



UNIVERSIDAD DE CASTILLA-LA MANCHA
ESCUELA SUPERIOR DE INFORMÁTICA

GRADO EN INGENIERÍA INFORMÁTICA
INGENIERÍA DE COMPUTADORES

TRABAJO FIN DE GRADO

Monitorización de ocupación y elaboración de planes de
desinfección en el contexto de una pandemia

David Camuñas Sánchez

Septiembre, 2021



UNIVERSIDAD DE CASTILLA-LA MANCHA
ESCUELA SUPERIOR DE INFORMÁTICA

Tecnologías y Sistemas de la Información

INGENIERÍA DE COMPUTADORES

TRABAJO FIN DE GRADO

**Monitorización de ocupación y elaboración de planes de
desinfección en el contexto de una pandemia**

Autor: David Camuñas Sánchez

Tutor: Dr. Javier Alonso Albusac Jiménez

Co-Tutor: Dr. Iván González Díaz

Septiembre, 2021

Monitorización de ocupación y elaboración de planes de desinfección
© David Camuñas Sánchez, 2021

Este documento se distribuye con licencia *Creative Commons 4.0*. El texto completo de la licencia puede obtenerse en <https://creativecommons.org/licenses/by-nc-sa/4.0/>.

La copia y distribución de esta obra está permitida en todo el mundo, sin regalías y por cualquier medio, siempre que esta nota sea preservada. Se concede permiso para copiar y distribuir traducciones de este libro desde el español original a otro idioma, siempre que la traducción sea aprobada por el autor del libro y tanto el aviso de copyright como esta nota de permiso, sean preservados en todas las copias.

Este texto ha sido preparado con la plantilla \LaTeX de TFG para la UCLM publicada por *Jesús Salido* en GitHub y Overleaf, como parte del curso « *\LaTeX esencial para preparación de TFG, Tesis y otros documentos académicos*» impartido en la Escuela Superior de Informática de la Universidad de Castilla-La Mancha.



TRIBUNAL:

Presidente: _____

Vocal: _____

Secretario: _____

FECHA DE DEFENSA: _____

CALIFICACIÓN: _____

PRESIDENTE

VOCAL

SECRETARIO

Fdo.:

Fdo.:

Fdo.:

*A mis padres y mis abuelos
por su apoyo incondicional.*

Resumen

Nos encontramos en una situación excepcional, conviviendo con la pandemia COVID-19 que está dificultando de sobremanera nuestra vida diaria. En este contexto, proteger y garantizar la salud de los ciudadanos en los espacios públicos es un tema de preocupación. Para acometer esta labor y contener el virus, se deben cumplir con rigor determinados preceptos de higienización y protocolos de seguridad sanitaria, especialmente en espacios cerrados del interior de edificios. En este tipo de infraestructuras es común la existencia de espacios transitados donde es complicado llevar un control sobre las áreas con mayor trasiego de personas o con mayor densidad de ocupación a lo largo del tiempo. Son estas áreas, las que requieren ser desinfectadas prioritariamente y con mayor frecuencia. Por tanto, surge la necesidad de automatizar una distribución óptima de los recursos de desinfección disponibles en ese momento.

Con el fin de proporcionar datos cuantitativos de trasiego y ocupación que aporten información relevante para la elaboración de planes de desinfección, se desplegará una red de sensores inalámbricos (**WSN**) compuesta por nodos que incorporan sensores de movimiento (**PIR**) y transceptores de comunicación para análisis periódico de señales inalámbricas (WiFi/Bluetooth). El procesamiento de dichas señales permitirá estimar los niveles de presencia y densidad de ocupación.

Tras la obtención de los distintos datos relacionados con el estado del entorno de monitorización, se llevará a cabo un análisis del mismo, mediante una serie de gráficos y estadísticas. En base a dichos resultados se realizará una estimación de ocupación con el uso de las variables correspondientes al número de dispositivos **WiFi/BLE** y del sensor PIR. Esta estimación junto al nivel de concentración de CO_2 (obtenido a partir del sensor de CO_2 NDIR) representan un parámetro necesario a la hora de realizar una distribución óptima de los recursos de limpieza disponibles, en este caso, un equipo humano de limpieza, indicando así a que áreas (salas) de la infraestructura debe de ir destinado cada componente.

Abstract

We are in an exceptional situation living together with the COVID-19 pandemic which is making our daily lives absolutely difficult. In this context, the matter of concern is protecting and ensuring the health of citizens in public spaces. To undertake this work and stop the virus, certain sanitation precepts and health safety protocols must be strictly carried out, especially in indoor places inside buildings. In this kind of infrastructures, it is common the existence of transited spaces where it is difficult to carry out a control on the areas with the highest traffic of people or with the highest density of occupation over time. These areas are required to be disinfected as a priority and frequently. Therefore, the necessity arises to automate an optimal distribution of disinfection resources available at that time.

With the aim of providing quantitative data on transit and occupation that provide relevant information for the preparation of disinfection plans, a wireless sensor network will be deployed (WSN) made up of nodes that incorporate motion sensors (PIR) and communication transceivers for periodic analysis of wireless signals (WiFi/Bluetooth). The processing of these signals will allow the estimation of presence levels and occupancy density.

After obtaining the different data related to the state of the monitoring environment, an analysis of the same will be carried out, by means of a series of graphs and statistics. Based on these results, an occupancy estimation will be carried out using the variables related to the number WiFi/BLE devices and the PIR sensor. This estimation together with the CO_2 concentration level (obtained from the CO_2 NDIR sensor) represents a necessary parameter for making an optimal distribution of the available cleaning resources, in this case, a human cleaning team, indicating to which areas (rooms) of the infrastructure each component should be assigned.

AGRADECIMIENTOS

Quisiera comenzar dando las gracias a mis padres y mi hermana, *Ana Belén, Oscar José y Helena*, por hacerme ser quien soy, creer siempre en mí y ofrecerme la oportunidad de formarme en lo que he querido.

Gracias a mis abuelos, Francisco y Adoración, por su constante preocupación por mí y cuidarme como a un hijo.

Dar las gracias a mi entorno familiar más cercano, como mis tíos, por apoyarme en todo momento.

Gracias a *Elena*, por tratar de entenderme hasta en los peores momentos y ayudarme a lo largo de esta etapa. Por transmitirme tu calma y convertirte en un pilar fundamental en mi día a día.

Agradecer a mi compañero y buen amigo Mario, todos los momentos vividos a lo largo de esta etapa, atrás quedan las incontables horas de trabajo y esfuerzo conjunto para conseguir este tan ansiado objetivo. He de recalcar, que sin tu amistad nada hubiese sido igual.

Por último, he de agradecer a mis tutores, Javier e Iván, haberme ofrecido la oportunidad de realizar este proyecto y guiarme a lo largo de todo su desarrollo.

David Camuñas Sánchez
Ciudad Real, 2021

ÍNDICE GENERAL

Resumen	IX
Abstract	XI
Agradecimientos	XIII
Índice de figuras	XIX
Índice de tablas	XXI
Índice de listados	XXIII
Índice de algoritmos	XXV
Listado de acrónimos	XXV
1. Introducción	1
1.1. Estructura del documento	4
2. Objetivos	7
2.1. Objetivo General	7
2.2. Objetivos Específicos	7
3. Estado del Arte	9
3.1. Monitorización de densidad de ocupación en entornos de interior	9
3.1.1. Aplicación de cámaras para la detección de ocupación	12
3.1.2. Aplicación del sensor o transceptor BLE para la detección de dispositivos inteligentes	13
3.1.3. Aplicación del sensor o transceptor WiFi para la detección de dispositivos inteligentes	14
3.1.4. Aplicación de dispositivos para la captación de señales LTE	14
3.2. Sistemas orientados a la monitorización de la densidad de ocupación en entornos de interior	15
3.3. Estudio y elección del entorno <i>hardware</i>	23
4. Metodología	27
4.1. Metodología de trabajo	27
4.2. Distribución del trabajo	28
4.2.1. Paquetes de trabajo	28
4.2.2. Iteraciones	32
4.3. Recursos	34

4.3.1.	Recursos <i>Hardware</i>	34
4.3.2.	Recursos <i>Software</i>	35
5.	Arquitectura	39
5.1.	Visión General	39
5.2.	Arquitectura del Sistema	40
5.2.1.	Aplicación de monitorización del tránsito humano	40
5.2.2.	Aplicación para la distribución óptima de recursos de limpieza	41
5.2.3.	<i>Backend</i>	43
5.3.	Despliegue de nodos para la sensorización del entorno	44
5.3.1.	Visión general	44
5.3.2.	<i>Door Node</i>	45
5.3.3.	<i>Scan Node</i>	49
5.3.4.	Comunicación con el <i>Backend</i> del sistema	53
5.4.	Aplicación de monitorización del tránsito humano	55
5.4.1.	Entradas y salidas	56
5.4.2.	Representación de la información	57
5.5.	Aplicación de distribución óptima de recursos de limpieza	57
5.5.1.	Entradas y salidas	58
5.5.2.	Obtención y tratamiento de los datos	59
5.5.3.	Definición de la prioridad de cada sala	60
5.5.4.	Generación de planes de desinfección	61
5.5.5.	Definición del plan de desinfección más óptimo	62
5.6.	<i>Backend</i>	63
5.6.1.	Base de datos	64
5.6.2.	Contenedores	64
6.	Resultados	67
6.1.	Sistema Resultante	67
6.1.1.	Aplicación de monitorización del tránsito humano	67
6.1.2.	Aplicación de distribución de recursos de limpieza	68
6.2.	Experimentación	69
6.2.1.	Monitorización del tránsito humano	69
6.2.2.	Comparación prototipo de sensor CO_2 NDIR con medidor de CO_2 comercial	73
6.2.3.	Distribución óptima de los recursos de limpieza	74
7.	Conclusiones	77
7.1.	Objetivos alcanzados	78
7.2.	Justificación de competencias adquiridas	79
7.3.	Trabajo Futuro	79
	Bibliografía	81
A.	Repositorios del sistema	89
A.1.	Repositorio de la red de nodos WSN	89
A.2.	Repositorio del <i>Backend</i> del sistema	89
A.3.	Repositorio de la aplicación de distribución de recursos de limpieza	90
B.	Experimentación en un escenario simulado	91
C.	Imágenes del Sistema	93
C.1.	Despliegue de la red de nodos de sensorización	93
C.2.	Imágenes del entorno de monitorización	94

C.3. Imágenes de la aplicación de distribución de recursos de limpieza	96
D. Listados de configuración del sistema	97
D.1. Listado de aprovisionamiento del <i>dashboard</i>	97
D.2. Listado de configuración de <i>Docker Compose</i>	98

ÍNDICE DE FIGURAS

1.1.	Principales modos de transmisión del COVID-19 [48].	2
1.2.	Ruta de propagación del aerosol en diferentes entornos: (1) evaporación; (2) interacción con otras gotas; (3) deposición en la superficie de objetos; (4) impacto de la temperatura y la humedad relativa; (5) proceso de ventilación; [82].	3
3.1.	Representación de la detección de movimiento de un individuo mediante un sensor PIR. [81]	10
3.2.	Representación y estructura de un sensor de CO ₂ basado en NDIR [23].	11
3.3.	Diagrama de funcionamiento de un sistema de estimación de ubicación basado en vibración de suelo [30].	12
3.4.	Representación de la estimación de densidad de ocupación en base al uso de señales inalámbricas LTE [14].	15
3.5.	Orientación de los sensores PIR [81].	16
3.6.	Diagramas de la variante del sensor PIR, C-PIR [74, 75].	16
3.7.	Prototipo SLEEPiR y sus respectivos componentes; (1) Obturador LC, (2) Sensor PIR, (3) Placa del circuito, (4) Microcontrolador, (5) Batería [76].	17
3.8.	Representación del sistema propuesto en [51].	18
3.9.	Prototipo de silla con sensor de temperatura y representación del escenario de prueba [69].	18
3.10.	Representación del sistema <i>BloT</i> propuesto en [26].	19
3.11.	Área perimetral del conjunto de edificios que forman el hospital, junto las rutas generadas [52].	20
3.12.	Representación del nodo del sistema, y su disposición por salas [35]	21
3.13.	Representación basada en los modos de funcionamiento WiFi de los dispositivos inteligentes; Círculo de color naranja como representación del área o zona de servicio de un punto de acceso (AP) [71].	22
3.14.	Diagrama de funcionamiento de la plataforma BOSSA [37].	23
3.15.	Representación de componentes y placas de desarrollo de la compañía <i>Arduino</i>	24
3.16.	Representación de microcontrolador y placa de desarrollo para ESP32-WROOM-32 11.	24
4.1.	Representación del desarrollo iterativo e incremental.	27
5.1.	Esquema general del sistema propuesto.	39
5.2.	Esquema de la Arquitectura del sistema	42
5.3.	Esquema general del comportamiento de los nodos.	44
5.4.	Diagrama del prototipo <i>hardware</i> de <i>Door Node</i>	45
5.5.	Esquema detallado sobre el funcionamiento de <i>Door Node</i>	46
5.6.	Comparación ambos campos de visión del sensor PIR.	47
5.7.	Sensor de detección PIR digital [53].	47
5.8.	Diagrama de conexiones del prototipo <i>hardware</i> de <i>Scan Node</i>	49
5.9.	Esquema detallado sobre el funcionamiento de <i>Scan Node</i>	50

5.10. Diagrama de comunicación entre la red de nodos WSN y el <i>Backend</i> , mediante el <i>broker</i> MQTT.	54
5.11. Flujo de comunicación entre la infraestructura que conforma la red de nodos de sensorización y la parte del <i>Backend</i> encargada de su gestión.	55
5.12. Diagrama de casos de uso de la aplicación de monitorización.	55
5.13. Esquema general aplicación monitorización.	56
5.14. Ejemplo de generación de una gráfico en base a una selección de datos mediante consulta SQL.	57
5.15. Diagrama de casos de uso de la aplicación de distribución óptima de recursos de limpieza.	58
5.16. Esquema general aplicación de distribución óptima de recursos de limpieza.	58
5.17. Planes de desinfección generados por la aplicación de distribución de recursos para un entorno determinado.	62
5.18. Esquema general <i>Backend</i> del sistema	64
5.19. Comparación del soporte ofrecido por un despliegue de máquinas virtuales y de <i>Docker</i>	65
6.1. Pantalla principal aplicación de motorización.	68
6.2. Pantalla principal de la aplicación de distribución de recursos	68
6.3. Resultados sobre la estimación de ocupación por cada sala.	72
6.4. Comparativa entre los valores promedios de cada una de las variables obtenidas por lo sensores y la ocupación real de cada sala.	72
6.5. Resultados de CO_2 obtenido a partir del sensor NDIR.	73
6.6. Nodo desarrollado encargado de la medición de CO_2 y el medidor comercial.	73
6.7. Comparación de resultados sobre el nivel de CO_2 obtenidos a través del sensor NDIR de <i>Scan Node</i> y del medidor comercial <i>Selensy</i>	74
A.1. Código QR del repositorio de la red WSN.	89
A.2. Código QR del repositorio del <i>Backend</i> del sistema.	90
A.3. Código QR del repositorio de la aplicación de distribución de recursos de limpieza.	90
C.1. Fotografías de los dos nodos instalados en una de las salas.	93
C.2. Fotografías de la Sala 1 en diferentes instantes de tiempo.	94
C.3. Acceso a Sala 1.	95
C.4. Interfaz de escritorio de la aplicación de distribución de recursos.	96

ÍNDICE DE TABLAS

3.1. Modos de funcionamiento del módulo WiFi integrado en los dispositivos inteligentes [71].	21
3.2. Comparación técnica de los microcontroladores <i>ESP8266</i> y <i>ESP32-WROOM-32</i> 11.	25
4.1. Representación de la iteración 1	32
4.2. Representación de la iteración 2	32
4.3. Representación de la iteración 3	33
4.4. Representación de la iteración 4	33
4.5. Representación de la iteración 5	34
4.6. Representación de la iteración 6	34
5.1. Relación entre los valores obtenidos fruto de la conversión analógico-digital y su equivalencia con el nivel de calidad del aire.	52
6.1. Resultados sala 1.	71
6.2. Resultados sala 2.	71
6.3. Resultados sala 3.	71
6.4. Resultados distribución más óptima.	75
6.5. <i>Ranking</i> de las diez mejores distribuciones de recursos.	75
B.1. Resultados de la distribución más óptima en el escenario simulado.	91
B.2. <i>Ranking</i> de las diez mejores distribuciones de recursos en el escenario simulado.	92

ÍNDICE DE LISTADOS

5.1.	Algoritmo de contabilización de ocupantes	48
5.2.	Algoritmo para el cálculo del nivel de concentración de CO_2	51
5.3.	Secuencia de código desarrollada para la definición del modo de conexión utilizado en Scan Node	52
5.4.	Secuencia de código desarrollada para el análisis de señales BLE	53
5.5.	Código correspondiente al bucle principal del <i>firmware</i> de <i>Scan Node</i>	53
5.6.	Resultado obtenido en formato JSON sobre el estado de una las salas que componen el entorno	59
5.7.	Algoritmo para la definición de prioridad de cada sala	61
5.8.	Algoritmo para la generación de planes de desinfección	62
5.9.	Algoritmo para la generación de planes de desinfección	63
D.1.	Fichero de aprovisionamiento configuración del <i>dashboard</i>	97
D.2.	Configuración del fichero <i>docker compose</i> de los contenedores del <i>Backend</i>	98

LISTADO DE ACRÓNIMOS

BLE	Bluetooth Low Energy
WiFi	Wireless Fidelity
GPS	Global Positioning System
CSI	Channel State Information
LTE	Long Term Evolution
MQTT	Message Queue Telemetry Transport
RFID	Radio Frequency Identification
WSN	Wireless Sensor Network
PIR	Passive Infrared
NIC	Network Interface Controller
RSSI	Received Signal Strength Indicator
FOV	Field of Vision
MAC	Media Access Control
SQL	Structured Query Language
JSON	JavaScript Object Notation
PIR	Passive Infrared
SoC	System on Chip
GUI	Graphical User Interface
YAML	Ain't Markup Language
TSDB	Time Series Database
SSL	Secure Sockets Layer
PCB	Printed Circuit Board

INTRODUCCIÓN

Las pandemias han formado parte de la vida del ser humano. A lo largo de la historia ha existido una batalla constante contra este tipo de adversidades, cuyo objetivo ha sido la erradicación y control de los diferentes tipos de enfermedades infecciosas que los provocan. La supervivencia de la mayoría de agentes microbiológicos, como los virus y bacterias, depende de su capacidad de infectar a la población humana. En cambio, las infecciones han ganado muchas batallas, provocando la muerte de millones de personas y cambiando en la mayoría de ocasiones la historia de la humanidad [42].

A comienzos de 2019 se pronosticó la aparición de nuevas enfermedades por coronavirus, presuntamente de origen animal en el mercado de mariscos de *Hunan*, perteneciente a la ciudad china de *Wuhan*. El coronavirus causante fue denominado *COVID-19* por parte de la OMS. Posteriormente, diversos estudios demostraron que, realmente, un alto porcentaje de los pacientes no habían visitado el mercado, pero sí zonas cercanas a la ciudad de *Wuhan*. Al conocer esta información, la mayoría de los países del mundo informaron sobre *la propagación de enfermedades causada por la inhalación de microgotas o aerosoles en el tracto respiratorio y los pulmones*. Las infecciones por COVID-19 siguieron un crecimiento exponencial reportando numerosos casos en todo el mundo, declarándose así el COVID-19 como pandemia a nivel mundial. En marzo de 2020, un modelo matemático estudió posibles formas de controlar o frenar la transmisión de este virus, concluyendo en que la aplicación de una cuarentena domiciliaría voluntaria de las personas que hubiesen mantenido contacto con un infectado por COVID-19, sería fundamental para frenar su propagación [38].

Como se ha mencionado anteriormente, la propagación del COVID-19 ocurre mediante el contacto cercano de un individuo con otro, a través de micro-gotas respiratorias, cuando una persona contagiada tose, estornuda e incluso habla. Por lo general, el recorrido de estas gotas pueden llegar a sobrepasar los dos metros de distancia y permanecer en el aire durante un tiempo limitado (este tiempo puede llegar hasta un máximo de tres horas). Por lo tanto, la ventilación en interiores y la aplicación adecuada de desinfectante es una de las medidas más importantes contra este virus.

El nivel de riesgo de infección está fuertemente relacionado a ciertos comportamientos y características de las personas. Una de las principales causas de propagación de muchas enfermedades respiratorias son los contactos cercanos. Con este tipo de contactos, el distanciamiento social se ve claramente decrementado, lo que facilita la inhalación de este tipo de gotas o aerosoles. De este modo, el riesgo de transmisión o propagación entre distintos individuos, siempre y cuando estos no hagan uso de ningún medio de protección, depende exclusivamente de la distancia que se establezca entre un individuo y otro; junto al tiempo de exposición. Por lo tanto, un distanciamiento social menor implica una mayor inhalación del virus, aumentando así el riesgo de infección.

En este contexto, varios organismos y países del mundo han recomendado mantener una distancia mínima interpersonal entorno a 1 y 2 metros de distancia, al ser esta una medida razonable para reducir el riesgo de infección, siempre que no se vea alterado por otros factores que favorezcan la transmisión. Sin embargo, diversos estudios de investigación intentan establecer la distancia de seguridad más adecuada, debido a que no existe un distanciamiento único que establezca una total

seguridad frente a estos riesgos. En este sentido, pueden darse otros factores que obliguen a variar la distancia de seguridad, como pueden ser la expulsión en forma de aerosoles por parte de uno de los individuos o incluso haber mantenido contacto con superficies contaminadas [40]. La Figura 1.1 muestra los principales modos de transmisión identificados [48].

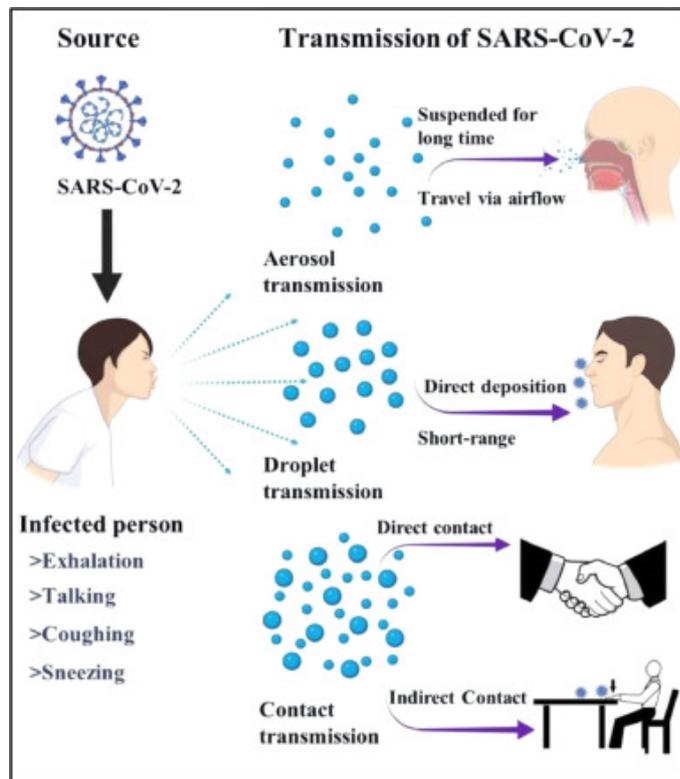


Figura 1.1: Principales modos de transmisión del COVID-19 [48].

Recientes estudios de investigación [49] y expertos como *E.Bromage* [7] indicaron que la mayoría de las transmisiones son el resultado de la propagación de la infección en el hogar. Si un miembro de la familia se infecta fuera de la vivienda, los otros miembros, generalmente, se contagian cuando la persona infectada regresa. Este estudio ha sido confirmado por *H.Qian* [49], que identificó 318 brotes donde se daban tres o más casos de este tipo, involucrando así a 1245 casos confirmados en más de 120 ciudades. Los resultados muestran una fuerte correlación entre la infección y la convivencia en espacios interiores.

Los estudios científicos relacionados con la transmisión de COVID-19 en el interior de edificios, y la propagación en los entornos y hogares familiares, junto con los propios contactos que se generen, ponen en sobre aviso de que en un entorno interior (dentro de una arquitectura cerrada), como viviendas o espacios similares, existe un elevado riesgo de transmisión por proximidad, en especial si un ocupante está infectado. Debido a las características que representan este tipo de espacios y el comportamiento de los ocupantes dentro de ellos, ya que por lo general suele haber un mayor grado de interacción, especialmente en viviendas modestas. En estos casos, el riesgo de transmisión está estrechamente relacionado con: la distancia, la calidad del aire, las acciones relacionadas, como la expulsión de gotas contaminadas provocada por cualquier acción (hablar, estornudar, toser o simplemente respirar), además del tiempo de exposición. Si bien, estornudar o toser implica la expulsión de aerosoles suficientes para infectar a otro individuo de forma directa, unos minutos hablando cara a cara con una persona infectada o simplemente veinte minutos expuesto a su respiración pueden ser suficientes para infectarse, en función de la cantidad de gotas expulsadas y recibidas por unidad de tiempo [7].

Los aerosoles son el principal medio de transmisión aérea del COVID-19, caracterizado por un

tamaño de partícula pequeño y un tiempo de supervivencia prolongado en el aire, por lo que la capacidad de transmisión del virus se ve muy afectada por el flujo de aire. La mala ventilación puede incrementar la transmisión de aerosoles. *D.Guo*, junto a otros investigadores [25] encontraron una gran cantidad de muestras positivas al analizar el aire en diferentes ubicaciones del *Hospital Huoshenshan*. Una situación similar surgió en Francia el 12 de marzo de 2020, donde 19 participantes fueron diagnosticados con COVID-19 después de una práctica de coro en un espacio no ventilado [11].

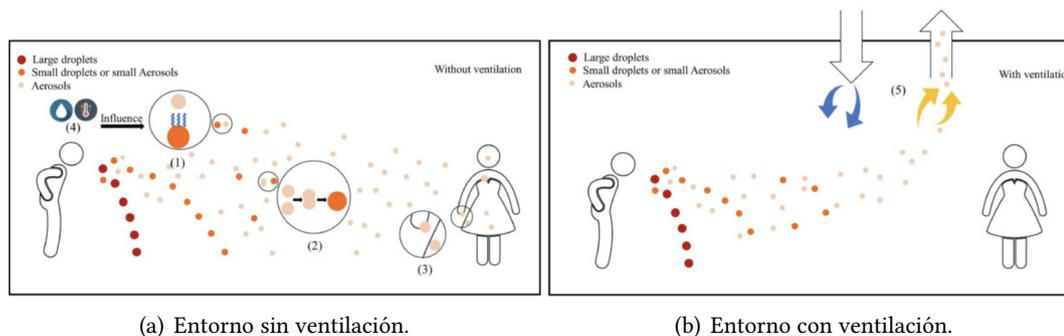


Figura 1.2: Ruta de propagación del aerosol en diferentes entornos: (1) evaporación; (2) interacción con otras gotas; (3) deposición en la superficie de objetos; (4) impacto de la temperatura y la humedad relativa; (5) proceso de ventilación; [82].

Como se muestra en la Figura 1.2, tanto en el entorno (a) como en el (b) la evaporación afecta a la vida útil que puede tener un aerosol dependiendo de su tamaño. Los aerosoles grandes se sedimentan rápidamente debido a la fuerza que ejerce la gravedad sobre ellos. En cambio, los aerosoles más pequeños pueden mantenerse suspendidos en el aire durante un largo periodo de tiempo. En este caso, la evaporación de este tipo de gotas depende en gran medida de la humedad. Esto se debe a que en ambientes caracterizados por su baja humedad y su alta temperatura, los aerosoles aceleran su evaporación. Teniendo esta información en cuenta, *X.Xie* [77] indicó que el proceso de evaporación conduce a una rápida reducción del tamaño de este tipo de partículas, obteniendo como resultado que más gotas en forma de aerosol se dispersen en el aire, aumentando así el riesgo de propagación y la durabilidad del virus. Por el contrario, en el escenario (b) se puede observar como el sistema de ventilación juega un papel fundamental en la propagación de aerosoles. Donde la ruta de propagación de estas partículas se ve afectada, pudiendo ser expulsadas hacia el exterior y disminuir así la *adhesión humana*. Por lo tanto, un sistema de ventilación puede mitigar en gran medida los riesgos de infección a través de la transmisión aérea. [82].

En este contexto, surge la necesidad de proteger y garantizar la salud de los ciudadanos tanto en los espacios públicos, como en los entornos más familiares. Para llevar a cabo esta labor y contener el virus, se debe acometer, con rigor, determinados preceptos de higienización y protocolos de seguridad sanitaria, especialmente en espacios cerrados en el interior de edificios. En infraestructuras públicas es común la existencia de espacios transitados donde es complicado llevar un control sobre las áreas con mayor densidad de ocupación a lo largo del tiempo, estas áreas son las que requieren ser desinfectadas prioritariamente y con mayor frecuencia.

Por lo general, las labores de desinfección realizadas por parte de equipos humanos mediante el uso de productos de limpieza, están produciendo buenos resultados. Este tipo de desinfección forma parte de las estrategias clásicas de desinfección previamente conocidas [12], donde la asepsia se realiza mediante contacto directo con la superficie infectada, asegurando así una esterilización eficaz y rápida frente a los virus. Por otra parte, la propagación de aparatos de desinfección por ozono ha aumentado, de forma notoria, desde que se declaró la pandemia por COVID-19, debido a que se trata de una solución de bajo coste en comparación de los equipos humanos de limpieza y evita, en cierta manera, la exposición directa al patógeno. El ozono se ha notificado como sustancia biocida

para uso ambiental, si bien, no han sido aprobados todavía a nivel europeo los usos que se le pueden dar, esto lo convierte en una solución poco eficaz. Ante todo esto, surge la necesidad de maximizar la eficiencia que ofrecen los equipos de limpieza formados por recursos humanos. Una de las ideas más novedosas en este sector es la introducción de dispositivos de monitorización, para así poder recopilar toda la información posible del estado en el que se encuentra el entorno, y en base a esta información realizar una redistribución optimizada de los recursos humanos que forman el equipo de limpieza.

El presente proyecto propone una solución tecnológica de bajo coste que podría ser de ayuda a la hora de monitorizar el estado de entornos cerrados, basada en un buen control de la densidad de ocupación y calidad de aire. A partir de la información adquirida, se pretende llevar a cabo la elaboración de una serie de planes de desinfección, con el objetivo de maximizar la eficiencia de los recursos disponibles. Se dotará al usuario de información útil y detallada sobre aspectos relacionados con el estado de las distintas salas de las que está formada su infraestructura, así como la automatización de la toma de decisiones en lo relativo a la planificación de desinfección, en aras de optimizar los recursos disponibles.

Para lograr materializar la propuesta anterior, se pretende realizar el despliegue de una red de nodos del tipo WSN (*Wireless Sensing Network*), donde habrá dos clases de nodos; un primer nodo o *nodo de escaneo* exclusivamente dedicado al análisis de señales WiFi y Bluetooth provenientes de los dispositivos inteligentes dentro del entorno a monitorizar, junto con la evaluación de la calidad del aire de dicho entorno. El análisis de las señales se realizará con el uso de sensores de captación de este tipo de señales (también denominados transeptores), y para el análisis de la calidad del aire se hará uso de un sensor de medición de CO_2 NDIR. En cuanto al segundo nodo, denominado *nodo principal* será el encargado de realizar un conteo de las personas que vayan entrando en el entorno o espacio cerrado mediante el uso de un sensor PIR (*sensor de infrarrojos pasivo*). Cada nodo enviará los datos recopilados por los distintos sensores a un servidor desplegado en la misma red, el cual, se encargará de almacenarlos. En dicho servidor, se iniciarán dos aplicaciones distintas, la primera mostrará información relacionada con el estado del entorno; por su parte, la segunda realizará, de forma automática, la elaboración de planes de desinfección en base a los recursos de desinfección disponibles y los datos previamente obtenidos del entorno monitorizado. Posteriormente, esta aplicación mostrará el resultado obtenido de dicha planificación, para ello, se representará un plano del entorno monitorizado donde se verá de forma detallada los recursos destinados a cada uno de los espacios cerrados (salas) que conformen parte de este junto con su prioridad de desinfección.

1.1. ESTRUCTURA DEL DOCUMENTO

En esta sección se describirá la estructura por la que está compuesto este Trabajo Fin de Grado.

- **Capítulo 1: *Introducción***

Se trata del capítulo actual del documento, cuya finalidad es presentar el contexto de este proyecto junto a una propuesta general a la problemática expuesta.

- **Capítulo 2: *Objetivos***

Este capítulo se centra en explicar de forma detallada el objetivo principal de este proyecto. El objetivo principal estará compuesto por una serie de hitos u objetivos parciales, los cuales, cumpliéndolos se logrará obtener el objetivo principal.

- **Capítulo 3: *Estado del arte***

El capítulo del estado del arte del proyecto, se trata de un estudio detallado de las soluciones existentes dentro del contexto actual del proyecto. Este capítulo servirá como base para poder conseguir los distintos objetivos definidos, creando una base teórica ayudará a tomar mejores decisiones de diseño. En el caso del presente proyecto, se realizará un estudio detallado de

sobre diversas tecnologías usadas a la hora de estimar la densidad de ocupación en espacios cerrados, también denominados entornos de interior. Este estudio servirá de gran ayuda a la hora de elegir la tecnología más apropiada para este proyecto.

▪ **Capítulo 4: Metodología**

En este capítulo se describe la metodología de desarrollo utilizada en el proceso de desarrollo de este proyecto, además se tratarán las distintas herramientas y recursos usados para poder llevarlo a cabo.

▪ **Capítulo 5: Arquitectura**

Este capítulo hace hincapié en la descripción técnica del sistema resultante del proyecto. Para ello, se explicarán los distintos problemas encontrados y qué soluciones han sido elegidas, explicando el por qué ha sido elegida dicha solución.

▪ **Capítulo 6: Resultados**

Dentro de este capítulo se realizará un plan de pruebas para así poder comprobar la eficacia del prototipo diseñado.

▪ **Capítulo 7: Conclusión**

Por último, en este capítulo se resaltarán los aspectos más importantes relacionados con la realización del proyecto, justificando las competencias adquiridas y poniendo potenciales líneas de trabajo futuro.

OBJETIVOS

En este capítulo se pretende llevar a cabo una descripción detallada sobre el objetivo principal que este proyecto pretende cumplir y, al mismo tiempo, el conjunto de subobjetivos para poder alcanzarlo.

2.1. OBJETIVO GENERAL

El objetivo general de este proyecto es desarrollar un sistema que permita la monitorización automática en espacios cerrados, en un contexto de pandemia, poniendo el foco en el análisis de tránsito de personas y la calidad del aire. Posteriormente, con los datos obtenidos, se pretende generar, de forma automática, planes de desinfección que faciliten una distribución de recursos adecuada. Principalmente, recursos humanos para la limpieza y desinfección.

2.2. OBJETIVOS ESPECÍFICOS

En base al objetivo general descrito anteriormente, se requiere identificar objetivos más específicos los cuales, una vez alcanzados, permitirán lograr el objetivo general.

Los objetivos específicos son los siguientes:

Estudio y evaluación de dispositivos *hardware* para la monitorización de salas

En primer lugar será necesario estudiar las capacidades computacionales y las limitaciones de diferentes dispositivos *hardware* dentro del ámbito *IoT* que se puedan encontrar en el mercado, y así poder elegir los que mejor se adapten a las necesidades del proyecto.

Concretamente, se usarán dispositivos relacionados con la detección de movimiento y la presencia de dispositivos cercanos, mediante el análisis de señales WiFi y BLE. Por otra parte, se usarán dispositivos relacionados con el análisis de la calidad del aire, a través de la medición del nivel de *CO2*.

Despliegue de una red de sensores para la adquisición de datos

Para cumplir con el objetivo general de este proyecto será necesario el despliegue de una infraestructura formada por varios nodos. Los nodos integrarán distintos tipos de sensores que actuarán de una manera conjunta para poder obtener datos relacionados con la presencia y la ocupación en el entorno a monitorizar, además de la calidad del aire en dicho entorno.

Análisis de ocupación y calidad del aire en el entorno

A partir de los datos procedentes de los nodos distribuidos en el entorno monitorizado, se llevará a cabo el análisis la densidad de ocupación en el entorno y la calidad del aire. Estos datos deben mantenerse de forma persistente en el sistema para así poder mantener un histórico sobre la evolución de dicha sala a lo largo del tiempo.

Elaboración automática de planes de desinfección

En base a los resultados obtenidos del análisis anterior, el sistema tendrá la capacidad de elaborar un plan de desinfección de forma automática. El plan de desinfección se basará, principalmente, en la distribución espacial y temporal de los recursos humanos disponibles.

Desarrollo de varias interfaces de usuario para la visualización de los resultados de monitorización y planes elaborados

En primer lugar, se desarrollará una interfaz web donde se mostrará información de la red de nodos desplegada en el entorno, como el valor actual de los sensores, estadísticas y gráficos relacionados con los valores previamente obtenidos. Por otro lado, se desarrollará una interfaz de escritorio, donde se podrá visualizar de una forma intuitiva y detallada un mapa con el estado de cada sala, y el plan de desinfección más óptimo para las mismas. Este plan será realizado en base a los recursos de desinfección disponibles en ese momento.

ESTADO DEL ARTE

En este capítulo se van a presentar una serie de conceptos teóricos considerados relevantes para el desarrollo de este proyecto. Se trata de conceptos totalmente necesarios para poder llegar a comprender las decisiones que se han ido tomando en esta etapa de desarrollo. En primer lugar, se estudiará el uso de distintos tipos de tecnologías basadas en la detección de ocupación y localización de ocupantes en entornos de interior. En segundo lugar, se llevará a cabo un estudio de diversos trabajos considerados de relevancia relacionados con el uso de estas tecnologías. A continuación, para finalizar este capítulo se presentarán los dispositivos *hardware* utilizados para dar solución a esta propuesta.

3.1. MONITORIZACIÓN DE DENSIDAD DE OCUPACIÓN EN ENTORNOS DE INTERIOR

La estimación de densidad de ocupación puede ser de gran importancia en diversos ámbitos, sobre todo aquellos en los que la presencia humana es frecuente y desempeña un papel relevante. Este tipo de monitorización se utiliza a menudo en escenarios relacionados con la supervisión de seguridad humana [37], como es el caso del control de multitudes [59], control del tráfico [32], o incluso como guía inteligente en museos [79], entre otros. La estimación de la densidad de ocupación es primordial para que la administración evite un posible peligro potencial durante la realización de cualquier actividad de carácter social y/o público. Este tipo de actividades como reuniones, conferencias, zonas de transporte público, celebraciones, deportes, conciertos, por lo general son eventos donde una gran cantidad de personas son reunidas en un lugar determinado en un periodo de tiempo determinado. Los organizadores deben guiar y controlar a los asistentes, llevando a cabo la monitorización de ocupación, y así predecir el movimiento que realicen las personas, evitando cualquier potencial accidente. De lo contrario, en este tipo de actividades ante algún problema, pueden ocasionarse desastres masivos, debido a la pérdida de control. Por ejemplo, lo ocurrido en el *Madrid Arena* en noviembre del 2012, cuando murieron hasta cinco personas durante la celebración de una fiesta de *Halloween*. Tras realizar un estudio se concluyó que la causa fundamental de la tragedia fue el exceso de aforo y la apertura de un portón de carga, el cual, al ser utilizado súbitamente por miles de jóvenes, provocó la saturación de la pista central y de los pasillos de evacuación, provocando en uno de ellos una avalancha humana [24]. Por lo tanto, surge la necesidad de adoptar nuevas técnicas para evitar que desastres de esta magnitud puedan volver a ocurrir.

A continuación se presentarán las modalidades de detección más comunes para realizar una estimación precisa de la densidad de ocupación, además, se estudiará en detalle las características que ofrece cada modalidad.

Captación de movimiento empleando sensores PIR (*Passive InfraRed*)

Los sensores de infrarrojos piroeléctricos o PIR se usan para la detección de movimiento, aprovechando la piroelectricidad. Esta técnica permite detectar un cuerpo que no se encuentra en equilibrio térmico dentro de su entorno [45]. Este tipo de sensores son de gran utilidad en aplicaciones en las que es necesaria la detección de presencia de personas para activar algún tipo de evento. También se utilizan en aplicaciones más complejas, sirvan de ejemplo, la obtención de imágenes térmicas [3], la biometría [19, 20] y su empleo en termómetros [64]. Es habitual ver este tipo de sensores integrados en sistemas combinados con otra clase de tecnologías, como es el caso de los sistemas de vigilancia, donde los sensores PIR son de gran ayuda a la hora de diferenciar entre un individuo inmóvil y el fondo del escenario, o para detectar algunos cambios de dirección por parte de los individuos que estén actuando dentro del entorno capturado.

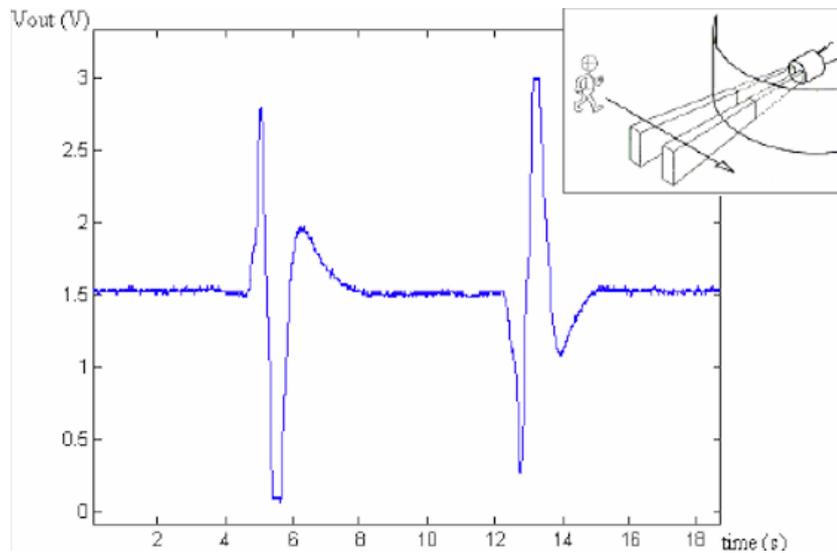


Figura 3.1: Representación de la detección de movimiento de un individuo mediante un sensor PIR. [81]

Los sensores piroeléctricos emiten un tipo de señal eléctrica como respuesta a un cambio en la radiación térmica incidente. El sensor lleva incorporado varias lentes de tipo *fresnel* que están diseñadas para dar forma al campo de detección. Por consiguiente, a medida que un individuo se mueve a través de este campo de detección, estas lentes detectan el cambio de flujo de radiación. En la Figura 3.1, se puede observar como funciona este tipo de sensor cuando un individuo pasa caminando por su campo de detección, generando dos picos opuestos, cada uno correspondiente al paso del individuo entre dos lentes de fresnel.

La principal limitación que presentan los sensores piroeléctricos es que son detectores de movimiento, y por lo tanto, son incapaces de detectar actividades estacionarias. El artículo [76] menciona una encuesta realizada, con el objetivo de saber el tiempo promedio que una persona pasa al día realizando actividades estacionarias. El resultado obtenido indica que ese tiempo se sitúa sobre las 11,8 horas. En cambio, entorno a esta problemática se han desarrollado una serie de proyectos que pretenden darle una solución óptima y que son mostrados en el citado artículo.

Aplicación de sensores ambientales

Existen diversos sensores de este tipo, como sensores de temperatura, CO, CO₂, humedad, luz o presión atmosférica [72]. POo lo general, se trata de sensores analógicos (que requieren una conversión

analógico-digital) y que generan datos brutos que contienen bastante *ruido*¹, en este contexto el ruido suele producirse por dispositivos/agentes que causen interferencias dentro del entorno. Por lo tanto, se requiere un proceso de extracción y selección de las características principales que estos ofrezcan. Los sensores ambientales suelen estar disponibles en la gran mayoría de entornos construidos y suelen utilizarse, junto con otros sensores, para la detección de ocupación y el conteo de multitudes, además son relativamente económicos, no intrusivos, y conservan la privacidad. es decir, el sistema no capta ninguna información personal de los individuos. Sin embargo, la eficacia de la estimación realizada tras obtener los datos del sensor puede verse afectada si surge un cierto retraso de tiempo en la adquisición (latencia) [86].

Sensor de CO_2 para la medición de la calidad del aire

El control del dióxido de carbono (CO_2) es clave a la hora de medir la calidad del aire tanto en los entornos de interior como de exterior. Los sensores de CO_2 han sido reconocidos como el sensor ambiental más eficaz para la estimación sobre la calidad del aire [9]. Una exposición prolongada en un ambiente donde por sus características el nivel de CO_2 es alto, puede dar origen a la aparición de diversos síntomas, como dolor de cabeza, temblor, sudoración, visión borrosa e incluso pérdida del conocimiento [67]. Actualmente, existen varios tipos de sensores de CO_2 , esencialmente, hay dos clases: sensores de tipo NDIR (*Non-Dispersive InfraRed*) [23] y sensores con utilización de métodos químicos [33]. Ambos suelen utilizarse para monitorizar las concentraciones de CO_2 en la calidad del aire interior. Sin embargo, los sensores con mayor usos comerciales y, por lo tanto, más utilizados para medir la calidad del aire en interiores son los sensores basados en NDIR, debido a que los sensores basados en métodos químicos presentan muchas limitaciones a la hora de su aplicación. Algunas de Las limitaciones de estos sensores son la ausencia de estabilidad a largo plazo y una baja durabilidad, dado que las sustancias que componen este tipo de sensores pueden ser contaminadas fácilmente con los gases y partículas que se encuentren en el entorno.

Los sensores de CO_2 basados en la tecnología NDIR (*Non-Dispersive InfraRed*) proporcionan estabilidad a largo plazo, una tasa de consumo energética constante durante su medición y mayor precisión. Por lo tanto, los sensores de CO_2 basados en NDIR son los más utilizados cuando se requiere una estimación de la concentración de CO_2 en tiempo real. En la Figura 3.2 (a) se muestra uno de los modelos de sensor de CO_2 NDIR más comunes [34].

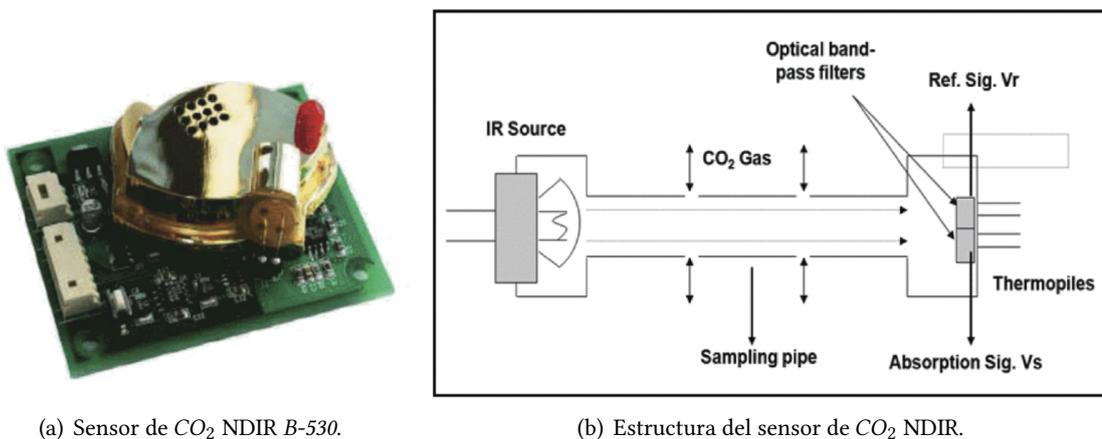


Figura 3.2: Representación y estructura de un sensor de CO_2 basado en NDIR [23].

¹Un dato ruidoso hace referencia a datos que están distorsionados, debido a su baja relación señal-ruido. Este tipo de datos contienen una gran cantidad de información adicional sin sentido, a la que se le denomina ruido [13].

La estructura del sensor de CO_2 NDIR se muestra en la Figura 3.2 (b), consta de cuatro componentes principales, como la lámpara de infrarrojos (la fuente de luz infrarroja, con una frecuencia determinada), la cavidad óptica (por donde se difunde el gas detectado), el filtro de banda óptico y el detector de infrarrojos (IR), en este modelo, se puede apreciar como la lámpara de infrarrojos y el detector están instalados a ambos lados de la cavidad óptica. Este tipo de dispositivos necesitan un tiempo de calentamiento para realizar una pequeña calibración dentro del entorno a monitorizar, dicho proceso tiene una duración de entre uno y tres minutos [34].

Sensor piezoeléctrico para detección de vibraciones en el suelo

Este tipo de sensores están basados en el *efecto piezoeléctrico*², a partir del cual se puede realizar una estimación de la posición de un individuo. El sistema [2] debe de asumir las posiciones en las que se encuentran los objetos que forman parte del entorno, como el mobiliario. Teniendo en cuenta que las posiciones de este tipo de objetos se conoce de antemano, el sistema realiza una estimación del posicionamiento de los individuos en base al tipo de vibración que estos producen en el mobiliario cuando realizan cualquier tipo de acción. De esta forma, el sistema no capta ninguna información relacionada con la privacidad de los individuos.

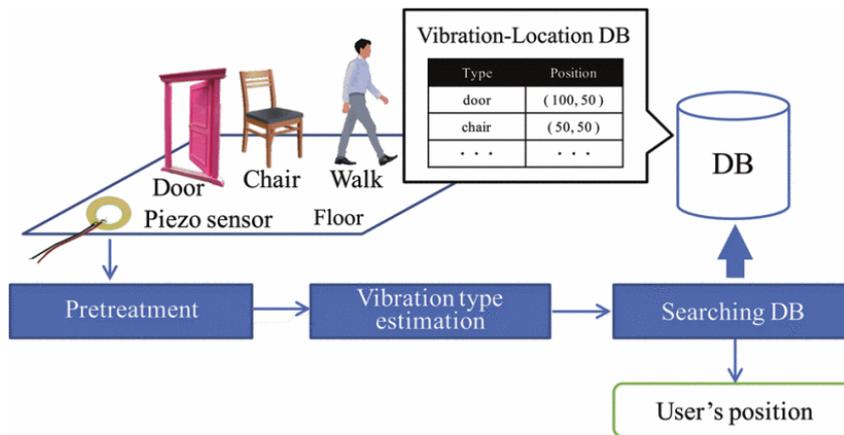


Figura 3.3: Diagrama de funcionamiento de un sistema de estimación de ubicación basado en vibración de suelo [30].

En el diagrama de la Figura 3.3, se muestra el funcionamiento de este tipo de sistemas. En este caso, el sensor piezoeléctrico se encuentra fijado al suelo, encargado de capturar la vibración generada por parte del individuo y el propio mobiliario del entorno. El sistema realiza una estimación de la posición del individuo en tres etapas. En la primera etapa se realiza un tratamiento previo, donde el sistema procesa la vibración capturada, reduciendo el efecto del ruido. En la segunda etapa, el sistema clasifica las señales recibidas por el sensor según el momento o tiempo en la que se han realizado. Finalmente, en la tercera etapa el sistema consulta los datos de posicionamiento en su base de datos. La base de datos se encarga de almacenar la ubicación correspondiente al tipo de vibración estimada anteriormente [2, 30].

3.1.1. Aplicación de cámaras para la detección de ocupación

La mayoría de las soluciones clásicas para obtener una distribución precisa es utilizar el sistema de vigilancia basado en imágenes y vídeos para el análisis del entorno. Analizando los fotogramas

²El *efecto piezoeléctrico*, es un fenómeno que tiene lugar en determinadas superficies, cuando al ser sometidas a tensiones mecánicas, su masa adquiere una polarización eléctrica, dando lugar a la aparición de una diferencia de potencial y cargas eléctricas en su superficie [6].

que forman el vídeo se puede obtener información precisa de la densidad ocupación existente en el entorno [70]. El algoritmo utilizado para el procesamiento de imágenes está compuesto por varias etapas, como la realización de modelado de fondo, detección de cambios, agrupación e interpretación de eventos ocurridos [17]. Sin embargo, las soluciones basadas en vídeo o captación de imágenes requieren una línea de visión directa entre la cámara y los individuos, además pueden verse afectadas por condiciones de iluminación desfavorables. La privacidad de los ocupantes puede suponer un tema crítico en la utilización de cámaras.

En [88], se propone un novedoso sistema con el objetivo de ahorrar en el consumo energético de las oficinas de trabajo. El sistema está basado en un algoritmo de análisis de vídeo para detectar ocupación. La detección humana se realiza en base a la detección de la cabeza del individuo desde el vídeo de vigilancia. Por lo tanto, no es necesaria la utilización de ningún tipo de sensor adicional. La característica fundamental de este algoritmo es la aplicación de una estructura en cascada de tres etapas, la cual, mediante un clasificador combina las diferentes características distintivas de la cabeza humana y la información del movimiento corporal realizado por el usuario. A través de su aplicación en una oficina de trabajo, se concluyó, que el algoritmo propuesto ofrecía una precisión relativamente alta a la hora de detectar la cabeza del usuario, así como una velocidad alta de procesamiento.

3.1.2. Aplicación del sensor o transceptor BLE para la detección de dispositivos inteligentes

El conocimiento con precisión de las personas que se encuentran en el interior de una edificio es de gran relevancia dentro de un amplio abanico de aplicaciones, desde su utilización para ahorrar en el consumo de energético de la instalación [84], la navegación en interiores [22] o la monitorización remoto de atención médica [58], hasta ser clave para responder a una crisis (como puede ser un ataque terrorista o una catástrofe). En este contexto, surge la necesidad de tecnologías de comunicación inalámbrica y es aquí donde el *Bluetooth Low Energy* (BLE) es particularmente prometedor, su aplicabilidad para la localización en interiores ha sido reconocida por diversas instituciones a nivel mundial, siendo una de ellas la Comisión Federal de Comunicaciones de Estados Unidos (FCC). Dicha comisión ha propuesto un informe donde estudian el uso de BLE y WiFi para ayudar a la localización de individuos cuando llaman al número de emergencias (en este caso el 911) desde el interior de edificios u otro tipo de entornos y construcciones cerradas. Todo ello a partir de la comunicación entre los dispositivos móviles de los individuos y una red de seguridad pública, basándose en la ubicación BLE del dispositivo y enviando dicha información a la red mediante vía WiFi [37].

La mayoría de los dispositivos móviles están equipados con un módulo *Bluetooth* de tipo BLE (Bluetooth 4.X o 5.x), que pueden comunicarse con nodos *Beacon*³. Las balizas *Bluetooth* (nodos *Beacon*) emiten un identificador único, que puede ser recogido por una aplicación compatible. La principal ventaja que ofrece el uso de esta tecnología de comunicación, a diferencia de otras, es su bajo consumo de batería. Existen investigaciones basadas en la utilización de la tecnología BLE para la detección de ocupación y la estimación de multitudes en el interior de los edificios, sirva de ejemplo [84], donde se realiza una estimación de la ubicación de los dispositivos móviles que llevan consigo los usuarios, en función de los indicadores de intensidad de señal recibida o RSSI (*Received Signal Strength Indicator*), emitidos por sus transceptores BLE.

En [21] se realiza un estudio sobre las ventajas que ofrece el uso de la tecnología BLE en interiores frente a la tecnología WiFi. Este estudio concluyó que, a partir del posicionamiento marcado por una baliza BLE, se logra una precisión de posicionamiento de menos de 2,6 metros de diferencia durante el 95 % del tiempo. Para ello, se investigan diferentes frecuencias de transmisión de onda, obviando el

³*Beacon* o baliza electrónica es un dispositivo que actúa como transmisor, utilizado para emitir una señal de tipo BLE a dispositivos que se encuentren alrededor suyo, sin la necesidad de una sincronización previa.

efecto que esto podría causar en el consumo de energía de los dispositivos móviles.

3.1.3. Aplicación del sensor o transceptor WiFi para la detección de dispositivos inteligentes

La tecnología WiFi se utiliza como alternativa al GPS⁴ para aplicaciones de localización en entornos interiores, basadas en el posicionamiento o estimación de ocupación en estos espacios. La reciente integración de tecnologías de detección inalámbricas de dispositivos móviles están permitiendo la adaptación de aplicaciones de monitorización más livianas y de bajo coste. Este tipo de tecnologías están basadas en el despliegue de de redes de sensores inalámbricos WSN [86].

Las infraestructuras WiFi basadas en el estándar IEEE 802.11 [73] están ampliamente presentes en los entornos construidos (entornos de interior), además, la mayoría de los dispositivos móviles integran un transceptor WiFi. De esta forma, el movimiento de un ocupante puede variar la propagación de la señal WiFi emitida por el dispositivo móvil que lleva consigo. El reciente desarrollo llevado a cabo en el ámbito de las tarjetas de interfaz de red WiFi (NIC), ha permitido el acceso a una medición de canal de grano fino a partir de la *capa física*, lo que se conoce como información de estado del canal o *Channel State Information* (CSI) [8], El CSI ofrece una descripción de la propagación producida por las señales WiFi desde el dispositivo transmisor Tx (dispositivo móvil de los usuarios) al receptor Rx (dispositivo que forme parte del sistema de monitorización). En base a la información obtenida del CSI, es posible revelar, con el procesamiento adecuado, la actividad humana realizada dentro del entorno monitorizado. Este proceso de captación se realiza de forma no intrusiva de cara al usuario, debido a que los movimientos del cuerpo durante el desarrollo de una actividad física conllevan una serie de interferencias en los caminos de propagación de la señal, obteniéndose como resultado una alta variación del CSI. Sin embargo, para llevar a cabo una estimación lo más precisa posible, es necesario que las señales emitidas por los dispositivos sean capturadas en su totalidad. Estas transmisiones son generadas en tiempo real, lo que plantea un gran desafío a la hora de realizar la recopilación y el procesamiento de los datos [87].

La identificación de actividades físicas con un alto nivel de reconocimiento puede ser factible mediante la realización de un análisis de medición del canal CSI en el dispositivo receptor (Rx). Diversos proyectos han propuesto sistemas de reconocimiento de actividades basados en el CSI [15, 61, 85], pero el marco de aprendizaje automático sigue requiriendo una compleja y engorrosa ingeniería de características, basada en la recolección de patrones relacionados con el dominio de tiempos y frecuencia, seguido de los métodos clásicos de clasificación como SVM⁵ (*Support Vector Machine*) o el árbol de decisión [83].

3.1.4. Aplicación de dispositivos para la captación de señales LTE

Este tipo de sistemas [14] estudia la densidad de ocupación sin la detección por radiofrecuencia (WiFi o Bluetooth) proveniente de los dispositivo inteligente, como *smartphones* o *wearables*. De esta forma, se descarta la necesidad de que los usuarios lleven consigo algún tipo de dispositivo adicional. Estos sistemas están empleando para ello mediciones de la intensidad de la señal (RSSI) de señales LTE. Debido a que las señales de radio son atenuadas y reflejadas por los cuerpos humanos, la presencia de individuos y/o sus acciones por intuitivas y pequeñas que sean, ofrecen como resultado una variación en el canal de propagación entre el emisor (Tx) y el receptor (Rx) de la señal. Mediante la realización

⁴GPS (*Global Positioning System*), son sistemas de posicionamiento mediante radio, estos sistemas están basados en satélites que ofrecen a los usuarios información sobre la posición global, velocidad y tiempo en la que se encuentran desde cualquier punto del planeta y bajo cualquier condición climática [36].

⁵*Support Vector Machine* (SVM), en castellano Máquina de Vectores de Apoyo, es un método muy potente de clasificación utilizado para resolver problemas relacionados con el reconocimiento de patrones [83].

de un análisis de la señal recibida, podría detectarse la presencia o la realización de ciertas actividades por parte de un individuo. La utilización de estos sistemas está ofreciendo resultados prometedores en entornos o espacios de interior, donde la variación del canal de propagación de este tipo de señales depende en gran medida de la atenuación causada por los individuos que se encuentran en el entorno de monitorización. Una muestra de ello, puede verse en el trabajo descrito en [14], donde se lleva a cabo una evaluación sobre un sistema de estimación de densidad de ocupación en entornos de interior. Este es un sistema de tipo *radar*, basado en las variación de las señales de tipo LTE, concretamente en el análisis del canal de información de estado o CSI (*Channel State Information*), calculado a partir de señales emitidas entre el eNodeB y un receptor de señal LTE. En la Figura 3.4 se muestra una representación gráfica de lo descrito anteriormente. Cuando un cuerpo se interpone entre la transmisión realizada a través de este tipo de dispositivos, provoca una dispersión de señal, provocando así una serie de alteraciones de las que, bajo análisis, puede obtenerse información sobre la densidad de ocupación en el entorno.



Figura 3.4: Representación de la estimación de densidad de ocupación en base al uso de señales inalámbricas LTE [14].

3.2. SISTEMAS ORIENTADOS A LA MONITORIZACIÓN DE LA DENSIDAD DE OCUPACIÓN EN ENTORNOS DE INTERIOR

Si se centra el foco de estudio en este tipo de sistemas se comprueba que para realizar una estimación de la densidad de ocupación en un espacio cerrado determinado, generalmente, se requiere desplegar una red de sensores que permita detectar presencia, para así poder realizar dicha estimación. Este tipo de sistemas, como ya se ha remarcado con anterioridad, son clave a la hora de reducir el consumo de energía y mantener una calidad de ventilación adecuada del aire en viviendas o edificios públicos, haciendo uso así de un control climático más eficiente. Además, la estimación de ocupación en interiores puede ser de gran importancia para la supervisión de la seguridad, evacuación de emergencia e incluso para las operaciones de rescate [65].

Existen dos tipos de sistemas enfocados a la detección de ocupación, aquellos basados en el despliegue de sensores físicos para la detección de ocupación [76]; y los sistemas basados en la captación de señales provenientes de dispositivos inteligentes para la estimación de ocupación. Estos últimos emplean, principalmente, etiquetas de radiofrecuencia o señales activas. El sistema tiene la necesidad de obtener la información sobre la fuente de transmisión de la señal, o la agregación de etiquetas identificativas (RFID) a los usuarios objetivos [44]. A partir de estos dispositivos, se lleva a cabo la adquisición de datos para así poder realizar acciones de estimación de ocupación o de posicionamiento dentro del entorno. A continuación, se presentarán diversos proyectos basados en

estas modalidades de detección y/o estimación de ocupación en el interior de edificios.

Sistemas basados en el despliegue de sensores físicos para la detección de ocupación

En [81] se lleva a cabo el desarrollo de un proyecto basado en este tipo de detectores de movimiento. La red de sensores estaba formada por un nodo principal que actuaba como coordinador, y por tres nodos sensores, en este caso sensores PIR (modelo *Murata IRA E710*). El objetivo de este sistema era reducir el campo de visión o FOV de cada uno de los sensores, reduciendo la cantidad de lentes utilizadas por el sensor. Para conseguir este objetivo, se protegieron las lentes innecesarias del sensor (podrían ocasionar falsos positivos), dejando al descubierto únicamente las tres lentes centrales. En la Figura 3.5 se muestra la configuración de los tres sensores que formaban parte del sistema, éstos estaban colocados en una fila horizontal, donde cada uno de ellos tenía una orientación distinta.

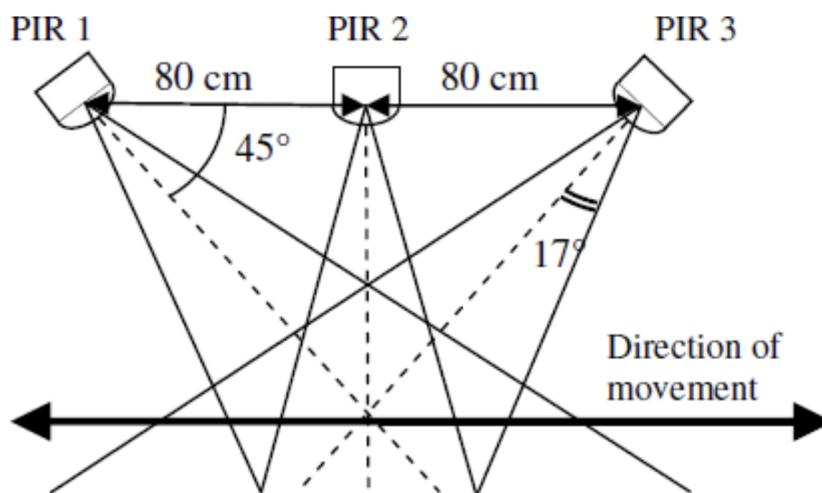
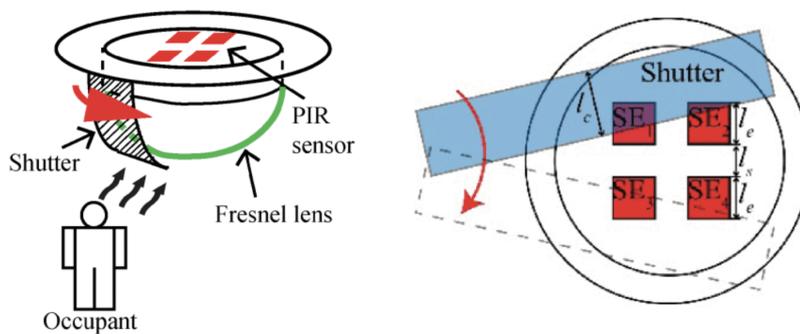


Figura 3.5: Orientación de los sensores PIR [81].

A lo largo del tiempo, se han estudiado diversas formas de mejorar estas soluciones clásicas, como es el caso de la presentada en el proyecto anterior [81]. Las investigaciones presentadas a continuación, ofrecen una serie de soluciones novedosas para mejorar la funcionalidad de los sensores PIR convencionales, mediante la integración del componente obturador, permitiéndose añadir nuevas características a este tipo de sensores y, por lo tanto, ampliar el abanico de funcionalidades que éstos ofrecen.



(a) Diagrama esquemático del prototipo. (b) Diagrama basado en la vista superior del prototipo.

Figura 3.6: Diagramas de la variante del sensor PIR, C-PIR [74, 75].

En [74, 75] se presenta *C-PIR*, una variante del sensor PIR que posee un obturador giratorio⁶ para así explorar más allá de las funcionalidades de detección de presencia. De este modo, se pretende realizar una estimación de la localización y seguimiento de la actividad de los usuarios, debido a que con la utilización del obturador el rango de detección se ve fuertemente incrementado. En la Figura 3.6 se muestra una representación del prototipo de este sensor *C-PIR*, donde se puede observar, de forma detallada, el funcionamiento del obturador.

SLEEP-PIR [76], es un detector de presencia de bajo coste y consumo energético que utiliza un obturador electrónico de cristal líquido, el cual, corta las señales recibidas por el sensor PIR integrado, con el objetivo de reducir la intensidad de la radiación IR recibida por parte de cualquier cuerpo. De esta manera, el prototipo *SLEEP-PIR* se amplía la función básica de los sensores PIR tradicionales hacia la detección de ocupantes estacionarios. En la Figura 3.7 se puede ver este dispositivo; a diferencia de un sensor PIR tradicional, se observa como posee un peso y tamaño mayor.

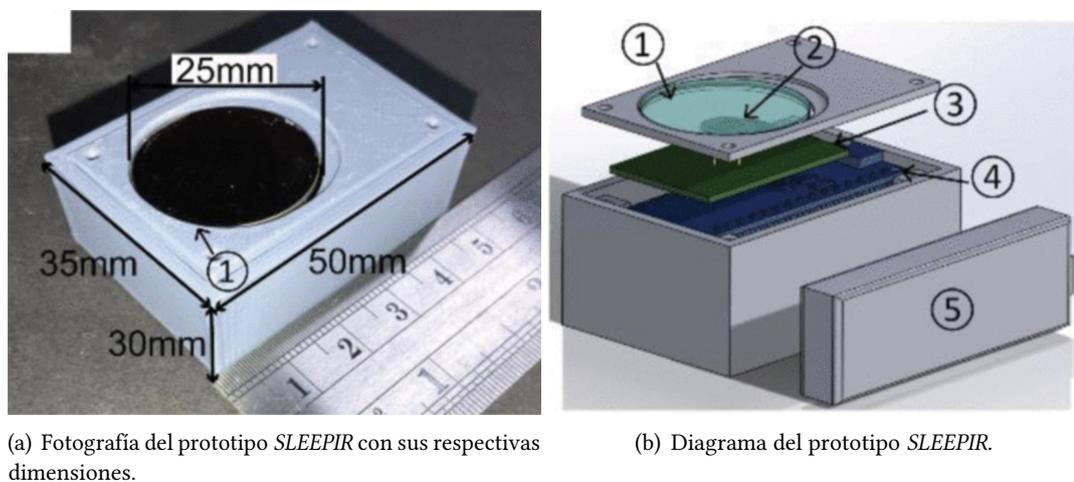


Figura 3.7: Prototipo *SLEEP-PIR* y sus respectivos componentes; (1) Obturador LC, (2) Sensor PIR, (3) Placa del circuito, (4) Microcontrolador, (5) Batería [76].

Por otra parte, se han realizado diversas investigaciones en las que se utiliza una agrupación de diversos tipos de sensores, como sensores PIR, sensores ambientales, e incluso la integración de sensores PIR junto con cámaras. Un claro ejemplo puede observarse en [51] donde se lleva a cabo el diseño de un algoritmo centrado en la fusión de sensores térmicos basados en la tecnología PIR y cámaras, para así garantizar que la detección de ocupación sea realizada con precisión. Específicamente, se utilizan dos algoritmos, el primero basado en el procesamiento de imágenes obtenidas a partir de las cámaras, y así detectar tanto eventos estáticos como dinámicos mediante la sustracción de fondo y modelos de segregación de imagen basados en *ROI*⁷. El segundo algoritmo está enfocado al procesamiento de la señal recibida por parte del sensor PIR. En la Figura 3.8 se puede observar como por cada fila de mesas hay instalados nodos con sensores PIR integrados, estos sensores son utilizados básicamente para detectar movimiento en el entorno, de esta forma cuando uno de estos nodos sensores detectan movimiento activan las cámaras y entra en funcionamiento la parte dedica al procesamiento de imágenes.

⁶Obturador giratorio utilizado principalmente en cámaras de cine, es un disco semicircular que gira delante de la puerta de la película, permitiendo alternativamente que la luz de la lente incida sobre la película o bloqueándola [75].

⁷<https://la.mathworks.com/help/images/roi-based-processing.html>.

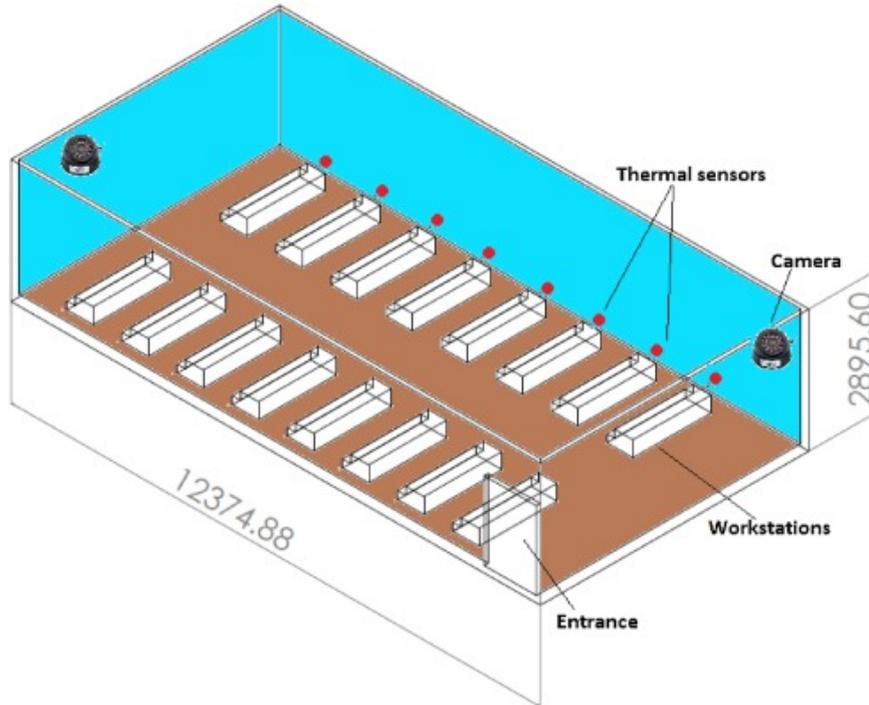


Figura 3.8: Representación del sistema propuesto en [51].

El estudio realizado en [69], presenta una nueva técnica de detección de ocupación basada en la utilización de sensores ambientales, concretamente sensores de temperatura. Se utiliza una matriz de sensores de temperatura, cada sensor está colocado en una silla de oficina, con el fin de estimar la ocupación en la oficina. El diferencial de temperatura entre el ocupante y la superficie del asiento permite de forma rápida y eficaz detectar la presencia de un usuario sentado. En la Figura 3.9 se muestra una representación del prototipo construido, junto con la representación de la oficina en formato de matriz.

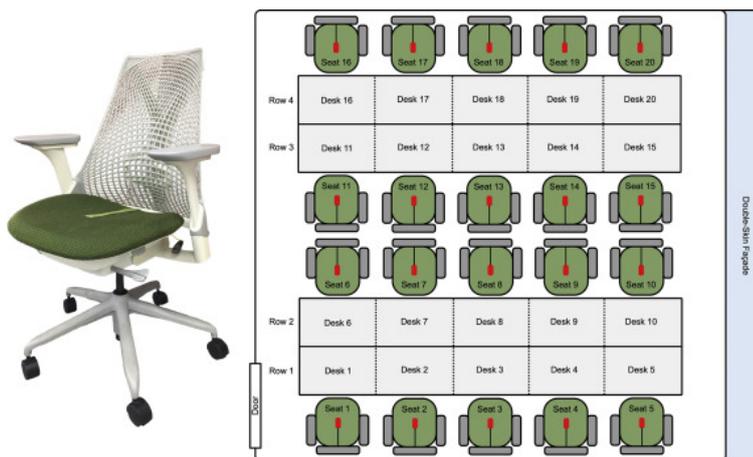


Figura 3.9: Prototipo de silla con sensor de temperatura y representación del escenario de prueba [69].

BloT [26] es un caso de demostración de la eficacia y el potencial de un método basado en la comprobación periódica de la calidad del aire IAQ (*Indoor Air Quality*) y la densidad de ocupación orientado a edificios de pequeño y mediano tamaño. En base a estos parámetros se lleva a cabo un control inteligente de la ventilación, aire acondicionado e iluminación del edificio. Además, el estudio intentó lograr el máximo nivel de practicidad en su implementación y en el coste, a través de la integración de un sistema simple y conciso, con el uso de una red de tipo WSN formada en su mayoría por sensor PIR y sensores de CO₂.



Figura 3.10: Representación del sistema *BIoT* propuesto en [26].

En la Figura 3.10 se muestra los prototipos utilizados en este proyecto, junto con los escenarios donde fueron instalados. Como se puede apreciar, en cada cubículo perteneciente al puesto de trabajo de un empleado, hay instalado un nodo integrado por sensores PIR, para la detección de movimiento y un sensor CO_2 para la medición de la calidad de aire. También se puede observar la representación de un nodo principal encargado de coordinar a los nodos anteriores, denominados *esclavos*.

Sistemas basados en la captación de señales provenientes de dispositivos inteligentes para la estimación de ocupación

La reciente integración de tecnologías de detección inalámbricas de dispositivos móviles están permitiendo la adaptación de aplicaciones de monitorización más livianas y de bajo coste. Este tipo de tecnologías están basadas en el despliegue de una red de sensores inalámbricos WSN (*Wireless Sensor Network*), las cuales, mediante la recopilación pasiva de señales que transmiten los distintos dispositivos inteligentes se puede realizar una estimación de la densidad de ocupación dentro de un área. Esto depende de las tecnologías de comunicación para las que los dispositivos sean compatibles, como es el caso de las tecnologías de comunicación inalámbricas *WiFi* basadas en el estándar IEEE 802.11 [73] y *Bluetooth Low Energy* (BLE) [66]. Si bien, para llevar a cabo una estimación lo más precisa posible, una opción es que todas aquellas señales de comunicación emitidas por los dispositivos (capa física) sean capturadas y analizadas en combinación. La generación de estas solicitudes en tiempo real plantea un gran desafío a la hora de realizar la recopilación y el procesamiento de los datos [35].

En [52] se presentan diversos estudios relacionados con el comportamiento que presentan las personas en los edificios públicos. En particular, se realizaron las evaluaciones pertinentes en *Hospital Universitario de Aarhus*. Se utilizaron las señales *WiFi* obtenidas de sus propios dispositivos. Siguiendo varios métodos de análisis, se elaboró una estimación (en base a la ubicación de dichos dispositivos) de las rutas realizadas por los usuarios durante su estancia en el edificio, marcando su

entrada y su respectiva salida. Además, el sistema otorgaba roles a cada usuario, diferenciados entre personal y visitante del hospital, con el fin de obtener información relevante relacionada sobre el comportamiento de cada grupo dentro del edificio. En la Figura 3.11, se muestran varias de las rutas detectadas seguidas por los individuos en el interior del hospital. Se puede observar como estas rutas se encuentran dentro del área perimetral del hospital.



Figura 3.11: Área perimetral del conjunto de edificios que forman el hospital, junto las rutas generadas [52].

UIM (University of Illinois Movement) [68] es un sistema de escaneo compatible con los dispositivos móviles Android de Google. El sistema está basado en la recopilación de la información relacionada con la **MAC** de los puntos de acceso (AP) WiFi a los que el dispositivo se conecta; y en las búsquedas realizadas automáticamente de forma pasiva por parte del módulo BLE (*Bluetooth Low Energy*) que lleva integrado el dispositivo. De esta forma, el sistema almacena las ubicaciones del usuario en base a la MAC única de los puntos de acceso WiFi a los que se conecta el dispositivo y los contactos sociales cercanos que mantiene el usuario, a partir de las MAC BLE obtenidas como resultados en las búsquedas automáticas del dispositivo. Las pruebas se realizaron en el campus universitario de la *Universidad de Illinois*⁸, obteniendo resultados prometedores como los sitios con un mayor índice de popularidad, es decir, en base a las MAC WiFi de los puntos de acceso a los que se conectaban los dispositivos de los usuarios, de esta forma el sistema estimaba las MAC más frecuentes y creaba así un ranking de los sitios del campus más populares o visitados por los usuarios. Además, también se analizaba los contactos sociales que mantenían los usuarios en base a la MAC BLE de los dispositivos que se obtenían como resultado en cada búsqueda automática, es decir, desde un dispositivo se lleva a cabo una búsqueda de dispositivos cercanos que mantengan *Bluetooth* activado. Los dispositivos encontrados se catalogan por parte del sistema como contacto social, cuanto más repeticiones tenga la MAC de un dispositivo éste se catalogará como frecuente y, por lo tanto, el dueño del dispositivo como un contacto social cercano al propio usuario. En estas pruebas se mantiene la privacidad de los usuarios, dado que en la estimación de los contactos sociales solo se almacena por parte del sistema la MAC BLE de los dispositivos. El estudio concluyó que la mayoría de los usuarios (estudiantes y profesorado) durante su vida en el campus suelen visitar lugares habituales y hacer contactos frecuentes.

Con la integración de las redes WSN (*Wireless Sensor Networks*) [86] en los ámbitos relacionados con la detección de presencia u ocupación en el interior de edificios, se ampliaron aún más las posibilidades de aplicación para este tipo de sistemas basados en tecnología IoT. En el artículo [35], se muestra un sistema formado por una red de nodos de tipo WSN, donde cada nodo está conectado de

⁸<https://www.uic.edu>

forma dual a dos interfaces. La primera interfaz representa la conexión a un Dongle USB Wi-Fi para la recopilación de señales recibidas vía WiFi. La segunda interfaz está conectada a la red WiFi existente, para así poder transmitir los datos recopilados anteriormente a un servidor desplegado en la misma red. Con este tipo de arquitectura se logra realizar la recopilación de las señales WiFi y la transmisión de los datos obtenidos de forma simultánea. Este sistema fue utilizado para comprender la densidad de ocupación en un área pública, como se muestra en la Figura 3.12. Se realizaron evaluaciones en dos escenarios; una sala de laboratorio de investigación para observar las horas de oficina de los trabajadores; y en cuatro salas ubicadas en el campus de la *SUTD University*⁹ para así estudiar la densidad de ocupación de estas salas a distintas horas del día.

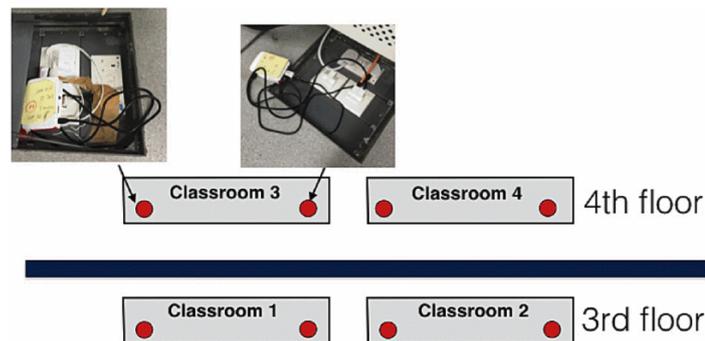


Figura 3.12: Representación del nodo del sistema, y su disposición por salas [35]

Hoy en día los dispositivos inteligentes como los *smartphones* [10] llevan integrado la función de ahorro de batería [29]. Esta función desconecta periódicamente la comunicación WiFi activando el modo reposo, provocando una serie de irregularidades a la hora de realizar la detección de las señales emitidas por estos dispositivos, suponiendo todo un reto para el desarrollo de sistemas basados en la detección de señales WiFi. Sin embargo, la mayoría de proyectos centrados en la aplicación de esta tecnología asumen una captación continua de las señales WiFi. En el trabajo [71] se lleva a cabo una investigación sobre la detección de densidad de ocupación basada en el uso de la tecnología WiFi en un entorno interior complejo y con gran ocupación. Se propone un método de actualización activado por eventos, para así poder hacer frente a la discontinuidad en la comunicación WiFi de los dispositivos móviles. En la primera parte del artículo se estudia el desarrollo de un filtro para la identificación basado en la captación de direcciones MAC de los dispositivos, seleccionando así los dispositivos válidos para el sistema y descartando los irrelevantes. Los resultados obtenidos mostraron que el filtro de identificación propuesto es capaz de mejorar la precisión de detección entre un 77,3 % y un 96,8 %. Demostrando así ser bastante eficaz para eliminar dispositivos externos o dispositivos *estáticos no humanos*, como impresoras, computadores de escritorio, *smartTVs*, etc.

	WiFi activado		WiFi desactivado
	Modo normal conectado a red WiFi	Modo de escaneo, no conectado a una red WiFi	
Pantalla encendida	Transmisión de señal a alta frecuencia (alrededor de 5 segundos).	Transmisión de señal a alta frecuencia (alrededor de 5 segundos).	Sin transmisión de señal.
Pantalla apagada	Transmisión de señal a baja frecuencia (alrededor de 10 segundos).		

Tabla 3.1: Modos de funcionamiento del módulo WiFi integrado en los dispositivos inteligentes [71].

La segunda parte del artículo [71], está centrada en el estudio de los distintos modos de funcionamiento de los dispositivos inteligentes. En la Tabla 3.1, se muestra un resumen de los modos de

⁹<https://www.sutd.edu.sg>

funcionamiento de los dispositivos móviles inteligentes que suelen llevar consigo las personas [10]. Esta tabla se encuentra basada en la representación de la Figura 3.13, donde se puede apreciar el modo de funcionamiento de los dispositivos inteligentes a la hora de realizarse un desplazamiento desde un área a otro. Cuando una persona entra en uno de los áreas, el dispositivo activa el modo de escaneo (alta frecuencia de transmisión de la señal), estableciendo así una conexión al punto de acceso (AP) WiFi ubicado en el área. Una vez ya realizada la conexión el dispositivo cambia a modo ahorro de energía (pantalla apagada), emitiendo una señal de transmisión a baja frecuencia. Este tipo de señal dificulta la realización de una detección precisa. En cambio, cuando el ocupante se encuentra en el área de escaneo situado en el exterior de ambos áreas, vuelve a activarse el modo de escaneo a alta frecuencia, lo que permite llevar a cabo una detección continuada de la señal WiFi transmitida por el dispositivo. Obteniendo como conclusión, la transición de un ocupante entre habitaciones o zonas puede detectarse siempre de forma continuada, mientras que la realización de una actividad estacionaria en el interior de un área no siempre es detectable.

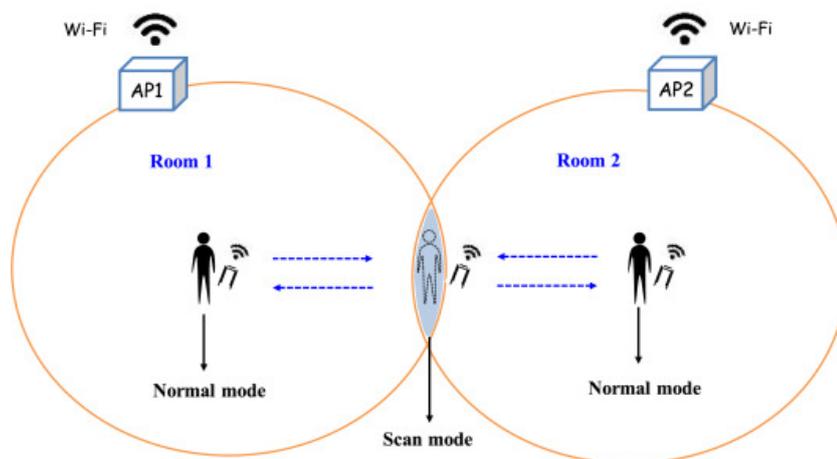


Figura 3.13: Representación basada en los modos de funcionamiento WiFi de los dispositivos inteligentes; Círculo de color naranja como representación del área o zona de servicio de un punto de acceso (AP) [71].

Para finalizar este apartado, se presenta una investigación [37] realizada a petición de la CCF (Comisión Federal de Comunicaciones de Estados Unidos), mencionada en la sección 3.1.2. La CCF ha estimado que existe la posibilidad de salvar alrededor de 10.000 vidas por año, si se dispusiese de dispositivos que proporcionasen la ubicación de las personas que realizan llamadas de emergencia desde el interior de edificios. La investigación describe y analiza el funcionamiento de BOSSA (Bluetooth and Sensors Array) un sistema basado en la captación de señales de tipo BLE emitidas por el dispositivo para identificar la ubicación de una persona que llama mediante una aplicación de teléfono inteligente para solicitar asistencia de emergencia. BOSSA utiliza señales BLE emitidas por una serie de nodos con tecnología *iBeacons* [60] desplegados por la infraestructura de un edificio. Los resultados obtenidos en base a la ubicación del ocupante, proporcionan una identificación del 100 % del nivel de planta del edificio, junto a un fallo de precisión máximo de 5 metros. Por lo tanto, la capacidad de diferenciación de las diferentes plantas que constituyen el edificio por parte del sistema y su alto nivel de precisión, lo posicionan como una solución razonable al problema de la localización de ocupantes en infraestructuras complejas, como edificios formados por varias plantas.

En la parte superior de la Figura 3.14 se muestra la arquitectura del sistema BOSSA, cómo el sistema detecta el dispositivo inteligente que lleva consigo el ocupante, y cómo, por parte de los servidores de localización, se realiza una triangulación de su posición. Posteriormente, a través de la aplicación instalada en el propio dispositivo del ocupante, se emiten sus coordenadas a través de Internet (con el uso de la tecnología WiFi o LTE) a la red de los servicios de emergencia, en este caso conocida como *ESINet*.

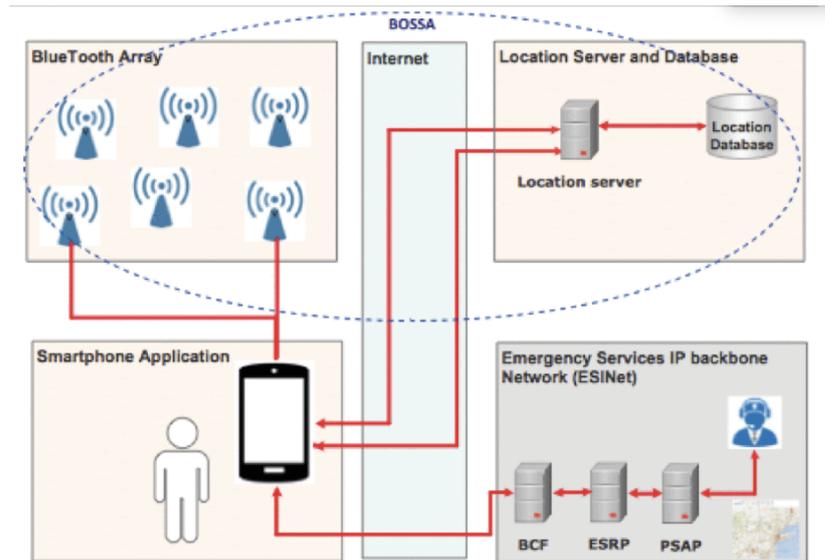


Figura 3.14: Diagrama de funcionamiento de la plataforma BOSSA [37].

3.3. ESTUDIO Y ELECCIÓN DEL ENTORNO *HARDWARE*

A lo largo de esta sección se va a llevar a cabo un estudio para elegir el entorno *hardware* que mejor se adapte al proyecto, teniendo en cuenta los proyectos e investigaciones anteriormente mencionados. Hasta hace poco tiempo, existía un inconveniente principal al realizar este tipo de proyectos sobre detección de ocupación en entornos de interior, debido al alto coste que suponía el *hardware* requerido y su instalación, principalmente se utilizaban cámaras de videovigilancia. Sin embargo, este problema se ha ido solucionando con el uso de redes de sensores de bajo coste (como es el caso de los sensores PIR), y su posterior actualización al despliegue de redes de sensores inalámbricos (WSN), mencionados en la sección anterior.

En el proyecto a desarrollar es necesario la detección de actividad y de trasiego producido por los ocupantes en los entornos de interior, para ello se requiere la utilización de diversos tipos de sensores, abarcando desde los sensores físicos como sensores PIR y sensores ambientales, hasta sensores basados en la captación de señales provenientes de dispositivos inteligentes (transceptores inalámbricos). Para que estos sensores ofrezcan una comunicación inalámbrica, tienen la necesidad de estar integrados o conectados a algún módulo transceptor que los dote de ese tipo de conectividad. Es aquí donde surge la necesidad de usar SoC que ofrezcan las funcionalidades demandas por el sistema, el cual, está basado en el despliegue de una red de *nodos sensores* de tipo WSN.

El mercado actual ofrece una gran cantidad de microcontroladores y de sistemas embebidos de bajo coste, por lo tanto, una característica principal del proyecto es la elección de la solución más apropiada y que se adapte de forma correcta a los requisitos del sistema. Se ha realizado un estudio comparativo de empresas con gran relevancia en el mercado de los microcontroladores y SOC's. como son *Arduino*¹⁰ y *Espressif*¹¹, para así comprobar qué dispositivo de los ofrecidos por ambas compañías se adapta de forma correcta al sistema. Además, surgen una serie de necesidades a tener en cuenta a la hora de su aplicación, como son: el tamaño del sistema embebido o de la placa de desarrollo; la posibilidad de realizar una conexión de red vía WiFi; y el precio por unidad, debido a que puede ser una característica interesante para replicar una gran cantidad de nodos que formen el sistema, y así hacer que el sistema posea el máximo nivel de escalabilidad posible.

La compañía *Arduino 10* está orientada al desarrollo de *hardware* y *software* libre, conocido

¹⁰ <https://www.arduino.cc/en/Guide/Introduction>

¹¹ <https://www.espressif.com/en/products/modules>

como *Open Hardware* y *Open Source*. Su primera placa de desarrollo y más popular hasta la fecha es *Arduino Uno*¹². Como se muestra en la Figura 3.15 a), la placa está equipada con un microcontrolador *ATmega328* de 8 bits, cuyas características son una frecuencia de 16MHz, un juego de instrucciones RISC, 32 KBytes de memoria FLASH, 1024 Byte de memoria EEPROM y 2 KBytes de memoria SRAM. Además, esta placa también ofrece una amplia variedad de pines tanto analógicos como digitales. Otra placa de desarrollo ofrecida por la compañía es *Arduino Nano*¹³, mostrada en la Figura 3.15 b), ofrece características técnicas similares a las del *Arduino Uno*. En cuanto al diseño, la característica principal es su reducido tamaño, siendo esta fundamental para el proyecto, ya que no es necesario la utilización de una gran cantidad de pines analógicos y digitales. Además, uno de los tipos de nodos sensores deberá ir instalado en el marco de la puerta, por lo tanto, se busca que el prototipo a desarrollar posea un tamaño y peso lo más reducido posible, para así resultar menos costosa su instalación. Sin embargo, ambas opciones fueron descartadas, debido al bajo nivel de conectividad que ofrecían estos sistemas por sí solos, al carecer de la integración de un módulo WiFi o Bluetooth. Un alto nivel de conectividad es clave a la hora de desarrollar un sistema basado en el despliegue de una red WSN.

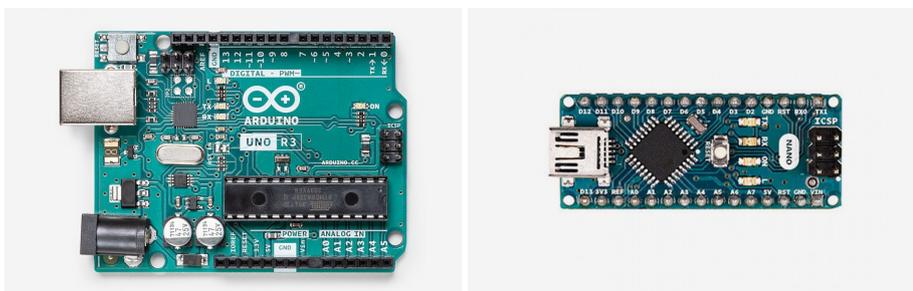
(a) Placa de desarrollo *Arduino Uno* 12.(b) Placa de desarrollo *Arduino Nano* 13.

Figura 3.15: Representación de componentes y placas de desarrollo de la compañía *Arduino*.

Las placas de desarrollo de la compañía *Arduino* se pueden programar en un lenguaje sencillo y potente, basado *C* y *C++*, existiendo una gran variedad de plantillas, denominadas *sketches*¹⁴ realizadas por su comunidad, con el fin de servir como ejemplo y facilitar la implementación de programas a los usuarios más principiantes. También ofrecen una amplia variedad de *firmwares* compatibles con otros sistemas embebidos. Esta posibilidad despertó el interés investigador, centrando dicho interés en la compañía *Espressif*.

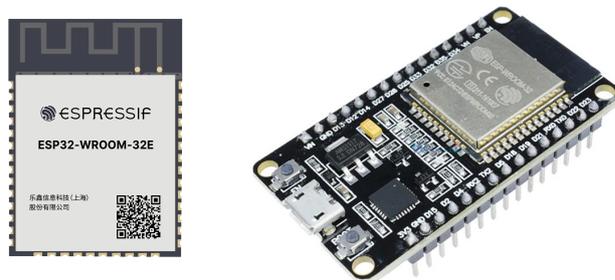
(a) Microcontrolador *ESP32-WROOM-32*.(b) Placa de desarrollo *ESP32DevKitC*.

Figura 3.16: Representación de microcontrolador y placa de desarrollo para *ESP32-WROOM-32* 11.

¹²<https://store.arduino.cc/arduino-uno-rev3?queryID=undefined>

¹³<https://store.arduino.cc/arduino-nano?queryID=undefined>

¹⁴<https://www.arduino.cc/en/Tutorial/Sketch>

Espressif 11 es una compañía que ofrece soluciones de bajo coste para el ámbito *IoT* que incorporan conectividad WiFi y/o *Bluetooth*. Esta compañía es conocida por haber desarrollado un SoC compatible con la tecnología WiFi de muy bajo coste, el conocido *ESP8266*. En base al SoC *ESP8266*, surgió un nuevo SoC mucha más potente, denominado *ESP32*. El *ESP32* es compatible tanto con la tecnología WiFi, como BLE, en mucha de sus variantes, siendo esta última una de las ventajas respecto a su antecesor. El *ESP32* ofrece diversas ventajas además de esta que se pueden observar en la Tabla 3.2. Es por ello, que finalmente se decidió utilizar esta placa, en concreto el *ESP32-WROOM-32*, el cual, se ajusta perfectamente a los requisitos que demandaba el proyecto, con la integración de un módulo WiFi y BLE en el mismo. A través del uso de los módulos transceptores de WiFi y BLE, se realizará la captación de señales de cada tipo que emitan los dispositivos inteligentes que lleven consigo los ocupantes. Y tras esto, vía WiFi se enviarán los datos recopilados al servidor correspondiente encargado de su tratamiento. En la Figura 3.16 se muestra el SoC *ESP32-WROOM-32* junto a la placa de desarrollo *ESP32DevKitC*, en la cual va integrado el mismo.

CARACTERÍSTICAS	ESP8266	ESP32-WROOM-32
Arquitectura	Harvard	Harvard
Procesador	Tensilica LX1060 de 32 bit Single Core a 80MHz	Tensilica XTensa LX6 de 32 bit Dual Core a 160 MHz
Memoria SRAM	64 KB juego de instrucciones y 96 KB datos	520 KB
Memoria Flash	4 MB	16 MB
Memoria ROM	Incompatible	448 KB
Coprocesador de bajo consumo	Incompatible	Entorno a 150 uA
UART	2, una destinada al pin Tx	3
SPI	2	3
I2C	1	2
GPIO Utilizables	11	32
GPIO Analógico	1	18
PWM	8	16
ADC	1 de 10 bits	3 de 12 bits
WiFi	IEE 802.11 (b/g) hasta 20 dBm	IEE 802.11 (b/g) hasta 20 dBm
Bluetooth	Incompatible	Versión 4.2 BLE

Tabla 3.2: Comparación técnica de los microcontroladores *ESP8266* y *ESP32-WROOM-32* 11.

Tras el estudio de los proyectos [74-76, 81], se ha optado por elegir el sensor basado en la tecnología PIR (*Passive Infrared*) para la detección de presencia en el entorno, debido a su bajo coste, y su flexibilidad para integrarse en cualquier tipo de sistema. Sin embargo, se ha optado por descartar la serie de reajustes que se realizan en los anteriores proyectos. En particular, en [75] exponen una solución novedosa basada en un obturador giratorio, cuyo objetivo es ampliar el rango de longitud que ofrece por defecto el sensor PIR, pero esta solución realmente no es eficaz para nuestro propósito. La finalidad del sensor PIR en el ámbito del presente proyecto esta orientada a la detección de ocupantes que entran al interior del espacio monitorizado, instalando así este sensor en el marco superior de la puerta por la que se accede a dicho espacio, siendo más que suficiente el rango de longitud que ofrecen por defecto este tipo de sensores, el cual, se encuentra entre los tres y diez metros de distancia. Por otra parte, en [76] se presenta el sensor *SLEEPPIR* una variante de sensor PIR basada en la integración de un obturador electrónico que amplía su funcionalidad inicial, otorgándole a este sensor la capacidad de detectar actividades estacionarias como hemos mencionado anteriormente, esta es una funcionalidad incompatible con los detectores PIR convencionales. En cambio, este funcionamiento sería ineficaz para el enfoque del presente proyecto, ya que simplemente la finalidad que se busca en el sensor PIR es la detección de ocupantes en movimiento en un espacio muy limitado, debido a la zona en la que se encontrará instalado.

Las investigaciones basadas en la utilización de sensores ambientales [26, 69] demuestran la gran utilidad de este tipo de sensores, tanto para la detección de ocupación, como a la hora de controlar la calidad de aire en un entorno determinado. Es por ello, que se ha obstado por la elección de un sensor de CO_2 basado en la tecnología NDIR [26], este tipo de sensor ofrece la funcionalidad de detectar una posible variación de ocupación en el entorno, mediante la evaluación de la concentración de partículas de CO_2 en el mismo, es por ello, que el uso correcto de este sensor es esencial para el proyecto. El presente proyecto no solo está orientado a la estimación de ocupación dentro de un entorno determinado, si no que además, está centrado en controlar la calidad de aire de dicho entorno.

En lo relativo a los proyectos e investigaciones centrados en la captación de señales provenientes de dispositivos inteligentes mostrados en la sección 3.2, como UIM [68] un sistema basado en el uso de señales BLE para la obtención de los contactos sociales que mantenía cada usuario, y en el uso de señales WiFi para obtener las ubicaciones más populares (más visitadas) dentro de un campus universitario. El estudio sobre UIM ha servido de gran ayuda para entender el uso de las tecnologías inalámbricas. Sin embargo, este enfoque ha sido descartado, debido a que la forma de utilizar estas tecnologías no es apta para el presente proyecto, ya que el enfoque del mismo está centrado en la detección de ocupación en entornos de interior, en base a las señales provenientes de los dispositivos inteligentes que lleven consigo los ocupantes.

Por otro lado, en [71] el sistema hace uso de un filtro de direcciones MAC catalogadas como *dispositivos estáticos*. Esta idea ha sido utilizada en el presente proyecto, evitando así que el sistema capte un gran número de dispositivos WiFi, cuando realmente pueda existir una diferencia notable respecto al número real de ocupantes.

Cabe destacar que no se han encontrado soluciones relacionadas de forma específica con los métodos técnicos de escaneo realizados en el siguiente trabajo. El sistema realiza un escaneo automático y periódico de señales WiFi y BLE existentes en el entorno, a través de los módulos inalámbricos integrados en el SoC *ESP32-WROOM-32* (Figura 3.16 b)), sin la necesidad de integrar ningún driver o dispositivo adicional para llevar a cabo esta labor, como los utilizados en el proyecto [35].

METODOLOGÍA

La utilización de una metodología es clave para el correcto desarrollo de un proyecto tecnológico, debido a que permite trabajar de forma eficiente y completa, para así lograr los objetivos previamente establecidos en la fase inicial de éste. En este capítulo se va a presentar la metodología que se ha empleado durante el desarrollo del proyecto, distribución del trabajo y los recursos que han sido necesarios para poder llevarlo a cabo.

4.1. METODOLOGÍA DE TRABAJO

La metodología elegida para llevar a cabo este proyecto es el *desarrollo iterativo e incremental*, también denominado *iterativo y creciente* [78]. Este desarrollo es un tipo de metodología ágil, donde el trabajo se divide en una secuencia de bloques temporales denominados *iteraciones*, presentes durante todo el ciclo de vida del desarrollo. Cada iteración está compuesta de las siguientes fases: *análisis*, *diseño*, *implementación*, *pruebas* y *evaluación*, como se muestra en la Figura 4.1. Al comienzo del proyecto, se realiza un planteamiento inicial para identificar los requisitos funcionales y no funcionales que se deben de satisfacer, como del alcance que debe tener el proyecto. Tras la finalización de todas las iteraciones que forman el proyecto, se llevará a cabo el despliegue del mismo en el entorno de ejecución elegido.

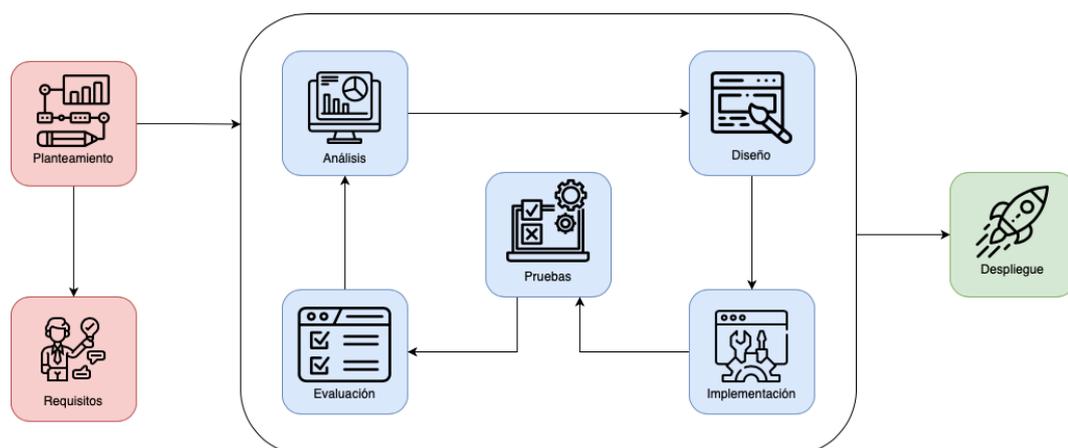


Figura 4.1: Representación del desarrollo iterativo e incremental.

Cada iteración puede tener una duración mínima de una semana y una duración máxima de cuatro semanas, obteniéndose como resultado de cada una de ellas una versión funcional del proyecto, permitiendo así que el proyecto evolucione de forma incremental hasta llegar a la versión final. Al finalizar una iteración se analizan los resultados obtenidos y se comprueba si se han alcanzado los objetivos definidos previamente al inicio de la iteración.

Para la elaboración de este proyecto se ha optado por un desarrollo iterativo e incremental debido a que se adapta a los objetivos y la estructura de este proyecto. De esta forma, se podría dividir el proyecto en dos partes claramente diferenciadas: la aplicación de monitorización, encargada de obtener los datos recopilados por cada uno de los nodos que se encuentran desplegados en el entorno a monitorizar; y la aplicación encargada de la elaboración automática de planes de desinfección. Gracias a esta división, en las primeras iteraciones se ha obtenido una primera versión funcional de cada una de ellas. En las siguientes iteraciones, se ha dotado de completa funcionalidad e integración a ambas partes. Previamente al comienzo de iteración se realiza una planificación de las tareas que se deben de llevar a cabo durante la misma, además de la corrección de los errores encontrados en las revisiones de la iteraciones posteriores.

4.2. DISTRIBUCIÓN DEL TRABAJO

En este capítulo se va a llevar a cabo una explicación de las iteraciones que componen el desarrollo de este proyecto, siguiendo la metodología presentada en la anterior sección. La carga de trabajo de este proyecto se encuentra dividida en cinco paquetes de trabajo distintos, donde cada uno de ellos está compuesto por diferentes tareas.

1. Estudio del estado del conocimiento.
2. Despliegue de nodos para la sensorización del entorno.
3. Aplicación de monitorización del tránsito humano.
4. Aplicación de distribución óptima de los recursos de limpieza.
5. *Backend*.

4.2.1. Paquetes de trabajo

Estudio del estado del conocimiento

El presente paquete de trabajo engloba todas las tareas relacionadas con la documentación y el estudio necesario para el correcto desarrollo del sistema resultante de este proyecto. El estudio inicial está dividido en las siguientes tareas:

1. **Definición de los requisitos del sistema.** Se lleva a cabo la definición de una serie de requisitos que debe de cumplir el sistema. Estos requisitos son acordados con los tutores que forman parte de este proyecto.
2. **Estudio de antecedentes de los sistemas de monitorización de tránsito de personas.** En esta tarea se estudió de forma detallada los distintos sistemas de monitorización diseñados en los últimos años, enfocados principalmente a la monitorización de espacios cerrados.
3. **Estudio de dispositivos *hardware*.** Una vez estudiados los antecedentes relacionados con el proyecto, se estudiaron los dispositivos *hardware* que mejor se adaptan a las necesidades del propio proyecto, concretamente diversos tipos de sensores y placas de desarrollo. Finalmente, se eligieron los que mejor se adapten al proyecto.

Despliegue de nodos para la sensorización del entorno

En este paquete de trabajo, se ha desarrollado la parte *hardware* y la parte *software* del sistema de monitorización de la densidad de ocupación del entorno. Este sistema es el encargado de obtener y recopilar diversos datos del entorno donde se encuentra desplegado.

1. **Diseño *hardware* de dispositivos.** Consiste en el diseño *hardware* de los dispositivos (*nodos*) utilizados en el proyecto, teniendo en cuenta el estudio realizado en el paquete anterior.

2. **Desarrollo del *firmware* del nodo de la puerta (*Door Node*).** Durante esta tarea se desarrolla la lógica interna del nodo (dispositivo) de la puerta, concretamente la conexión de este nodo con los dos sensores PIR que tiene asociados. Así cuando ambos sensores PIR estén activados, el sistema detectará que ha entrado o salido una persona. Para diferenciar estas dos actividades se construyó un modelo de máquina de estados que las diferenciaba.
3. **Desarrollo del *firmware* del nodo de escaneo (*Scan Node*).** A lo largo de esta tarea se implementó la lógica interna del nodo de escaneo, este nodo fue el encargado de escanear las señales WiFi y BLE procedentes de los dispositivos inteligentes que se encuentren en el entorno de monitorización. Además, se conectó un sensor de CO2 de tipo NDIR, el cual, analizaba el nivel de CO2 del entorno.
4. **Diseño e implementación de la comunicación entre la red de nodos y el *Backend* del sistema.** En esta tarea se lleva a cabo la comunicación de este sistema con la aplicación de monitorización. Esta comunicación se produce vía MQTT, donde encontraremos desplegado un *broker* encargado de gestionar la comunicación recibida por este tipo de canales. Una vez gestionada, se despliega un agente MQTT encargado de almacenar todos los datos recibidos pertenecientes a una serie de canales específicos, de forma persistente en la base de datos de la aplicación.
5. **Valoración y pruebas.** Durante la realización de este tarea, se llevó a cabo una serie de pruebas y validaciones para comprobar si el diseño de la infraestructura junto al despliegue de los nodos de sonorización se integra de forma correcta con la aplicación de monitorización.

Aplicación de monitorización del tránsito humano.

El siguiente paquete de trabajo está compuesto por una serie de tareas encargadas del diseño y desarrollo de la aplicación de monitorización del tránsito humano. Para el funcionamiento de esta aplicación es necesario el despliegue de una serie de servicios, cada uno con una funcionalidad específica. El objetivo de esta aplicación es la representación de información relacionada con la red de nodos desplegada en el entorno, como el valor actual de los sensores, estadísticas y gráficos relacionados con los valores previamente obtenidos.

1. **Configuración e integración de los servicios utilizados por la aplicación.** En esta tarea se lleva a cabo el despliegue de los servicios necesarios. Estos servicios son desplegados en forma de contenedores de forma automática con el uso de *Docker*.
2. **Desarrollo del módulo de comunicación.** A lo largo de esta tarea, se ha implementado la recepción de datos por parte del sistema encargado de la persistencia de datos alojado en el *Backend* del sistema global.
3. **Representación y tratamiento de los datos captados por la red de nodos sensores.** En esta tarea se realiza el tratamiento y representación de los datos recopilados a través de la red de nodos de sensorización desplegados en el entorno a monitorizar.
4. **Validación y pruebas.** La finalidad de esta tarea es realizar las pruebas necesarias para validar que esta aplicación encargada de mostrar información relevante sobre el estado del entorno, se integra de forma correcta con el sistema de monitorización.

Aplicación de distribución óptima de los recursos de limpieza

Este paquete de trabajo consta de las tareas de diseño y desarrollo de la aplicación encargada de la elaboración automática de planes de desinfección.

1. **Desarrollo del módulo de definición de prioridad de cada sala.** A lo largo de esta tarea, se ha implementado una primera fase del algoritmo, en la cual, se lleva a cabo una asignación de la prioridad correspondiente a cada sala, que será posteriormente utilizado para realizar la elaboración del plan.
2. **Desarrollo del módulo encargado de la generación de planes.** Durante esta tarea, se ha desarrollado la segunda fase del algoritmo, encargada de generar todas las distribuciones de recursos humanos (personal de limpieza) posibles para cada sala, generando una serie de planes de desinfección. Cada distribución será generada en base a los parámetros recogidos anteriormente, como la prioridad, el porcentaje de ocupación, nivel de CO_2 en sala y el número de salas desinfectadas obtenido con esa planificación.
3. **Desarrollo del módulo de definición del plan más óptimo.** En este módulo se ha implementado la parte encargada de la búsqueda del plan de desinfección más óptimo, basado en la distribución de los recursos de limpieza disponibles.
4. **Desarrollo del módulo encargado de la representación de los resultados.** En esta tarea se ha implementado el procesamiento de la información recibida por parte del módulo anterior, representando dicha información en una interfaz sencilla e intuitiva para el usuario final.
5. **Validación y pruebas.** A lo largo de esta tarea, se han realizado las pruebas y validaciones pertinentes, para así corregir los posibles problemas encontrados durante el desarrollo de la aplicación, y de esta forma mejorar su funcionamiento final.

Backend

El objetivo de este paquete de trabajo es el desarrollo e integración de los protocolos de comunicación elegidos y la persistencia de los datos del sistema resultante, como se ha comentado en secciones anteriores.

1. **Definición e integración de la base de datos.** Esta tarea está enfocada a la definición y despliegue de una base de datos adecuada para este tipo de sistema, que permita almacenar la información requerida para que el sistema cumpla con su finalidad.
2. **Configuración del broker MQTT.** En esta tarea se despliega el servicio encargado de gestionar la información recibida mediante el protocolo de comunicación MQTT.
3. **Integración del agente MQTT/BD.** Durante esta tarea se desarrolla un agente encargado de recopilar la información recibida por parte de los canales MQTT por los que se comunican los nodos de la red, y los guarda en la base de datos.
4. **Diseño e implementación del cliente BD.** La tarea ha consistido en el desarrollo de un cliente para la base de datos, para establecer una conexión y así obtener el estado de cada área que forma el entorno de monitorización.
5. **Validación y pruebas.** El objetivo principal de esta tarea es validar mediante una serie de pruebas si el *Backend* desarrollado en este paquete se comporta correctamente.

4.2.2. Iteraciones

El desarrollo de este proyecto se ha llevado a cabo en un total de siete iteraciones, basado en una metodología de *desarrollo iterativo e incremental* 4.1, como se muestra en la Figura 4.1. Las iteraciones que forman el proyecto se pueden dividir en dos partes: la primera iteración centrada en un estudio inicial de los antecedentes existentes, para así poder obtener los requisitos que debe de satisfacer el proyecto; y el resto de iteraciones más centradas en el diseño, implementación y validación del sistema, junto a las aplicaciones por las que está compuesto el proyecto. A continuación, cada iteración se representa mediante una tabla en la que se indica tanto el inicio, como el final de la misma, una estimación de la duración, los paquetes de trabajo implicados junto a las tareas realizadas y los objetivos alcanzados.

Iteración 1

La primera iteración ha consistido en realizar un estudio del estado del conocimiento del proyecto. Compuesto el estudio de los antecedentes de los sistemas de monitorización en entornos de interior y el análisis de los distintos dispositivos *hardware* existentes en el mercado. De este estudio se han extraído los diferentes requisitos que debe de satisfacer el sistema a desarrollar, y los dispositivos *hardware* utilizados para diseñar e implementar los distintos tipos de nodos de sensorización a utilizar, uno de ellos para la detección y el conteo de ocupantes que entran en el área de monitorización y otro encargado del análisis de señales BLE y WiFi procedentes de dispositivos inteligentes y de la calidad del aire.

Iteración 1		
Inicio: 25/01/2021	Fin: 28/02/2021	Estimación: 100 horas
Paquete de trabajo	Tareas	Objetivos alcanzados
Pt.1	T1.1 T1.2 T1.3	- Requisitos del sistema - Estudio de los antecedentes - Elección de <i>hardware</i>

Tabla 4.1: Representación de la iteración 1

Iteración 2

Esta segunda iteración se ha enfocado en el diseño *hardware* de los dispositivos utilizados, en concreto los dos tipos de nodos por los que están formados la red de nodos sensores. Además, se ha implementado el *firmware* correspondiente al nodo de tipo *Door Node* cuya funcionalidad es la captación de movimiento en las zonas de acceso al entorno monitorizado. Finalmente, se ha iniciado el desarrollo de la aplicación encargada de la distribución de los recursos de limpieza, comenzando con la primera etapa del algoritmo principal, enfocada a la definición de prioridad única a cada área (sala) que conforma el entorno a monitorizar.

Iteración 2		
Inicio: 01/03/2021	Fin: 30/03/2021	Estimación: 90 horas
Paquete de trabajo	Tareas	Objetivos alcanzados
Pt.2 Pt.4	T2.1 T2.2 T4.1	- Diseño <i>hardware</i> de los dispositivos - Desarrollo del <i>firmware</i> de <i>Door Node</i> - Desarrollo del módulo de definición de prioridad

Tabla 4.2: Representación de la iteración 2

Iteración 3

A lo largo de la tercera iteración se ha empezado a desarrollar el *firmware* del nodo *Scan Node*. Los servicios utilizados por la aplicación de monitorización han sido configurados, en concreto la herramienta de visualización *Grafana*, al igual de la definición y la integración de la base de datos que se ha decidido utilizar para el almacenamiento persistente de los datos. Por parte de la aplicación de distribución de recursos, se ha implementado la parte del algoritmo principal encargada de la generación de planes de desinfección, basados en la distribución de los recursos de limpieza disponibles (como ya ha comentado en secciones anteriores).

Iteración 3		
Inicio: 01/04/2021	Fin: 15/04/2021	Estimación: 50 horas
Paquete de trabajo	Tareas	Objetivos alcanzados
Pt.2	T2.3	- Desarrollo del <i>firmware</i> de <i>Scan Node</i>
Pt.3	T3.1	- Configuración e integración de los servicios utilizados por la aplicación
Pt.4	T4.2	- Desarrollo del módulo de generación de planes
	T5.1	- Definición e integración de la base de datos

Tabla 4.3: Representación de la iteración 3

Iteración 4

En esta iteración se ha obtenido una versión funcional del módulo de conectividad de *Scan Node*, es decir, del módulo utilizado para el análisis de señales BLE/WiFi. Además, por parte de la red de nodos, se ha implementado la parte software encargada del establecimiento de conexión con el *Backend* del sistema para el envío de los datos recopilados sobre el estado del entorno, y su posterior almacenamiento en la base de datos. El envío y recepción de los datos enviados desde los nodos ha sido posible, gracias a la utilización de un *broker* MQTT encargado de gestionar los canales de comunicación desde los cuales están conectado los nodos de la red. Para el almacenamiento de los datos recopilados por dichos nodos, se ha integrado un agente gestor de la base de datos, el cual, establece la comunicación con cada uno de los nodos a partir de una suscripción a los canales de comunicación MQTT.

Iteración 4		
Inicio: 15/04/2021	Fin: 30/04/2021	Estimación: 60 horas
Paquete de trabajo	Tareas	Objetivos alcanzados
Pt.2	T2.3	- Versión funcional módulo conectividad <i>Scan Node</i>
Pt.3	T2.5	- Comunicación nodos y <i>Backend</i> del sistema
Pt.4	T3.2	- Módulo comunicación aplicación monitorización
Pt.5	T4.3	- Desarrollo módulo definición plan más óptimo
	T5.2	- Configuración del broker MQTT
	T5.3	- Integración agente MQTT/BD

Tabla 4.4: Representación de la iteración 4

Iteración 5

La iteración cinco ha consistido en realizar una serie de dashboards para representar los datos captados por la red de nodos de sensorización. De forma simultanea se ha desarrollado la parte correspondiente al módulo de comunicación de la aplicación de distribución de recursos, encargado de establecer una conexión a la base de datos del sistema a través del cliente de la base de datos, previamente diseñado e implementado.

Iteración 5		
Inicio: 01/05/2021	Fin: 09/05/2021	Estimación: 40 horas
Paquete de trabajo	Tareas	Objetivos alcanzados
Pt.3	T3.4	- Representación de los datos captados por los nodos
Pt.4	T4.4	- Desarrollo del módulo de comunicación
Pt.5	T5.4	- Diseño e implementación del cliente BD

Tabla 4.5: Representación de la iteración 5

Iteración 6

Durante la iteración seis se ha finalizado la representación del estado de cada área que compone el entorno de monitorización. También se ha desarrollado el módulo encargado de la representación de los resultados obtenidos a partir del algoritmo de distribución de los recursos del limpieza, realizando así el diseño y la implementación de la interfaz de escritorio de dicha aplicación.

Iteración 6		
Inicio: 10/05/2021	Fin: 31/05/2021	Estimación: 80 horas
Paquete de trabajo	Tareas	Objetivos alcanzados
Pt.3	T3.4	- Finalización representación de los datos captados por los nodos
Pt.4	T4.5	- Módulo representación de los resultados obtenidos

Tabla 4.6: Representación de la iteración 6

Iteración 7

Finalmente, la iteración siete ha consistido en la corrección de los errores encontrados a lo largo de la ejecución de las pruebas del sistema. Una vez corregidos dichos errores el sistema ha quedado validado y preparado para su despliegue. Por otra parte, durante esta iteración se ha terminado de documentar el presente documento, el cual, representa la memoria final del proyecto.

4.3. RECURSOS

Para poder llevar a cabo un desarrollo correcto del proyecto, surge la necesidad de hacer uso de recursos *hardware* y *software*. En el presente apartado, se va a describir detalladamente los diferentes tipos de recursos utilizados a lo largo del proyecto.

4.3.1. Recursos Hardware

Teniendo en cuenta el presente proyecto, los recursos *hardware* son un aspecto clave para el correcto cumplimiento de los objetivos del mismo.

- **Computador de escritorio:** equipo utilizado durante el desarrollo del sistema, modelo *Intel Nuc* compuesto por el procesador *Intel i3* con 4 núcleos y una frecuencia de 3 GHz.
- **Placa de desarrollo basada en el microcontrolador ESP32 WROOM-32**¹: dispositivo desarrollado por la empresa *Spressif* compuesto por 4MB de memoria flash y 2MB de memoria PSRAM. Este dispositivo destaca por llevar integrado un módulo WiFi (*IEEE 802.11*) y un módulo BLE (*Bluetooth Low Energy*).

¹https://www.espressif.com/sites/default/files/documentation/esp32-wroom-32_datasheet_en.pdf

- **Sensores PIR:** conectados al microcontrolador, con la funcionalidad de detectar cuerpos en movimiento en base a la variación de la temperatura de su entorno.
- **Sensor de CO2 basado en la tecnología NDIR:** sensor basado en la tecnología NDIR, encargado de medir la concentración de partículas de CO2 en el aire. Su rango de medición efectivo es de 0 a 5000 ppm (*partículas por millón*). Conectado al microcontrolador que es capaz de procesar las señales emitidas por el sensor.

En base a estos componentes, se han diseñado dos tipos de dispositivos (nodos) que forman el sistema de monitorización.

- **Nodo de puerta, Door Node:** este tipo de nodo se encuentra instalado en el marco superior de las puertas que componen el espacio cerrado. En particular, este nodo está compuesto por dos sensores PIR (uno instalado a cada lado del marco de la puerta) y un microcontrolador ESP32, este último encargado de convertir las señales eléctricas recibidas por los sensores PIR y de establecer la comunicación mediante su módulo WiFi integrado.
- **Nodo de escaneo, Scan Node:** este tipo de nodo es el encargado de escanear los dispositivos WiFi y BLE del entorno. Además, de medir la calidad de aire en base a la concentración de partículas de CO2. El nodo está compuesto por un sensor de CO2 NDIR y de un microcontrolador ESP32.

4.3.2. Recursos Software

Este tipo de recursos son de gran relevancia en el proyecto, la elección correcta de los mismos es primordial para poder sacar el máximo beneficio de los recursos *hardware* utilizados.

Sistema operativo

El primer recursos *software* a tener en cuenta es la elección del sistema operativo que se vaya a utilizar. En este caso, se ha utilizado:

- **Ubuntu 20.04 LTS:** correspondiente a la última versión estable del sistema operativo *Ubuntu*². Es una distribución de *Linux* [63] basada en *Debian* [31]. Este sistema operativo ha sido elegido debido a la compatibilidad que ofrece a la hora de desplegar diversos tipos de servicios con el uso de *Docker*, además de resultar compatible con todo tipo de placas de desarrollo, esto ha sido de gran ayuda a la hora de implementar el *firmware* de los diversos nodos que forman el sistema.

Herramientas de desarrollo

- **Docker**³: proyecto de código abierto, encargado de automatizar el despliegue de aplicaciones o servicios dentro de contenedores, además de ser multiplataforma.
- **Git**⁴: sistema de control de versiones, utilizado para el correcto desarrollo de la parte *software* del proyecto.

²<https://ubuntu.com>

³<https://www.docker.com>

⁴<https://git-scm.com>

- **Visual Studio Code**⁵ editor de código desarrollado por la empresa *Microsoft*, utilizado para el desarrollo de las distintas partes del sistema, tanto la parte *firmware* como las aplicaciones.
- **Platformio**⁶: módulo de desarrollo integrado en editor *Visual Studio Code*, el cual, proporciona una serie de características para acelerar y simplificar el desarrollo *software* en *sistemas embebidos*. Utilizado para el desarrollo del sistema de monitorización.
- **Overleaf**⁷: editor de *Latex*, facilitando la edición colaborativa de la documentación del proyecto.
- **Darw.io**⁸: herramienta orientada a la creación de diversos tipos de diagramas, como diagramas UML, de procesos, de flujo, etc. La herramienta ha sido utilizada para la creación de los diagramas del presente proyecto.

Tecnologías de desarrollo

- **C++**: lenguaje creado con el objetivo de extender las funcionalidades del lenguaje de programación *C*. En *C++* encontramos mecanismos integrados que permiten la creación y manipulación de elementos objeto. Utilizado para la implementación del *firmware* (parte *software*) de los nodos que forman el sistema de monitorización (*SM*).
- **JSON**: *JavaScript Object Notation* es un formato de texto ligero utilizado para el intercambio de datos basado en la notación literal de los objetos de tipo *JavaScript*.
- **HTML**: *HyperText Markup Language* es un lenguaje de marcado orientado a la elaboración de paginas web. Utilizado en las aplicaciones desarrolladas en el proyecto.
- **JavaScript (JS)**: lenguaje de programación interpretado, definido como un lenguaje orientado a objetos, basado en prototipos, imperativo, dinámico y débilmente tipado. Usado para la elaboración de la aplicación encargada de la elaboración de planes de desinfección.
- **Node.js**⁹: entorno de desarrollo para aplicaciones web, el cual, permite llevar a cabo una programación basada *JavaScript* para el desarrollo de aplicaciones web y móviles. Esta tecnología se ha empleado para el desarrollo de la aplicación de escritorio encargada de la elaboración de planes de desinfección.
- **MQTT**¹⁰: protocolo de mensajería estándar, orientado a *IoT*. Basado en el mecanismo de mensajería *publicación/ suscripción*, extremadamente ligero, esto lo convierte en un protocolo de comunicación idóneo para conectar dispositivos remotos que disponen de un ancho de banda de red mínimo.
- **Influxdb**¹¹: base de datos relacional, específica para el almacenamiento de series temporales de código abierto, desarrollada por *InfluxData*. Utilizada en ámbitos relacionados con la recuperación de datos de series temporales en campos como la supervisión de operaciones, métricas de aplicaciones, datos de sensores de Internet de las cosas y análisis en tiempo real. Esta base de datos ha sido empleada para almacenar los datos recopilados por parte de los nodos del

⁵<https://code.visualstudio.com/>

⁶<https://platformio.org>

⁷<https://www.overleaf.com/>

⁸<https://drawio-app.com>

⁹<https://nodejs.org/en/about/>

¹⁰<https://mqtt.org>

¹¹<https://www.influxdata.com/products/influxdb/>

sistema de monitorización.

- **Telegraf**¹²: agente de servidor para recopilar y enviar métricas desde sistemas y/o sensores de IoT a bases de datos. Empleado para inyectar los datos recopilados por parte de los nodos del sistema. Los datos son publicados en sus respectivos canales *MQTT* donde *Telegraf* se encuentra suscrito.

Bibliotecas *software*

- **Electron**¹³: *framework* de código abierto compatible con *Node.js*, permite el desarrollo de interfaces gráficas de escritorio. *Electron* ha sido utilizado para el desarrollo de una gran variedad de aplicaciones comerciales como *Visual Studio Code*, *Facebook Messenger*, *Microsoft Teams* o *Twitch*. En el presente proyecto, ha sido utilizado para el desarrollo de la aplicación de escritorio orientada a la elaboración de planes de desinfección.
- **Grafana**: software libre que permite la visualización y el tratamiento de datos métricos. *Grafana* permite crear cuadros de mando y gráficos provenientes de varios tipos de fuentes, como pueden ser bases de datos de series temporales, como *InfluxDB* ¹¹ o *Graphite*. Además, puede realizarse su implementación con la utilización de *Docker* ³.
- **Mosquitto**¹⁴: *broker MQTT* utilizado para la gestión de los canales *MQTT*, basados en *topics*¹⁵ (temas). Este servicio se encuentra desplegado en el servidor principal utilizado en el presente proyecto.
- **PubSubClient**¹⁶: esta biblioteca proporciona un cliente para hacer mensajes simples de publicación/subscripción con servidores compatibles con el protocolo de comunicación *MQTT*. Utilizada para el establecimiento de conexión vía *MQTT* entre los nodos del sistema y el servidor donde se encuentra desplegado el *broker MQTT*.
- **NumPy**¹⁷: biblioteca compatible con el lenguaje de programación *Python*, encargada de dar soporte para la creación vectores y matrices multidimensionales, junto con una gran colección de funciones matemáticas de alto nivel para realizar diversos tipos de operaciones. Se ha utilizado en la parte relacionada con el *Backend* de la aplicación encargada de la elaboración de planes de desinfección.

¹²<https://www.influxdata.com/time-series-platform/telegraf/>

¹³<https://www.electronjs.org>

¹⁴<https://mosquitto.org>

¹⁵<https://mosquitto.org/man/mqtt-7.html>

¹⁶<https://github.com/knolleary/pubsubclient>

¹⁷<https://numpy.org>

ARQUITECTURA

En el presente capítulo se va a realizar una descripción técnica y detallada del sistema diseñado y, posteriormente, desarrollado. El capítulo pone el foco en el modo en que se han alcanzado los objetivos propuestos y cómo se han solucionado los distintos problemas que han surgido. La descripción comienza con una visión general del sistema, para así avanzar hacia detalles más específicos del mismo. De esta forma, se pretende aportar una versión global al lector y, de forma gradual, presentar los detalles de la propuesta.

5.1. VISIÓN GENERAL

Tal como se comentó en secciones anteriores, el objetivo principal de la solución desarrollada, es la monitorización del tránsito de personas en un contexto de pandemia, al mismo tiempo, la evaluación de la calidad del aire y, finalmente, la elaboración automática de planes de desinfección. En el caso particular de este trabajo, los planes se centran en la distribución del personal de limpieza por salas.

El modo en el que se interrelacionan las distintas partes que conforman la propuesta es el siguiente: el sistema se encuentra dividido en dos subsistemas con un único perfil de usuario, el *administrador*. El *administrador* podrá realizar un análisis del entorno que está siendo monitorizado, a partir de la información recopilada por el despliegue de una red de sensores inalámbricos en dicho entorno. Por otro lado, en base a la información previamente recopilada, el *administrador* podrá obtener de forma automática un plan de desinfección basado en la distribución óptima del personal de limpieza, logrando así una desinfección eficaz del entorno.

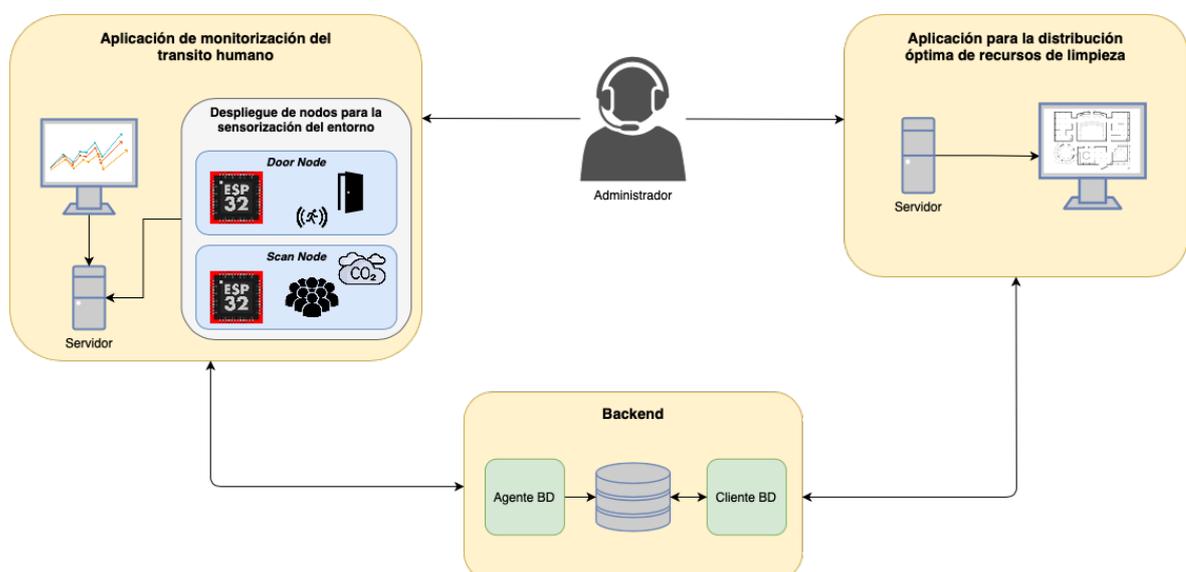


Figura 5.1: Esquema general del sistema propuesto.

Para mayor detalle, se realizará una breve descripción de las partes del sistema mencionadas anteriormente (mostradas en la Figura 5.1):

- **Aplicación de monitorización del tránsito humano.** Encargada de llevar a cabo la monitorización del tránsito humano dentro del entorno. Esta parte del sistema hará uso de la infraestructura desplegada previamente, formada por una red de sensores inalámbricos (WSN), a partir de la cual, se recopilará información del entorno relacionada con la ocupación y la calidad de aire en el mismo. La aplicación realiza un tratamiento de los datos y los representa de forma detallada al *administrador*.

- **Despliegue de nodos para la sensorización del entorno.** Se realiza un despliegue de nodos inalámbricos, encargados de la sensorización del entorno. A partir de éstos, se estima información del nivel de presencia y densidad de ocupación de dicho entorno. Como se muestra en la Figura 5.1, este despliegue está compuesto por dos tipos de sensores:
 - **Door Node o nodo de la puerta.** Colocado en la/s puerta/s que dan acceso al espacio cerrado. Este nodo está formado por un SoC y dos sensores basados en la tecnología PIR. El SoC es el encargado de procesar las señales recibidas por ambos sensores PIR y enviar la información a la aplicación correspondiente.
 - **Scan Node o nodo de escaneo.** Este nodo está compuesto por un SoC y un sensor de CO₂ basado en la tecnología NDIR. Este nodo realiza un control de la calidad del aire, y a su vez, lleva a cabo un análisis de la señales de tipo BLE y WiFi del entorno correspondiente. Tras esto, envía la información recopilada a la aplicación de monitorización del tránsito humano, encargada del tratamiento de los datos y de su representación.

- **Aplicación para la distribución óptima de recursos de limpieza.** Proporciona al *administrador* la herramienta para llevar a cabo una distribución óptima de los recursos de limpieza disponibles, en aras de realizar una desinfección del entorno lo más optimizada posible. Para ello, se le otorga una prioridad de desinfección y una cantidad de recursos de limpieza a cada espacio cerrado (sala) en el interior de un edificio.

- **Backend del sistema.** Esta parte del sistema es la encargada de la administración de la base de datos y de interactuar con las aplicaciones anteriores cuando sea necesario actuar sobre los datos persistentes del sistema.

En la siguiente sección, se describirá con un mayor nivel de detalle cada una de las partes que componen el presente esquema general (Figura 5.1).

5.2. ARQUITECTURA DEL SISTEMA

En la sección anterior se ha realizado una descripción general del sistema propuesto, con el objetivo de ofrecer al lector una visión global sin entrar aún en detalles técnicos. En la Figura 5.2, se puede apreciar la arquitectura del sistema, con una división en capas independientes, en la que se encuentran los componentes principales junto a sus respectivos módulos, los cuales, desempeñan una función concreta dentro del sistema global.

5.2.1. Aplicación de monitorización del tránsito humano

Esta aplicación es uno de los componentes principales del sistema. El objetivo primordial de la aplicación es ofrecer toda la información relacionada con el nivel de ocupación y de calidad del aire

del interior de cada sala que compone la infraestructura o edificio. La aplicación posee un módulo de comunicación para interactuar con el resto de subsistemas del sistema global.

A través del módulo de comunicación, la aplicación interactúa con el *Backend* del sistema para obtener los datos recopilados por parte de la red de nodos de sensorización inalámbricos, tipo WSN. El subsistema que forma la red de nodos para la sensorización del entorno está formado por dos tipos de nodos:

- **Door Node.** Como se ha comentado, este nodo está instalado en la/s puerta/s que dan acceso a la sala (espacio cerrado), más concretamente en la parte superior de la/s misma/s. De esta forma, se consigue realizar una detección correcta de los ocupantes que entran en el entorno que se está monitorizando. El nodo dispone de un módulo de detección de acceso, el cual, a través de los eventos de detección lanzados por los sensores PIR, permite realizar el conteo de los ocupantes que acceden al entorno, mediante el adecuado procesamiento en el SoC integrado en el nodo.
- **Scan Node.** Este es el segundo tipo de nodo que forma el subsistema encargado de la obtención de datos a través de los nodos sensores. El nodo de tipo *Scan Node*, dispone de dos módulos, el primero de ellos se encarga de realizar una detección de los dispositivos inteligentes que se encuentran en el entorno, todo ello a partir del análisis periódico de señales BLE y WiFi que emiten este tipo de dispositivos. El segundo módulo, está dedicado al control de la calidad del aire empleando el sensor de CO₂ NDIR. Dicho sensor estima métricas relacionadas con el nivel de concentración de partículas de CO₂ en el entorno, convertidas en valores digitales por el conversor analógico-digital del SoC. La parte relacionada con la captación de señales BLE y WiFi se lleva a cabo haciendo uso del módulo de conectividad integrado en el propio SoC (transceptor)..

La conexión entre la red de nodos sensores y la aplicación de monitorización es inalámbrica, gracias a la utilización del *broker MQTT Mosquitto*, basado en el protocolo de comunicación **MQTT**. De esta forma, se consigue que los datos recopilados por la red sean obtenidos de forma eficiente y en tiempo real.

El módulo de comunicación es el encargado de la interacción entre la aplicación de monitorización y el resto de subsistemas. Para recibir los datos recopilados a través de la red de nodos sensores, dicho módulo se comunica con la base de datos, en la cual, se encuentran almacenados de forma persistente los mismos. Todo ello, mediante la utilización de *telegraf* un agente encargado de inyectar en la base de datos, la información proveniente del *broker MQTT* al que está conectada la red de nodos inalámbricos.

La aplicación web se ha desarrollado con la utilización del software libre *Grafana* que ofrece una amplia variedad de *dashboard* y extensiones para representar de forma detallada e intuitiva toda la información del entorno. Por otro lado, en la parte del desarrollo de la red de sensores inalámbricos se ha usado *C++* como lenguaje de programación en el desarrollo del *firmware*; la parte *software* que gobierna el SoC y que encontramos en cada tipo de nodo sensor. Este lenguaje se ha elegido debido al alto nivel de rendimiento que ofrece en este tipo de componentes físicos. El rendimiento es un parte clave, ya que surge la necesidad de que el funcionamiento de la red nodos sea constante y en tiempo real.

5.2.2. Aplicación para la distribución óptima de recursos de limpieza

Esta aplicación es la encargada de dotar al *administrador* de la información necesaria para poder realizar una distribución óptima de los recursos de limpieza disponibles, en base a la ocupación reciente de las salas, el número de individuos que la han ocupado, y la calidad del aire en cada sala.

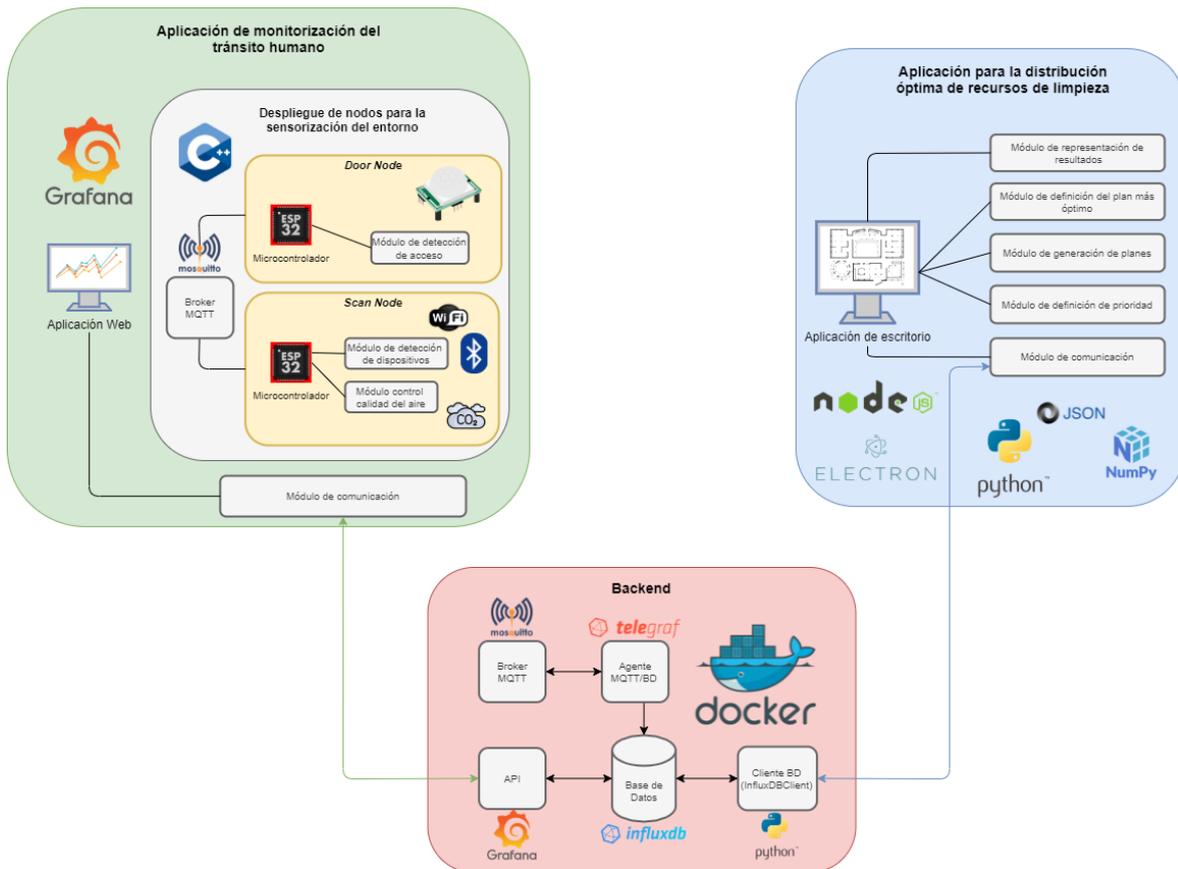


Figura 5.2: Esquema de la Arquitectura del sistema

En concreto, en este trabajo, los recursos de limpieza disponibles serán relativos a un equipo humano de limpieza. La aplicación ofrece un alto grado de flexibilidad y adaptación a cualquier entorno, con posibilidad de modificar tanto el número de salas o áreas de desinfección y sus características, como al conjunto de componentes que forman el equipo de limpieza.

La aplicación, a su vez, está compuesta por cinco módulos, cada uno con una función específica.

- Módulo de definición de prioridad:** encargado de dar un rango de prioridad único a cada área de desinfección. Esta prioridad dependerá de una serie de parámetros característicos de cada área, como el porcentaje de ocupación y el nivel de concentración de CO₂ en dicha sala. El sistema diseñado es lo suficientemente flexible como para que el administrador, en un entorno determinado a monitorizar, decida qué criterios deben ser los de mayor prioridad.
- Módulo de generación de planes:** cuyo objetivo es la generación de planes de desinfección, basados en la distribución de los recursos disponibles. De esta forma, se distribuirán los recursos disponibles (en este caso, de los integrantes que forman el equipo de limpieza). Para cada distribución posible se calculará un beneficio, para así obtener la distribución óptima (aquella con el valor de beneficio mayor). Los criterios mencionados en el punto anterior, con mayor prioridad, serán los que tendrán más peso en el cálculo de beneficio.
- Módulo de elaboración del plan más óptimo:** este módulo ha sido el encargado de obtener el plan de desinfección óptimo, en base a la distribuciones realizadas por parte del módulo anterior (módulo de generación de planes). De esta forma, se consigue la distribución óptima de los recursos disponibles para un entorno interior en concreto, ya que como se ha mencionado anteriormente, el sistema posee un alto nivel de adaptación al entorno interior correspondiente.

- **Módulo de representación de los resultados:** encargado de la representación de los resultados en la interfaz gráfica de escritorio, de manera detallada e intuitiva para el usuario. Para ello, desde el *Backend* de la aplicación deben almacenarse los resultados obtenidos, para posteriormente ser obtenidos por parte del *frontend* de la aplicación para su representación.
- **Módulo de comunicación:** la tarea de este módulo es la de ofrecer a la aplicación la capacidad de comunicarse con otros subsistemas.

La aplicación se ha desarrollado haciendo uso de diferentes tecnologías y lenguajes. En la parte correspondiente al *Backend* de la aplicación se ha utilizado el lenguaje de programación *Python* debido al amplio conjunto de librerías que ofrece, como es el caso de *NumPy* frecuentemente utilizado en el tratamiento de datos. Además, la principal motivación de haber elegido este lenguaje es la compatibilidad que ofrece con *frameworks* utilizados para el desarrollo del *frontend* de la aplicación. El *frontend* se ha desarrollado en *Node.js* y *Electron*, ambos *frameworks* basados en los lenguajes de programación web, como *JavaScript*, *CSS* y *HTML*.

5.2.3. *Backend*

El sistema desarrollado en el presente proyecto requiere un almacenamiento persistente de los datos, debido a que tanto la aplicación de monitorización del tránsito, como la aplicación enfocada a la distribución óptima de los recursos deben disponer de los datos previamente guardados. Por lo tanto, surge así la necesidad de garantizar el acceso a dichos datos siguiendo una arquitectura conforme al proyecto. Es por ello, que el componente elegido para alcanzar los requisitos de la persistencia de datos es la creación de una base de datos, que se encuentra gestionada por una agente, y dispone de un cliente para realizar peticiones. Por lo tanto, el *Backend* estaría formado por los siguientes componentes:

- **Base de datos:** se ha utilizado una base de datos dirigida al almacenamiento de series temporales (**TSDB**), denominada *InfluxDB*. Se ha elegido este tipo de base de datos, debido a la rapidez que ofrece a la hora de realizar el almacenamiento y procesamiento de datos acompañados de una marca temporal (*timestamp*), algo que resulta primordial en el sistema propuesto, debido a la gran cantidad de datos almacenados en un corto - medio periodo de tiempo. De esta forma, al disponer de un índice identificativo basado en una marca temporal, el sistema encargado de su gestión dispone de un índice sencillo a la hora de realizar la escritura de datos, sin dedicar parte de su tiempo a la organización de los mismos, como es el caso de las base de datos relacionales clásicas.
- **Broker MQTT:** encargado de la gestión de los canales de comunicación MQTT, a través de los cuales la red de nodos sensores se comunica con el sistema. El *broker* MQTT utilizado ha sido *Mosquitto*, debido a su método ligero de mensajería basado en el patrón publicador/subscriptor definido por el protocolo de comunicación MQTT, y la fácil adaptación que ofrece, haciéndolo idóneo para su utilización en este tipo de sistemas.
- **Agente MQTT/DB:** este agente es el encargado de la gestión de la base de datos. Se ha utilizado como agente *telegraf*, debido a la compatibilidad que ofrece con la base de datos *InfluxDB*. Además, *telegraf* dispone de extensiones compatibles con el protocolo de comunicación MQTT, lo que le permite conectarse al servidor *Mosquitto*, mediante una suscripción a los canales donde cada nodo (que forma la red de nodos de sensorización) publica los datos recopilados. Tras esto, *telegraf* guarda dichos datos en la base de datos.
- **Cliente BD:** el cliente de la base de datos, *InfluxDBClient* es un cliente específico de esta base

de datos. Se utiliza para obtener los datos necesarios por parte de la aplicación encargada de la distribución óptima de recursos de limpieza. Además, este cliente ha sido elegido debido a la compatibilidad que ofrece con el lenguaje de programación *Python*, ya que la parte relacionada con el *Backend* de la aplicación de distribución de recursos también ha sido desarrollada con este lenguaje, como se ha mencionado anteriormente en el apartado 5.2.2.

Cabe destacar que todos los servicios anteriormente descritos que componen el *Backend* del sistema global, son desplegados en forma de contenedores haciendo uso del *software* de virtualización *Docker*, aportándole así una mayor grado de flexibilidad y portabilidad al sistema.

5.3. DESPLIEGUE DE NODOS PARA LA SENSORIZACIÓN DEL ENTORNO

Anteriormente, en el Capítulo 3, se justificó la necesidad de desplegar una red de nodos inalámbricos (WSN) para la sensorización del entorno, algo imprescindible para el desarrollo de este proyecto. El dispositivo elegido, en el que están basados estos nodos, ha sido la placa de desarrollo *ESP32-DevKitC* (equipada con el SoC *ESP32-WROOM-32*), ya que cumple con las especificaciones del proyecto, comentadas en la Sección 3.3.

5.3.1. Visión general

El despliegue de la red de nodos está basado en la recopilación de datos que están relacionados con el estado del entorno. En esta sección se realizará una descripción en detalle de cada uno de los dos tipos de nodos que componen la red, denominados *Door Node* y *Scan Node*.

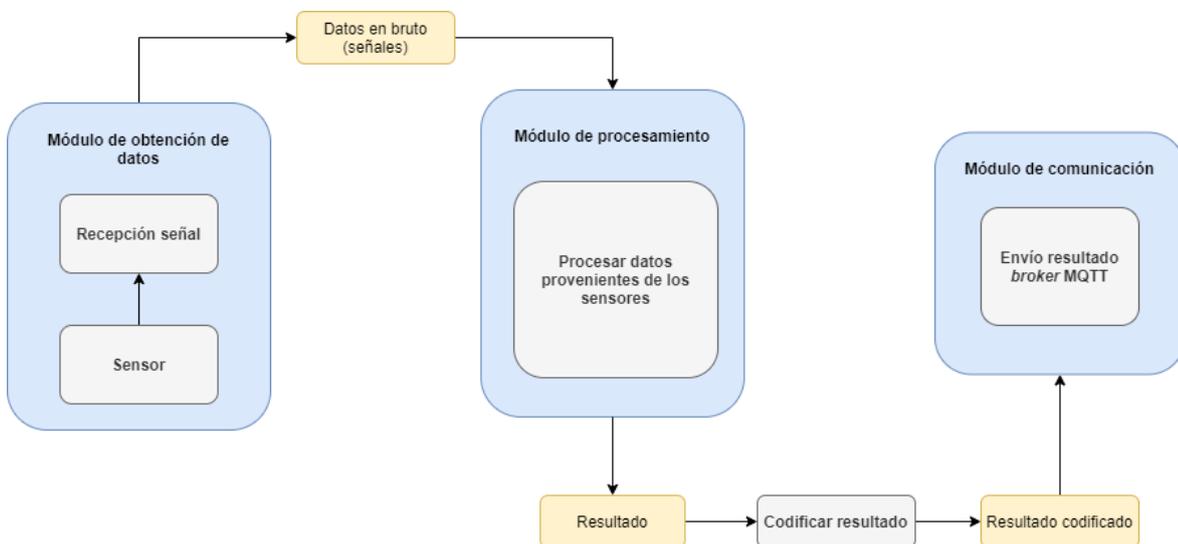


Figura 5.3: Esquema general del comportamiento de los nodos.

Para cada uno de los dos tipos de nodos, empleados en el proyecto, se ha implementado un patrón de módulos en común, mostrados en la Figura 5.3. El primero de ellos es el módulo encargado de la obtención de los datos adquiridos por parte de los sensores utilizados, en este caso, los datos obtenidos realmente son diversos tipos de señales provenientes de dichos sensores (de tipo digital y analógico). A continuación, las señales obtenidas (como datos en bruto) son tratadas por el módulo de procesamiento, obteniendo un resultado en base a dichos datos en bruto. Una vez obtenido el resultado (el cual, indica información sobre el estado del entorno de monitorización en ese momento, como aforo, nivel de CO_2 , cantidad de dispositivos inalámbricos, etc), este será codificado en un

formato ligero y compatible para su envío. Finalmente, el módulo de comunicación es el encargado de enviar el resultado codificado al servicio encargado de la gestión de la comunicación de la red (*broker* MQTT), a través del canal correspondiente.

Una peculiaridad de esta red de sensorización WSN, es la gran cantidad de nodos que se pueden utilizar, teniendo en cuenta que en cada área a monitorizar es necesario, como mínimo, el despliegue de dos nodos: *Door Node* encargado de controlar el acceso a dicho área, y *Scan Node* encargado del análisis de señales proveniente de los dispositivos inalámbricos y el control de la calidad del aire. Sin embargo, todo esto puede dar lugar a una posible saturación del servicio encargado de la gestión de su comunicación. Por ello, se ha utilizado el protocolo de comunicación MQTT, ya que en la red desplegada se utilizan dispositivos de escasa potencia y con un tiempo de funcionamiento elevado, el uso de este protocolo se adapta a las condiciones de la red, al requerir un ancho de banda mínimo, siendo esta última una característica fundamental en las redes inalámbricas.

En los siguiente apartados, se describirá con un mayor nivel de detalle el funcionamiento de ambos tipos de nodos que componen la red de sensorización WSN.

5.3.2. *Door Node*

Door Node es un nodo de detección de presencia basado en la tecnología PIR, encargado de contabilizar el número de ocupantes que acceden a cada área o sala que compone el entorno de monitorización (ver Sección 5.2).

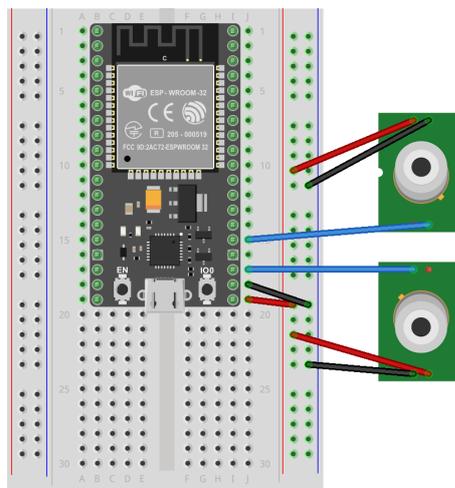


Figura 5.4: Diagrama del prototipo *hardware* de *Door Node*.

En la Figura 5.4, se puede observar un diagrama basado en el prototipo *hardware* de *Door Node*. En él aparecen los componentes que forman este nodo, entre los que se encuentran la placa de desarrollo utilizada, en concreto *ESP32 DevKitC* con el SoC *ESP32-WROOM-32*, una *protoboard* donde va anclada la placa de desarrollo y los dos sensores PIR de detección de presencia utilizados. El uso de la *protoboard*, se debe a la facilidad ofrecida a la hora de realizar las conexiones, ya que en el caso de los sensores PIR, se puede observar como se ha sacado una conexión desde el pin de alimentación (3.3V) y el pin de tierra (GND) de la placa, para así poder alimentar ambos sensores PIR. Además, la utilización de la *protoboard* ofrece mayor seguridad a la hora de conectar los componentes a través de los cables, dando un mayor agarre, disminuyendo así la aparición de posibles problemas provocados por una mala conexión.

Centrándonos en su funcionamiento, en la Figura 5.5 se muestran los diversos módulos en los que está basada la implementación de la parte *software* de este nodo, con la utilización del SoC

ESP32-WROOM-32 (Figura 3.16) para la ejecución de instrucciones de dicho *software*. Los sensores de detección PIR están conectados al SoC encargado de la obtención, procesamiento y envío de datos. El primer módulo es el encargado de la obtención de datos en bruto provenientes de los sensores de detección PIR, al tratarse de sensores digitales, los datos obtenidos son realmente valores de respuesta binaria 0 ó 1 si se detecta presencia o no. Por lo tanto, una vez recibido el dato en bruto por el módulo de procesamiento, se lleva a cabo un tratamiento del mismo. Este tratamiento está basado en la comprobación de la secuencia de datos, es decir, en el marco o *frame* recibido. En base a esta secuencia se llevan a cabo dos acciones, las cuales son i) un incremento o un decremento del contador de aforo, ya que dependiendo del *frame* recibido se interpreta si un ocupante entra o sale de la sala. El módulo de procesamiento devuelve un resultado, que representa el aforo (número de ocupantes que hay en ese momento) de la sala. El resultado debe de ser preparado para su envío, por lo que es codificado para posteriormente ser interpretado por *telegraf* (agente MQTT y de la base de datos). Finalmente, ii) mediante el módulo de comunicación se establece la comunicación con el *broker* MQTT y el dato resultante es enviado para su posterior gestión y almacenamiento.

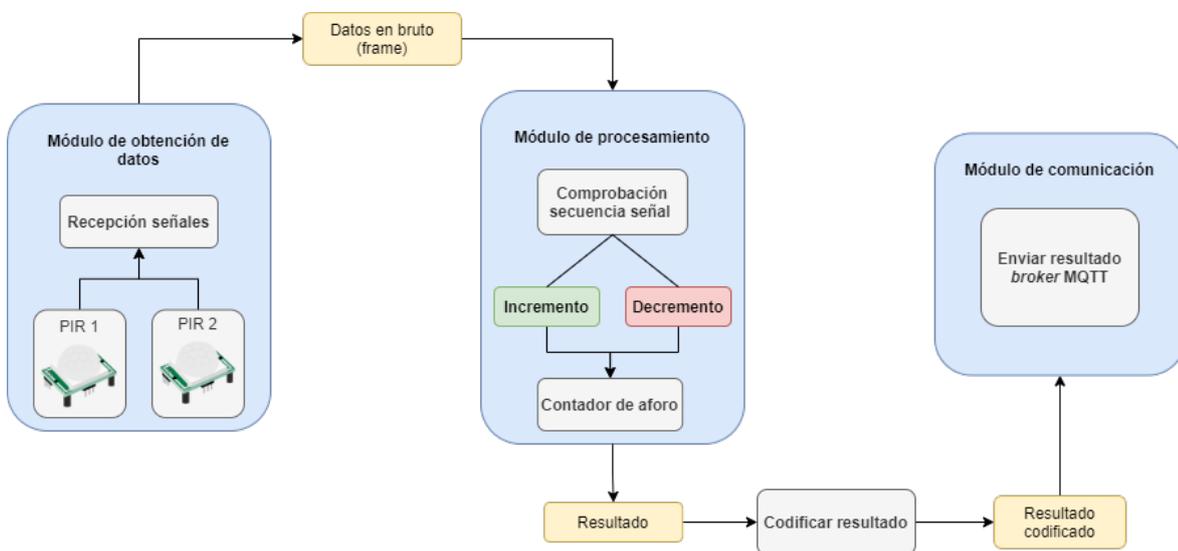


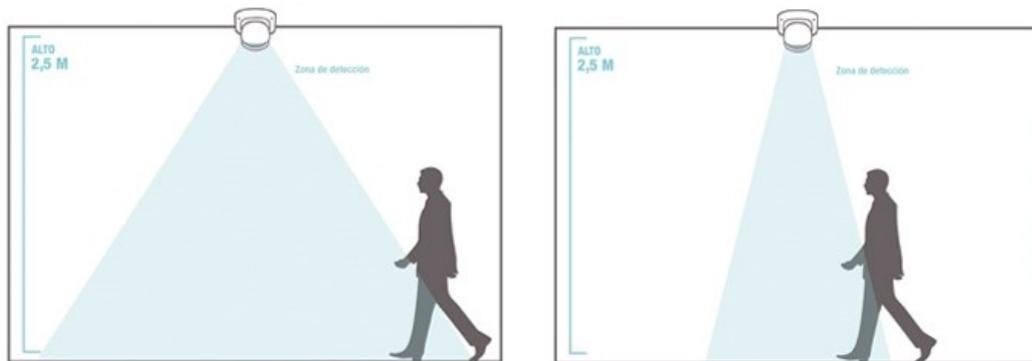
Figura 5.5: Esquema detallado sobre el funcionamiento de *Door Node*.

Manejo de los sensores de detección PIR

El componente principal de un sensor de detección basado en la tecnología PIR es el sensor piroeléctrico que lleva integrado detrás de la lente de *fresnel* (tapa compuesta por plástico), mostrado en la Figura 5.7 (a). El sensor piroeléctrico se encuentra dividido en dos mitades. Cuando no se detecta ningún tipo de movimiento por parte del sensor, ambas mitades reciben la misma cantidad de radiación infrarroja (IR) del entorno. En cambio, cuando un objetivo cruza el sensor, el nivel de radiación IR recibido en una de las mitades es bastante superior respecto al de la otra mitad. El sensor PIR reacciona a dicho cambio, modificando la señal de salida a modo Alto (1), provocando así un cambio en la señal emitida al pin digital del SoC, donde se encuentra la parte lógica del *Door Node*. Por lo tanto, el valor de la señal recibida es modificado de 0 a 1 [53].

Los sensores PIR pueden usarse en una gran cantidad de situaciones enfocadas a la detección de presencia, como se ha mencionado a lo largo del Capítulo 3. En este trabajo, el propósito de este tipo de sensor es contabilizar el número de personas que hay en el área (sala) de monitorización. Sin embargo, con el uso de un único sensor de detección se hace complicado llevar a cabo esta tarea, ya que no se puede detectar de forma exacta la dirección del movimiento, es decir, si el ocupante está accediendo o saliendo de la sala. Por esta razón, surge la necesidad de utilizar dos sensores de detección, y así dependiendo de la secuencia de activación de dichos sensores, producida al detectar movimiento, se podrá estimar el tipo de acción que está siendo llevada a cabo por el ocupante: entrada

a la sala vigilada, o salida.



(a) Sensor PIR por defecto.

(b) Sensor PIR modificado (prototipo).

Figura 5.6: Comparación ambos campos de visión del sensor PIR.

Ambos sensores de detección PIR han sido instalados en una posición específica dentro de la sala, para así captar de forma precisa la entrada y salida de los ocupantes. La posición elegida ha sido en el marco superior de la/s puerta/s que dan acceso a dicha sala. Al tratarse de una posición relativamente cercana al espacio donde se está llevando a cabo la acción por parte del ocupante (máximo un metro de distancia, dependiendo de la altura de la/s puerta/s), como se presentó en el apartado 3.1, el campo de visión de este tipo de sensores es bastante amplio. Por ello, se ha limitado el mismo, encapsulando ambos sensores en un prototipo de caja abierta, donde las paredes de la misma obstruyen las lentes de fresnel innecesarias del sensor, reduciendo así su campo de visión (ver Sección C.1 del Anexo C). En la Figura 5.6 se muestra una representación sobre la disminución producida en el rango de detección.



(a) Parte frontal del sensor.

(b) Parte de conexión del sensor (Pinout).

Figura 5.7: Sensor de detección PIR digital [53].

El sensor PIR utilizado es de tipo digital (Figura 5.7), se ha realizado esta elección respecto a un sensor PIR de tipo analógico debido a que aunque ambos sensores se ajustan de manera óptima a la tarea a realizar por este tipo nodo (*Door Node*), un sensor analógico basado en la tecnología PIR requiere un post-procesamiento de la onda digital resultante, fruto de la conversión analógico-digital de la señal de variación de tensión/voltaje que genera este tipo de sensor. Por su parte, un sensor digital proporciona una respuesta digital binaria que determina presencia o no presencia en el entorno monitorizado. En entornos complejos en los que se debe detectar presencia con mayor precisión o cubriendo mayores distancias un sensor PIR analógico (con el conversor analógico-digital

y post-procesamiento adecuados) puede aportar un mejor rendimiento, si bien, para el caso de un sensor PIR dispuesto en el marco de una puerta con un rango de alcance corto y restringido el sensor PIR digital proporciona ventajas por su rápido despliegue y su fácil comunicación con el SoC.

Proceso de contabilización de individuos

Como se ha mencionado con anterioridad, el objetivo de *Door Node* es llevar a cabo un conteo del número de individuos que se encuentran en el interior de una sala (área a monitorizar). Para ello, es necesario que este nodo pueda diferenciar qué tipo de acción está realizando el ocupante, es decir, si es de acceso a la sala o de salida de la misma.

En el Listado 5.1 se puede observar como, en base a la secuencia de activación de los sensores PIR, se lleva a cabo dos acciones diferentes. Estas acciones son el incremento y el decremento del número de individuos en la sala. En este caso, se han utilizado dos variables *booleanas*: *input* y *output* ambas inicializadas como *false*, el propósito de ambas variables es marcar la secuencia recibida como una posible entrada (*input*) o salida (*output*) de un individuo de la sala. Además, se define el valor de cada sensor, como: *s1_value* (valor sensor uno) y *s2_value* (valor sensor dos). De esta forma, dependiendo de la secuencia de activación de los sensores, se le aplicará un incremento o un decremento al contador global del número de ocupantes.

Listado 5.1: Algoritmo de contabilización de ocupantes.

```
1 void people_counter()
2 {
3     s2_value = digitalRead(PIR2_PIN);
4     s1_value = digitalRead(PIR1_PIN);
5
6     /* States Machine */
7     if (s1_value == HIGH && s2_value == HIGH)
8     {
9         if (input)
10            people_counter++;
11        else if (output)
12            people_counter > 0 ? people_counter-- : people_counter = 0;
13
14        // Reset States
15        input = false;
16        output = false;
17    }
18
19    else if (s1_value == LOW && s2_value == HIGH)
20    {
21        input = true;
22        output = false;
23    }
24
25    else if (s1_value == HIGH && s2_value == LOW)
26    {
27        input = false;
28        output = true;
29    }
30
31    else
32    {
33        input = false;
34        output = false;
35    }
36 }
```

Por ejemplo, si la secuencia de activación fuese $s1_value = HIGH$ y $s2_value = LOW$: la variable *input* cambiaría su estado a *true* y la variable *output* cambiaría su estado a *false*. En cambio, habría que esperar a la siguiente iteración para confirmar que un ocupante esta accediendo a la sala realmente, evitando así falsos positivos. Si bien la siguiente secuencia de activación tuviese el valor $s1_value = HIGH$ y $s2_value = HIGH$, y la variable *input* tiene el valor *true*, finalmente, se confirmaría que un ocupante esta accediendo a la sala y se incrementaría en uno el valor del contador (***people_counter++***). Una vez realizada esta operación, se muestra como se realiza una eliminación del estado de las variables, para así comprobar de nuevo la secuencia de activación de los sensores PIR, y detectar cuándo un nuevo individuo accede a la sala o, por el contrario, sale algún otro de ella.

Por otra parte, se puede observar como si el resultado obtenido en el ejemplo anterior hubiese sido un decremento del contador (***people_counter--***), se llevarían a cabo las mismas comprobaciones, pero dando lugar a una primera secuencia de activación inversa ($s1_value = LOW$, $s2_value = HIGH$) activando así la variable *booleana output* a *true*, y obteniendo una secuencia de activación en la siguiente iteración donde el valor de ambos sensores se encontrase a *HIGH*, confirmando así que alguien está saliendo de la sala.

5.3.3. Scan Node

Scan Node es un nodo cuya tarea principal es el control de la calidad del aire y el análisis de señales de tipo BLE y WiFi provenientes de dispositivos inalámbricos con este tipo de conectividad, y así contabilizar la cantidad de dispositivos que se encuentran en la sala, como ya se ha mencionado en la Sección 5.2 y estimar el número de ocupantes de la sala.

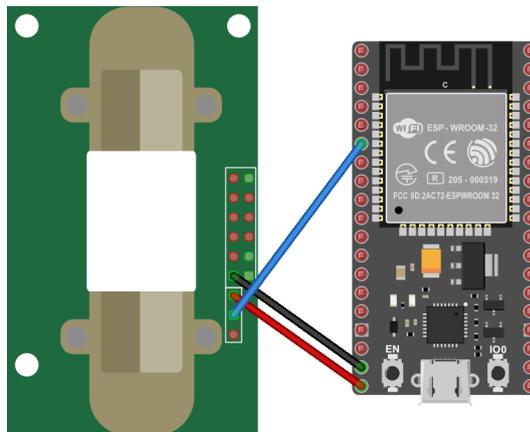


Figura 5.8: Diagrama de conexiones del prototipo *hardware* de *Scan Node*.

En la Figura 5.8 se muestra un diagrama basado en el prototipo *hardware* de *Scan Node*. En él aparecen los componentes por los que este nodo está compuesto, junto a sus conexiones. *Scan Node* está compuesto por la placa de desarrollo *ESP32 DevKitC* y el sensor CO_2 basado en la tecnología NDIR, utilizado para medir la calidad del aire. El sensor de CO_2 es alimentado a través del pin Vcc de 5 voltios de la placa de desarrollo, a la vez se encuentra conectado a un pin analógico de la misma, para así poder emitir una serie de valores relacionados con la calidad del aire del entorno. Sin embargo, estos valores son datos obtenidos en bruto, en concreto señales analógicas enviadas desde el sensor a la placa. A continuación, se presentará como estas señales son procesadas por el microcontrolador *ESP32-WROOM-32* integrado en la placa de desarrollo. Además, el propio microcontrolador será el encargado de tratar las señales BLE y WiFi obtenidas de los dispositivos inteligentes del entorno de monitorización.

La parte *software* de *Scan Node* (encargada de otorgarle funcionalidad al nodo), se compone de una serie de módulos, mostrados en la Figura 5.9. Cada uno de estos módulos está enfocado en la

realización de una tarea específica, como la obtención de datos, el procesamiento de los mismos y la comunicación con el resto de subsistemas que forman el sistema global desarrollado. El primer módulo está enfocado a la obtención de datos en bruto provenientes tanto del sensor de CO_2 , como de los transceptores WiFi y BLE. Tras recibir los datos en bruto, se lleva a cabo su tratamiento, el cual está compuesto por varias etapas: i) la primera etapa se encarga del tratamiento de la señal analógica para la obtención del nivel de CO_2 en la sala, ii) la segunda etapa es la encargada de realizar un filtrado de las MACs BLE y WiFi de los dispositivos inteligentes detectados por el nodo. Posteriormente, iii) el módulo de procesamiento devuelve como resultado el nivel de CO_2 y la cantidad de dispositivos BLE y WiFi de la sala, y el resultado es codificado en un formato óptimo su envío. Finalmente, iv) se establece comunicación con el *broker* MQTT, para la transmisión del resultado que será gestionado y almacenado de manera persistente en el sistema.

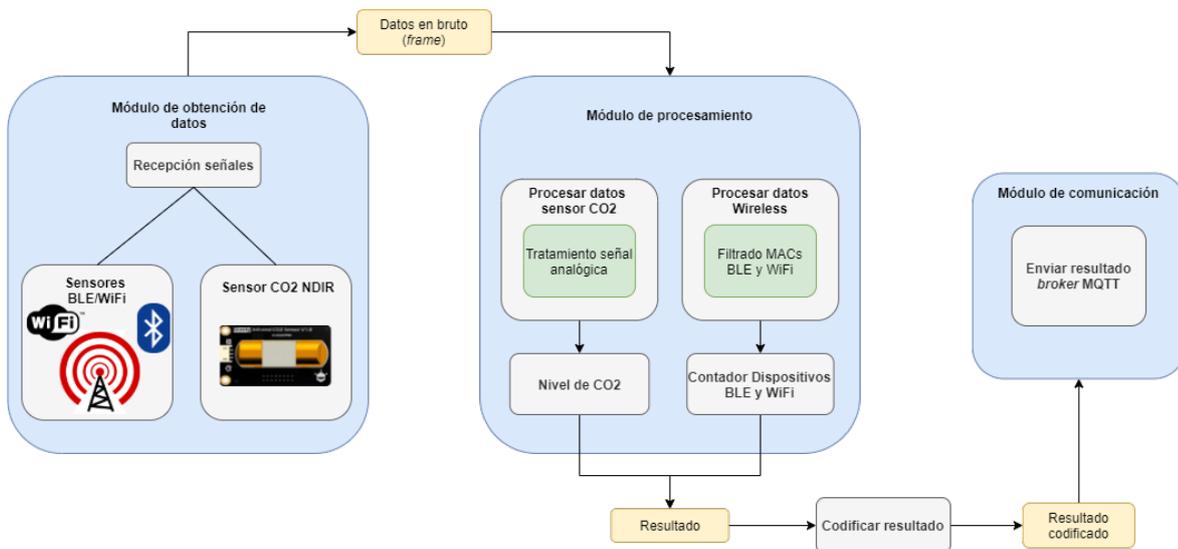


Figura 5.9: Esquema detallado sobre el funcionamiento de *Scan Node*.

Medición de la concentración de CO_2 en el entorno

Este modelo de sensor al igual que la gran mayoría de los sensores de CO_2 basados en la tecnología NDIR pueden funcionar con una tensión de alimentación entre 4,5 y 5,5 voltios, emitiendo una señal analógica cuyo valor está comprendido entre 0 y 2 voltios. Cuando el valor de la señal analógica emitida por el sensor se encuentra entre 0 y 0,4 voltios, no se presenta ningún tipo de medida y los valores entre este rango son ignorados, ya que significa que el sensor está en su etapa de precalentamiento. La etapa de precalentamiento tiene una duración máxima de tres minutos. Durante su funcionamiento normal, el sensor genera una tensión entre 0,4 y 2 voltios, correspondiendo a concentraciones de CO_2 entre 0 y 5000 ppm (*partes por millón*). En el caso particular de la placa *ESP32-DevKitC*, que lleva integrados dos ADC de 12 bits [1], se pueden obtener valores entre 0 y 4095 ($2^{12} - 1$), como medidas entre las que se encuentra la señal analógica.

En el Listado 5.2, se muestra el algoritmo implementado para este cálculo. A través de la función *analogRead()* se obtiene el valor de la señal analógica enviada por parte del sensor al Pin especificado; como se ha mencionado anteriormente, este valor se encuentra comprendido entre 0 y 4095. Este valor será convertido a su correspondiente voltaje mediante la siguiente expresión:

$$voltaje = sv * \frac{av}{max_av} \quad (5.1)$$

Donde *sv* representa el valor de la señal analógica emitida por el sensor; *av* representa el voltaje de alimentación que le ofrece la placa de desarrollo al sensor (en Listado 5.2 como *ALIM_VOLTAGE*),

en este caso es de 3,3 voltios (3300 milivoltios); y max_av es el máximo valor que puede alcanzar la señal analógica, en este caso 4095.

Tras calcular el voltaje, se procede a la comprobación del voltaje calculado, donde se pueden dar tres casos: i) si el voltaje obtenido es de 0 voltios, significa que ha surgido algún problema con el sensor; ii) si el voltaje se encuentra dentro del rango de 0 a 0,4 voltios, significa que el sensor todavía está en su etapa de precalentamiento; iii) si finalmente el voltaje obtenido es superior a 0,4 voltios el sensor esta trabajado de forma normal y se debe de convertir ese voltaje a su respectivo valor de concentración de CO_2 , medido en ppm .

El primer paso es obtener la diferencia del voltaje medido y el umbral de calentamiento, debido a que cuando el sensor emite una tensión de 0,4 voltios, corresponde a una concentración de CO_2 de 0 ppm . De esta forma, se consigue una relación lineal entre el voltaje medido y la concentración de CO_2 . La diferencia de voltaje obtenido está definida como dv , y el valor del mínimo voltaje que se puede obtener como min_volt .

$$dv = voltaje - min_volt \quad (5.2)$$

Calculada la diferencia de voltaje únicamente es necesario aplicarla para obtener así el nivel de concentración de CO_2 que hay en la sala en ese momento. Donde $conc$ representa el nivel de concentración de CO_2 calculado, max_conc representa el valor máximo de concentración de CO_2 , y max_volt como el valor máximo de tensión.

$$conc = \frac{vd * max_conc}{max_volt - min_volt} \quad (5.3)$$

Listado 5.2: Algoritmo para el cálculo del nivel de concentración de CO_2 .

```

1 float get_co2_level()
2 {
3     sensor_value = analogRead(PIN);
4     voltage = sensor_value * (ALIMENTATION_VOLTAGE /
5         ↪ MAX_ANALOGIC_VALUE);
6
7     /* Check preheating */
8     if (voltage == 0)
9     {
10        Serial.println("Fault");
11    }
12    else if (voltage < MIN_VOLTAGE)
13    {
14        Serial.println("Preheating");
15    }
16    else
17    {
18        voltage_dif = voltage - MIN_VOLTAGE;
19        concentration = (voltage_dif * MAX_PPM) / (MAX_VOLTAGE ↪
20            ↪ - MIN_VOLTAGE);
21
22        return concentration;
23    }

```

En la Tabla 5.1, se muestra la relación existente entre el valor obtenido a partir del sensor de CO_2 (fruto de la conversión analógico-digital) y su valor correspondiente medido en ppm (partes por millón), en base a los cálculos realizados anteriormente. Además, también se marca una etiqueta para

categorizar la calidad de aire en relación a cada valor, siendo esta categorización desde muy buena a inaceptable calidad del aire.

Valor Analógico	Valor ppm	Calidad del aire
205 - 328	250 - 400	Muy buena
328 - 820	400 - 1000	Buena
820 - 1640	1000 - 2000	Mala
1640 - 4095	2000 - 5000	Muy mala
4095	5000	Inaceptable

Tabla 5.1: Relación entre los valores obtenidos fruto de la conversión analógico-digital y su equivalencia con el nivel de calidad del aire.

Análisis de señales BLE y WiFi

Para el análisis de las señales BLE y WiFi provenientes de los dispositivos con conectividad que llevan consigo los ocupantes del entorno de monitorización se debe de tener en cuenta una parte clave dentro del módulo de comunicación: el modo de conexión WiFi utilizado. Dentro de los modos de conexión ofrecidos por este tipo de SoC (*ESP32*), se ha utilizado una combinación entre el *modo estación* y el *modo promiscuo*.

- **Modo estación (*Station Mode*):** este modo le ofrece al SoC la posibilidad de conectarse a una red WiFi, actuando éste como cualquier dispositivo cliente de la red.
- **Modo promiscuo (*Promiscuous Mode*):** modo que ofrece la posibilidad de actuar como un punto de acceso WiFi, y cualquier dispositivo compatible con el uso de la tecnología WiFi 802.11 puede conectarse a él. La finalidad de este modo es obtener la señal de conexión que realizan la mayoría de los dispositivos a la hora de escanear los puntos de acceso WiFi cercanos. Esta operación se realiza de forma automática y periódica por parte del dispositivo.

Como se ha mencionado anteriormente, se ha utilizado una combinación (*Station Mode + Access Point*), para así llevar a cabo esta tarea de análisis. De esta forma, se mantiene una conexión a la red WiFi existente para el envío de resultados obtenidos (por parte de estos nodos) a los servicios de gestión y almacenamiento desplegados en la misma red. En el Listado 5.3 se puede observar como se lleva a cabo este modo de conexión, activando el modo promiscuo antes de realizar el escaneo de las señales WiFi (función *sniffer*) y, una vez finalizado dicho análisis, vuelve a desactivarse dicho modo, quedando así únicamente el modo estación activado para enviar los resultados al *broker* MQTT desplegado en el sistema global.

Listado 5.3: Secuencia de código desarrollada para la definición del modo de conexión utilizado en Scan Node

```

1 void setup_promiscuous_mode()
2 {
3     wifi_init_config_t config = WIFI_INIT_CONFIG_DEFAULT();
4     esp_wifi_init(&config);
5     esp_wifi_set_storage(WIFI_STORAGE_RAM);
6     esp_wifi_set_mode(WIFI_MODE_STA);
7     esp_wifi_set_promiscuous(true);
8     esp_wifi_set_promiscuous_filter(&filter);
9     esp_wifi_set_promiscuous_rx_cb(&sniffer);
10    esp_wifi_set_promiscuous(false);
11 }

```

Para realizar el análisis de las señales WiFi, también se tiene en cuenta los distintos dispositivos existentes en el entorno de monitorización que realmente no pertenecen a ningún individuo, como impresoras, monitores, *smartTVs*, etc. Estos dispositivos han sido categorizados como *dispositivos estáticos*, a los cuales se les ha aplicado un filtrado de sus direcciones físicas (MACs). De esta forma, se pretende ignorar este tipo de dispositivos, y que así el análisis de señales WiFi sea lo más óptimo y preciso posible. Por lo tanto, cuando un nuevo dispositivo es detectado por *Scan Node*, se le realiza un filtrado a su dirección MAC, si dicha MAC no corresponde a ningún dispositivo estático existente en el entorno, se contabilizará dicho dispositivo. Si por el contrario, corresponde a algún dispositivo de este tipo, finalmente el dispositivo es ignorado.

Cabe destacar, que la utilización de varios modos de conexión WiFi se debe a que una vez realizado el análisis de este tipo de señales, el resultado debe de ser enviado a través de la red WiFi a la que esta conectado el nodo, y en la cual se encuentra desplegado el sistema. Sin embargo, a la hora de analizar las señales BLE no es necesario ningún cambio en el modo de conexión, ya que esta tecnología solo es utilizada para el análisis de señales BLE, convirtiéndose así su realización en una tarea menos laboriosa. En el Listado 5.4, se muestra el algoritmo desarrollado en lenguaje C/C++ para la realización del análisis.

Listado 5.4: Secuencia de código desarrollada para el análisis de señales BLE

```
1
2 void scan_ble(boolean &send_mqtt)
3 {
4     BLEScanResults found_devices = ble_scan->start(SCAN_TIME, false);
5     ble_devs_count = found_devices.getCount();
6     ble_scan->setActiveScan(false);
7     ble_scan->clearResults();
8 }
```

Una vez definidos el modo de conexión, como se muestra en el Listado 5.5, se lleva a cabo el análisis BLE y WiFi, junto a la medición de la concentración de CO_2 en el aire del entorno monitorizado. Finalmente, el resultado obtenido será enviado al *broker* MQTT. El resultado está compuesto por los parámetros adquiridos previamente, como el nivel de CO_2 , y el número de dispositivos BLE y dispositivos WiFi detectados. Antes de enviar el resultado, debe ser validado, comprobándose así si ha cambiado el valor de alguno de estos parámetros y si se ha alcanzado el tiempo de espera del anterior envío realizado.

Listado 5.5: Código correspondiente al bucle principal del *firmware* de *Scan Node*.

```
1 void loop()
2 {
3     boolean send_mqtt = false;
4     scan_wifi(send_mqtt);
5     scan_ble(send_mqtt);
6     my_ndir.read_value(analogRead(CO2_PIN));
7     send_mqtt += check_co2_level() + check_send_time();
8     if (send_mqtt) send_mqtt_data();
9 }
```

5.3.4. Comunicación con el *Backend* del sistema

El *firmware* que compone cada uno de los tipos de nodos de sensorización utilizados, donde ambos tipos de nodos tienen los módulos que integran dicho *firmware*. En el presente apartado se hace hincapié en el módulo de comunicación, este módulo es el encargado de comunicar cada nodo con la

parte del *Backend* del sistema global. El objetivo principal de esta comunicación es el almacenamiento de forma persistente de los datos recopilados por los mismos, relacionados con el estado de cada sala que conforma el entorno de monitorización. A través del diagrama representado en la Figura 5.10, se muestra como se lleva a cabo esta comunicación. Por cada sala se despliega un nodo de cada tipo, es decir un *Door Node* y un *Scan Node* que establecen conexión mediante un canal de comunicación MQTT con el *broker* MQTT encargado de gestionar dicha comunicación. Los canales de comunicación son diferenciados por un identificador de canal, denominado *topic* (tema). Por cada sala existe un único canal de comunicación, por el cual ambos nodos de la sala establecen una conexión para el envío de datos. De esta forma, se consigue que el sistema disponga de un nivel de escalabilidad bastante alto, debido a que todos los nodos de la misma sala comparten el mismo canal de comunicación.

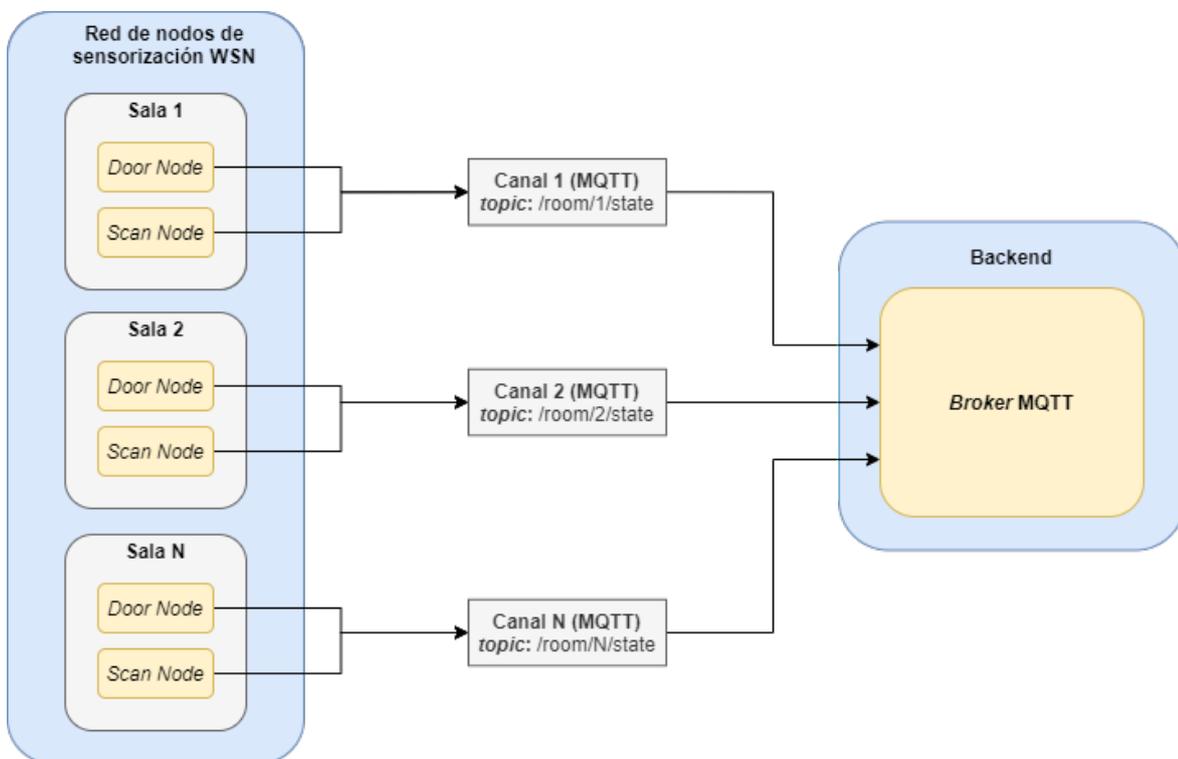


Figura 5.10: Diagrama de comunicación entre la red de nodos WSN y el *Backend*, mediante el *broker* MQTT.

Cabe destacar como a la hora de enviar el resultado obtenido, éste debe ser codificado (Figura 5.3). Dicho resultado está compuesto por una serie de parámetros relevantes sobre el estado de la sala, y su respectivo valor. Parámetros como el nivel de CO_2 , la cantidad de dispositivos BLE, la cantidad de dispositivos WiFi y el contador de aforo.

La finalidad de codificar el resultado antes de ser enviado, se debe a que el formato del mensaje que contiene dicho resultado debe de ser compatible con la línea de protocolos utilizada por la base de datos *InfluxDB*. De esta forma, cuando *telegraf* (agente MQTT/BD) obtiene cualquier mensaje proveniente de un nodo de la red, procederá al almacenamiento del mismo en la base de datos.

Por otro lado, en la Figura 5.11 se aprecia el flujo de comunicación entre los nodos que componen la infraestructura y el *broker* MQTT, encargado de la gestión de dicha comunicación. El flujo de comunicación es continuo, donde cada nodo establece de forma individual la conexión con el *broker* MQTT, una vez establecida la conexión, el nodo publica (*publish*) periódicamente datos relevantes del estado de su respectiva sala.

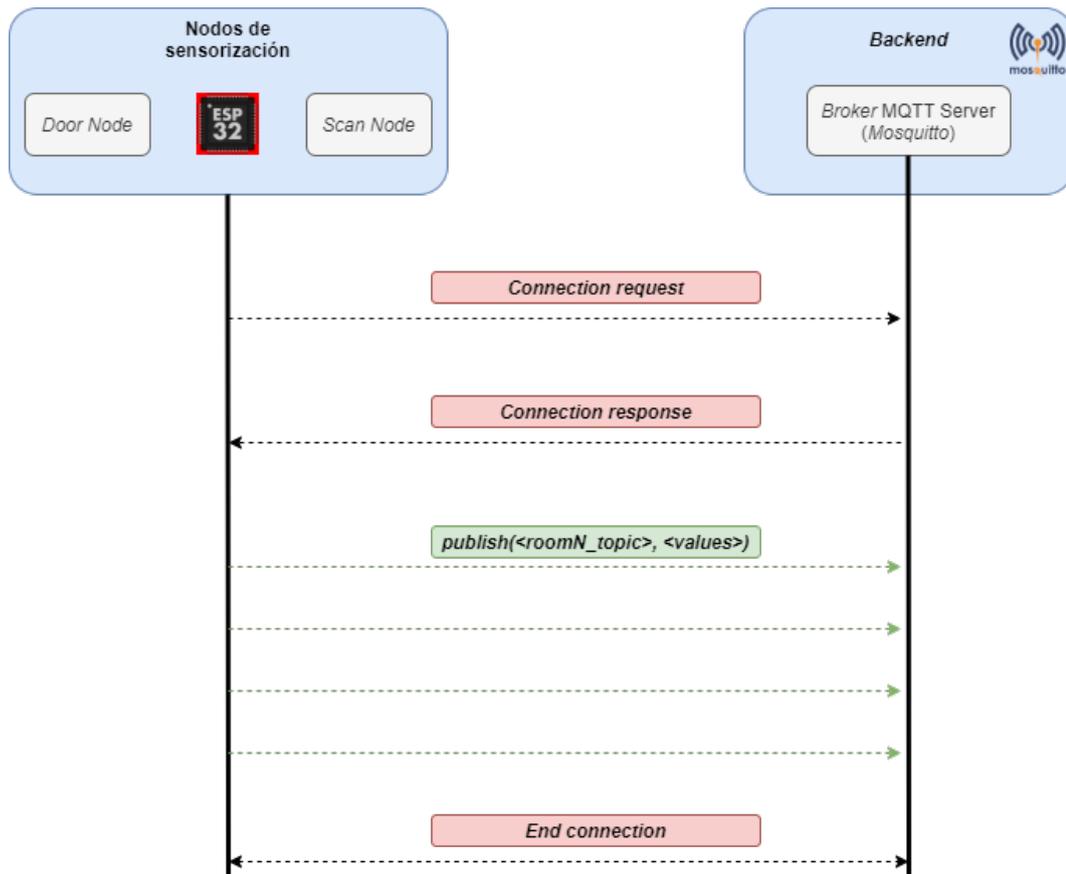


Figura 5.11: Flujo de comunicación entre la infraestructura que conforma la red de nodos de sensorización y la parte del Backend encargada de su gestión.

5.4. APLICACIÓN DE MONITORIZACIÓN DEL TRÁNSITO HUMANO

El actor principal del sistema es el *administrador*, debido a que el sistema desarrollado está enfocado a la recopilación de información relevante sobre el estado de un entorno monitorizado. Así pues, el administrador podrá realizar una serie de acciones dentro del sistema, como obtener diversas estadísticas sobre la información más relevante del entorno, obtener valores de la red de sensorización en tiempo real o comparaciones de datos en distintas áreas que conforman el entorno. Estas formas de interacción entre el sistema y el *administrador* son realizadas gracias a la aplicación específicamente diseñada para esta labor. Estas acciones se realizarán en base a la adquisición y tratamiento de los datos provenientes de la red de nodos de sensorización, como se observa en la Figura 5.12.

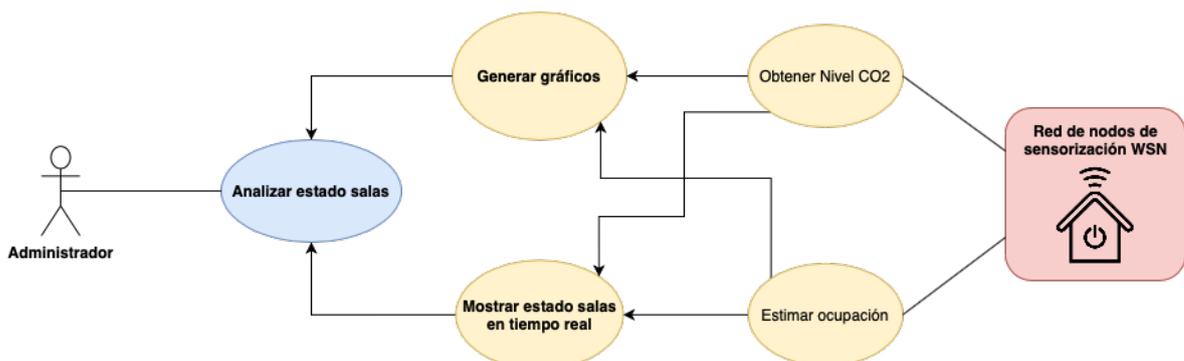


Figura 5.12: Diagrama de casos de uso de la aplicación de monitorización.

En la Figura 5.13, se muestra como la aplicación se ha dividido en varios módulos principales, cada uno de ellos con una funcionalidad diferente y un objetivo específico. La idea de usar la modularización se debe a que en ambas aplicaciones se utilizan varias partes de *software* con un objetivo similar, aunque el modo de llevar a cabo su implementación difiera en cierta medida de una aplicación a otra. De esta forma, se representa, de forma simple, la tarea a realizar por cada una de las partes que conforman el subsistema. Además, se observa cómo los módulos interaccionan entre sí, y con otro subsistema, como es el *Backend* del sistema, presentado en la Sección 5.6.

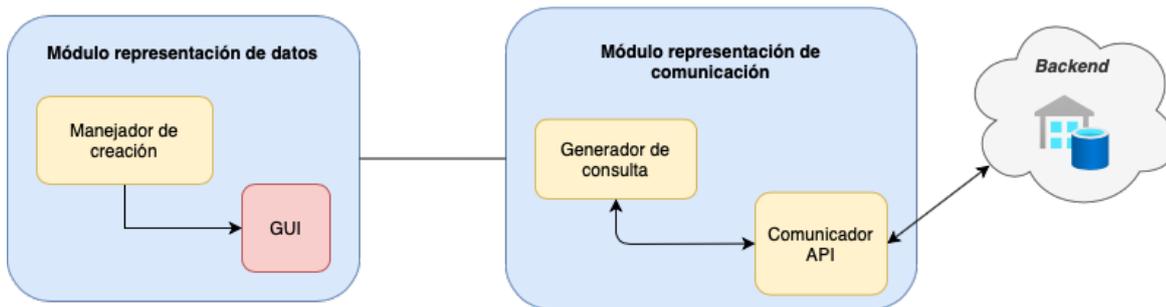


Figura 5.13: Esquema general aplicación monitorización.

5.4.1. Entradas y salidas

La aplicación basada en monitorización del tránsito humano recibe distintos datos de entrada a interactuar con el entorno. Mediante este tipo de entradas se reciben una serie de datos, a partir de los cuales el sistema produce varios tipos de salidas correspondientes. Los diferentes tipos de entradas pueden categorizarse de la siguiente manera:

- **Información del entorno.** La adquisición constante de parámetros específicos sobre el estado actual del entorno de monitorización se considera el flujo de información más importante recibido por el sistema. Dicha información se compone de datos de tipo entero como el número de dispositivos BLE/WiFi detectados, el contador de ocupación con un rango entre 0 y 100, y el nivel de CO_2 con un rango desde 0 a 5000 (medido en ppm, como se ha mostrado en la Tabla 5.1). Gracias a estos datos, se pueden realizar los demás tipos de acciones que ofrece el sistema, tanto las relacionadas con la presente aplicación, como aquellas relacionadas con la aplicación de distribución de recursos que se presentará en la siguiente sección.
- **Computador principal.** A partir de la utilización de los periféricos del computador principal, como son el teclado y el ratón, se reciben una serie de acciones llevadas a cabo por el *administrador* en la aplicación web desarrollada. El sistema guarda la información producida por estas entradas y comienza las acciones indicadas en las mismas, en este caso, relacionado con el análisis de la información previamente obtenida del entorno de monitorización.

Tras el procesamiento por parte de la aplicación de los datos provenientes de ambas entradas, se producen varios tipos de salidas o resultados. En este caso, la salida principal corresponde a una serie de gráficos sobre el estado de las salas (áreas) que conforman el entorno de monitorización, como se muestra en el siguiente apartado (5.4.2).

En la Figura 5.14, se observa un ejemplo sobre la generación de uno de estos gráficos. El gráfico es generado con la utilización de una sentencia *SQL* (a), a través de la cual, se obtiene de la base de datos la selección de datos requerida. En concreto, el ejemplo se centra en la obtención de datos relacionados con el estado de la calidad del aire en base al nivel de CO_2 de la sala número 1.

```
SELECT mean("co2") AS "mean_co2" FROM "rooms_db"."state" WHERE time > :dashboardTime: AND time < :upperDashboardTime: AND "room_id"='1' GROUP BY time(:interval:)
```

(a) Consulta SQL.



(b) Gráfico representativo.

Figura 5.14: Ejemplo de generación de un gráfico en base a una selección de datos mediante consulta SQL.

5.4.2. Representación de la información

Uno de los objetivos del presente trabajo es el análisis de ocupación y calidad del aire en el entorno, representando dicha información de una forma intuitiva y de fácil entendimiento para del usuario, en este caso, el *administrador*. Para representar esta información se ha utilizado todo tipo de datos métricos relacionados con el estado de las salas que componen el entorno de monitorización, mediante el uso de un *dashboard* (tablero). Es por ello, que esta aplicación esta basada en *Grafana*, un *software* libre de visualización de datos métricos utilizado para este tipo de tareas. Este *software* facilita la generación de consultas y permite según las necesidades del usuario la personalización de los distintos paneles que componen el tablero donde se mostrarán los resultados. De esta forma, cada panel puede interactuar con los diferentes tipos de bases de datos compatibles con *Grafana*, en este caso, se utiliza la base de datos de tipo *InfluxDB*. En la Figura 5.14, se muestra la creación de uno de estos tipos de paneles.

El *dashboard* principal se puede observar en la Figura 6.1 del Apartado 6.1.1, este ha sido diseñado para que aparezcan todos los datos relevantes sobre el estado del entorno. La configuración del *dashboard* utilizado ha sido aprovisionado (guardado) en un archivo con formato **YAML** (Listado D.1 del Anexo D), permitiendo así la carga automática del mismo cada vez que se inicie esta aplicación. Además, cabe destacar que la aplicación ha sido integrada en el sistema a través de un contenedor de *Docker*, debido a que la parte que corresponde al *Backend* del sistema también ha sido diseñado e implementada para integrarse en otro contenedor, como se presentará con un mayor nivel de detalle en la Sección 5.6.

5.5. APLICACIÓN DE DISTRIBUCIÓN ÓPTIMA DE RECURSOS DE LIMPIEZA

En la aplicación de distribución óptima de recursos, el actor principal, al igual que en la aplicación de monitorización del tránsito (Sección 5.4) es el *administrador*. El sistema desarrollado en esta aplicación está orientado a la elaboración de planes de desinfección basados en la distribución óptima de recursos, en este caso, un equipo humano de desinfección. Por lo tanto, el *administrador* puede iniciar la elaboración de un plan de desinfección en un momento determinado, en base a los recursos disponibles y las características del entorno que se esté monitorizando, como se representa en la Figura 5.15.

Por otro lado, en la Figura 5.16, se observa como la aplicación se encuentra dividida en diversos módulos. Además, se puede apreciar la importancia del módulo de comunicación, del cual dependen el resto de los módulos. Como se ha indicado con anterioridad, el módulo de comunicación es el

encargado de realizar la petición a la base de datos para poder obtener los datos necesarios para poder iniciar este subsistema. Una vez obtenidos los datos, el módulo de definición de prioridad es el encargado de dar cierta prioridad de desinfección a cada una de las salas que conforman el entorno, en base a una serie de parámetros considerados de gran relevancia. Una vez definida la prioridad de desinfección de las salas, se procede a la generación de planes de desinfección y la búsqueda del plan óptimo. Finalmente, los resultados serán representados en el **GUI** de la aplicación de escritorio.

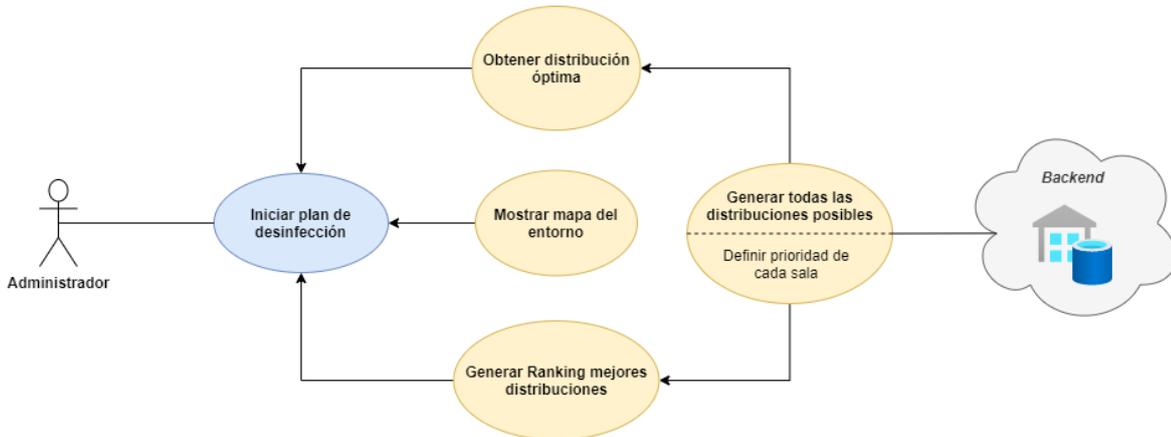


Figura 5.15: Diagrama de casos de uso de la aplicación de distribución óptima de recursos de limpieza.

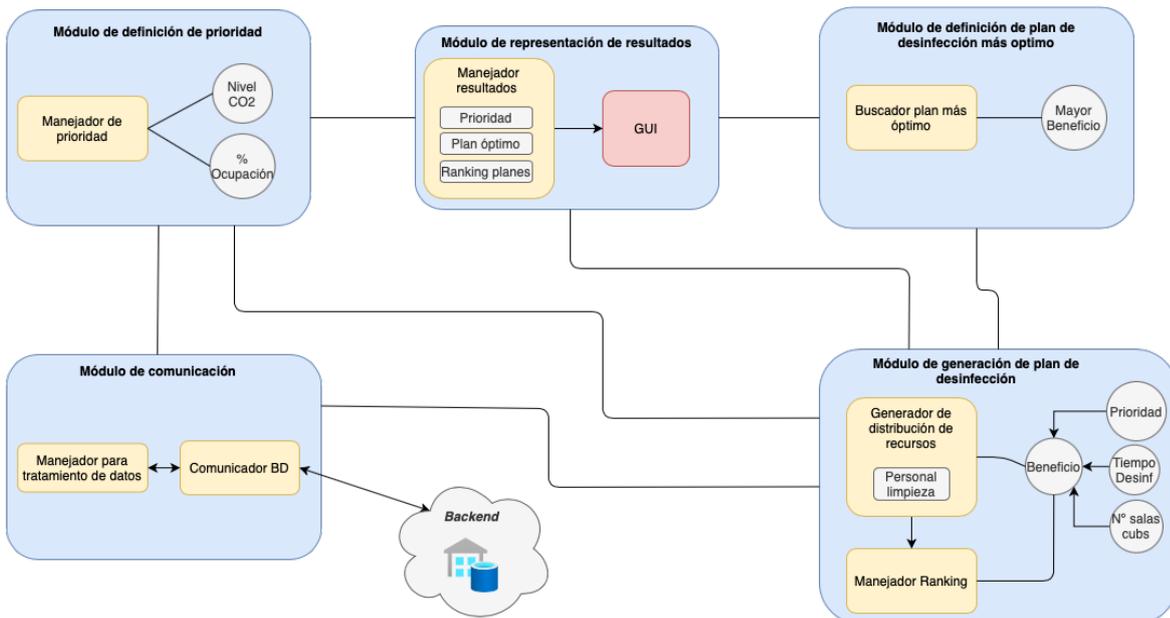


Figura 5.16: Esquema general aplicación de distribución óptima de recursos de limpieza.

5.5.1. Entradas y salidas

La aplicación de distribución de recursos recibe datos de diferentes entradas cuando interactúa con el entorno, posteriormente en base a dichos datos de entrada produce una serie de datos de salida. Por lo tanto, las entradas de la aplicación podrían categorizarse en los siguientes tipos:

- **Computador principal.** Los datos recibidos desde el computador principal serán las acciones realizadas por el usuario, en este caso el administrador, de la misma forma que ocurre en la aplicación de monitorización.

- **Datos del entorno de monitorización.** Los datos relacionados con el estado del entorno de monitorización, se considera uno de los flujos de información más importantes que recibe el sistema, estos datos son cargados de la base de datos del sistema, donde previamente han sido recopilados por parte de la red de nodos de sensorización **WSN**.

Una vez que el subsistema (aplicación de distribución de recursos) ha procesado los datos de las diferentes entradas ya mencionadas, produce una serie de salidas. Dichas salidas corresponden a los datos resultantes de la elaboración del plan de desinfección basados en la distribución óptima de recursos de limpieza, mostrándole al *administrador* el plan de desinfección más óptimo y las mejores alternativas a dicho plan calculadas por el subsistema. De este modo, se proporciona al *administrador* información relevante para llevar a cabo una distribución eficiente del equipo de limpieza (recursos disponibles).

5.5.2. Obtención y tratamiento de los datos

Los datos recopilados previamente mediante la red de sensorización WSN y registrados en la base de datos, serán obtenidos por la aplicación a través de la conexión con cliente de la base de datos (*InfluxDBClient*), de esta forma, se obtendrán una serie de datos de gran importancia sobre el estado de cada una de las salas por las que esta formada el entorno monitorización. Dichos datos están relacionados con la media del número de dispositivos WiFi y BLE que ha habido en la sala, valor del contador de ocupantes en ella y la media del nivel de CO_2 en la misma. En el Listado 5.6 se muestra un ejemplo del resultado obtenido fruto del establecimiento de esta conexión. En este caso, simplemente se muestra la estructura en formato **JSON** de los datos recibidos para una de las salas. Donde el campo *name* corresponde con el nombre de la tabla de la base de datos donde se encuentran almacenados, posteriormente se indica el identificador de la sala, en este caso la sala 1, y las columnas que conforman la tabla, como son el número de dispositivos BLE, el contador de personas dentro de la sala, el nivel de CO_2 y el número de dispositivos WiFi. Finalmente se muestran los valores para cada una de las columnas en el momento dado.

Listado 5.6: Resultado obtenido en formato JSON sobre el estado de una las salas que componen el entorno

```
1 {
2   "name": "states", "tags":
3     {"room_id": "1"},
4     "columns": ["ble", "co2", "people_counter", "wifi"],
5     "values": [{"2.0, 531.0, 5.0, 8.0}]
6 }
```

Una vez obtenidos los datos, se procede a un tratamiento de los mismos, ya que en base únicamente a los datos relacionados con el contador de ocupantes y el número de dispositivos WiFi/BLE no se puede obtener ningún tipo de información válida relacionada con la ocupación de cada sala. Por lo tanto, se ha realizado una estimación sobre estos datos, mediante una distribución de pesos para cada uno de ellos. Esto ofrece una nueva funcionalidad al sistema, ya que tiene la posibilidad de otorgar un mayor nivel de importancia a unos datos respecto de otros. De este modo, se logra la adaptación del sistema a diferentes tipos de escenarios, ya que dependiendo del entorno en el que se encuentre desplegado puede ser más relevante la información relacionada con la contabilización de ocupantes (realizada por *Door Node*), que la información relacionada con la cantidad de dispositivos WiFi/BLE (obtenida a través de *Scan Node*), o viceversa. Por ejemplo, este tipo de casos podrían darse en entornos más rurales donde la cantidad de dispositivos inteligentes compatibles con la tecnología BLE y WiFi no es muy abundante, o en escenarios más aislados donde tampoco se usan este tipo de tecnologías como las residencias de mayores. A continuación, se muestra la distribución de pesos mencionada.

$$eo = (co * p1) + (med_dw * p2) + (med_dw * p3) \quad (5.4)$$

Donde eo es la estimación de ocupación obtenida tras la aplicación de la distribución de pesos, co como contador de ocupantes, (med_dw, med_db) representa la media del número de dispositivos WiFi/BLE y $(p1, p2, p3)$ como los pesos aplicados. En este caso, los pesos pertenecen al intervalo $[0,1]$, de tal forma que la suma total de los mismos es igual a 1.

5.5.3. Definición de la prioridad de cada sala

La primera etapa del algoritmo es la definición dinámica de prioridad de cada sala (en la Figura 5.16 representado como Módulo de de definición de prioridad). En base a los datos relacionados con el nivel de CO_2 y el porcentaje de ocupación de cada sala se define la prioridad de la misma. El valor relacionado con el nivel de CO_2 de la sala ya ha sido previamente obtenido, pero el porcentaje de ocupación debe ser calculado en base a la estimación ya realizada (eo) y el tamaño de la sala correspondiente (indicado por el *administrador* del sistema). Para calcular el porcentaje de ocupación de cada sala, se debe obtener previamente el aforo máximo (am) o número máximo de ocupantes de la misma, para ello, se ha tenido en cuenta la distancia de seguridad interpersonal (dsi) en interiores de 2 metros, establecida por parte del *Ministerio de Sanidad*, siendo ésta equivalente a un área de $4m^2$ por ocupante.

$$am = \frac{ts}{dsi} \quad (5.5)$$

Una vez obtenido el valor correspondiente al aforo máximo de la sala, se calcula el porcentaje de ocupación (po), obteniéndose como resultado de la división entre la estimación de ocupación de la sala (eo) y aforo máximo (am).

$$po = \frac{eo}{am} * 100 \quad (5.6)$$

Tras calcular el porcentaje de ocupación de cada sala, se obtienen el valor máximo de nivel de CO_2 y del porcentaje de ocupación entre todas las salas que componen el entorno. Después se aplica una distribución de pesos al nivel de CO_2 y el porcentaje de ocupación de cada una de las salas, dividiéndose a la vez entre los valores máximos anteriormente calculados. De esta forma, se calcula el beneficio de cada una de las salas, a partir del cual se ordenarán las salas según el beneficio aportado por cada una de ellas.

$$bs = \frac{nco2 * pco2}{max_co2} + \frac{ocup * poc}{max_ocup} \quad (5.7)$$

Donde el beneficio de la sala está representado por bs , el nivel de CO_2 de la sala se representa como ($nco2$), su peso correspondiente como $pco2$, la ocupación de la sala como $ocup$, y el nivel máximo de CO_2 y la ocupación máxima como (max_co2, max_ocup).

En el Listado 5.7 se muestra el algoritmo desarrollado para llevar a cabo esta tarea. Donde se puede apreciar como se devuelve una lista ordenada en base a la prioridad calculada, y cada sala aparece representada por su identificador. Un ejemplo de salida de ejecución de esta parte del algoritmo sería: $[2,3,4,1]$, lo que indica el orden de prioridad de cada una de las salas de izquierda a derecha. Por lo tanto, a la sala número 2 se le otorgaría prioridad 1 (mayor prioridad), y así hasta la sala número 1 con prioridad 4 (menor prioridad).

Listado 5.7: Algoritmo para la definición de prioridad de cada sala

```

1 def create_list(self, rooms, co2_weight, occupation_weight):
2     plist = []
3     max_co2, max_occupation = get_max_values(rooms)
4
5     for room in rooms:
6         benefit = round(((room.co2_level * co2_weight) / ↵
7             ↵ max_co2) + ((room.occupation * ↵
8             ↵ occupation_weight) / max_occupation), 2)
9         plist.append(PriorityRoom(room.id, benefit))
10    plist.sort()
11
12    def gen_list_id_only():
13        id_list = []
14        for pr in plist:
15            id_list.append(pr.id)
16        return id_list
17
18    return gen_list_id_only()

```

5.5.4. Generación de planes de desinfección

La segunda etapa del algoritmo se centra en la distribución de los recursos disponibles, en este caso el personal del equipo de limpieza. De tal forma, que dichos recursos son asignados a cada una de las salas dependiendo de su prioridad, tiempo de desinfección y número de salas cubiertas de entre todas las existentes, lo que constituye un plan de desinfección. Tras la aplicación de una distribución de pesos a los parámetros anteriormente mencionados se obtiene un beneficio por cada una de las salas, el cual, varía dependiendo de la cantidad de recursos que se le asignen a cada una de ellas. Finalmente, los planes de desinfección son categorizados por el beneficio total obtenido que se calcula como el sumatorio de los beneficios obtenidos por cada una de las salas que componen dicho plan. Por lo tanto, el beneficio del plan de desinfección se obtiene a través del cálculo de la Función 5.8.

$$\sum_{i=1}^n (p_i + b_i) + \frac{sc}{n} \quad (5.8)$$

Donde p representa el beneficio obtenido por la prioridad de cada de la sala, b representa el beneficio obtenido por el tiempo de desinfección de dicha sala, sc representa el número de salas cubiertas, n que representa el número total de salas y cada i representa cada sala monitorizada.

En el Listado 5.8 se muestra la parte del algoritmo correspondiente a este apartado, donde se puede apreciar la existencia de dos tipos de funciones principales: *gen_priority_benefit()* y *gen_ct_benefit()*. La primera de ellas, se encarga del cálculo del beneficio a partir de la prioridad de la sala, mediante esta función se obtendría el valor representado por la p en la Función 5.8. La segunda de las funciones *gen_ct_benefit()* (representada en la Función 5.8 como b) se encarga de calcular el beneficio obtenido en base al tiempo de desinfección de la sala, según la cantidad de recursos que se le hayan asignado. Ambas funciones son llamadas dentro de un bucle *for* para así poder llevar a cabo el sumatorio, pasándole a cada función los argumentos requeridos. Una vez sumados los beneficios obtenidos por cada una de las salas, se calcula el beneficio proporcional a la cantidad de salas cubiertas en dicho plan, haciendo uso de la variable *rooms_count*, representada en la Función 5.8 como sc .

En la Figura 5.17, se muestra una captura de la aplicación desarrollada, donde para un entorno determinado se han generado distintos tipos de planes de desinfección en base a la distribución del personal de limpieza. Los planes son clasificados en un *ranking* según el beneficio obtenido por cada uno de ellos.

Listado 5.8: Algoritmo para la generación de planes de desinfección

```

1 def calculate_benefit(self, priority_list, max_rrs_room, ↵
  ↵ sizes_room, time_cl_m2, priority_weight, time_cl_weight):
2
3     rooms_count = 0
4     benefit = 0
5
6     for i in range(0, len(rrs_per_room)):
7
8         if self.rrs_per_room[i] != 0:
9             benefit += gen_ct_benefit(sizes_room[i], ↵
  ↵ rrs_per_room[i], max_rrs_room[i]) + ↵
  ↵ gen_priority_benefit(priority_list[i])
10
11             rooms_count += 1
12
13         benefit += (rooms_count / len(rrs_per_room)) * (1 - ↵
  ↵ (priority_weight + time_cl_weight))
14
15     return round(benefit, 3)

```

• Información del entorno

- Número personal de limpieza: 4

#	Sala	Tamaño	Ocupación %	Nivel CO2	Prioridad
	1	16	75 %	775	2
	2	46	36 %	601	3
	3	28	86 %	862	1

• Ranking de salas

Plan	Recursos por Sala:			Beneficio Obtenido
				
1				1,616
2				1,604
3				1,6
4				1,35
5				1,334

Figura 5.17: Planes de desinfección generados por la aplicación de distribución de recursos para un entorno determinado.

5.5.5. Definición del plan de desinfección más óptimo

La tercera y última parte del algoritmo tiene la finalidad de definir el plan de desinfección en base al beneficio ya mencionado. Para ello, todos los planes generados se han representado como una clase objeto llamada *solución*, compuesta por dos atributos: i) la distribución realizada en cada sala y ii) el beneficio generado por dicha solución (mencionado en el anterior apartado 5.5.4). De esta forma, todas las soluciones generadas se han almacenado en la lista denominada *other_solutions*. Sin embargo, antes de guardar la solución generada en dicha lista se ha comparado si el beneficio aportado por la solución es el mayor de todas las soluciones existentes, si no es así la solución acabará almacenada en dicha lista, si por el contrario, es la solución que más beneficio aporta en relación a las ya generadas, se almacena una copia de dicha solución en la variable *best_solution*. Finalmente, cuando se hayan generado todas las soluciones posibles, en la variable *best_solution* se encontrará almacenada la mejor solución, correspondiente al plan de desinfección óptimo y, por otra parte, en la

lista denominada *other_solution* se ha almacenado todas las soluciones alternativas que aportan un beneficio menor que la mejor solución. Todo este proceso se puede observar detalladamente en el Listado 5.9.

Listado 5.9: Algoritmo para la generación de planes de desinfección

```

1  for p in product(rr_numbers, repeat=len(rooms)):
2      solution = None
3
4      if sum(p) == len(rr_numbers)-1 and check_permutation(p):
5          solution = Solution(priority_list, p, max_rr_room, ←
6              ↪ sizes_room, time_cl_m2, priority_weight, ←
7              ↪ time_cl_weight)
8
9          if best_solution is None:
10             best_solution = copy.deepcopy(solution)
11
12         else:
13             if best_solution.benefit < solution.benefit:
14                 other_solutions.append(best_solution)
15                 best_solution = copy.deepcopy(solution)
16
17             else:
18                 sother_solutions.append(solution)

```

Cabe destacar, que el bucle de tipo *for* que aparece al comienzo del Listado 5.9 ha sido implementado para realizar todas las permutaciones posibles en base al número de recursos disponibles (personal de limpieza) que se podría asignar a cada sala. Para ello, dichas permutaciones deberían cumplir con una serie de condiciones, como que la suma total de la aplicación de las mismas no sobrepasase el número total de recursos disponible, o que el número de recursos disponibles destinados a la sala no fuese mayor o igual al número de ocupantes que haya habido en la misma, dado que, en este caso, se estarían desaprovechando recursos, los cuales podrían destinarse para la desinfección de otras salas.

En la Figura 5.17, aparece un ejemplo del resultado generado por este subsistema para un entorno determinado. En la tabla situada en la parte inferior se muestra un *ranking* con varios de los mejores planes de desinfección para ese entorno, donde el mejor plan de desinfección es el primero (resaltado en color verde), ya que ofrece el mejor beneficio en base a la distribución del personal de limpieza disponible. Por lo tanto, dicho plan representaría la mejor solución (*best_solution*) y los demás planes serían las soluciones alternativas a este (representadas anteriormente como *other_solutions*).

5.6. BACKEND

El sistema desarrollado en el presente trabajo tiene la necesidad de que los datos recopilados por parte de la red de sensorización de tipo WSN sean persistentes, y así poder acceder a ellos más adelante. Por esta razón, este tipo de sistemas que trabajan con esta clase de datos requieren de una arquitectura específica para poder llevar a cabo este cometido. Por lo tanto, se ha diseñado un subsistema encargado del almacenamiento de dichos datos, para su posterior uso por parte de las aplicaciones de monitorización del tránsito y de la distribución óptima de recursos.

Seguidamente, en la Figura 5.18 se muestra un diagrama general sobre el *Backend* del sistema. El Backend esta formado por una serie de servicios o aplicaciones que interactúan de manera conjunta, estas aplicaciones han sido desplegadas en forma de *contenedores* de *Docker*, como se presentará de forma detallada a continuación, en el apartado 5.6.2.

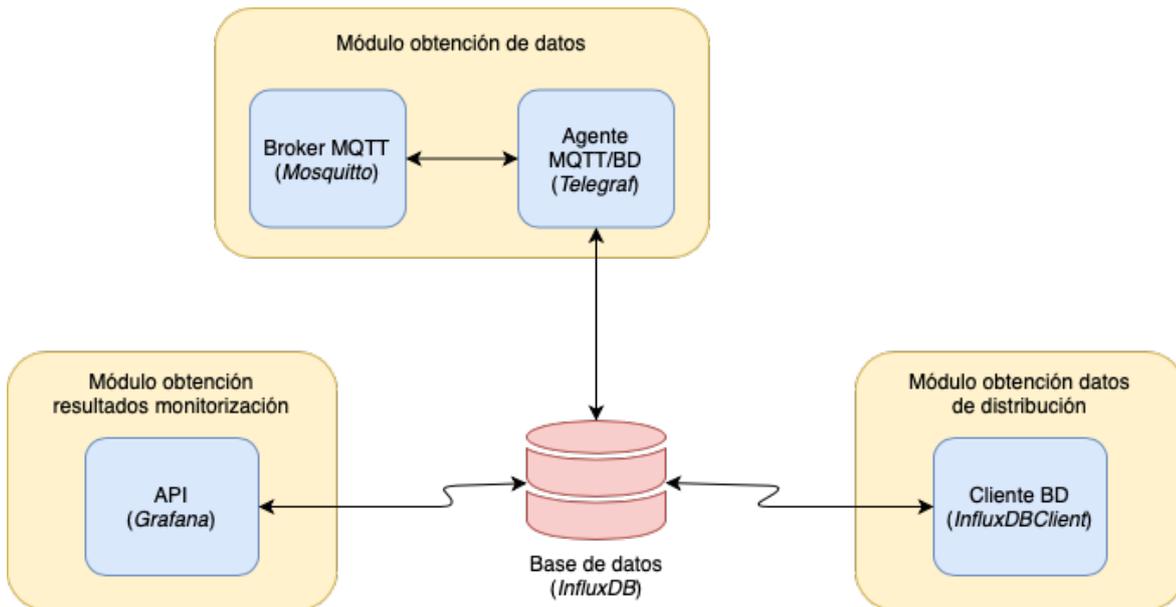


Figura 5.18: Esquema general *Backend* del sistema

5.6.1. Base de datos

Para la persistencia de los datos se ha diseñado una base de datos con *InfluxDB*, orientada al almacenamiento de series temporales. Una de las características de este proyecto es el envío periódico de datos, los cuales deben de ser almacenados. Por esta razón, se ha elegido este modelo de base de datos, optimizado para el almacenamiento rápido y eficiente de series temporales, ofreciendo una alta disponibilidad para la recuperación de este tipo de datos provenientes de sensores *IoT*, como es el caso del presente trabajo.

La base de datos *InfluxDB* es una base de datos de tipo relacional, que permite la relación entre las distintas tablas que la componen mediante el uso de clave primaria y foránea. Sin embargo, para este proyecto se ha diseñado una base de datos compuesta por una tabla principal, encargada del almacenamiento de los estados de cada sala. De esta forma, se favorece a la rapidez y eficiencia por parte del agente administrador de la base de datos, a la hora de almacenar y obtener los datos necesarios, con aras de ahorrar tiempo en el ordenamiento de los distintas claves primarias o foráneas y sus relaciones con las demás tablas. Por ello, se define una única clave primaria correspondiente al tiempo en el cual se ha almacenado dicho dato, siendo así más sencilla y eficaz su obtención más adelante.

La tabla principal de la base de datos diseñada observar se denomina *States*, en ella se almacena la información relativa al estado de las salas. Para ello, la tabla dispone de una serie de atributos, como *time* (clave primaria); *room_id*, identificador de la sala sobre la que se esta almacenando el estado; *people_counter*, dominio perteneciente a la contabilización de los ocupantes realizada por el sensor **PIR**; *disp_ble*, tiene como dominio el número de dispositivos **BLE** que se encuentre en el entorno; *disp_wifi*, dominio relacionado con el número de dispositivos **WiFi** que se encuentren en el entorno; y *co2_level*, cuyo dominio es el nivel e concentración de CO_2 en el ambiente (medido en *ppm*).

5.6.2. Contenedores

El uso y despliegue del sistema se ha realizado a partir de *Docker*. *Docker* es una plataforma de *software* libre que permite la creación de *contenedores*. Dichos contenedores incluyen todo lo necesario para la correcta ejecución del *software*, como herramientas, bibliotecas, etc. Aparte del uso de plataformas como *Docker* para estas funciones, también se pueden usar *software* de virtualización. Sin embargo, el soporte del sistema operativo de la máquina virtual y el de *Docker* es muy diferente (Ver Figura

5.19). Las máquinas virtuales requieren utilizar el *hipervisor*¹ o monitor de máquina virtual. En la Figura 5.19 (a), se muestra como el hipervisor se encuentra situado en una capa superior al sistema operativo. Sobre el hipervisor se instalará el sistema operativo completo correspondiente a cada una de las máquinas virtuales desplegadas, al igual, que sus respectivas herramientas y bibliotecas necesarias para el correcto funcionamiento de las aplicaciones iniciadas en cada sistema [4].

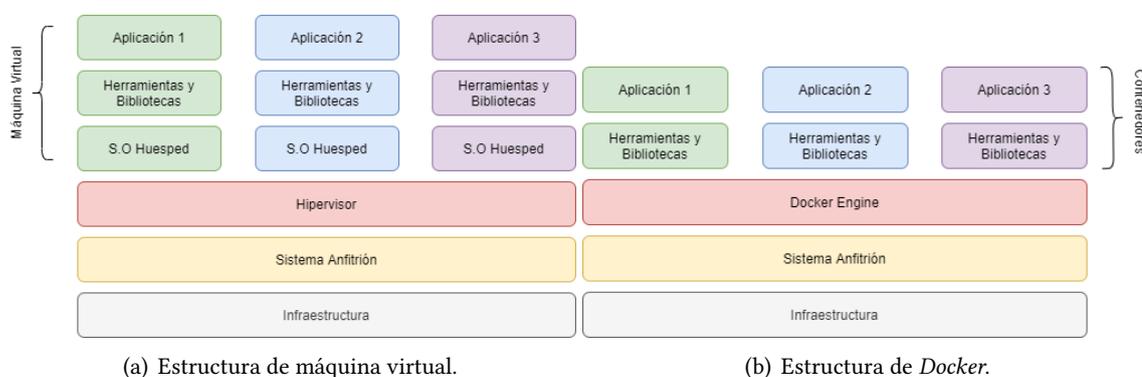


Figura 5.19: Comparación del soporte ofrecido por un despliegue de máquinas virtuales y de *Docker*.

Con la utilización de *Docker*, desaparece el sistema operativo huésped (de la máquina virtual) y el hipervisor (Figura 5.19 (b)), sustituyéndose ambos por *Docker Engine*. A simple vista puede parecer un cambio mínimo, pero las diferencias de optimización y eficiencia entre estos dos tipos de soportes son enormes. En este caso *Docker Engine* es el encargado del lanzamiento y de la gestión de los contenedores, compartiendo con cada contenedor los recursos *hardware* de los que dispone el sistema operativo anfitrión. De esta forma, se consigue un máximo nivel de optimización del uso de estos recursos, y permitiendo la interacción de las bibliotecas o herramientas (denominadas imágenes) entre las diversas aplicaciones que componen el sistema global.

Debido a las ventajas comentadas anteriormente, se ha optado por la utilización de *Docker*, creando así un contenedor por cada una de las partes que forman el *Backend*. Dichos contenedores contienen lo siguiente:

- **Mosquitto:** servicio que hace la función de *broker MQTT*, encargado de gestionar los canales de comunicación MQTT a partir de los cuales se comunica la red de sensorización con el sistema desarrollado.
- **Telegraf:** actúa como agente MQTT/BD, encargado de guardar los datos provenientes de los nodos de sensorización en la base de datos.
- **Grafana:** entorno de visualización de datos métricos, utilizado para mostrar los resultados obtenidos por la red de sensorización. Este *software* se mantiene conectado a la base de datos mediante una conexión a través de su propia API HTTP, encapsulada a nivel interno. Esta API se encarga de obtener de la base de datos del sistema los datos requeridos por parte de la aplicación a través de peticiones en forma de sentencias SQL.
- **InfluxDB:** base de datos encargada del almacenamiento persistente de los datos del sistema.

La creación y el posterior despliegue de estos contenedores puede realizarse mediante la introducción de una serie de comandos compatibles con *Docker*. Sin embargo, *Docker* ofrece su propia

¹El *hipervisor* es una capa *software* que permite la virtualización del *hardware*, y así poder hacer uso de diferentes sistemas operativos sobre un mismo computador

herramienta denominada *Docker Compose*, con la que se pueden definir y ejecutar múltiples aplicaciones (contenedores). Para ello es necesario utilizar un fichero de lenguaje de marcado **YAML**, para definir la configuración del sistema correspondiente. De tal forma, que con la ejecución de un solo comando se pueden crear e iniciar las aplicaciones que han sido configuradas en dicho fichero. Por lo tanto, en el fichero de configuración se pueden definir la imagen de cada uno de los contenedores a utilizar, volúmenes, exposición de puertos, etc. En la Sección **D.2** del Anexo **D**, se muestra un listado con el fichero de configuración correspondiente a las aplicaciones que forman el *Backend* del sistema desarrollado en el presente trabajo.

RESULTADOS

En este capítulo se mostrarán y analizarán los resultados obtenidos tras la aplicación del método de trabajo presentado en el Capítulo 4. Al principio se van a exponer los resultados obtenidos del desarrollo de cada aplicación y, posteriormente, los resultados obtenidos en la fase de experimentación del sistema global.

El objetivo de la fase de experimentación ha sido probar el sistema en un contexto realista en el que se lleve a cabo una monitorización del tránsito humano dentro de un entorno determinado. De esta forma, los ocupantes realizarán distintos tipos de actividades a lo largo del periodo de monitorización, transitando entre las distintas salas que componen dicho entorno. Tras un periodo de tiempo, se obtendrán la suficiente información sobre el estado en el que se encuentran las salas tras este trasiego. Seguidamente, en base a los datos obtenidos sobre el estado del entorno en la etapa de monitorización, se realizará por parte de la aplicación de elaboración automática de planes de desinfección una distribución óptima de los recursos de limpieza disponibles, en este caso, un equipo de limpieza (formado un número determinado de personas).

6.1. SISTEMA RESULTANTE

El sistema resultante del desarrollo de este proyecto se compone principalmente de dos aplicaciones, y una serie de subsistemas necesarios para el correcto funcionamiento de las mismas, como el despliegue de la red de nodos de sensorización de tipo WSN y el *Backend* del propio sistema, presentado previamente en el Capítulo 5. En los siguientes apartados se presentarán los resultados y funcionales ofrecidas por estas dos aplicaciones.

6.1.1. Aplicación de monitorización del tránsito humano

Esta aplicación ha sido diseñada con el uso de *Grafana*. La pantalla principal de esta aplicación corresponderá a un *dashboard* (tablero), compuesto por múltiples paneles, en los cuales, se mostrarán varios tipos diferentes de gráficos, junto a los valores en tiempo real de cada uno de los sensores desplegados en las salas. Dicha interfaz se puede ver representada en la Figura 6.1, donde podemos ver como por cada sala se muestra un gráfico histórico sobre los valores obtenidos por cada uno de los sensores y el valor en tiempo real de los mismos, como número de dispositivos BLE, número de dispositivos WiFi, nivel de concentración de CO_2 y número de detecciones de movimiento por parte del sensor PIR, en el cual, se podrá observar como el resultado puede ser incrementado o decrementado, dependiendo si el ocupante accede o sale de la sala. De esta forma se le ofrecerá el suficiente *feedback* al *administrador*, para así analizar por su parte el estado en el que se encuentran cada una de las salas que forman el entorno de monitorización.

En la Figura 6.1, se muestra la pantalla principal en forma de *dashboard* de la aplicación de monitorización de tránsito humano. En este caso se muestra toda la información relevante a la sala número 1

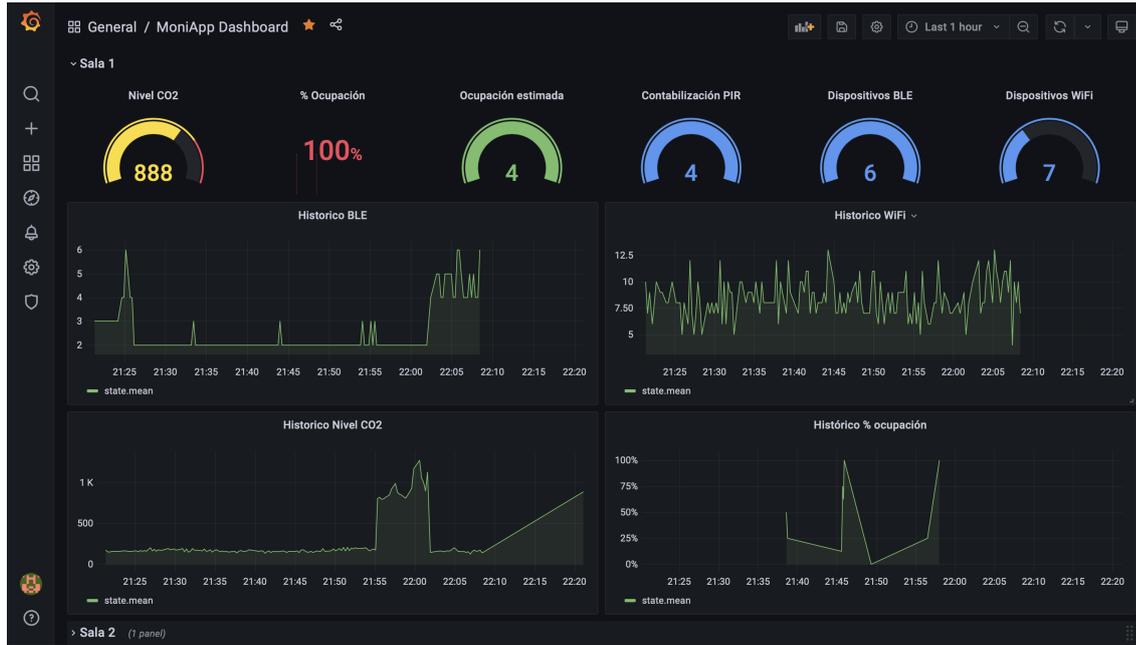
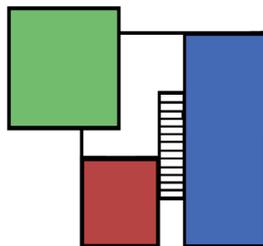
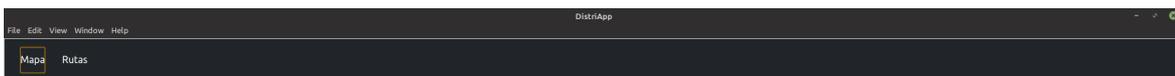


Figura 6.1: Pantalla principal aplicación de motorización.

6.1.2. Aplicación de distribución de recursos de limpieza

La aplicación de distribución de recursos se ha desarrollado en base a la arquitectura presentada en la Sección 5.5. La funcionalidad principal de la aplicación es la generación automática de planes de desinfección, para ello, el *administrador* simplemente debe de ejecutar la aplicación. Ésta, de forma automática, obtendrá la información sobre el estado de las salas en ese momento, y elaborará una serie de planes de desinfección, mostrando el plan óptimo en la pestaña principal junto a un mapa del entorno o la infraestructura que está monitorizando, como se muestra de forma detallada en la Figura 6.2. Esta aplicación también está compuesta por una segunda pestaña, conocida como *Rutas*, presentada (junto a la pestaña *Mapa*) con una mayor calidad de imagen en el apartado C.3 del Anexo B.



- Plan de desinfección
 - Número personal de limpieza: 4

#	Sala	Tamaño	Limite aforo	Aforo	Ocupación %	Nivel CO2	Prioridad	Recursos
1	1	17	4	4	75 %	775	2	
2	2	46	11	6	36 %	601	3	
3	3	29	7	6	86 %	862	1	

Figura 6.2: Pantalla principal de la aplicación de distribución de recursos

6.2. EXPERIMENTACIÓN

En esta sección se va a presentar la fase de experimentación del sistema desarrollado en el presente proyecto. Este proceso ha consistido en realizar una serie de pruebas en el interior del domicilio familiar, debido a las medidas especiales que se han adoptado a causa de la COVID-19. Por ello, en las pruebas han participado entre cuatro y seis personas pertenecientes al entorno familiar del autor del trabajo, de las cuales, se han guardado diversos datos considerados como relevantes para la realización de un estudio, de modo que permitan extraer conclusiones sobre el sistema desarrollado. Además, durante estas pruebas se ha realizado una comparación entre el medidor de sensor de CO_2 integrado en uno de los nodos desarrollados y un medidor comercial con fiabilidad de medición alta, con el objetivo de asegurar el correcto funcionamiento del sensor integrado en el nodo.

En los siguientes apartados se analizarán los resultados obtenidos, en base a varias pruebas sobre la utilización del sistema resultante. Por parte del apartado de monitorización, primero se va a medir la cantidad de dispositivos BLE/WiFi, junto a los valores de contabilización proporcionados por el sensor PIR. Con esto se pretende analizar la diferencia existente entre el número real de personas en la sala con los valores de los parámetros mencionados. Seguidamente, se analizará la fiabilidad demostrada por parte de la función de estimación calculada. Dicha función de estimación ha sido presentada en la Sección 5.5. Finalmente, se analizará la evolución del nivel de concentración de CO_2 en relación con el nivel de ocupación en la sala, a partir de los valores obtenidos por el sensor NDIR. Además, se realizará una comparación entre el prototipo desarrollado y un sensor comercial. Tras realizar las pruebas relacionadas con la aplicación de monitorización, se analizarán los resultados obtenidos por parte de la aplicación de distribución de recursos de limpieza, demostrando así si es coherente o no los resultados obtenidos. Para ello se realizará una evaluación sobre cómo el algoritmo de distribución otorga un mayor beneficio a un plan de desinfección respecto de otros, explicando así las distintas ventajas que puedan ofrecer.

6.2.1. Monitorización del tránsito humano

Para las pruebas realizadas en este apartado se ha tenido en cuenta la siguiente información acerca de la configuración del entorno de monitorización (en este caso el domicilio familiar).

- **Tamaño de cada sala (d):** este parámetro corresponde al área de la sala, calculado en metros cuadrados.
- **Aforo máximo de cada sala (am):** número de ocupantes máximo que pueden encontrarse de forma simultánea en una sala. Este dato ha sido calculado en base a la Función 6.2, compuesta por la distancia de seguridad interpersonal (ds) en interiores, actualmente de 2 metros, dando como resultado un área por ocupante de $4 m^2$. Por otra parte, se representa el tamaño de la sala (D).

$$am = \frac{d}{ds} \quad (6.1)$$

- **Cantidad de nodos en cada sala:** se debe remarcar que por cada sala se utilizan un único nodo de escaneo de tipo *Scan Node*, debido a que las señales BLE/WiFi operan en la banda de 2,4 GHz (802.11b/g/n), ofreciendo así un amplio rango de red (de escaneo). Además, se utiliza un único nodo de tipo *Door Node* por cada puerta que de acceso a dicha sala, mostrados en la Sección C.1

Por otro lado, los datos obtenidos en tiempo real fruto de la actividad de monitorización, son los siguientes:

- **Número real de personas:** indica el número real de personas que se encuentran en cada una de las salas. Con el uso de este dato se pretende poder obtener diferencias entre los datos capturados por el sensor.
- **Dispositivos BLE¹:** promedio de dispositivos compatibles con la tecnología BLE existentes en el entorno de monitorización, su cálculo se realiza a través del nodo de escaneo (*Scan Node*). Este parámetro se ha utilizado para, posteriormente, realizar una estimación de ocupación (explicada detalladamente en la sección Capítulo 5).
- **Dispositivos WiFi [1]:** promedio de dispositivos compatibles con la tecnología WiFi existentes en el entorno de monitorización, calculado con el uso del nodo de escaneo (*Scan Node*). Al igual, que el parámetro correspondiente a los dispositivos BLE, posteriormente se le aplicará una distribución de pesos para así realizar una estimación de la ocupación en la sala (como se describirá a continuación).
- **Detecciones de movimiento por parte del sensor PIR (Detecciones PIR):** número de ocupantes que acceden al área de monitorización (sala), con el uso de los sensores PIR integrados en el nodo instalado en la puerta de la sala (*Door Node*).
- **Estimación de ocupación (eo):** se realiza una estimación de la ocupación en base a una distribución de pesos entre el promedio de dispositivos BLE (*db*), promedio de dispositivos WiFi (*dw*) y detecciones PIR (*dp*). De tal forma, que al número de detecciones del sensor PIR se le ha otorgado un mayor peso (*pp*), ya que los resultados de este sensor son más fiables que los disp BLE (*pb*) y disp WiFi (*pw*) detectados. Tal y como se ha mencionado en la Sección 5.5, y representado a partir de la Función 6.2

$$eo = (dp * pp) + (db * pb) + (dw * pw) \quad (6.2)$$

- **Porcentaje de ocupación (po):** cálculo del porcentaje de ocupación que hay en un momento dado en la sala, de acuerdo a la Función 6.3. Esta función está compuesta por la estimación de ocupación (*eo*) y el aforo máximo (*am*). Al igual que la estimación de ocupación se ha presentado este cálculo de forma detallada en la Sección 5.5.

$$po = \frac{eo}{am} * 100 \quad (6.3)$$

- **Nivel de CO₂:** calidad del aire en la sala en base a la concentración de CO₂, medido en *ppm* (partes por millón). Este parámetro es calculado por *Scan Node* (presentado en la Sección 5.3)
- **Nivel de CO₂ comercial Selensy:** este parámetro indica la concentración de CO₂ en el aire de la sala (al igual que en el apartado anterior). Sin embargo, los valores representados por este parámetro han sido calculados con el uso de este sensor comercial. Con esto se pretende estudiar la diferencia entre los valores obtenidos a través del sensor desarrollado y el comercial. El sensor desarrollado será presentado en el Apartado 6.2.2.

Resultados obtenidos

Las pruebas se han desarrollado en un espacio formado por tres salas de con un tamaño de 17, 29 y 46 m² respectivamente.

En las Tablas 6.1, 6.2 y 6.3, se muestran los resultados obtenidos para la experimentación de este primer tipo de prueba, relacionada con la monitorización del entorno. Esta serie de tablas siguen un

¹Los dispositivos BLE y WiFi representan el cálculo de la media de dispositivos existentes en el entorno compatibles con cada una de estas tecnologías.

formato sencillo, donde dependiendo de la cantidad real de personas que se encuentren en cada una de las sala (en total tres), se reporta la cantidad de dispositivos BLE y WiFi en el entorno, número de detecciones PIR de la sala correspondiente, la estimación de ocupación realizada en base a los datos anteriores, el porcentaje de ocupación de la sala según la estimación previamente calculada, así como el nivel de concentración de CO_2 de la sala medido con el prototipo desarrollado (*Scan Node*) y con el medidor comercial *Selensy*.

Sala 1			Tamaño: 17 m ²			Aforo Máximo: 4 personas	
Personas	Disp BLE	Disp. WiFi	Detect PIR	Estim ocup	% ocupación	Nivel de CO ₂	Nivel CO ₂ comer
1	1	4	1	1	25	510	553
2	2	6	2	2	50	620	633
3	2	8	3	3	75	745	801
4	3	9	4	4	100	897	885
Coef. Correlación							0,97

Tabla 6.1: Resultados sala 1.

Sala 2			Tamaño: 46 m ²			Aforo Máximo: 11 personas	
Personas	Disp BLE	Disp. WiFi	Detect PIR	Estim ocup	% ocupación	Nivel de CO ₂	Nivel CO ₂ comer
1	1	5	1	1	9	310	332
2	2	7	2	3	18	389	411
3	3	7	3	3	27	421	429
4	3	9	4	4	36	553	513
5	4	10	5	5	45	611	598
6	4	10	6	6	55	734	714
Coef. Correlación							0,98

Tabla 6.2: Resultados sala 2.

Sala 3			Tamaño: 29 m ²			Aforo Máximo: 7 personas	
Personas	Disp BLE	Disp. WiFi	Detect PIR	Estim ocup	% ocupación	Nivel de CO ₂	Nivel CO ₂ comer
1	2	5	1	2	14	450	465
2	2	6	2	2	29	505	491
3	2	7	3	3	43	560	556
4	3	8	4	4	57	645	687
5	3	8	5	5	71	734	713
6	4	10	6	6	86	862	875
Coef. Correlación							0,98

Tabla 6.3: Resultados sala 3.

En estas tablas se puede observar como la cantidad de dispositivos WiFi detectados es bastante mayor que la cantidad de dispositivos BLE, tal y como se había comentado. En algunas ocasiones, como en la datos obtenidos con un número real de cuatro personas, se observa como la diferencia de dispositivos compatibles con la tecnología WiFi detectados (9) es entorno a un 44 %. Por esta razón, es necesario realizar una estimación de ocupación en base a la Función 6.2, otorgándole una mayor distribución de peso a los resultados obtenidos por parte de las detecciones del sensor PIR (0,7), un peso de 0,2 a los resultados relacionados con la cantidad de dispositivos BLE y finalmente un 0,1 a los resultados obtenidos sobre la cantidad de dispositivos WiFi.

En la Figura 6.3, se presenta un diagrama de barras comparativo sobre los resultados obtenidos a partir de la función de estimación 6.2, ofreciendo un alto grado de fiabilidad (en concreto un 93,75 % de acierto) con respecto al número real de personas que se encuentran en distintos instantes de tiempo en el interior de cada sala. En dicho diagrama se aprecia como el máximo de ocupantes de la Sala 1 son cuatro, como ya se ha comentado debido a las medidas COVID-19 para espacios cerrados.

Por otra parte, puede observar un error de estimación en el instante que se encuentra un único ocupante en la Sala 3, como se puede apreciar en la Tabla 6.3 esto se debe a que la cantidad de dispositivos WiFi en ese instante es un 80 % mayor respecto al número real de personas, además de la existencia de dos dispositivos BLE (50 % mayor respecto al número real de personas) dentro del

entorno de monitorización, lo que provoca una ligera variación (no crítica) en la estimación.

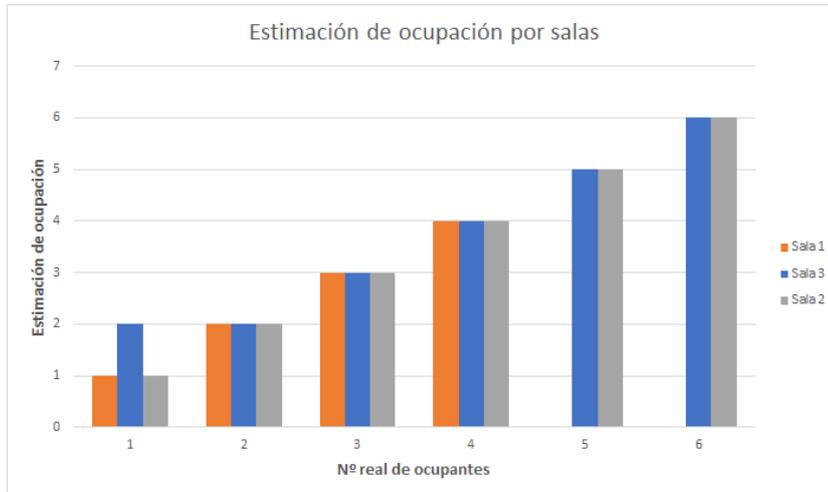


Figura 6.3: Resultados sobre la estimación de ocupación por cada sala.

En base a los resultados obtenidos por la estimación de ocupación se calcula el porcentaje de ocupación respecto al aforo máximo de la sala (ver Función 6.3). En las tablas anteriores, se muestra el porcentaje de ocupación de cada sala en función al número de ocupantes, apreciándose como dependiendo del tamaño de la sala, se llega a alcanzar el porcentaje máximo de ocupación. En esta prueba, debido a la cantidad de personas que han participado en ella (seis), y el tamaño de las salas, únicamente se ha alcanzado la ocupación máxima (según la medidas de distanciamiento en interiores por la COVID-19) en la sala 1.

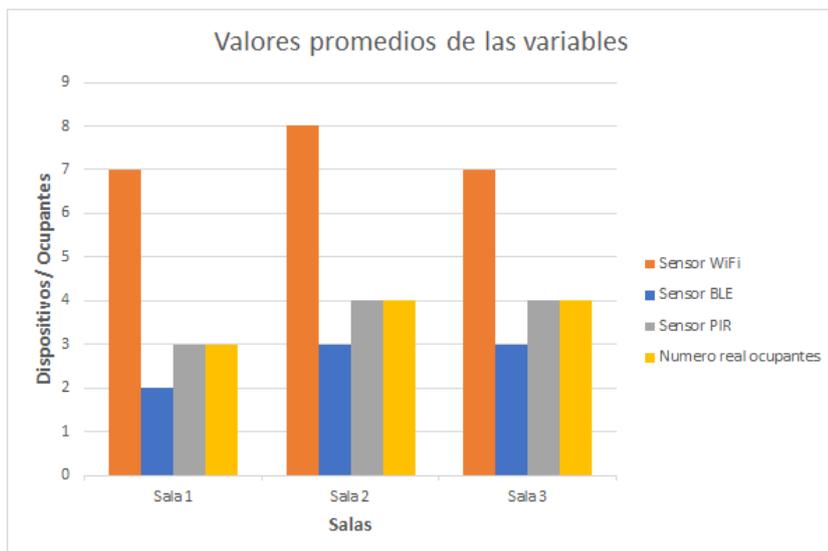


Figura 6.4: Comparativa entre los valores promedios de cada una de las variables obtenidas por los sensores y la ocupación real de cada sala.

En la Figura 6.4 se muestra el valor promedio perteneciente a las variables de los sensores en cada una de las salas del entorno de monitorización respecto a la ocupación real de las mismas. Se puede observar una diferencia mayor entre el promedio de dispositivos WiFi respecto al promedio del número real de ocupantes, que con el promedio de dispositivos BLE. En concreto, el promedio de dispositivos WiFi respecto al de ocupantes es alrededor de un 50 % mayor. Por el contrario, realizando una comparación sobre el promedio del número de ocupantes y el promedio de dispositivos BLE se

obtiene una diferencia del 25 % mayor por parte de los ocupantes, lo que demuestra que existe un alto porcentaje de variación entre la cantidad de dispositivos WiFi y el número de ocupantes, por esta razón se ha tomado la decisión de darle una importancia mayor a esta variable (dispositivos WiFi) respecto a otras. Además, se demuestra que los ocupantes realizan un mayor uso de los dispositivos WiFi que de los dispositivos BLE, obteniendo un porcentaje del 63,7 % del mayor uso de esta tecnología.

Por otro lado, en la Figura 6.4 se demuestra el alto grado de fiabilidad ofrecida por parte del sensor PIR, con un 0 % de error en base a la media calculada.

Por último, se ha realizado un análisis sobre los resultados obtenidos por el *Scan Node* con el uso del medidor de CO_2 . En la Figura 6.5, se puede apreciar la tendencia ascendente sobre el nivel de CO_2 de cada una de las salas, demostrando una clara relación entre el aumento de dicha medición (cuyo rango de valores ha sido definido en la Tabla 5.1) respecto al número de ocupantes de la misma.

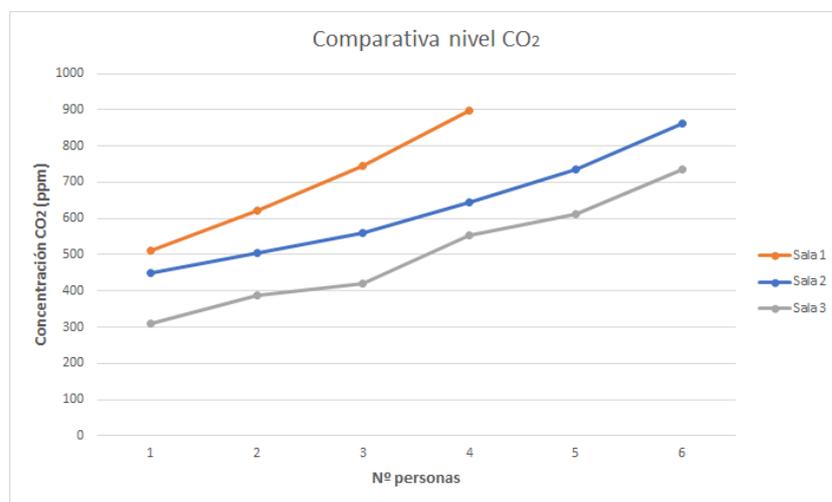


Figura 6.5: Resultados de CO_2 obtenido a partir del sensor NDIR.

6.2.2. Comparación prototipo de sensor CO_2 NDIR con medidor de CO_2 comercial

En este apartado se va a llevar a cabo la comparación entre el prototipo de sensor de CO_2 resultante en el desarrollo del presente proyecto, y un medidor de CO_2 comercial. Para ello, se debe tener en cuenta que el prototipo está basado en la tecnología NDIR [57], por lo que el medidor comercial debe ser compatible con ésta.



(a) Prototipo *Scan Node*.

(b) Medidor de CO_2 *Selensy*.

Figura 6.6: Nodo desarrollado encargado de la medición de CO_2 y el medidor comercial.

Tras analizar los diferentes sensores de control de calidad del aire en base a la concentración de CO_2 y compatibles con el uso de la tecnología NDIR actualmente disponibles en el mercado, se ha optado por el *Medidor de CO_2 portátil a Tiempo Real desarrollado por Selensy* (mostrado en la Figura 6.6 (b)), ya que es un sensor que se encuentra disponible a un precio asequible en el mercado, por lo que dispone de un alto porcentaje de ventas. Además, al igual que el prototipo desarrollado ofrece un rango de medición de entre 0 y 5000 ppm (partes por millón). En la Figura 6.6, se muestra el prototipo desarrollado en el proyecto (a) y el medidor comercial utilizado para esta comparación (b).

En la Figura 6.7 se muestra el diagrama de dispersión para la comparación de los valores sobre el nivel de CO_2 obtenidos a través del sensor NDIR de *Scan Node* y el medidor comercial *Selensy*. Se pudo observar como los niveles de CO_2 calculados por ambos tipos de medidores son bastante similares. Tras la aplicación del coeficiente de correlación de *Pearson*² a estas dos variables, se ha obtenido una correlación alrededor de 0,98 (ver Tablas 6.1, 6.2, 6.3), demostrando una correlación lineal directa (casi perfecta). Por lo tanto, se deduce en que los cálculos realizados con el uso de *Scan Node* sobre la concentración de CO_2 en el entorno ofrecen unos resultados fiables en relación a los ya ofrecidos por otro sensores destinados a este fin.

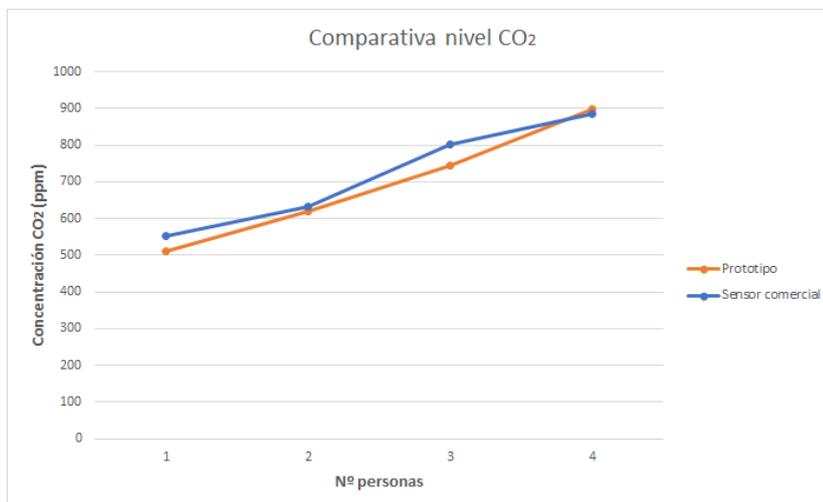


Figura 6.7: Comparación de resultados sobre el nivel de CO_2 obtenidos a través del sensor NDIR de *Scan Node* y del medidor comercial *Selensy*.

6.2.3. Distribución óptima de los recursos de limpieza

Para llevar a cabo este apartado de pruebas se ha tenido en cuenta los resultados obtenidos sobre la monitorización del entorno. Además, gran parte de la información sobre la configuración de cada sala es necesaria para llevar a cabo esta distribución. Estos datos son el **tamaño**, el **porcentaje de ocupación** y el **nivel de CO_2** (mencionados anteriormente en 6.2.1).

Dicha distribución se ha realizada en base a los siguientes de cálculos:

- **Prioridad:** prioridad de desinfección de cada sala. La prioridad se ha calculado siguiendo la serie de funciones explicadas en el apartado 5.5.3. Cuya función principal, es la Función 6.4, donde p representa la prioridad obtenida a partir de una distribución de pesos (p_{co2} , p_{oc}) entre el nivel de CO_2 de la sala (n_{co2}) y el porcentaje de ocupación de la misma (o_{cup}). Además, max_co2 y max_ocup representan los valores máximos obtenidos de entre todas las salas.

²El coeficiente de correlación de *Pearson* es una medida de dependencia lineal entre dos variables aleatorias cuantitativas bastante utilizado en estadística. De manera menos formal, se podría definir como un índice perteneciente al intervalo [0,1] utilizado para definir el nivel de relación existente entre dos variables cuantitativas y continuas.

$$p = \frac{ncO_2 * pco_2}{max_co_2} + \frac{ocup * poc}{max_ocup} \quad (6.4)$$

- **Beneficio:** este parámetro hace referencia al beneficio obtenido por cada plan de desinfección (basado en la distribución de recursos), para así una vez generados todos los planes definir el más óptimo, el cual, según su distribución haya obtenido un mayor beneficio según la aplicación. El beneficio obtenido ha sido calculando en base a una serie de funciones, que se pueden encontrar detalladas en el Apartado 5.5.4.

Resultados obtenidos

La prueba sobre una distribución óptima de los recursos de limpieza ha sido realizada entre las tres salas monitorizadas en el Apartado 6.2.1. En las Tablas 6.4 y 6.5, se muestran los resultados obtenidos por la aplicación de distribución, los cuales serán analizados a continuación. Cabe destacar, que los recursos disponibles son cuatro componentes del equipo de limpieza.

Sala	Tamaño (m ²)	Ocupación %	Nivel CO ₂	Prioridad	Recursos
1	17	100	693	2	1
2	46	55	503	3	2
3	29	86	626	1	2

Tabla 6.4: Resultados distribución más óptima.

En la Tabla 6.4, se puede apreciar el resultado de la distribución óptima calculada por el sistema, en otras palabras, el plan de desinfección óptimo. En este caso, el sistema ha evaluado la necesidades de desinfección de cada sala dependiendo del nivel de CO₂, la prioridad, el porcentaje de ocupación y el tamaño de la sala. De tal forma, que el plan que más beneficio ha sido el de destinar un personal de limpieza a la sala 1 (la más pequeña) y dos a cada una de las demás salas (2 y 3), consiguiendo así la desinfección de todas las salas que conforman el entorno. Por lo tanto, mientras uno de los componentes de limpieza desinfecta la sala 1, los demás componentes desinfectaran en un tiempo similar las otras dos salas.

Posición	Recursos - Sala			Beneficio
	Sala 1	Sala 2	Sala 3	
1	1	2	2	1,545
2	2	2	1	1,534
3	2	1	2	1,526
4	1	3	1	1,515
5	0	2	3	1,312
6	0	3	2	1,297
7	1	4	0	1,282
8	3	0	2	1,212
9	1	0	4	1,150
10	2	0	3	0,867

Tabla 6.5: Ranking de las diez mejores distribuciones de recursos.

En la Tabla 6.5 se muestra el ranking de las diez mejores distribuciones de recursos realizada por parte del sistema (posicionando en primer lugar la más óptima, mostrada anteriormente), indicándose la cantidad de recursos que han sido distribuidos por cada una de las salas. Cada una de estas configuraciones ha sido ordenada en el ranking en base al beneficio obtenido por cada una de ellas.

Se puede observar como los mejores planes de desinfección son aquellos donde el equipo de limpieza (recursos disponibles) cubre todas las salas, esto se debe a que el sistema otorga un mayor beneficio aquel plan donde la distribución de recursos cubre al completo todas las salas por las que está compuesto el entorno. Por lo tanto, el plan de desinfección más óptimo calculado demuestra que es la opción más adecuada de distribución de recursos, pudiéndose observar como se ha distribuido un mayor número de personal de limpieza en las salas con un alto nivel de prioridad y de mayor tamaño, surgiendo así la necesidad de asignar un mayor porcentaje de recursos a las mismas (en este caso un 40 % a cada una, dos componentes de los cinco totales), y así poder lograr la desinfección en el menor tiempo posible. También cabe destacar, como esta distribución de recursos cubre el 100 % del espacio de desinfección, aportando un mayor beneficio al plan de desinfección.

Por último, en el Anexo B se presenta el resultado de la aplicación de distribución de recursos en base a un escenario simulado que representa un contexto de mayor complejidad. Este escenario se encuentra formado por un total de seis salas, y dispone de un equipo humano de limpieza formado por nueve personas. Como se podrá observar se han obtenido resultados prometedores, los cuales, demuestran la adaptación de este sistema a cualquier tipo de entorno, en este caso, desde un entorno con un grado de complejidad menor como es el domicilio familiar del autor (en el cual han sido realizadas las distintas pruebas ya vistas), hasta entornos como el presentado en dicho anexo con un tamaño y grado de complejidad mayor.

CONCLUSIONES

Actualmente, nos encontramos en un contexto de pandemia provocado por la COVID-19, donde surge la necesidad de proteger y garantizar la salud de los ciudadanos tanto en los espacios públicos, como en los entornos más familiares. Para llevar a cabo esta labor y contener la propagación del virus, se debe acometer, con rigor, determinados preceptos de higienización y protocolos de seguridad sanitaria, especialmente en espacios cerrados en el interior de edificios. En infraestructuras públicas es común la existencia de espacios transitados donde es complicado llevar un control sobre las áreas con un mayor trasiego humano y ocupación, estas áreas son las que requieren ser desinfectadas prioritariamente y con mayor frecuencia.

En base a esta problemática, en este TFG se ha diseñado y desarrollado una solución que permite la monitorización del nivel de ocupación en salas, análisis de la calidad del aire y la elaboración de planes de desinfección con la distribución eficiente de recursos. En concreto, se han desarrollado dos tipos diferentes de aplicaciones. La primera orientada a labores de supervisión de este tipo de espacios, basada en la monitorización del tránsito humano dentro del entorno, para ello ha sido necesario el despliegue de una red de nodos sensorización de tipo **WSN** para la recopilación de dicha información. La segunda aplicación busca la distribución óptima de recursos de limpieza, siendo en este trabajo recursos humanos responsables de la desinfección de salas.

En el capítulo de Resultados se ha demostrado una fiabilidad alta del sistema desarrollado, siendo el sistema capaz de llevar a cabo una monitorización correcta del entorno, al igual de una estimación de ocupación bastante precisa, con un porcentaje de error por debajo del 5 %, según el experimento realizado, el cual, se ha llevado a cabo en un entorno compuesto por tres salas, con la participación total de seis ocupantes. Estos ocupantes disponían de un gran número de dispositivos compatibles con el uso de la tecnología WiFi, entorno a un 60 % mayor que el uso de tecnología BLE. En cuanto a la medición de la calidad del aire, se ha demostrado una clara relación respecto al porcentaje de ocupación, demostrando así que cuanto mayor es el porcentaje de ocupación mayor concentración de CO_2 hay en el entorno, y viceversa. Durante la realización de estas mediciones, también se ha comparado los valores sobre la concentración de CO_2 obtenidos por el sistema con los valores obtenidos por parte de un medidor de CO_2 comercial que hace uso del mismo tipo de tecnología (basado en NDIR), mostrando por parte de ambos dispositivos resultados muy similares, demostrando por tanto una medición correcta por parte del prototipo desarrollado. Además, una vez recopilada toda la información relacionada con el estado del entorno, el sistema es capaz de realizar una distribución adecuada de los recursos, tal como demuestran también los resultados obtenidos. En las situaciones sobre las que se han elaborado los planes, la solución elegida por el algoritmo diseñado, trata de distribuir personal de limpieza entre las salas que han sido ocupadas por un mayor número de personas, y las que tienen mayor prioridad (peor calidad del aire y mayor índice de ocupación). Esta serie de razones convierte el sistema propuesto, en una posible solución a parte de los problemas con los que convive la sociedad en la actualidad.

A lo largo del desarrollo del presente proyecto, se han aplicado gran parte de los conocimientos

adquiridos a lo largo del grado de Ingeniería Informática, relacionados con el diseño y la implementación de las distintas partes que forman el sistema, al igual que en la gestión del proyecto. Las habilidades técnicas que más se han aplicado han sido adquiridas en la intensificación de ingeniería de computadores, presentadas a continuación en la Sección 7.2. Los conocimientos adquiridos a lo largo de mi etapa en el grado, y en especial los relacionados con la intensificación de ingeniería de computadores, me han ayudado a afrontar los retos surgidos durante el presente proyecto. Por lo tanto, para el desarrollo de este trabajo he adquirido nuevos conocimientos sobre el despliegue de redes de tipo WSN, el diseño y creación de prototipos *hardware* y el manejo de datos.

7.1. OBJETIVOS ALCANZADOS

A modo de conclusión es importante recalcar que se han alcanzado con éxito todos los objetivos planteados inicialmente en este trabajo. A continuación, se describe brevemente el modo en el que se ha cumplido con cada uno de ellos:

Estudio y evaluación de dispositivos *hardware* para la monitorización de salas

Al comienzo del proyecto, una vez especificados los requisitos, se realizó un estudio sobre los diferentes dispositivos *hardware* orientados a la monitorización de ocupación, concluyendo que la mejor solución para esta tarea es el despliegue de una red de nodos de sensorización, siendo el resultado de esta la combinación de varios tipos de dispositivos como son el SoC ESP32-WROOM-32, el sensor PIR para la detección de movimiento y el sensor de CO₂ NDIR para medir la concentración de CO₂ y la calidad del aire del entorno.

Despliegue de una red de sensores para la adquisición de datos

Uno de los objetivos principales de este proyecto era el despliegue de una red de sensores para la adquisición de datos relacionados con el estado del entorno, como el porcentaje de ocupación y la calidad del aire de cada uno de los áreas que lo conformasen. Esto se ha conseguido con el desarrollo de una red de nodos WSN, con el uso de los nodos *Door Node* y *Scan Node* (ver Sección C.1 del Anexo C).

Análisis de ocupación y calidad del aire en el entorno

A partir de los datos procedentes de la red de sensorización que ha sido desplegada, se ha llevado a cabo un análisis sobre la densidad de ocupación y la calidad del aire del entorno. Este análisis es de vital importancia para que el *administrador* del sistema conozca el estado en el que se encuentra el entorno que está monitorizando. Para ello, se han mostrado diferentes tipos de resultados en forma de gráficos y métricas en tiempo real a través del uso de la aplicación de monitorización desarrollada en este proyecto, presentada en la Sección 5.4.

Elaboración automática de planes de desinfección

Para satisfacer este objetivo se ha diseñado e implementado una aplicación para la elaboración automática de planes de desinfección (ver Sección 5.5) basada en la distribución óptima de los recursos de limpieza disponibles. En este caso, los recursos de limpieza disponibles, es un equipo humano de limpieza. Por lo tanto, la aplicación devuelve la distribución más óptima de los componentes del equipo por las salas o áreas que componen el entorno de monitorización.

Desarrollo de interfaces gráficas de usuario para la visualización de los resultados de monitorización y planes elaborados

El *administrador* debe ser consciente del estado en todo momento del entorno que está siendo monitorizado, recibiendo así el mayor *feedback* posible. Para ello, se ha desarrollado una interfaz web que muestra los datos de la red de nodos desplegada en el entorno, como el valor actual de los sensores de cada nodo, estadísticas y gráficos relacionados con dichos valores previamente obtenidos, presentado de forma detallada en la Sección 5.4, y su funcionamiento y resultados obtenidos en el Apartado 6.1.1. Por otro lado, se ha desarrollado una interfaz de escritorio para la aplicación de distribución de recursos de limpieza (presentada en la Sección 5.5), donde se puede visualizar de una forma intuitiva y detallada el estado de cada sala, y el plan de desinfección óptimo (en base a dicha distribución), junto a un mapa del entorno, donde se representan los límites de cada una de las salas que lo componen. El funcionamiento de esta última aplicación y los resultados se encuentran comentados en el Apartado 6.1.2.

7.2. JUSTIFICACIÓN DE COMPETENCIAS ADQUIRIDAS

IC1: [*Capacidad de diseñar y construir sistemas digitales, incluyendo computadores, sistemas basados en microprocesador y sistemas de comunicaciones*]. Para el despliegue de la red de nodos de sensorización de tipo WSN, se han diseñado tres prototipos *hardware* de cada tipo de nodo, y así a partir de ellos recopilar información sobre el estado del entorno. Además, se ha estudiado la colocación más adecuada dentro del entorno para este tipo de nodos y sus componentes.

IC4: [*Capacidad de diseñar e implementar software de sistema y de comunicaciones*]. El sistema resultante se encuentra compuesto por varios subsistemas, donde cada uno de ellos representa un parte *software* del sistema global, ofreciendo cada uno de ellos una serie funcionalidades específicas. Por otro lado, para el correcto funcionamiento de este sistema ha sido necesaria la utilización del protocolo de comunicación MQTT, para la comunicación en tiempo real entre la red de nodos WSN y el *Backend* del sistema. Seguidamente, la aplicación de monitorización y la aplicación de distribución se comunican con el *Backend* para obtener los datos necesarios.

IC5: [*Capacidad de analizar, evaluar y seleccionar las plataformas hardware y software más adecuadas para el soporte de aplicaciones empotradas y de tiempo real*]. Para poder llevar a cabo una sensorización del entorno correspondiente en tiempo real, se requería del uso de SoC que incluyesen las especificaciones necesarias para poder llevar a cabo esta tarea. Finalmente se ha elegido el SoC ESP32-WROOM-32, debido a que disponía de un esquema de pines (pinout) compatibles con los sensores utilizados, y de los transceptores WiFi/BLE necesarios para la detección de dispositivos inteligentes, además de ser necesario el transceptor WiFi para establecer la comunicación con el resto de subsistemas.

7.3. TRABAJO FUTURO

Los objetivos propuestos, han sido planteados teniendo en cuenta el alcance y las limitaciones que tiene que tener un trabajo de esta categoría (TFG), dejando ciertas aplicaciones finales para esfuerzos futuros.

Las líneas de trabajo futuro más destacables relacionadas con el presente proyecto son las siguientes:

- Realizar una mejora de los prototipos de nodos diseñados, omitiendo el uso de placas de desarrollo y protoboards, e integrando tanto SoC como los sensores necesarios para cada uno de

los nodos en una única PCB. Esto le otorgaría a cada nodo un tamaño más compacto, menos visible a los individuos que transitan por las salas.

- Mejorar el algoritmo de captación de señales BLE/WiFi, ya que esta versión no diferencia la distancia a la que se encuentra el dispositivo detectado, tal que se encuentra con problemas para averiguar si el dispositivo se encuentra en la sala monitorizada o en otra adyacente. Esto se puede realizar si se consigue calcular el RSSI de cada paquete capturado (señal de dispositivo), a través del cual se podría realizar una estimación lo suficientemente fiable como para averiguar si el dispositivo detectado se encuentra dentro de la sala monitorizada por dicho nodo o en una sala adyacente a la misma.
- Sobre el protocolo de comunicación, se ha desplegado un *broker* MQTT con el uso de un contenedor de *Docker*, todo ello integrado en el computador principal. Sin embargo, cualquier usuario puede acceder al puerto 1883 habilitado para este servicio, y realizar cualquier acción de tipo publicación/ suscripción a alguno de los *topic* (canales MQTT), al no estar cifrado. Para ello, se propone añadir mayor seguridad a dicho *broker* MQTT, siendo necesario realizar algún tipo de autenticación para acceder al mismo, como puede ser el uso de una conexión de tipo SSL.

BIBLIOGRAFÍA

- [1] M. Y. Z. Abouleish, «Indoor air quality and coronavirus disease (COVID-19)», *Public Health*, 2020.
- [2] S. Akiyama y col., «Estimation of Walking Direction with Vibration Sensor based on Piezoelectric Device», en *2020 IEEE International Conference on Pervasive Computing and Communications Workshops (PerCom Workshops)*, IEEE, 2020, págs. 1-6.
- [3] R. W. Astheimer y F. Schwarz, «Thermal imaging using pyroelectric detectors», *Applied optics*, vol. 7, n.º 9, págs. 1687-1695, 1968.
- [4] D. Bernstein, «Containers and cloud: From lxc to docker to kubernetes», *IEEE Cloud Computing*, vol. 1, n.º 3, págs. 81-84, 2014.
- [5] B. Blocken y col., «Ventilation and air cleaning to limit aerosol particle concentrations in a gym during the COVID-19 pandemic», *Building and Environment*, vol. 193, pág. 107 659, 2021.
- [6] S. D. Brewster, *Observations on the pyro-electricity of minerals*. William Blackwood, 1824.
- [7] E. Bromage, «The Risks-Know Them-Avoid Them», *Erin Bromage: COVID-19 Musings blog*, 2020.
- [8] G. Caire y S. Shamai, «On the capacity of some channels with channel state information», *IEEE Transactions on Information Theory*, vol. 45, n.º 6, págs. 2007-2019, 1999.
- [9] V. r. Candanedo Luis M y Feldheim, «Detección precisa de ocupación de una habitación de oficina a partir de mediciones de luz, temperatura, humedad y CO2 utilizando modelos de aprendizaje estadístico», *Energy and Buildings*, Elsevier, ed.,
- [10] P. R. Center, «Mobile fact sheet», *Pew Research Center: Internet, Science & Tech*, 2018.
- [11] N. Charlotte, «High rate of SARS-CoV-2 transmission due to choir practice in France at the beginning of the COVID-19 pandemic», *Journal of Voice*, 2020.
- [12] H. Choi y col., «Classical and alternative disinfection strategies to control the COVID-19 virus in healthcare facilities: a review», *Environmental Chemistry Letters*, págs. 1-7, 2021.
- [13] R. N. Dave, «Characterization and detection of noise in clustering», *Pattern Recognition Letters*, vol. 12, n.º 11, págs. 657-664, 1991.
- [14] M. De Sanctis y col., «Lte signals for device-free crowd density estimation through csi secant set and svd», *IEEE Access*, vol. 7, págs. 159 943-159 951, 2019.
- [15] S. Depatla e Y. Mostofi, «Crowd counting through walls using WiFi», en *2018 IEEE International Conference on Pervasive Computing and Communications (PerCom)*, IEEE, 2018, págs. 1-10.
- [16] S. Di Domenico y col., «Trained-once device-free crowd counting and occupancy estimation using WiFi: A Doppler spectrum based approach», en *2016 IEEE 12th International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob)*, IEEE, 2016, págs. 1-8.
- [17] V. L. Erickson, M. Á. Carreira-Perpiñán y A. E. Cerpa, «OBSERVE: Occupancy-based system for efficient reduction of HVAC energy», en *Proceedings of the 10th ACM/IEEE international conference on information processing in sensor networks*, IEEE, 2011, págs. 258-269.
- [18] J. Esparza, «Epidemias y pandemias virales emergentes: ¿Cuál será la próxima?», *Investigación Clínica*, vol. 57, págs. 231-235, sep. de 2016, ISSN: 0535-5133. dirección: http://ve.scielo.org/scielo.php?script=sci_arttext&pid=S0535-51332016000300001&nrm=iso.

- [19] J.-S. Fang y col., «Path-dependent human identification using a pyroelectric infrared sensor and Fresnel lens arrays», *Optics express*, vol. 14, n.º 2, págs. 609-624, 2006.
- [20] J.-S. Fang y col., «Real-time human identification using a pyroelectric infrared detector array and hidden Markov models», *Optics express*, vol. 14, n.º 15, págs. 6643-6658, 2006.
- [21] R. Faragher y R. Harle, «Location fingerprinting with bluetooth low energy beacons», *IEEE journal on Selected Areas in Communications*, vol. 33, n.º 11, págs. 2418-2428, 2015.
- [22] A. Filippoupolitis, W. Oliff y G. Loukas, «Bluetooth low energy based occupancy detection for emergency management», en *2016 15th International Conference on Ubiquitous Computing and Communications and 2016 International Symposium on Cyberspace and Security (IUCC-CSS)*, IEEE, 2016, págs. 31-38.
- [23] D. Garcia-Romeo y col., «A NDIR-based CO2 monitor system for wireless sensor networks», en *2012 IEEE 3rd Latin American Symposium on Circuits and Systems (LASCAS)*, IEEE, 2012, págs. 1-4.
- [24] E. H. García, «Responsabilidad penal por imprudencia en la celebración de grandes eventos: un análisis de algunas cuestiones problemáticas en el caso Madrid Arena», *InDret*, 2017.
- [25] Z.-D. Guo y col., «Aerosol and surface distribution of severe acute respiratory syndrome coronavirus 2 in hospital wards, Wuhan, China, 2020», *Emerging infectious diseases*, vol. 26, n.º 7, pág. 1586, 2020.
- [26] K. Han y J. Zhang, «Energy-saving building system integration with a smart and low-cost sensing/control network for sustainable and healthy living environments: Demonstration case study», *Energy and Buildings*, vol. 214, pág. 109 861, 2020.
- [27] N. Hussain y col., «CDES: A pixel-based crowd density estimation system for Masjid al-Haram», *Safety Science*, vol. 49, n.º 6, págs. 824-833, 2011.
- [28] I. C. II, «Guía para la evaluación de la gravedad de las epidemias y pandemias de gripe en España», *Centro Nacional de Epidemiología*, 2019.
- [29] H. Joe y col., «Output-oriented power saving mode for mobile devices», *Future Generation Computer Systems*, vol. 72, págs. 49-64, 2017.
- [30] Y. Kashimoto y col., «Floor vibration type estimation with piezo sensor toward indoor positioning system», en *2016 International Conference on Indoor Positioning and Indoor Navigation (IPIN)*, IEEE, 2016, págs. 1-6.
- [31] M. F. Krafft, *The Debian system: concepts and techniques*. No Starch Press, 2005.
- [32] T. Kumar y D. S. Kushwaha, «An approach for traffic congestion detection and traffic control system», en *Information and communication technology for competitive strategies*, Springer, 2019, págs. 99-108.
- [33] U. Kumar y B. Yadav, «Synthesis of carbon nanotubes by direct liquid injection chemical vapor deposition method and its relevance for developing an ultra-sensitive room temperature based CO2 sensor», *Journal of the Taiwan Institute of Chemical Engineers*, vol. 96, págs. 652-663, 2019.
- [34] J. Kwon y col., «A study on NDIR-based CO2 sensor to apply remote air quality monitoring system», en *2009 Iccas-Sice*, IEEE, 2009, págs. 1683-1687.
- [35] K. Li y col., «Understanding crowd density with a smartphone sensing system», en *2018 IEEE 4th World Forum on Internet of Things (WF-IoT)*, IEEE, 2018, págs. 517-522.
- [36] D. D. Lin y T. J. Lim, «The variational inference approach to joint data detection and phase noise estimation in OFDM», *IEEE Transactions on Signal Processing*, vol. 55, n.º 5, págs. 1862-1874, 2007.
- [37] L. Logan, C. Davids y C. Davids, «Determining the Indoor Location of an Emergency Caller in a Multi-story Building», en *2020 IEEE International Workshop Technical Committee on Communications Quality and Reliability (CQR)*, IEEE, 2020, págs. 1-6.
- [38] M. Lotfi, M. R. Hamblin y N. Rezaei, «COVID-19: Transmission, prevention, and potential therapeutic opportunities», *Clinica Chimica Acta*, vol. 508, págs. 254-266, 2020, ISSN: 0009-8981. DOI: <https://doi.org/10.1016/j.cca.2020.05.044>. dirección: <https://www.sciencedirect.com/science/article/pii/S0009898120302503>.

- [39] R. P. Mahler, «Multitarget Bayes filtering via first-order multitarget moments», *IEEE Transactions on Aerospace and Electronic systems*, vol. 39, n.º 4, págs. 1152-1178, 2003.
- [40] D. Marín-García, J. J. Moyano-Campos y J. D. Bienvenido-Huertas, «Distances of transmission risk of COVID-19 inside dwellings and evaluation of the effectiveness of reciprocal proximity warning sounds», *Indoor air*, vol. 31, n.º 2, págs. 335-347, 2021.
- [41] L. Morawska, «Droplet fate in indoor environments, or can we prevent the spread of infection?», *Indoor air*, vol. 16, n.º 5, págs. 335-347, 2006.
- [42] F. Moreno-Sánchez y col., «Las grandes epidemias que cambiaron al mundo», *Anales Médicos de la Asociación Médica del Centro Médico ABC*, vol. 63, n.º 2, págs. 151-156, 2018.
- [43] M. Moussa y M. Youssef, «Smart cevices for smart environments: Device-free passive detection in real environments», en *2009 IEEE International Conference on Pervasive Computing and Communications*, IEEE, 2009, págs. 1-6.
- [44] H. Mu y col., «Human Indoor Positioning via Passive Spectrum Monitoring», en *2021 55th Annual Conference on Information Sciences and Systems (CISS)*, IEEE, 2021, págs. 1-6.
- [45] P. Muralt, «Micromachined infrared detectors based on pyroelectric thin films», *Reports on progress in physics*, vol. 64, n.º 10, pág. 1339, 2001.
- [46] Y. Peng y col., «Occupancy learning-based demand-driven cooling control for office spaces», *Building and Environment*, vol. 122, págs. 145-160, 2017.
- [47] F. Pérez y J. Maudos, «Impacto económico del coronavirus en el PIB y el empleo de la economía española y valenciana», *Valencia: Instituto Valenciano de Investigaciones Económicas (Ivie). Documento disponible aquí*, 2020.
- [48] O. P. C. Priyanka, I. Singh y G. Patra, «Aerosol transmission of SARS-CoV-2: The unresolved paradox», *Travel Medicine and Infectious Disease*, 2020.
- [49] H. Qian y col., «Indoor transmission of SARS-CoV-2», *Indoor Air*, vol. 31, n.º 3, págs. 639-645, 2021.
- [50] Z.-H. Qian e Y.-j. Wang, «IoT technology and application», *Acta Electronica Sinica*, vol. 40, n.º 5, págs. 1023-1029, 2012.
- [51] J. P. Roselyn y col., «Development and implementation of novel sensor fusion algorithm for occupancy detection and automation in energy efficient buildings», *Sustainable Cities and Society*, vol. 44, págs. 85-98, 2019.
- [52] A. J. Ruiz-Ruiz y col., «Analysis methods for extracting knowledge from large-scale wifi monitoring to inform building facility planning», en *2014 IEEE International Conference on Pervasive Computing and Communications (PerCom)*, IEEE, 2014, págs. 130-138.
- [53] K. C. Sahoo y U. C. Pati, «IoT based intrusion detection system using PIR sensor», en *2017 2nd IEEE International Conference on Recent Trends in Electronics, Information & Communication Technology (RTEICT)*, IEEE, 2017, págs. 1641-1645.
- [54] L. Schauer, M. Werner y P. Marcus, «Estimating crowd densities and pedestrian flows using wi-fi and bluetooth», en *Proceedings of the 11th International Conference on Mobile and Ubiquitous Systems: Computing, Networking and Services*, 2014, págs. 171-177.
- [55] H. Sidenbladh, «Multi-target particle filtering for the probability hypothesis density», *arXiv preprint cs/0303018*, 2003.
- [56] S. Sigg y col., «Passive, device-free recognition on your mobile phone: tools, features and a case study», en *International Conference on Mobile and Ubiquitous Systems: Computing, Networking, and Services*, Springer, 2013, págs. 435-446.
- [57] M. G. Srabanti, «Design of a non-dispersive infrared (NDIR) based CO₂ sensor to detect the human respiratory CO₂», Tesis doct., Universiti Teknologi Malaysia, 2019.
- [58] K. Sugino y col., «A bluetooth-based device-free motion detector for a remote elder care support system», en *2015 IIAI 4th International Congress on Advanced Applied Informatics*, IEEE, 2015, págs. 91-96.
- [59] K. Sun, Q. Zhao y J. Zou, «A review of building occupancy measurement systems», *Energy and Buildings*, vol. 216, pág. 109 965, 2020.

- [60] E. R. Sykes, S. Pentland y S. Nardi, «Context-aware mobile apps using iBeacons: towards smarter interactions.», en *CASCON*, 2015, págs. 120-129.
- [61] C. Tang y col., «Occupancy Detection and People Counting Using WiFi Passive Radar», en *2020 IEEE Radar Conference (RadarConf20)*, IEEE, 2020, págs. 1-6.
- [62] K. W. Tham, «Indoor air quality and its effects on humans—A review of challenges and developments in the last 30 years», *Energy and Buildings*, vol. 130, págs. 637-650, 2016.
- [63] L. Torvalds y col., «Linux», URL: <http://www.linux.org>, vol. 2, págs. 263-297, 2002.
- [64] C. Tsai y M.-S. Young, «Pyroelectric infrared sensor-based thermometer for monitoring indoor objects», *Review of scientific instruments*, vol. 74, n.º 12, págs. 5267-5273, 2003.
- [65] A. Tyndall, R. Cardell-Oliver y A. Keating, «Occupancy estimation using a low-pixel count thermal imager», *IEEE Sensors Journal*, vol. 16, n.º 10, págs. 3784-3791, 2016.
- [66] D. Van Merode y col., «Flexible technologies for smart campus», en *2016 13th International Conference on Remote Engineering and Virtual Instrumentation (REV)*, IEEE, 2016, págs. 64-68.
- [67] T. Vehviläinen y col., «High indoor CO₂ concentrations in an office environment increases the transcutaneous CO₂ level and sleepiness during cognitive work», *Journal of occupational and environmental hygiene*, vol. 13, n.º 1, págs. 19-29, 2016.
- [68] L. Vu y col., «Characterizing and modeling people movement from mobile phone sensing traces», *Pervasive and Mobile Computing*, vol. 17, págs. 220-235, 2015.
- [69] D. N. Wagner, A. Mathur y B. E. Boor, «Spatial seated occupancy detection in offices with a chair-based temperature sensor array», *Building and Environment*, vol. 187, pág. 107360, 2021.
- [70] J. Wang y Z. Xu, «Texture-based homogeneity analysis for crowd scene modelling and abnormality detection», en *2014 20th International Conference on Automation and Computing*, IEEE, 2014, págs. 182-187.
- [71] J. Wang, N. C. F. Tse y J. Y. C. Chan, «Wi-Fi based occupancy detection in a complex indoor space under discontinuous wireless communication: A robust filtering based on event-triggered updating», *Building and Environment*, vol. 151, págs. 228-239, 2019.
- [72] K. Weekly y col., «Building-in-Briefcase: A rapidly-deployable environmental sensor suite for the smart building», *Sensors*, vol. 18, n.º 5, pág. 1381, 2018.
- [73] Y. Wen y T. Liu, «WiFi Security Certification through Device Information», en *2018 International Conference on Sensor Networks and Signal Processing (SNSP)*, IEEE, 2018, págs. 302-305.
- [74] L. Wu e Y. Wang, «A low-power electric-mechanical driving approach for true occupancy detection using a shuttered passive infrared sensor», *IEEE Sensors Journal*, vol. 19, n.º 1, págs. 47-57, 2018.
- [75] L. Wu, Y. Wang y H. Liu, «Occupancy detection and localization by monitoring nonlinear energy flow of a shuttered passive infrared sensor», *IEEE Sensors Journal*, vol. 18, n.º 21, págs. 8656-8666, 2018.
- [76] L. Wu y col., «SLEEPiR: Synchronized Low-Energy Electronically Chopped PIR Sensor for True Presence Detection», *IEEE Sensors Letters*, vol. 4, n.º 3, págs. 1-4, 2020.
- [77] X. Xie y col., «How far droplets can move in indoor environments—revisiting the Wells evaporation–falling curve», *Indoor air*, vol. 17, n.º 3, págs. 211-225, 2007.
- [78] M. Yadav, N. Goyal y J. Yadav, «Agile Methodology over iterative approach of Software Development-A review», en *2015 2nd International Conference on Computing for Sustainable Global Development (INDIACom)*, IEEE, 2015, págs. 542-547.
- [79] Y. Yoshimura y col., «Analysis of visitors' mobility patterns through random walk in the Louvre Museum», *Journal of Ambient Intelligence and Humanized Computing*, págs. 1-16, 2019.
- [80] T. Zajic y R. P. Mahler, «Particle-systems implementation of the PHD multitarget tracking filter», en *Signal Processing, Sensor Fusion, and Target Recognition XII*, International Society for Optics y Photonics, vol. 5096, 2003, págs. 291-299.
- [81] P. Zappi, E. Farella y L. Benini, «Enhancing the spatial resolution of presence detection in a PIR based wireless surveillance network», en *2007 IEEE Conference on Advanced Video and Signal Based Surveillance*, IEEE, 2007, págs. 295-300.

-
- [82] W. Zheng y col., «COVID-19 Impact on Operation and Energy Consumption of Heating, Ventilation and Air-Conditioning (HVAC) Systems», *Advances in Applied Energy*, pág. 100 040, 2021.
- [83] Y. Zhou, Z. Kang y C. Spanos, «Parametric dual maximization for non-convex learning problems», en *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 31, 2017.
- [84] H. Zou y col., «Bluedetect: An ibeacon-enabled scheme for accurate and energy-efficient indoor-outdoor detection and seamless location-based service», *Sensors*, vol. 16, n.º 2, pág. 268, 2016.
- [85] H. Zou y col., «Freedetector: Device-free occupancy detection with commodity wifi», en *2017 IEEE International Conference on Sensing, Communication and Networking (SECON Workshops)*, IEEE, 2017, págs. 1-5.
- [86] H. Zou y col., «Device-free occupancy detection and crowd counting in smart buildings with WiFi-enabled IoT», *Energy and Buildings*, vol. 174, págs. 309-322, 2018.
- [87] —, «Towards occupant activity driven smart buildings via WiFi-enabled IoT devices and deep learning», *Energy and Buildings*, vol. 177, págs. 12-22, 2018.
- [88] J. Zou y col., «Occupancy detection in the office by analyzing surveillance videos and its application to building energy conservation», *Energy and Buildings*, vol. 152, págs. 385-398, 2017.

ANEXOS

REPOSITORIOS DEL SISTEMA

La parte *software* relacionada con el proyecto ha sido alojada en diversos repositorios de la plataforma *Github*. Se ha optado por dividir esta parte del sistema en tres repositorios, para así poder diferenciar cada una de ellas.

A.1. REPOSITORIO DE LA RED DE NODOS WSN

En este repositorio se encuentra almacenado el *firmware* de los dos tipos de nodos que han sido diseñados e implementados, estos son *Door Node* y *Scan Node*, presentados en la Sección 5.3.

Enlace del repositorio: <https://github.com/dcamunas/tfg-wsn>

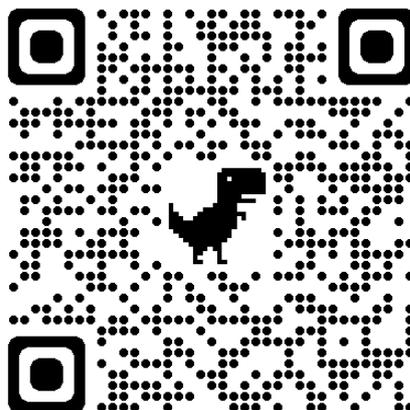


Figura A.1: Código QR del repositorio de la red WSN.

A.2. REPOSITORIO DEL BACKEND DEL SISTEMA

En este repositorio se encuentra almacenada la parte relacionada con el *Backend* del sistema desarrollado, junto a la *aplicación de monitorización de tránsito humano*, debido a que esta aplicación, al igual que los servicios que componen el *Backend*, ha sido desplegado con el uso de *Docker*. Ambas partes presentadas en las Secciones 5.4 y 5.6.

Enlace del repositorio: <https://github.com/dcamunas/tfg-backend>

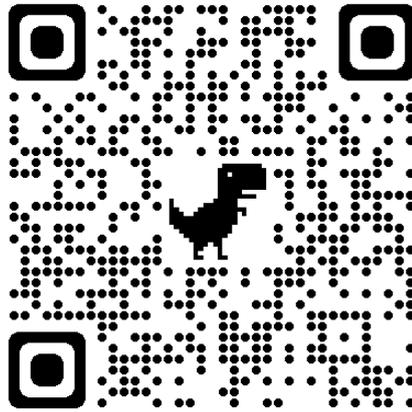


Figura A.2: Código QR del repositorio del *Backend* del sistema.

A.3. REPOSITORIO DE LA APLICACIÓN DE DISTRIBUCIÓN DE RECURSOS DE LIMPIEZA

En este repositorio se encuentra almacenada la *aplicación de distribución óptima de recursos de limpieza*, presentada en la Sección 5.5.

Enlace del repositorio: <https://github.com/dcamunas/tfg-distriapp>

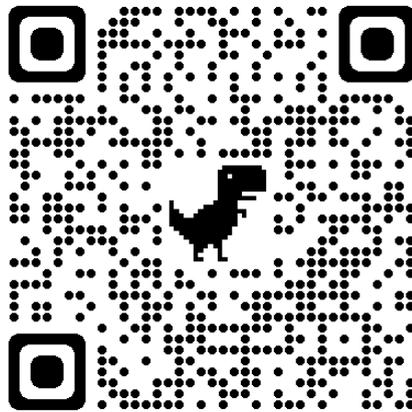


Figura A.3: Código QR del repositorio de la aplicación de distribución de recursos de limpieza.

EXPERIMENTACIÓN EN UN ESCENARIO SIMULADO

En este Anexo se ha experimentado sobre la aplicación de distribución de recursos de limpieza a partir del uso de un escenario simulado, el cual, se caracteriza por un mayor grado de complejidad respecto al presentado en el Apartado 6.2.3. Este escenario está formado por seis salas de distintos tamaños (100, 120, 50, 29, 20 y 17 m^2), y se dispone de un equipo humano de limpieza de nueve componentes. Los resultados relacionados con esta simulación se pueden encontrar en las Tablas B.1 y B.2, donde la Tabla B.1 representa la definición del plan de desinfección más óptimo (distribución más beneficiosa según el sistema), y la Tabla B.2 presenta en forma de tabla un *ranking* con los diez mejores distribuciones de recursos generadas por la aplicación.

Sala	Tamaño (m^2)	Ocupación %	Nivel CO_2	Prioridad	Recursos
1	100	78	889	3	2
2	120	90	956	1	3
3	50	68	820	4	1
4	20	100	901	3	1
5	17	88	811	5	1
6	29	55	789	6	1

Tabla B.1: Resultados de la distribución más óptima en el escenario simulado.

En la Tabla B.1, se muestra el resultado obtenido basado en la distribución más óptima (mejor plan de desinfección) calculada por el sistema. Al igual que en el caso anterior, el sistema ha analizado las necesidades de desinfección de cada una de las salas según el nivel de CO_2 , la prioridad de desinfección, el porcentaje de ocupación y el propio tamaño de la sala. El plan que más beneficio reportado ha sido la distribución de 1 personal de limpieza a las salas con una menor prioridad (salas 3, 4, 5 y 6), dos a la sala número 1 y tres a la sala con mayor prioridad de desinfección (sala número 2). En base a esta distribución de recursos, se ha conseguido la desinfección de todas las salas que componen el entorno, consiguiendo así que mientras los componentes de limpieza se encuentran desinfectando sus respectivas salas de forma individual, los grupos formados por más de un componente desinfecten en el mismo periodo de tiempo las salas de mayor tamaño.

Por último, en la Tabla B.2 se muestra el *ranking* donde el sistema ha definido las diez mejores distribuciones de recursos, posicionando en primer lugar la mejor distribución, correspondiente al plan de desinfección más óptimo. Este *ranking* sigue el mismo formato que el visto en el Apartado 6.2.3, a diferencia del número de salas y los recursos de limpieza disponibles. De igual forma, se puede observar como los mejores planes de desinfección son aquellos donde el equipo de limpieza llega a cubrir todas las salas que componen el entorno, esto se debe a que el sistema premia este tipo de distribuciones otorgándole un mayor beneficio (incrementando su beneficio un 10 %). Seguidamente,

si se hace hincapié en el plan más óptimo de este escenario, se puede observar como a pesar de que la sala número 4 dispone de un nivel de prioridad de desinfección mayor (2) respecto a algunas de las demás salas, se le ha otorgado un solo componente de limpieza (sobre un 11 % de los recursos), esto se debe a que la sala número 1 con un nivel de prioridad menor (3) posee un tamaño mayor. Por lo tanto, en aras de optimizar los recursos y desinfectar el entorno en el menor tiempo posible se le han otorgado un grupo formado por dos componentes del equipo de limpieza, es decir, alrededor del 22 % de los recursos disponibles. Además, se puede observar como en la sala que se ha definido el mayor nivel de prioridad (sala 2 con un nivel de prioridad 1), se le ha otorgado un grupo formado por tres componentes del equipo de limpieza, es decir, más del 33 % de los recursos disponibles.

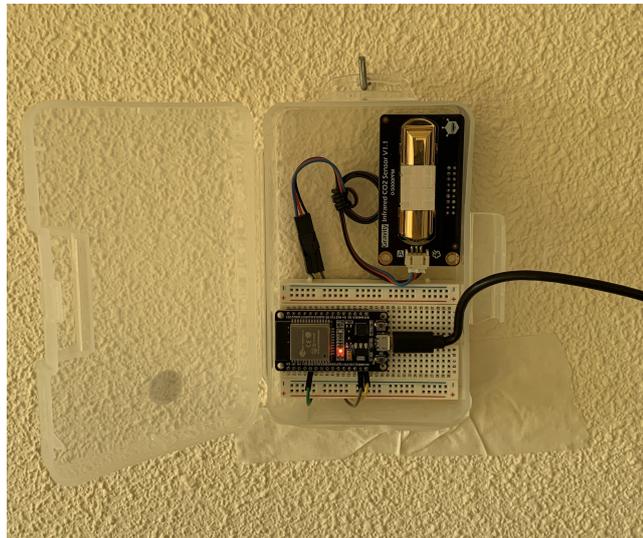
Posición	Recursos - Sala						Beneficio
	Sala 1	Sala 2	Sala 3	Sala 4	Sala 5	Sala 6	
1	2	3	1	1	1	1	1,759
2	1	4	1	1	1	1	1,651
3	1	3	2	1	1	1	1,626
4	2	2	2	1	1	1	1,565
5	2	4	1	1	1	0	1,445
6	3	3	1	1	0	1	1,321
7	4	3	1	1	0	0	1,286
8	2	4	1	1	1	0	1,189
9	3	1	2	2	1	0	1,06
10	3	4	1	0	1	0	0,883

Tabla B.2: Ranking de las diez mejores distribuciones de recursos en el escenario simulado.

IMÁGENES DEL SISTEMA

C.1. DESPLIEGUE DE LA RED DE NODOS DE SENSORIZACIÓN

En la Figura C.1 se muestran los dos tipos de nodos que han sido desplegados para la sensorización del entorno, instalando en cada una de las salas dos nodos, uno de cada tipo.



(a) *Scan Node.*



(b) *Door Node.*

Figura C.1: Fotografías de los dos nodos instalados en una de las salas.

C.2. IMÁGENES DEL ENTORNO DE MONITORIZACIÓN

En la Figura C.2, se muestran una serie de fotografías sobre las pruebas de monitorización realizadas, donde se puede apreciar el tránsito humano. En concreto, estas fotografías pertenecen a diferentes momentos de ocupación capturados en la Sala 1 del entorno de monitorización, cuyos resultados se muestran en la Tabla 6.1 del Capítulo de Resultados 6.



(a) Sala 1 vacía (0 % de ocupación).



(b) Sala 1 ocupada (50 % ocupación).

Figura C.2: Fotografías de la Sala 1 en diferentes instantes de tiempo.

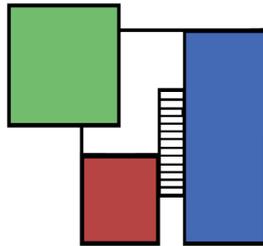
Por otro lado, en la Figura C.3 se muestra en una fotografía el instante en el que un ocupante (1) está accediendo a la Sala 1. En la fotografía aparece señalado el nodo encargado de la contabilización de los ocupantes que acceden y salen de dicha sala, conocido como *Door Node* (2).



Figura C.3: Acceso a Sala 1.

C.3. IMÁGENES DE LA APLICACIÓN DE DISTRIBUCIÓN DE RECURSOS DE LIMPIEZA

Por otro lado, en la Figura C.4 se muestran las dos pestañas que componen la interfaz de escritorio correspondiente a la aplicación de distribución de recursos de limpieza. (a) corresponde a la pestaña de *Mapa* donde se muestra el mapa del entorno y la definición del plan más óptimo calculado. Por otra parte, (b) representa la pestaña de *Rutas*, donde se muestra información relevante sobre el entorno de monitorización y un *ranking* en forma de tabla de los diez mejores planes de desinfección (rutas).

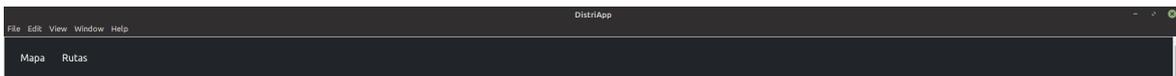


• Plan de desinfección

◦ Número personal de limpieza: 4

#	Sala	Tamaño	Limite aforo	Aforo	Ocupación %	Nivel CO2	Prioridad	Recursos
●	1	17	4	4	75 %	775	2	
●	2	46	11	6	36 %	601	3	
●	3	29	7	6	86 %	862	1	

(a) Pestaña *Mapa* (principal).



• Información del entorno

◦ Número personal de limpieza: 4

#	Sala	Tamaño	Ocupación %	Nivel CO2	Prioridad
●	1	16	75 %	775	2
●	2	46	36 %	601	3
●	3	28	86 %	862	1

• Ranking de salas

Plan	Recursos por Sala:			Beneficio Obtenido
	●	●	●	
1				1,616
2				1,604
3				1,6
4				1,35
5				1,334
6				1,334
7				1,24
8				1,203
9				0,883
10				0,867

(b) Pestaña de *Rutas*.

Figura C.4: Interfaz de escritorio de la aplicación de distribución de recursos.

LISTADOS DE CONFIGURACIÓN DEL SISTEMA

D.1. LISTADO DE APROVISIONAMIENTO DEL *DASHBOARD*

En el Listado [D.1](#) se muestra el fichero de configuración correspondiente al *dashboard* que representa la pantalla principal de la aplicación de monitorización del tránsito humano.

Listado D.1: Fichero de aprovisionamiento configuración del *dashboard*

```
1  apiVersion: 1
2  providers:
3  - name: InfluxDB
4    folder: ''
5    type: file
6    disableDeletion: false
7    editable: true
8    options:
9      path: /etc/grafana/provisioning/dashboards
```

D.2. LISTADO DE CONFIGURACIÓN DE *DOCKER COMPOSE*

En el Listado D.2 se muestra el fichero de configuración correspondiente a las aplicaciones que forman el *Backend* del sistema desarrollado en este proyecto.

Listado D.2: Configuración del fichero *docker compose* de los contenedores del *Backend*

```
1 version: '3'
2 services:
3   mosquito:
4     image: eclipse-mosquitto
5     container_name: mosquito
6     ports:
7       - 1883:1883
8     volumes:
9       - ./mosquitto/mosquitto.conf:/mosquitto/config/mosquitto.conf
10      - mosquito_data:/mosquitto/data
11      - mosquito_log:/mosquitto/log
12
13   telegraf:
14     image: telegraf:1.18
15     container_name: telegraf
16     volumes:
17       - ./telegraf/telegraf.conf:/etc/telegraf/telegraf.conf
18     depends_on:
19       - influxdb
20
21   influxdb:
22     image: influxdb:1.8
23     container_name: influxdb
24     ports:
25       - 8086:8086
26     volumes:
27       - influxdb_data:/var/lib/influxdb
28     environment:
29       - INFLUXDB_DB=${INFLUXDB_DB}
30       - INFLUXDB_ADMIN_USER=${INFLUXDB_USERNAME}
31       - INFLUXDB_ADMIN_PASSWORD=${INFLUXDB_PASSWORD}
32
33   grafana:
34     image: grafana/grafana
35     container_name: grafana
36     ports:
37       - 3000:3000
38     volumes:
39       - grafana_data:/var/lib/grafana
40       - ./grafana-provisioning/:/etc/grafana/provisioning
41     depends_on:
42       - influxdb
```