

Informe final

Juan Enrique Azar, Santiago Garaventta, Kevin Giribuela

8 de Agosto - 2022

Índice

1. Introducción	2
2. Componentes del proyecto	3
3. Etapas del Proyecto	3
3.1. Laboratorio 1: Constitución del Proyecto	3
3.2. Laboratorio 2: Diseño de la aplicación	4
3.3. Laboratorio 3: Incorporación de Beebotte	5
3.4. Laboratorio 4:	6
3.5. Laboratorio 5: Construcción del Circuito	8
4. Registro temporal de las actividades del proyecto	9
4.1. Programación del ESP32	10
4.2. Diseño del gabinete	10
4.3. Diseño lógico de la App	10
5. Resultados obtenidos	10

1. Introducción

En el presente informe se realiza el cierre de proyecto presentado a principio de semestre. Con el fin de visualizar los avances realizados, conocimientos adquiridos, modificaciones y obstáculos que se fueron presentando a lo largo del proyecto se compara con lo propuesto en informes anteriores.

Como se sabe, AirQuality es un dispositivo diseñado para poder realizar tareas de monitoreo de variables ambientales en un aula. La idea surge a partir de la problemática presente en los tiempos post COVID-19 donde la vuelta a la presencialidad en las aulas comenzó a ser una actividad frecuente. Los protocolos vigentes en sus momentos eran estrictos (en relación a los actuales) y era de suma importancia mantener las aulas ventiladas para evitar que el aire se estanque y sea mas probable que una persona contraiga COVID-19 debido a la pobre circulación de aire.

Una forma de monitorear este tipo de situaciones es la de medir el nivel de concentración de dióxido de carbono presente en el aire. En base a esta medida, se puede tener conocimiento acerca de qué tan mala es la ventilación que existe en el salón.

Sin embargo, medir el CO₂ para saber cuántas partículas por millón hay presente en el aire resulta una tarea en la cuál sólo es necesario un sensor y un microcontrolador que señalice de alguna forma tal parámetro.

La idea propuesta para AirQuality es la de diseñar y sintetizar un dispositivo de fácil instalación, cuya fuente de alimentación sea la tensión de línea presente en las aulas, mida (además de niveles de CO₂) temperatura, humedad y luminosidad (parámetros con los cuales pueden realizarse estudios de ergonomía dentro del aula), y envíe esos valores medidos a una plataforma de *Internet of Things* para poder tener una visualización pública de los datos.

Dado a que la tarea a realizar necesita de algún microcontrolador capaz de establecer una comunicación a internet, la elección del dispositivo que realice dicha tarea es discutida y analizada en profundidad tomando conocimiento de los posibles problemas que puedan surgir a futuro según la elección del microcontrolador. El criterio de selección debe contemplar costos, disponibilidad en el mercado y nivel de complejidad.

Dicho criterio es aplicado en igual forma a la selección de los sensores que midan las variables ambientales mencionadas previamente.

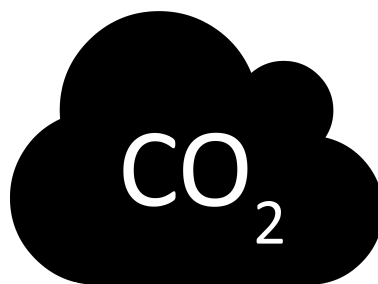


Figura 1: Logo del proyecto

2. Componentes del proyecto

Para la elaboración de este proyecto, se utilizaron relativamente pocos componentes, a diferencia de la propuesta inicial, esto motivado principalmente por el la utilización de un sensor de CO2 el cual dispone también de medición de temperatura y humedad. Los componentes utilizados y sus respectivos precios son:

- ESP32: Cantidad 1, Precio \$2.200[c/u].
- Plac Fenolico: Cantidad 1, Precio \$350[c/u].
- Caja Estanca 10x10x6: Cantidad 1, Precio \$400[c/u].
- SCD30 (Sensor CO2): Cantidad 1, Precio U\$ 62,7[c/u].
- BH1750 Sensor Luminosidad: Cantidad 1, Precio \$750[c/u].
- LED (RGB): Cantidad 1, Precio \$ 50[c/u].
- Componentes Varios: Precio \$500[c/u].

Resulta en un precio final para el proyecto de aproximadamente \$23.000.

3. Etapas del Proyecto

3.1. Laboratorio 1: Constitución del Proyecto

La primera etapa del proyecto consistió principalmente en la planificación del mismo, es decir, el establecimiento de metas tanto temporales como de entregables al finalizar el proyecto.

Estas etapas pueden verse en el diagrama de Gantt de la Figura 2:

Nro	Tareas	Inicio	Final	Fin de proyecto
1	Diseño de fuente	1/4/2022	6/4/2022	30/6/2022
2	Diseño de esquemático	7/4/2022	17/4/2022	
3	Diseño de PCB	18/4/2022	18/5/2022	
4	Diseño de gabinete	19/5/2022	29/5/2022	
5	Diseño gráfico de App	1/4/2022	11/4/2022	
6	Diseño lógico de App	12/4/2022	11/6/2022	
7	Programación ESP32	1/4/2022	30/6/2022	

Figura 2: Diagrama de Gantt del Inicio del Proyecto

- **Diseño Fuente:** en esta etapa del proyecto se realizó no solo la investigación respecto a las distintas alternativas de alimentación, sino que también se decantó por el uso de una fuente la cual permite directamente convertir 220Vac a 3,3Vdc.
- **Diseño esquemático:** En esta etapa, se diseñaron todas las interconexiones entre el microcontrolador y los sensores utilizados, así como también del circuito externo necesario para flasharlo.
- **Diseño PCB:** esta etapa consistió del diseño del circuito impreso, considerando la restricción espacial del mismo, dado que este debía estar contenido en una caja de 10x10x6, lo cual nos obligó a utilizar únicamente el módulo ESP32, en lugar de la placa de desarrollo.
- **Diseño de gabinete:** Esta etapa consistió de la elección de la caja contenedora del dispositivo, así como también de la determinación de la ubicación en el mismo, de la PCB, los sensores y el led.
- **Diseño gráfico de la App:** esta etapa constó de la optimización de la aplicación móvil, buscando que esta resultará lo más amigable posible para el usuario.
- **Diseño lógico:** en esta etapa se diseñaron y posteriormente implementaron, todas las comprobaciones, conexiones a bases de datos y comunicaciones por MQTT de la aplicación con el broker.
- **Programación ESP32:** en esta etapa se definió el esquema de funcionamiento del ESP32, la conmutación entre Acces Point y Station, así como la programación de estos modos junto con el resto de código y bibliotecas necesarias para poder utilizar los sensores.

3.2. Laboratorio 2: Diseño de la aplicación

Esta etapa del proyecto consistió del diseño de la aplicación móvil utilizada para el monitoreo de las variables del ambiente medidas por los sensores:

Nro	Tareas	Duración
1	Determinación del entorno de desarrollo de la App	2
2	Estudio de la plataforma seleccionada	10
3	Selección de parámetros de la aplicación	4
4	Desarrollo de la interfaz gráfica	4
5	Ensayos de la App	4
6	Corrección de errores y mejoras	4

Figura 3: Diagrama de Gantt de la aplicación

- **Determinación del Entorno de Desarrollo de la App:** Para esto se analizan las distintas alternativas disponibles para el diseño de aplicaciones como *App Inventor*, *Kodular*, etc. Haciendo principal hincapié en la disponibilidad de información sobre las mismas, de la comunidad que las utiliza y de la cantidad de extensiones disponibles para optimizar y brindar nuevas funcionalidades a nuestra aplicación, como por ejemplo retoques estéticos.
- **Estudio de la Plataforma Seleccionada:** Esta etapa se basa principalmente en una investigación a fondo de las herramientas disponibles por el entorno, así como del desarrollo de algunos componentes que permitan facilitar su programación posteriormente, como puede ser el caso de la conectividad MQTT entre otros.

- **Selección de Parámetros de la Aplicación:** Se determina el nombre de la misma *Air Quality UNLP*, así como también su esquema de colores y las diversas funcionalidades con las que contará.
- **Desarrollo de la Interfaz Gráfica:** Esta consiste en diseñar las distintas pantallas de la aplicación así como también de que forma se el usuario interactuará con estas.
- **Ensayos de la App:** Esta etapa consiste principalmente en la prueba a fondo de la aplicación, buscando *bugs* en la aplicación que deban ser solucionados. También se observa y se toma nota de las posibles dificultades en el uso que le puedan llegar a surgir al usuario.
- **Corrección de errores y mejoras:** Se utiliza este tiempo para la corrección de fallas detectadas en la etapa anterior, así como también para la implementación de posibles mejoras que puedan haber surgido de la prueba de la aplicación.

En esta etapa la modificación principal que se hizo sobre la app, respecto de la propuesta original, fue el hecho de no contar con notificaciones en segundo plano respecto al nivel de CO₂, debido a la ausencia de herramientas por parte de Kodular para este tipo de tareas.

3.3. Laboratorio 3: Incorporación de Beebotte

En esta etapa del proyecto se realizó la incorporación en la aplicación de un servidor como Beebotte, el cual cuenta con un broker MQTT, capaz de recibir datos y mostrarlos en un dashboard.

Configured resources		
Temperatura_BARCALA	23.98	3 days ago
Humedad_BARCALA	53.18	3 days ago
CO ₂ _BARCALA	1170.24	3 days ago
Luminosidad_BARCALA	0	3 days ago

Esto permitió la incorporación a la aplicación de la posibilidad de observar en tiempo real la evolución temporal de las variables así como también darle al usuario la posibilidad de visualizar distintos periodos de tiempo para poder así analizar los datos.



3.4. Laboratorio 4:

En esta etapa del proyecto, se realizaron las modificaciones necesarias según el avance del proyecto, las limitaciones y los problemas encontrados, es decir, la re definición del proyecto.

También se explicó la elección de los sensores y del microcontrolador y se lo comparó con las demás opciones que estuvieron presentes y no fueron seleccionadas.

De todos modos, lo más importante de esta entrega fue el funcionamiento de la conexión del microcontrolador con un dispositivo y la robustez en caso de que haya problemas de red o se le pasen datos incorrectos de red.

Los siguientes diagramas representan el funcionamiento en esos casos.

■ Puesta en marcha

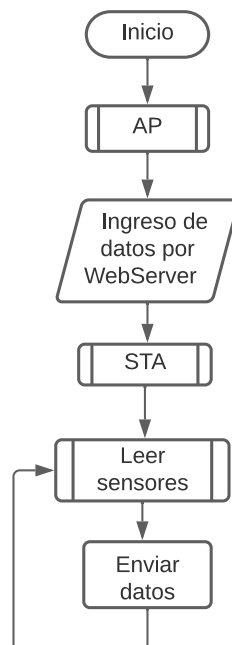


Figura 4: Puesta en marcha típica.

■ Pérdida de conexión o datos incorrectos

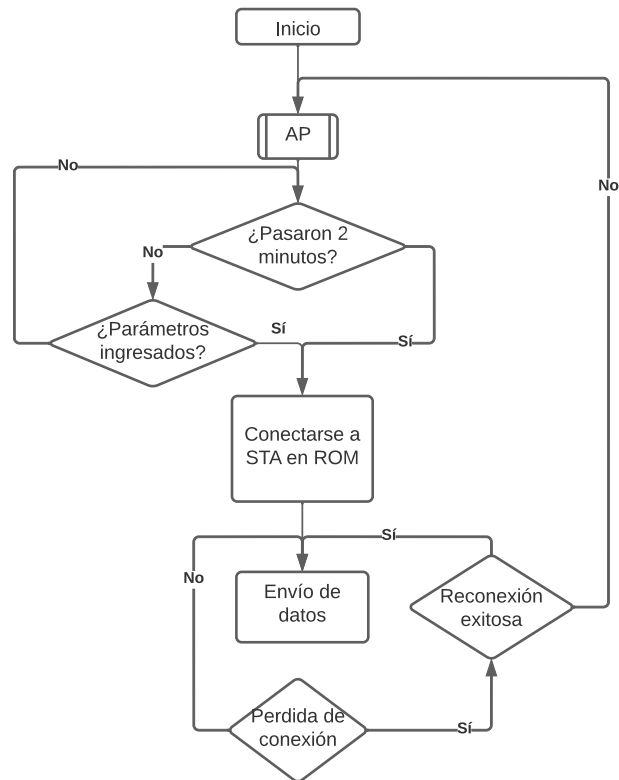


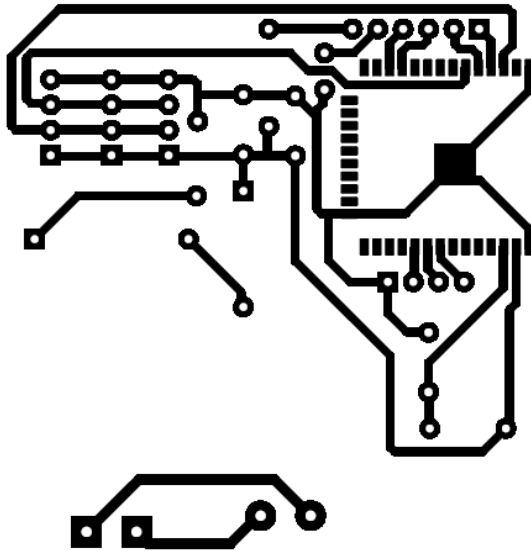
Figura 5: Esquema de reconexión.

Para finalizar, también se realizó la señalización con el led RGB y el significado de cada color.

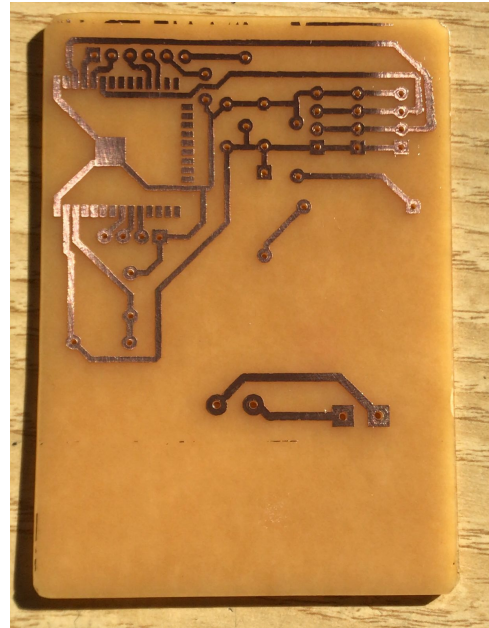
- Modo AccesPoint: Azul.
- WiFi desconectado: Rojo.
- WiFi conectado: Verde.
- MQTT correcto: Amarillo.
- MQTT incorrecto: Rojo.
- Envío de datos: Destello rojo.
- WiFi y MQTT correctos: Violeta.

3.5. Laboratorio 5: Construcción del Circuito

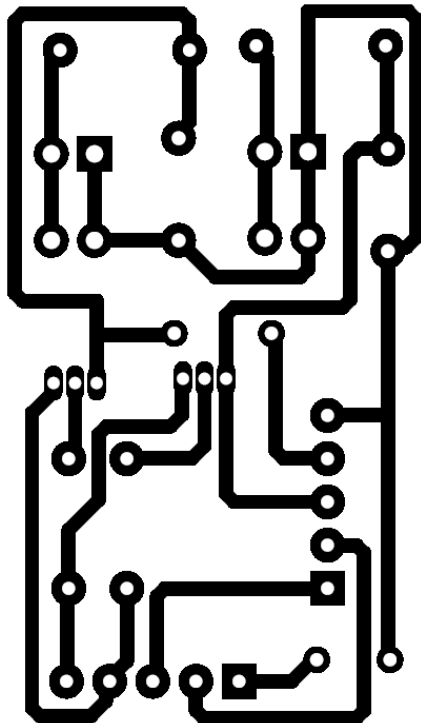
En esta etapa del proyecto, ya finalizando el mismo, se procedió a la fabricación y montaje del circuito final, tanto del que incluye al ESP32 como del circuito necesario para programar el mismo.



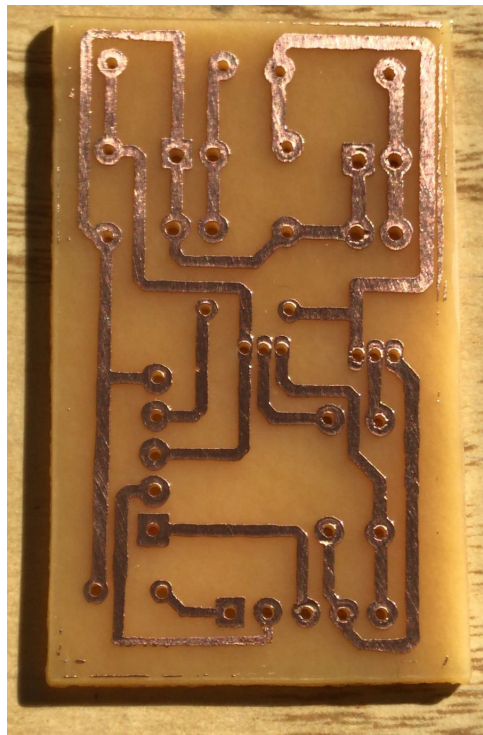
(a) PCB



(b) Placa



(a) PCB



(b) Placa

La construcción del circuito consistió principalmente de 4 etapas:

1. **Diseño circuito impreso:** el diseño de ambos circuitos fue realizado utilizando el software *Altium Designer*, definiendo en este la ubicación de los componentes así como de las respectivas conexiones entre estos.
2. **Transferencia del Circuito al PCB:** esta etapa se realizó mediante el método de la plancha, recibiendo ayuda del ATEI, imprimiendo el esquemático en una hoja de ilustración satinado de 145g y luego colocándola sobre la PCB transfiriendo la tinta de la misma mediante calor utilizando una plancha.
3. **Eliminación del cobre sobrante del PCB:** para retirar el cobre sobrante y dejar solo las pistas necesarias para nuestro circuito, se sumergió la placa, ya con el esquemático transferido en una solución de ácido clorhídrico para poder remover todo el cobre que no estuviera cubierto por la pista.
4. **Montaje de los componentes:** Por último se procedió al montaje de todos los componentes necesarios para el funcionamiento del proyecto, comenzando primero por el ESP32 ya que era smd, y siguiendo con todos el resto de componentes through hole.

Por último, previo a la programación final con el programa completo, se procedió a la prueba del circuito flasheador y al del propio proyecto, colocando un led en uno de los pines disponibles y haciéndolo titilar, previamente habiendo corroborado que no se encontrara nada en cortocircuito mediante la utilización de un voltímetro en modo de continuidad.

4. Registro temporal de las actividades del proyecto

Como se ha mencionado, en la Figura 2 se presenta la cantidad de tiempo proyectado a cada tarea cuando se dio inicio al proyecto. De manera similar, en la Figura 6 se muestra cuánto tiempo se le dedicó realmente a cada tarea.

Puede observarse como la cantidad de tiempo propuesto para el diseño lógico de la aplicación móvil y la programación del microcontrolador fueron subestimados.

Nro	Tareas	Inicio	Final	Fin de proyecto
1	Diseño de fuente	1/4/2022	6/4/2022	9/8/2022
2	Diseño de esquemático	7/4/2022	17/4/2022	
3	Diseño de PCB	18/4/2022	18/5/2022	
4	Diseño de gabinete	19/5/2022	9/8/2022	
5	Diseño gráfico de App	1/4/2022	11/4/2022	
6	Diseño lógico de App	12/4/2022	9/8/2022	
7	Programación ESP32	1/4/2022	9/8/2022	

Figura 6: Diagrama de Gantt del Final del Proyecto

Como puede apreciarse las diferencias entre la planificación inicial de los tiempos del proyecto, y los reales, obtenidos luego de la finalización del mismo se deben principalmente a tres etapas.

4.1. Programación del ESP32

Esta etapa, y en particular, la comunicación y lectura de datos del sensor de CO2 *SCD30*, cuya comunicación con el ESP32 estaba prevista que fuera mediante un BUS I2C, dado que el resto de sensores utilizaban el mismo protocolo de comunicaciones. Sin embargo luego de reiteradas pruebas e investigación, se encontró que el sensor requería de un *ClockStretching* de 13ms, mientras que el ESP32 solo soportaba hasta 7ms. Por lo tanto, fue necesario cambiar la forma de comunicación, utilizando el resto de sensores por I2C, mientras que para el SCD30 se hizo uso de MODBUS, lo cual llevo a tener que implementar otra nueva biblioteca para este, postergando significativamente la conclusión de esta etapa.

4.2. Diseño del gabinete

Esta etapa, se vio muy retardada principalmente debido a la incertidumbre de que sensor de CO2 iba a ser utilizado, ya que, dado el problema de comunicaciones que existía, se barajaron varios sensores distintos como posibles remplazos. Por lo tanto, el diseño final del gabinete, con las conexiones necesarias y las respectivas ubicaciones de los distintos sensores dentro del gabinete, se tuvieron que postergar hasta no tener definidos el grupo final de sensores, los cuales resultaron ser 2, el de luminosidad y el de CO2, ya que este ultimo también mide temperatura y humedad, haciendo que no sea necesario otro sensor extra.

4.3. Diseño lógico de la App

Esta etapa sufrió modificaciones principalmente debido a dos factores, uno de ellos fue la incorporación a la aplicación, de dos parámetros adicionales para la configuración del MQTT, el Usuario y Contraseña, los cuales permiten utilizar plataformas como Beebotte que brinda la opción de disponer de un Dashboard, el cual fue incorporado a la aplicación como otra opción para el usuario.

El otro factor que demoro el diseño lógico de la aplicación fue el requerimiento inicialmente propuesto de contar con notificaciones en segundo plano para el nivel de CO2, las cuales luego de muchas iteraciones y pruebas con herramientas de Kodular y externas, no fue posible lograr obtenerlas, ya que este entorno de desarrollo de aplicaciones no permite dejar procesos de aplicaciones corriendo en segundo plano, por lo tanto, no pueden obtenerse notificaciones sobre el nivel del CO2 cuando la aplicación no se encuentra abierta.

5. Resultados obtenidos

En esta sección se discuten los resultados obtenidos, la calidad del dispositivo y su alcance.

El dispositivo final integrado puede observarse en la Figura 7. Allí queda en evidencia varios de los puntos propuestos en primer momento tales como la de utilizar la tensión de línea para alimentar al dispositivo.



Figura 7: Resultado final

Nótese la necesidad de colocar perforaciones exteriores al gabinete para permitir la circulación de aire dentro del mismo, la visualización del led indicador y al mismo tiempo la incidencia de luz para obtener una correcta medida de intensidad lumínica.

El proyecto resultante está contenido en una caja estanca de $10\text{cm} \times 10\text{cm}$. Dentro de ella se encuentra la placa electrónica junto con los sensores encargados de medir las variables ambientales.