

Calidad de Aire

Introducción

Nuestro objetivo es mejorar la calidad de CO₂ en el aire en espacios públicos como hospitales, universidades, etc. Ya que mejorar la calidad de este ayudará a una mejora de salud y bienestar en las personas. Esto también repercute en el rendimiento de los trabajadores. Por ello hemos desarrollado un sistema capaz de controlar la calidad del aire y evitar exponernos a ambientes perjudiciales.

Al principio nuestra idea era un poco más grande de lo que finalmente desarrollamos en este proyecto.

Teníamos pensado que en vez de un dispositivo fueran varios comunicados entre sí para poder controlar no solo la calidad de aire en una sala sino de todo un espacio público como puede ser una universidad para que, si estás en una sala con exceso de CO₂ en el aire, localice cuál es la sala más próxima a ti con un nivel de CO₂ saludable.

Finalmente, por razones de tiempo no hemos podido desarrollar esta idea a tan gran escala, por lo que lo hemos desarrollado para hacerlo en una sola sala.

Es muy fácil construir un sistema simplificado para mostrar cuando los niveles de CO₂ están dentro de unos umbrales, para así poder monitorizar los espacios.

Funcionalidades

- **Sensor MQ-135:** este sensor permite obtener la lectura tanto de un valor analógico como de un valor digital cuando supera un umbral regulado a través de un potenciómetro ubicado en el sensor. Este

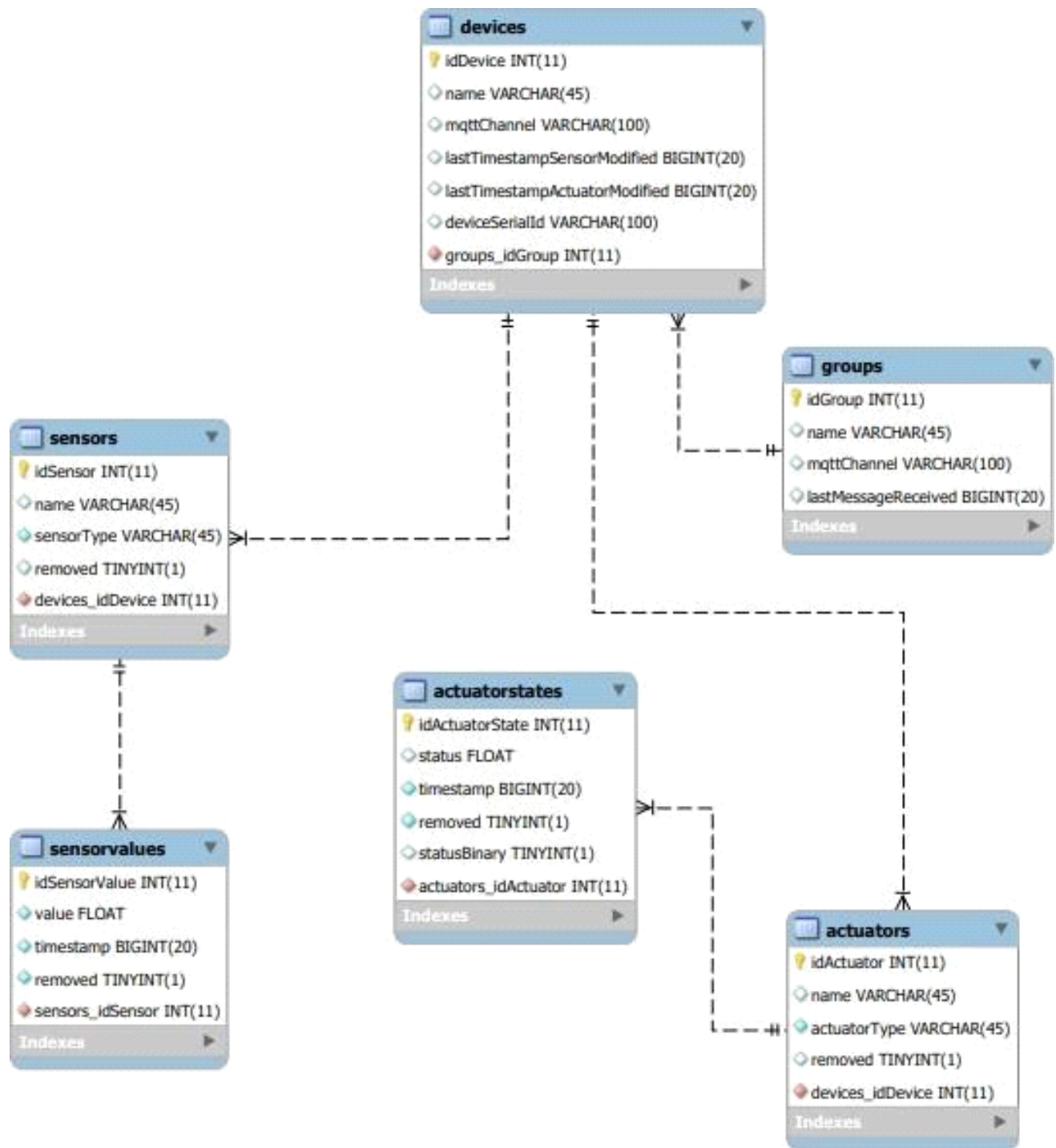
sensor permite obtener valores en todo momento de la calidad del aire. Cabe destacar que estos sensores son diseñados para detectar la presencia de distintos componentes químicos en el aire, en este caso CO₂.

- **Relé de monitorización:** este relé monitoriza la calidad del aire según una regla de negocio que indica que si se supera cierto umbral, se activa el relé, para así poder controlar cómo de perjudicial es la calidad del aire tomada por el sensor.

Base de datos.

La base de datos diseñada para nuestro proyecto no es compleja, almacenaremos los valores que se emiten por el sensor en cada momento, también la información relativa a los actuadores, dispositivos, etc...

Para ello se han implementado diferentes tablas para la información del sensor o actuador y para los valores que se recogen, se puede observar según la siguiente imagen:



Para todas las tablas tenemos los valores necesarios para obtener una base de datos no muy compleja.

Api Rest.

En el diseño de las API Rest principalmente se han creado funciones que nos conecten con la base de datos, está formada por lo siguiente:

- **Métodos GET:**

Hemos creado un get para cada tabla. Para los valores de sensores como los valores de actuadores hemos añadido la opción de filtrar mediante 'deviceId' (id del dispositivo al que nos corresponde).

También hemos añadido distintas funcionalidades para filtrar por distintas variables y que se muestren los datos que queremos ver..

- **Métodos POST:**

Para insertar nuevos elementos en la base de datos hemos usado los métodos POST. Uno para los sensores, con los valores que se recogen en el sensor específico y otro para los actuadores.

Para estas peticiones hemos usado URLs del siguiente modo:

La ruta principal de la petición es la siguiente: "/api/devices/id".

Dependiendo de que objeto es el que queremos insertar añadiremos nuevos parametros a la ruta de esta manera

- Para obtener los sensores o actuadores de un dispositivo específico, la nomenclatura es: "/api/devices/id/sensors" o "/api/devices/id/actuators".
- Si queremos obtener un sensor específico o un relé específico, usamos la siguiente nomenclatura: "/api/devices/id/sensors/(sensor)" o "/api/devices/id/actuators/(actuador)".

Hay que tener cuidado a la hora de realizar un método POST ya que debemos introducir todos los atributos, si no es así la petición fallará.

MQTT.

MQTT tiene una función importante en nuestro proyecto ya que permite que actúen los actuadores. En este proyecto usaremos un solo canal, el referente a nuestro dispositivo

En la estructura del servidor MQTT de nuestro proyecto, se usa una clase con métodos para inicializar el servidor, y para la conexión, desconexión, suscripción y para darnos de baja. Tenemos que especificar los canales que estamos usando, en nuestro caso, sólo el del dispositivo.

La estructura del servidor MQTT es genérica para todos los proyectos, es decir, el código es prácticamente el mismo. Se usan una clase con métodos para inicializar el servidor, y para indicar la conexión, desconexión, suscripción y para darnos de baja. El único punto importante son los canales, donde debemos de especificar el canal que estamos usando, en nuestro caso el referente al dispositivo.

ESP32.

El código para el ESP32 sigue una estructura en la que solo tenemos que adaptarlo a nuestro proyecto:

- **Setup:** Inicializamos la conexión Wifi e inicializamos el cliente MQTT, mostramos por pantalla la IP obtenida, configuramos los modos de pin para los actuadores y los sensores e iniciamos y obtenemos el tiempo actual.
- **Loop:** Si no estamos conectados al servidor MQTT nos volvemos a conectar, mediante la función loop() vamos comprobando si existen nuevos datos para el sensor. Añadimos un contador para que cada segundo se vaya haciendo las distintas operaciones, vamos obteniendo los valores, los subimos a la base de datos y los mostramos por pantalla.

- **Funciones MQTT:** La función 'InitMqtt', inicia la comunicación MQTTe inicia y establece el servidor y el callback al recibir un mensaje.
También tenemos la función 'ConnectMQtt' que conecta o reconecta el MQTT y, si consigue conectar se suscribe al topic y publica un mensaje, si no, espera 5 segundos. Gracias a la función 'HandleMqtt' podemos gestionar la comunicación MQTT, comprobamos si el cliente está conectado, si no lo está, intenta reconectar y, si lo está, llama al MQTT loop.

Por último, tenemos la función 'OnMqttReceived', que es un callback a ejecutar cuando se recibe un mensaje, verificamos si el contenido es igual a 1, para activar el relé, si no lo es, lo apaga.

- **Funciones Actuadores:** Hemos implementado las funciones de serializar y deserializar los actuadores para poder enviar nuevos a la base de datos, o modificar alguno añadido.
- **Funciones Sensores:** De igual forma que los actuadores, tenemos las funciones de serializar y deserializar los sensores, para poder enviar o modificar datos de la base de datos.