-Population Approach. *IEEE Access*, 8, 207794-207804.
7. Laput, G., Zhang, Y., & Harrison, C. (2017, May). Synthetic sensors: Towards generalpurpose sensing. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems* (pp. 3986-3999).
8. Ienca, M., Wangmo, T., Jotterand, F., Kressig, R. W., & Elger, B. (2018). Ethical design of intelligent assistive technologies for dementia: a descriptive review. *Science and engineering ethics*, 24(4), 1035-1055.
9. Ali Hamad, R., Järpe, E., and Lundström, J. (2018). Stability analysis of the t-sne algorithm for human activity pattern data. In *The 2018 IEEE International Conference on Systems, Man, and Cybernetics (SMC2018)*.
10. Bouchard, K., Hao, J., Bouchard, B., Gaboury, S., Moutacalli, M., Gouin-Vallerand, C., Kenfack Ngankam, H., Pigot, H., and Giroux, S. (2018). The cornerstones of

- smart home research for healthcare. *Smart Innovation, Systems and Technologies*, 93:185.
11. Bravo, J., Hervas, R., Fontecha, J., and González, I. (2018). M-health: lessons learned by m-experiences. *Sensors*, 18(5):1569.
 12. Rashidi, P. and Mihailidis, A. (2012). A survey on ambient-assisted living tools for older adults. *IEEE journal of biomedical and health informatics*, 17(3):579.
 13. López-Medina, M. A., Espinilla, M., Cleland, I., Nugent, C., & Medina, J. (2020). Fuzzy cloud-fog computing approach application for human activity recognition in smart homes. *Journal of Intelligent & Fuzzy Systems*, 38(1), 709-721.
 14. Al-Sarawi, S., Anbar, M., Alieyan, K., and Alzubaidi, M. (2017). Internet of things (iot) communication protocols. In *2017 8th International conference on information technology (ICIT)*, pages 685-690. IEEE.
 15. Medina-Quero, J., Zhang, S., Nugent, C., and Espinilla, M. (2018a). Ensemble classifier of long short-term memory with fuzzy temporal windows on binary sensors for activity recognition. *Expert Systems with Applications*, 114:441.
 16. Ordoñez, F. J. and Roggen, D. (2016). Deep convolutional and lstm recurrent neural networks for multimodal wearable activity recognition. *Sensors*, 16(1):115.
 17. Zhang, S., Wei, Z., Nie, J., Huang, L., Wang, S., and Li, Z. (2017b). A review on human activity recognition using vision-based method. *Journal of healthcare engineering*, 2017.
 18. Laput, G.; Zhang, Y.; Harrison, C. Synthetic sensors: Towards general-purpose sensing. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*, Denver, CO, USA, 6 May 2017; pp. 3986–3999.
 19. Cruz-Sandoval, D.; Beltran-Marquez, J.; Garcia-Constantino, M.; Gonzalez-Jasso, L.A.; Favela, J.; Lopez-Nava, I.H.; Cleland, I.; Ennis, A.; Hernandez-Cruz, N.; Rafferty, J.; et al. Semi-automated data labeling for activity recognition in pervasive healthcare. *Sensors* 2019, 19, 3035.
 20. Shih, P. C., Han, K., Poole, E. S., Rosson, M. B., & Carroll, J. M. (2015). Use and adoption challenges of wearable activity trackers. *ICConference 2015 Proceedings*.
 21. Patel, M. S., Asch, D. A., & Volpp, K. G. (2015). Wearable devices as facilitators, not drivers, of health behavior change. *Jama*, 313(5), 459-460.
 22. Heale, L. D., Dover, S., Goh, Y. I., Maksymiuk, V. A., Wells, G. D., & Feldman, B. M. (2018). A wearable activity tracker intervention for promoting physical activity in adolescents with juvenile idiopathic arthritis: a pilot study. *Pediatric Rheumatology*, 16(1), 66.
 23. Mercer, K., Giangregorio, L., Schneider, E., Chilana, P., Li, M., & Grindrod, K. (2016). Acceptance of commercially available wearable activity trackers among adults

- aged over 50 and with chronic illness: a mixed-methods evaluation. *JMIR mHealth and uHealth*, 4(1), e7.
24. Mercer, K., Li, M., Giangregorio, L., Burns, C., & Grindrod, K. (2016). Behavior change techniques present in wearable activity trackers: a critical analysis. *JMIR mHealth and uHealth*, 4(2), e40.
 25. Lim, S. S., Vos, T., Flaxman, A. D., Danaei, G., Shibuya, K., Adair-Rohani, H., ... & Aryee, M. (2012). A comparative risk assessment of burden of disease and injury attributable to 67 risk factors and risk factor clusters in 21 regions, 1990–2010: a systematic analysis for the Global Burden of Disease Study 2010. *The lancet*, 380(9859), 2224-2260.
 26. Espinilla, M., Quero, J. M., Irvine, N., Cleland, I., & Nugent, C. Fuzzy framework for activity recognition in a multi-occupant smart environment based on wearable devices and proximity beacons.
 27. King, C. E., & Sarrafzadeh, M. (2018). A survey of smartwatches in remote health monitoring. *Journal of healthcare informatics research*, 2(1-2), 1-24.
 28. Zheng, Y. L., Ding, X. R., Poon, C. C. Y., Lo, B. P. L., Zhang, H., Zhou, X. L., ... & Zhang, Y. T. (2014). Unobtrusive sensing and wearable devices for health informatics. *IEEE Transactions on Biomedical Engineering*, 61(5), 1538-1554.
 29. Torrado, J. C., Gomez, J., & Montoro, G. (2017). Emotional self-regulation of individuals with autism spectrum disorders: smartwatches for monitoring and interaction. *Sensors*, 17(6), 1359.
 30. Mo, L., Li, F., Zhu, Y., and Huang, A. (2016). Human physical activity recognition based on computer vision with deep learning model. In *2016 IEEE International Instrumentation and Measurement Technology Conference Proceedings*, pages 1{6. IEEE.
 31. Hiriyannaiah, S., Akanksh, B., Koushik, A., Siddesh, G., and Srinivasa, K. (2020). Deep learning for multimedia data in iot. In *Multimedia Big Data Computing for IoT Applications*, pages 101{129. Springer.
 32. Brauer, H. S. P. (2014). Camera based human localization and recognition in smart environments. PhD thesis, University of the West of Scotland.
 33. Waheed, S. A. and Khader, P. S. A. (2017). A novel approach for smart and costective iot based elderly fall detection system using pi camera. In *2017 IEEE International Conference on Computational Intelligence and Computing Research (ICCIC)*, pages 1{4. IEEE.
 34. Macek, N., Franc, I., Bogdanoski, M., and Mirkovic, A. (2016). Multimodal biometric authentication in iot: Single camera case study.

35. Sixsmith, A. and Johnson, N. (2004). A smart sensor to detect the falls of the elderly. *IEEE Pervasive computing*, 3(2):42{47.
36. Medina-Quero, J. M., Burns, M., Razzaq, M. A., Nugent, C., and Espinilla, M. (2018b). Detection of falls from non-invasive thermal vision sensors using convolutional neural networks. In *Multidisciplinary Digital Publishing Institute Proceedings*, volume 2, page 1236.
37. Griffiths, E., Assana, S., and Whitehouse, K. (2018). Privacy-preserving image processing with binocular thermal cameras. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, 1(4):1{25.
38. Mohammadmoradi, H., Munir, S., Gnawali, O., and Shelton, C. (2017). Measuring people-ow through doorways using easy-to-install ir array sensors. In *2017 13th International Conference on Distributed Computing in Sensor Systems (DCOSS)*, pages 35{43. IEEE.
39. Han, J. and Bhanu, B. (2005). Human activity recognition in thermal infrared imagery. In *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)-Workshops*, pages 17{17. IEEE.
40. López-Medina, M. Á., Espinilla, M., Nugent, C., & Quero, J. M. (2020). Evaluation of convolutional neural networks for the classification of falls from heterogeneous thermal vision sensors. *International Journal of Distributed Sensor Networks*, 16(5), 1550147720920485.
41. Medina-Quero, J.M.; Burns, M.; Razzaq, M.A.; Nugent, C.; Espinilla, M. Detection of falls from non-invasive thermal vision sensors using convolutional neural networks. *Multidiscip. Digit. Publ. Inst. Proc.* 2018, 2, 1236.
42. Wang, J.; Chen, Y.; Hao, S.; Peng, X.; Hu, L. Deep learning for sensor-based activity recognition: A survey. *Pattern Recognit. Lett.* 2019, 119, 3–11.
43. Albawi, S.; Mohammed, T.A.; Al-Zawi, S. Understanding of a convolutional neural network. In *Proceedings of the 2017 International Conference on Engineering and Technology (ICET)*, Antalya, Turkey, 21–23 August 2017; pp. 1–6.
44. Wyse, L. Audio spectrogram representations for processing with convolutional neural networks. *arXiv* 2017, arXiv:1706.09559.
45. Salamon, J.; Bello, J.P. Deep convolutional neural networks and data augmentation for environmental sound classification. *IEEE Signal Process. Lett.* 2017, 24, 279–283.
46. Kim, J. Urban sound tagging using multi-channel audio feature with convolutional neural networks. In *Proceedings of the Detection and Classification of Acoustic Scenes and Events*, Tokyo, Japan, 2–3 November 2020.

47. Lasseck, M. Acoustic bird detection with deep convolutional neural networks. In Proceedings of the Detection and Classification of Acoustic Scenes and Events 2018 Workshop (DCASE2018), Surrey, UK, 19–20 November 2018; pp. 143–147.
48. Su, Y.; Zhang, K.; Wang, J.; Madani, K. Environment sound classification using a two-stream CNN based on decision-level fusion. *Sensors* 2019, 19, 1733.
49. Beltrán, J.; Chávez, E.; Favela, J. Scalable identification of mixed environmental sounds, recorded from heterogeneous sources. *Pattern Recognit. Lett.* 2015, 68, 153–160.
50. Beltrán, J.; Navarro, R.; Chávez, E.; Favela, J.; Soto-Mendoza, V.; Ibarra, C. Recognition of audible disruptive behavior from people with dementia. *Pers. Ubiquitous Comput.* 2019, 23, 145–157.
51. Cruz-Sandoval, D.; Beltran-Marquez, J.; Garcia-Constantino, M.; Gonzalez-Jasso, L.A.; Favela, J.; Lopez-Nava, I.H.; Cleland, I.; Ennis, A.; Hernandez-Cruz, N.; Rafferty, J.; et al. Semi-automated data labeling for activity recognition in pervasive healthcare. *Sensors* 2019, 19, 3035.
52. Laput, G.; Ahuja, K.; Goel, M.; Harrison, C. Ubicoustics: Plug-and-play acoustic activity recognition. In Proceedings of the 31st Annual ACM Symposium on User Interface Software and Technology, Berlin, Germany, 14 October 2018; pp. 213–224.
53. Atzori, L.; Iera, A.; Morabito, G. The Internet of Things: A survey. *Comput. Netw.* 2010, 54, 2787–2805.
54. Chen, L.W.; Ho, Y.F.; Kuo, W.T.; Tsai, M.F. Intelligent file transfer for smart handheld devices based on mobile cloud computing. *Int. J. Commun. Syst.* 2015, 30, e2947.
55. Lara, O.D.; Labrador, M.A. A survey on human activity recognition using wearable sensors. *IEEE Commun. Surv. Tutorials* 2013, 15, 1192–1209.
56. Shi, W.; Dustdar, S. The promise of edge computing. *Computer* 2016, 49, 78–81.
57. Bonomi, F.; Milito, R.; Zhu, J.; Addepalli, S. Fog computing and its role in the internet of things. In Proceedings of the first edition of the MCC workshop on Mobile cloud Computing, Helsinki, Finland, 17 August 2012; pp. 13–16.
58. Espinilla, M.; Martínez, L.; Medina, J.; Nugent, C. The experience of developing the UJAmI Smart lab. *IEEE Access* 2018, 6, 34631–34642.
59. Ordóñez, F.J.; Roggen, D. Deep convolutional and lstm recurrent neural networks for multimodal wearable activity recognition. *Sensors* 2016, 16, 115.
60. Ruiz, A.R.J.; Granja, F.S. Comparing ubisense, bespoon, and decawave uwb location systems: Indoor performance analysis. *IEEE Trans. Instrum. Meas.* 2017, 66, 2106–2117.

61. Xu, X.; Tang, J.; Zhang, X.; Liu, X.; Zhang, H.; Qiu, Y. Exploring techniques for vision based human activity recognition: Methods, systems, and evaluation. *Sensors* 2013, 13, 1635–1650.
62. Cruz-Sandoval, D.; Beltran-Marquez, J.; Garcia-Constantino, M.; Gonzalez-Jasso, L.A.; Favela, J.; Lopez-Nava, I.H.; Cleland, I.; Ennis, A.; Hernandez-Cruz, N.; Rafferty, J.; et al. Semi-automated data labeling for activity recognition in pervasive healthcare. *Sensors* 2019, 19, 3035.
63. López-Medina, M.; Espinilla, M.; Cleland, I.; Nugent, C.; Medina, J. Fuzzy cloud-fog computing approach application for human activity recognition in smart homes. *J. Intell. Fuzzy Syst.* 2020, 38, 709–721.
64. Medina-Quero, J.; Zhang, S.; Nugent, C.; Espinilla, M. Ensemble classifier of long short-term memory with fuzzy temporal windows on binary sensors for activity recognition. *Expert Syst. Appl.* 2018, 114, 441–453.
65. Krishnan, N.C.; Cook, D.J. Activity recognition on streaming sensor data. *Pervasive Mob. Comput.* 2014, 10, 138–154.
66. Radu, V.; Lane, N.D.; Bhattacharya, S.; Mascolo, C.; Marina, M.K.; Kawsar, F. Towards multimodal deep learning for activity recognition on mobile devices. In *Proceedings of the 2016 ACM International Joint Conference on Pervasive and Ubiquitous Computing: Adjunct, Heidelberg, Germany, 12 September 2016*; pp. 185–188.
67. Cruciani, F., Sun, C., Zhang, S., Nugent, C., Li, C., Song, S., ... & Mccullagh, P. (2019, August). A public domain dataset for human activity recognition in free-living conditions. In *2019 IEEE SmartWorld, Ubiquitous Intelligence & Computing, Advanced & Trusted Computing, Scalable Computing & Communications, Cloud & Big Data Computing, Internet of People and Smart City Innovation (SmartWorld/SCALCOM/UIC/ATC/CBDCom/IOP/SCI)* (pp. 166-171). IEEE.
68. Ye, J., & Huang, G. (2017, January). Real time monitoring to the odour of excrement for health of infants and elderly completely bedridden. In *International Conference on Innovative Optical Health Science (Vol. 10245, p. 102450H)*. International Society for Optics and Photonics.
69. Zhu, M. R., & Guo, J. (2019, July). Design of Intelligent Defecation Monitoring System for Newborns. In *Proceedings of the 2019 International Conference on Intelligent Medicine and Health* (pp. 8-11).
70. Inoue, D., Iikuni, T., Kanayama, Y., Kim, J., & Nakajima, K. (2019). Distinction of urination or defecation using a gas sensor placed under the toilet seat. *Advanced Biomedical Engineering*, 8, 60-62.

71. Bae, J. H., & Lee, H. K. (2018). User health information analysis with a urine and feces separable smart toilet system. *Ieee Access*, 6, 78751-78765
72. C. Zhong, A. R. Reibman, H. M. Cordoba and A. J. Deering, "Hand-hygiene activity recognition in egocentric video," 2019 IEEE 21st International Workshop on Multimedia Signal Processing (MMSP), 2019, pp. 1-6, doi: 10.1109/MMSP.2019.8901753.
73. Kutafina E, Laukamp D, Jonas SM. Wearable Sensors in Medical Education: Supporting Hand Hygiene Training with a Forearm EMG. *Stud Health Technol Inform.* 2015;211:286-91. PMID: 25980884.
74. Brinke, J. K., Chiumento, A., & Havinga, P. (2021, February). Personal hygiene monitoring under the shower using Wi-Fi channel state information. In 1st Workshop on Computer Human Interaction in IoT Applications, CHIIoT 2021.
75. Sahakian, A. B., Jee, S. R., Pimentel, M. 2010. Methane and the gastrointestinal tract. *J. Dig Dis Sci*, 2010, 55: 2135--2143.
76. MQ8 Gas Sensor. <https://dlnmh9ip6v2uc.cloudfront.net/datasheets/Sensors/Biometric/MQ-8.pdf> (retrieved 24/12/2021)
77. MQ2 Gas Sensor. <http://gas-sensor.ru/pdf/combustible-gas-sensor.pdf> (retrieved 24/12/2021)
78. Srivastava, D., Kesarwani, A., & Dubey, S. (2018). Measurement of Temperature and Humidity by using Arduino Tool and DHT11. *International Research Journal of Engineering and Technology (IRJET)*, 5(12), 876-878.
79. Lopez Medina, M. A., Espinilla, M., Paggeti, C., & Medina Quero, J. (2019). Activity recognition for iot devices using fuzzy spatio-temporal features as environmental sensor fusion. *Sensors*, 19(16), 3512.
80. Medina, J., Espinilla, M., Zafra, D., Martínez, L., & Nugent, C. (2017, November). Fuzzy fog computing: A linguistic approach for knowledge inference in wearable devices. In *International conference on ubiquitous computing and ambient intelligence* (pp. 473-485). Springer, Cham.
81. Upton, E.; Halfacree, G. *Raspberry Pi User Guide*; John Wiley & Sons: Hoboken, NJ, USA, 2014.
82. Monteiro, A.; de Oliveira, M.; de Oliveira, R.; Da Silva, T. Embedded application of convolutional neural networks on Raspberry Pi for SHM. *Electron. Lett.* 2018, 54, 680–682.
83. Cooney, M., & Bigun, J. (2017). PastVision+: Thermovisual inference of recent Medicine intake by Detecting heated Objects and cooled lips. *Frontiers in Robotics and AI*, 4, 61.

84. Soni, D., & Makwana, A. (2017, April). A survey on mqtt: a protocol of internet of things (iot). In International Conference On Telecommunication, Power Analysis And Computing Techniques (ICTPACT-2017).
85. Soni, D., & Makwana, A. (2017, April). A survey on mqtt: a protocol of internet of things (iot). In International Conference On Telecommunication, Power Analysis And Computing Techniques (ICTPACT-2017) (Vol. 20).
86. Khan, N., McClean, S., Zhang, S., & Nugent, C. (2016). Optimal parameter exploration for online change-point detection in activity monitoring using genetic algorithms. *Sensors*, 16(11), 1784.
87. Min, P. Y. G. (2001). The Fundamental Principle and Application of Sliding Average Method [J]. *Journal of Gun Launch&Control*, 1.
88. Monk, S. *Programming the Raspberry Pi: Getting Started with Python*; McGraw-Hill Education: New York, NY, USA, 2016.
89. Gulli, A.; Pal, S. *Deep Learning with Keras*; Packt Publishing Ltd.: Birmingham, UK, 2017.
90. Wyse, L. Audio spectrogram representations for processing with convolutional neural networks. *arXiv* 2017, arXiv:1706.09559.
91. Ordóñez, F.; De Toledo, P.; Sanchis, A. Activity recognition using hybrid generative/discriminative models on home environments using binary sensors. *Sensors* 2013, 13, 5460–5477.
92. Salamon, J.; Bello, J.P. Deep convolutional neural networks and data augmentation for environmental sound classification. *IEEE Signal Process. Lett.* 2017, 24, 279–283.
93. Ciresan, D.C.; Meier, U.; Masci, J.; Gambardella, L.M.; Schmidhuber, J. Flexible, high performance convolutional neural networks for image classification. In *Proceedings of the Twenty-Second International Joint Conference on Artificial Intelligence*, Barcelona, Spain, 16–22 July 2011.
94. Cao, Z., Simon, T., Wei, S. E., & Sheikh, Y. (2017). Realtime multi-person 2d pose estimation using part affinity fields. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 7291-7299).
95. Cao, Z., Hidalgo, G., Simon, T., Wei, S. E., & Sheikh, Y. (2019). OpenPose: realtime multi-person 2D pose estimation using Part Affinity Fields. *IEEE transactions on pattern analysis and machine intelligence*, 43(1), 172-186.
96. He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 770-778).

97. Lupión, M., Medina-Quero, J., Sanjuan, J. F., & Ortigosa, P. M. (2021). DOLARS, a Distributed On-Line Activity Recognition System by Means of Heterogeneous Sensors in Real-Life Deployments—A Case Study in the Smart Lab of The University of Almería. *Sensors*, 21(2), 405.
98. Bajer, M. (2017, August). Building an IoT data hub with Elasticsearch, Logstash and Kibana. In 2017 5th International Conference on Future Internet of Things and Cloud Workshops (FiCloudW) (pp. 63-68). IEEE.