

# Manual del desarrollador

## 1.5.0

Envira Sostenible S.A.

Nanoenvi IAQ

ENVIRA

Copyright © 2021 Envira Sostenible S.A.

Quedan reservados todos los derechos a la modificación y corrección de los contenidos de este documento sin notificaciones previas. Estas especificaciones aplican sobre los pedidos recibidos. Envira Sostenible S.A. no acepta responsabilidades derivadas de posibles erratas o información no incluida en este documento. Todos los derechos sobre el contenido, imágenes e ilustraciones incluidos en este documento quedan reservados. Prohibida la reproducción, transmisión o utilización, total o parcial, de este documento o sus contenidos, por terceras partes sin el consentimiento de Envira Sostenible S.A.

## Tabla de contenidos

1. Control de revisiones .....	5
2. Alcance del documento .....	6
3. Introducción .....	7
4. Interfaces .....	8
4.1. Envío de medidas de sensores vía MQTT .....	8
4.1.1. Formato de mensaje de medidas .....	8
4.2. API de configuración remota .....	9
4.2.1. Funcionamiento de configuración remota .....	9
4.2.2. Métodos públicos expuestos .....	11
4.2.3. Descripción métodos públicos .....	11

## Lista de tablas

1.1. Control de revisiones .....	5
----------------------------------	---

# Capítulo 1. Control de revisiones

Tabla 1.1. Control de revisiones

Versión	Cambios
V1.3	Creada guía de integración
V1.4	Documentado sensor de ruido
V1.5	Añadidos métodos para resetear, factory reset y poner en modo configuración. Corrección de erratas.

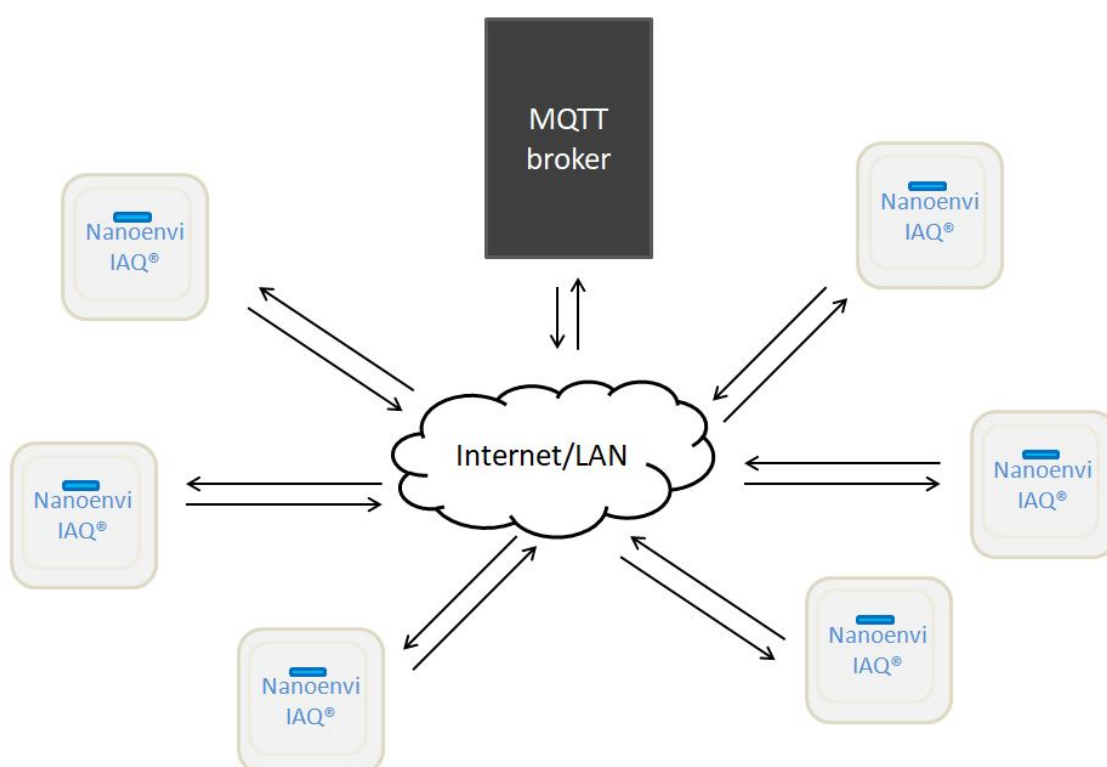
## Capítulo 2. Alcance del documento

El presente documento detalla aspectos de integración de Nanoenvi IAQ™ con broker MQTT. Se documentan las interfaces de comunicación, formato de mensaje y aspectos relacionados.

## Capítulo 3. Introducción

Nanoenvi IAQ™ ha sido desarrollado bajo la filosofía #CloudFree. Es un dispositivo que ofrece una comunicación abierta y es integrable con plataformas de visualización o explotación de datos sin depender de software propietario o suscripciones a servicios de pago. Para ello, utiliza el protocolo MQTT, que es un protocolo de comunicación abierto y orientado a la transferencia de datos de sensores. A lo largo de los últimos años se ha posicionado como uno de los protocolos utilizados en sistemas IoT por su sencillez, ligereza y facilidad de integración.

De este modo, el único requerimiento de Nanoenvi IAQ™ es el uso de un broker MQTT. A partir de este punto, el integrador puede consumir los datos con la libertad de explotarlos como desee: bien sea desde un BMS o desde una plataforma de visualización de datos.



Durante su instalación, Nanoenvi IAQ™ requiere la configuración de las credenciales de la red WiFi usada para el envío de datos, dirección del broker MQTT y parámetros de acceso al mismo. Una vez configurado, envía las medidas obtenidas al broker MQTT y ofrece una API para facilitar su gestión, consultar su estado y notificar nuevas actualizaciones de firmware.

Para más detalles sobre el proceso de instalación, configuración y operación, consulte el manual de usuario de Nanoenvi IAQ™.

# Capítulo 4. Interfaces

## 4.1. Envío de medidas de sensores vía MQTT

El envío de datos de medidas de los diferentes sensores se realiza a través del protocolo MQTT, al broker configurado cada T segundos, siendo T el periodo de envío de datos establecido. A continuación se detalla el formato del mensaje MQTT recibido por el broker:

### 4.1.1. Formato de mensaje de medidas

#### 4.1.1.1. Formato de medidas

A partir de la versión de firmware V1.2, las medidas de sensores se envían al topic configurado en formato SenML (Sensor Measurement Lists) tal y como define la RFC8428. [<https://tools.ietf.org/html/rfc8428>] A continuación, se muestra el formato de mensaje.

Nótese que en función de los sensores incluidos en el dispositivo algunos de los campos no están disponibles. Por ejemplo en el caso de no disponer de sensor de ruido, el campo correspondiente no estará presente en el mensaje. Ídem en el caso de la ausencia de sensor de CO.

```
{
  "device_info": {
    "uuid": "<UUID>",
    "fw_ver": "<versión de firmware>"
  },
  "measures": [
    {
      "bn": "",
      "bver": "<versión de Nanoenvo IAQ fimware>"
    },
    {
      "n": "co2",
      "u": "ppm",
      "v": "<valor numérico de medida de CO2>"
    },
    {
      "n": "voc",
      "u": "ppb",
      "v": "<valor numérico de medida de VOC>"
    },
    {
      "n": "co",
      "u": "ppm",
      "v": "<valor numérico de medida de CO>"
    },
    {
      "n": "noise",
      "u": "%",
      "v": "<valor numérico de medida de nivel de
        ruido (expresada como porcentaje)>"
    },
    {
      "n": "pm10",
      "u": "ug/m3",
      "v": "<valor numérico de medida de PM10>"
    },
    {
      "n": "pm2.5",
      "u": "ug/m3",
      "v": "<valor numérico de medida de PM2.5>"
    }
  ]
}
```



```

    "n": "temp",
    "u": "Cel",
    "v": <valor numérico de medida de T>
  },
  {
    "n": "hum",
    "u": "%RH",
    "v": <valor numérico de medida de RH>
  },
  {
    "n": "prb",
    "u": "mmHg",
    "v": <valor numérico de medida de P>
  },
  {
    "n": "pm1",
    "u": "ug/m3",
    "v": <valor numérico de medida de PM1>
  },
  {
    "n": "pm4",
    "u": "ug/m3",
    "v": <valor numérico de medida de PM4>
  },
  {
    "n": "iaqi",
    "u": "count",
    "v": <valor numérico IAQI (Indoor Air Quality Index) calculado>
  },
  {
    "n": "tci",
    "u": "count",
    "v": <valor numérico TCI (Thermal Comfort Index) calculado>
  },
  {
    "n": "eiaqi",
    "u": "count",
    "v": <valor numérico EIAQI (Environmental Indoor Air Quality
      Index) calculado>
  }
]
}

```

El mensaje contiene el UUID del dispositivo y la versión de firmware del mismo. Las entradas correspondientes a cada medida contienen el nombre abreviado de la variable medida, las unidades de la medida según lo definido en la RFC8428 [<https://tools.ietf.org/html/rfc8428>] y el valor de la medida.



Las medidas de partículas en suspensión se presentan por defecto de forma acumulativa. Es decir, el valor de concentración de PM10 incluye la concentración de partículas con menores tamaños (PM4, PM2.5 y PM1), el valor de concentración de PM4 incluye la concentración de PM2.5 y PM1 y PM2.5 incluye también a PM1.

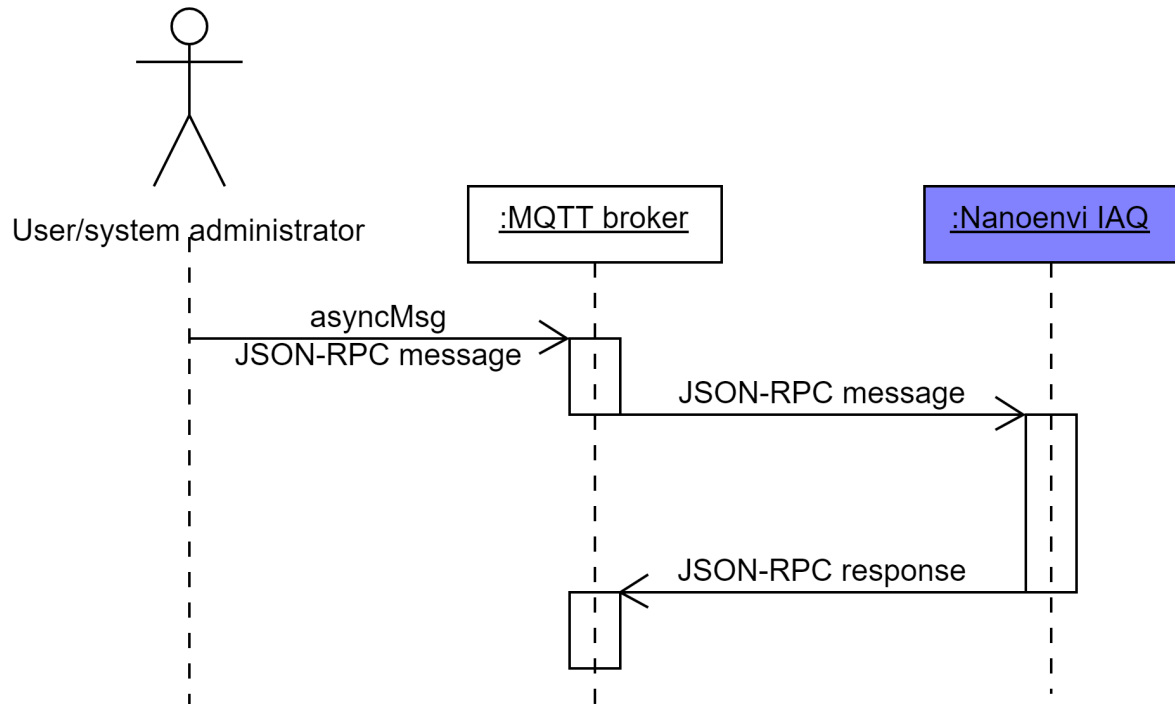
Si se desea que estas medidas se den por intervalo (de forma no acumulativa), se debe enviar un mensaje de configuración al dispositivo (ver documentación sobre JSON-RPC).

## 4.2. API de configuración remota

### 4.2.1. Funcionamiento de configuración remota

Cuando Nanoenvi IAQ™ se encuentra en *modo comunicación*, se suscribe al topic `IAQ_RPC/<UUID>` en el broker MQTT configurado. A través de este topic, el dispositivo puede recibir mensajes de configuración. Las respuestas a estos mensajes son publicadas en el topic `IAQ_RESP/<UUID>`.

La siguiente imagen se corresponde con el diagrama de mensajes entre Nanoenvi IAQ™ y el broker MQTT, cuando el usuario o administrador del sistema lanza mensajes de configuración remotos:



El formato de mensajes de configuración se basa en el estándar JSON-RPC 1.0 ([https://www.jsonrpc.org/specification\\_v1](https://www.jsonrpc.org/specification_v1)), con las siguientes modificaciones:

- Una respuesta/petición no válida no cierra la comunicación.
- El tamaño del *payload* del mensaje JSON-RPC no debe ser mayor de 982 bytes.
- Respuestas mayores de 937 bytes no serán devueltas.
- Los mensajes de notificación también reciben una respuesta.

Las modificaciones citadas anteriormente tienen como objetivo añadir seguridad para evitar posibles desbordamientos de memoria debido al envío de mensajes demasiado largos o un número de peticiones demasiado alto.

Internamente, en el dispositivo, los mensajes recibidos se guardan en una cola y son procesados y respondidos a la mayor brevedad posible en orden de recepción.



Nótese que los mensajes son procesados por orden de llegada.

Adicionalmente algunos mensajes (tal como el cambio de configuración WiFi) pueden producir el reinicio de Nanoenvi IAQ™. Esto puede provocar la pérdida de mensajes recibidos con posterioridad.

Se deben tener en cuenta también los siguientes extremos:

- Si un mensaje es demasiado largo o la cola de mensajes del dispositivo está llena, se responde con un mensaje de error.
- Si se invoca a métodos de lectura de configuración en cortos espacios de tiempo, puede que algunas de las respuestas no sean recibidas (esto podría darse si el buffer de salida del dispositivo se llena).

- Algunos métodos implican el reinicio del dispositivo, por lo que debe tenerse en cuenta que todos los mensajes recibidos con posterioridad serán ignorados hasta que se inicie de nuevo.

## 4.2.2. Métodos públicos expuestos

- **rpc.list**: permite listar los métodos expuestos.
- **config.get.aqi\_params**: permite leer la configuración de parámetros relacionados con los índices de calidad del aire tales como las medidas tenidas en cuenta para el cálculo del IAQI o qué parámetro se muestra en los LEDs de estado (TCI, IAQI o EIAQI).
- **config.set.aqi\_params**: permite cambiar la configuración de parámetros relacionados con los índices de calidad del aire tales como las medidas tenidas en cuenta para el cálculo del IAQI o qué parámetro se muestra en los LEDs de estado (TCI, IAQI o EIAQI).
- **config.get.sensor\_params**: permite leer los parámetros de operación de sensores configurados como puede ser, por ejemplo, la autocalibración del sensor de CO2.
- **config.set.sensor\_params**: permite cambiar los parámetros de operación de sensores configurados como puede ser, por ejemplo, la autocalibración del sensor de CO2.
- **co2.frc**: permite forzar la calibración del sensor de CO2, enviando un valor de CO2 conocido.
- **config.get.measures\_coefcs**: permite leer los factores de corrección aplicados a las medidas.
- **config.set.measures\_coefcs**: permite cambiar los factores de corrección aplicados a las medidas.
- **config.get.mqtt**: permite leer la configuración MQTT y el periodo de envío de datos.
- **config.set.mqtt**: permite cambiar la configuración MQTT y el periodo de envío de datos.
- **config.get.wifi**: devuelve el SSID de la WiFi a la que se conecta el dispositivo.
- **config.set.wifi**: permite cambiar la configuración WiFi del dispositivo.
- **device\_info**: devuelve la dirección MAC y versión de firmware del dispositivo
- **ping**: permite conocer si un dispositivo está en funcionamiento y la conexión con el broker MQTT es correcta.
- **download\_ota**: notifica al dispositivo la presencia de una actualización.
- **reset**: reinicia el dispositivo.
- **factory\_reset**: restablece la configuración del Nanoenvi IAQ™ a los valores de fábrica.

## 4.2.3. Descripción métodos públicos

### 4.2.3.1. rpc.list

#### Descripción:

El método `rpc.list` devuelve un mensaje con los métodos JSON-RPC expuestos por Nanoenvi IAQ™

#### Formato:

```
{
  "id": <id de mensaje>,
  "method": "rpc.list"
```

```
}

```

**Ejemplo:**

```
{
  "id": 3,
  "method": "rpc.list"
}
```

**Respuesta:**

Si el mensaje recibido es correcto y hay espacio en la cola de mensajes del dispositivo, la respuesta tendrá el siguiente formato:

```
{
  "id": <id de mensaje>,
  "result": [<array de métodos expuestos de Nanoenvi IAQ>]
}
```

En caso de mensaje incorrecto o falta de espacio en la cola de mensajes del dispositivo se responderá un mensaje de error (consultar documentación de mensajes de error).

**4.2.3.2. download\_ota****Descripción:**

El método `download_ota` indica a Nanoenvi IAQ™ que debe descargar una actualización OTA del firmware.

Tras la recepción del mensaje se comprueba que binario de actualización es compatible con la CPU del dispositivo. Esto se realiza comprobando el nombre del mismo. Si el nombre del binario indica que no es compatible con la CPU, se descarta la actualización **OTA**.

**Formato:**

```
{
  "id": <id de mensaje>,
  "method": "download_ota",
  "params": {
    "url": "<URL de con archivo binario de firmware>",
    "hash": "<hash MD5 del nuevo binario de firmware>"
  }
}
```

**Ejemplo:**

```
{
  "id": 1,
  "method": "download_ota",
  "params": {
    "url": "http://192.168.137.1:8080/firmware_binary.bin",
    "hash": "e44a5eea993e8a34e938feab4c2f"
  }
}
```

**Respuesta:**

Si el mensaje recibido es correcto y hay espacio en la cola de mensajes del dispositivo, la respuesta tendrá el siguiente formato:

```
{
  "id": <message id>,

```

```
}
  "result": "OTA notified successfully"
}
```

En caso de mensaje incorrecto o falta de espacio en la cola de mensajes del dispositivo se responderá un mensaje de error (consultar documentación de mensajes de error).

En el caso de que el nombre del binario de la actualización indique que este no es compatible con la CPU, el mensaje retornado por el dispositivo será el siguiente:

```
{
  "id":<message id>,
  "error": {
    "code": -32602,
    "message": "download_ota",
    "data":"OTA update suspected to be incompatible with CPU"}}}
```

### 4.2.3.3. ping

#### Descripción:

El método `ping` se envía a Nanoenvi IAQ™ a modo de ping para comprobar que esté conectado y disponible.

#### Formato:

```
{
  "id": <message id>,
  "method": "ping"
}
```

#### Ejemplo:

```
{
  "id": 1,
  "method": "ping"
}
```

#### Respuesta:

Si el mensaje recibido es correcto y hay espacio en la cola de mensajes del dispositivo, la respuesta tendrá el siguiente formato:

```
{
  "id":<message id>,
  "result": "ACK"
}
```

En caso de mensaje incorrecto o falta de espacio en la cola de mensajes del dispositivo se responderá un mensaje de error (consultar documentación de mensajes de error).

### 4.2.3.4. device\_info

#### Descripción:

El método `device_info` se envía a Nanoenvi IAQ™ para solicitar metadatos del dispositivo. En la presente versión se devuelven:

- Dirección MAC de módulo WiFi
- IP del dispositivo

- RSSI de la red WiFi a la que está conectado el dispositivo. El valor de RSSI se lee solamente en el arranque de Nanoenvi IAQ™.



El valor de RSSI es leído en el arranque de Nanoenvi IAQ, por lo que si se quiere conocer el valor real/actual del mismo, se aconseja realizar previamente un reinicio del dispositivo.

- ID de microcontrolador.
- Versión de firmware

#### Formato:

```
{
  "id": <message id>,
  "method": "device_info"
}
```

#### Ejemplo:

```
{
  "id": 1,
  "method": "device_info"
}
```

#### Respuesta:

Si el mensaje recibido es correcto y hay espacio en la cola de mensajes del dispositivo, la respuesta tendrá el siguiente formato:

```
{
  "id":<message id>,
  "result": {
    "mac_adress": "<MAC address>",
    "ip_address": "<firmware version>",
    "rssi":<WiFi signal strength in dBm>,
    "fw_ver": "<firmware version>",
    "cpu": "<microcontroller device signature as string representing device signature in hexadecimal format>"
  }
}
```

En caso de mensaje incorrecto o falta de espacio en la cola de mensajes del dispositivo se responderá un mensaje de error (consultar documentación de mensajes de error).

### 4.2.3.5. config.set.wifi

#### Descripción:

El método `config.set.wifi` permite enviar la configuración de la red WiFi.



Tras el envío de método el dispositivo se reiniciará pudiendo perder los mensajes recibidos con posterioridad.



Tras el reinicio del dispositivo, se conectará a la red WiFi configurada, por lo que si no está disponible o si se configuró un red incorrecta se perderá la comunicación con el mismo.

**Formato:**

```
{
  "id": <message id>,
  "method": "config.set.wifi",
  "params": {
    "ssid": "<SSID of the new WiFi network (string, max 32 characters)>",
    "pass": "<string, WiFi network password (string, max 63 characters)>"
  }
}
```

El mensaje debe incluir todos los campos (SSID y password) para que sea tenido en cuenta.

**Ejemplo:**

```
{
  "id": 1,
  "method": "config.set.wifi",
  "params": {
    "ssid": "SSID_EXAMPLE",
    "pass": "PASSWORD"
  }
}
```

**Respuesta:**

Si el mensaje recibido es correcto y hay espacio en la cola de mensajes del dispositivo, la respuesta tendrá el siguiente formato:

```
{
  "id":<message id>,
  "result": "wifi config updated"
}
```

En caso de mensaje incorrecto o falta de espacio en la cola de mensajes del dispositivo se responderá un mensaje de error (consultar documentación de mensajes de error).

**4.2.3.6. config.get.wifi****Descripción:**

El método `config.get.wifi` permite consultar la configuración WiFi de Nanoenvi IAQ™. Sólo se devuelve el nombre de la red a la que está conectado Nanoenvi IAQ™.

**Formato:**

```
{
  "id": <message id>,
  "method": "config.get.wifi"
}
```

**Ejemplo:**

```
{
  "id": 1,
  "method": "config.get.wifi"
}
```

**Respuesta:**

Si el mensaje recibido es correcto y hay espacio en la cola de mensajes del dispositivo, la respuesta tendrá el siguiente formato:

```
{
  "id": <message id>,
  "result": {
    "ssid": "<SSID of the new WiFi network>"
  }
}
```

En caso de mensaje incorrecto o falta de espacio en la cola de mensajes del dispositivo se responderá un mensaje de error: consultar documentación de mensajes de error.

#### 4.2.3.7. config.set.mqtt

##### Descripción:

El método `config.set.mqtt` permite cambiar la configuración de conexión MQTT del dispositivo de forma remota.



Tras el envío de método el dispositivo se reiniciará pudiendo perder los mensajes recibidos con posterioridad.



Tras el reinicio del dispositivo, se conectará al broker MQTT con los nuevos parámetros configurados, por lo que si no está disponible o si se configuró de forma incorrecta se perderá la comunicación con el dispositivo.

##### Formato:

```
{
  "id": <message id>,
  "method": "config.set.mqtt",
  "params": {
    "host": "<MQTT broker URL (string, 31 characters max)>",
    "port": <port used to connect to MQTT broker (integer, max. 65535)>,
    "ssl": <indicates if connection is secure or not (boolean: true or false)>,
    "comm_period": <communication period in seconds(integer, 4 digits max.)>,
    "user": "<MQTT user (string, max 16 characters)>",
    "pass": "<MQTT password (string, max 16 characters)>",
    "topic": "<MQTT topic where data measurements are received (string, max. 64 characters)>"
  }
}
```

Para ser tenido en cuenta, el mensaje debe contener todos los campos.

##### Ejemplo:

```
{
  "id": 1,
  "method": "config.set.mqtt",
  "params": {
    "host": "mqtt-broker-example-name",
    "port": 1883,
    "ssl": false,
    "comm_period": 1200,
    "user": "my_mqtt_user",
    "pass": "my_mqtt_password",
    "topic": "sensor_data_measures"
  }
}
```

##### Respuesta:



Si el mensaje recibido es correcto y hay espacio en la cola de mensajes del dispositivo, la respuesta tendrá el siguiente formato:

```
{
  "id": <message id>,
  "result": "mqtt config updated"
}
```

En caso de mensaje incorrecto o falta de espacio en la cola de mensajes del dispositivo se responderá un mensaje de error (consultar documentación de mensajes de error).

#### 4.2.3.8. config.get.mqtt

##### Descripción:

El método `config.get.mqtt` permite obtener la configuración MQTT del dispositivo.

##### Formato:

```
{
  "id": <message id>,
  "method": "config.get.mqtt"
}
```

##### Ejemplo:

```
{
  "id": 1,
  "method": "config.get.mqtt"
}
```

##### Respuesta:

Si el mensaje recibido es correcto y hay espacio en la cola de mensajes del dispositivo, la respuesta tendrá el siguiente formato:

```
{
  "id": <message id>,
  "method": "config.set.mqtt",
  "result": {
    "broker": "<MQTT broker URL (string, 31 characters max)>",
    "port": <port used to connect to MQTT broker (integer, 4 digits max.)>,
    "ssl": <indicates if connection is secure or not (boolean: true or false)>,
    "comm_period": <communication period in seconds(integer, 4 digits max.)>,
    "user": "<MQTT user (string, max 16 characters)>",
    "pass": "<MQTT password (string, max 16 characters)>",
    "topic": "<MQTT topic where data measurements are received (string, max. 64 characters)>"
  }
}
```

En caso de mensaje incorrecto o falta de espacio en la cola de mensajes del dispositivo se responderá un mensaje de error (consultar documentación de mensajes de error).

#### 4.2.3.9. config.set.measures\_coefcs

##### Descripción:

El método `config.set.measures_coefcs` permite cambiar de forma remota los valores de ajuste lineal de las medidas de los sensores del dispositivo.

Tras adquirir datos de los sensores, Nanoenvi IAQ™ aplica un ajuste lineal que permite realizar compensaciones sobre el valor medido:

$$\text{medida\_corregida} = a * \text{medida\_sensor} + b \quad (4.1)$$



Nótese que si  $a=1$  y  $b=0$ ,  $\text{medida\_corregida} = \text{medida\_sensor}$  (compensación desactivada).

Si bien se permite aplicar esta compensación para todas las medidas, no tiene sentido hacerlo en los siguientes casos (para los cuales el coeficiente  $a$  deberá ser 1 y el coeficiente  $b$  deberá ser 0):

- CO2: si la calibración automática está activa. En ese caso, los parámetros  $a$  y  $b$  pierden su validez cuando el sensor se autocalibre dando lugar a medidas erróneas.
- VOC: el sensor de VOC implementa una autocalibración, por lo que los parámetros de ajuste lineal pierden su validez cada vez que se realiza una autocalibración.

#### Formato:

```
{
  "id": <message id>,
  "method": "config.set.measures_coefcs",
  "params": {
    "co2": {
      "a": <co2 measurement compensation slope, (float)>,
      "b": <co2 measurement compensation offset, (float)>
    },
    "voc": {
      "a": <voc measurement compensation slope, (float)>,
      "b": <voc measurement compensation offset, (float)>
    },
    "co": {
      "a": <co measurement compensation slope, (float)>,
      "b": <co measurement compensation offset, (float)>
    },
    "pm1": {
      "a": <pm1 measurement compensation slope, (float)>,
      "b": <pm1 measurement compensation offset, (float)>
    },
    "pm2_5": {
      "a": <pm2.5 measurement compensation slope, (float)>,
      "b": <pm2.5 measurement compensation offset, (float)>
    },
    "pm4": {
      "a": <pm4 measurement compensation slope, (float)>,
      "b": <pm4 measurement compensation offset, (float)>
    },
    "pm10": {
      "a": <pm10 measurement compensation slope, (float)>,
      "b": <pm10 measurement compensation offset, (float)>
    },
    "t": {
      "a": <temperature measurement compensation slope, (float)>,
      "b": <temperature measurement compensation offset, (float)>
    },
    "rh": {
      "a": <relative humidity measurement compensation slope, (float)>,
      "b": <relative humidity measurement compensation offset, (float)>
    },
    "noise": {
      "a": <noise level measurement compensation slope, (float)>,
      "b": <noise level measurement compensation offset, (float)>
    }
  }
}
```

```
}
}
}
```

Para que el mensaje sea considerado como correcto, no es necesario que incluya los factores de ajuste de todas las variables medidas. Para cada variable configurada es requerido que se envíen los coeficientes a y b. En caso contrario, el parámetro no será tenido en cuenta.

### Ejemplo:

Ejemplo de configuración de factores de ajuste de todos los parámetros medidos:

```
{
  "id": 3,
  "method": "config.set.measures_coefcs",
  "params": {
    "co2": {
      "a": -111.11111,
      "b": -111.11111
    },
    "voc": {
      "a": -111.11111,
      "b": -111.11111
    },
    "co": {
      "a": -111.11111,
      "b": -111.11111
    },
    "pm1": {
      "a": -111.11111,
      "b": -111.11111
    },
    "pm2_5": {
      "a": -111.11111,
      "b": -111.11111
    },
    "pm4": {
      "a": -111.11111,
      "b": -111.11111
    },
    "p": {
      "a": -111.11111,
      "b": -111.11111
    },
    "pm10": {
      "a": -111.11111,
      "b": -111.11111
    },
    "t": {
      "a": -111.11111,
      "b": -111.11111
    },
    "rh": {
      "a": -111.11111,
      "b": -111.11111
    },
    "noise": {
      "a": -111.11111,
      "b": -111.11111
    }
  }
}
```

Ejemplo de configuración de factores de ajuste de medidas de temperatura:

```
{
```

```

    "id": 3,
    "method": "config.set.measures_coefcs",
    "params": {
      "t": {
        "a": 1,
        "b": -2.0
      }
    }
  }
}

```

**Respuesta:**

Si el mensaje recibido es correcto y hay espacio en la cola de mensajes del dispositivo, la respuesta tendrá el siguiente formato:

```

{
  "id": <message id>,
  "result": "updated : [<name of measurement parameters which factors have been updated>]"
}

```

En caso de mensaje incorrecto o falta de espacio en la cola de mensajes del dispositivo se responderá un mensaje de error (consultar documentación de mensajes de error).

**4.2.3.10. config.get.measures\_coefcs****Descripción:**

El método `config.get.measures_coefcs` permite obtener los parámetros de compensación lineal de las medidas adquiridas por los diferentes sensores.

**Formato:**

```

{
  "id": <message id>,
  "method": "config.get.measures_coefcs"
}

```

**Ejemplo:**

```

{
  "id": 1,
  "method": "config.get.measures_coefcs"
}

```

**Respuesta:**

Si el mensaje recibido es correcto y hay espacio en la cola de mensajes del dispositivo, la respuesta tendrá el siguiente formato:

```

{
  "id": <message id>,
  "method": "config.get.measures_coefcs",
  "result": {
    "co2": {
      "a": <co2 measurement compensation slope, (float)>,
      "b": <co2 measurement compensation offset, (float)>
    },
    "voc": {
      "a": <voc measurement compensation slope, (float)>,
      "b": <voc measurement compensation offset, (float)>
    },
    "co": {

```

```

    "a": <co measurement compensation slope, (float)>,
    "b": <co measurement compensation offset, (float)>
  },
  "pm1": {
    "a": <pm1 measurement compensation slope, (float)>,
    "b": <pm1 measurement compensation offset, (float)>
  },
  "pm2_5": {
    "a": <pm2.5 measurement compensation slope, (float)>,
    "b": <pm2.5 measurement compensation offset, (float)>
  },
  "pm4": {
    "a": <pm4 measurement compensation slope, (float)>,
    "b": <pm4 measurement compensation offset, (float)>
  },
  "pm10": {
    "a": <pm10 measurement compensation slope, (float)>,
    "b": <pm10 measurement compensation offset, (float)>
  },
  "t": {
    "a": <temperature measurement compensation slope, (float)>,
    "b": <temperature measurement compensation offset, (float)>
  },
  "p": {
    "a": <pressure measurement compensation slope, (float)>,
    "b": <pressure measurement compensation offset, (float)>
  },
  "rh": {
    "a": <relative humidity measurement compensation slope, (float)>,
    "b": <relative humidity measurement compensation offset, (float)>
  },
  "noise": {
    "a": <noise level measurement compensation slope, (float)>,
    "b": <noise level measurement compensation offset, (float)>
  }
}

```

En caso de mensaje incorrecto o falta de espacio en la cola de mensajes del dispositivo se responderá un mensaje de error (consultar documentación de mensajes de error).

#### 4.2.3.11. config.set.sensor\_params

##### Descripción:

El método `config.set.sensor_params` permite cambiar de forma remota parámetros de operación de diferentes sensores. En concreto, permite cambiar las opciones de operación citadas a continuación:

- CO2:
  - activar/deactivar la autocalibración.
  - activar/deactivar compensación de medidas en función de la presión atmosférica.
- PM:
  - flag para activar/deactivar medidas de PM acumulativas. Por defecto, las medidas de PM se devuelven de forma acumulativa. Por ejemplo, la concentración de partículas PM10 incluye a todas las partículas que tengan un tamaño inferior a 10 micras, PM4 a todas las partículas que tengan un valor menor de 4 micras, etc.

Si se desactiva esta opción, la medidas de concentraciones de partículas en suspensión se darán por intervalo: Por ejemplo, la concentración de partículas PM10, incluye sólo a las partículas cuyo tamaño está comprendido entre 10 y 4 micras, PM4 incluye sólo a las partículas cuyo tamaño está comprendido entre 4 y 2.5 micras, etc.

**Formato:**

```
{
  "id": <message id>,
  "method": "config.set.sensor_params",
  "params": {
    "co2": {
      "autocalibration_active": <autocalibration active or not (boolean: true or
"false")>,
      "pres_compensation": <pressure compensation active or not (bool: true or false)>
    },
    "pm": {
      "cumulative_reads": <boolean value to indicate whether pm measurements are
cumulative or not (boolean: true or false)>
    }
  }
}
```

Para que el mensaje sea considerado como correcto no es necesario que se incluyan entradas correspondientes a todos los sensores.

**Ejemplo:**

```
{
  "id": 3,
  "method": "config.set.sensor_params",
  "params": {
    "co2": {
      "autocalibration_active": true,
      "pres_compensation": true
    },
    "pm": {
      "cumulative_reads": false
    }
  }
}
```

**Respuesta:**

Si el mensaje recibido es correcto y hay espacio en la cola de mensajes del dispositivo, la respuesta tendrá el siguiente formato:

```
{
  "id":<message id>,
  "result": {
    "updated": [<list of sensors which compensation factors have been updated
successfully>]
  }
}
```

En caso de mensaje incorrecto o falta de espacio en la cola de mensajes del dispositivo se responderá un mensaje de error (consultar documentación de mensajes de error).

**4.2.3.12. config.get.sensor\_params****Descripción:**

El método `config.set.sensor_params` permite obtener de forma remota parámetros de operación de diferentes sensores.

**Formato:**

```
{
  "id": <message id>,
  "method": "config.get.sensor_params"
}
```

**Ejemplo:**

```
{
  "id": 8,
  "method": "config.get.sensor_params"
}
```

**Respuesta:**

Si el mensaje recibido es correcto y hay espacio en la cola de mensajes del dispositivo, la respuesta tendrá el siguiente formato:

```
{
  "id":<message id>,
  "result": {
    "co2": {
      "autocalibration_active": <autocalibration active or not (boolean: true or
false)>,
      "pres_compensation": <pressure compensation active or not (boolean: true or
false)>
    },
    "pm": {
      "cumulative_reads":<boolean value to indicate whether pm measurements are
cumulative or not (boolean: true or false)>
    }
  }
}
```

En caso de mensaje incorrecto o falta de espacio en la cola de mensajes del dispositivo se responderá un mensaje de error (consultar documentación de mensajes de error).

**4.2.3.13. co2.frc****Descripción:**

El método `co2.frc` forzar la calibración del sensor de CO2 enviando un valor de referencia conocido. El valor debe ser expresado en ppm y estar entre 400 y 2000 ppm.

**Formato:**

```
{
  "id": <message id>,
  "method": "co2.frc",
  "params": {
    "c_ref_ppm":<CO2 reference concentration in ppm>
  }
}
```

**Ejemplo:**

```
{
  "id": <message id>,
  "method": "co2.frc",
  "params": {
    "c_ref_ppm": 550
  }
}
```

```
}

```

**Respuesta:**

Si el mensaje recibido es correcto y hay espacio en la cola de mensajes del dispositivo, la respuesta tendrá el siguiente formato:

```
{
  "id": <message id>,
  "result": {
    "updated": [<list of sensors which compensation factors have been updated
successfully>]
  }
}
```

En caso de mensaje incorrecto o falta de espacio en la cola de mensajes del dispositivo se responderá un mensaje de error (consultar documentación de mensajes de error).

**4.2.3.14. config.set.aqi\_params****Descripción:**

El método `config.set.aqi_params` permite configurar los parámetros relacionados con el cálculo del AQI (Air Quality Index):

- Qué parámetro de AQI se muestra en las luces de estado :
  - índice de calidad del aire interior (IAQI)
  - índice de confort térmico (TCI)
  - una combinación de los dos anteriores (EIAQI)
- Qué medidas se utilizan para el cálculo del IAQI. Para el cálculo de la calidad del aire se pueden tener en cuenta las medidas de concentraciones de:
  - CO
  - CO2
  - VOC
  - PM10

De este modo, se puede seleccionar cuáles de estas variables son incluidas en el cálculo del IAQI y cuáles no.

Un ejemplo de aplicación del uso de este mensaje es la configuración de Nanoenvi IAQ™ para que muestre el nivel de CO2 a través de las luces de estado.

**Formato:**

```
{
  "id": <message id>,
  "method": "config.set.aqi_params",
  "params": {
    "aqi_displayed": "<type of AQI index shown by status LEDs: "iaqi", "tci", "eiaqi">",
    "iaqi_used_measures": [<array of names of measurements used to calculate IAQI, values:
"co2", "co", "voc", "pm10">]
  }
}
```



**Ejemplo:**

```
{
  "id": 8,
  "method": "config.set.aqi_params",
  "params": {
    "aqi_displayed": "iaqi",
    "iaqi_used_measures": [
      "co2"
    ]
  }
}
```

En este ejemplo, se configura el dispositivo para que las luces de estado muestren el nivel de CO2 medido: el parámetro `iaqi_used_measures` se configura para que el índice de calidad de aire interior (IAQI) se calcule usando sólo la medida de CO2 y el parámetro `aqi_displayed` se configura como "iaqi" para que en las luces de estado se muestre el índice de calidad de aire interior. De este modo en las luces de estado se muestra el nivel de CO2.

**Respuesta:**

Si el mensaje recibido es correcto y hay espacio en la cola de mensajes del dispositivo, la respuesta tendrá el siguiente formato:

```
{
  "id": <message id>,
  "result": "aqi parameters updated"
}
```

En caso de mensaje incorrecto o falta de espacio en la cola de mensajes del dispositivo se responderá un mensaje de error: consultar documentación de mensajes de error.

**4.2.3.15. config.get.aqi\_params****Descripción:**

El método `config.get.aqi_params` permite leer la configuración de los parámetros relacionados con el cálculo del AQI (Air Quality Index):

- Qué parámetro de AQI se muestra en las luces de estado :
  - índice de calidad del aire interior (IAQI)
  - índice de confort térmico (TCI)
  - la combinación de los dos índices anteriores, dado como resultado el índice de calidad de aire ambiental (EIAQI)
- Qué medidas se utilizan para el cálculo del IAQI. Para el cálculo de la calidad del aire se pueden tener en cuenta las medidas de concentraciones de:
  - CO
  - CO2
  - VOC
  - PM10

**Formato:**

```
{
```

```
"id": <message id>,
"method": "config.get.aqi_params"
}
```

**Ejemplo:**

```
{
  "id": 6,
  "method": "config.get.aqi_params"
}
```

**Respuesta:**

Si el mensaje recibido es correcto y hay espacio en la cola de mensajes del dispositivo, la respuesta tendrá el siguiente formato:

```
{
  "id":<message id>,
  "result": {
    "aqi_displayed": "<type of AQI index shown by status LEDs: 'iaqi','tci','eiaqi'>",
    "iaqi_used_measures": [<array of names of measurements used to calculate IAQI, values:
'co2', 'co', 'voc', 'pm10'>]
  }
}
```

En caso de mensaje incorrecto o falta de espacio en la cola de mensajes del dispositivo se responderá un mensaje de error (consultar documentación de mensajes de error).

**4.2.3.16. reset****Descripción:**

El método `reset` reinicia el dispositivo Nanoenvi IAQ™.



Tras el envío de este método el dispositivo se reiniciará pudiendo perder los mensajes recibidos con posterioridad.

**Formato:**

```
{
  "id": <id de mensaje>,
  "method": "reset"
}
```

**Ejemplo:**

```
{
  "id": 3,
  "method": "reset"
}
```

**Respuesta:**

Si el mensaje recibido es correcto y hay espacio en la cola de mensajes del dispositivo, la respuesta tendrá el siguiente formato:

```
{
```

```

    "id":<id de mensaje>,
    "result": "device will be reset"
  }

```

En caso de mensaje incorrecto o falta de espacio en la cola de mensajes del dispositivo se responderá un mensaje de error (consultar documentación de mensajes de error).

#### 4.2.3.17. factory\_reset

##### Descripción:

El método `factory_reset` actualización la configuración del dispositivo Nanoenvi IAQ™ a los valores de fábrica



Tras el envío de este método el dispositivo se reiniciará pudiendo perder los mensajes recibidos con posterioridad.

El dispositivo pasará a tener la configuración de fábrica. Esto implica que perderá la conexión con el broker MQTT y red WiFi, entrará en modo configuración (necesitará ser configurado desde la página Web de configuración desde un punto cercano al dispositivo) y perderá los parámetros de calibración y de uso de sensores.

##### Formato:

```

{
  "id": <id de mensaje>,
  "method": "factory_reset"
}

```

##### Ejemplo:

```

{
  "id": 3,
  "method": "factory_reset"
}

```

##### Respuesta:

Si el mensaje recibido es correcto y hay espacio en la cola de mensajes del dispositivo, la respuesta tendrá el siguiente formato:

```

{
  "id":<id de mensaje>,
  "result": "device will be reset to factory settings"
}

```

En caso de mensaje incorrecto o falta de espacio en la cola de mensajes del dispositivo se responderá un mensaje de error (consultar documentación de mensajes de error).

# Nanoenvi

[www.enviraiot.es](http://www.enviraiot.es)