



Proyecto Final de Carrera

Ingeniería en Telecomunicaciones

---

**“Nodo GPRS para Control y Monitoreo de Material Particulado”**

Francou Agustín

Directores:

Farina Mario

Primer semestre 2021.

## ÍNDICE GENERAL

|                                                           |    |
|-----------------------------------------------------------|----|
| ÍNDICE DE FIGURAS Y TABLAS .....                          | 4  |
| PALABRAS CLAVE .....                                      | 6  |
| RESUMEN .....                                             | 7  |
| Capítulo 1: Introducción al proyecto .....                | 8  |
| 1.1 TÍTULO .....                                          | 8  |
| 1.1.1 MOTIVACIÓN .....                                    | 8  |
| 1.2 OBJETIVO GENERAL.....                                 | 10 |
| 1.2.1 OBJETIVOS ESPECÍFICOS .....                         | 10 |
| 1.3 ESTRUCTURA DEL PROYECTO .....                         | 11 |
| Capítulo 2: Contextualizando IoT y conceptos claves ..... | 12 |
| 2.1 ANTECEDENTES .....                                    | 12 |
| 2.2 MARCO TEÓRICO .....                                   | 16 |
| 2.2.1 CALIDAD DE AIRE EN ARGENTINA .....                  | 16 |
| 2.2.2 INTERNET OF THINGS .....                            | 17 |
| 2.2.3 Principios y Tecnologías IoT.....                   | 18 |
| Actuación .....                                           | 19 |
| Geolocalización .....                                     | 19 |
| Plataformas IoT .....                                     | 19 |
| 2.2.4 HARDWARE IoT .....                                  | 20 |
| Microprocesadores y Microcontroladores .....              | 20 |
| Sensores .....                                            | 20 |

|                                                              |    |
|--------------------------------------------------------------|----|
| Conectividad y normalización .....                           | 21 |
| La gestión de dispositivos .....                             | 21 |
| Almacenamiento de datos .....                                | 22 |
| Procesamiento y Gestión de las acciones .....                | 22 |
| La Analítica .....                                           | 22 |
| 2.2.5 TECNOLOGÍAS DE COMUNICACIÓN .....                      | 24 |
| LoRa .....                                                   | 24 |
| LoRaWAN .....                                                | 24 |
| GPRS .....                                                   | 24 |
| Capítulo 3: Diseño y modelado de la solución .....           | 26 |
| 3.1 INTERCONEXIÓN DE HARDWARE .....                          | 30 |
| 3.2 PARAMETRIZACIÓN DE LA SOLUCIÓN .....                     | 32 |
| 3.3 DISEÑO DE SOFTWARE .....                                 | 33 |
| 3.4 PROTOTIPO .....                                          | 45 |
| Capítulo 4: Experimentación, resultados y conclusiones ..... | 50 |
| 4.1 MEDICIONES .....                                         | 50 |
| 4.2 RESULTADOS .....                                         | 60 |
| 4.3 PROPUESTAS DE ACCIONES FUTURAS .....                     | 61 |
| 4.4 CONCLUSIONES .....                                       | 62 |
| ANEXOS .....                                                 | 63 |
| REFERENCIAS .....                                            | 75 |

## ÍNDICE DE FIGURAS Y TABLAS

### Figuras:

|     |                                                                             |    |
|-----|-----------------------------------------------------------------------------|----|
| 1   | Arquitectura General de IoT .....                                           | 23 |
| 2   | Arquitectura de la solución propuesta .....                                 | 26 |
| 3   | Esquema de Interconexión entre módulos.....                                 | 30 |
| 4   | Batería/Fuente PROBATTERY 12V 2A .....                                      | 31 |
| 5   | Primera Máquina de Estado del sistema de la solución .....                  | 34 |
| 5.1 | Segunda Máquina de Estado del sistema de la solución Primera Parte .....    | 35 |
| 5.2 | Segunda Máquina de Estado del sistema de la solución Segunda Parte .....    | 36 |
| 6   | Diagrama de flujo de función base de la solución .....                      | 37 |
| 7   | Interior del prototipo con vista superior delantera .....                   | 47 |
| 7.1 | Interior del prototipo con vista trasera .....                              | 48 |
| 7.2 | Vista frontal del prototipo .....                                           | 48 |
| 7.3 | Lateral Izquierdo del gabinete .....                                        | 49 |
| 7.4 | Lateral Derecho del gabinete .....                                          | 49 |
| 7.5 | Entrada de Aire y Frente del sensor .....                                   | 50 |
| 7.6 | Proyección interior .....                                                   | 50 |
| 8   | Visualización total de las muestras. ....                                   | 54 |
| 8.1 | Valores implicados parar escenario 1.....                                   | 55 |
| 8.2 | Valores implicados para escenario 2 .....                                   | 56 |
| 8.3 | Valor máximo reportado en escenario 3 .....                                 | 57 |
| 8.4 | Promedio de las muestras cada 15 minutos.....                               | 58 |
| 8.5 | Promedio de todo el período de mediciones .....                             | 58 |
| 8.6 | Panel de configuración de vista para obtener promedio cada 15 minutos ..... | 59 |
| 9   | SIM 800l v2 EVB reverso y frente del módulo .....                           | 64 |
| 9.1 | SIM 800l v2 EVB + Antena .....                                              | 64 |
| 10  | Módulo Sharp GP2Y1010AUoF .....                                             | 66 |

|    |                                                              |    |
|----|--------------------------------------------------------------|----|
| 11 | Ejemplo de conexiónado Sensor-Placa .....                    | 67 |
| 12 | Proceso de muestreo .....                                    | 68 |
| 13 | Método de detección de partículas que utiliza el sensor..... | 69 |
| 14 | Display I2C 16x2 .....                                       | 70 |
| 15 | Logo de la plataforma IOT ThingSpeak .....                   | 72 |

**Tablas:**

|     |                                                                                                 |    |
|-----|-------------------------------------------------------------------------------------------------|----|
| 1   | Listado de contaminantes y proporciones estipuladas permitidas por la Ley Nacional 20.284 ..... | 16 |
| 2   | Consumo de corriente por dispositivo .....                                                      | 32 |
| 3   | Especificaciones técnicas ARDUINO UNO R3 .....                                                  | 63 |
| 4   | Características técnicas SIM800l v2 [1] .....                                                   | 65 |
| 4.1 | Características técnicas del SIM800l v2 [2] .....                                               | 65 |
| 4.2 | Características técnicas del SIM800l v2 [3] .....                                               | 66 |
| 5   | Condiciones óptimas de trabajo del Sensor .....                                                 | 68 |
| 6   | Limitaciones de Licencia Gratuita ThingSpeak.....                                               | 73 |

PALABRAS CLAVE

- Internet of Things (IoT)
- ARDUINO UNO R3
- SIM800 L v2.0 EVB
- SHARP GP2Y1010AUoF
- PM 2.5
- PM 10
- ThingSpeak

## RESUMEN

El presente Proyecto Final de Carrera tiene por objeto el diseño, desarrollo, programación, puesta en marcha y ensamblado de un punto de sensado o nodo final, de calidad de aire y monitoreo, utilizando para ello, un sensor óptico encargado de la obtención de muestras de la concentración de partículas finas y gruesas presentes en el aire, con la cual se puede saber leyendo su valor, cuán limpio se encuentra el mismo. Seguido de obtener los datos, el nodo ha de elevar el reporte a una plataforma web remota en la cual se realiza el almacenamiento, analítica y presentación de los mismos, para lo cual, es necesario implementar un dispositivo y enlace de comunicación que permita tal fin cuando fuese necesario. Con el fin anterior, se implementa el módulo celular SIM800 L, el cual previamente a comunicar los datos que obtiene del sensor, se configura y asegura de contar con registro y acceso a la red del proveedor, además de las condiciones de enlace y conexión a la plataforma realizando las solicitudes y conexiones pertinentes.

Debido a que el sensor por sí mismo no puede comunicarse directamente con el módulo celular, se hace uso de una placa Arduino UNO, la cual permite la interconexión e intercomunicación de dispositivos, encargado de llevar la programación y rutinas de funcionamiento de todo el nodo en general, resolución de problemas, notificaciones y avisos, métodos de debugging/desarrollo, y asimismo, mostrar todo el proceso en tiempo real mediante un display de 16x2, para, en caso de fallas, poder determinar donde ha ocurrido el problema y/o cuándo se realizó el último reporte a la plataforma.

*Capítulo 1:**Introducción al proyecto*1.1 TÍTULO

“Nodo GPRS para Control y Monitoreo de Material Particulado”

1.1.1 MOTIVACIÓN

Este proyecto tiene como objeto hablando a grandes rasgos la construcción, testing y puesta en marcha de un prototipo enmarcado como prueba de concepto bajo el paradigma de comunicaciones conocido como Internet de las Cosas o por sus siglas en inglés IoT. El prototipo en cuestión procura ser un End Point de recabación de datos acerca de la concentración de partículas finas (como lo son las de polvo, humo, polen, etc.) en el ambiente circundante, el cual pretende transmitir dichos datos a una plataforma remota, donde, una vez allí, los mismos puedan ser visualizados por medio de gráficas analíticas por el usuario final, utilizando como enlace de comunicación, la red celular.

Siendo que el desarrollo de objetos inteligentes y el almacenamiento de datos en la nube se encuentra en un constante crecimiento día a día a nivel mundial, y la escasa presencia en la ciudad de Concepción del Uruguay de este tipo de soluciones ambientales o bien, de usos específicos de laboratorios o instituciones que precisan en su funcionamiento de aire limpio y de un monitoreo constante, hace factible la idea de, con un redimensionamiento a un producto comercial del prototipo, brindar una herramienta de fabricación local que permita monitorear y controlar la presencia de polvo o partículas finas en el ambiente, sirviendo como instrumento de prevención y análisis de los interesados o instituciones gubernamentales de la ciudad.

IoT representa de una forma completa y englobadora todos aquellos conocimientos adquiridos durante el transcurso de la carrera, permitiendo obtener una solución totalmente íntegra, de la cual es posible diferenciar a la perfección cada capa o etapa de desarrollo. Desde

aspectos relacionados al diseño, programación, interconexión e intercomunicación de módulos electrónicos, hasta configuración, acceso, respuesta ante fallos y consumo de datos en Internet, son todas aquellas temáticas y desafíos que el prototipo en cuestión propone e impulsa al momento de su desarrollo.

## 1.2 OBJETIVO GENERAL

- Diseñar, desarrollar y puesta en marcha de una prueba de concepto de nodo de sensado 2G para monitoreo y control de material particulado.

### 1.2.1 OBJETIVOS ESPECÍFICOS

- Lograr que el nodo en caso de fallas, cuente con los mecanismos suficientes para solucionar problemas referidos al establecimiento de conexión con el proveedor celular y, de no ser posible, notificar de alguna forma la causa del problema.
- Obtener y visualizar datos de relevancia por medio de una plataforma IoT para uso estadístico, predictivo y preventivo cómo será el caso de la concentración de partículas finas, en sus diferentes niveles moleculares o lo que la tecnología y el capital destinado al presente proyecto permita.
  - Objetivo indirecto: Fomentar e incentivar el desarrollo e implementación con IoT.

### 1.3 ESTRUCTURA DEL PROYECTO

Para lograr una mayor armonía y organización con el desarrollo de la prueba de concepto en cuestión, el proyecto se divide en 3 capítulos bien diferenciados entre sí. En el primero, más precisamente el Capítulo 2, se realiza una introducción teórica de todos aquellos fundamentos que deben ser tenidos en cuenta y conocidos al momento de llevar a cabo cualquier tipo de investigación, desarrollo o prueba relacionada con Internet de las Cosas o alguno de sus componentes o etapas. Se añaden aquí también, una recopilación de desarrollos e investigaciones realizadas algunos años atrás por particulares e instituciones relacionadas con la ingeniería que, a modo de expandir el pensamiento referido a los alcances de la materia resulta de un buen complemento.

Sin lugar a dudas, en donde está el núcleo del trabajo y de los desarrollos llevados a cabo durante este proyecto, es en el Capítulo 3, donde en el mismo se partirá desde la base tecnológica y electrónica elegida, subiendo paulatinamente en cada capa o nivel de desarrollo en detalle, dando las justificaciones correspondientes y motivos de elección tomados en cada caso ya sean tanto a nivel de hardware como de software.

Como última sección, se exponen los resultados obtenidos de todo el proceso de elaboración y diseño del prototipo, en conjunto con las respectivas mediciones realizadas en diferentes puntos de la ciudad, consideraciones generales y parámetros de aire limpio, y por último y no menos importante, las conclusiones obtenidas de todo el trabajo realizado complementado con las propuestas o mejoras que se le podrían aplicar a esta prueba de concepto para su mayor performance y futura comercialización.

## Capítulo 2:

### Contextualizando IoT y conceptos claves

#### 2.1 ANTECEDENTES

En la búsqueda de complementos, desarrollos, implementaciones, lo más relacionado posible a la idea y fines del prototipo, o asimismo al paradigma de IOT, se puede hacer mención y referencia a nivel nacional en Argentina, al CASE (Congreso Argentino de Sistemas Embebidos) donde año a año en cada una de las diferentes ediciones, representantes tanto de laboratorios, facultades e instituciones de renombre, concurren al congreso exponiendo sus avances, desarrollos y conclusiones sobre diferentes temáticas, que, dependiendo del año, se van rotando para lograr un complemento y relación entre tópicos para todos los oyentes.

La edición 2020, fue una edición especial, la cual se realizó de forma virtual, donde la misma se desarrolló a través de sesiones orales con expositores tanto de Argentina como de otros países hispanohablantes, haciendo foco a categorías como: “Bioingeniería” “Comunicaciones inalámbricas”, “Robótica” y “Protocolos y comunicaciones”, entre otras, donde IOT se vio fuertemente involucrada<sup>1</sup>.

Con respecto a Bioingeniería, el trabajo presentado fue la invención de un “Uroflujómetro IoT de uso domiciliario”, el cual fue desarrollado por integrantes de la Universidad Nacional de Mar del Plata y el CONICET. En el presente, se expone el diseño de un esquema de IOT para la rama de la Telemedicina y Salud, más precisamente un uroflujómetro basado en una balanza de precisión, donde luego de que las muestras son obtenidas llegan a un conversor analógico/digital (ADC), procesadas y elevadas a la nube, donde el profesional luego puede acceder remotamente y corroborar en tiempo real tanto los

---

<sup>1</sup> Enlace de acceso a los trabajos del CASE 2020 por área y categoría donde se brinda el acceso general a los documentos: <http://www.sase.com.ar/case/ediciones/case2020/trabajos2020/>

resultados, como el minuto a minuto del proceso, utilizando un microprocesador típico de redes WiFi como es el caso del ESP32 y para el seguimiento local de los datos o rutinas de trabajo, los mismos se muestran por un display. Bajo el contexto de la pandemia del virus COVID19, el objetivo del mismo fue no solamente generar una estructura remota de trabajo para el sistema de salud y atención a distancia de pacientes, preservando así el distanciamiento obligatorio que la propia pandemia requiere, agilizando el proceso de evaluación y diagnóstico de los análisis y el fácil acceso por parte del profesional hacia ellos. También, pero no menos importante, que, según los autores detallan, la solución tiene a favor el factor de la comodidad generando una notable mejora de efectividad al momento de realizar los estudios ya que, el paciente se encuentra en completa comodidad y tranquilidad, debido al confort de su domicilio, evitando así concurrir al consultorio con todo lo que conlleva, además de la solución en sí ser de fácil transporte y uso, lo que agiliza todo el ciclo ampliamente.<sup>2</sup>

En referencia a otra de las categorías de exposición del congreso, más precisamente “Protocolos y Comunicaciones”, se exhibieron desarrollos muy interesantes y sobre todo relevantes para complemento de este proyecto y en general, de todo el mundo del IOT y las Telecomunicaciones. Uno de ellos, titulado como “Aplicación de Internet de las Cosas con procesamiento de Datos”, a cargo de profesionales recibidos del Departamento de Control Automático, de la Universidad Central “Marta Abreu” de Las Villas, Cuba y del Departamento de Ingeniería Eléctrica y Electrónica, Universidad del Bio-Bio, Chile. En el mismo, se hace foco puntualmente en el desarrollo de una aplicación para una arquitectura IOT completa, la cual tiene por objeto utilizando Inteligencia Artificial, ser capaz de recaudar datos, almacenarlos localmente, procesarlos y segmentarlos por grado de importancia y categoría dependiendo de la variable que se trate, para una posterior transmisión a una plataforma de análisis y

---

<sup>2</sup> Uroflujómetro IoT de uso domiciliario:

[https://drive.google.com/file/d/126qNwDtVcMgS5\\_2qtoVWV5gRWdiE-j8K/view](https://drive.google.com/file/d/126qNwDtVcMgS5_2qtoVWV5gRWdiE-j8K/view). Consultado el 27 de enero de 2021.

visualización utilizando algoritmos de aprendizaje inteligente. Implementan un gateway IoT que funciona como servidor local, lo que posibilita el acceso a la información de los nodos sensores, aún sin conexión a Internet, con lo cual, para ser capaces de cumplir tal fin hacen uso de una Raspberry Pi 3 Modelo B, la cual no es más ni menos que una Computadora de Placa Simple sobre la cual se monta y se encuentran alojados tanto el sistema operativo de la computadora, la plataforma ThingsBoard IoT Gateway, la base de datos PostgreSQL para el almacenamiento de la información y el bróker de mensajería MQTT Mosquitto. El procesamiento de la información se realiza en un servidor Big Data en la nube, mediante Apache Spark, Apache Kafka y la plataforma IoT ThingsBoard, utilizando como objeto experimental, una base datos en conjunto para las prueba de campo, grandes quintas de cultivo de tomates, donde el fin último, era el monitoreo, control y comparación de eficiencia de los métodos de aprendizajes inteligentes propuestos por los implementadores, donde en palabras finales, resulta de conformidad total la respuesta del sistema, toma de decisiones y performance de la aplicación en general.<sup>3</sup>

Por último, en relación más específicamente con el prototipo desarrollado en el corriente proyecto final de carrera, además de, observar como en otros países del mundo los estudiantes pueden, bajo otras oportunidades, hacer desarrollos complementarios a la misma idea propuesta, implementando diferentes arquitecturas, plataformas, electrónica y programación. El proyecto en cuestión es presentado por Ronald Estiven Ríos Hernández y titulado como: “Prototipo de monitoreo de calidad de condiciones ambientales a bajo costo con tecnologías IOT” de la Facultad de Ingeniería de la Universidad de La Salle, Colombia donde en el mismo pretende realizar control y monitoreo de la calidad de aire y PM 2.5 y 10, pero con la diferencia en la cantidad de sensores, donde estos últimos requieren electrónica

---

<sup>3</sup>Aplicación de Internet de las Cosas con procesamiento de datos:

[https://drive.google.com/file/d/1CHJ77sHOUGVQEmlh\\_5xmb4Xzjzfl93G/view](https://drive.google.com/file/d/1CHJ77sHOUGVQEmlh_5xmb4Xzjzfl93G/view) Consultado el 27 de enero de 2021.

de conversión lógica adicional, y como buen punto ante las fallas de comunicación, suma una tarjeta SSD para respaldar y evitar pérdidas de muestras las cuales se guardan en un archivo de texto simple con fecha y hora, para que luego las mismas sean elevadas a la plataforma elegida por el estudiante como lo es Ubidots. La solución estaría conformada por una Raspberry pi como fuente principal de procesamiento, y como medio de interconexión, acceso y configuración para con los sensores y el módulo de comunicaciones, un Arduino Nano. Como protocolos y medios de enlace, utiliza el módulo LoRa EBYTE 433MHz y LoRaWAN como protocolo de enrutamiento en la red. Detalles interesantes que el ahora profesional recibido adhiere a su proyecto, es toda la normativa vigente en Colombia en relación a los aspectos que el pretende abarcar con su prototipo, así como también diversos análisis y conclusiones de la importancia de recabar datos de estas variables tan importantes para el cambio climático e indirectamente, la calidad del aire que todos constantemente llevamos a nuestro cuerpo.

## 2.2 MARCO TEÓRICO

### 2.2.1 CALIDAD DE AIRE EN ARGENTINA

La autoridad Sanitaria Nacional de la República Argentina es aquel ente encargado de fijar las normas, regulaciones y concentraciones para el control de la calidad del aire a lo largo de todo el territorio, y la misma lo hace a través de los estados del Plan de Prevención de Situaciones Críticas de Contaminación Atmosférica, conforme se establece en el Anexo II de la Ley Nacional 20.284 del año 1973.

Extraído directamente del Anexo mencionado anteriormente, se muestra a continuación la lista de los diversos contaminantes en conjunto con su proporción crítica permitida:

| Contaminante [unidad]           | Norma de calidad de aire                       |
|---------------------------------|------------------------------------------------|
| <b>CO ppm</b>                   | <b>10 ppm en 8 hs</b><br><b>50 ppm en 1 hs</b> |
| <b>NO<sub>x</sub> ppm</b>       | <b>0,45 ppm en 1 hs</b>                        |
| <b>SO<sub>2</sub> ppm</b>       | <b>0,03 ppm promedio mensual</b>               |
| <b>O<sub>3</sub> ppm</b>        | <b>0,1 ppm en 1 hs</b>                         |
| <b>Partículas en suspensión</b> | <b>150 mg/m<sup>3</sup> promedio mensual</b>   |
| <b>Partículas sedimentares</b>  | <b>1 mg/cm<sup>2</sup> cada 30 días</b>        |

Tabla 1: Listado de contaminantes y proporciones estipuladas permitidas por la Ley Nacional 20.284.

Dentro del apartado se incluyen definiciones que permiten en su conjunto, modelar el problema de la contaminación ambiental y control de calidad del aire, dentro de las cuales se detallan, según palabras textuales de la presente Ley como:

“Contaminación atmosférica: Se entiende por contaminación atmosférica la presencia en la atmósfera de cualquier agente físico, químico o biológico, o de combinaciones de los mismos en lugares, formas y concentraciones tales que sean o puedan ser nocivos para la

salud, seguridad o bienestar de la población, o perjudiciales para la vida animal y vegetal o impidan el uso y goce de las propiedades y lugares de recreación.

Fuente de contaminación: Entiéndese por fuente de contaminación los automotores, maquinarias, equipos, instalaciones e incineradores, temporarios o permanentes, fijos o móviles, cualquiera sea su campo de aplicación u objeto a que se lo destine, que desprendan a la atmósfera sustancias que produzcan o tiendan a producir contaminación atmosférica.

Emisión: Se entiende por emisión cualquier contaminante que pase a la atmósfera como consecuencia de procesos físicos, químicos o biológicos. Cuando los contaminantes pasen a un recinto no diseñado específicamente como parte de un equipo de control de contaminación del aire, serán considerados como una emisión a la atmósfera.

Norma de calidad de aire: Se entiende por norma de calidad de aire todo valor límite de la concentración de uno y más contaminantes en la atmósfera.”

### 2.2.2 INTERNET OF THINGS:

Lo primero a definir para dar encuadre y contexto a este proyecto, es básicamente el paradigma y esquema de comunicación sobre el que se asienta el prototipo desarrollado, IoT (Internet of Things por sus siglas en inglés) o más bien conocido como el Internet de las Cosas. Es definido como aquella red que pretende desde todo punto de vista técnico/tecnológico, crear escenarios de monitoreo y control remoto, emulando situaciones reales que requerirían presencia humana, en donde gracias a objetos (cosas) de todo tipo, los cuales se caracterizan principalmente por poseer INTELIGENCIA, son capaces de tomar decisiones, operar y transmitir datos a cualquier parte del planeta haciendo uso de la red de redes (Internet), sin la necesidad siquiera de la intervención humana. El humano en este paradigma, pasa a ser el consumidor final de todos aquellos resultados (informes, reportes, mediciones, etc) que las cosas obtienen recabando, procesando y elevando datos del mundo real hacia la red, u hasta incluso, otro objeto.

De manera simbólica, si se habla de IoT, se habla de objetos inteligentes, donde justamente la inteligencia es la clave que hace que todo el ecosistema de cosas funcione y provea los resultados que se observan en la actualidad, donde la misma puede definirse como:

$$\text{SENSORIZACIÓN} + \text{CONECTIVIDAD} + \text{PROCESAMIENTO} = \text{INTELIGENCIA}$$

### 2.2.3 PRINCIPIOS Y TECNOLOGÍAS IOT

IoT de forma muy genérica y simple, es una representación a gran escala de la red sensorial humana, es decir, en base a un conjunto de puntos de captura, los mismos recaban datos periódicamente, los cuales son reportados a un sistema superior que se encarga de, en base a lo adquirido, procesar esos datos y tomar las decisiones correspondientes a los reportes obtenidos.

Por ello, las bases de IOT como la red propia de objetos, comprende:

Comunicación y cooperación: Los objetos son capaces de estar conectados entre sí a través de plataformas y protocolos de comunicación que establecen la “forma de hablar” de los objetos en cuestión.

Capacidad de direccionamiento: Los objetos, al igual que las personas, poseen datos únicos de identificación que nos hace poder ser identificados con respecto a los demás individualmente. Para los objetos, esta normativa se mantiene, pero son configurados a través de IDs o códigos, ya sea por unidad, o por grupo en el caso de que existan varias unidades del mismo tipo.

Detección y obtención: Es la función principal y primordial de los componentes finales de una red IOT, adquirir datos, es decir, que en base al entorno donde se encuentre el objeto, éste último debe ser capaz de detectar cambios en ese entorno (o de alguna de las variables que lo conforma) y a su vez, ser capaz de reportar ese dato que acaba de obtener por algún medio de transmisión que el objeto tenga.

## Actuación

Los objetos pueden y en muchos casos debería tener actuadores que les permita manifestarse a su entorno, es decir que, los mismos posean algún mecanismo de inteligencia ya sea en base a una petición, o acciones periódicas, que le permitan al objeto interactuar con el entorno que lo rodea.

## Geolocalización

Los objetos inteligentes en muchos casos deben reportar su ubicación física para su control, seguimiento y estado remoto.

Interfaces de usuario: Es decir, la forma en que los reportes de los objetos y posibles acciones que las personas podemos tomar u accionar de ellos, ordenadas y presentadas en pantalla de la manera más amigable posible.

## Plataformas IoT

Las Plataformas IoT son las soluciones en una implementación de internet de las cosas que nos permite observar la interacción entre los distintos dispositivos de la infraestructura.

Los ejes estructurales de toda plataforma de IoT con las que toda aquella mínimamente debería contar serían:

- **API:** La Interfaz de programación de aplicaciones, la cual es el conjunto de rutinas que proveerá acceso a las funciones que tengan implementados los dispositivos de una determinada red o esquema IoT, además de ser el lugar donde se reciben todos los reportes hechos por los objetos para su futuro análisis.
- **Bases de datos:** Si dijimos anteriormente que todos los reportes vienen a parar a la API, estamos diciendo que los datos van a permanecer allí, y el espacio donde los datos se almacenan dentro de las APIs, son las bases de datos, sin estas sería imposible tener acceso, y utilizar toda aquella información que llegase a una plataforma. Las bases de

datos para poder almacenar, deben hacerlo de una forma ordenada, estructurada, segura y escalable.

- Tratamiento de datos: Una vez que los datos son recibidos en la Plataforma IoT, ésta suele brindar, algunas veces, un catálogo de servicios adicionales como:
  - Agregación de datos (medias, máximos, mínimos) en distintos intervalos temporales, creación de reportes.
  - Análisis Big Data, o tratamiento de datos: el cual debería incluir tanto su seguridad (encriptación). como su transporte seguro.
  - Desarrollo y Hosting de aplicaciones.

#### 2.2.4 HARDWARE IoT

Hardware IoT comprende a todos aquellos componentes electrónicos, que dentro de sus principales funciones o características deben ser capaces de recolectar datos del mundo real, procesarlos internamente e intercomunicarse entre ellos, permitiendo así lograr conectarse a Internet por el medio disponible.

##### Microprocesadores y Microcontroladores

En el marco de IoT, los mismos tienden cada vez más a ser físicamente más pequeños, donde a su vez las tensiones de alimentación y su disipación de energía térmica son cada vez menores sin descuidar, la capacidad de procesamiento y almacenamiento interno del propio procesador, que recordemos es el cerebro y mente lógico de todo el sistema, que comanda toda acción y cálculo a realizar.

##### Sensores

Una de las características principales de los dispositivos IoT es su capacidad para observar el entorno y determinar cuál es su evolución, por lo que al fin y al cabo, un sensor no es más ni menos que un transductor, el cual puede ser definido como un traductor de áreas, con capacidades de interrelacionar áreas muy lejanas entre sí, por ejemplo, el mundo real, con

cualquiera de sus variables, al mundo eléctrico por ejemplo, pudiendo realizar una conversión de ese dato de una determinada área a un nuevo dato equivalente de la otra respectivamente.

Por lo que, se termina concluyendo, que los sensores son las piezas claves en cualquier sistema IoT, existiendo de diversos tipos y costos, los cuales dependen de la tecnología de fabricación, el elemento, o la física de la transducción, y también su grado de repetibilidad, exactitud, sensibilidad y capacidad de obtención de datos.

En cuanto al conjunto de bloques de una arquitectura IoT y, por ende, tanto de los objetos como de la propia aplicación, se fijan los siguientes bloques básicos que se deben disponer:

### Conectividad y normalización

Es necesario que incluyan diferentes protocolos y formatos de datos que permitan garantizar una transmisión precisa y la interacción con todos los dispositivos. Es decir, es necesario que garanticen la interoperabilidad entre sistemas y plataformas. Interoperabilidad hace referencia a la capacidad de la plataforma en cuestión para interrelacionar equipos y sistemas de diferentes proveedores o fabricantes para que, entre ellos, a pesar de comunicarse de formas diferentes, la plataforma logre adaptar la comunicación entre ambos extremos. Con el concepto de interoperabilidad, se logra un concepto mencionado al inicio que es de suma importancia en el ambiente tecnológico, el cual es: ECOSISTEMA. La interoperabilidad genera ecosistemas.

### La gestión de dispositivos

Una plataforma debe asegurar que los objetos funcionan correctamente, para ello deben monitorizarse, gestionar sus fallos y actualizarlos en el caso de que sea necesario.

### Almacenamiento de datos

El almacenamiento de los datos procedentes de los dispositivos trae consigo los requisitos de las bases de datos híbridas (relacionales/no relacionales), basadas en una arquitectura en nube, que permitan hacer frente al tipo de datos, su variedad, velocidad y veracidad.

### Procesamiento y gestión de las acciones

La generación de datos en tiempo real, basado en reglas (evento-acción), permite la ejecución de acciones. Estas acciones pueden ser más o menos inteligentes en función de los niveles de inteligencia que se le hayan dado al dispositivo.

### La analítica

Permite llevar a cabo una serie de procesamientos complejos a partir de la agrupación de datos básicos. La posterior incorporación de técnicas de aprendizaje automático, permite el análisis predictivo, lo que al final extraerá el máximo valor a los datos (información/conocimiento).

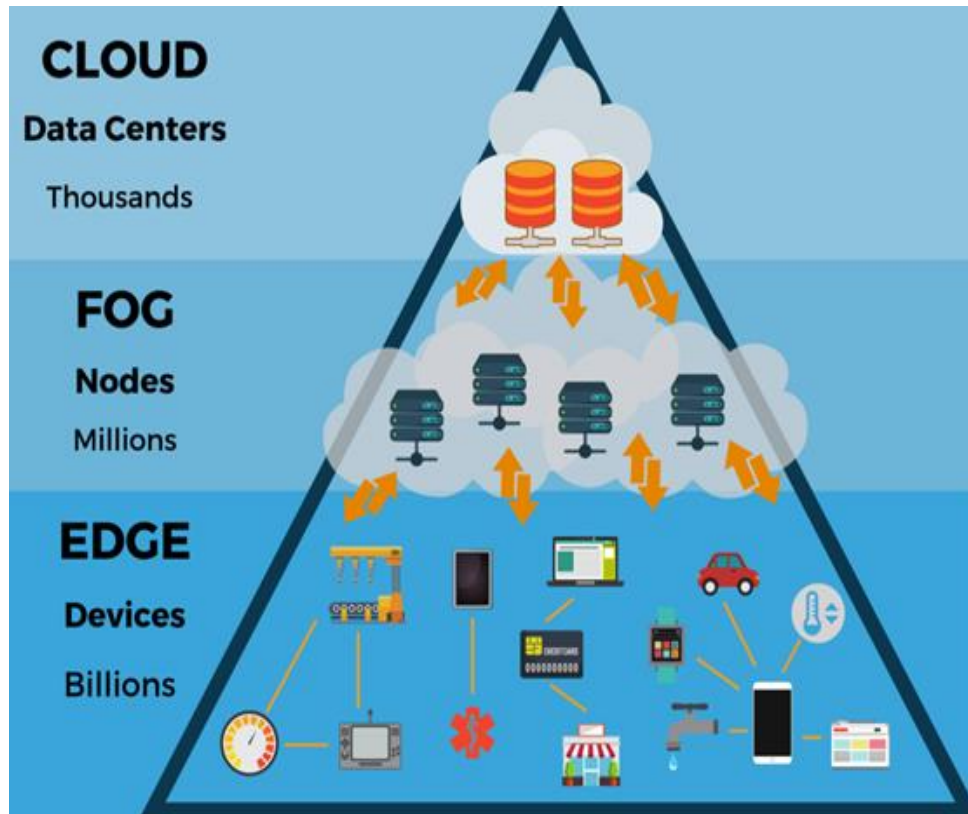


Fig.1: Arquitectura General de IoT.

Fuente: <https://aprendiendoarduino.wordpress.com/tag/fog-computing/>  
Consultado el 1 de Febrero de 2021

En Edge Computing (EC), el procesamiento se distribuye entre los nodos de la red o puertas de enlace, cerca del borde (Edge), o incluso en los propios dispositivos.

En Fog Computing (FC) el procesamiento se lleva a nivel de la arquitectura de red, por lo que también tiene lugar entre nodos Fog distribuidos por la red o Gateways IoT (Puertas de enlace).

El paradigma de Cloud Computing (CC) se amplía hasta el borde de la red. Es aquí donde se realizan todas las acciones relacionadas al almacenamiento, permanencia y analítica de los datos.

### 2.2.5 TECNOLOGÍAS DE COMUNICACIÓN

#### LoRa

Es una tecnología inalámbrica que opera en una banda de frecuencia libre, con buena penetración en paredes, ofrece disponibilidad de hardware, una buena relación costo/calidad además de, la libertad para que cualquiera pueda montar su propia red. Actualmente una de las mayores tecnologías implementadas para soluciones IoT.

#### LoRaWAN

Es un protocolo de red de capa de enlace, que usa la tecnología de radio LoRa para comunicar y administrar dispositivos (gestionando sus canales y parámetros de conexión: canal, ancho de banda, cifrado de datos, etc.). Es un protocolo bidireccional; incluye encriptación de extremo a extremo, permitiendo actualizaciones Over-The-Air de los nodos finales (carga de firmware en el módulo mediante una conexión inalámbrica), incluyendo la capacidad de realizar multicast (envío de la información en múltiples redes a múltiples destinos simultáneamente).

#### GPRS (General Packet Radio Service)

Es una tecnología que subsana las falencias de GSM en cuanto a la transmisión de datos, introduciendo así una red de conmutación de paquetes que funciona de forma paralela a la conmutación de circuitos de GSM. En su momento GPRS apareció como una evolución de la actual red GSM: la cual no llevó a grandes inversiones y se pudo reutilizar parte de las infraestructuras actuales de GSM. Por este motivo, es que GPRS tiene desde sus inicios la misma cobertura que la red GSM. GPRS es considerada la generación 2.5, ya que, en base a sus especificaciones y cambios en la transmisión, fue el paso previo a lo que serían luego las redes 3G.

Características generales de 2G, más específicamente GPRS:

- Velocidad de transferencia de hasta 144 Kbps o hasta 21,4 kbps por TS (Time Slot)
- Red de conmutación de paquetes
- Se aprovecha gran parte de la estructura de red GSM
- Reserva flexible de canales radio
- Adecuado para tráfico a ráfagas
- Sistema FDMA/TDMA con FDD
- Separación de radiocanales 200 kHz
- Bandas de frecuencias: 850 / 900 / 1800 / 1900 MHz
- Separación dúplex: 45 / 45 / 95 / 80 MHz
- Modulación: GMSK a 270,833 kb/s
- Potencia máxima de terminal móvil:
  - 900 MHz: 8 W–0,8 W (típico 2 W)
  - 1800 MHz: 1 W–0,25W (típico 0,25 W).
- Relación de protección: cocanal 9 dB, canal adyacente -9 dB
- Posibilidad de salto en frecuencias (Frequency Hopping, FH)
- Distancia de cobertura máxima: 35 km; puede ampliarse
- Se utilizan nuevos métodos de codificación de canal (Coding Schemes, CS), que se seleccionan dinámicamente, en función de las condiciones del enlace.

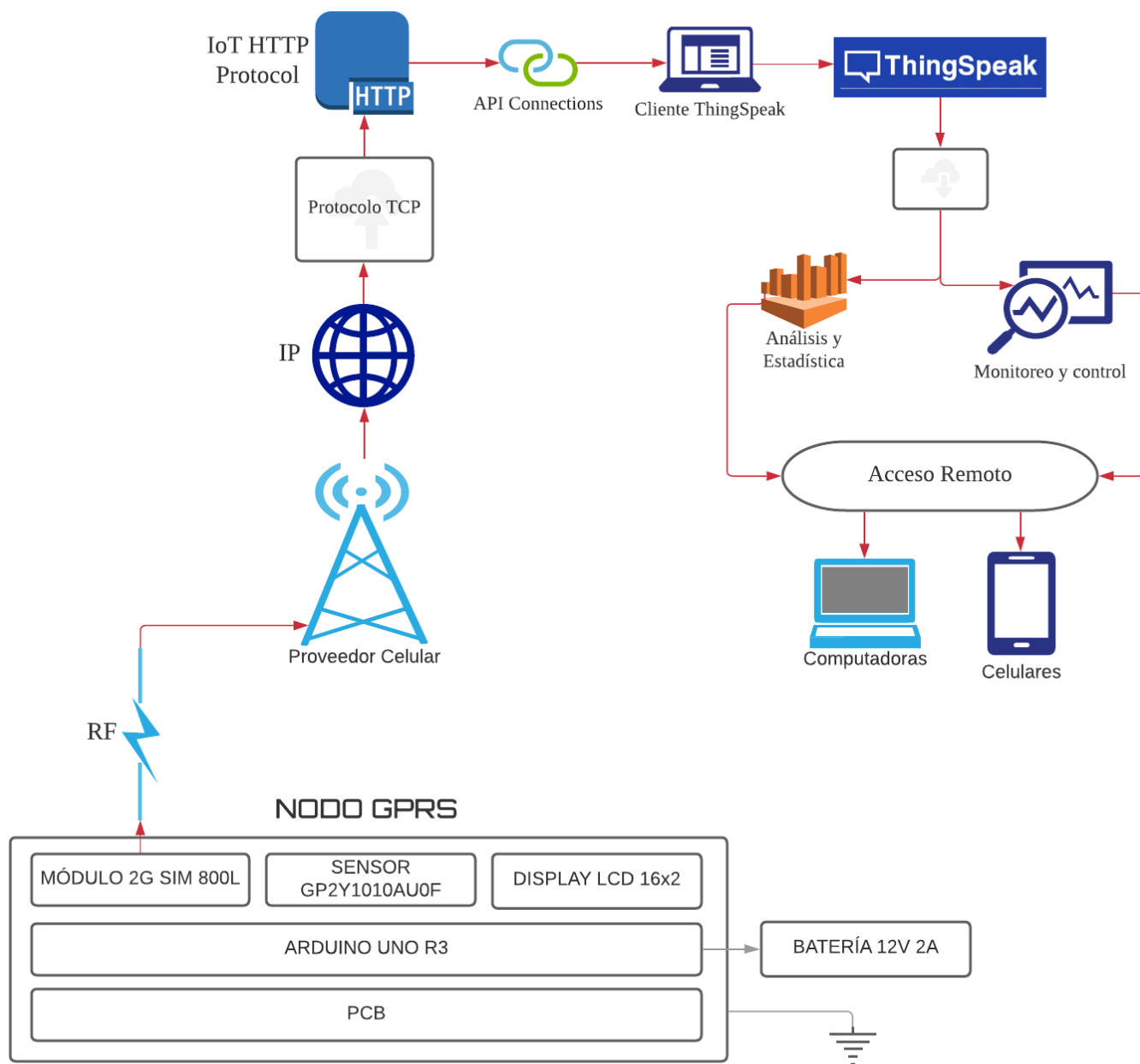


Fig.2: Arquitectura de solución propuesta. Fuente: Diseño del autor

Partiendo de la base de la Figura 2, será como en el presente capítulo se desarrollarán todas aquellas consideraciones y justificaciones técnicas implementadas en todo el proceso de preparación, programación y puesta en marcha de la prueba de concepto en cuestión, pretendiendo así, comenzar desde lo más bajo, como lo es la elección de electrónica de

comunicación e interconexión de la misma, con sus respectivas condiciones de diseño, hasta los niveles superiores relacionados a protocolos y elevación de datos a la plataforma elegida.

Como se mencionó en puntos anteriores, el proyecto se basó en el diseño y la implementación de un sistema capaz de cuantificar la cantidad de partículas finas presentes en puntos determinados. Esta solución permitirá alertar en forma temprana la presencia de polvo, humo de cigarrillo, polen, permitiendo llevar un control debido al constante monitoreo. Al mismo tiempo, una vez que la información se encuentra en la plataforma, esta podrá ser consultada vía web por terceros permitiéndole por ejemplo, a entidades gubernamentales, utilizar esta valiosa información para la toma de decisiones, contribuyendo al cuidado del medio ambiente y concientización de la población respecto a esta temática, entre otros casos de uso.

#### **CONSIDERACIONES DE IMPORTANCIA:**

- Se deja expresamente aclarado que el presente proyecto pretende ser una prueba de concepto en calidad de prototipo de monitoreo y control de calidad de aire, sin fines de lucro. En su diseño, se hace uso de licencias no aptas para fines comerciales como lo son la implementación de placas Arduino y el uso de la licencia gratuita de ThingSpeak, las cuales ambas tienen por objeto su utilización para fines hobbistas, de educación y/o desarrollos no remunerativos.
- En caso de fabricación en serie para su futuro lucro, se deberán tomar primeramente las siguientes medidas:
  - Adquisición y/o cambio dependiendo el caso, por licencias y módulos aptos para comercialización por parte de las marcas en cuestión, y en su defecto, de alternativas compatibles.
  - Diseño y desarrollo de un PCB estándar, que cumpla con las condiciones óptimas de funcionamiento y requerimientos.

Inicialmente al momento de seleccionar la tecnología y electrónica para el futuro diseño de la prueba de concepto, se pensó en la utilización de tecnología LoRa basada en Arduino, implementando módulos tales como Arduino Mkr Wan 1300 (el cual cuenta con compatibilidad con LoRa), SX1278 y/o HM-TRLR-433S. Con la inclusión de los anteriores, el ancho de banda y capacidad de transmisión necesario para los fines de este proyecto alcanzaría perfectamente a pesar de ser estos últimos relativamente bajos. En referencia al alcance del enlace donde el máximo teórico ronda los 5 km según la hoja de datos del HM-TRLR-433S, sería más que suficiente siendo que el Gateway a desarrollar estaría a no más de 2km del End Point, además de ser LoRa uno de las tecnologías con las más altas sensibilidades de recepción pudiendo así, llegar hasta -168dB.

Lo que sentenció la elección para otra dirección no fue solamente el tema de disponibilidad y acceso a esta tecnología en los mercados nacionales, que, para la fecha en la que se hizo el análisis (aproximadamente a mediados del 2020) los distribuidores y sucursales de electrónica en el país no contaban con stock y variedad, sino que hasta incluso el costo de los mismos era demasiado elevado para la fecha. Sin contar que, una vez habiendo profundizado en el análisis, se terminó por concluir que la idea presentada demandaba demasiado trabajo de adaptación con la posibilidad de, finalmente no ser compatibles los módulos y/o shields.

Es aquí cuando se inicia nuevamente la búsqueda de alternativas, teniendo ahora un sondeo previo de la disponibilidad tecnológica existente en el país, que termina con la inclusión de los módulos en cuestión, siendo las 2 cuestiones previas a resolver:

- ¿Arduino UNO R3 o Arduino MEGA 2560? La diferencia a grandes rasgos que existen entre estos dos módulos radica principalmente en el microcontrolador que cada placa utiliza (ATmega 328P y ATmega 2560 correspondientemente) y luego, mayor cantidad de memoria y puertos por parte del 2560, que, a fin de cuentas, para los fines de este proyecto, serían innecesarios. Observando y analizando los módulos a conectar se estimó que la placa que mejor se adaptaba y cumplía a los requisitos era el Arduino

UNO. En lo que efectivamente, luego de su adquisición y la conexión correspondiente, la placa siguió contando con I/O analógicas y digitales de sobra, pero en lo que sí se estuvo casi al límite fue en cuestiones de SRAM, lo que más tarde se solucionó aplicando optimización en código, logrando así liberar en gran medida esta memoria. Por lo que la decisión final y correcta estuvo por el lado del Arduino UNO.

- ¿Módulo de comunicación 4G o 2G? Ésta fue otra de las grandes interrogantes planteadas al momento de adquirir electrónica de comunicación, que si bien para el momento del análisis se desconocía exactamente cuál era la carga a transportar por la red cuando se realizara la elevación de los datos, se tenía la certeza de que era muy baja. Entonces, un módulo 4G sería innecesario por cuestiones de capacidad de transmisión, aunque no está para nada mal contar con altas velocidades de navegación, pero si se analiza desde la perspectiva del alcance de una red y la otra, 2G prevalece mucho más que la anterior, pudiendo la misma tener acceso por ejemplo en zonas de alejadas de la ciudad donde dependiendo el caso de uso, podría ser necesario y 4G no estar a la altura de la situación siendo que la misma está pensada como una red urbana de corto a medio alcance (5-10km máximo de la antena más cercana). Por último y no menos importante, la diferencia de costos existente entre módulos es terriblemente alta, donde un módulo LTE con respecto a uno GPRS puede llegar a valer 10 veces más, por lo que estuvo más que claro después de analizar específicamente el caso, que 2G era la tecnología correcta para este proyecto.

### 3.1 INTERCONEXIÓN DE HARDWARE

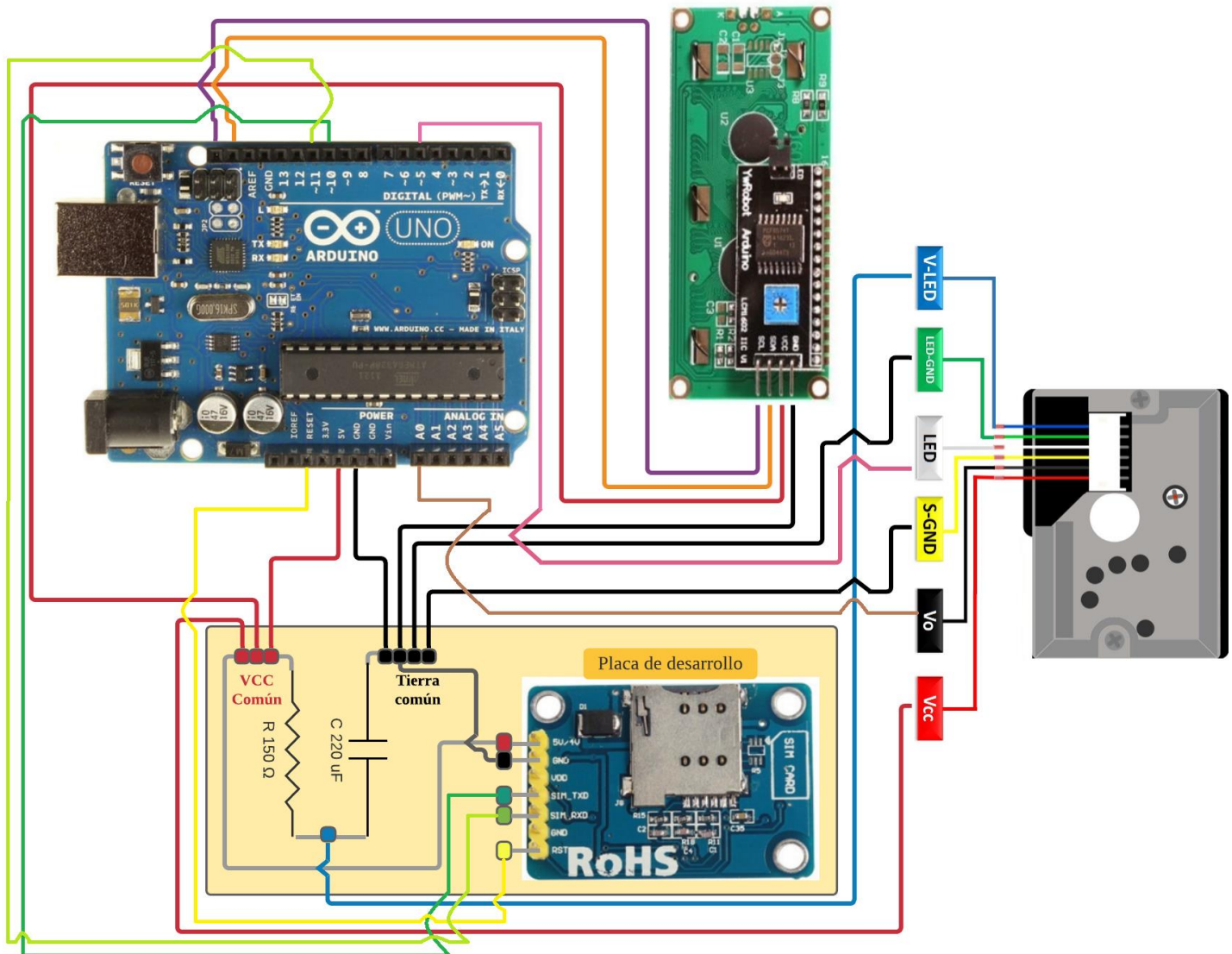


Fig.3: Esquema de Interconexión entre módulos

La interconexión del hardware de todo el prototipo fue llevada a cabo según el esquema presentado en la Figura 9. Debido a que la cantidad de conexiones intermedias entre la Placa de Desarrollo con el resto de dispositivos no son demasiadas, se decidió realizar el diseño de un PCB casero utilizando una Placa de Desarrollo agujereada, pines hembra y macho y por último como se mencionó en los datos técnicos del sensor, la resistencia de  $150\Omega$  y el capacitor de  $220\mu F$ .

Las conexiones que se observan en la placa de desarrollo de **color GRIS** representan conexiones locales de la placa que se encuentran soldadas entre sí, para evitar el uso de cables

innecesarios entre módulos y los pines. El módulo celular, NO se encuentra directamente soldado al PCB, sino que, si bien en el gráfico no se muestra, pero para su modularidad se soldó una tira de 7 pines hembra sobre los cuales se encastra el mismo, y luego, se puentea cada pata del módulo directamente con su conector hembra para distribuir la conexión por cable hacia fuera de la placa. El pin VDD y GND de TTL (el pin GND sobrante del módulo) no cuentan con su conexión puenteada debido a que los mismos para este propósito no se utilizan.

Para alimentar la solución en su totalidad se utilizó una batería de 12V-2A la cual se conecta a través del Arduino, por su entrada Jack de 5.5mm.



Fig. 4: Batería/Fuente PROBATTERY 12V 2A

### 3.2 PARAMETRIZACIÓN DE LA SOLUCIÓN

| Dispositivo                  | Consumo <sup>i</sup>                                               | Alimentación                         |
|------------------------------|--------------------------------------------------------------------|--------------------------------------|
| ARDUINO UNO R3               | 50 mA (en reposo)                                                  | 7-12V (recomendada)<br>Operación: 5V |
| SIM800 L                     | 0.7 mA (Sleep mode – Full Functionallity) – 2 A (Modo Ráfaga GPRS) | 3.4 V – 5V                           |
| Display I <sup>2</sup> C LCD | 25 mA (máximo con linterna encendida)                              | 2.5 – 5 V                            |
| GP2Y1010AU0F                 | 11 mA (promedio), 20 mA (máx.)                                     | 5 V                                  |

<sup>i</sup> Consumo de corriente estimado por el fabricante en su hoja de datos.

Tabla.2: Consumo de corriente por dispositivo

Para este proyecto no existe dificultad al momento de diagramar la forma de alimentar la solución ya que la totalidad de dispositivos funcionan con 5V, los cuales, si bien la alimentación utilizada es de 12V, el Arduino cuenta con su respectivo regulador de voltaje a la entrada para que dicha alimentación esté asegurada en todos los dispositivos, además de los propios reguladores que cada dispositivo trae consigo internamente. En la Tabla 6 se observan con mayor detalle los valores de consumo de corriente para cada dispositivo respetando lo expresado por cada fabricante en la hoja de datos.

Con este apartado concluye la justificación y desarrollo de todo lo relacionado a los aspectos electrónicos y tecnológicos de la prueba de concepto, que según la Figura 2, representa el primer escalón de la solución implementada.

### 3.3 DISEÑO DE SOFTWARE

En el presente apartado será donde se desarrollarán todos los eslabones o capas restantes del proyecto, siguiendo con la guía base propuesta en la Figura 2, donde dicha implementación se realiza mediante software, siendo el SIM 800 L el encargado principal y clave de llevar a cabo todas estas rutinas y ejecuciones. Partiendo desde el registro y acceso a la red celular del operador, habilitar servicios de consumo de datos, sincronizar hora y fecha mediante un servidor NTP, hasta la apertura y configuración de las conexiones HTTP a la API de ThingSpeak y la elevación de los datos son las grandes temáticas y cuestiones que el software resuelve a su vez de permitir la reiteración del ciclo de elevación sin problemas cada vez que sea necesario, y en caso de fallas o errores, notificarlos y resolverlos en lo posible.

Para la programación se utilizó el IDE de Arduino ya que el mismo durante la carrera ha sido uno de los más utilizados y de los que más conocimientos se tiene utilizándolo y programando en #C, además de las grandes facilidades que este presenta como por ejemplo el grandísimo repositorio de librerías y código existente para la gran mayoría de los módulos que existen hoy en el mercado, llevándolo más puntualmente al presente caso, todos los módulos utilizados cuentan con sus respectivas librerías o código, además de la compatibilidad con el IDE y los demás dispositivos. Llevado a otras cuestiones, cuando se presentan errores o fallas desconocidas, al ser uno de los IDEs más utilizados por la mayoría de estudiantes, electrónicos y hobbistas, resulta fácil obtener soporte desde Internet donde se encuentra muchísimo contenido disponible, foros y páginas de creadores de contenido.

Para la manipulación y utilización del sensor, se reutilizó y readaptó código desarrollado por Viktor Lytsus, de su repositorio oficial de GitHub, donde el mismo desarrolla a la perfección todos los métodos de corrección y obtención de muestras por parte del sensor, utilizando de referencia la hoja de datos provista por el fabricante y teniendo en consideración todos aquellos factores que podrían afectar en la medición.

A continuación, se presentan las máquinas de estado de funcionamiento de toda la solución en general, y por último el diagrama de flujo de la función más relevante del software que debe ser detallada:

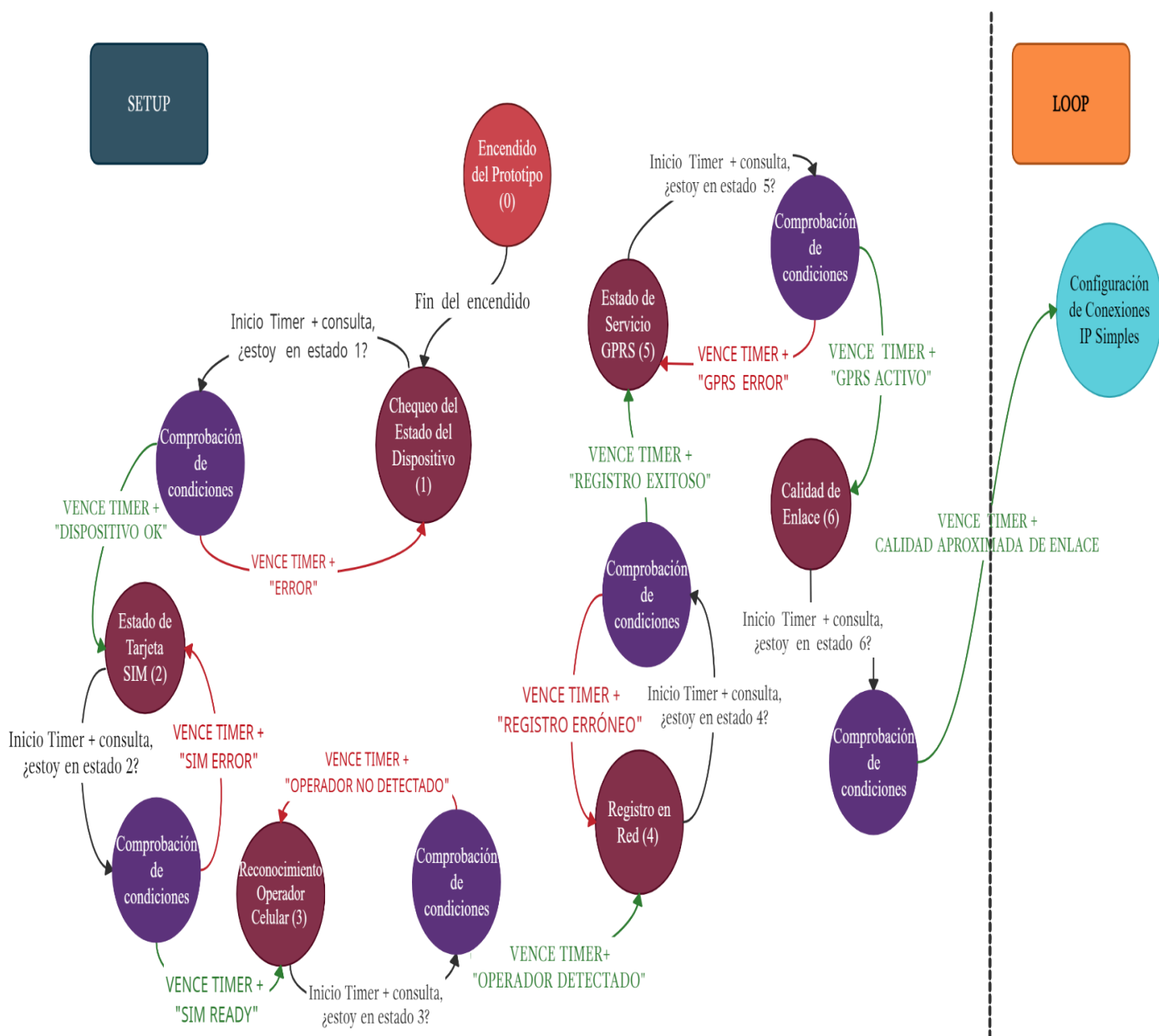


Fig.5: Primera Máquina de Estado del sistema de la solución

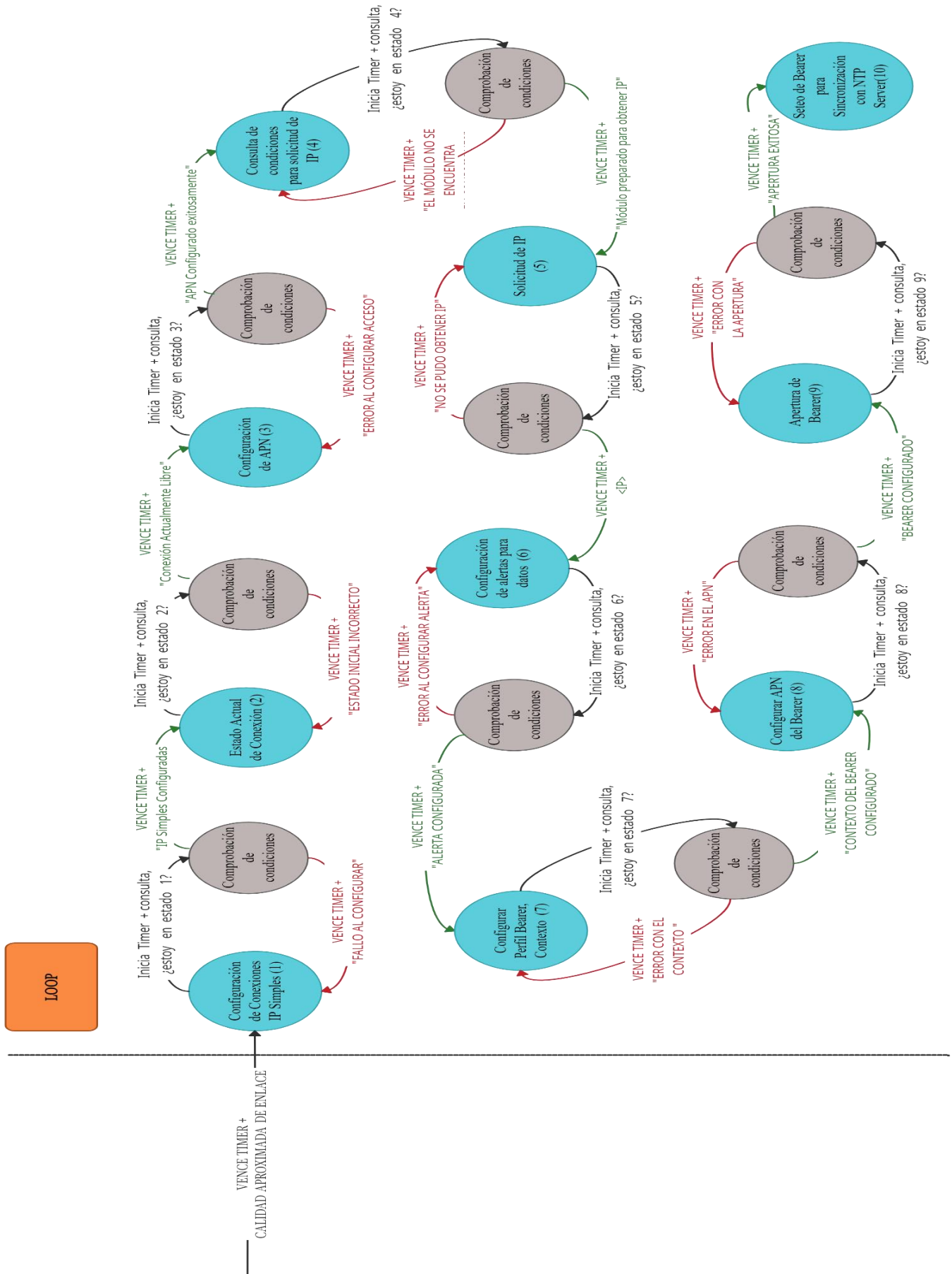


Fig.5.1: Segunda Máquina de Estado del sistema de la solución Primera Parte

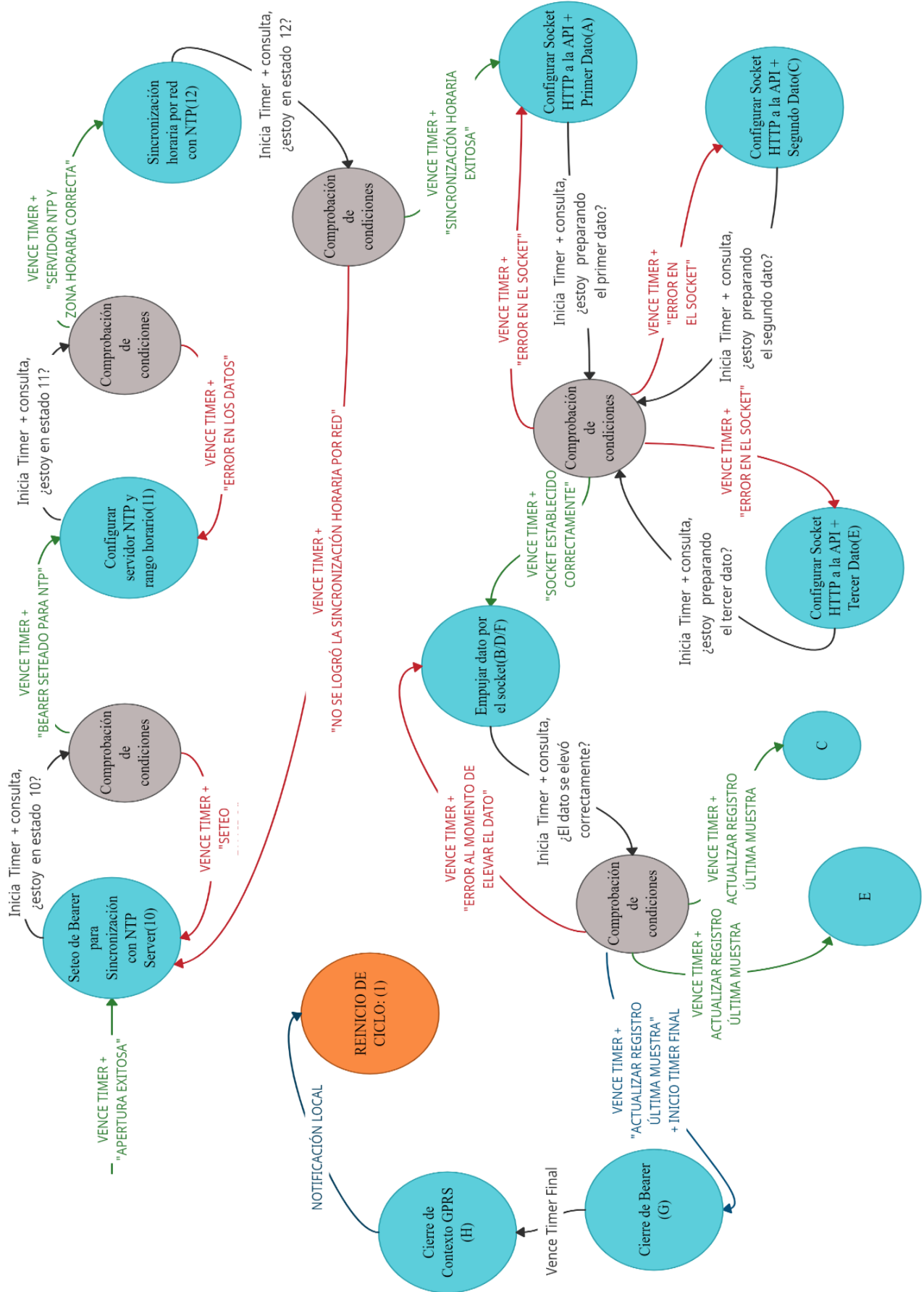


Fig.5.2: Segunda Máquina de Estado del sistema de la solución Primera Parte

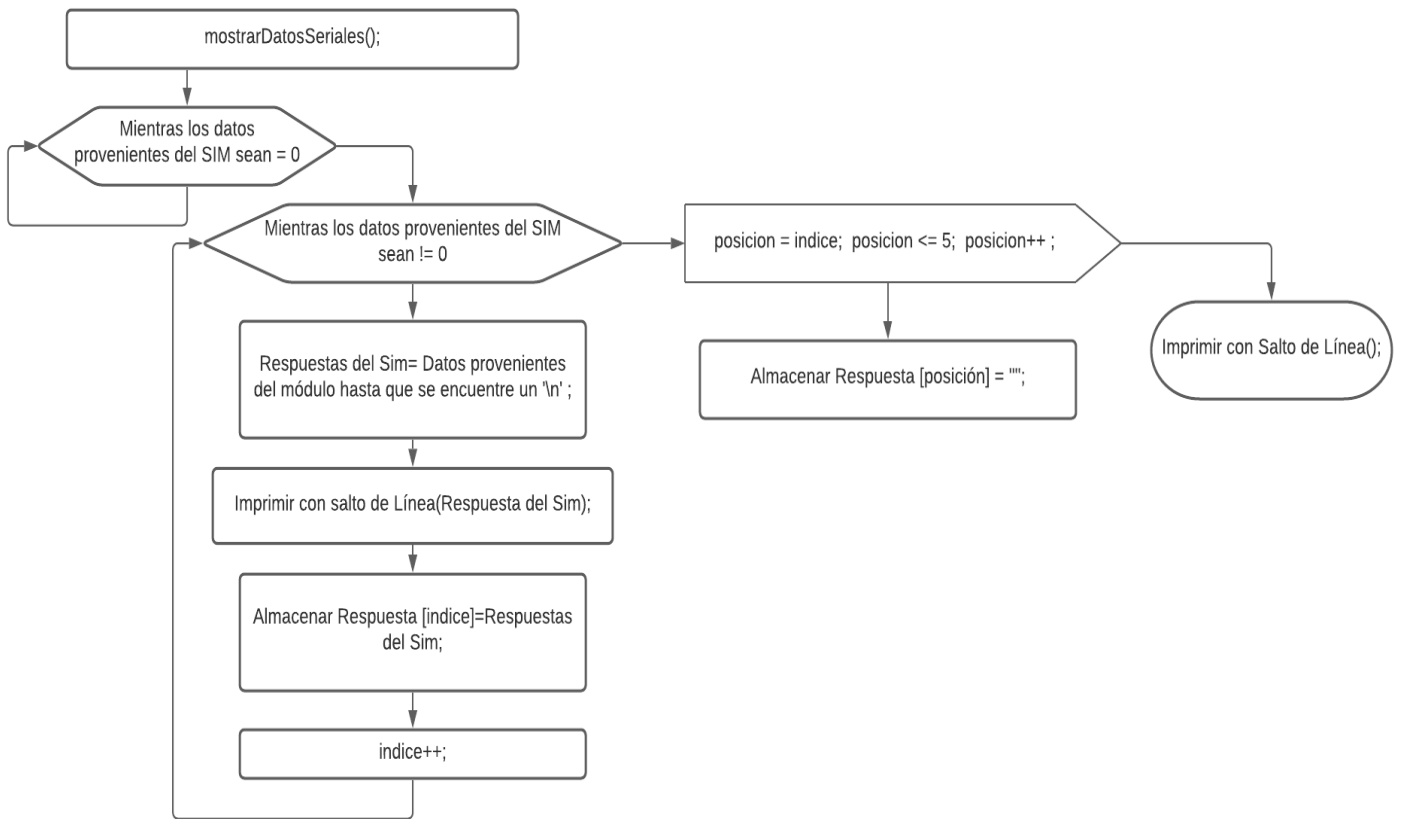


Fig.6: Diagrama de flujo de función base de la solución

Como bien se mencionó anteriormente, el sistema de la solución propuesto para esta prueba de concepto, consiste en, fundamentalmente dos máquinas de estado finitas cíclicas, asistidas por una función de captura y comparación de parámetros (estado intermedio llamado “Comprobación de Condiciones” en las Figuras 5, 5.1 y 5.2) en base a justamente las respuestas que el módulo da cuando se le ejecuta una configuración o comando, en conjunto con los respectivos timers que cada ejecución conlleva para poder dar tiempo al módulo a procesar y responder, así como informar el estado de lo sucedido por display.

Cuando se dice que el módulo responde en base a comandos y configuraciones, es porque el mismo utiliza comandos AT o comandos Hayes. Los comandos AT son instrucciones codificadas que conforman un lenguaje de comunicación entre el hombre y un terminal MÓDEM. Los mismos fueron desarrollados en 1977 por Dennis Hayes como una interfaz de comunicación con un MÓDEM para así poder configurarlo y proporcionarle instrucciones de todo tipo. Tiempo después, con el avance del baudio, fueron las compañías Microcomm y US

Robotics las que siguieron desarrollando y expandiendo su desarrollo hasta lograr que el conjunto de comandos Hayes, se convirtiese en un estándar abierto, vigente hasta la actualidad. Los caracteres «AT», que preceden a todos los comandos, provienen del significado de «Atención», los cuales hicieron que en su conjunto se terminase por definir como actualmente se los conoce, comandos AT.

Un dispositivo que implemente el conjunto de comandos Hayes se considera compatible Hayes. Parte del conjunto de comandos Hayes fue incluido por la ITU-T en el protocolo V.25ter (1995), actualmente V.250(2003 última reforma, actualmente vigente). Aunque la finalidad principal de los comandos AT es la comunicación con módems, la telefonía móvil GSM/GPRS también ha adoptado como estándar este lenguaje para poder comunicarse con sus terminales.

Para comprender más a fondo cómo funcionan este conjunto de instrucciones, se extrae desde el blog de Rafael Nova (septiembre 2013) el siguiente apartado:

#### “Formato de los comandos Hayes

Como se dijo anteriormente la mayoría de los comandos Hayes empieza con la secuencia “AT”, siendo las excepciones el comando “A/” que repite el último comando introducido y la secuencia triple del carácter de escape. Los otros comandos luego de la secuencia “AT” siguen con las letras del alfabeto. Además, muchos de ellos necesitan a continuación un valor numérico, que en el caso que no se escriba se tomará como que dicho valor es cero.

A medida que se fueron requiriendo más funciones en los módems se tuvo la necesidad de agregar más comandos. A estos comandos se los llama extendidos y tienen la forma AT&X donde “&” indica que el comando X es extendido. Así mismo cada fabricante introdujo otros que no fueron estándares y cumplían funciones específicas. No todos los módems responden a estos comandos. En resumen, a los comandos Hayes se los puede dividir en 4 grupos:

- 1- Básicos (AT...): estos comandos fueron los que inicialmente fueron definidos y cumplen funciones elementales.
- 2- De registro (ATSi=, ó ATSi?): modifican los valores de los registros internos del módem o solicitan sus valores.
- 3- Extendidos (AT&.): comandos adicionales que se agregaron posteriormente a las definiciones de los comandos básicos. Generalmente cumplen funciones poco más complejas que los básicos.
- 4- Propietarios (AT/...): definidos por el fabricante del equipo.

Ante cualquier comando enviado al módem, este responde con el resultado de la operación: “OK”, “ERROR”, CONNECT, etc.”

Una vez aclarado lo anterior, entendiendo como funciona el SIM 800 L, es necesario primeramente poder capturar las respuestas dadas por el módulo ante un determinado seteo o configuración para luego, una vez capturada dicha respuesta, utilizarla como parámetro de comprobación, ya que leyendo la hoja de datos del fabricante es posible saber qué es lo que el módulo debe devolver si un comando se ha ejecutado correctamente ya sea por sintaxis y/o aplicación correcta del mismo. Es de esta forma en la cual las máquinas de estado van procediendo en la ejecución, y en el caso de fallas, volver a ejecutar el comando o recomenzar la configuración desde donde sea necesario, logrando así setear y adaptar el terminal para que el mismo luego sea capaz de registrarse, acceder y elevar datos por la red.

Todo este algoritmo de funcionamiento, es lo que se realiza en cada estado intermedio de las máquinas entre la ejecución de un comando y el siguiente, llevado a cabo por la función descrita en la Figura 6, y una vez obtenida la respuesta o error, informar la situación por display haciendo uso de su respectiva librería y una serie de prints sobre cada línea propia del módulo. La función graficada, básicamente consta de dos salidas dependiendo de si existen o no datos que reportar por parte del SIM: mientras no existan datos en el terminal que informar, la función es totalmente bloqueante y permanecerá en ese loop hasta que si los haya, es entonces cuando la función salta a la otra salida donde se van mostrando línea por línea

hasta un total de 5 (número determinado por la máxima cantidad de líneas requeridas para leer la respuesta más extensa según lo referido a dicho comando en la hoja de datos y la propia experimentación con el módulo). Luego de esto, se borra el búfer de la función para evitar sobreescritura y guardado de datos no correspondientes entre ellos (debido a la cantidad de líneas de respuesta de diferencia entre un comando y otro, más basura que el módulo propiamente puede devolver) para nuevamente terminado esto volver a esperar una próxima entrada de datos.

Siendo que el terminal celular utiliza la convención de sistemas Unix y basados en Unix, cada fin de línea utiliza “\n” como carácter terminador, siendo justamente este parámetro el utilizado para identificar cada una de las respuestas por línea.

Al momento de programar las rutinas que finalmente terminaron por ser las máquinas de estado finitas, inicialmente dicho código era lo que se conocía como “código espagueti”, es decir, líneas y líneas de código sin estructura alguna y con una enorme cantidad de ciclos anidados el cual entorpecían enormemente la interpretación y seguimiento del mismo. Si bien, inicialmente lo que se quería comprobar era cómo el módulo respondía a estas rutinas y si el fin principal se cumplía, es decir, la correcta y ordenada elevación de los datos a la plataforma, el mismo a su vez hacía uso de funciones de código bloqueante como lo era la utilización constante de la función “delay();” de Arduino debido a que se requería algunos segundos de espera hasta que el SIM devolvía las respuestas a los diferentes comandos, cuestión totalmente necesaria ya que al parecer el procesador del módulo no cuenta con la suficiente capacidad para realizar esta tarea más rápidamente.

Por lo que, luego de comprobar que los datos llegaban correctamente a la web, lo siguiente fue estructurar el código, separándolo por funciones en base a la necesidad e importancia de cada comando (si su ejecución era necesaria una o más veces) y por sobre todo, evitar a toda costa el bloqueo del procesador del Arduino quitando, la cantidad de delays utilizados durante toda la ejecución del software desarrollado. Para lograr lo anterior, se reemplazó cada delay por un temporizador particular por cada comando o espera a ejecutar

haciendo uso de otra función de Arduino encargada de temporizar, la cual es “millis()”, que a diferencia del delay el cual bloquea el procesador por la cantidad de tiempo indicado, esta lleva el conteo de la cantidad de milisegundos pasados desde que el Arduino se encuentra encendido. En conjunto con la anterior, definiendo una constante (valor de segundos que se quiere esperar) y una variable entera de 16 bits (ya que dependiendo del momento en el que se quiera iniciar la temporización, el valor de millis puede ser lo suficientemente grande superando así los 8 bits de tamaño), el mecanismo del temporizador sería el siguiente:

```
#define temporizadorA 3000;           //CONSTANTE QUE LLEVA LA REFERENCIA
                                       //DE SEGUNDOS QUE SE QUIEREN
                                       //ESPERAR

uint_16t tiempoA;                     //VARIABLE DE REFERENCIA REQUERIDA
                                       // PARA COMPARAR

millis()=tiempoA;                     //INICIO LA TEMPORIZACIÓN, ASIGNANDO
                                       //A LA VARIABLE EL VALOR DE MILLIS() EN
                                       //ESE INSTANTE

if ((millis()- tiempoA) > temporizadorA){

// SECUENCIA DE ACCIONES A REALIZAR

}
```

Entonces, una vez realizada la asignación, y la diferencia entre millis y el valor asignado sea mayor a la constante, significa que ha transcurrido el tiempo que se quería esperar y sin bloquear al procesador en su ejecución, para luego por ejemplo mostrar los datos que el SIM tenga en su búfer y así comparar las respuestas obtenidas, siendo que si se tiene el parámetro querido, es posible continuar con la rutina reportando lo sucedido por display y pasando al siguiente comando, y de no serlo, reportarlo y actuar en consecuencia.

Con respecto a las máquinas de estado, todas las acciones realizadas en la primera (Figura 5) son aquellas consideradas de única ejecución, es por ello que aparecen en el SETUP, como lo son consultar el estado del módulo, de la tarjeta SIM y las relacionadas a la detección y registro en la red del operador, así como habilitar el servicio de consumo de datos por GPRS. Una vez que todas estas condiciones se han cumplido, se puede decir que el módulo se encuentra en estado de comenzar a realizar conexiones por la web, pasando así a la segunda máquina.

A diferencia de la primera, esta máquina es la encargada de configurar primeramente el modo de conexión, las cuales son conexiones IP simples, es decir, un socket a la vez y no todos los datos en simultáneo, setear el acceso a la red del proveedor con su respectivo APN, el proceso de solicitud de IP hasta su obtención y luego, pero no menos importante, la sincronización horaria por red estableciendo una conexión a un servidor NTP. Para configurar dicha conexión se requiere antes el establecimiento de un perfil de Bearer para uso exclusivo de este tráfico, en el cual es necesario establecer ciertos parámetros previamente como contexto de la conexión (GPRS), APN a utilizar, apertura del mismo, seteo que indica que el tráfico que utilizará dicho bearer corresponde a conexiones de servicios NTP, URL del servidor al que se consulta con el correspondiente huso horario y por último la orden de comienzo de sincronización al servidor. Es muy importante que la sincronización se realice de manera exitosa ya que con ella luego se deja constancia de la fecha y hora de la última elevación de datos realizada, es por ello que si al iniciar la comunicación al servidor la misma falla, se ha de reiniciar todo el proceso hasta lograr su correcto funcionamiento.

Siguiendo la Figura 5.2 se llega al ciclo de subida de los 3 datos que se elevan a la plataforma, donde, en resumen y suponiendo que:

- No existen problemas de conexión tanto con el proveedor ni con los servidores de la API.
- Entre cada acción ejecutada, existe una mínima cantidad de segundos de espera y todo el proceso es seguido en tiempo real, actualizándose en el display.

La secuencia realizada por el módulo con cada uno de los datos es la siguiente:

- 1- Iniciar obtención de muestras por parte del sensor.
- 2- Configuración del Socket: Indicando tipo de transporte a utilizar (TCP) + IP y/o URL de Destino (API de ThingSpeak) + Puerto de Destino (80).
- 3- Habilitar Socket y espacio de envío.
- 4- Solicitud HTTP GET + API Link-API Key & Ubicación de parada del dato en la plataforma + Dato a elevar (Variable).
- 5- Obtención del dato a elevar por parte del módulo.
- 6- Se ordena el “push” o el “Enter (<cr>)” del dato por el Socket.
- 7- Recibimiento del mensaje “SEND OK” debido al “Echo Mode” activado durante las rutinas anteriores + Fecha-Hora del reciente reporte transmitido.
- 8- Espera de 30 segundos para próximo dato.
- 9- **UNA VEZ LLEGADO AL TERCER DATO: Cierre de Bearer**
- 10- **Espera de 2 minutos 30 segundos.**
- 11- **Cierre de todo el contexto GPRS.**
- 12- **Reinicio de Ciclo.**

Una aclaración que se debe hacer, que si bien se encuentra detallado en la Tabla 6 es el hecho de que al momento de elevar un dato a la plataforma existe una limitación impuesta por esta misma con el tiempo mínimo de espera entre cada dato elevado, el cual es de 15 segundos aproximadamente, pero para evitar cualquier problema relacionado con este margen de tiempo y que la petición HTTP y por ende el socket TCP fallase, se dobló el tiempo de espera entre cada dato. Como se dijo en puntos anteriores de la sección, la manera en que se transmiten los datos es a través de conexiones simples y no múltiples en simultáneo debido a justamente, otra de las limitaciones impuestas por la licencia de usuario gratuito de ThingSpeak, aunque perfectamente el módulo admite hasta 5 sockets en simultáneo. La última de las limitaciones impuestas viene dada por la cantidad máxima de datos de esta licencia, la cual ronda actualmente en 3 millones por año, que si bien, para la realización de esta prueba

de concepto, y testing, dicho valor resulta excelente, pero debe ser tenido en cuenta si es que la solución se encuentra funcionando por tiempo ininterrumpido durante largos períodos.

Lo que no se ha justificado hasta aún es la elección de los protocolos utilizados durante todo el proceso de upload de los datos tanto para transporte como el de aplicación o mensajería. En relación al transporte se ha preferido TCP debido a cuestiones tales como:

- TCP es un protocolo orientado a conexión. Un protocolo orientado a la conexión requiere que los puntos finales de la red establezcan un canal entre ellos antes de transmitir los mensajes.
- Fiabilidad del protocolo: TCP garantiza la llegada de los datos a destino y de una forma ordenada.
- TCP cuenta con retransmisión en caso de pérdida del datagrama o la falta de confirmación.

En una aplicación de IoT, donde cada dato/señal/actuación a transmitir tiene un valor excepcional dependiendo del caso de uso lo que podría representar una alerta de incendio, una alarma de un sensor que se dispara, una acción a tomar que puede tener grandes costos y generar daños si el mismo no llega a destino hace que para esta tecnología sea siempre uno de los protocolos preferidos para el transporte de paquetes a través de la web.

Por último, como protocolo de Aplicación/Mensajería realmente la investigación de alternativas fue muy escasa, además que por parte del autor se desconocía realmente que el módulo en conjunto con Arduino podía implementar por ejemplo MQTT, que hubiera sido una muy buena alternativa y una gran oportunidad de aprendizaje debido a que dicho protocolo en las grandes soluciones comerciales de IoT que se encuentran en el mercado hoy en día (independientemente de la tecnología pero en su mayoría LoRa/SigFox) lo utilizan por sobre el seleccionado. Es de ahí por qué básicamente se implementó HTTP por ser el protocolo convencional de comunicación web de estructuras cliente-servidor, asimismo también su estudio durante la carrera en las materias relacionadas a redes genera cercanía y confianza al

momento de utilizarlo debido al conocimiento que se tiene al respecto sobre este, teniendo en cuenta a su vez que su implementación resultó sencilla ya que el dispositivo celular cuenta dentro de sus posibilidades con comandos específicos para su rápida utilización.

### 3.4 PROTOTIPO

El diseño, modificación e injertos colocados para la confección del prototipo fueron realizados en su totalidad por el autor. Para su realización se utilizó un gabinete de tipo estanco “Welt WT 04-407 IP 56” de aproximadamente 19 cm x 14 cm x 7 cm (Largo x Ancho x Profundidad) los cuales normalmente son utilizados en instalaciones eléctricas para paso derivación, adaptándose perfectamente a las necesidades y hasta incluso contar con espacio de sobra en caso de mayores cambios o alguna necesidad de readaptación.

Las reglas de diseño implicadas fueron:

- La principal regla de diseño estuvo en la ubicación y orientación de la antena del módulo celular, prefiriéndose colocarla fuera del gabinete, lo más cerca posible del borde superior del lateral, para evitar que la misma al momento de colocar la tapa del gabinete no sea opacada totalmente por esta, ya que, el gabinete es de plástico PVC, un material totalmente polar, que en su naturaleza ya posee una constante dieléctrica 4 veces superior a la del aire, lo que genera un enlentecimiento de la onda (factor de velocidad), provocando pérdidas por absorción propias del material en conjunto con las respectivas de la frecuencia de trabajo de la antena más todos aquellos obstáculos presentes en el camino. Es por ello, que dejar la antena dentro del gabinete representaría un gran problema tanto al momento de la recepción como de transmisión, sumado a que si estuviese dentro, la misma se encontraría inmóvil, generando aún mucho más problemas si es que el prototipo debe colocarse en alguna posición diferente a la normal, siendo que estando fuera la misma puede ser rotada con total libertad, evitando así, el paralelismo con el suelo lo que prácticamente arruinaría la comunicación con la estación base del proveedor.

- Debido a que el prototipo en sí mismo no cuenta con algún mecanismo que provoque un flujo de aire forzado(ventilador) a través del sensor, se ha colocado el mismo sobre una de las paredes laterales con su respectivo agujero en contacto directo con el ambiente, adosando un codo de plástico a la salida del mismo, de mayor diámetro que el de recepción propio del sensor, para así, por otro extremo del gabinete permitir una entrada/salida mayor de aire. Con el objetivo de preservar al máximo la vida útil del mismo se lo ha dejado también, dentro del gabinete, para que los dos grandes factores que pueden perjudicar tanto el funcionamiento como la integridad de las mediciones, como lo son todo haz de luz externo al del fotodiodo y rocío/humedad queden fuera de alcance.
- Todos los componentes colocados dentro del estanco, son totalmente modulares ya que ninguno de ellos se encuentra soldado o encastrado de forma total en el gabinete, sino que, aprovechando las rendijas y guías que este provee, se utilizan tornillos, tuercas y arandelas para darle flexibilidad al momento del quite de alguno de alguno de estos por el motivo que sea, pero por sobre todo, para recambio de alguno de ellos en caso de fallas o defectos.
- Aunque no existen grandes cantidades de cables utilizados para la conexión de la solución en su totalidad, se adosaron en partes estratégicas del gabinete, grampas de pared pretendiendo servir de cable canal o derivadores de conexión, para mayor prolijidad y orden del sistema en general.

A continuación, se incluyen fotos en la mayoría de sus vistas apreciando los detalles anteriores:

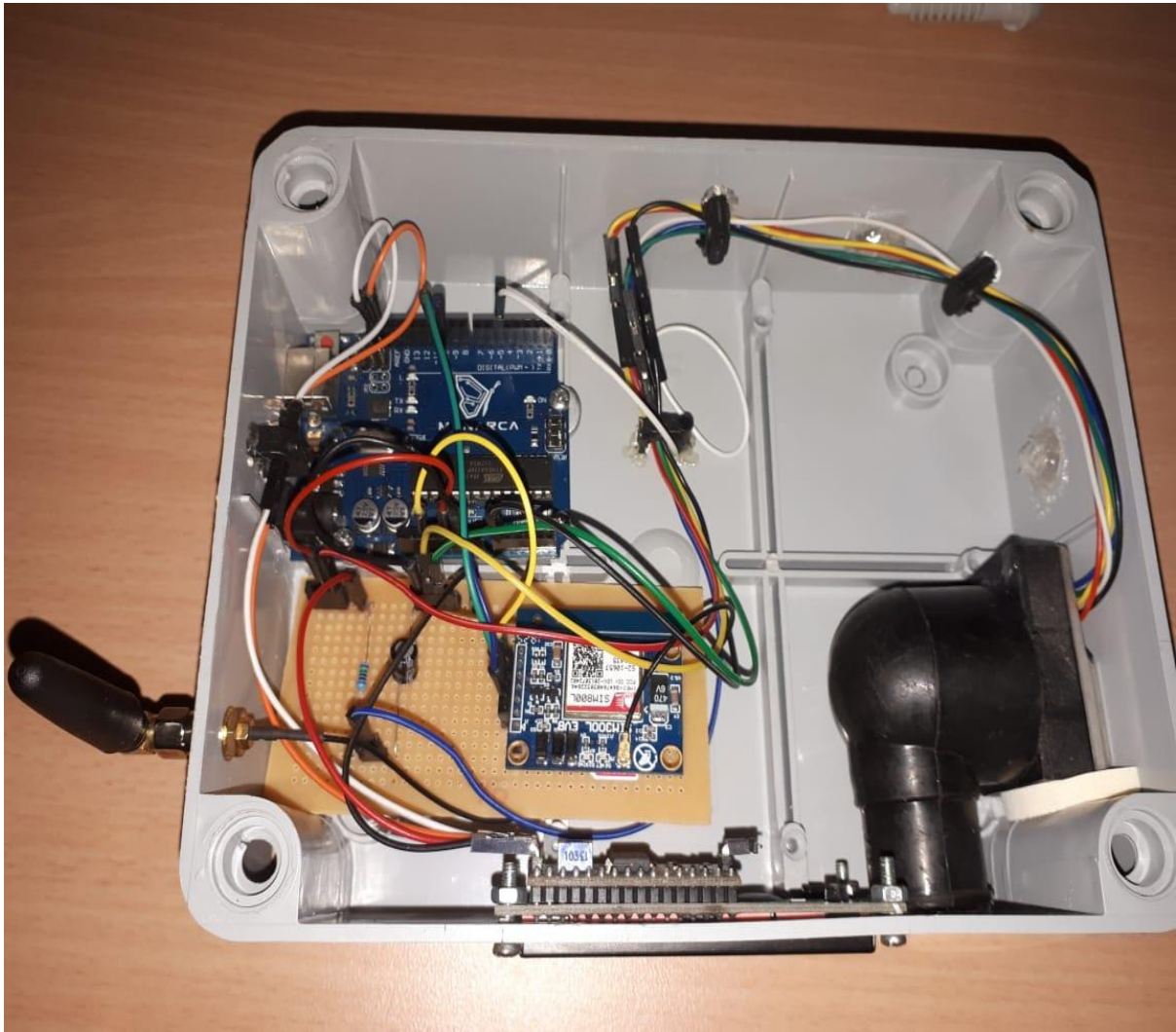


Fig.7: Interior del prototipo con vista superior delantera

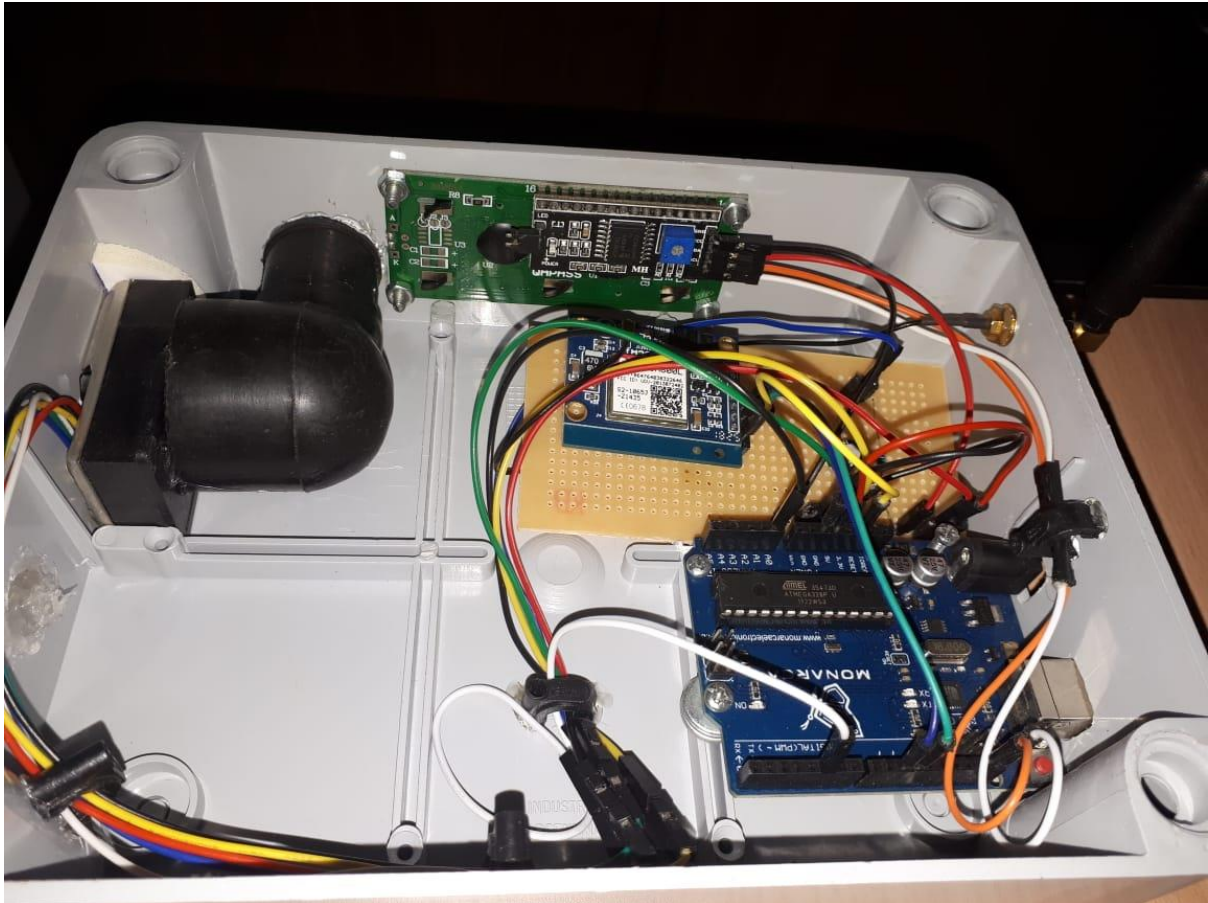


Fig.7.1: Interior del prototipo con vista trasera



Fig.7.2: Vista frontal del prototipo



Fig.7.3: Lateral Izquierdo del gabinete

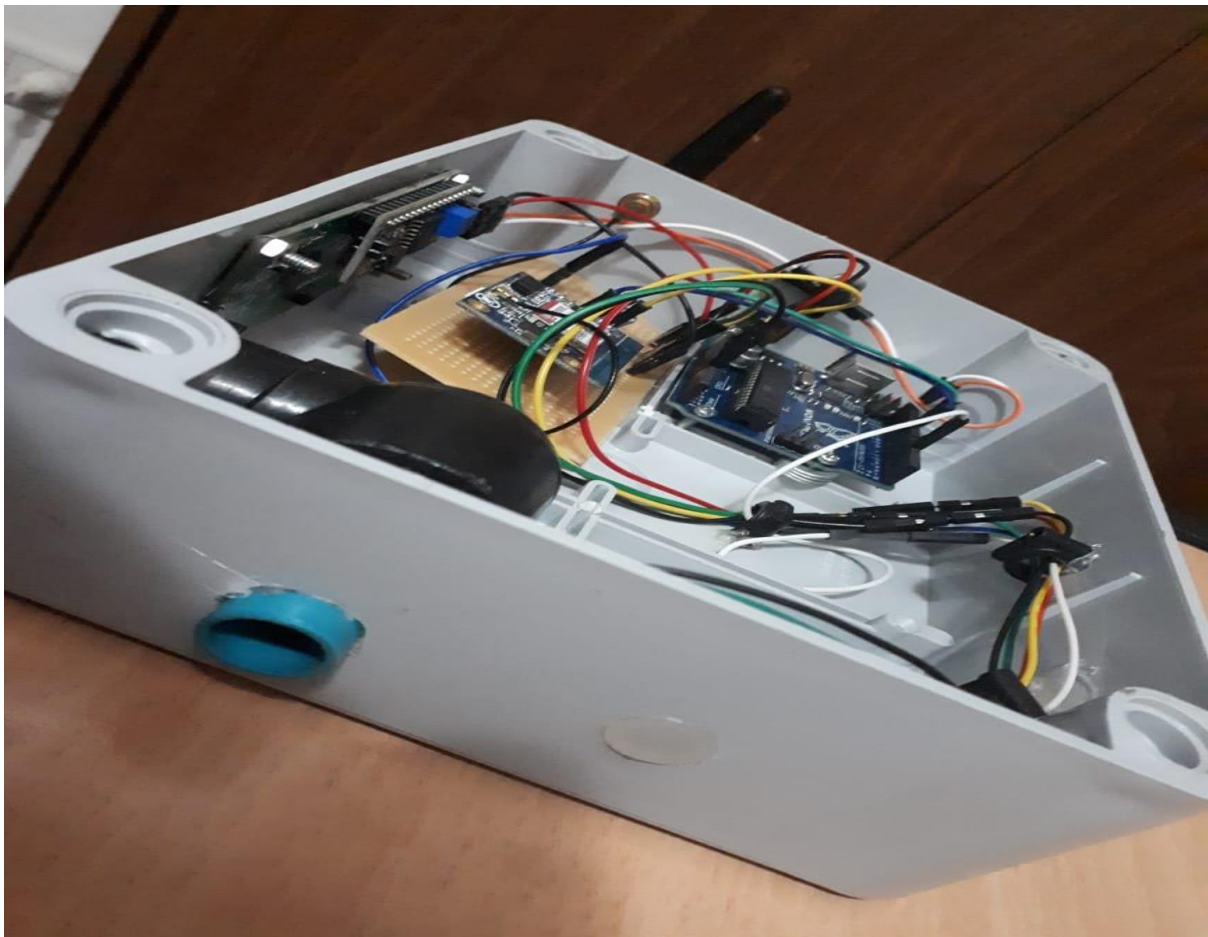


Fig.7.4: Lateral Derecho del gabinete



Fig.7.5: Entrada de Aire y Frente del sensor

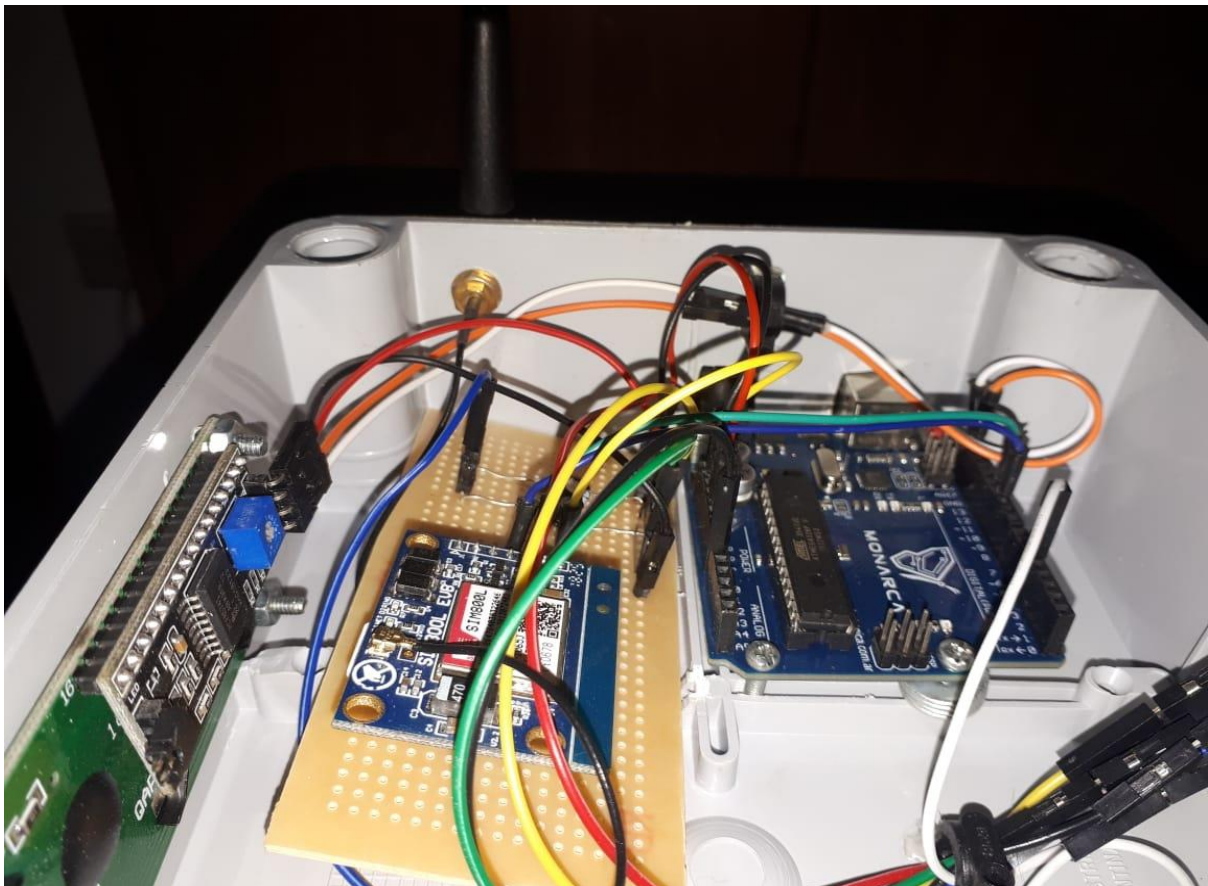


Fig.7.6: Proyección interior

## Capítulo 4:

### Experimentación, resultados y conclusiones

#### 4.1 MEDICIONES

Es necesario para este apartado, antes de pasar a más detalles, hacer algunas aclaraciones de consideración relacionadas a las mediciones y visualizaciones que se observarán más adelante. El sensor para poder obtener la concentración de PM presente en el ambiente utiliza dos variables, las cuales son:

- El valor del ADC (Conversor Analógico-Digital) que contiene el Arduino, el cual es obtenido a partir de las lecturas de las entradas/salidas analógicas. Dicho valor no tiene unidad y va desde 0 a 1023.
- Valor de tensión que representa la concentración obtenida, expresada en Volts [V]. Este valor se obtiene en base al valor que el ADC obtenga de la lectura de la entrada analógica del sensor.

De las 3, la que resulta de relevancia para el usuario final es justamente la concentración o polución de PM, las otras dos, se deja expresamente aclarado que las mismas no se las eleva a la plataforma, ya que su fin y uso son exclusivamente para debug, testing y troubleshooting en caso de fallas y/o mediciones sospechosas presentes en los reportes. Es posible acceder a estas dos variables utilizando el “Debug Mode” presente en el código.

Otras cuestiones, se responden en base a las siguientes preguntas:

- Se dice que el sensor es apto para la medición de partículas finas y de mayor tamaño incluso, pero ¿cuáles son las características de estas partículas?

Es aquí donde se debe presentar el concepto de Material Particulado (PM), es decir todas aquellas partículas que se encuentran flotando en el aire. Según la Organización Mundial de la Salud (OMS), este concepto es definido como: “Las PM son un indicador representativo común de la contaminación del aire. Afectan a más personas que cualquier otro contaminante.

Los principales componentes de las PM son los sulfatos, los nitratos, el amoníaco, el cloruro de sodio, el hollín, los polvos minerales y el agua. Consisten en una compleja mezcla de partículas sólidas y líquidas de sustancias orgánicas e inorgánicas suspendidas en el aire. Si bien las partículas con un diámetro de 10 micrones o menos ( $\leq PM_{10}$ ) pueden penetrar y alojarse profundamente dentro de los pulmones, existen otras partículas aún más dañinas para la salud, que son aquellas con un diámetro de 2,5 micrones o menos ( $\leq PM_{2.5}$ ). Las  $PM_{2.5}$  pueden atravesar la barrera pulmonar y entrar en el sistema sanguíneo. La exposición crónica a partículas contribuye al riesgo de desarrollar enfermedades cardiovasculares y respiratorias, así como cáncer de pulmón.”

- ¿Cuáles son las condiciones o parámetros para decir que el aire está limpio o puro?  
¿Cómo se lo mide?

Para responder estas preguntas nuevamente se citará textualmente lo dicho por la OMS: “Generalmente, las mediciones de la calidad del aire se notifican como concentraciones medias diarias o anuales de partículas  $PM_{10}$  por metro cúbico ( $m^3$ ) de aire. Las mediciones sistemáticas de la calidad del aire describen esas concentraciones de PM expresadas en microgramos ( $\mu$ )/ $m^3$ . Cuando se dispone de instrumentos de medición suficientemente sensibles, se notifican también las concentraciones de partículas finas ( $PM_{2.5}$  o más pequeñas).”

Dispuesto por las directrices internacionales, los parámetros establecidos a nivel mundial sobre la condición de aire limpio es la siguiente:

- Valores fijados en las Directrices
  - Partículas finas ( $PM_{2.5}$ ):
    - $10 \mu g/m^3$  de media anual
    - $25 \mu g/m^3$  de media en 24h

- Partículas gruesas (PM<sub>10</sub>)
  - 20 µg/m<sup>3</sup> de media anual
  - 50 µg/m<sup>3</sup> de media en 24h

En referencia más específicamente a calidad aire, existe otra organización a nivel global encargada de medir el ICA (Índice de Calidad de Aire) donde en su página web es posible observar en los puntos que haya disponible de un determinado país, cuál es la calidad del aire en tiempo real en esa zona, donde las mediciones de PM 2.5 y 10 son de grandísima importancia, con la diferencia que el ICA se calcula utilizando más variables que solamente el Material Particulado.

A nivel cualitativo, el rango del ICA está dividido en seis tramos:

- Buena: Color verde (ICA de 0 a 50)
- Moderada: Color amarillo (ICA de 51 a 100)
- Dañina a la salud para grupos sensibles: Color naranja (ICA de 101 a 150)
- Dañina a la salud: Color rojo (ICA 151 a 200)
- Muy dañina a la salud: Color morado (ICA 201 a 300)
- Peligrosa: Color marrón (ICA superior a 300)

Una vez realizadas las aclaraciones pertinentes, recordando que ThingSpeak tiene como poder de análisis, cómputo y procesamiento a las herramientas gráficas-estadísticas provistas por MATLAB, en las gráficas es posible obtener visualizaciones de una forma totalmente personalizada, siendo capaz de ejecutar de una forma muy sencilla cálculos estadísticos, diferentes tipos de vistas que en casos de aplicación como los de este proyecto son totalmente útiles.

Haciendo uso de una batería USB portátil, se procedió a realizar las mediciones bajo el siguiente contexto y planteando los siguientes 3 escenarios, con el objetivo de observar la diferencia existente entre los mismos:

- El día en el que fueron llevadas a cabo las mediciones, el clima se encontraba soleado, siendo que dos días antes de esto, llovió torrencialmente ambos días, con una temperatura aproximada de 22° C, sin presencia vientos fuertes y una humedad del 65%. Cantidad de muestras realizadas: 62.

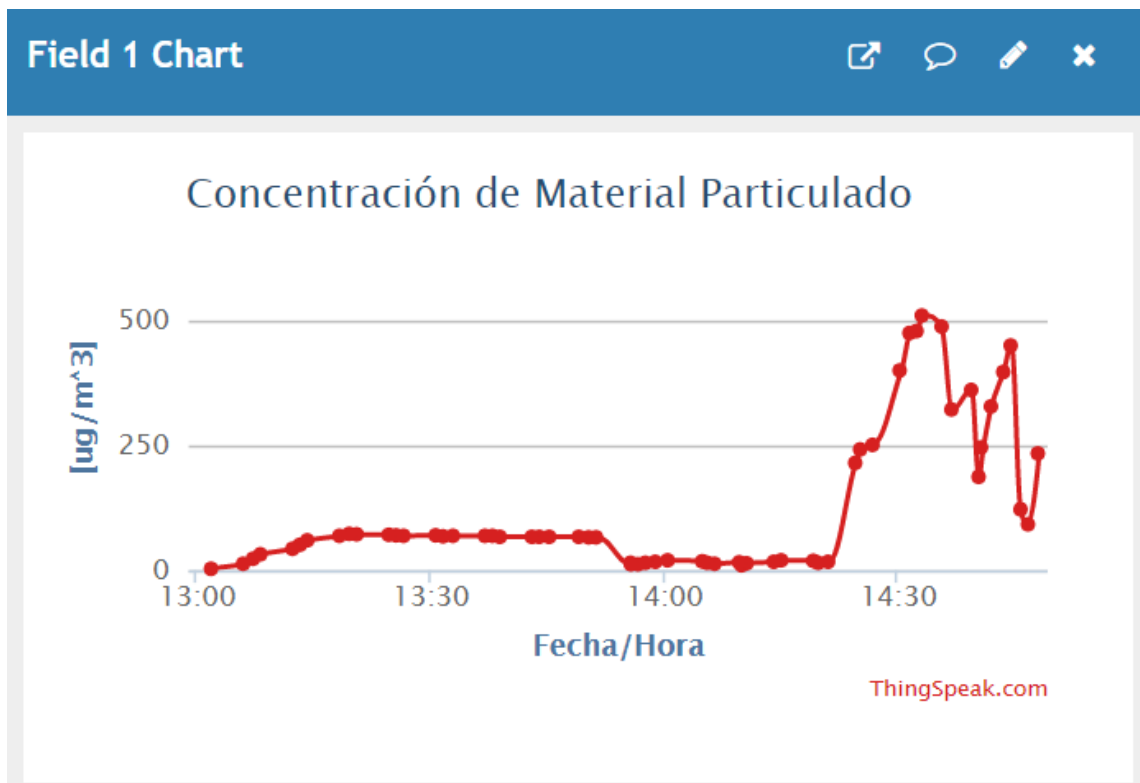


Fig.8: Visualización total de las muestras.

- Escenario 1: Frente de una casa, calle de tierra y con una circulación escasa de tráfico a dicha hora. Franja horaria de las mediciones en el escenario: 13 ~ 13:50hs.

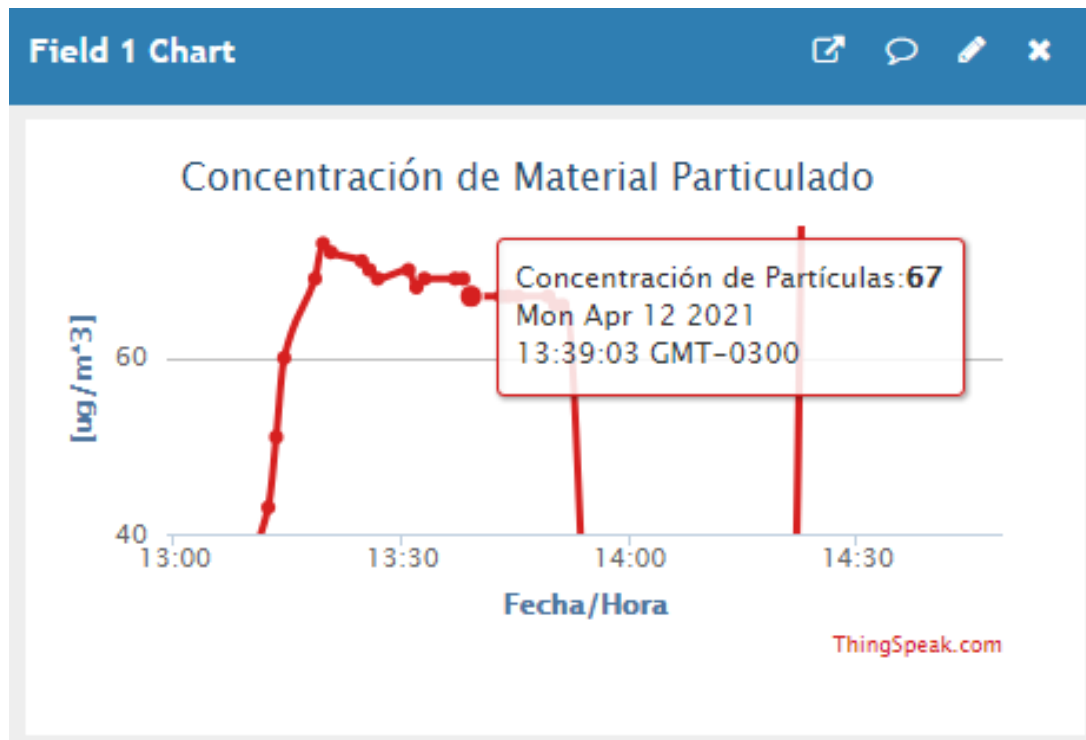


Fig.8.1: Valores implicados para Escenario 1

Como se observa en la imagen anterior, los valores implicados en este escenario comprenden valores entre 67 a 70  $\mu\text{g}/\text{m}^3$  en promedio, siendo que el tráfico fue realmente bajo, lo que da a pensar si este mismo fuese un poco más elevado y más aun tratándose de una superficie compuesta por brosa, piedras y materiales que suelen desprender fácilmente partículas al aire (sumado a todos los compuestos que los vehículos propiamente emiten) por lo que se deduce que si se hubieran cumplido dichas condiciones, perfectamente se hubiesen obtenido muestras cercanas a los 100  $\mu\text{g}/\text{m}^3$  o hasta incluso más. Siendo que dicho escenario ha sido donde se encendió el prototipo, se puede observar en la gráfica como el sensor inicialmente necesita de un tiempo de estabilización para poder comenzar a obtener muestras reales hasta tender a estabilizarse. Según los índices de calidad de aire el mismo estaría catalogado como “Moderado” según la tabla.

- Escenario 2: Habitación interior de la casa, ambiente sumamente limpio, ventanas cerradas, equipo posicionado encima del escritorio. Franja horaria de las mediciones en el escenario: 13:50 ~ 14:20 hs.

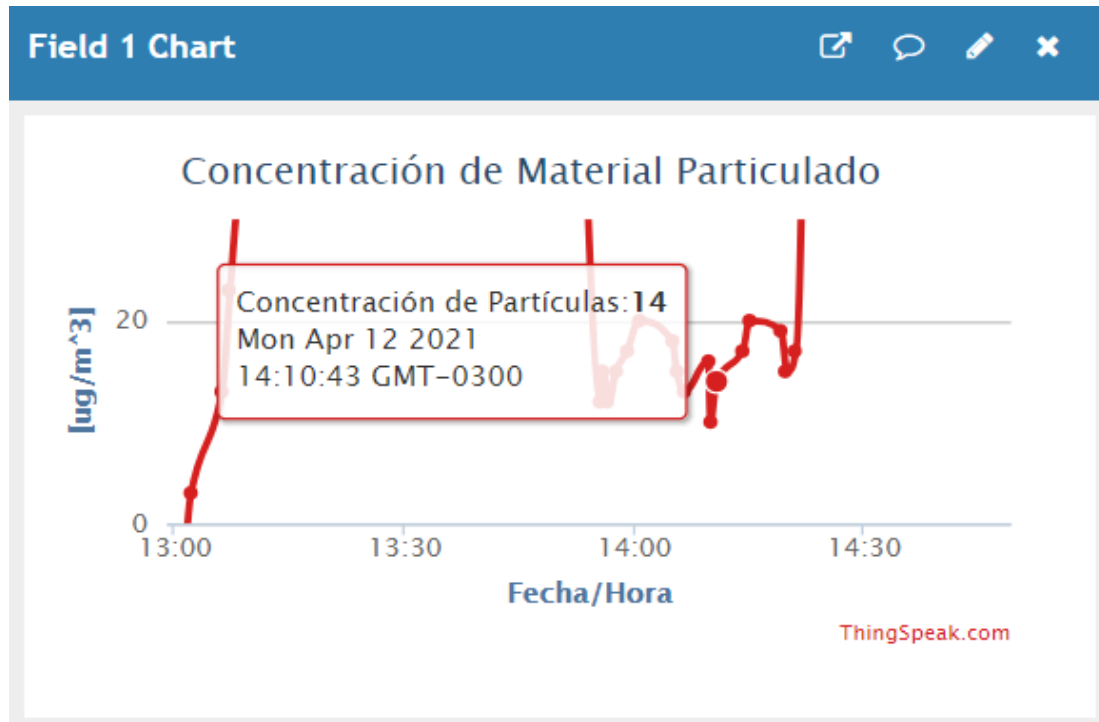


Fig.8.2: Valores implicados para Escenario 2

En dicho escenario los valores rondaron entre los 10 y 20  $\mu\text{g}/\text{m}^3$ , teniendo una distribución más inestable para el momento de que el sensor tomaba las muestras. En referencia a lo dicho por los estándares mundiales de índice de calidad de aire, el ambiente estaría catalogado como “Aire Limpio” inicialmente para el tiempo en el que ha permanecido el prototipo funcionando.

- Escenario 3: Mismo habitáculo, pero ahora en presencia de un sahumerio encendido e intercambiado por humo provocado por fósforos. Prototipo sobre el escritorio, materiales colocados debajo de este, provocando intencionalmente que la combustión se dirigiese hacia la zona de detección del sensor. Franja horaria de las mediciones en el escenario: 14:20 ~14:50 hs.

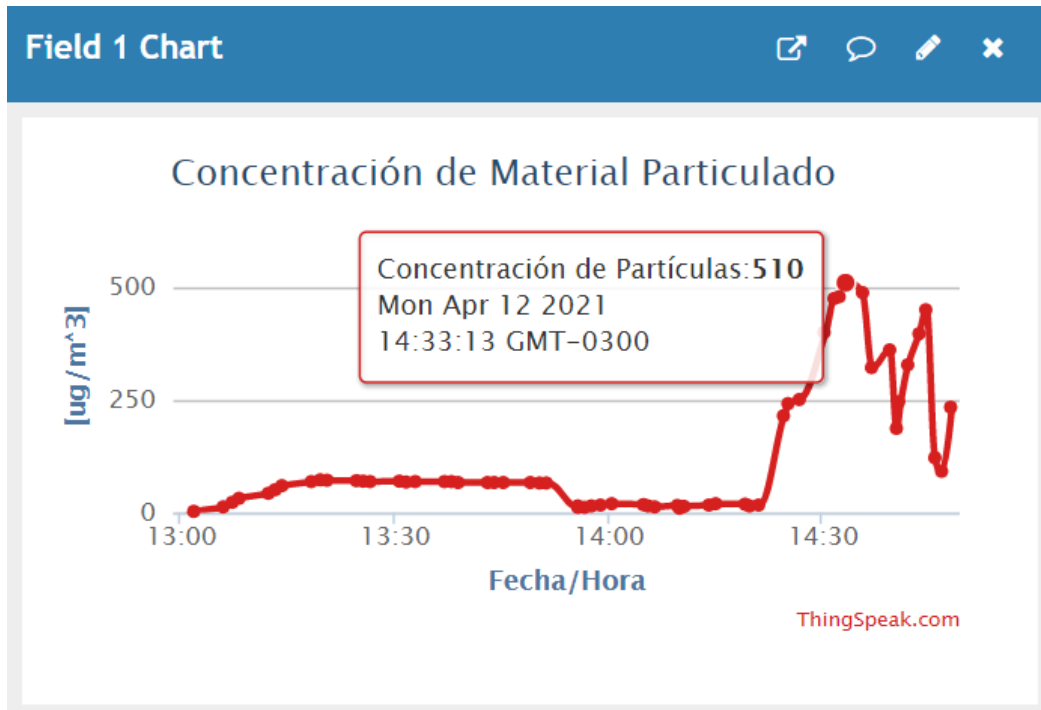


Fig.8.3: Valor máximo reportado en escenario 3

Es aquí donde realmente se dio a la sorpresa de la grandísima concentración que existe cuando el sensor detecta estas cantidades de partículas, y hasta incluso, no logrando poder recabar una muestra reportada en la gráfica, pero si visualizada en el Debug Mode en algunas pruebas previas a realizar la subida de datos (ya que aquí es posible obtener mayor cantidad de muestras por segundo) donde se alcanzaron incluso valores superiores a los 600  $\mu\text{g}/\text{m}^3$ , lo que realmente es alarmante siendo que no se saben los efectos nocivos que podrían traer para la salud el aire que se está respirando en ese momento y que normalmente la población no toma en consideración estos hechos debido a la falta de conocimiento en el tema de contaminación ambiental, y sobre todo, que no es indiferente al lugar, ya que estas partículas perfectamente pueden acabar o provenir desde el interior de nuestros hogares. Esto es lo que según la OMS y el índice de calidad aire cataloga desde “Dañina” hasta “Muy peligrosa” dichas concentraciones para la salud, permaneciendo en contacto con esta durante largos períodos de tiempo.

A continuación, se incluyen capturas del promedio de las mediciones efectuadas en un intervalo de cada 15 minutos, y cuál fue el promedio en el día, seguido de la vista de configuración de como configurar las vistas para obtener dichos valores:

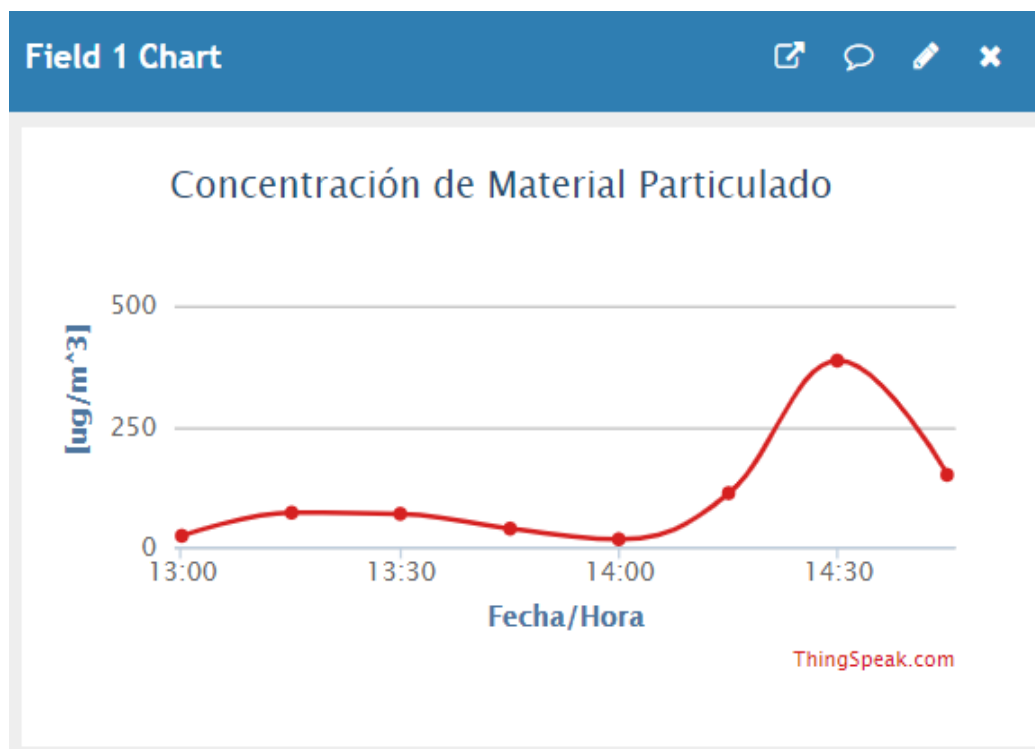


Fig.8.4: Promedio de las muestras cada 15 minutos

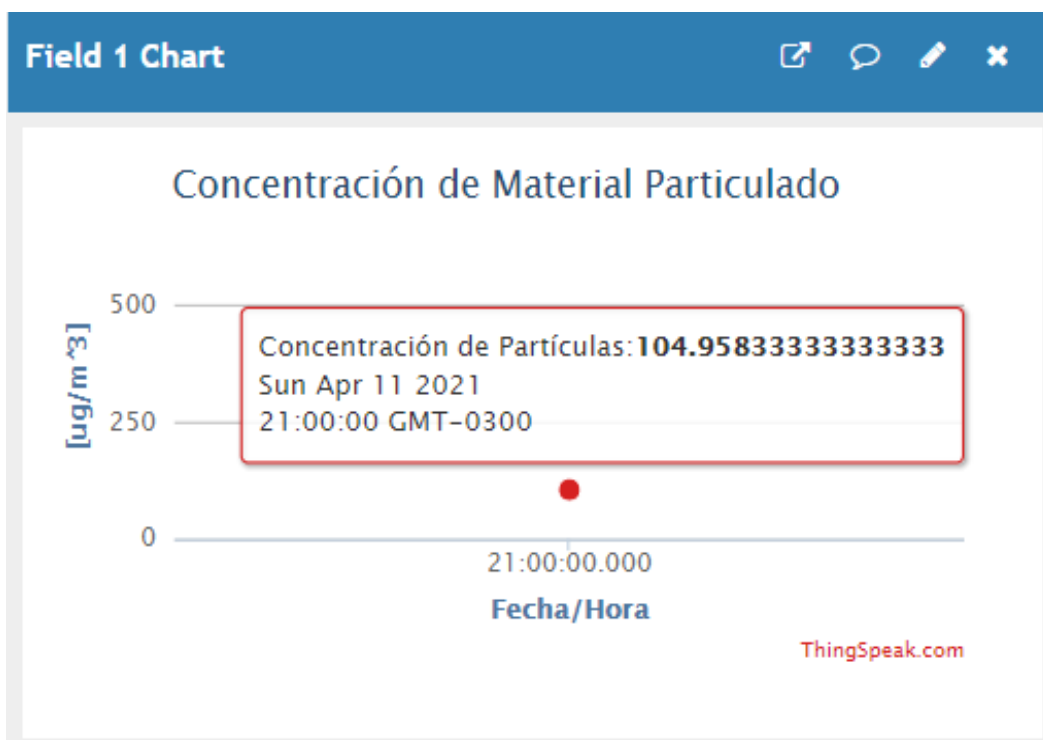


Fig.8.5: Promedio de todo el período de mediciones

Field 1 Chart Options ×

|             |                           |             |      |
|-------------|---------------------------|-------------|------|
| Title:      | Concentración de Material | Timescale:  | ▼    |
| X-Axis:     | Fecha/Hora                | Average:    | 15 ▼ |
| Y-Axis:     | [ug/m <sup>3</sup> ]      | Median:     | ▼    |
| Color:      | #d62020                   | Sum:        | ▼    |
| Background: | #ffffff                   | Rounding:   |      |
| Type:       | spline ▼                  | Data Min:   |      |
| Dynamic?:   | true ▼                    | Data Max:   |      |
| Days:       |                           | Y-Axis Min: | 0    |
| Results:    | 62                        | Y-Axis Max: | 650  |

Save

Cancel

Fig.8.6: Panel de configuración de vista para obtener promedio cada 15 minutos

Asimismo, para configurar la vista del promedio de las muestras en el día, basta con ir a la misma solapa de “Average” (Promedio en inglés) y tildar en la opción que nos aparecía una vez desplegada la misma, en “Daily” (Diaria). Lo mismo es posible realizar con cada cálculo estadístico propuesto por este panel, permitiendo por ejemplo variar la cantidad de datos involucrados en los cálculos, los intervalos de valores con los que se quiere obtener el resultado de dicho cálculo, cantidad de días sobre los que se quiere calcular, entre otras tantas. Hasta incluso, si la función o necesidad del usuario no se encontrase dentro de las ofrecidas, ThingSpeak deja la posibilidad dentro de la API, de utilizar código MATLAB para manejar las vistas y algoritmos matemáticos que sean requeridos al elevar datos a la plataforma, permitiendo también por ejemplo, descargar una planilla de cálculo con el histórico de todas las muestras reportadas, y luego, subir la tabla nuevamente, lo que acondicionaría los gráficos en caso de errores o fallos de elevación, lo que realmente hace de ThingSpeak una de las opciones más versátiles y sobre todo potentes en la manipulación y manejo de los datos en cualquier proyecto, solución o infraestructura IoT.

## 4.2 RESULTADOS

Tras la implementación, prueba, puesta a punto y adaptaciones realizadas al prototipo, los resultados obtenidos son más que satisfactorios. Teniendo en cuenta los objetivos principales de este proyecto, la solución logra recuperarse en la gran mayoría de las situaciones ante los fallos que puedan ocurrir, se logró cuantificar con éxito las diferentes densidades de las partículas que por el sensor se han atravesado, inicialmente con algunos problemas de resolución pero que luego fueron resueltos adaptando el software, los módulos en general y por ende el sistema en su totalidad, salvo por cuestiones de diseño y ensamblado que por razones de presupuesto no se han podido implementar, así como también la inclusión de mayor cantidad de sensores. El sistema al utilizar la infraestructura y red celular, dependiendo la situación se encuentra limitado al momento del encendido debido a si el módulo consigue un registro y acceso rápido en la red o no, y si la conexión durante el momento de la transmisión es óptima y no concluye en la caída del socket a la API de la plataforma, donde en algunos casos se ha tenido que reiniciar manualmente el módulo, pero dejando de lado estos dos factores anteriores, la solución, al ser relativamente económica y simple funciona de la forma esperada.

#### 4.3 PROPUESTA DE FUTURAS ACCIONES

En la presente sección se dará mención a un conjunto de propuestas/ideas que podrían implementarse a futuro si se decidiese continuar con el perfeccionamiento de la solución y que son vitales para su evolución:

- Para dar una respuesta completa en lo que a “Calidad de Aire” se refiere, se deberían incluir mayor cantidad de sensores que captasen demás variables físicas indispensables para las mediciones, deducciones y estudios del aire en general como puede ser: Presión Atmosférica, Temperatura, Humedad y Concentración de Gases.
- Incluir en la solución para evitar la pérdida de muestras que podrían resultar significativas, algún sistema de almacenamiento local como puede ser una memoria extraíble SSD.
- Estudiar la posibilidad de incluir a la solución, algoritmos de aprendizaje inteligente para la toma de decisiones en real-time que no dependan solamente de las posibilidades consideradas por el propio sistema.
- Implementar como alternativa de enlace y conectividad, electrónica LoRa, LoRaWAN como protocolo de red y MQTT como protocolo de mensajes para evitar dependencia de la infraestructura móvil.
- Diseño de un gabinete especial para el prototipo que se adapte perfectamente en tamaño, medidas y requisitos especiales para este tipo de soluciones.
- Para garantizar el funcionamiento independiente del prototipo y su continuidad de trabajo donde no pueda ser conectado directamente a la red eléctrica, analizar la posibilidad de incorporar baterías sustentadas por paneles solares.

#### 4.4 CONCLUSIONES

Las conclusiones obtenidas luego de todo el proceso recorrido en la elaboración, trabajo y desarrollo de esta prueba de concepto, realmente son enriquecedoras en muchos aspectos, ya sea desde lo técnico/tecnológico, hasta cuestiones relacionadas a lo que el mundo IoT, las redes y en general, las Telecomunicaciones pueden aportar enormemente a un ámbito sumamente importante como es el de la salud. El PM pasa desapercibido por el común de la gente y no se tiene una noción clara y establecida de lo que la misma puede llegar a provocar en el cuerpo, y que, sin darnos cuenta, estamos en contacto permanente con ella en mayores o menores concentraciones.

Ya desde una perspectiva más general de todo el proceso, personalmente considero que ha sido una grandísima oportunidad de aprendizaje y contacto con el mundo actual en lo que se refiere a encarar un proyecto de IoT, el proceso de selección de tecnologías, teniendo en cuenta los objetivos propuestos, el presupuesto con el que se cuenta y cuán mayor o menor puede ser la prestación final del sistema en base a todo lo decidido anteriormente. La utilización de tecnología celular y todo lo que conlleva esto, partiendo desde la base que el módulo por sí solo no funciona, se debe configurar, registrar y acceder a la red del operador, ha sido otro de los grandes puntos que rescato como beneficio de esta solución, que si bien se implementó una tecnología cada vez más en desuso como es 2G, el proceso es siempre el mismo independientemente de la red, destacando la gran cantidad de cosas que un celular hace para poder brindar conectividad y por lo general, nunca nos percatamos de esto.

Como punto final, volver a destacar la suma importancia, beneficios, versatilidad y eficacia en la toma de decisiones que una API permite hoy en día, con la cual nunca había trabajado con una de ellas utilizando dispositivos externos a una computadora, siendo una experiencia gratificante, la cual permite más adelante continuar adaptándose hacia las demás haciendo uso de otros módulos/tecnologías de mayores y complejas soluciones con posibilidad de implementación comercial.

## ANEXOS

|                                |                                                       |
|--------------------------------|-------------------------------------------------------|
| Microcontroller                | ATmega328P                                            |
| Operating Voltage              | 5V                                                    |
| Input Voltage<br>(recommended) | 7-12V                                                 |
| Input Voltage (limit)          | 6-20V                                                 |
| Digital I/O Pins               | 14 (of which 6 provide PWM output)                    |
| PWM Digital I/O Pins           | 6                                                     |
| Analog Input Pins              | 6                                                     |
| DC Current per I/O Pin         | 20 mA                                                 |
| DC Current for 3.3V Pin        | 50 mA                                                 |
| Flash Memory                   | 32 KB (ATmega328P) of which 0.5 KB used by bootloader |
| SRAM                           | 2 KB (ATmega328P)                                     |
| EEPROM                         | 1 KB (ATmega328P)                                     |
| Clock Speed                    | 16 MHz                                                |
| LED_BUILTIN                    | 13                                                    |
| Length                         | 68.6 mm                                               |
| Width                          | 53.4 mm                                               |
| Weight                         | 25 g                                                  |

Tabla 3: Especificaciones técnicas ARDUINO UNO R3. Store Oficial de Arduino. Consultado 3 de febrero 2021.  
Fuente: <https://store.arduino.cc/usa/arduino-uno-smd-rev3>

## SIM800 L v2.0 EVB



Fig.9: SIM 800l v2 EVB reverso y frente del módulo



Fig.9.1: SIM 800l v2 EVB + Antena

SIM800L en su versión 2.0 de SIMCOM Company en conjunto con Microchip Technology Inc. es un módulo QUAD BAND GSM / GPRS en el cual se pueden utilizar y aprovechar la gran mayoría de los servicios que prestan estas dos tecnologías como lo son SMS, llamadas de corta y larga distancia, sincronización horaria por red, transmisión de datos según lo estipulado por las limitaciones de 2G, configuración de alarmas, manejo y almacenamiento de contactos, entre otras tantas funcionalidades extras. Una de las mayores ventajas de este módulo son los niveles de serie VCC y TTL que tienen un voltaje de operación 5 V, siendo que, habiendo tantos módulos GPRS / GSM en el mercado que necesitan agregar un regulador de 5V y un circuito convertidor de nivel, el mismo ya los trae incorporado lo que hace que su conexión e incorporación a otros sistemas de este rango de voltaje sea totalmente directo, además de contar con un puerto serie para depuración y en su defecto también, actualizar el firmware del dispositivo. Para utilizar el módulo es necesario la incorporación y anexo de un chip de un operador registrado.

Se añaden a continuación, las principales características técnicas del módulo extraídas de su correspondiente hoja de datos:

| Feature            | Implementation                                                                                                                                                                                                                                                                                                |
|--------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Power supply       | 3.4V ~4.4V                                                                                                                                                                                                                                                                                                    |
| Power saving       | typical power consumption in sleep mode is 0.7mA (AT+CFUN=0 )                                                                                                                                                                                                                                                 |
| Frequency bands    | <ul style="list-style-type: none"> <li>● Quad-band: GSM 850, EGSM 900, DCS 1800, PCS 1900. SIM800L can search the 4 frequency bands automatically. The frequency bands can also be set by AT command "AT+CBAND". For details, please refer to document [1].</li> <li>● Compliant to GSM Phase 2/2+</li> </ul> |
| Transmitting power | <ul style="list-style-type: none"> <li>● Class 4 (2W) at GSM 850 and EGSM 900</li> <li>● Class 1 (1W) at DCS 1800 and PCS 1900</li> </ul>                                                                                                                                                                     |
| GPRS connectivity  | <ul style="list-style-type: none"> <li>● GPRS multi-slot class 12 ( default )</li> <li>● GPRS multi-slot class 1~12 (option)</li> </ul>                                                                                                                                                                       |
| Temperature range  | <ul style="list-style-type: none"> <li>● Normal operation: -40°C ~ +85°C</li> </ul>                                                                                                                                                                                                                           |

Tabla 4: Características técnicas SIM800L v2 [1]

|                  |                                                                                                                                                                                                                                                                                                                                                                                                                    |
|------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Data GPRS        | <ul style="list-style-type: none"> <li>● GPRS data downlink transfer: max. 85.6 kbps</li> <li>● GPRS data uplink transfer: max. 85.6 kbps</li> <li>● Coding scheme: CS-1, CS-2, CS-3 and CS-4</li> <li>● PAP protocol for PPP connect</li> <li>● Integrate the TCP/IP protocol.</li> <li>● Support Packet Broadcast Control Channel (PBCCH)</li> <li>● CSD transmission rates: 2.4, 4.8, 9.6, 14.4 kbps</li> </ul> |
| CSD              | <ul style="list-style-type: none"> <li>● Support CSD transmission</li> </ul>                                                                                                                                                                                                                                                                                                                                       |
| USSD             | <ul style="list-style-type: none"> <li>● Unstructured Supplementary Services Data (USSD) support</li> </ul>                                                                                                                                                                                                                                                                                                        |
| SMS              | <ul style="list-style-type: none"> <li>● MT, MO, CB, Text and PDU mode</li> <li>● SMS storage: SIM card</li> </ul>                                                                                                                                                                                                                                                                                                 |
| SIM interface    | Support SIM card: 1.8V, 3V                                                                                                                                                                                                                                                                                                                                                                                         |
| External antenna | Antenna pad                                                                                                                                                                                                                                                                                                                                                                                                        |

Tabla 4: Características técnicas del SIM800l v2 [2]

| Coding scheme | 1 timeslot | 2 timeslot | 4 timeslot |
|---------------|------------|------------|------------|
| CS-1          | 9.05kbps   | 18.1kbps   | 36.2kbps   |
| CS-2          | 13.4kbps   | 26.8kbps   | 53.6kbps   |
| CS-3          | 15.6kbps   | 31.2kbps   | 62.4kbps   |
| CS-4          | 21.4kbps   | 42.8kbps   | 85.6kbps   |

Tabla 4.2: Características técnicas del SIM800l v2 [3]

### Sensor GP2Y1010AUoF



Fig.10: Módulo Sharp GP2Y1010AUoF

El sensor GP2Y1010AU0F de la Compañía Sharp está especialmente diseñado para detectar partículas muy finas de polvo (PM 2.5 por ejemplo), polen y humo de cigarrillo en el aire, siendo es muy utilizado como detector en purificadores y acondicionadores de aire. En la Figura 10 se puede observar el módulo, el cual internamente contiene un diodo emisor de luz (LED) y un fotodiodo colocados de forma tal que el mismo es capaz de detectar las reflexiones de luz sobre las partículas de polvo y de esta forma determinar sus características.

Es un dispositivo que su consumo, de lo que a corriente se refiere, es realmente bajo, alrededor de 20 [mA] y soporta alimentación de hasta 7 [V] DC. Su salida analógica es un voltaje lineal y proporcional a la cantidad de polvo detectado. La sensibilidad del sensor es de 0.5 [V] por cada 0,1 [mg/m<sup>3</sup>].

Para su conexión tiene un conector JST3 de 6 pines y se entrega con su cable correspondiente, y, si se lo pretende conectar a una placa Arduino UNO, se debe añadir una resistencia de 150 [ $\Omega$ ] y un capacitor de 220 [ $\mu$ F].

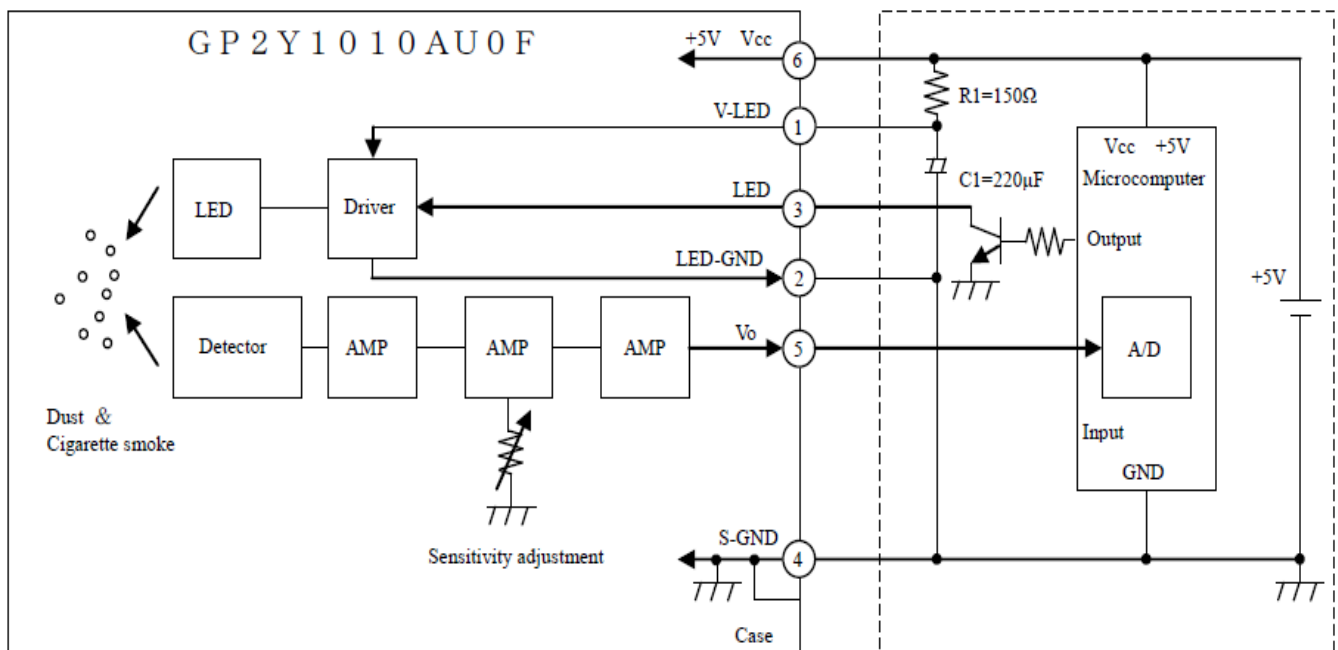


Fig.11: Ejemplo de conexionado Sensor-Placa.  
Fuente: Datasheet oficial Sharp Company- GP2Y1010AU0F

La explicación y motivos de añadir al conexionado los dos componentes anteriormente mencionados viene dado a que, gracias a esos componentes, y a la configuración interna del sensor, son necesarios para que, la detección por pulsos del módulo funcione correctamente.

| Parameter   | Symbol | Specified condition | Recommended condition | Unit |
|-------------|--------|---------------------|-----------------------|------|
| Pulse cycle | T      | 10                  | $10 \pm 1$            | ms   |
| Pulse width | Pw     | 0.32                | $0.32 \pm 0.02$       | ms   |

Tabla.5: Condiciones óptimas de trabajo del Sensor.  
Fuente: Datasheet oficial Sharp Company- GP2Y1010AUoF

La muestra obtenida por el fotodetector pasa a la siguiente etapa, donde la misma es amplificada por el circuito amplificador al mismo tiempo que el diodo se detiene sincronizadamente. El valor que se obtiene para la salida del dispositivo es el que se mide 0,28 ms después de que se enciende el LED. Por lo tanto, se recomienda que el microordenador al que se encuentra conectado el sensor lea la salida 0,28 ms después de la emisión del LED también.

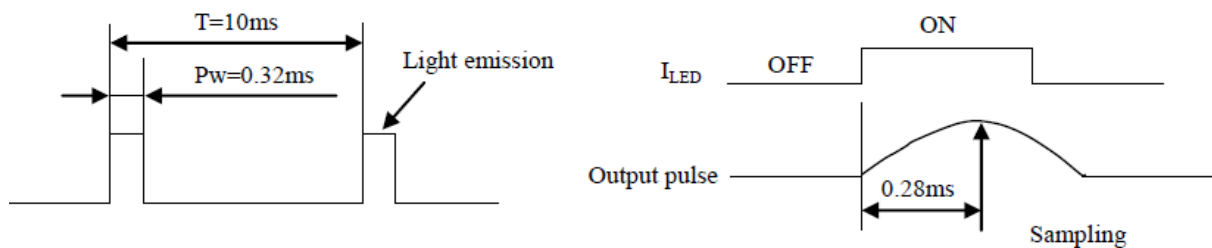


Fig.12: Proceso de muestreo  
Fuente: Datasheet oficial Sharp Company- GP2Y1010AUoF

Por último, se describe el proceso de como el módulo detecta las diferentes partículas circundantes en el aire.

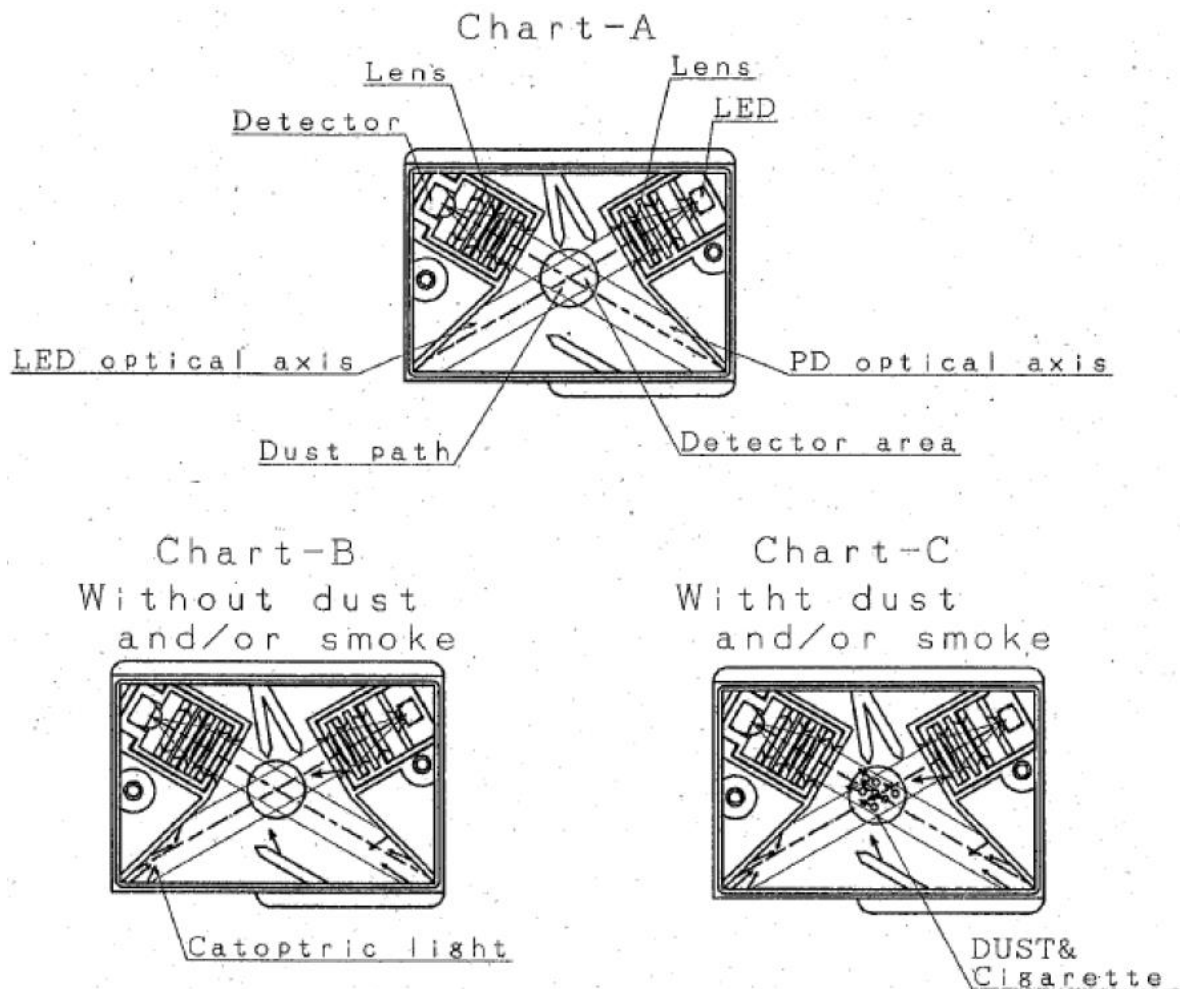


Fig.13: Método de detección de partículas que utiliza el sensor.  
Fuente: Datasheet oficial Sharp Company- GP2Y1010AUoF

La luz generada por el LED se detecta con una lente y una rendija como se muestra en el gráfico-A. También para el fotodiodo se coloca una lente y una rendija frente a él para cortar la luz de la cual no es producto del reflejo de alguna partícula, y de esta forma, detectar el verdadero reflejo que es generado por la presencia de material circundante en el aire. El área donde se cruzan esos dos ejes ópticos es el área de detección del dispositivo. El gráfico-B muestra lo que ocurre dentro del dispositivo cuando no hay polvo y el gráfico-C cuando existe presencia de partículas en el área de detección. Aquí, el fotodiodo es capaz de detectar la partícula debido a justamente la reflexión que la misma genera sobre la luz del emitida por el

LED, donde, el consumo de corriente por partícula, viene dada en proporción a la cantidad de luz detectada en el fotodiodo, donde luego se generará una salida de voltaje analógica (salida de pulso) después de que el circuito amplificador amplifica la muestra de corriente del detector.

Un dato curioso y a tener en cuenta, es que el dispositivo genera una salida de voltaje incluso cuando no se detecta polvo. Esto se debe a que la luz emitida por el LED se refleja en la carcasa del dispositivo, donde una parte termina incidiendo en el detector.

Display LCD I<sup>2</sup>C 16x2

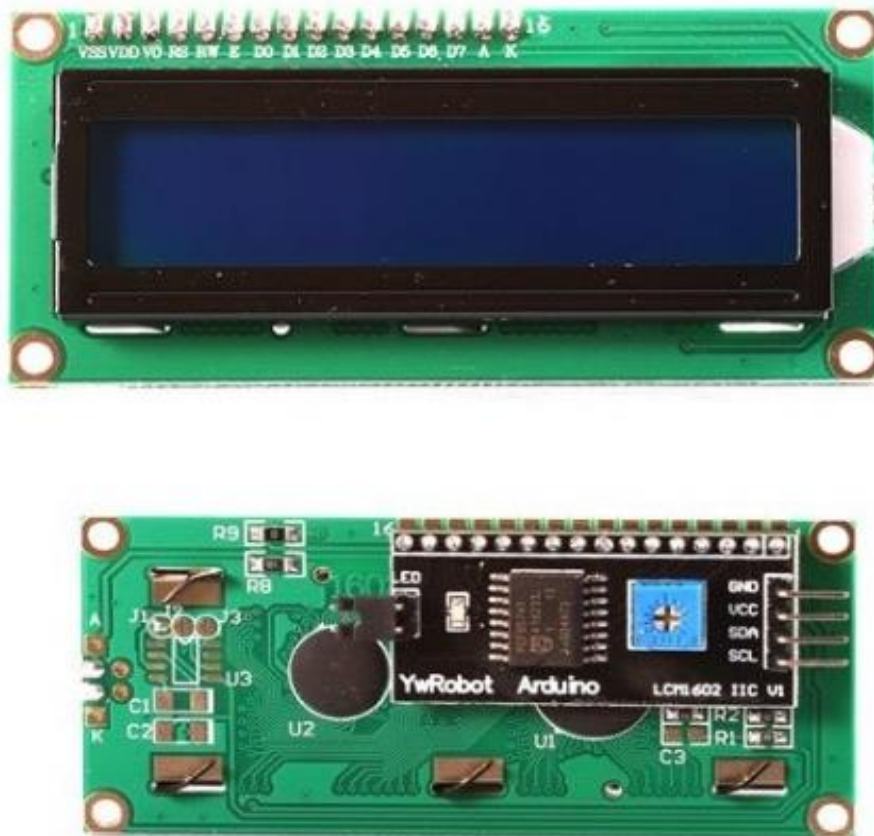


Fig.14: Display I2C 16x2

El último de los componentes de la solución es un display de 16 caracteres por 2 líneas (por ello la abreviatura 16x2), el cual como se mencionó en otras secciones, el mismo tiene por finalidad sólo la visualización de todo el proceso que el prototipo está realizando en real-time, permitiendo así, en caso de fallas, o atascos en campo cuando el mismo se encuentre operando, detectar puntualmente donde se ha detenido, y tener un estimativo desde hace cuánto ha dejado de reportar, no pudiendo así, reponerse para continuar el proceso.

Un detalle interesante a mencionar, es el que se observa en la Figura 14 en la parte superior del display, por encima de la pantalla, la cantidad de pines que el módulo tiene, y que luego, en el reverso del mismo, se observa que terminan siendo 4 pines tipo hembra los necesarios para su conexión: GND-VCC-SDA-SCL. Esto se debe justamente al protocolo de bus serie de datos desarrollado por Philips Semiconductors más conocido como I<sup>2</sup>C (Inter Integrated Circuits), el cuál dentro de tantas ventajas técnicas a nivel transmisión, la más importante a nivel físico y desarrollo, es justamente el ahorro de conexiones, remitiéndose solamente a conectarle al módulo, alimentación (VCC- 5V en este caso), tierra (GND), el sistema de reloj (Clock) o sincronización del dispositivo (SCL) y por último el sistema de datos o transferencia de data (SDA).

El jumper que se observa en el extremo contrario a las salidas I<sup>2</sup>C no es nada más ni nada menos que el controlador de la linterna que el módulo trae consigo para iluminar la pantalla, además del propio controlador de brillo de los caracteres que se encuentra entre las salidas y el jumper de color celeste.

## THINGSPEAK



Fig.15: Logo de la plataforma IOT ThingSpeak

Según palabras de sus propios desarrolladores, ellos la definen como: “ThingSpeak™ es un servicio de plataforma de análisis de IoT que le permite agregar, visualizar y analizar flujos de datos en vivo en la nube. ThingSpeak proporciona visualizaciones instantáneas de los datos publicados por sus dispositivos en ThingSpeak. Con la capacidad de ejecutar código MATLAB® en ThingSpeak, puede realizar análisis y procesamiento en línea de los datos a medida que ingresan. ThingSpeak se usa a menudo para la creación de prototipos y pruebas de sistemas de IoT de concepto que requieren análisis.”

Dentro de sus capacidades o funcionalidades, ThingSpeak ofrece:

- Configurar fácilmente dispositivos para el envío de datos utilizando protocolos populares de IoT.
- Visualización de datos de sus sensores en tiempo real.
- Agregar datos a pedido de fuentes de terceros.
- Utilizar el poder de MATLAB para dar sentido a sus datos de IoT.
- Ejecutar análisis automáticamente en función de horarios o eventos.
- Crear prototipos/sistemas IoT sin configurar servidores ni desarrollar software web.
- Actuar automáticamente sobre los datos además de comunicarlos mediante servicios como Twilio® o Twitter®.

Debido a todo lo anterior y la cantidad de facilidades y beneficios que la plataforma tiene para ofrecer es por lo que se ha elegido a ThingSpeak para ser la que almacene y realice la analítica de todas las muestras que el sensor obtenga, utilizando su servicio de licencia gratuita la cual, a pesar de tener algunas limitaciones, para los fines del prototipo cumple a la perfección todos los requerimientos que el mismo requiere. Los mismos se encuentran detallados a continuación:

|                                           | FREE<br>For time-limited commercial evaluation of the service |
|-------------------------------------------|---------------------------------------------------------------|
| Scalable for larger projects              | <b>×</b><br>No. Annual usage is capped.                       |
| Number of messages                        | <b>3 million/year</b><br>(~8200/day) <sup>(2)</sup>           |
| Message update interval limit             | <b>Every 15 seconds</b>                                       |
| Number of channels                        | <b>4</b>                                                      |
| MATLAB Compute Timeout                    | <b>20 seconds</b>                                             |
| Number of simultaneous MQTT subscriptions | <b>Limited to 3</b>                                           |
| Private channel sharing                   | <b>Limited to 3 shares</b>                                    |
| Technical Support                         | <b>Community Support</b>                                      |

Tabla.6: Limitaciones de Licencia Gratuita ThingSpeak. Sitio Oficial de ThingSpeak. Consultado 8 de febrero 2021. Fuente: [https://thingspeak.com/prices/thingspeak\\_standard](https://thingspeak.com/prices/thingspeak_standard)

## REFERENCIAS

### BIBLIOGRAFÍA PRINCIPAL

- *CASE: Web general del Congreso Argentino de Sistemas Embebidos. Disponible en: <http://www.sase.com.ar/case/ediciones/case2020/>*
- *GP2y1010auof Hoja de datos del sensor GP2y1010auof [online], Sharp, 2010. Disponible en: <http://www.sharp-world.com>*
- *SIM800l v2 EVB Hoja de datos del módulo celular y Manual de Comandos AT. Disponible en: <https://drive.google.com/drive/folders/15NpPQyA4GqqnqzNYzdQbd2YPk5RaBupg>*
- *Arduino Plataforma Arduino: [www.arduino.cc](http://www.arduino.cc)*
- *Politi, M. (septiembre 2020). Charla Virtual "Internet de las cosas". Mes de la Ciencia y la Tecnología - FCyT UADER. Recuperado de: <https://youtu.be/RTxL5lB8gwg>.*
- *ThingSpeak Plataforma IoT. Disponible en: <https://thingspeak.com/>*
- *Viktor Lytsus, repositorio personal del código del sensor Sharp. Disponible en: <https://github.com/vlytsus/arduinosenzor>*
- *Normativa Ley Nacional n° 20248 [online]. Disponible en: <https://www.argentina.gob.ar/normativa/nacional/ley-20284-40167/texto>*
- *Organización Mundial de la Salud, Sitio Oficial: Calidad de Aire. Disponible: [https://www.who.int/es/news-room/fact-sheets/detail/ambient-\(outdoor\)-air-quality-and-health](https://www.who.int/es/news-room/fact-sheets/detail/ambient-(outdoor)-air-quality-and-health)*
- *Índice de Calidad de Aire. World Project. Disponible en: <https://aqicn.org/here/>*
- *Unión Internacional de Telecomunicaciones (ITU): Recomendación V.25ter: Marcación y control automáticos asíncronos en serie. Disponible en: <https://www.itu.int/rec/T-REC-V.25ter/es>*

- *Unión Internacional de Telecomunicaciones (ITU): Recomendación V.250: Marcación y control automáticos asíncronos en serie. Disponible en: <https://www.itu.int/rec/T-REC-V.250/es>*
- *Rafael Nova (26 de septiembre, 2013). Blog personal. Disponible en: <https://sites.google.com/site/telecomunicacionesredes/comandos-hayes-1>*

#### MATERIAL EXTRA

- *Material provisto por el Ing. Farina Mario en la Cátedra de: “Comunicaciones Móviles” (2019).*