



TRABAJO FIN DE GRADO EN INGENIERÍA QUÍMICA INDUSTRIAL

DESARROLLO DEL CONTROL AUTOMÁTICO DE LOS PROCESOS QUÍMICOS Y BIOLÓGICOS DE LA CÁMARA ATMOSFÉRICA TAR

Autor:

Carlos Bretthauer Yedro

Tutores:

Julián Lebrato Martínez

María Dolores Garvi Higuera

Fecha de entrega: 02/2024

DEDICATORIA

A Julián, Dolores y Miguel Ángel por su ayuda y apoyo.

ÍNDICE

1. INTRODUCCIÓN	5
1.1. Fitorremediación	5
1.2. Parámetros.....	6
2. MATERIALES Y MONTAJE	9
2.1. Elementos fundamentales.....	9
2.2. Conectividad	9
3. HARDWARE	10
3.1. ESP32/ESP32 CAM.....	10
3.2. BME680	11
3.3. MH-Z19B.....	11
3.4. Raspberry PI 4 Model B	12
4. INSTALACIÓN DE SOFTWARE	13
4.1. Instalación del sistema operativo Raspberry Pi OS	13
4.2. Instalación de Docker.....	13
4.3. Instalación de Portainer.....	14
4.4. Instalación de FileBrowser	14
4.4.1. Instalación e FileBrowser por ejecución de comandos	15
4.4.2. Instalación de FileBrowser desplegando un <i>stack</i>	15
4.5. Instalación de Mosquitto.....	16
4.6. Instalación de Home Assistant	18

5. AUTOMATIZACIÓN	19
5.1. Configuración de los sensores	19
5.1.1. Tasmota.....	19
5.2. Integraciones.....	23
5.3. Automatizaciones	25
5.4. Muestreo	26
6. Control remoto	31
6.1. VNC.....	31
6.2. Telegram	31
6.2.1. Enviar estado actual de los sensores	33
6.2.2. Enviar el fichero de valores muestreados.....	35
7. ENSAYOS.....	37
7.1. Ensayos de calibración inicial	37
7.2. Otros ensayos	39
8. CONCLUSIONES	41
Bibliografía.....	42

1. INTRODUCCIÓN

Con el presente prototipo se pretende cuantificar parámetros referentes a la polución del aire y a su calidad.

Su objetivo principal es optimizar la fitorremediación atmosférica en núcleos urbanos, obteniendo datos acerca de qué clase de masa verde es más favorable para el procesamiento de ciertos gases o elementos nocivos presentes.

Los ensayos pueden hacerse de diferentes formas, no necesariamente requiriendo plantas. Algunas posibles metodologías son las siguientes:

- Suelo + Vegetal.
- Vegetal.
- Suelo.

Otro gran objetivo es la conectividad de los sensores. Todos están conectados a la red, pudiendo tener acceso en tiempo real a los datos y control. De esta forma se puede estudiar la evolución del comportamiento de los sistemas formados en la cámara frente al aire patrón.

Se podrá trabajar en estático o en dinámico y en estufa a temperatura calibrada (para muestras que liberen gases a elevadas temperaturas como, por ejemplo, el asfalto).

1.1. Fitorremediación

La fitorremediación es una técnica que utilizan las plantas para procesar contaminantes del suelo, el agua y la tierra. Se engloban dentro de las técnicas de fitorremediación diferentes modalidades dependiendo de la parte de la planta que absorba el contaminante y del tipo de contaminante en sí: rizofiltración, fitoestabilización, fitoinmovilización, fitodegradación, fitoextracción y fitovolatilización son algunas de estas.²

El dispositivo aquí desarrollado se enfoca en la medición de la degradación, acumulación o absorción de contaminantes atmosféricos. Sin embargo, puede adaptarse para la medición de cualquier contaminante si se poseen los sensores adecuados.

Un factor a tener en cuenta es que algunas especies de plantas tienen mayor aptitud fitorremediadora que otras debido a la influencia de las características propias de la planta. Las plantas con capacidad fitorremediadora se denominan “acumuladoras”. Por ejemplo, algunas especies de álamos y sauces tienen capacidad para acumular metales pesados, funcionando como bioindicadores para la polución.⁹

Para llevar a cabo la experimentación, habrá que tener en cuenta una serie de variables. Con la propia experimentación se llevará a cabo la optimización del método, pudiendo descartar variables que no sean de elevada importancia y averiguar qué parámetros tienen mayor influencia en la fitorremediación del aire y qué modelo sería el adecuado.

1.2. Parámetros

La influencia de ciertos factores afectará al comportamiento de la planta en términos de absorción de contaminantes. Por ser un prototipo inicial, solo se podrá experimentar con un número limitado de los factores mostrados.

- **Planta:**

- **Especie:** Algunas plantas tienen mayor afinidad por ciertos contaminantes atmosféricos y pueden ser más efectivas en su captura. Se presentan, a continuación, una serie de plantas que son conocidas por poder procesar el aire y eliminar contaminantes:
 - ➔ **Sansevieria trifasciata (Lengua de suegra):** Esta planta es conocida por su capacidad para eliminar compuestos orgánicos volátiles, como el formaldehído y el benceno, del aire. ⁵
 - ➔ **Epipremnum aureum (Potos):** Esta planta es conocida por su resistencia y capacidad para eliminar formaldehído y otros contaminantes del aire. Un estudio publicado en la revista "HortScience" en 2009 encontró que el Potos fue efectivo en la eliminación de varios COV del aire interior. ¹

→ *Chamaedorea seifrizii* (Palma de bambú): Esta palma es conocida por su capacidad para eliminar formaldehído y otros contaminantes del aire. ³

- Edad.
- SLA (Superficie específica de hoja): También conocida como Área Foliar Específica, describe la relación entre el área de hoja de una planta y su masa.

$$SLA = \frac{A}{M_L}$$

Siendo A el área de hojas y M_L la masa seca de todas esas hojas.

En general, un alto SLA indica hojas delgadas y ligeras beneficioso en condiciones de baja luminosidad. Las hojas delgadas pueden estar más extendidas y tener, así, mayor superficie expuesta a la luz solar, además de poder disponerse en ángulos variados. Sin embargo, esto puede variar según la especie de planta y el entorno, ya que la adaptación de una planta a su entorno depende de más variables. Existen unos valores típicos de SLA dependiendo del tipo de planta. El área específica que desarrolle una planta estará íntimamente relacionada con el estrés al que se vea sometida la planta durante su desarrollo. ⁶

- **Suelo:**

- pH.
- Cantidad de materia orgánica.
- Humedad.

- **Ambiente.**

- Humedad relativa.
- Presión.
- Temperatura.
- Longitud de onda (Luz).

- **Concentración de contaminantes:**

- NO₂: de todos los NO_x, el NO₂ es el más producido por los humanos. Al igual que con el dióxido de azufre, la habilidad de las plantas para absorber NO₂ reside en los estomas.⁶
- SO₂: los dióxidos de azufre fueron los primeros contaminantes aéreos en causar daños al ser humano y su concentración ha aumentado con el uso de combustibles fósiles.¹⁸ Las hojas de las plantas absorben este contaminante por los estomas y se hidratan y oxidan a sulfito y sulfato. Sin embargo, cuando se acumula hasta altas concentraciones, se puede inhibir la fotosíntesis y otros procesos, siendo de esta forma inconveniente para la planta.¹⁴
- Compuestos orgánicos volátiles: los compuestos orgánicos volátiles son eliminados por ciertas plantas a tan buen ritmo que puede comprarse con el efecto de máquinas de acondicionamiento de aire.^{9,16}
- CO₂.
- PM_{2.5}: quizás el contaminante menos nocivo de los que se nombran. Es extraído del aire por deposición en las hojas de las plantas y, por lo tanto, se espera una gran dependencia del SLA.⁴

2. MATERIALES Y MONTAJE

2.1. Elementos fundamentales

El dispositivo consta de los siguientes elementos fundamentales:

- Dos cámaras de metacrilato de tamaño reducido.
- Una cámara de metacrilato de gran volumen.
- Fuente de luz: HL-3050F de 16W, alcanzando 2400LM (20'5x2'6x0'4 pulgadas): permite simular ciclos de día y noche, luminosidad baja e incluso tiempo desfavorable.
- Ventiladores de CPU para la homogeneización interna del aire.

Todas las cámaras de metacrilato poseen 4 orificios sellables para la introducción de cableado y tubos de conexión con las demás.



Figura 1. A la izquierda, la cámara de metacrilato de mayor tamaño. A la derecha, las dos cámaras de menor tamaño junto con la lámpara y sensores.

2.2. Conectividad

La elección de ciertos dispositivos y la toma de decisiones se vio condicionada por la intención de hacer la toma de datos pública y accesible desde dispositivos particulares. La conectividad Wi-Fi de todos los dispositivos encargados de procesar datos es un requerimiento indispensable.

3. HARDWARE

Se introduce brevemente a continuación el hardware básico empleado.

3.1. ESP32/ESP32 CAM

Se trata de una placa de desarrollo de bajo coste y reducido gasto. Posee tecnología Wi-Fi y Bluetooth. Se usará también su versión con cámara incorporada.

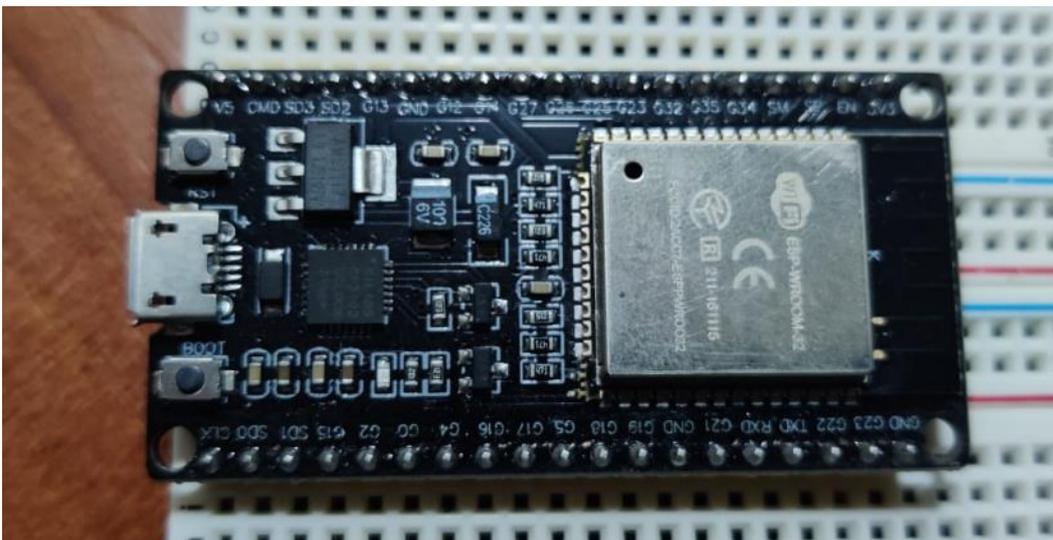


Figura 2. Placa de desarrollo ESP32

La elección de esta placa de desarrollo frente a otras radica en que, por su bajo coste, pueden adquirirse gran número de éstas en función de la cantidad de sensores a instalar. Al poseer conectividad Wi-Fi, existe la posibilidad de configurar todos los sensores conectados a esta placa de desarrollo de forma remota.

Esta placa posee interfaz I2C, esencial para la conexión de ciertos sensores que se comunican mediante este protocolo. El protocolo I2C utiliza dos conexiones para llevar a cabo su comunicación: SDA (Serial Data Line) y SCL (Serial Clock Line). La conexión SDA se encarga de recibir y enviar datos, mientras que la conexión SCL marca el ritmo de reloj para la sincronización.

Se utilizarán también los pines RX (recepción) y TX (transmisión) para la incorporación de un sensor de CO₂ por infrarrojo.

3.2. BME680

Sensor ambiental I2C que contiene un pequeño sensor MOX, que cambia su resistencia según los compuestos orgánicos volátiles del aire (servirá para detectar gases, alcoholes y monóxido de carbono). Su desempeño en cuanto a humedad relativa es destacable. No es capaz de diferenciar entre gases y alcoholes.

El uso que se le dará en este prototipo es el de medir parámetros generales de la cámara en la que esté situado. Estos parámetros son la temperatura, humedad relativa, presión y compuestos orgánicos volátiles.



Figura 3. Sensor BME680

3.3. MH-Z19B

Sensor de dióxido de carbono Infrarrojo No Dispersivo (NDIR). El principio de funcionamiento se basa en la absorción de infrarrojos por parte de ciertos gases. Sus principales componentes son los siguientes:

- Fuente de luz infrarroja: emite radiación infrarroja con longitudes de onda específicas para la absorción por parte del CO₂.
- Celda de gas: cámara donde estará el CO₂.
- Detector de infrarrojos: mide la cantidad de luz que no absorbe el gas, relacionada con la concentración del gas en la muestra.

Consta con dos pines TX y RX para la conexión con la placa de desarrollo, además de los respectivos pines de tensión y GND.

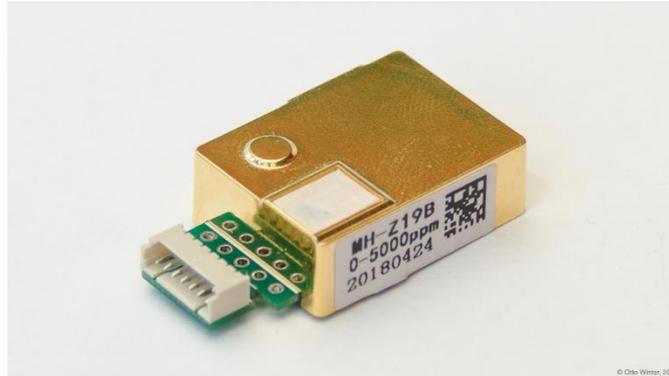


Figura 4. Sensor MH-Z19B

3.4. Raspberry PI 4 Model B

Ordenador de placa simple, de reducido tamaño y con la suficiente potencia para llevar a cabo las funciones de muestreo y procesamiento de datos para la que se usará. Integra conexiones USB ideales para la incorporación de periféricos para su correcto manejo y un puerto Micro HDMI para la conexión de un monitor. Fue importante la adquisición de un chasis con ventiladores para asegurar la refrigeración de la Raspberry durante un uso continuado. Permite la conexión Wi-Fi.

4. INSTALACIÓN DE SOFTWARE

En este apartado se describe el software a instalar en la Raspberry, además de detallar la instalación.

4.1. Instalación del sistema operativo Raspberry Pi OS

Se instalará el sistema operativo Raspberry Pi OS con escritorio, de 64 bits. Este sistema operativo está basado en Debian, y viene con el software recomendado para la mayoría de computadoras de trabajo.

Una de las grandes ventajas de este sistema operativo es su liviandad, pudiendo incluso instalarse en computadoras antiguas si la situación lo requiriera.

Es opcional e interesante que, durante o tras la instalación, se active la comunicación SSH y el escritorio remoto. De esta forma, el usuario puede acceder a sus servidores remotos a través de Internet.

La imagen del sistema operativo puede descargarse en la página oficial de software Raspberry Pi, siendo de uso libre. La instalación puede realizarse desde un USB, previamente *flasheada* la imagen del sistema operativo.

4.2. Instalación de Docker

Docker es una plataforma de uso libre para ejecutar aplicaciones separadamente en un entorno denominado *container* (contenedor). Estos *containers* son parecidos a máquinas virtuales pequeñas y aisladas, no afectando a otras aplicaciones fuera del propio container.

Para su instalación, en una consola dentro del sistema se ejecutan los siguientes comandos:

```
curl -fsSL https://get.docker.com -o get-docker.sh  
  
sudo sh get-docker.sh
```

De esta forma, instalamos Docker en nuestro dispositivo. Para comprobar si la instalación se llevó a cabo correctamente, ejecutamos el siguiente comando:

```
sudo docker run hello-world
```

4.3. Instalación de Portainer

Portainer es una interfaz gráfica, accesible mediante navegador web, para el manejo de *containers*.

Crearemos una carpeta llamada *docker*, donde guardaremos las aplicaciones instaladas. Dentro de esta carpeta, creamos otra llamada *portainer*, donde lo instalaremos mediante la ejecución de este comando:

```
sudo docker run -d -p 8000:8000 -p 9443:9443 --name=portainer  
--restart=unless-stopped -v  
/var/run/docker.sock:/var/run/docker.sock -v  
/home/grupotar/docker/portainer:/data portainer/portainer-ce
```

Este comando habrá instalado Portainer, y lo estará alojando en la dirección especificada en el propio comando. Por ello, para acceder a la interfaz de Portainer, deberemos dirigirnos al navegador web e ingresar la dirección en la que se aloja: <https://localhost:9443/>, siendo "localhost" sustituible por la IP local que tenga el dispositivo. Esta última forma sería la adecuada para acceder a la dirección, evitando así posibles errores.

4.4. Instalación de FileBrowser

Aunque no es obligatorio, es recomendable, ya que las alternativas para el uso que desempeña presentan mayor complejidad. Una alternativa es usar Visual Studio Code, con extensiones SSH y YAML para poder entrar y editar, siendo más tedioso. Otra alternativa es instalar una versión supervisada de Home Assistant, permitiendo la instalación de Add-Ons sin tener que editar la configuración a mano.

FileBrowser se usa para navegar por el dispositivo, presentando una ventaja clara sobre el navegador de archivos propio del sistema operativo: permite la edición de archivos y carpetas ocultos, pudiendo así editar archivos de configuración indispensables más adelante.

Tenemos dos alternativas para su instalación:

4.4.1. Instalación e FileBrowser por ejecución de comandos

Se ejecuta el siguiente comando:

```
sudo docker run -d --name=Filebrowser --restart=unless-stopped  
-p 8086:80 -v /:/srv filebrowser/filebrowser
```

Añadiendo “-p 8086:80” se permite exponer el puerto 80 (el que generalmente usa FileBrowser) al puerto 8086, permitiendo que otros servicios usen el puerto 80.

Añadir “-v /:/srv filebrowser/filebrowser” vincula la carpeta de la izquierda (solo escribimos “/”, ya que se refiere a la carpeta raíz que queremos poder explorar con FileBrowser) con la carpeta de la derecha (la carpeta del contenedor que FileBrowser usa por defecto), separadas por “:”.

4.4.2. Instalación de FileBrowser desplegando un *stack*

Esta alternativa de instalación aprovecha la instalación previa de Portainer. Consiste en dirigirse al apartado de *stacks* y añadir como texto las siguientes líneas:

```
version: '3'
services:
  filebrowser:
    container_name: filebrowser
    image: filebrowser/filebrowser:latest
    volumes:
      - /:/srv
    ports:
      - 8086:80
    restart: unless-stopped
```

Se despliega pulsando *deploy* y, si no hay ningún error, el contenedor de FileBrowser debería estar ejecutándose correctamente.

4.5. Instalación de Mosquitto

Mosquitto es un bróker de uso libre que implementa el protocolo MQTT. Este protocolo de comunicación permite a dispositivos y aplicaciones intercambiar mensajes de manera ligera y rápida y, por ello, es ideal para la comunicación en tiempo real. Se usará más adelante para conectar dispositivos Tasmota a Home Assistant.

Para instalarlo, debemos crear tres carpetas dentro de nuestra, ya creada, carpeta *docker*:

/config

/data

/log

Dentro de la carpeta *config*, creamos un archivo al que llamaremos “*mosquitto.conf*”. Se edita con las siguientes líneas:

```
persistence true
persistence_location /mosquitto/data/
log_dest file /mosquitto/log/mosquitto.log

listener 1883
allow_anonymous false
password_file /mosquitto/config/passwd
```

La última línea permite darle una contraseña al *broker*, indicando el nombre del archivo de texto que tendría la contraseña dentro.

La penúltima línea permite la conexión anónima sin autenticar en caso de que cambiemos *false* por *true*.

Una vez creados los archivos anteriores, procedemos a desplegar el siguiente *stack* dentro de Portainer:

```
version: '3'
services:
  mosquitto:
    container_name: mosquitto
    image: "eclipse-mosquitto"
    volumes:
      -
        /home/grupotar/docker/mosquitto/config:/mosquitto/config/
        - /home/grupotar/docker/mosquitto/data:/mosquitto/data
        - /home/grupotar/docker/mosquitto/log:/mosquitto/log
    ports:
      - "1883:1883"
      - "9001:9001"
    restart: unless-stopped
    network_mode: bridge
```

Donde las direcciones locales escritas de los volúmenes son particulares del dispositivo, ateniéndose en este caso a la instalación indicada en los anteriores pasos. Esto quiere decir que, en caso de haber creado archivos en carpetas diferentes a las indicadas, deberíamos especificar las nuevas direcciones locales del *broker*.

Al igual que con FileBrowser, el contenedor de Mosquitto debería estar montado y ejecutándose sin fallos.

4.6. Instalación de Home Assistant

Se despliega el siguiente *stack*:

```
version: '3'
services:
  homeassistant:
    container_name: homeassistant
    image: "homeassistant/home-assistant:stable"
    volumes:
      - /home/grupotar/docker/homeassistant/./config
      - /etc/localtime:/etc/localtime:ro
    restart: unless-stopped
    privileged: true
    network_mode: host
```

Tras desplegarlo, Home Assistant estará disponible y accesible desde la dirección: <http://localhost:8123> , pudiendo ser también la dirección IP de la Raspberry seguida de “:8123”.

5. AUTOMATIZACIÓN

Una vez instalado todo el software necesario, se procede a configurarlo para la medición. Se explicará la configuración dentro de la Raspberry y el montaje de los sensores físicamente.

5.1. Configuración de los sensores

La conexión física debe llevarse a cabo según el siguiente montaje

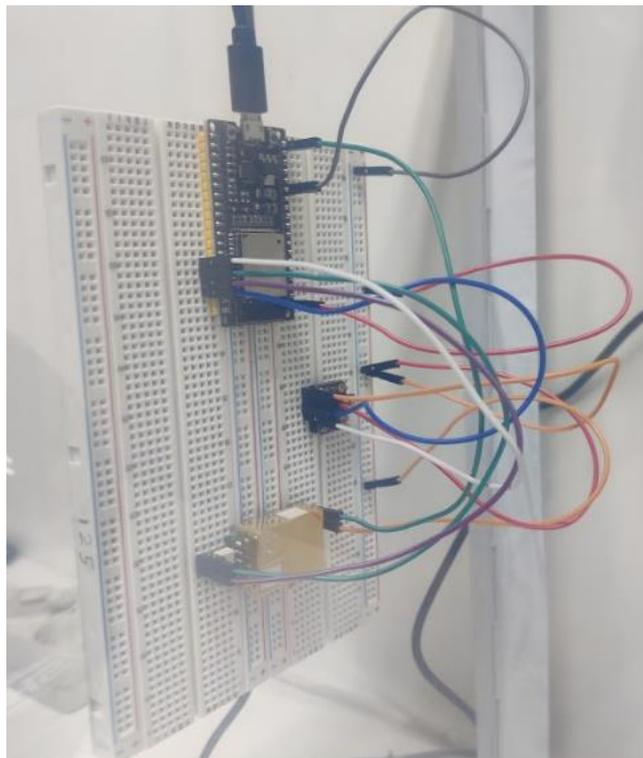


Figura 5. Conexión de los sensores a la placa de desarrollo

Para poder acceder a ellos se usará Tasmota. Se trata de una interfaz con configuraciones precargadas de los sensores, debiendo especificarse tan solo los puertos donde se conectan los sensores.

5.1.1. Tasmota

Tasmota debe instalarse previamente en la placa de desarrollo. Para poder realizar la conexión de la placa ESP32 deben instalarse los drivers CP210x, disponibles en la web. De esta forma, al acceder a la web <https://tasmota.github.io/install/> (un instalador web Tasmota) y seleccionando la

opción adecuada de Tasmota (en nuestro caso se usó la versión normal, “Tasmota (english)”), se comienza a instalar dentro de la placa de desarrollo.

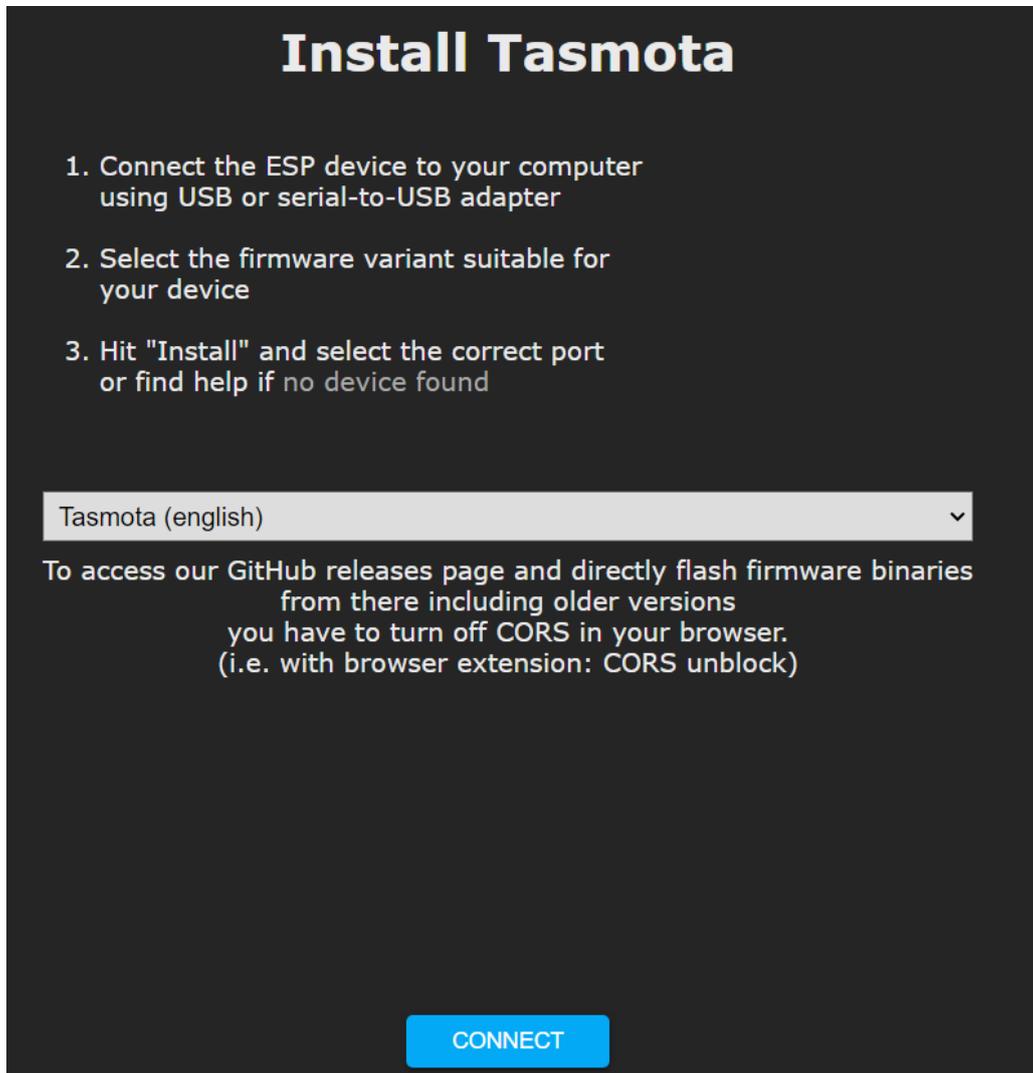


Figura 6. Menú principal para la instalación de Tasmota

Al pulsar CONNECT se debe seleccionar el puerto donde se tenga conectada la placa de desarrollo, identificable por empezar por “CP2102 USB to UART Bridge Controller”.



Figura 7. Desplegable de instalación, donde deberá configurarse la red

Al instalar Tasmota, a la placa de desarrollo se le asigna una IP (consultable en la barra de búsqueda) que debe conocerse para poder acceder a la configuración.

Para configurar los puertos y los sensores conectados a estos, se debe entrar a “Configure Module” dentro de “Configuration”. Se usa la siguiente configuración:

Module parameters

Module type (ESP32-DevKit)

ESP32-DevKit (1) ▾

IO GPIO0	None ▾	
TX GPIO1	MHZ Rx ▾	
IO GPIO2	None ▾	
RX GPIO3	MHZ Tx ▾	
IO GPIO4	None ▾	
IO GPIO5	None ▾	
FL GPIO9	None ▾	
FL GPIO10	None ▾	
IO GPIO12	None ▾	
IO GPIO13	None ▾	
IO GPIO14	None ▾	
IO GPIO15	None ▾	
IO GPIO16	None ▾	
IO GPIO17	None ▾	
IO GPIO18	None ▾	
IO GPIO19	None ▾	
IO GPIO21	I2C SDA ▾	1 ▾
IO GPIO22	I2C SCL ▾	1 ▾
IO GPIO23	None ▾	
IO GPIO25	None ▾	
IO GPIO26	None ▾	
IO GPIO27	None ▾	
AO GPIO32	None ▾	
AO GPIO33	None ▾	
IA GPIO34	None ▾	
IA GPIO35	None ▾	
IA GPIO36	None ▾	
IA GPIO39	None ▾	

Figura 8. Configuración de los sensores en Tasmota según los pines de la placa de desarrollo

Al salvar y volver al menú principal, podrá observarse que ya está configurado el dispositivo:

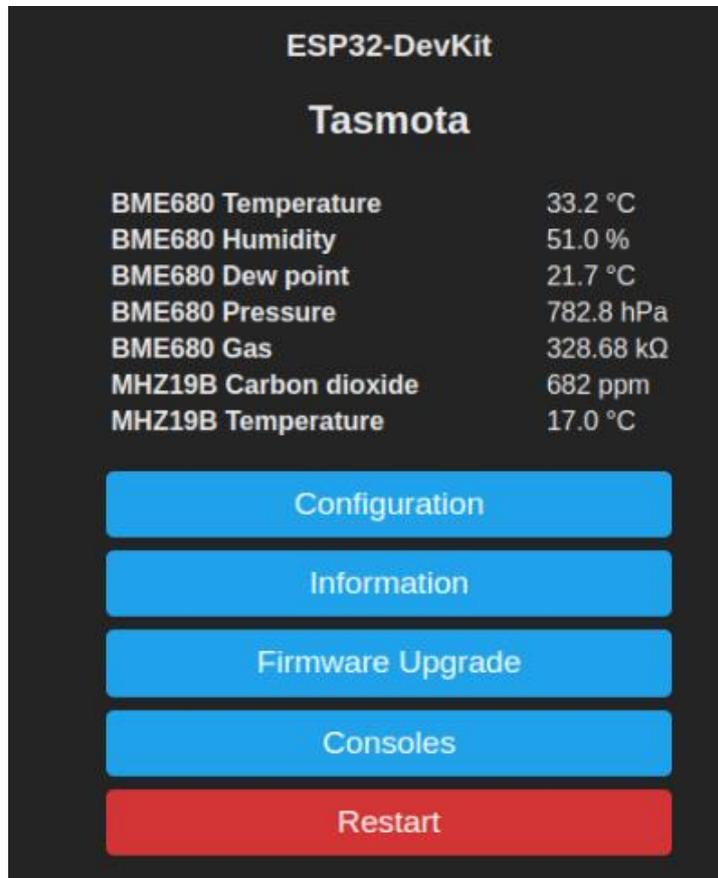


Figura 9. Dispositivo en funcionamiento dentro de Tasmota

5.2. Integraciones

Las integraciones son los procesos que se añaden a Home Assistant para que diferentes dispositivos de diferentes servicios puedan conectarse y comunicarse con él. Para la versión instalada de Home Assistant no puede usarse el Modo Supervisor para poder instalar integraciones.

Dentro de la carpeta de Docker creada, en la carpeta Home Assistant, podremos encontrar los siguientes ficheros:



Figura 10. Ficheros requeridos dentro de la carpeta Docker

Estos ficheros son de gran importancia, ya que editarlos es la forma de añadir y configurar integraciones y automatizaciones.

La conectividad entre Mosquitto y Tasmota debe configurarse. Esto permite que Home Assistant detecte los dispositivos que tengan instalado Tasmota y, de esta forma, puedan usarse y recogerse los datos enviados por los sensores.

Para ello, se debe ingresar en el apartado “Integraciones” dentro de Home Assistant y desde ahí añadir la integración MQTT.

Los datos a ingresar dentro de la configuración de MQTT pueden hallarse en la configuración de MQTT dentro de Tasmota:

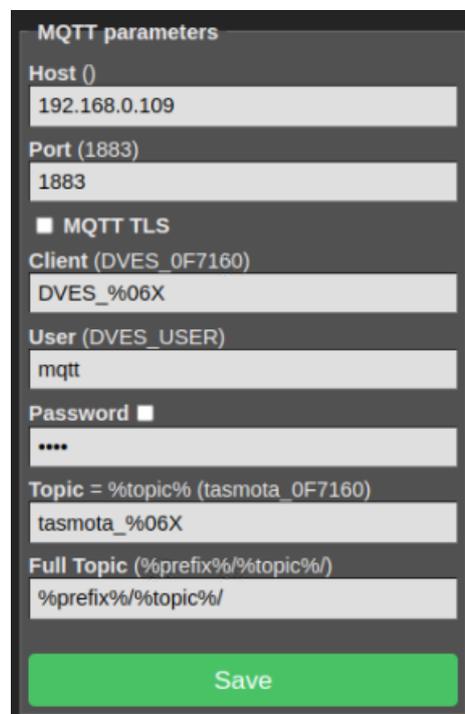


Figura 11. Configuración del bróker dentro de Tasmota, donde el usuario y la contraseña serán propios y la IP será la de la Raspberry

5.3. Automatizaciones

Las automatizaciones son un conjunto de reglas y condiciones que responden a eventos particulares. De esta forma, se le puede comunicar a Home Assistant qué hacer cuando, por ejemplo, pasa una cantidad determinada de tiempo o un sensor toma cierto valor.

A la hora de trabajar con automatizaciones, existen unos entornos denominados plantillas (*templates*). Con ellas, se pueden visualizar rápidamente resultados de operaciones llevadas a cabo que se implementen en las automatizaciones. El editor de plantillas en Home Assistant lo encontraremos en las Herramientas de Desarrollador (*Developer Tools*). Desde ahí, la ventana de Estados (*States*) sirve para identificar más fácilmente las IDs de los diferentes sensores o dispositivos, además de los atributos que poseen. Existe la posibilidad de buscar el que necesitamos, filtrándolo de la lista. Las plantillas se renderizan usando el motor Jinja2. Es un motor para plantillas desarrollado en Python.

Por ejemplo, si escribimos este código en el editor de la ventana de Templates:

```
;{{states.sensor.tasmota_mhz19b_carbondioxide.state}};
{{states.sensor.tasmota_mhz19b_carbondioxide_2.state}};
{{states.sensor.tasmota_bme680_temperature.state}};
{{states.sensor.tasmota_bme680_temperature_2.state}};
{{states.sensor.tasmota_bme680_pressure.state}};
{{states.sensor.tasmota_bme680_pressure_2.state}};
{{states.sensor.tasmota_bme680_humidity.state}};
{{states.sensor.tasmota_bme680_humidity_2.state}};
{{states.sensor.tasmota_mhz19b_carbondioxide.state |float -
states.sensor.tasmota_mhz19b_carbondioxide_2.state |float }};
{{states.sensor.tasmota_bme680_temperature.state |float -
states.sensor.tasmota_bme680_temperature_2.state |float}};
{{states.sensor.tasmota_bme680_pressure.state |float -
states.sensor.tasmota_bme680_pressure_2.state |float}};
{{states.sensor.tasmota_bme680_humidity.state |float-
states.sensor.tasmota_bme680_humidity_2.state |float}};
```

Tipo de resultado: string

```
message: >-
;659;720;19.5;20.3;1023.7;1018.7;35.1;39.3;-61.0;-0.8000000000000007;5.0;-4.1999999999999996;
```

Esta plantilla escucha los siguientes eventos de cambio de estado:

- Entidad: sensor.tasmota_bme680_humidity
- Entidad: sensor.tasmota_bme680_humidity_2
- Entidad: sensor.tasmota_bme680_pressure
- Entidad: sensor.tasmota_bme680_pressure_2
- Entidad: sensor.tasmota_bme680_temperature
- Entidad: sensor.tasmota_bme680_temperature_2
- Entidad: sensor.tasmota_mhz19b_carbondioxide
- Entidad: sensor.tasmota_mhz19b_carbondioxide_2

Figura 12. Código escrito en la plantilla y su salida

Al escribir: `{{states.sensor.tasmota_mhz19b_carbondioxide.state}}` , se está llamando al estado del sensor MH-Z19B Integrado con Tasmota y lo imprime

como una cadena (*string*) de caracteres. Sin embargo, para poder hallar la diferencia, debe convertirse a decimal (*float*, punto flotante), y así operar con él. Para pasarlo a decimal, se escribe *|float* tras la llamada al estado. Para hallar la diferencia, simplemente se resta:

```
{{states.sensor.tasmota_bme680_temperature.state |float -
states.sensor.tasmota_bme680_temperature_2.state |float}}.
```

Obsérvese que, todo lo que queramos que sea llamado desde un sensor o cualquier otra entidad, debemos escribirlo entre "{{ }}". Para los condicionales y otras operaciones lógicas, escribiremos entre "{% %}".

Los dispositivos se representan con entidades. Estas entidades poseen:

- ID
- Nombre
- Icono.
- Estado.
- Atributos.

Entidad	Estado	Atributos <input checked="" type="checkbox"/>
Filtrar entidades <input type="text" value="tasmota"/>	Filtrar estados <input type="text" value=""/>	Filtrar atributos <input type="text" value=""/>
<input type="checkbox"/> sensor.tasmota_bme680_dewpoint <input type="info"/> Tasmota BME680 DewPoint	3.6	state_class: measurement unit_of_measurement: °C device_class: temperature icon: mdi:weather-rainy friendly_name: Tasmota BME680 DewPoint

Figura 13. Imagen obtenida directamente de Home Assistant donde se han filtrado las entidades que contengan "*tasmota*" en su designación. En las siguientes columnas, pueden visualizarse los Estados y los Atributos que estén asignados a esa entidad.

5.4. Muestreo

Una de las prácticas obligatorias a la hora de tomar datos es la manipulación de estos: obtener gráficas y medidas estadísticas. Para ello, la versión supervisada de Home Assistant permite la instalación de Add-On's que facilitan la labor. En

nuestro caso, estamos usando un sistema operativo que no admite la versión supervisada. Por ese motivo, para usar otro tipo de servicios, se necesita modificar el código interno de Home Assistant (*configuration.yaml*).

Al haber instalado FileBrowser, se puede navegar por el sistema de la Raspberry usando el navegador web instalado por defecto en el sistema operativo. Si se ingresa al directorio donde se haya instalado Home Assistant (en nuestro caso `/home/grupotar/docker/homeassistant/`), se observa que ahí se encuentra el mencionado *configuration.yaml*. Se encuentra escrito en Yaml la primera vez que se abre sin haberlo editado:

```
# Loads default set of integrations. Do not remove.
default_config:

# Load frontend themes from the themes folder
frontend:
  themes: !include_dir_merge_named themes

# Text to speech
tts:
  - platform: google_translate

automation: !include automations.yaml
script: !include scripts.yaml

scene: !include scenes.yaml
```

Para poder guardar nuestros datos muestreados en un fichero, debe buscarse una integración que sea adecuada para ello. En nuestro caso, se usó el servicio "file" que permite guardar las notificaciones en un fichero local. Para integrar el servicio, se añaden las siguientes líneas al final del fichero anterior:

notify:

- platform: file #La plataforma a usar es 'file'

name: filenotify #El nombre que le demos al servicio.

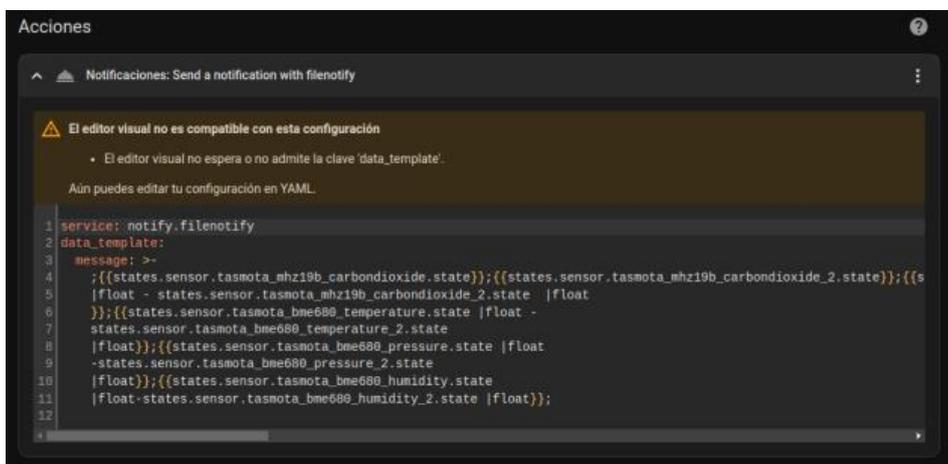
filename: /config/file.csv #La dirección y el nombre del fichero donde guardaremos los datos. Si no existiera, se crearía automáticamente.

timestamp: true #Añade una etiqueta de tiempo a la medida, sabiendo en qué momento se tomó la muestra. Incluye milésimas de segundo, que más tarde se separarán en Excel.

De esta forma, tenemos un servicio que nos permite guardar datos en un fichero. Para configurar la forma en la que se van a guardar nuestros datos en el fichero y dependiendo de qué evento se va a desencadenar una toma de datos, se debe crear una automatización. Entrando en Home Assistant/Ajustes/Automatizaciones y escenas, puede crearse la primera automatización.

En desencadenantes, se localiza 'Time pattern trigger', donde puede escribirse el patrón de tiempo con el que se guardarán las notificaciones. En este caso, se usa /10 en segundos (nótese la barra previa al número). Escribiéndolo de esa forma, nos aseguramos de que muestrea cada 10 segundos, y no en los 10 primeros segundos de cada hora.

En acciones, se escribe lo siguiente:



```
1 service: notify.filenotify
2 data_template:
3   message: >-
4     ;{{states.sensor.tasmota_mhz19b_carbondioxide.state}};{{states.sensor.tasmota_mhz19b_carbondioxide_2.state}};{{s
5     |float - states.sensor.tasmota_mhz19b_carbondioxide_2.state |float
6     }};{{states.sensor.tasmota_bme680_temperature.state |float -
7     states.sensor.tasmota_bme680_temperature_2.state
8     |float}};{{states.sensor.tasmota_bme680_pressure.state |float
9     -states.sensor.tasmota_bme680_pressure_2.state
10    |float}};{{states.sensor.tasmota_bme680_humidity.state
11    |float-states.sensor.tasmota_bme680_humidity_2.state |float}};
12
```

Figura 14. Estructura de la acción

Donde la estructura del mensaje es la siguiente:

```
message:
;{{states.sensor.tasmota_mhz19b_carbondioxide.state}};{{state
s.sensor.tasmota_mhz19b_carbondioxide_2.state}};{{states.sens
or.tasmota_bme680_temperature.state}};{{states.sensor.tasmota
_bme680_temperature_2.state}};{{states.sensor.tasmota_bme680_
pressure.state}};{{states.sensor.tasmota_bme680_pressure_2.st
ate}};{{states.sensor.tasmota_bme680_humidity.state}};{{state
s.sensor.tasmota_bme680_humidity_2.state}};{{states.sensor.ta
smota_mhz19b_carbondioxide.state
    |float -
states.sensor.tasmota_mhz19b_carbondioxide_2.state |float
}};{{states.sensor.tasmota_bme680_temperature.state
    |float -
states.sensor.tasmota_bme680_temperature_2.state
|float}};{{states.sensor.tasmota_bme680_pressure.state
    |float -states.sensor.tasmota_bme680_pressure_2.state
|float}};{{states.sensor.tasmota_bme680_humidity.state
    |float-states.sensor.tasmota_bme680_humidity_2.state
|float}};
```

Se está llamando al servicio *notify* con el alias que se le puso. El mensaje será la estructura escrita que se guarde en el fichero "file.csv", donde se van a estar guardando las muestras separadas por ";". Se estará generando entonces un fichero con la misma estructura que la salida de la plantilla (además de haberle indexado el tiempo actual), exportable a una hoja de cálculo. Tiene la siguiente forma:

```
2023-02-03T12:17:30.007360+00:00 ;800;884;19.6;19.4;164.47;122.07;37.9;40.0
2023-02-03T12:17:40.006963+00:00 ;800;884;19.6;19.4;164.47;122.07;37.9;40.0
2023-02-03T12:17:50.007528+00:00 ;800;884;19.6;19.4;164.47;122.07;37.9;40.0
2023-02-03T12:18:00.005717+00:00 ;800;884;19.6;19.4;164.47;122.07;37.9;40.0
2023-02-03T12:18:10.006719+00:00 ;800;884;19.6;19.4;164.47;122.07;37.9;40.0
2023-02-03T12:18:20.006872+00:00 ;800;884;19.6;19.4;164.47;122.07;37.9;40.0
2023-02-03T12:18:30.012492+00:00 ;800;884;19.6;19.4;164.47;122.07;37.9;40.0
2023-02-03T12:18:40.005773+00:00 ;800;884;19.6;19.4;164.47;122.07;37.9;40.0
2023-02-03T12:18:50.007578+00:00 ;800;884;19.6;19.4;164.47;122.07;37.9;40.0
2023-02-03T12:19:00.007333+00:00 ;800;884;19.6;19.4;164.47;122.07;37.9;40.0
2023-02-03T12:19:10.006923+00:00 ;800;884;19.6;19.4;164.47;122.07;37.9;40.0
2023-02-03T12:19:20.007383+00:00 ;800;884;19.6;19.4;164.47;122.07;37.9;40.0
2023-02-03T12:19:30.007443+00:00 ;800;884;19.6;19.4;164.47;122.07;37.9;40.0
2023-02-03T12:19:40.007061+00:00 ;800;884;19.6;19.4;164.47;122.07;37.9;40.0
2023-02-03T12:19:50.007022+00:00 ;800;884;19.6;19.4;164.47;122.07;37.9;40.0
2023-02-03T12:20:00.006983+00:00 ;800;884;19.6;19.4;164.47;122.07;37.9;40.0
2023-02-03T12:20:10.009050+00:00 ;800;884;19.6;19.4;164.47;122.07;37.9;40.0
2023-02-03T12:20:20.006599+00:00 ;800;884;19.6;19.4;164.47;122.07;37.9;40.0
```

Figura 15. Ejemplo de respuesta que se obtiene

Al generar un archivo .csv con los datos separados por “;”, puede exportarse fácilmente a una hoja de cálculo.

6. Control remoto

El control remoto se realizará de dos formas distintas dependiendo de si se quiere controlar la Raspberry en sí o si se desea mandar órdenes predefinidas.

6.1. VNC

VNC es un servicio que viene integrado en el sistema operativo instalado por defecto. Puede instalarse, además, en otro dispositivo móvil u ordenador para el control remoto. La persona que quiera usarlo debe registrarse con una cuenta e iniciar sesión en ambos dispositivos.

VNC permite la transferencia de archivos, pudiendo manualmente de esta forma enviar el archivo de muestreo, ubicado en la dirección establecida anteriormente. Para ello, se debe ingresar en el apartado "Transferencia de archivos" del menú desplegable.

6.2. Telegram

Con Telegram, se puede crear un bot que se encargue de mandar órdenes predefinidas a Home Assistant. Se describirán los pasos para la creación de un bot de Telegram.

Los pasos previos son compartidos con la creación de cualquier Bot de Telegram. Para la creación de Bots, abrimos una conversación con [BotFather](#), un asistente de Telegram con el que podemos generar Bots.

Para comenzar, se escribe en el chat /start. A nosotros nos interesa la creación de un nuevo Bot y, por lo tanto, se escribe el comando /newbot, y después se selecciona el nombre que se quiera y su nombre de usuario.

Por último, se da un link para poder acceder a la conversación con el Bot y una clave, que se usará más adelante.

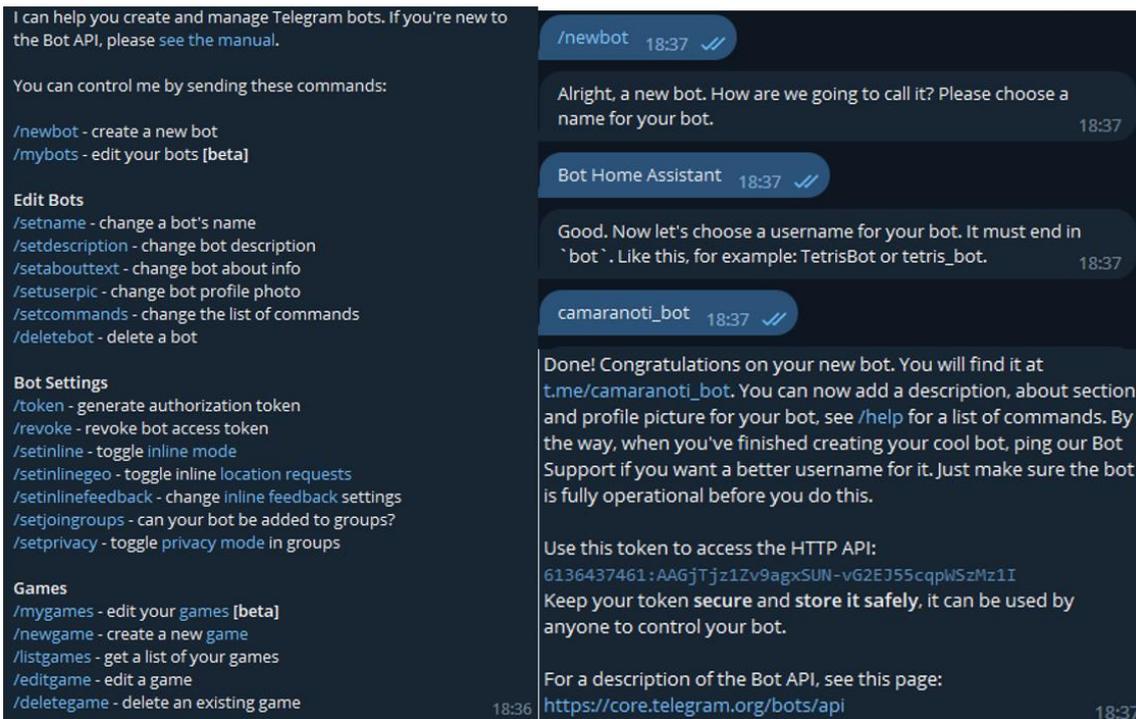


Figura 16. Chat con el creador de bots

Iniciando una conversación el bot creado por medio del enlace generado, se escribe el comando `/start` y un mensaje cualquiera. Desde este momento, pueden seguirse dos caminos: usar el bot desde un chat privado o integrarlo en un grupo. Se usará la segunda opción, siendo fácil que la información de la cámara atmosférica llegue a más usuarios.

Se crea un grupo de Telegram, añadiendo al bot entre sus integrantes.

Se necesita la ID del grupo, siendo una serie de números que referencian al chat, obtenibles de la siguiente manera:

- Se vuelve a iniciar una conversación con un nuevo bot: Chat ID Bot.
- Se añade a nuestro grupo y, al salir de este, dará la ID de nuestro grupo. Empieza por un signo "-". Si fuera un chat privado, sería un número positivo.

Para la integración de Telegram, se añaden manualmente las siguientes líneas al archivo de `configuration.yaml`

```
telegram_bot:  
  - platform: polling  
    api_key: 6136437461:AAGjTjz1Zv9agxSUN-vG2EJ55cqpWSzMz1I  
#Se trata de la clave final proporcionada por el BotFather.  
    allowed_chat_ids: -[CHAT ID] #Aquí se debe especificar la ID  
proporcionada por el bot anteriormente, pudiendo poner varias (los chats que queremos que  
sean accesibles por el bot)
```

El código para las automatizaciones de Telegram puede consultarse en el fichero *automations.yaml*. El control por Telegram se configura más fácilmente desde el editor de automatizaciones del propio Home Assistant. Se muestran a continuación algunas automatizaciones para el control remoto.

6.2.1. Enviar estado actual de los sensores

Para ello, se debe ingresar en el apartado de automatizaciones dentro de Home Assistant y crear una nueva automatización de la siguiente manera:

- **Desencadenantes:** se establece que cada 6 horas se enviará el estado de la cámara al chat de Telegram. Sin embargo, se podrá llamar al estado actual mediante un comando y por un botón que aparecerá en una botonera en el chat.

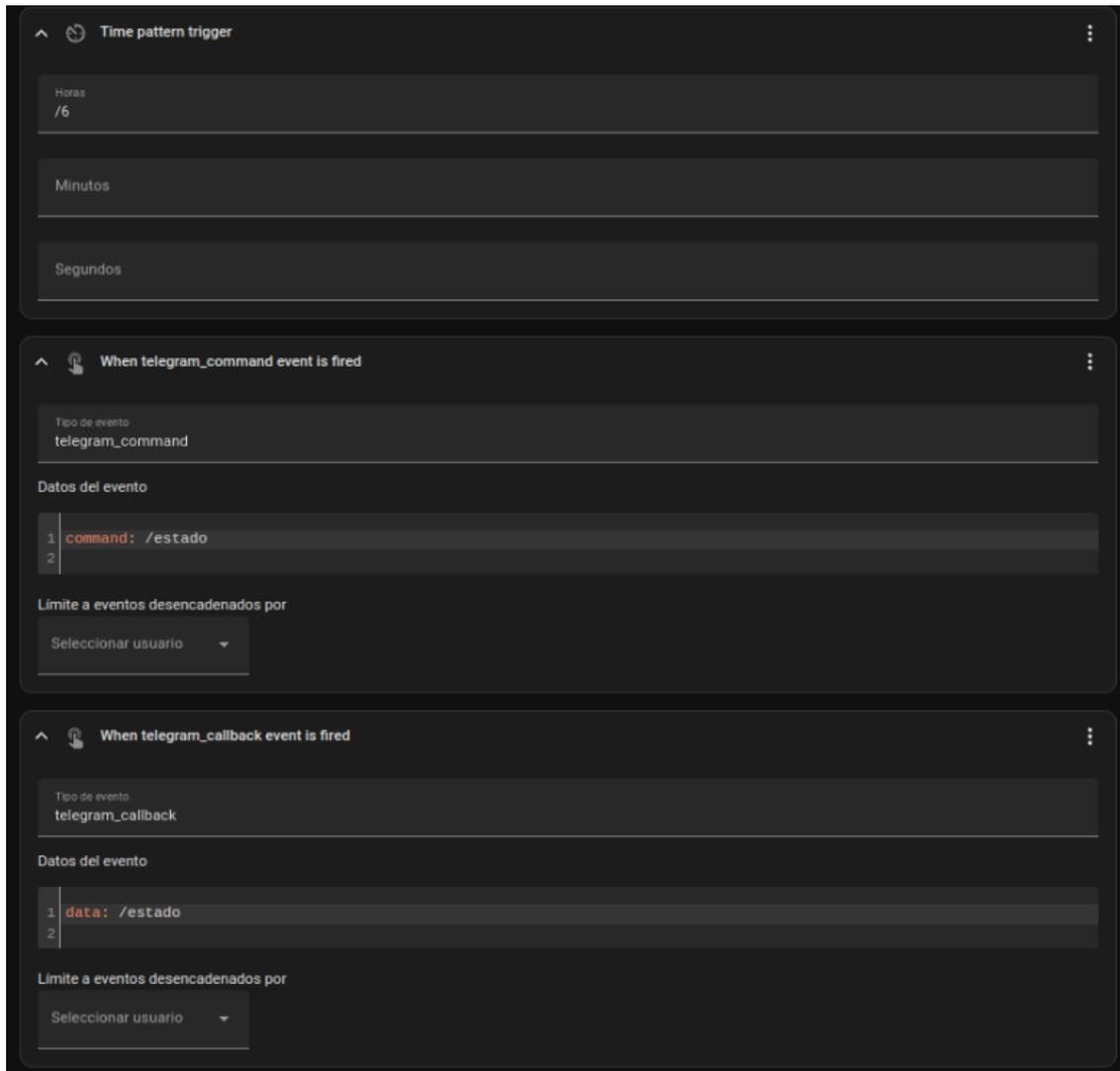


Figura 17. Desencadenantes para enviar el estado actual de la cámara por Telegram

- Acciones: lo que se quiere es que se envíe el estado de todos los sensores como mensaje. Para ello, usamos la siguiente estructura (debe editarse a mano):

```
Notificaciones: Send a notification with bot_telegram

⚠ El editor visual no es compatible con esta configuración
  • Las plantillas no son compatibles con el editor visual
  Aún puedes editar tu configuración en YAML.

1 service: notify.bot_telegram
2 data:
3   title: "**ESTADO ACTUAL || {{states.sensor.date_time.state}}*"
4   message: |2-
5     TEMPERATURA 125: {{states.sensor.tasmota_bme680_temperature.state}} °C
6     TEMPERATURA 121: {{states.sensor.tasmota_bme680_temperature_2.state}} °C
7     CO2 125: {{states.sensor.tasmota_mhz19b_carbondioxide.state}} ppm
8     CO2 121: {{states.sensor.tasmota_mhz19b_carbondioxide_2.state}} ppm
9     Humedad 125: {{states.sensor.tasmota_bme680_humidity.state}} %
10    Humedad 121: {{states.sensor.tasmota_bme680_humidity_2.state}} %
11  data:
12    inline_keyboard:
13      - Reiniciar Home Assistant:/reinicio
14      - Borrar Muestreo:/borrar_muestreo
15      - Enviar Muestreo:/enviar_muestreo
16      - Iniciar Muestreo:/iniciar_muestreo
17      - Pausar Muestreo:/pausar_muestreo
18      - Estado actual:/estado
19
```

Figura 18. Acciones a llevar a cabo cuando se pida el estado actual, siendo las últimas líneas una botonera con comandos asociados

6.2.2. Enviar el fichero de valores muestreados

- Desencadenantes: solo se enviará el muestreo cuando se pida, es decir, escribiendo un comando o pulsando un botón de la botonera:

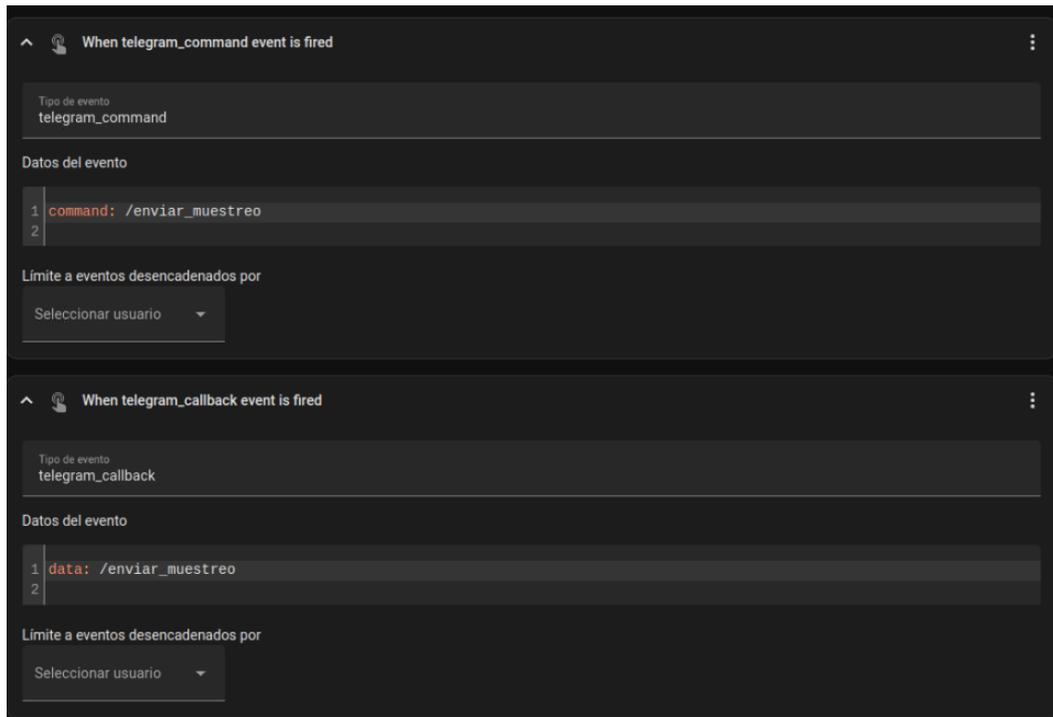


Figura 19. Desencadenantes para enviar el muestreo de la cámara por Telegram

- Acciones: se envía el archivo, especificando la ruta hasta él. Se realiza mediante una notificación que permite el envío de archivos.

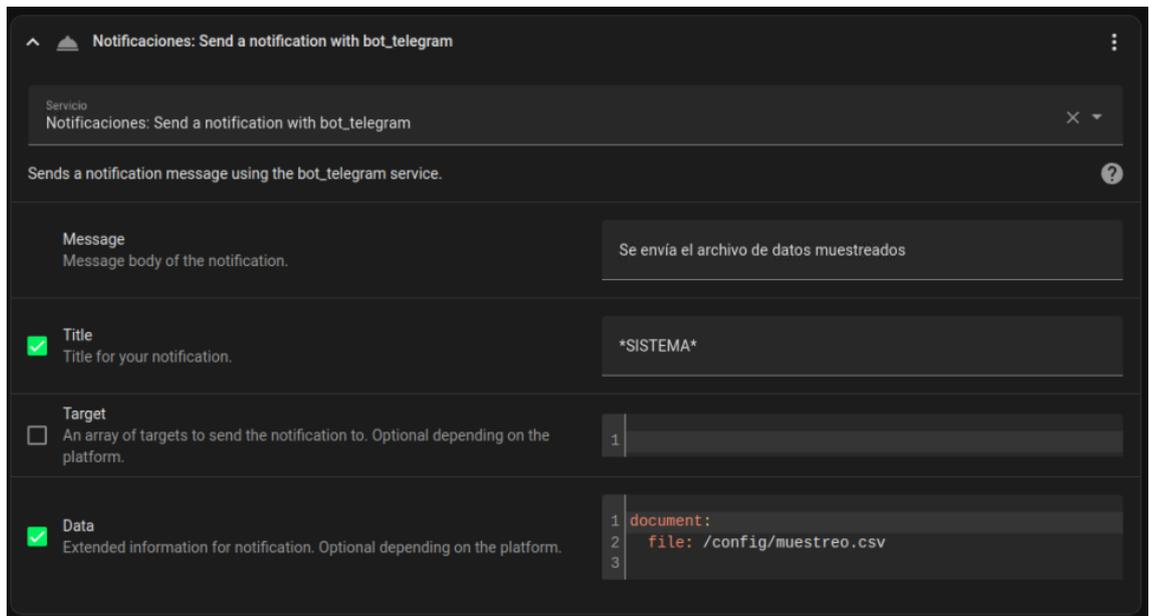


Figura 20. Acciones a llevar a cabo cuando se pida el envío del archivo de datos muestreados

7. ENSAYOS

En este apartado se describen los ensayos realizados.

7.1. Ensayos de calibración inicial

Se realizaron 3 ensayos sin agentes externos, es decir, simplemente con las dos cámaras de metacrilato y sensores en ambas. De esta forma, se pudo comprobar el error entre parejas de sensores y cómo se comporta la atmósfera dentro de la cámara.

- MH-Z19B: CO₂.
- BME680: Temperatura, humedad, presión y punto de rocío.

7.1.1. Calibración inicial 1

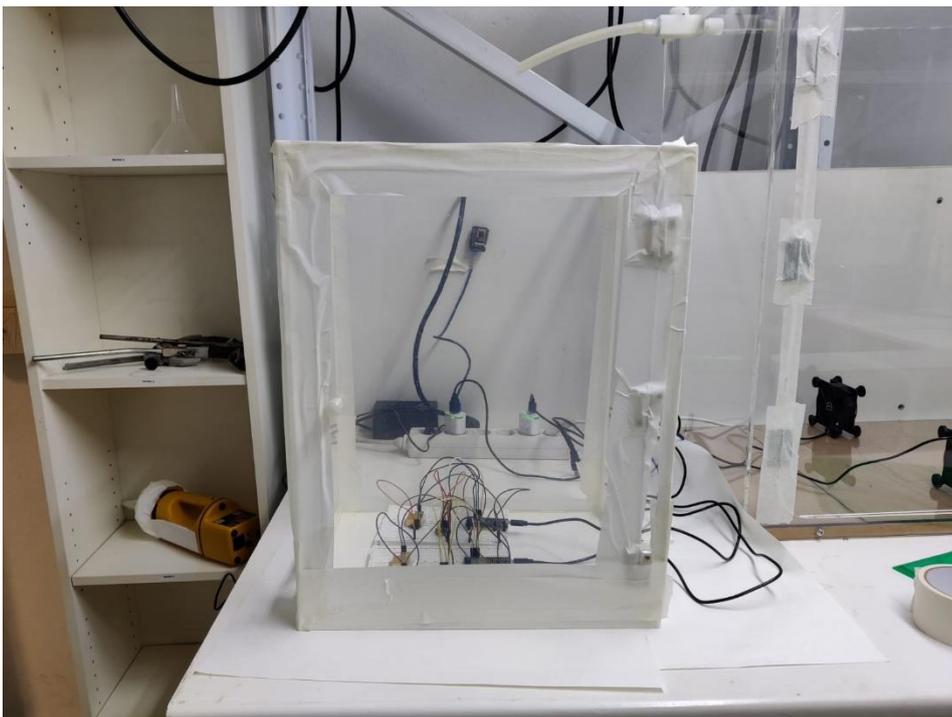


Figura 21. Cámara sellada

- Cámara sellada.
- Sensores en la misma cámara, que será una de las de menor tamaño.
- Tiempo transcurrido: 21 horas y 16 minutos.

- Tiempo de muestreo: 10 segundos.

Los resultados del ensayo fueron los siguientes:

- Media de la diferencia de temperatura: 0.28°C.
- Media de la diferencia de dióxido de carbono: 12.77 ppm. La diferencia se hace nula en varios momentos y, además, en los últimos momentos del ensayo se anuló completamente.
- Rango de variación de la medida de dióxido de carbono: 112 ppm.
- Máxima diferencia de medida de dióxido de carbono: 70 ppm.
- Mínima diferencia de medida de dióxido de carbono: -42 ppm.
- Media de la diferencia de humedad relativa: 2.18%. Este valor es negativo, indicando que el primer sensor es, generalmente, menor que el segundo.

En vista de los resultados de este ensayo, se aceptarían las medidas tomadas. Se observó también que, para el final de la experiencia, la diferencia de señal entre los dos sensores MHZ19B se estabilizó y se anuló.

7.1.2. Calibración inicial 2

Este ensayo se realizó para corroborar los datos del anterior ensayo y medir el tiempo de estabilización. La experiencia tuvo una duración aproximada de 3 días. El tiempo de estabilización es de poco más de 24h.

Para otra medida importante, la humedad relativa, se observa que el error fluctúa y se mantiene entre valores bajos, pero nunca se anula.

7.1.3. Calibración inicial 3

Este último ensayo se realizó para observar el comportamiento de los sensores de CO₂ de la cámara al manipular las puertas.

- 12:29: Se abre la cámara.
- 13:04: Se cierra la cámara. Habrían transcurrido 5 minutos.
- 13:05: Cámara de medida queda sellada.

Se observan los resultados durante 2h de muestreo de CO₂, iniciando cuando se abre la puerta de la cámara. Como era de esperar, ambas concentraciones medidas aumentan vigorosamente, sin embargo, se observa que no tienen el mismo tiempo de respuesta. Hay una diferencia de respuesta de 2 minutos entre ambos sensores, que se deberá tener en cuenta para experiencias futuras.

7.2. Otros ensayos

Además de los ensayos anteriores, se realizaron una serie de ensayos adicionales. El objetivo fue comprobar el comportamiento del dispositivo en diferentes situaciones.

- **Ensayo con cámaras abiertas:** se pretendía observar el comportamiento de los sensores al abrir las cámaras que, en esta ocasión, estaban cerradas herméticamente. No hay un cambio significativo en las medidas
- **Ensayo con bomba de vacío:** se pretendía observar la influencia de la bomba de aire disponible en el laboratorio sobre las cámaras.
 - Tiempo de aspiración: 10 minutos.
 - Caudal: 1'6 L/s.

El caudal del aire es demasiado pequeño como para notar la aspiración en la cámara más alejada. En la cámara donde se conecta directamente la bomba existe una leve subida de CO₂.



Figura 22. Diagrama del ensayo

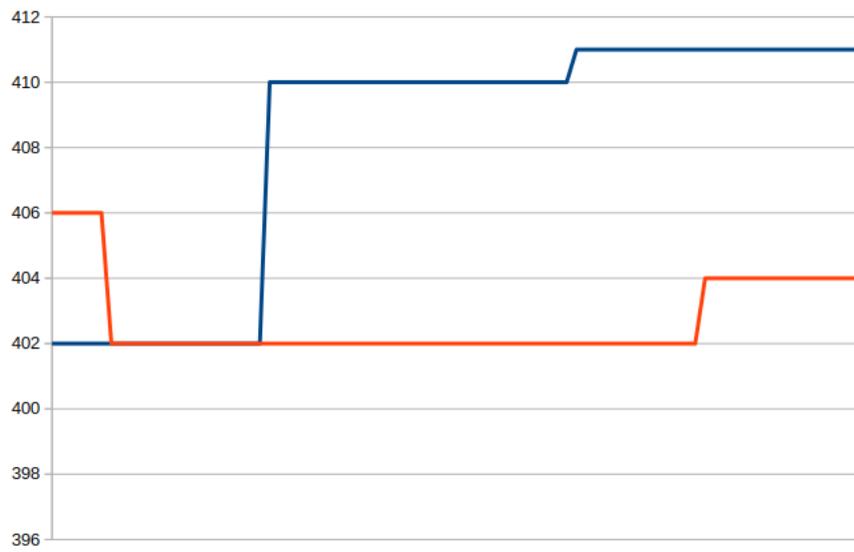


Figura 23. Medidas de CO₂ en ambas cámaras (ppm). En la cámara donde se conecta directamente la bomba existe una leve subida de CO₂ (naranja).

8. CONCLUSIONES

- La automatización de la cámara se ha llevado a cabo con la ayuda indispensable del profesor Miguel Ángel Leal Díaz, del departamento de Tecnología Electrónica.
- Se ha comprobado, con numerosos ensayos, que el dispositivo funciona correctamente y además presenta gran accesibilidad. De esta forma no solo es un dispositivo funcional, sino que también es un dispositivo fácil de usar por cualquier persona sin necesidad de estar capacitada.
- Inicialmente, su uso será:
 - ➔ Diseño del boulevard saludable de San Francisco Javier en base a los ensayos del Índice de Salud Ambiental (ISA) más favorables con plantas, arbustos y árboles introducidos en el diseño, usando los datos experimentales obtenidos juntos con las condiciones estéticas y forestales correspondientes.
 - ➔ Estudio experimental de la capacidad de la salud ambiental o Índice de la Salud Ambiental (ISA) de aquellas calles existentes en la ciudad que se quieran mejorar, en base a la evaluación de la situación inicial en la calle, la detección experimental de los puntos débiles en el ISA y el diseño de las compensaciones necesarias para mejorar el ISA o la salud ambiental en la propia calle.
- Entre otros usos que se le puede dar a la cámara están todos aquellos que requieran la medición de algún componente:
 - ➔ Optimización de todo tipo de instalaciones verdes, como paredes y techos verdes. Podrá observarse cuál es el tipo de masa verde más adecuada además de ver la eficiencia según las condiciones ambientales y el tipo de contaminante predominante.
 - ➔ Medición de emisión de contaminantes por determinadas muestras.
- Por la forma de desarrollar el dispositivo, existe la posibilidad de no solo medir contaminantes atmosféricos, sino medir también parámetros de muestras sólidas y líquidas si se tienen los sensores adecuados.

Bibliografía

1. C. C. Wolverton, et al. (2009). Interior plants: their influence on airborne microbes inside energy-efficient buildings. *HortScience*, 44(2), 132-137
2. Delgadillo-López, Angélica Evelin, González-Ramírez, César Abelardo, Prieto-García, Francisco, Villagómez-Ibarra, José Roberto, & Acevedo-Sandoval, Otilio. (2011). Fitorremediación: una alternativa para eliminar la contaminación. *Tropical and subtropical agroecosystems*, 14(2), 597-612. Recuperado en 27 de enero de 2024, de http://www.scielo.org.mx/scielo.php?script=sci_arttext&pid=S1870-04622011000200002&lng=es&tlng=es.
3. H. A. Al-Dhabi, et al. (2014). Potential of *Chamaedorea seifrizii* for Indoor Air Biofiltration of Volatile Organic Compounds. *The International Journal of Phytoremediation*, 16(9), 870-883
4. Janhäll, S. (2015). Review on urban vegetation and particle air pollution—Deposition and dispersion. *Atmospheric environment*, 105, 130- 137.
5. M. A. Awang, et al. (2015). Phytoremediation of indoor air by potted plants. *Environmental Science and Pollution Research*, 22(21), 16514-16527
6. Poorter, Hendrik; Niinemets, Ülo; Poorter, Lourens; Wright, Ian J.; Villar, Rafael (2009). "Causes and consequences of variation in leaf mass per area (LMA): a meta-analysis". *New Phytologist*. 182 (3): 565–588. doi:10.1111/j.1469-8137.2009.02830.x. ISSN 1469-8137. PMID 19434804. S2CID 6350349.
7. Sternberg T, Viles H, Cathersides A, Edwards M. Dust particulate absorption by ivy (*Hedera helix* L) on historic walls in urban environments. *Sci Total Environ*. 2010 Dec 1;409(1):162-8. doi: 10.1016/j.scitotenv.2010.09.022. Epub 2010 Oct 13. PMID: 20947134.
8. Tharrey M, Darmon N. Urban collective garden participation and health: a systematic literature review of potential benefits for free-living adults. *Nutr Rev*. 2021 Dec 8;80(1):6-21. doi: 10.1093/nutrit/nuaa147. PMID: 33997887.

9. Tomašević, M., Rajšić, S., Đorđević, D. et al. Heavy metals accumulation in tree leaves from urban areas. *Environ Chem Lett* 2, 151–154 (2004). <https://doi.org/10.1007/s10311-004-0081-8>
10. Tzoulas, K., Galan, J., Venn, S. et al. A conceptual model of the social–ecological system of nature-based solutions in urban environments. *Ambio* 50, 335–345 (2021). <https://doi.org/10.1007/s13280-020-01380-2>
11. Van den Bosch M, Ode Sang Å. Urban natural environments as nature-based solutions for improved public health - A systematic review of reviews. *Environ Res.* 2017 Oct;158:373-384. doi: 10.1016/j.envres.2017.05.040. Epub 2017 Jul 4. PMID: 28686952.
12. Vujcic M, Tomicevic-Dubljevic J, Grbic M, Lecic-Tosevski D, Vukovic O, Toskovic O. Nature based solution for improving mental health and well-being in urban areas. *Environ Res.* 2017 Oct;158:385-392. doi: 10.1016/j.envres.2017.06.030. Epub 2017 Jul 5. PMID: 28688269.
13. Walsh, É., Kuehnhold, H., O'Brien, S., Coughlan, N. E., & Jansen, M. A. K. (2021). Light intensity alters the phytoremediation potential of *Lemna minor*. *Environmental Science and Pollution Research*, 28(13), 16394–16407. doi:10.1007/s11356-020-11792-y
14. Wei, X., Lyu, S., Yu, Y., Wang, Z., Liu, H., Pan, D., & Chen, J. (2017). Phylloremediation of air pollutants: exploiting the potential of plant leaves and leaf-associated microbes. *Frontiers in plant science*, 8, 1318.
15. Wolch, Jennifer & Byrne, Jason & Newell, Joshua. (2014). Urban green space, public health, and environmental justice: The challenge of making cities 'just green enough'. *Landscape Urban Plann.* 125. 10.1016/j.landurbplan.2014.01.017.
16. Wood R.A., Orwell R.L. and Burchett M.D. (1999). Study of absorption of VOC's by commonly used indoor plants. University of Technology, Sydney.
17. Wróblewska K, Jeong BR. Effectiveness of plants and green infrastructure utilization in ambient particulate matter removal. *Environ Sci Eur.*

2021;33(1):110. doi: 10.1186/s12302-021-00547-2. Epub 2021 Sep 25.
PMID: 34603905; PMCID: PMC8475335.

18. Zhang, X., Zhou, P., Zhang, W., Zhang, W., and Wang, Y. (2013). Selection of landscape tree species of tolerant to sulfur dioxide pollution in subtropical China. *Open J. Forestry* 3, 104–108. doi: 10.4236/ojf.2013.34017