



# IES Gran Capitán

## Departamento de Informática

Ciclo Formativo de Grado Superior  
de Administración de Sistemas  
Informáticos

Módulo de Proyecto Integrado

Emilian Scurtu – 2ºASIR

Proyecto: Sistema IoT Cloud

Curso 2020/2021

## Contenido

1. Introducción .....	2
2. Objetivos y requisitos del proyecto.....	3
3. Estudio previo.....	5
3.1 Estado actual .....	5
3.2 Estudio de soluciones existentes .....	6
4. Plan de trabajo .....	7
5. Diseño.....	8
5.2 Diseño detallado .....	10
6. Implantación.....	12
7. Recursos .....	12
7.1 Herramientas hardware .....	12
7.2 Herramientas software .....	12
7.3 Personal.....	15
7.4 Presupuesto .....	15
8. Conclusiones.....	16
8.1 Grado de consecución de objetivos .....	16
8.2 Problemas encontrados .....	17
8.3 Futuras mejoras.....	17
9. Referencias/bibliografía .....	19
10. Anexos .....	20

## 1. Introducción

Hablar, toser, reír o gritar son actividades cotidianas que hasta hace unos meses no suponían ningún peligro. Ahora en espacios cerrados son bombas de relojería. Y es que ventilar es tu aliado para frenar los contagios de la COVID-19. Abrir las ventanas cinco veces en un aula de 100 metros cuadrados, con 25 personas, permite renovar 14 litros de aire por persona y segundo. Reemplaza el aire contaminado por aire limpio del exterior y, además, se eliminan las partículas en suspensión, que pueden contener el virus.

Eso sí, en ningún caso esta renovación de aire sustituye al mantenimiento de la distancia, el uso de mascarilla y el lavado de manos. Son herramientas complementarias para luchar contra el coronavirus. Juntas forman un escudo perfecto. Ahora bien, ¿cómo es posible comprobar si la ventilación es la adecuada? Los expertos tienen claro que el riesgo cero no existe, si bien es posible trabajar para reducir el número de infectados.

Con este objetivo hemos desarrollado un sistema que mide el dióxido de carbono, la temperatura y la humedad.

¿Cómo funciona? Hemos creado un sistema que recibe datos medioambientales emitidos por unos sensores que se sitúan tanto en el interior de las clases. En ellos encuentra la información sobre el estado actual de la ventilación.

«Esto es muy relevante, ya que los sensores han de ubicarse en los lugares más desfavorables de ventilación y los que tienen pantalla no sirven, puesto que hay que acercarse a ellos para leer los valores».

Por ello nuestro sistema es seguro ya que emite alertas a una sola persona puede controlar todas las ventilaciones de todas las aulas.

De esta forma creamos un entorno más seguro y todo el personal que compartimos espacios (profesores, alumnos, personal de otro tipo) tenemos la seguridad de que la calidad del aire está controlada. «Los encargados pueden ventilar lo justo, sin gasto energético innecesario», y comentar que la red de datos que se genera se puede usar para transmitir a su vez otro tipo de mediciones de forma gratuita.

Las causas de una deficiente ventilación interior multiplica los contagios, según los epidemiólogos especialistas en esta enfermedad COVID 19. Con independencia de la pandemia, agrega que se relaciona con problemas de salud, dolores de cabeza, somnolencia, falta de concentración... En este punto, advierte de que en Europa los problemas de salud por calidad del aire han causado la muerte prematura de muchas personas.

La prueba de concepto en la que se basa este proyecto, es en redes abiertas, pero encriptados de baja energía para sensores y medidores de dióxido de carbono, humedad y temperatura gobernados por inteligencia artificial.

Un aliado que, a su parecer, ayuda a conocer el estado sin estar en el lugar, lo que posibilita a los padres conocer cómo están las aulas donde están estudiando sus hijos en tiempo real.

Esta información ayuda a tomar decisiones sobre la necesidad de ventilación o mejorar los sistemas. «Una única persona y mediante el sistema de alertas puede gestionar todos los puntos y proteger a los alumnos, profesores y demás personal.

Tenemos claro que en estos tiempos es muy importante ventilar lo necesario para protegerse del coronavirus, eso sí, manteniendo a raya el bolsillo.

De ahí, considera esencial la información que proporciona un ahorro económico en calefacción y aire acondicionado muy importante, que evita a su vez un aumento de emisiones de efecto invernadero.

## 2. Objetivos y requisitos del proyecto

Como ya he explicado en el índice, el principal problema que hemos tenido en cuenta para hacer este proyecto es la situación de cuarentena y de todo lo que nos ha traído la crisis del Covid-19, para eso hemos desarrollado una PRUEBA DE CONCEPTO consistente en un sistema medidor medioambiental de CO2 y de temperatura, mediante un conjunto de sensores y una placa de IoT que está programada mediante Arduino.

Como todo esta situación de pandemia es variable, y actualmente la situación ha mejorado considerablemente, lo que no quiere decir que se haya controlado totalmente, hemos propuesto un sistema de medición de la calidad del aire, que se puede adaptar a cualquier tipo de espacio/s, con el único requisito de tener una wifi, para poder receptionar y enviar datos en tiempo real.

Esta prueba de concepto la hemos implementando usando los servicios de AWS Educate, lo que nos ha permitido trabajar con una máquina virtual con las mismas características que la máquina física, en caso de que se llegue a producción, y que será la que tenga los servicios de Mosquitto, Telegraf, Influxdb y Grafana, instalados. Tecnologías muy ágiles y de poco peso.

El despliegue de operación que se intenta llevar a cabo es que los sensores estén enviando datos a un BROKER y mediante unas tecnologías de software libre ya indicadas anteriormente, esos datos se recopilen y se monitoricen en base a unas métricas, y mostrar los resultados mediante una herramienta de monitorización. Esto permitiría que si se supera unos valores límites, decidir si ventilar o no ventilar las aulas, se decide en función de unas alertas que se van a programar para, enviar un mensaje al equipo educativo o responsable COVID, para que así decidan si ventilar o no el aula.

Niveles de CO2 ppm	400-600	600-800	800-1000	1000-
Ventilación	EXCELENTE	MUY BUENA	ACEPTABLE	MALA VENTILACIÓN (*)

Actualmente los valores permitidos de CO2 en un espacio cerrado, son los indicados en la tabla anterior.

Este sería el gráfico por el cual se decide si se ventila o no, dependiendo de los valores que recolecta el sensor se enviará la alerta y se ventilara el aula.

Esos son los principales objetivos que busca este proyecto, dar una solución casera y eficaz al problema de la ventilación en las aulas, y así evitar comprar componentes que valen como máximo 60€.

Con la ventaja de que hemos configurado un sistema medioambiental que puede ser ampliable, añadiéndole más sensores y programable, situado dentro del IoT, que ya es una realidad y no una visión de futuro.

La fama de esta tecnología radica principalmente en todas las aplicaciones y posibilidades que nos proporciona tanto para mejorar tanto la vida cotidiana de las personas como los entornos empresariales, dónde ya se está implantando desde hace algún tiempo.

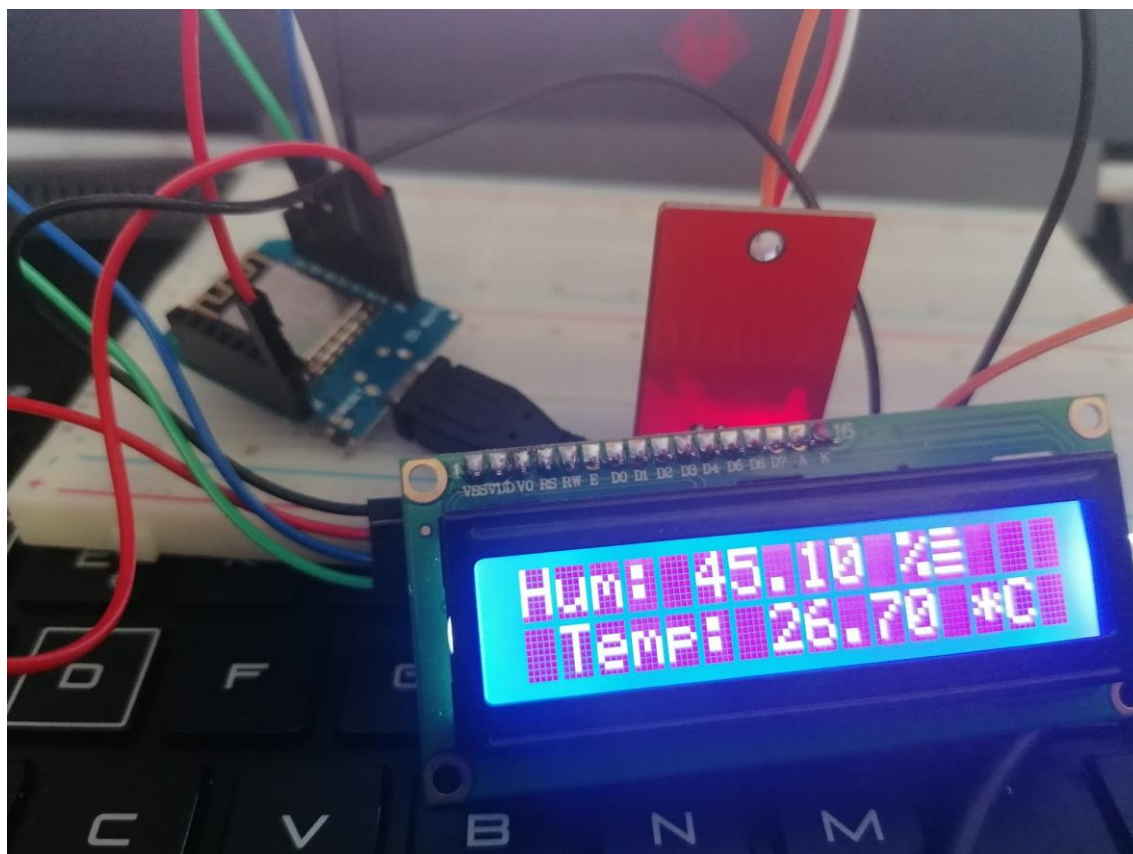
### 3. Estudio previo

#### 3.1 Estado actual

Actualmente en nuestro Instituto no hay casi nada implementado en relación a este sistema de medición. Solamente hemos visto por algunos espacios del Instituto algunos medidores de temperatura y CO2, que parece ser que no son muy fiables. Desde la aparición de la Pandemia, algunas empresas han visto una oportunidad de negocio en la fabricación de este tipo de medidores y ha surgido un mercado de productos relacionados con la medición de CO2, donde podemos encontrar medidores fiables y medidores falsos.

Nuestra propuesta ha sido fabricar nosotros mismos los medidores de CO2 (Dióxido de Carbono), partiendo de una prueba de concepto, para que posteriormente y si se le ve utilidad, se implante en nuestro Instituto, todo esto en función de cómo vaya evolucionando la crisis del coronavirus. Esta prueba de concepto lo que hace es mezclar tanto parte de informática como parte de electrónica, en cuanto a la parte de informática es la instalación y configuración de 4 programas, todo la descripción está en la guía técnica que la puedes consultar aquí.

Y por la parte de la electrónica lo que hemos hecho ha sido mediante una placa que se consigue conectar a la wifi, un sensor de temperatura, uno de dióxido de carbono y una pantalla LCD para poder visualizar los valores que recogen los sensores.



### 3.2 Estudio de soluciones existentes



Este es una de las soluciones existentes que existe en el mercado de hecho el departamento de informática lo compró para utilizarlo, pero no va bien.

En cuanto a los precios que suelen rondar estos aparatos es desde los 55€ hasta los 100€, pero como ya he mencionado anteriormente los aparatos no dan los valores adecuados o dan valores surrealistas.



Tackly Medidor de co2 ambiente mini- detector co2 con medidor calidad de aire interior - higrometro digital medidor de humedad y...

★★★★☆ ~ 17

69,99€

Ahorra 5% con un cupón

✓prime Entrega GRATIS viernes, 18 de junio



Eliviv Detector de Calidad del Aire, Medidor de CO2 Formaldehído TVOC HCHO Portátil con 3,5" Pantalla con Sensor de Temperatura y Humedad...

★★★★☆ ~ 6

58,98€

✓prime GRATIS Envío 1 día  
Recíbelo mañana, 15 de junio



Medidor CO2 Portátil, 400 a 5000 ppm Mini Detector CO2 Medidor de CO2, Probador de Humedad Relativa de Temperatura de Alta Precisión co...

★★★★☆ ~ 6

55,00€

Entrega GRATIS para miembros de Prime

Estos serían los tres productos más vendidos de Amazon, otro factor por el que no son buenos serían las alertas, en el sistema que yo he diseñado se pueden configurar límites para poder enviar alertas y así poder ventilar el aula.

#### 4. Plan de trabajo

El plan de trabajo que he seguido ha sido el siguiente, he empezado la semana final de marzo y lo he terminado en la semana del 14 de junio.

En cuanto al plan de trabajo, lo he dividido en 5 partes, la parte de la información que es la que he pasado la mayoría de las veces haciendo, ya sea solucionando errores o buscando información acerca de los componentes y del software que he utilizado.

Después viene la parte de la configuración electrónica, ya que ha sido de la que más he tardado, casi 6 semanas ya que yo nunca he trabajado con componentes electrónicos y me he tenido que poner a aprender para así poder llevarlo a cabo. En cuanto a la configuración informática, ha sido un poco más corta pero también he tenido algún que otro error con el software que he utilizado. Y por último y no menos importante la documentación del proyecto, para redactarla he tardado alrededor de 2 semanas, con documentación me refiero tanto a la guía técnica, como a este propio documento y la presentación.





## Plan de trabajo Marzo/Junio

Este es el plan de trabajo para la prueba de conceptos

	Semana 1/2	Semana 3/4	Semana 5/6	Semana 7/8	Semana 9/10	Semana 11/12
Información						
Configuración electrónica						
Configuración informática						
Pruebas						
Documentación						

## 5. Diseño

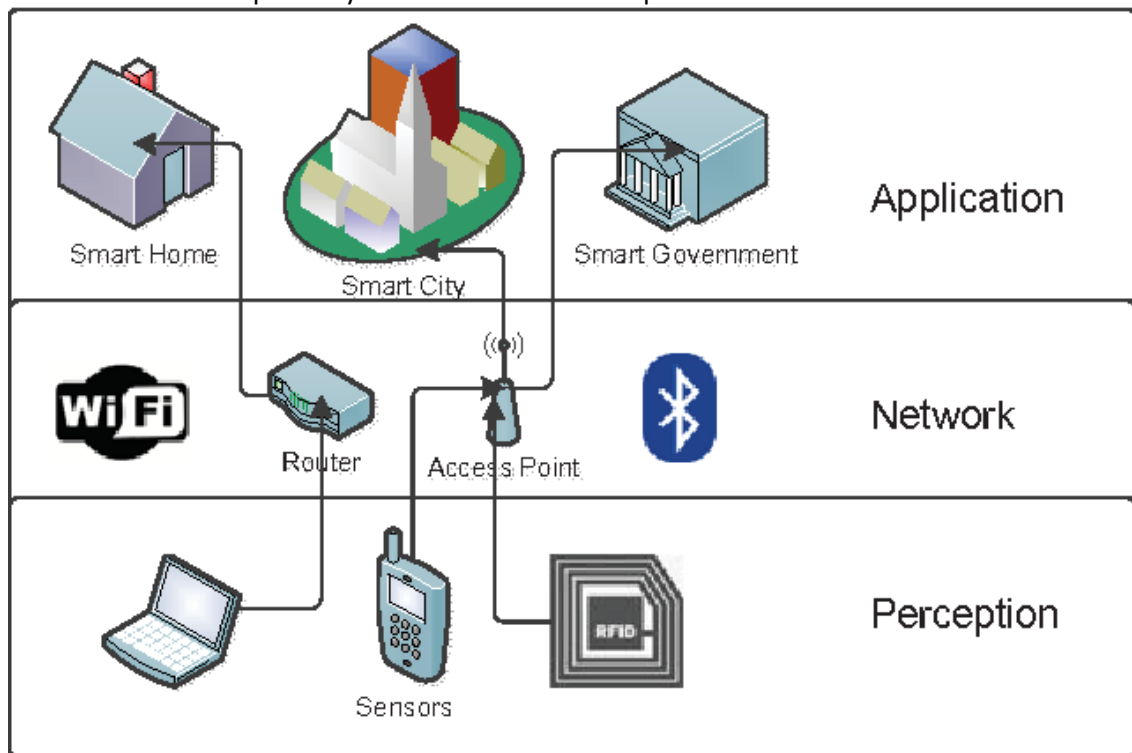
### 5.1 Diseño general

El mercado IoT ofrece cada vez más opciones de plataformas, soluciones y aplicaciones en la nube. Algunas son más o menos cerradas y otras permiten la construcción de la plataforma IoT desde la base.

Para las aplicaciones de IoT basadas en la captación de datos y en el análisis y visualización de estos es muy habitual el uso de la arquitectura de tres niveles (en la foto de abajo se puede apreciar). En esta arquitectura se captura el flujo de información desde los dispositivos los cuales lo transfieren a diversos servicios ubicados en la nube. Las tres capas que la componen son las siguientes:

- **Capa de dispositivos:** es la capa inferior de la arquitectura. Hay varios tipos de dispositivos, pero para que se consideren dispositivos IoT deben tener algún tipo de comunicación, directa o indirecta, que lo enlaza con Internet.
- **Capa de comunicación:** esta capa soporta la conectividad de los dispositivos. Hay múltiples protocolos para la comunicación entre los dispositivos y la nube entre los que destacan HTTP, MQTT y CoAP.

- **Capa de agregación:** es una capa importante de la arquitectura ya que hace de bróker de comunicaciones, ofreciendo soporte para un servidor MQTT para hablar con los dispositivos, enrutamiento de comunicaciones a dispositivos específicos y la habilidad de hacer un puente y transformar diferentes protocolos.



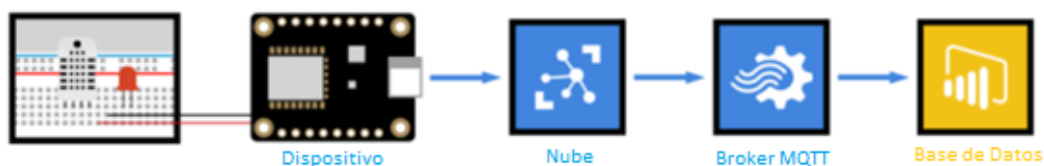
Además de estas tres capas básicas, una arquitectura de IoT más completa también incluiría capas que ayudan a la gestión de identidades o la seguridad de los datos.

La integración de un sistema IoT a cualquier proceso productivo requiere tres fases: conexión, análisis y visualización y automatización.

**Fase 1: Conexión.** Los dispositivos conectados a una red con sensores inteligentes empiezan a enviar información sobre sí mismos y su entorno a un centro de comunicaciones en la nube. Conectar cosas (darles sentido y abrirles una conexión a Internet para que puedan enviar sus datos) representa el comienzo de la evolución del IoT.

**Fase 2: Análisis y Visualización.** A medida que los datos se acumulan hay que empezar a ejecutar análisis inteligentes de las pilas de datos y visualizar los resultados en los dashboards.

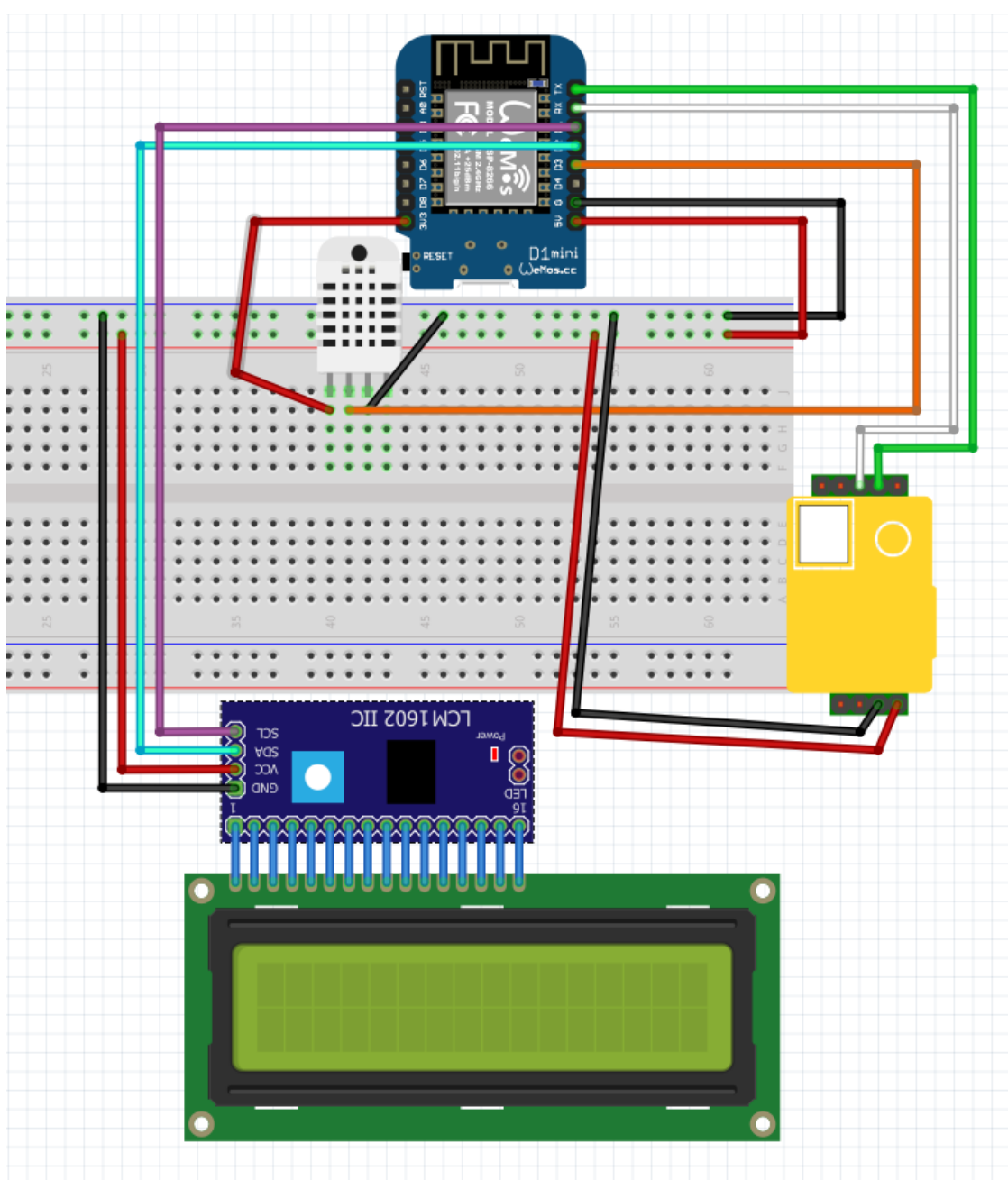
**Fase 3: Automatización.** Con una visión clara de todos los datos que se recopilan ya se puede pensar en hacer automatizaciones para aprovechar todo el potencial del IoT. Entre las más usadas se encuentran la configuración de alarmas, tareas y análisis.



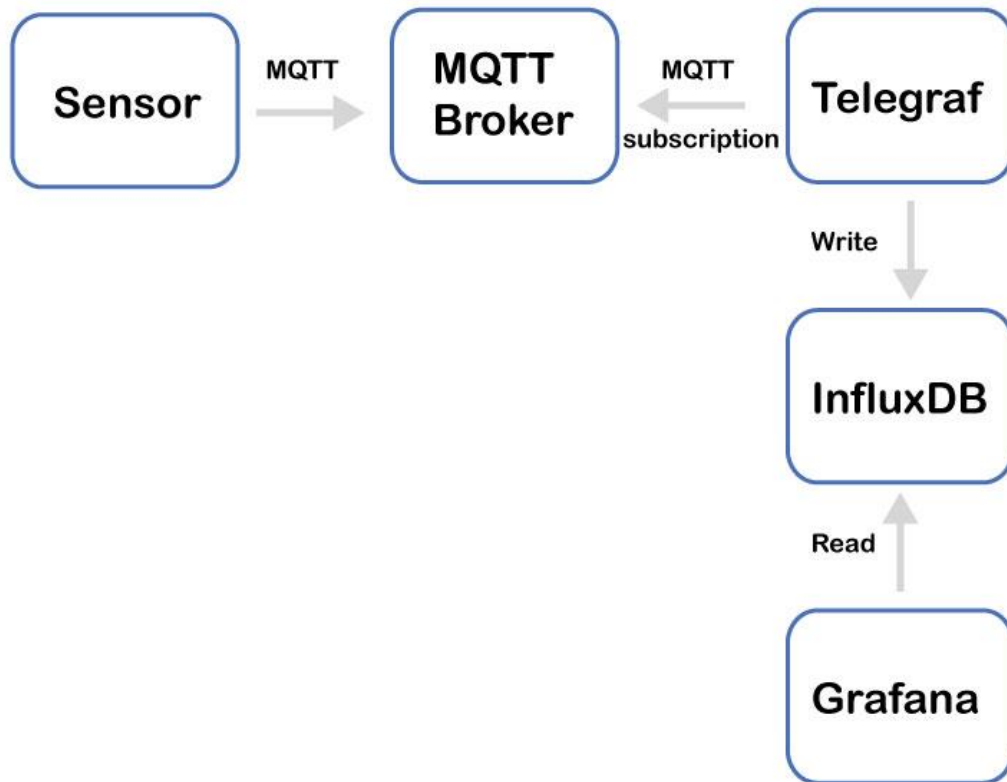
## 5.2 Diseño detallado

La propuesta que yo ofrezco a este problema del COVID-19, es hacer un medidor casero de CO<sub>2</sub>, en cuanto al diseño es muy sencillo, lo único que necesitamos es un servidor donde se alojan los programas que instalaremos, y materiales electrónicos, es decir una placa Wemos D1 mini, un sensor de temperatura/humedad, el DHT22 y un sensor de CO<sub>2</sub>, que sería el MHZ-19B. Esa sería la base de nuestro proyecto, aparte para la visualización de todos los datos necesitaremos una pantalla LCD, está la tendremos para así poder ver todos los datos sin necesidad de consultarlos en el servidor.

Otros materiales que necesitaremos sería una protoboard, unos jumpers, estos son unos cables que sirven para hacer conexiones en electrónica, aparte de eso necesitaremos un soldador para poder soldar todos los componentes si es necesario. A continuación adjunto foto del esquema.



En cuanto a la parte software lo que necesitaremos será instalar una serie de programas, Mosquitto, Telegraf, InfluxDB y por último Grafana, el esquema que utilizaremos es el siguiente.



Este es el esquema que utilizaremos, una vez cogemos los datos, los enviaremos mediante Mosquitto a un topic de Mosquitto, habrá un suscriptor que será la place Wemos D1 Mini y el propio Mosquitto que recibirá los datos, enviaremos los mensajes y una vez enviados estos mensajes.

El otro programa que actuará será Telegraf, este lo que hará es recoger los mensajes que el propio Mosquitto ha recogido, una vez que los coge los publicará en la base de datos InfluxDb, esta es una base de datos temporal que cada x tiempo lo que hará es borrar todos los mensajes que habrá en la base de datos y así mantendrá los más nuevos.

Y por último llega el turno de Grafana que lo que hará es recoger los datos de InfluxDB y los mostrará mediante gráficas para que así se puedan ver y comprender mejor.

Este es el diseño que he propuesto.

## 6. Implantación

Toda la implementación se llevará a cabo en una máquina de AWS, ya que es una prueba de concepto y para probarlo lo vamos a probar allí y ver si funciona, una vez funcionando se podrá poner en producción en los aulas. Toda la implementación, tanto de la máquina de AWS, como la instalación de los demás programas para que se pueda poner en marcha están en la [guía técnica](#).

## 7. Recursos

### 7.1 Herramientas hardware

En cuanto a la parte hardware lo que necesitaremos sería lo siguiente;

- Servidor: El servidor lo necesitaremos para así poder alojar todos los datos que recojan los sensores y además de alojar los datos nos sirve para instalar el software.
- Wemos D1 Mini: La placa que necesitamos para el proyecto es la Wemos D1 mini pro de 16 mb de memoria con el chip ESP8266, esta placa es una de las mejores que hay por ese rango de precios, entre 4 y 5 euros. Además con esta placa han trabajado unos compañeros míos realizando un proyecto y ellos también se decantaron por ella. Hemos elegido esta placa del fabricante Wemos frente a otros fabricantes como sería Olimex u Adafruit, en el caso de Olimex no posee ningún puerto USB y en cuanto a Adafruit sería una placa con más características de las que necesitamos.

En cuanto a las ventajas que tiene es que posee un puerto USB mediante el cual nos conectamos a ella para introducirle las librerías y demás información, tiene un botón de reset y la más importante de todas tiene 16 mb de memoria.

- Dht22: El DHT22 es un sensor que se utiliza para hacer mediciones de temperatura y de humedad, este sensor es del fabricante Adafruit. La ventaja más clara que he tenido en cuenta para decantarme por este y no por otro sensor es que este es digital, a diferencia de otros sensores como el LM35, este utiliza un pin digital mediante el cual se envía la información, esto hace que haya menos ruido, es decir que la información que recibamos sea más acorde con la realidad.
- Mhz-19B: El sensor de vamos a utilizar es el MH-Z19B, este sensor es de los mejores que hay calidad precio, hay uno que es un poco más caro el Senseair s8, este es un poco mejor que este sensor, pero por falta de tiempo a la hora de pedir los componentes, nos hemos decantando por el MH-Z19B que también es uno de los mejores sensores que hay actualmente en el mercado.
- Pantalla LCD LiquidCrystal I2C: El módulo adaptador LCD a I2C que usaremos está basado en el controlador I2C PCF8574, el cual es un expansor de entradas y salidas digitales controlado por I2C. Por el diseño del PCB este módulo se usa especialmente para controlar un LCD Alfanumérico.

Mediante este módulo solo tenemos que utilizar dos pines de nuestra wemos d1 mini, aparte de la tierra y el que le da la corriente.

### 7.2 Herramientas software

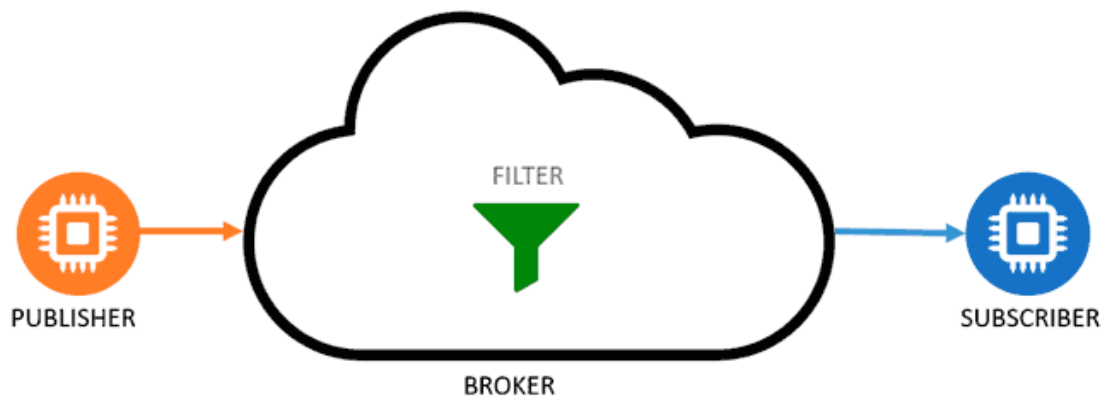
En cuanto a las herramientas software lo que utilizare son cuatro programas que los voy a comentar a continuación;

- Mosquitto: MQTT (Message Queue Telemetry Transport) es un protocolo de comunicación diseñado específicamente para la comunicación entre máquinas (M2M), por lo que posee características pensadas exclusivamente para este tipo de aplicaciones. Su modo de funcionamiento es a través de publicación/suscripción a un tópico, que se publica o lee. Sus mensajes son cortos, por lo que las transmisiones de datos también lo son y esto impacta en el consumo de energía de los nodos, algo crucial en dispositivos alimentados a baterías.

El protocolo MQTT consta de dos tipos de entidades en la red: un bróker de mensajería y numerosos clientes. El bróker es un servidor que recibe todos los mensajes de los clientes y los redirige a los clientes de destino relevantes. Por otra parte, un cliente es cualquier dispositivo que pueda interactuar con el bróker y enviar y recibir mensajes. Por tanto, un cliente puede ser desde un sensor de IoT en campo hasta una aplicación en un centro de datos que procesa datos de IoT.

Para que un cliente reciba un mensaje de un cliente tienen que pasar los siguientes acontecimientos:

1. El cliente se conecta al bróker. Puede suscribirse a cualquier "topic" de mensajería del bróker. Esta conexión puede ser una conexión TCP/IP simple o una conexión TLS cifrada para mensajes sensibles.
2. Otro cliente publica los mensajes en un "topic", enviando el mensaje y el "topic" al bróker.
3. Después, el bróker remite el mensaje a todos los clientes que se suscriben a este "topic".



MQTT dispone de un mecanismo de calidad del servicio o QoS, entendido como la forma de gestionar la robustez del envío de mensajes al cliente ante fallos (por ejemplo, de conectividad). MQTT tiene tres niveles QoS posibles:

- **QoS 0 unacknowledged (at most one):** El mensaje se envía una única vez. En caso de fallo el suscriptor puede que no reciba algún mensaje.
- **QoS 1 acknowledged (at least one):** El mensaje se envía hasta que se garantiza la entrega. En caso de fallo, el suscriptor puede recibir algún mensaje duplicado.
- **QoS 2 assured (exactly one):** Se garantiza que cada mensaje se entrega al suscriptor, y únicamente una vez.

Usar un nivel u otro depende de las características y necesidades de fiabilidad del sistema. Lógicamente, un nivel de QoS superior requiere un mayor intercambio de mensajes de verificación con el cliente y, por tanto, mayor carga al sistema y consumo energético. En este proyecto, se utilizará el protocolo QoS 0 ya que se envían mensajes cada poco tiempo y perder alguno no influye mucho.

El protocolo MQTT dispone de distintas medidas de seguridad que podemos adoptar para proteger las comunicaciones. En muchos casos, la autenticación consiste en una contraseña y usuario que son enviados como texto plano, aunque también es posible configurar el bróker para aceptar conexiones anónimas. Para verificar la correcta configuración del bróker se pueden usar los comandos de publicación y suscripción. Por ejemplo, para verificar un el topic "test" con usuario y contraseña se podría usar:

```
mosquitto_sub -h <brokerhost> -t "test" -u "<username>" -P "<password>"
mosquitto_pub -h <brokerhost> -t "test" -m "Hello World" -u "<username>" -P "<password>"
```

- **Telegraf:** Telegraf es un agente de servidor impulsado por complementos para recopilar y enviar métricas y eventos desde bases de datos, sistemas y sensores de IoT. Podemos encontrar 4 tipos de complementos distintos:
  - Complementos de entrada: recopilan métricas del sistema, servicios o API de terceros.
  - Complementos del procesador: transforman, decoran y / o filtran métricas.
  - Complementos de agregadores: crean métricas agregadas (p. Ej., Promedio, mínimo, máximo, cuantiles, etc.).
  - Complementos de salida: escriben métricas en varios destinos.

Dado que este servicio no requiere de acceso al exterior, el contenedor de Telegraf está enlazado directamente con el contenedor de InfluxDB, es decir, los datos que le transmite Telegraf a InfluxDB no llegan al exterior, sino que pasan a través del contenedor Docker0. En este proyecto, Telegraf se encargará de recopilar los datos que llegan al bróker de MQTT y guardar los datos en la base de datos. Para llegar a hacer esto, dispone de un archivo de configuración donde se configuran tanto las salidas como las entradas.

- **InfluxDB:** Telegraf es un agente de servidor impulsado por complementos para recopilar y enviar métricas y eventos desde bases de datos, sistemas y sensores de IoT. Podemos encontrar 4 tipos de complementos distintos:
  - Complementos de entrada: recopilan métricas del sistema, servicios o API de terceros.
  - Complementos del procesador: transforman, decoran y / o filtran métricas.
  - Complementos de agregadores: crean métricas agregadas (p. Ej., Promedio, mínimo, máximo, cuantiles, etc.).
  - Complementos de salida: escriben métricas en varios destinos.

Dado que este servicio no requiere de acceso al exterior, el contenedor de Telegraf está enlazado directamente con el contenedor de InfluxDB, es decir, los datos que le transmite Telegraf a InfluxDB no llegan al exterior, sino que pasan a través del contenedor Docker0. En este proyecto, Telegraf se encargará de recopilar los datos que llegan al bróker de MQTT y guardar los datos en la base de datos. Para llegar a hacer esto, dispone de un archivo de configuración donde se configuran tanto las salidas como las entradas.

- Grafana: Grafana es una herramienta de software libre hecha para visualizar datos de series temporales de una manera fácil y agradable a través de una interfaz web. Esta información es recolectada y procesada por aplicaciones de terceros como pueden ser Cloudwatch, Graphite, Elasticsearch, OpenTSDB, Prometheus, Hosted Metrics o InfluxDB. Grafana nos ayuda a estudiar, analizar y monitorear datos durante un período de tiempo, permitiendo consultar, visualizar, configurar alertas y, en general, comprender los datos fácilmente con la ayuda de métricas.

El tablero de instrumentos está bastante equipado y evoluciona continuamente para dar sentido a los datos complejos. La herramienta tiene una gran cantidad de opciones de visualización para comprender los datos según nuestros requisitos comerciales, desde la visualización de gráficos hasta mapas de calor, histogramas, mapas geográficos.

En este proyecto, Grafana está instalada en el puerto 3000 y se usa principalmente para monitorizar todos los datos que se reciben de los sensores instalados en los dispositivos representándolos en gráficas y tablas. También se encarga de enviar alertas por correo electrónico en el caso en el que los valores de alguna medida sobrepasaran unos límites establecidos. En Figura 62 se puede observar el “dashboard” de un módulo ESP82, donde se ven representados todos los datos de los sensores analizados anteriormente en tiempo real.

### 7.3 Personal

En cuanto al personal que ha elaborado este proyecto han sido mi tutora de proyecto Amelia, que es la que me ha dado la idea y el que ha realizado el proyecto he sido yo, con la ayuda de Amelia.

### 7.4 Presupuesto

El presupuesto que he elaborado tampoco es tan elevado, teniendo en cuenta que el que he dado es para una sola aula, posteriormente si se quiere hacer en las demás aulas lo que se tendrá que hacer es multiplicar el total por todas las aulas que se desee instalar.



Presupuesto											
Nombre			Emilian Scurtu			Datos cliente					
Dirección			C\ La Vendimia Nº16			Nombre			I.E.S Gran Capitan		
Teléfono			643354395			Dirección			C\ Arcos de la Frontera		
E-mail			a19scemem@iesgrancapitan.org			Teléfono			957379710		
Fecha presupuesto				25-mar-21							
DESCRIPCIÓN						CANTIDAD	% DTO.	PRECIO DTO.	TOTAL		
Wemos D1 mini Pro						1	0%	4,43 €	4,43 €		
Jumpers						1	0%	0,20 €	0,20 €		
Cable conexión USB						1	0%	2,66 €	2,66 €		
DHT22						1	0%	7,52 €	7,52 €		
Pantalla LCD						1	0%	3,60 €	3,60 €		
Protoboard						1	0%	3,00 €	3,00 €		
MHZ-19B						1	0%	28,00 €	28,00 €		
						Total Bruto			49,41 €		
Total presupuesto									49,41 €		
Firma de la persona que confecciona el presupuesto.						ACEPTO EL PRESUPUESTO. Nombre, apellidos y firma del cliente.					

Este precio es muy inferior al precio de mercado de un medidor de CO2 ya montado y que se pueden encontrar, con una seguridad en la medición de los datos, a partir de 50€.



Medidor de CO2 Dioxcare PDF, Sensor NDIR de Alta precisión con envío de Datos PDF al Ordenador, Detector de CO2, Temperatura y Humedad

Marca: dioxcare

★★★★☆ 31 valoraciones | 12 preguntas respondidas

Precio recomendado: 429,00 €

Precio: **118,90 €** prime GRATIS Envío 1 día

Ahorras: 10,10 € (8%)

Precio final del producto

☐ Cupón ☐ Aplicar cupón de 5%

Detalles

**Financiación:** 30,47 € x 4 cuotas en 90 días con Paga en 4 de Cofidis (Total adeudado 121,87€, Coste del crédito 2,97€,TAE 21,28%.)

- ✓ El medidor de CO2 DIOXCARE detecta el Dióxido de Carbono del ambiente, la temperatura y la humedad en tiempo real a través del sensor NDIR de alta precisión que incorpora basado en el principio de absorción de fuentes de luz infrarroja.
- ✓ Permite exportar en PDF los datos al Ordenador de forma rápida y sencilla. Puede almacenar hasta 999 grupos de registros de datos y ofrece un histórico de tendencias.
- ✓ Medidor de CO2 de tamaño compacto y portátil con batería.

Como se puede apreciar en esta imagen este medidor de CO2 vale unos 120€ aproximadamente.

## 8. Conclusiones

### 8.1 Grado de consecución de objetivos

Todos los objetivos que me he propuesto para el proyecto lo hemos implementado tanto la pantalla LCD, que ha sido la última que ha venido, lo único que ha fallado ha sido la implementación del sensor de CO2, ya que el que compramos era falso y estamos a la espera de que llegue el nuevo sensor para poder ponerlo. El código ya está hecho y los pines a los que se tienen que conectar también, solo falta que llegue el sensor y conectarlo.

En cuanto al sensor de temperatura, tuvimos el mismo problema, el primero que compramos fue el DHT11 y tampoco iba bien, lo tuvimos que cambiar por el DHT22.

Objetivos:

- Configuración e implementación pantalla LCD: totalmente terminado y puesto en marcha para que funcione.
- Configuración e implementación sensor temperatura: totalmente terminado y puesto en marcha para que funcione.
- Configuración e implementación sensor CO2: implementado a nivel software, pero a nivel hardware está pendiente que llegue, ya que la configuración está hecha pero no hemos llegado a implementarlo debido a que el sensor de Co2 no ha llegado a tiempo.

Todos los objetivos se han cumplido, menos el del sensor de Co2, ya que todavía no ha llegado.

## 8.2 Problemas encontrados

Los principales problemas que me he encontrado han sido a la hora de la instalación de la parte de electrónica, ya que nunca he trabajado con componentes electrónicos. Para esta parte lo que he tenido que hacer es informarme más para así poder hacer que todos los sensores puedan recolectar datos. Para esta parte me ha sido de gran ayuda un grupo de Telegram que he encontrado que se llama el Chat de Mariete, este grupo está lleno de personas que le gustan la electrónica y que no dudarán en ayudar cuando lo necesites.

Al estar en este grupo y ver que no funciona el sensor de CO2, les he mandado fotos y me han dicho que ese sensor es falso, al final tuvimos que comprar otro para sustituir el falso pero para la hora de la presentación todavía no había llegado.

El código ha sido también un gran reto, ya que he tenido que buscar todas las librerías, había librerías que no funcionaban y se han tenido que buscar en páginas como GitHub.

Tras la puesta en marcha del código el problema que he tenido es que no conseguía que la placa Wemos D1 Mini se conectará a internet, hay que tener en cuenta que tiene que ir por la banda de 2,4GHZ y no la de 5GHZ.

Después de eso hay que tener en cuenta también los puertos que se tienen que tener abiertos, todos los puertos que se tienen que tener abiertos están en la guía técnica.

En cuanto a la parte de informática, también he tenido problemas, sobre todo en la parte de las alertas, ya que se tenían que configurar para gmail y en el archivo de configuración me he liado un poco, la configuración que se tiene que tener para que vaya bien está en la guía técnica.

## 8.3 Futuras mejoras

Si la prueba de concepto se considera de valor, el siguiente paso es implementar este sistema de forma real, es decir ponerlo en producción. Usando los servidores del instituto como máquina física para instalar los servicios que se han detallado en la guía técnica.

Desarrollar una página web, que cuelgue de la página del Instituto, y con acceso público para que los padres y alumnos, puedan ver en todo momento y en tiempo real, la calidad del aire del aula donde sus hijos están desarrollando su actividad educativa. Para tranquilidad de padres, alumnos y comunidad educativa.

Aumentar otros parámetros (a través de sensores) que se consideren oportunos para recopilar más datos que sean de importancia para el control de las aulas u otros espacios.

Diseñar y fabricar mediante impresoras 3D una carcasa para ocultar todos los elementos (sensores, placa, jumpers, portoboard) del medidor medioambiental. De forma que sólo sea visible la pantalla para consultar valores.



Así es como se podría quedar, con unas rejillas para que los sensores recolecten los datos y con unos hueco para las pantallas.

Otra posible mejora sería que si por ejemplo se llega a una temperatura bastante elevada en el aula, salte una alerta y la alerta lo que hace es que encienda el aire acondicionado de la clase, para que así se refresque el aula.

## 9. Referencias/bibliografía

<https://www.jorgedelacruz.es/2020/11/23/en-busca-del-dashboard-perfecto-influxdb-telegraf-y-grafana-parte-i-instalando-influxdb-telegraf-y-grafana-sobre-ubuntu-20-04-lts/>

<https://www.luisllamas.es/wemos-d1-mini-una-genial-placa-de-desarrollo-con-esp8266/>

<https://www.luisllamas.es/como-instalar-mosquitto-el-broker-mqtt/>

<http://josejuansanchez.org/iot-dashboard/>

<https://github.com/josejuansanchez/co2-celia>

<https://www.vultr.com/docs/how-to-install-mosquitto-mqtt-broker-server-on-ubuntu-16-04>

<https://sg.com.mx/revista/51/hola-mundo-iot>

También tengo que mencionar el chat de Mariete de Telegram, que me han ayudado mucho, se accede [aquí](#).

También decir que he buscado más información en muchos foros de electrónica.

## 10. Anexos

Lo que hemos utilizado principalmente ha sido la nube de [AWS](#) educate, creando una máquina virtual, con la cuenta corporativa del instituto.

Después nos hemos ayudado de [Grafana](#), [Influxdb](#), [Telegraf](#) y [Mosquitto](#).

También destacar la guía técnica, que la tengo subida a mi Drive, se puede consultar [aquí](#).