



INSTITUTO UNIVERSITARIO AERONÁUTICO
DEPARTAMENTO DE ELECTRÓNICA Y TELECOMUNICACIONES

Trabajo final de grado

SISTEMA DE MONITOREO DE LA CALIDAD DEL AIRE INTEGRADO A IoT

Este trabajo es presentado por Ezequiel Campoli Marciszack para
obtener el título de Ingeniero en Telecomunicaciones.

Supervisado por:
Esp. Ing. José María Ducloux



INSTITUTO UNIVERSITARIO AERONÁUTICO
FACULTAD DE INGENIERÍA

Aprobado por el Departamento de Electrónica y Telecomunicaciones en cumplimiento de los requisitos exigidos para otorgar el título de Ingeniero en Telecomunicaciones al Señor: CAMPOLI MARCISZACK, Ezequiel - DNI 32080833

Revisó

Tutor de trabajo:

Esp. Ing. José María Ducloux

Directora de Departamento:

Esp. Ing. Marcela Beatriz Busnardo

Firma

Firma

Fecha

Fecha

Tribunal evaluador

Titular 1:

Ing. Carlos Chicala

Titular 2:

Ing. Walter Lancioni

Firma

Firma

Fecha

Fecha



Dedicatoria

Este trabajo final de grado está dedicado a mis padres Adriana y Oscar quienes son un motor incansable de apoyo en mi crecimiento.



Agradecimientos

A mi familia que me brindó apoyo incondicional en todo momento para concretar este objetivo, desde lo económico hasta lo afectivo. También a mi novia quien dedicó parte de su tiempo a esta meta. Finalmente dar las gracias al Esp. Ing. José María Ducloux, cuyo aporte es parte de este trabajo y a toda la comunidad de profesores del IUA quienes forjaron mi educación durante tantos años.



Resumen

Con la necesidad de poder monitorear y ser consciente de las condiciones atmosféricas presentes para un contexto en particular, en este trabajo final de grado se abordará la interacción de sistemas embebidos, componentes electrónicos, Internet y la exigencia poder cuantificar y calificar las condiciones del aire que nos rodean.

Este proyecto aspira entre sus objetivos principales, cuantificar el grado de polución del aire de una región o ciudad con la finalidad de recolectar y registrar datos para su posterior procesamiento, facilitar el acceso a la información y la toma de decisiones.

Para llevar a cabo esta idea, se diseñará e implementará una solución que permitirá a los usuarios finales acceder a través de un portal web a una gran cantidad de información en tiempo real relacionada con el medio ambiente y las condiciones climáticas circundantes.

Finalmente se mostrarán los resultados obtenidos en base a las pruebas realizadas sobre el prototipo.



Índice General

Resumen	IV
Índice de Figuras	VIII
Índice de Tablas	IX
Palabras Clave	X
1. Introducción	1
2. Marco Teórico	4
2.1. Cambio climático	4
2.2. <i>IoT</i>	6
2.3. Normativa actual	8
3. Estudio Técnico	10
3.1. Hardware	11
3.1.1. Raspberry Pi	11
3.1.2. Arduino	12
3.1.3. Sensores	14
3.2. Software	18
3.2.1. Arduino Software	18
3.2.2. Elasticsearch	19
3.2.3. Grafana	20
3.2.4. <i>Script</i> de Python	21
3.3. Resumen técnico	22
4. Diseño	24
4.1. Interconexión de Hardware	24
4.1.1. <i>Southbound</i>	24
4.1.2. <i>Northbound</i>	27
4.2. Modelado del Software	29
4.2.1. Elasticsearch	29
4.2.2. Grafana	30
4.2.3. Corolario	30
4.3. Parametrización del prototipo	31
4.4. Adquisición de datos	32
4.4.1. Humedad y temperatura	32
4.4.2. Presión atmosférica	33
4.4.3. Polvo ambiental	36
4.4.4. Gases	39



5. Implementación	40
5.1. Ensamble de las partes	40
5.2. Prototipo	43
5.3. Mediciones	45
5.4. <i>Test Cases</i>	47
5.4.1. De conectividad	47
5.4.2. De funcionalidad	51
6. Resultados	52
7. Conclusiones	53
8. Bibliografía	54
Anexos	56



Índice de Figuras

2.1.	Imágen ilustrativa del cambio climático.	5
2.2.	Arquitectura propuesta por IEEE para <i>IoT</i>	6
2.3.	Alcance de posibles sectores interesados en <i>IoT</i> propuestos por IEEE. . .	7
3.1.	Raspberry Pi 3.	11
3.2.	Arduino UNO R3.	12
3.3.	Sensor DHT11.	14
3.4.	Sensor electroquímico del tipo MQ-XX.	14
3.5.	Sensor MQ-XX modular.	15
3.6.	Módulo Sharp GP2Y1010AUf.	16
3.7.	Módulo Bosch BMP180.	17
3.8.	Impresión de pantalla de Arduino IDE.	18
3.9.	Logotipo de Elastic.	19
3.10.	Ejemplo de tableros múltiples en Grafana.	20
3.11.	Esquema de la solución propuesta.	22
4.1.	Diagrama en bloques para la interconexión de hardware.	24
4.2.	Detalle del área lógica <i>Southbound</i>	25
4.3.	Detalle del área lógica <i>Northbound</i>	27
4.4.	<i>timestamp</i> en formato Año-mes-día Hora-minutos-segundos.	29
4.5.	<i>Elasticsearch Head plugin</i>	30
4.6.	Señal que da inicio a la comunicación del sensor DHT11 y el microcontrolador que lo gobierna.	33
4.7.	Proceso de lectura de la presión atmosférica efectuado por el sensor BMP180 de Bosch según especifica el fabricante en su hoja de datos.	35
4.8.	Conexionado tipo para el sensor GP2Y1010 a un microcontrolador.	36
4.9.	Interior del sensor GP2Y1010.	37
4.10.	Interior del dispositivo (a) sin la presencia de contaminantes y (b) con la presencia de contaminantes.	37
4.11.	Respuesta del sensor GP2Y1010, relación entre tensión de salida Voc y la densidad del polvo captado en el ambiente.	38
4.12.	Períodos para la toma de datos correspondientes al sensor GP2Y1010. . .	38
5.1.	Conexión lógica entre el sensor DHT11 y la placa Arduino UNO.	40
5.2.	Conexión lógica entre el sensor BMP180 y la placa Arduino UNO.	41
5.3.	Conexión lógica entre el sensor MQ-135 y la placa Arduino UNO.	41
5.4.	Conexión lógica entre el sensor MQ-7 y la placa Arduino UNO.	41



5.5. Conexión lógica entre el sensor GP2Y1010AUf y la placa Arduino UNO.	42
5.6. Camino lógico de la información a través de Raspberry Pi.	42
5.7. Gabinete cerrado.	43
5.8. Interior del gabinete.	43
5.9. Vista lateral del gabinete.	44
5.10. Vista inferior de los sensores.	44
5.11. Histórico de temperatura promedio.	45
5.12. Histórico de humedad promedio.	45
5.13. Histórico de polvillo promedio disperso en el ambiente.	46
5.14. Histórico de la concentración de gases.	46
5.15. Proceso de sensado para DHT11.	48
5.16. Impresión de pantalla de Grafana, configuración de <i>datasources</i>	50



Índice de Tablas

2.1. Listado de contaminantes y las proporciones estipuladas permitidas por la Ley Nacional 20.284.	8
3.1. Características generales de Raspberry Pi detalladas por su fabricante. . .	12
4.1. Descripción de cada sensor y su relación con la placa Arduino UNO para la interconexión.	26
4.2. Consumo de corriente por cada sensor.	31
4.3. Formato de datos para el sensor DHT11.	32
4.4. Formato de datos para el sensor BMP180 de Bosch.	34



Palabras clave

- *Internet of Things (IoT)*
- Redes de sensores
- *Smart Cities*
- Arduino UNO
- Raspberry Pi
- Sistemas embebidos
- Polución ambiental
- Cambio climático



Capítulo 1

Introducción

El cambio climático es un problema de naturaleza global, cuyos mayores impactos serán en el largo plazo e involucran interacciones entre procesos naturales y procesos sociales, económicos y políticos a escala mundial. Actuar en forma proactiva frente a esto es sin dudas crucial para poder prepararse y aventajar el futuro de la humanidad.

La OMS¹ publica anualmente un modelo² con estimaciones a nivel mundial sobre la exposición a la contaminación del aire y sus repercusiones para la salud, el cual constituye un gran paso hacia la formulación de evaluaciones aún más fiables sobre la enorme carga mundial que representan los más de 6 millones de muertes —una de cada nueve muertes en todo el mundo— provocadas por la exposición a la contaminación del aire de interiores y de exteriores.

En la actualidad, las metrópolis más avanzadas consideran la polución ambiental entre sus prioridades. Visionarios tecnológicos, como el diseñador holandés *Daan Roosegaarde*³, quien se ha comprometido en la cuantificación de la polución a nivel mundial, diseñó un sistema⁴ colosal para capturar el monóxido de carbono excesivo presente en el medio ambiente, entre otros desarrollos sustentables.

Por otro lado, existen organizaciones destinadas a recoger información sobre las condiciones atmosféricas mediante dispositivos compactos comercializados y distribuidos a nivel mundial, tal es el caso de *Smart Citizen*⁵ y *Air Quality Egg*⁶.

¹Organización Mundial de la Salud.

²Referencia:<http://www.who.int/mediacentre/news/releases/2016/air-pollution-estimates/es/>

³Referencia:<https://www.studioroosegaarde.net/info/profile/>

⁴Referencia:<https://www.studioroosegaarde.net/project/smog-free-project/photo/#smog-free-project>

⁵Referencia:<https://smartcitizen.me/kits/>

⁶Referencia:<http://airqualityegg.com>



A nivel nacional, investigadores de la Universidad Tecnológica Nacional de Mendoza desarrollaron un equipo que determina la concentración de contaminantes presentes en la atmósfera denominado “Monitoreo Abierto de la Calidad del Aire”, (*MACA*)⁷. Este dispositivo almacena los datos en una tarjeta de memoria y cuenta con un panel solar y una batería, lo que permite que sea instalado incluso en lugares remotos. El proyecto *MACA* fue presentado a un grupo de científicos denominado *Agencia CTyS*, Ciencia Tecnología y Sociedad, pertenecientes a la Universidad Nacional de La Matanza, *UNLaM*, como una propuesta⁸ de innovación agraria en el sur de la región Cuyana.

Por otra parte, la *IEEE*⁹ es una organización de ingeniería global y profesional cuya misión es fomentar la innovación tecnológica y la excelencia en beneficio de la humanidad.

En un intento de plantear una definición general y útil para la comunidad científica, esta organización creó un documento llamado “*Towards a Definition of the Internet of Things (IoT)*”¹⁰, «“Hacia la definición de Internet de las Cosas”» publicado el 27 de Mayo de 2015, donde normaliza la estructura de este nuevo concepto.

IoT es un concepto que se refiere a la interconexión digital de objetos cotidianos con Internet. Alternativamente, *IoT* es el punto en el tiempo en el que se conectarían a internet más “cosas u objetos” que personas.

La atmósfera terrestre protege la vida de la Tierra, absorbiendo en la capa de ozono parte de la radiación solar ultravioleta, y reduciendo las diferencias de temperatura entre el día y la noche. El dióxido de carbono participa en procesos muy importantes para el desarrollo de la vida en la tierra, pero el exceso del mismo puede degradar la calidad del aire. Para poder cuantificar estas magnitudes, dispositivos electrónicos y sistemas embebidos pueden ser utilizados para obtener información muy valiosa, pero esta información luego debe ser debidamente procesada e interpretada. El concepto de *IoT* apunta a satisfacer la necesidad de capturar, procesar, exponer y compartir esa valiosa información con el resto de la comunidad a través de Internet.

El propósito de este trabajo final de grado es diseñar e implementar un sistema capaz de cuantificar el grado de polución del aire de una región o ciudad con la finalidad de recolectar y registrar datos para su posterior procesamiento, facilitar el acceso a la información y la toma de decisiones. Para el desarrollo del mismo, se utiliza una gran variedad de sensores y diferentes plataformas de sistemas embebidos conectados a un servidor central mediante Internet. Diferentes usuarios podrán acceder al servidor mediante una página web y/o una aplicación móvil para conocer los parámetros de la calidad del aire de su región o ciudad. Bajo los conceptos aplicados de *IoT*, se pretende proporcionar un servicio capaz de generar conciencia en la población sobre la contaminación del aire y facilitar la toma de decisiones a entidades gubernamentales. Entre los objetivos principales del proyecto se destacan los siguientes:

⁷Referencia:<http://monitorabierto.wikidot.com>

⁸Referencia:<http://www.ctys.com.ar/index.php?idPage=20&idArticulo=2832>

⁹Sitio web: <https://www.ieee.org/index.html>

¹⁰Referencia: Documento adjunto en Anexo.



- Medir la calidad del aire con sensores afines para obtener diferentes grados de presencia de contaminantes.
- Crear una red de sensores conectados a Internet para la obtención de muestras en diferentes lugares de una región o ciudad.
- Diseñar e implementar un sistema embebido para adquisición y envío de datos a través de Internet.
- Implementar un servidor conectado a Internet que permita el acceso a los datos y realice el procesamiento de los mismos.

El resto del documento se organiza de la siguiente manera. En el capítulo 2 se incluyen los aspectos teóricos sobre el cambio climático, la *IoT* y la normativa vigente sobre la calidad del aire en Argentina. Un estudio técnico sobre las plataformas de hardware y de software utilizadas en el proyecto puede encontrarse en el capítulo 3. El capítulo 4 se destina al diseño del hardware para adquisición y envío de datos como del software del lado servidor. La implementación se muestra del sistema completo se describe en el capítulo 5. Los resultados obtenidos se discuten en el capítulo 6 y, por último, en las conclusiones se comentan los logros, posibles mejoras y trabajo a futuro.



Capítulo 2

Marco Teórico

2.1. Cambio climático

Junto a GREENPEACE¹, quienes declaran abiertamente en una de sus campañas² lo siguiente,

“Desde la revolución industrial hasta hoy, la quema de combustibles fósiles (petróleo, carbón y gas), que se usan para producir energía, libera gases de efecto invernadero (CO₂) a la atmósfera, aumentando la temperatura de la Tierra y provocando una distorsión en el sistema climático global.”

*National Geographic Society*³ es una de las organizaciones internacionales más grandes del mundo inspiradas en la educación y la ciencia. Recientemente investigadores de esta organización han doblado sus esfuerzos para determinar con mayor precisión el impacto del cambio climático en el planeta tierra.

El último gran aporte en éste tópico realizado por esta agrupación, denominado *“Before the Flood”*, lo que se traduce como «“Antes de la inundación”», profundiza sobre la importancia que tiene la contaminación ambiental en nuestro planeta y, mediante la concientización, prevenir la destrucción de la tierra.

A continuación se observa a modo ilustrativo, en la Figura 2.1, el impacto visual que tiene sobre la población el calentamiento global.

¹<http://www.greenpeace.org/international/en/>

²Referencia: <http://www.greenpeace.org/argentina/es/campanas/cambio-climatico/>

³Comúnmente denominado “NatGeo”, <http://www.nationalgeographic.com>



Figura 2.1: Imágen ilustrativa del cambio climático.

En un informe publicado por el Grupo Intergubernamental de Expertos sobre el Cambio Climático⁴ (*IPCC*), se indica que el calentamiento del sistema climático es inequívoco y que en su mayor parte se debe, muy probablemente, al aumento de las concentraciones de gases de efecto invernadero provocado por actividades humanas como el uso extendido e indiscriminado de combustibles fósiles -el petróleo, el gas o el carbón-, la descomposición de residuos urbanos o ganaderos y los cambios en el uso de la tierra como consecuencia.

Lamentablemente en las grandes metrópolis de la República Argentina, tal es el caso de Buenos Aires, Córdoba, Rosario, etc., las entidades gubernamentales no cuentan con una herramienta que les permitan poder cuantificar y calificar la calidad del aire que los rodea. La necesidad de poder tener esa información y valerse de la misma para tomar decisiones para el futuro, a corto y largo plazo, es de trascendental.

⁴<https://www.ipcc.ch>

2.2. *IoT*

La asociación IEEE define a la tecnología *IoT* como “Una red de dispositivos -con sensores- que están conectados a Internet”, si bien no es una especificación oficial, denota la intencionalidad de esta organización de estudiar esta nueva tecnología desde su aspecto físico.

La IEEE tiene una cartera de más de 900 estándares activos y más de 500 en desarrollo. En su investigación sobre *IoT*, ha identificado más de 140 normas y proyectos existentes que son relevantes para el *IoT*, los cuales están disponibles para la comunidad a través de su sitio web⁵.

En su documento “*Towards a Definition of the Internet of Things (IoT)*”, esta comunidad presenta un diagrama en bloques simplificado con las intenciones de definir la arquitectura de *IoT*, como se observa en la Figura 2.2.

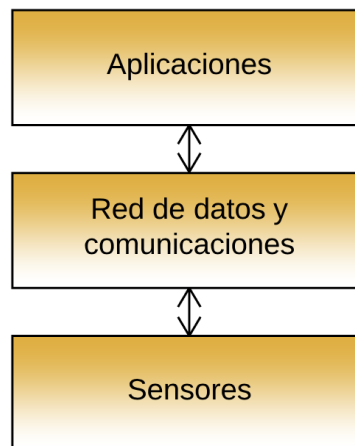


Figura 2.2: Arquitectura propuesta por IEEE para *IoT*.

⁵<http://standards.ieee.org/innovate/iot/>

IEEE también postula una catalogación teniendo en cuenta cuales son las potenciales partes interesadas en extraer provecho de esta nueva tecnología. A continuación se observa en la Figura 2.3 el esquema propuesto por esta organización.

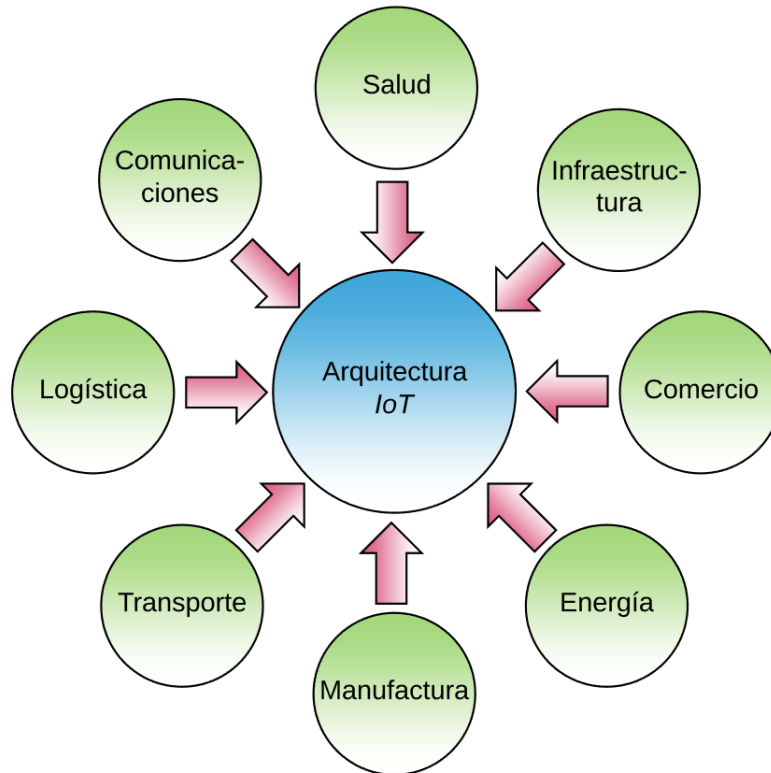


Figura 2.3: Alcance de posibles sectores interesados en *IoT* propuestos por IEEE.

Como se observa en la Figura 2.3, el alcance de esta nueva tecnología es tan amplia como la necesidad del ser humano en poder percibir y captar información del medio y, en base a ésta, interactuar con su entorno.

Desde la grandes industrias productoras de energía hasta el sector de la salud pública, el abanico de posibilidades en donde *IoT* puede ser útil abarca un gran espectro de necesidades. Éstas obligaciones podrían ir desde lo más básico como por ejemplo averiguar la ubicación de una maquinaria en un determinado momento, hasta lo más peculiar como conocer el estado y la vida útil de una batería de un paciente cardíaco cuyo marcapasos esté vinculado a una red de *IoT*.



2.3. Normativa actual

La autoridad Sanitaria Nacional de la República Argentina queda facultada para fijar las normas de calidad de aire y las concentraciones de contaminantes correspondientes a 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.

A continuación en la Tabla 2.1, se observa un extracto de dicha ley.

Contaminante [unidad]	Norma calidad del aire
CO ppm	10 ppm en 8 hs 50 ppm en 1 hs
NO _x ppm	0,45 ppm en 1 hs
SO ₂ ppm	0,03 ppm promedio mensual
O ₃ ppm	0,1 ppm en 1 hs
Partículas en suspensión	150 mg/m ³ promedio mensual
Partículas sedimentares	1 mg/cm ² cada 30 días

Tabla 2.1: Listado de contaminantes y las proporciones estipuladas permitidas por la Ley Nacional 20.284.

Los procesos involucrados en las mediciones de estos parámetros datan de la misma época que la ley, entre estos se destacan dispositivos tales como el “Analizador infrarrojo modificado” de *Jaccobs, M. B.* y/o métodos como “Determinación colorimétrica del NO_z en la atmósfera” asignado a *Griess-Saltzman*.

Finalmente las definiciones planteadas en esta ley permiten modelar el problema de la contaminación ambiental. Entre estos dictámenes se pueden destacar los siguientes:

- *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 los 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.



Capítulo 3

Estudio Técnico

Junto con la medición en la calidad del aire, el sistema será capaz de comunicarse a través de Internet para que la información sea monitoreada desde una estación remota en tiempo real. Este sistema entonces permitirá obtener en tiempo real parámetros específicos del aire lo cual contribuye a la caracterización del mismo en sectores donde sea crítica su medición.

Este proyecto se compone de dos partes principales las cuales se integran para cumplir con los objetivos propuestos: el hardware o *end-devices*¹ y el software.

Respecto a la primera, para poder llevar a cabo este trabajo se utilizaron dos dispositivos basados en microprocesador (μP) y en microcontrolador (μC) como es el caso de las placas de desarrollo Raspberry Pi y Arduino UNO R3, respectivamente. Junto a las placas de desarrollo se utilizaron sensores compatibles con la finalidad de poder capturar variables físicas del entorno que los rodea como, por ejemplo, valores de temperatura, humedad relativa, polvo disperso en el aire, concentración de dióxido de carbono, monóxido de carbono, alcoholes, etc.

Por otra parte, en lo que respecta al software, existe una segunda catalogación. Por un lado se utilizó el programa denominado Arduino Software (IDE), el cual se encarga de facilitarle al programador un entorno de desarrollo y programación de la plataforma Arduino; y, por otro lado, se hizo uso de herramientas diseñadas para administrar datos, tanto para almacenar como para mostrar los históricos, como es el caso de Elasticsearch y de Grafana, respectivamente.

¹*end-devices* - dispositivos terminales, en ingeniería se le denomina comúnmente de esta manera a los equipos involucrados en una solución tecnológica.

3.1. Hardware

3.1.1. Raspberry Pi

Raspberry Pi es un ordenador de placa reducida u ordenador de placa simple (*SBC*, *single board computer*) de bajo costo desarrollado en Reino Unido por la Fundación Raspberry Pi, con el objetivo de promover las ciencias de la computación en forma masiva.

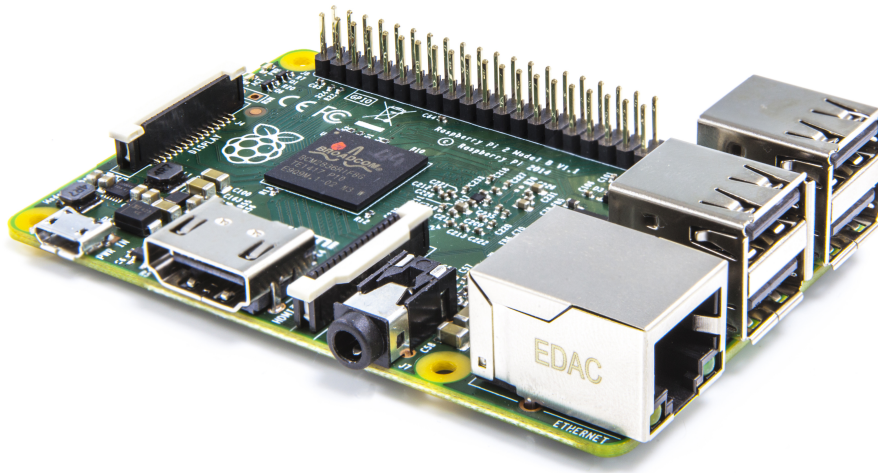


Figura 3.1: Raspberry Pi 3.

Básicamente es una computadora basada en una placa de circuito impreso equipada con un procesador ARM, sistema de video integrado HDMI, conexión Ethernet y puerto USB para conectar periféricos y monitores.

Para este proyecto se utilizó la 3^{ra} versión de Raspberry Pi, como puede observarse en la Figura 3.1, la cual, además de lo anteriormente mencionado, cuenta con conectividad Wi-Fi y Bluetooth.

El sistema operativo que se ejecuta en esta computadora es Raspbian², el cual está basado la distribución de Linux denominada Debian.

²Raspbian es un sistema operativo libre basado en Debian optimizado para el hardware de Raspberry Pi. Website - <https://www.raspbian.org>

A continuación se observa, en la Tabla 3.1, las características del modelo Raspberry Pi 3.

Fecha de lanzamiento	29/2/2016
Soc	BCM2837
CPU	Quad Cortex A53 @ 1.2 GHz
Arquitectura	ARMv8-A
GPU	400 MHz VideoCore IV
RAM	1 GB SDRAM
Almacenamiento	Micro-SD
Ethernet	10/100
Wireless	802.11n / Bluetooth 4.0
Video	HDMI / RCA compuesto
Audio	HDMI / 3,5 mm
GPIO	40
Precio	u\$d 35

Tabla 3.1: Características generales de Raspberry Pi detalladas por su fabricante.

3.1.2. Arduino

Arduino es una plataforma de prototipos electrónicos de código abierto (*open-source*) basada en hardware y software flexibles y fáciles de usar. Está pensado para artistas, diseñadores, como hobby y para cualquiera interesado en crear objetos o entornos interactivos.



Figura 3.2: Arduino UNO R3.



El hardware de Arduino puede capturar o adquirir señales del entorno mediante sus entradas desde una variedad de sensores, y puede afectar a su alrededor mediante sus salidas, por ejemplo, controlar luces, motores y otros artefactos.

Para poder desarrollar aplicaciones y programar el microcontrolador de la placa se utiliza el Arduino Software (IDE), el cual provee el compilador, librerías, ejemplos y drivers necesarios para correr aplicaciones de usuario.

En este proyecto se utilizó la versión de Arduino UNO R3 la cual se observa en la Figura 3.2.

3.1.3. Sensores

Humedad y temperatura

El módulo DHT11, que se observa en la Figura 3.3, es un sensor de humedad y temperatura, con salida digital, sencillo y de muy bajo costo. Internamente utiliza un sensor capacitivo de humedad y un termistor para obtener los valores de temperatura del ambiente que lo rodea, para luego suministrar los resultados a través de su pin digital de salida mediante una transmisión serial.

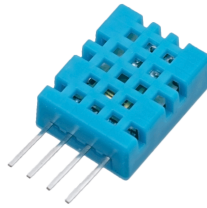


Figura 3.3: Sensor DHT11.

Es relativamente simple de usar, y debido a la respuesta interna de sus componentes debe tenerse en cuenta que el tiempo mínimo entre lecturas no puede ser inferior a 2 segundos para que los resultados obtenidos sean los correctos.

Sensores del tipo MQ-XX

Estos tipos de dispositivos son sensores electroquímicos y varían su resistencia cuando se exponen a determinados gases. Internamente poseen un calentador encargado de aumentar la temperatura interna y con esto el sensor pueda reaccionar con los gases provocando un cambio en el valor de la resistencia.

Frente a la aparición de ciertos gases en el ambiente que los rodea, la salida analógica de este tipo de dispositivos se ve alterada provocando diferentes valores de tensión, los cuales son convertidos a impulsos digitales tras pasar por la placa Arduino UNO.



Figura 3.4: Sensor electroquímico del tipo MQ-XX.

Debido al calentador interno al sensor es necesario esperar un tiempo de calentamiento para que la salida sea estable y tenga las características que el fabricante muestra en su datasheet. Dicho tiempo, dependiendo del modelo puede estar comprendido entre 12 y 48 horas.

Generalmente, los sensores MQ de la Figura 3.4 se encuentran disponibles en el mercado en módulos como los mostrados en la Figura 3.5, lo que simplifica el hecho de tener que conectarlo al microcontrolador, facilitando así su uso. Sólo basta con alimentar el módulo y empezar a leer el sensor. Estos módulos también tienen una salida digital la cual internamente trabaja con un comparador y con la ayuda de un potenciómetro se puede calibrar el umbral y así poder interpretar la salida digital como presencia o ausencia del gas medido.

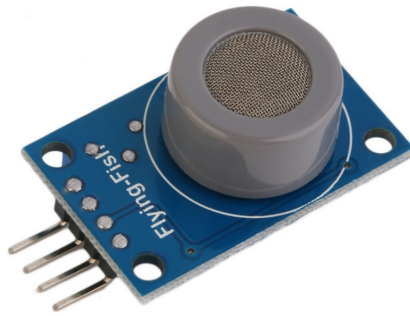


Figura 3.5: Sensor MQ-XX modular.

Sensor de polvo Sharp

El sensor GP2Y1010AU0F de la marca Sharp está especialmente diseñado para detectar partículas muy finas de polvo en el aire y es muy utilizado como detector en purificadores de aire.

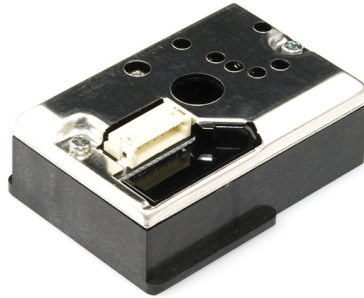


Figura 3.6: Módulo Sharp GP2Y1010AUf.

En la Figura 3.6 se puede observar el módulo, el cual internamente contiene un diodo LED y un fototransistor colocados de tal forma que son capaces de detectar las reflexiones de luz sobre las partículas de polvo.

Es un dispositivo que consume muy poco en lo que respecta a corriente, alrededor de 20 [mA] y puede ser alimentado hasta con 7 [V] DC. La salida analógica de este sensor 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/m³].

El sensor tiene un conector JST³ de 6 pines y se entrega con su cable correspondiente. Para conectarlo con una placa Arduino UNO, se debe añadir una resistencia de 150 [Ω] y un capacitor de 220 [μ F].

³JST es un conector eléctrico manufacturado por *J.S.T. Mfg. Co. (Japan Solderless Terminal)*

Presión atmosférica absoluta y relativa

El sensor BMP180 de la marca Bosch está especialmente diseñado para entregar a través de su salida digital los valores de presión atmosférica absoluta y relativa, así como la temperatura del medio en el cual está inmerso.

Debido a su bajo costo y dimensiones diminutas es de gran utilidad para sistemas de navegación moderna como son los UAVs, drones, etc.



Figura 3.7: Módulo Bosch BMP180.

En la Figura 3.7 se observa el sensor BMP180 integrado a un módulo facilitando así su interconexión con un microcontrolador.

El consumo de corriente de este sensor es extremadamente bajo, alrededor de $3 [\mu A]$, adecuado para sistemas alimentados por baterías como smartphones, sistemas de navegación, predicción del clima, etc.

3.2. Software

3.2.1. Arduino Software

Arduino Software (IDE) contiene un editor de texto para escribir código, un área de mensajes, una consola de texto, una barra de herramientas con botones para las funciones comunes y una serie de menús como se observa en la Figura 3.8. Es a través de este software que el desarrollador puede diseñar, probar, compilar para así luego cargar sus programas llamados “*sketches*” a la placa Arduino UNO.

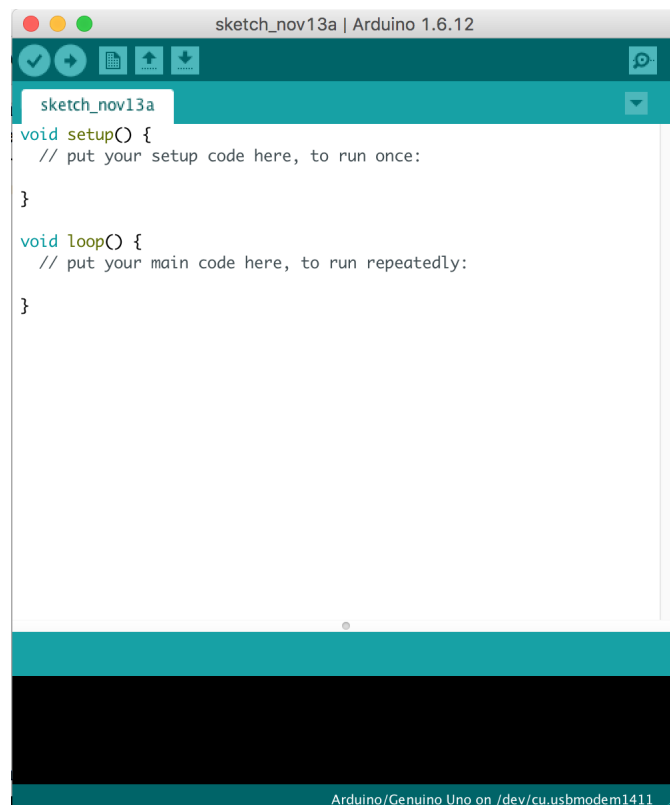


Figura 3.8: Impresión de pantalla de Arduino IDE.

Arduino Software es una aplicación *cross-platform*, lo que significa que puede ser instalado y utilizado en Linux, Windows, Mac y en dispositivos con arquitectura ARM (experimental).

3.2.2. Elasticsearch

Elasticsearch es un servidor de búsqueda basado en Lucene⁴. Provee un motor de búsqueda de texto completo, distribuido y con capacidad de multi-tenencia con una interfaz web RESTful y con documentos JSON. Elasticsearch está desarrollado en Java y está publicado como código abierto bajo las condiciones de la licencia Apache. A continuación se puede observar el logo de la Elasticsearch en la Figura 3.9.



Figura 3.9: Logotipo de Elastic.

Elasticsearch es una potente herramienta que permite indexar un gran volumen de datos y posteriormente hacer consultas sobre estos en forma ordenada. Permite entonces almacenar documentos (estructurados o no) e indexar todos los campos de estos casi en tiempo real, a diferencia de otros sistemas parecidos, no necesita declarar un esquema para la información que se añadirá al mismo a posterior.

⁴Apache LuceneTM es una librería de alta performance, diseñada completamente para motores de búsqueda de texto escrita completamente en Java. Es prácticamente adaptable a cualquier aplicación que precise realizar búsquedas de texto completo, especialmente para proyectos *cross-platform*

3.2.3. Grafana

Grafana es una herramienta de código abierto diseñada para el análisis de métricas y visualización de las mismas en forma gráfica. Se utiliza más comúnmente para la visualización de los datos en series de tiempo, infraestructura y aplicaciones analíticas. En la actualidad se destina a otros dominios que incluyen sensores industriales, automatización del hogar, el clima y el control de procesos.

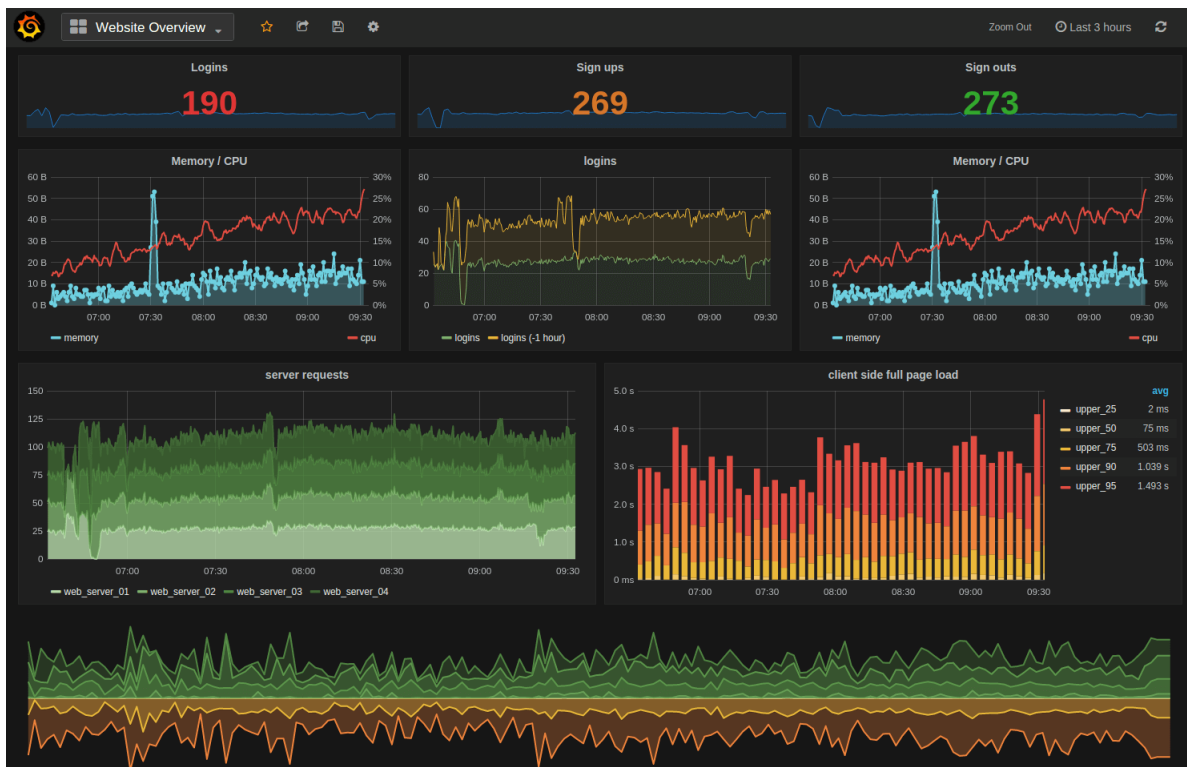


Figura 3.10: Ejemplo de tableros múltiples en Grafana.

Claramente el potencial de Grafana se encuentra en sus tableros llamados *dashboards* como se puede apreciar en la Figura 3.10. Estos tableros están compuestos de paneles individuales ordenados en filas, dedicados a mostrar en forma minuciosa toda la información proveniente de la base de datos asociada.



3.2.4. *Script* de Python

Por último es importante mencionar que para este proyecto se desarrolló en su totalidad, en el lenguaje de programación Python, un script que les permite a Elasticsearch y Grafana poder interactuar con el hardware anteriormente mencionado, tal es el caso de la placa Arduino UNO y Raspberry Pi.

Este *script*⁵ se ejecuta dentro de la placa de desarrollo Raspberry Pi en forma periódica con instrucciones de leer el puerto USB por el cual llegan los datos provenientes de la placa Arduino UNO. Una vez que la información está presente, el mismo *script* se encarga de crear los índices en Elasticsearch y depositar los datos en los mismos.

⁵*Script* - Programas escritos para un entorno especial que automatizan la ejecución de tareas que, alternativamente, podrían ser ejecutados en forma manual por un operador.

3.3. Resumen técnico

Como puede apreciarse, en la Figura 3.11, el sistema está compuesto por dispositivos terminales o *end-devices* una conexión a Internet y los usuarios finales. Los *end-devices* son los encargados de capturar los datos provenientes de los sensores de contaminación del aire, humedad y temperatura para luego ser enviados a un servidor Web a través de Internet.

La interfaz de acceso a la red se realizó por en forma inalámbrica, valiéndose de la conectividad Wi-Fi ofrecida por Raspberry Pi. El servidor es el encargado de recibir, almacenar y procesar los datos de los *end-devices*, y permite el acceso de usuarios o clientes a la información a través de una página web.

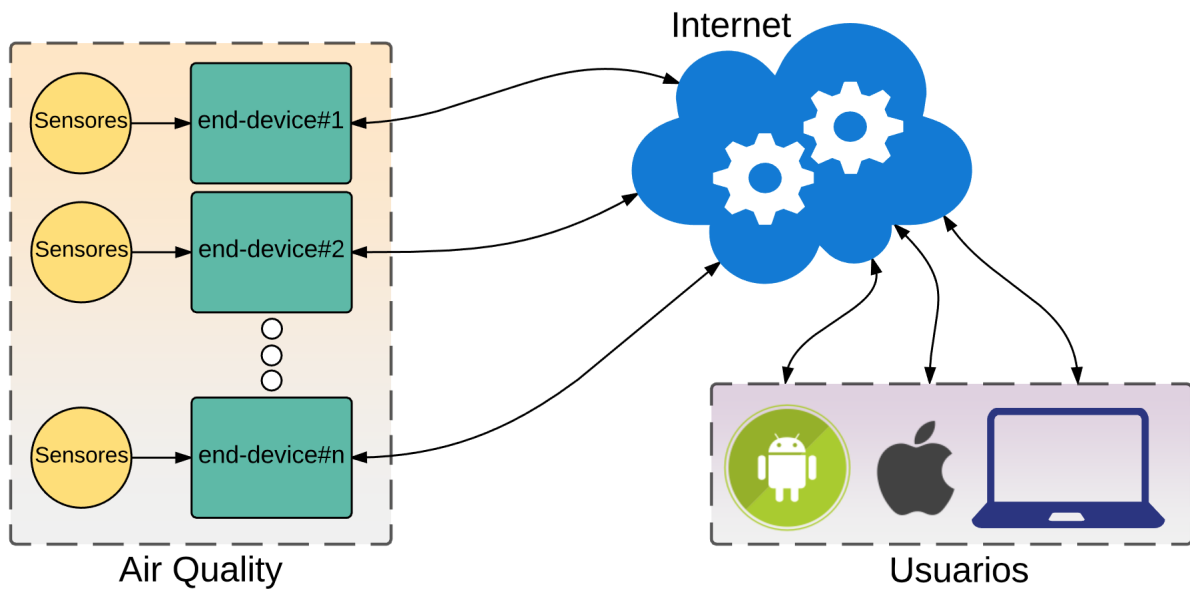


Figura 3.11: Esquema de la solución propuesta.



En este trabajo se diseñaron e implementaron los *end-devices* y el servidor central con tecnologías basadas en sistemas embebidos. Para ambos casos se utilizaron placas de desarrollo basadas en microcontrolador y microprocesador.

El software de aplicación que se ejecuta en los *end-devices* fue desarrollado en su totalidad. Del mismo modo, el software de aplicación del servidor fue desarrollado en este trabajo.

Como se mencionó en puntos anteriores, el proyecto se basó en el diseño y la implementación de un sistema capaz de cuantificar la calidad del aire en puntos determinados. Esta solución permitirá alertar en forma temprana la presencia de contaminantes o simplemente regular la existencia de los mismos.

Las alertas generadas por el sistema serán enviadas a un servidor central a través de Internet para ser consultadas a posterior según sea necesario, ésta capacidad de comunicación con Internet es clave para los sistema que conforman una red de *IoT*. Al mismo tiempo, una vez que la información se encuentra presente en el servidor, esta podrá ser consultada vía web por terceros quienes, por ejemplo, deseen publicar datos de polución ya sea en su página web o utilizarla como entrada de datos para una aplicación móvil, una persona que desea realizar actividad física podría querer consultar la calidad del aire en dos puntos determinados de la ciudad y en base a esto decidir a cual de ellos acudir, permitirle a entidades gubernamentales tomar desiciones inteligentes para contribuir a la concientización de la población respecto a la contaminación ambiental, entre otros casos de uso de esta valiosa información

Capítulo 4

Diseño

El diseño de este prototipo se puede dividir en dos partes bien diferenciadas. Por un lado se encuentra el hardware, encargado de la adquisición de datos y, por el otro, el software, responsable de almacenar y darle un valor agregado a la información recolectada por el dispositivo al mostrarla en un entorno rico en gráficos y métricas accesible desde Internet.

4.1. Interconexión de Hardware

La Figura 4.1 muestra, desde una perspectiva a gran escala, la interacción física de los componentes principales de este prototipo, además define dos áreas lógicas denominadas *Southbound* y *Northbound*, las cuales serán descritas a continuación.

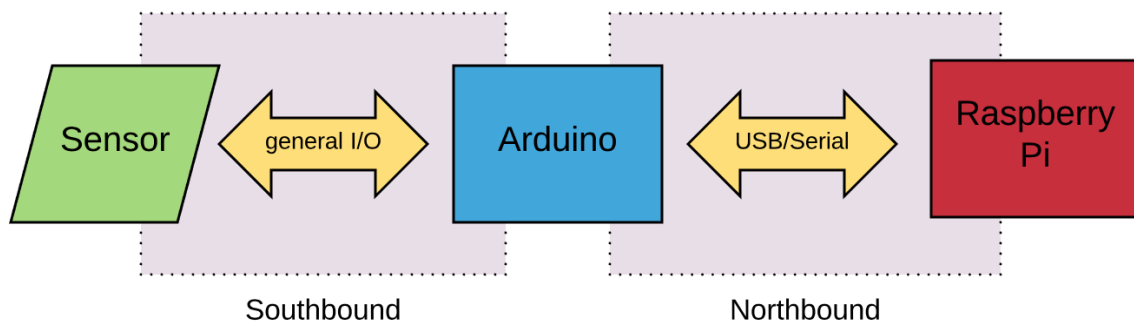


Figura 4.1: Diagrama en bloques para la interconexión de hardware.

4.1.1. *Southbound*

La placa Arduino UNO posee, en forma preestablecida, entradas y salidas, tanto digitales como analógicas, que le sirven al usuario como vía de comunicación con el medio que lo rodea. Haciendo uso de esos puertos, los sensores presentes en este proyecto fueron conectados a la placa de desarrollo y a través de este medio es que la información es capturada por el microcontrolador ATmega328 de la placa Arduino UNO, como se observa en la Figura 4.2.

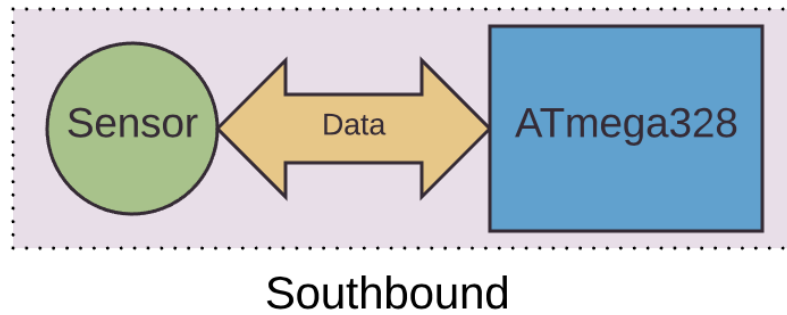


Figura 4.2: Detalle del área lógica *Southbound*.

Dentro del microcontrolador se ejecuta un programa diseñado para leer periódicamente los puertos analógicos y digitales donde se encuentran conectados los sensores, para luego enviar la información por el puerto serie de la placa Arduino UNO hacia *Northbound*.

La frecuencia con la que el programa de la placa Arduino UNO realiza el muestreo de los sensores es de 2 segundos debido a que este tiempo mínimo que le toma a todos los dispositivos tener información precisa de lo que están captando en ese momento.

Es de gran importancia destacar que todas las métricas y/o parámetros capturados por los sensores en este prototipo son variables ambientales que por una materialidad física no se ven alteradas tan rápidamente, con lo cual no es necesario aumentar la velocidad de muestreo de los dispositivos a costas de perder información. Tomando por ejemplo la temperatura, esta tiene una inercia tal que las variaciones se dan en décimas de grado por minuto, situaciones similares se dan con el resto de los parámetros.

Este tiempo está gobernado principalmente por el sensor de temperatura y humedad DHT11 que en su hoja de datos expresa dicha frecuencia de muestreo como el tiempo mínimo estipulado entre lecturas para garantizar la validez de los datos entregados.

En el mismo código escrito para la placa Arduino UNO se estipula que la tasa de transferencia para la comunicación serial en este proyecto fuese preestablecida en 9600 baudios.



Conexiones para *Southbound*

En la Tabla 4.1 se puede observar existe la correspondencia directa entre cada sensor y los puertos de E/S utilizados en la placa Arduino UNO.

Sensor	Parámetro/Métrica	Arduino E/S	Tipo de Señal
DHT11	Humedad y temperatura	D2	Digital
BMP180	Presión atmosférica	A4, A5	Analógica - I2C
GP2Y1010AUf	Polvo ambiental	D12, A3	Analógica
MQ-135	Dióxido de carbono	A2	Analógica
MQ-7	Monóxido de carbono	A0	Analógica

Tabla 4.1: Descripción de cada sensor y su relación con la placa Arduino UNO para la interconexión.

4.1.2. *Northbound*

Para poder establecer una comunicación de datos entre la placa Arduino UNO y Raspberry Pi, se codificó un *script*, desarrollado en su totalidad por el autor, que se encarga de emular un puerto serial en Raspberry Pi a través de la comunicación USB con la placa Arduino UNO, el cual estará ocupando el área lógica de *Northbound*.

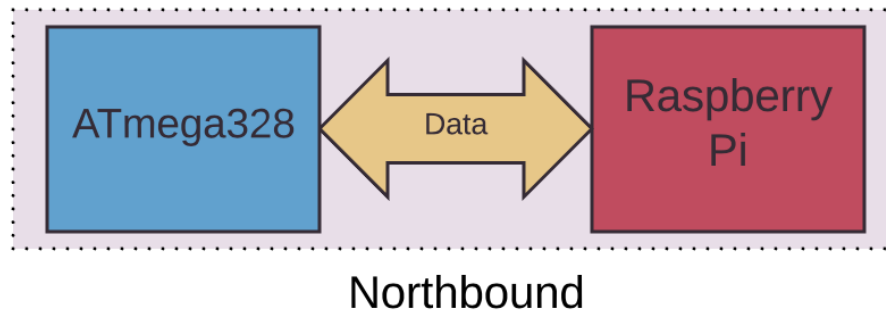


Figura 4.3: Detalle del área lógica *Northbound*.

En la Figura 4.3, la comunicación entre el microcontrolador de la placa Arduino UNO y el microprocesador de Raspberry Pi se da a través de una interfaz física compuesta por conectores USB de tipo-B y tipo-A para la placa Arduino UNO y Raspberry Pi, respectivamente.

El script, codificado en Python¹, se encarga de leer los datos provenientes de la placa Arduino UNO en forma periódica y enviarlos a la base de datos, previamente modelada, denominada Elasticsearch.

Como se observa en la Código 4.1, el script está constantemente esperando la llegada de datos provenientes del puerto USB en la placa Raspberry Pi para agregar dichos valores al índice creado en Elasticsearch con la finalidad de poder analizar la información a posterior.

¹Python - es un lenguaje de programación interpretado cuya filosofía hace hincapié en una sintaxis que favorezca un código legible. Se trata de un lenguaje de programación multiparadigma, ya que soporta orientación a objetos, programación imperativa y, en menor medida, programación funcional. Es un lenguaje interpretado, usa tipado dinámico y es multiplataforma.

Código 4.1: Código de Python para *Northbound*.

```
from elasticsearch import Elasticsearch
from datetime import datetime
from time import sleep
import netifaces
import serial

ser = serial.Serial('/dev/ttyACM0', 9600)
# time_format = "%Y-%m-%d %H-%M-%S"
ip_addr = netifaces.ifaddresses("wlan0")[netifaces.AF_INET][0]["addr"]
es = Elasticsearch([{'host': ip_addr, 'port': 9200}])

index_name = "air_qa"

if not es.indices.exists(index_name):
    request_body = {
        "settings" : {
            "number_of_shards": 1,
            "number_of_replicas": 0
        }
    }
    es.indices.create(index=index_name, body=request_body)

while 1 :
    line = ser.readline()
    try:
        CO2, hum, temp, pressabs, pressrel, dust, CO = line.split(",")
        js = { 'CO2': float(CO2), 'humidity': float(hum), 'temperature': float(temp),
              'pressabs': float(pressabs), 'pressrel': float(pressrel), 'dust': float(dust),
              'CO': float(CO), 'timestamp': datetime.now() }
        res = es.index(index="air_qa", doc_type="record", body=js, request_timeout=30)
    except ValueError:
        print(line)
    #print(res)
    sleep(1)
```

4.2. Modelado del Software

El objetivo de este trabajo es poder otorgarle un valor agregado, a través de *IoT*, a los datos capturados por los sensores mencionados anteriormente. Para ello en este proyecto se utilizaron principalmente dos herramientas web para procesar, almacenar y finalmente mostrar la información capturada, como es el caso de las ya mencionadas soluciones provistas por Elasticsearch y Grafana, para base de datos y *front-end*, respectivamente.

4.2.1. Elasticsearch

Es la base de datos maestra de este proyecto. Elasticsearch está conformada por un índice denominado *air_qa* (por su abreviación en inglés, *air_quality*) que alberga todos los datos de humedad, temperatura, presión atmosférica, polvo ambiental, etc., capturados por los sensores previamente.

Estos índices están ordenados por una estampa de tiempo llamada *timestamp* que le garantiza a todas las herramientas que a posterior consuman estos datos, se aseguren completamente del sincronismo de los mismos. El formato del *timestamp* está bien definido en el *script* de Python anteriormente mencionado. A continuación en la Figura 4.4, se observa con mayor detalle el formato convenido.

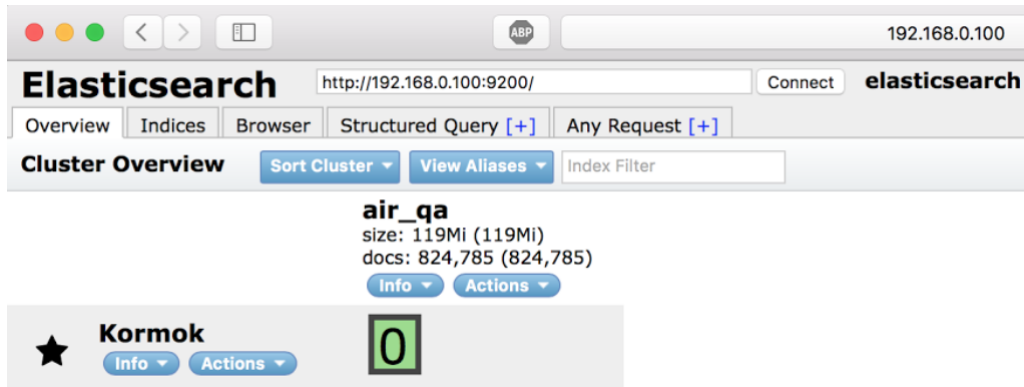
"%Y-%m-%d %H-%M-%S"

Figura 4.4: *timestamp* en formato Año-mes-día Hora-minutos-segundos.

Elasticsearch provee de un portal web para visualizar la información almacenada sin formato, sólo ordenados por *timestamp*, lo cual es de suma importancia para asegurar que la información proveniente de la placa Arduino UNO está siendo correctamente guardada y organizada dentro del índice apropiado.

Existe un *plugin*² provisto por Elasticsearch que ayuda a poder visualizar el contenido de sus índices en forma gráfica y directa a través de un portal web. Este portal denominado *Head* fue instalado junto a la base de datos. A continuación se puede observar en la Figura 4.5 una impresión de pantalla de dicho *plugin*.

²*Plugin* - es un programa que se adhesion a otro con la finalidad de poder ampliar sus capacidades.

Figura 4.5: *Elasticsearch Head plugin.*

4.2.2. Grafana

Para poder observar los datos en forma ordenada y obtener de estos un beneficio, se configuró Grafana como entorno gráfico para la visualización de los mismos. Organizado en dashboards o tableros, Grafana ofrece la posibilidad de organizar y mostrar la información presente en la base de datos con total libertad.

Tomando como *datasource*³ al índice provisto por Elasticsearch, Grafana automáticamente interpreta los tipos de datos, escalas de muestreos, etc. y le facilita al usuario la capacidad de crear gráficos específicos a sus necesidades.

4.2.3. Corolario

Ambas herramientas conforman lo que se conoce como *stack*⁴. Tanto Elasticsearch como Grafana fueron instalados en Raspberry Pi como un servicio o *daemon*⁵ desde el arranque del dispositivo, lo cual lo hace transparente para el usuario final encargado de administrar la solución. Ambos servicios fueron preparados para ejecutarse ordenadamente y de esta manera garantizar el correcto funcionamiento del conjunto.

³*Datasource* se denomina de esta manera a las fuentes de datos de donde Grafana toma la información

⁴En informática, un stack es un tipo de datos abstracto que sirve como una colección de elementos

⁵*Daemon (computing)* Es un programa que se encuentra funcionando en segundo plano en lugar de estar en constante interacción con el usuario.



4.3. Parametrización del prototipo

Para este proyecto casi todos los sensores están alimentados con 5 [V], el único que difiere es el sensor de presión atmosférica modelo BMP180 de la marca Bosch, el cual necesita ser alimentado con 3,3 [V] según indica el fabricante.

En la Tabla 4.2 se observan con mayor detalle los valores de consumo de corriente para cada sensor respetando lo expresado por cada fabricante en la hoja de datos.

Dispositivo	Consumo ⁱ	Alimentación
Raspberry Pi	800 mA	5 V
Arduino UNO r3	50 mA (en reposo)	6-20 V
DHT11	0,5 - 2,5 mA (midiendo) 0,2 - 1 mA (reposo)	5 V
BMP180	1 μ A (una lectura por segundo)	3,3 V
GP2Y1010AUf	11 mA (promedio), 20 mA (max)	5 V
MQ-135	\leq 700 mA	5 V
MQ-7	\leq 800 mA	5 V

ⁱConsumo de corriente estimado por el fabricante en su hoja de datos.

Tabla 4.2: Consumo de corriente por cada sensor.

Para poder calcular el consumo de corriente total de este prototipo, se realizó la estimación bajo el peor de los casos posibles, esto es, con todos los dispositivos involucrados requiriendo el máximo de corriente convenida por su fabricante.

La cantidad de corriente total estipulada para este conjunto de dispositivos, los cuales conforman al prototipo presentado en este proyecto es de 2,3 [A] aproximadamente. Dicha cantidad de corriente será suministrada por una fuente capaz abastecerlos en todo momento.



4.4. Adquisición de datos

El envío de datos desde cada uno de los sensores hacia la placa Arduino UNO dependerá de cada caso, ya sea digital o analógica la información.

4.4.1. Humedad y temperatura

El dispositivo DHT11 capaz de capturar temperatura y humedad del ambiente que lo rodea posee un complejo sensor de temperatura y humedad con una salida provista en formato digital.

Mediante el uso de una técnica exclusiva de adquisición de señal digital y una tecnología de detección de temperatura y humedad, este dispositivo garantiza una alta fiabilidad y una excelente estabilidad a largo plazo.

Este sensor incluye un componente de medición de humedad de tipo resistivo y un componente de medición de temperatura NTC, y se conecta a un microcontrolador de alto rendimiento de 8 bits, que ofrece una excelente calidad, una rápida respuesta e inmunidad a interferencias, lo que lo hace muy rentable por su bajo costo.

Cada componente dentro de este dispositivo está estrictamente calibrado en laboratorio, lo cual lo convierte en una herramienta de precisión para mediciones de humedad y temperatura del ambiente que lo rodea.

Los coeficientes de calibración se almacenan en la memoria OTP⁶, que luego son utilizados en el proceso de detección de la señal interna del sensor.

Proceso de comunicación

Se utiliza un único canal de comunicación entre el microcontrolador de la placa Arduino UNO y el sensor DHT11 para transmitir la información y sincronismo. Un proceso de comunicación completo y satisfactorio se realiza en 4 ms.

La información enviada consiste en una parte entera y otra decimal para cada uno de los parámetros. Una transmisión de datos completa ocupa 40 bits distribuidos como se observa en la Tabla ?? .

Formato de los datos				
8 bits entero H	8 bits decimal H	8 bits entero T	8 bits decimal T	8 bits <i>checksum</i> ⁱ

ⁱEn caso de que la transmisión sea satisfactoria, el *checksum* estará compuesto de los últimos 8 bits de los campos anteriores.

Tabla 4.3: Formato de datos para el sensor DHT11.

⁶OTP - *one-time programmable non-volatile memory*. Memoria de programación única, permanente.

Principio de funcionamiento

Cuando el microcontrolador envía una señal de inicio al sensor, éste cambia su estado de *standby* a *ready* (latente a listo, respectivamente) para comenzar a tomar lecturas y espera que el microcontrolador finalice la señal de inicio. Una vez completada, el sensor envía una respuesta de 40 bits hacia el microcontrolador que incluye la humedad relativa y la temperatura.

Una vez que la información es recolectada por el microcontrolador, el sensor DHT11 vuelve al estado de *standby* hasta tanto no reciba una señal de inicio y dar comienzo nuevamente al proceso de lectura.

A continuación, en la Figura 4.6, se puede observar el proceso completo descrito anteriormente.

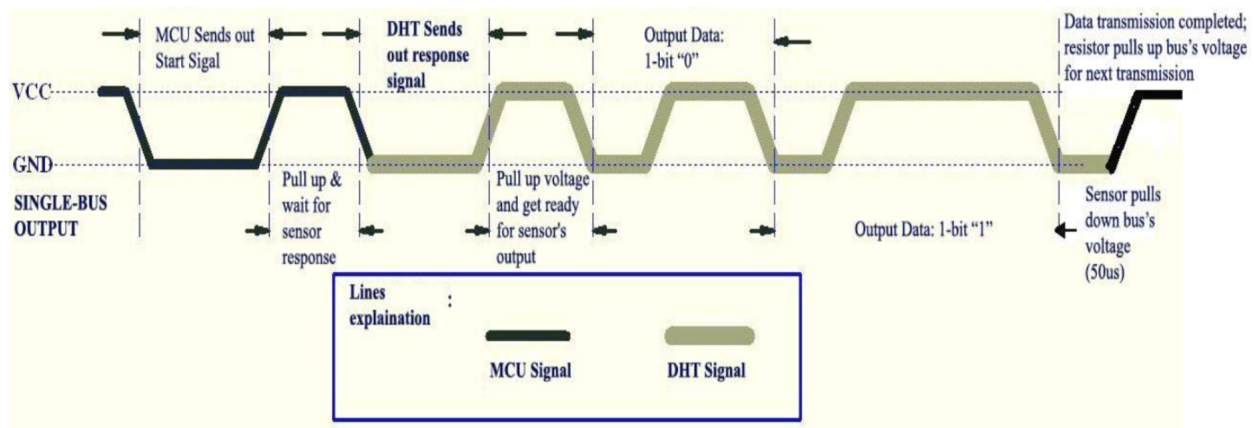


Figura 4.6: Señal que da inicio a la comunicación del sensor DHT11 y el microcontrolador que lo gobierna.

4.4.2. Presión atmosférica

El sensor BMP180 está diseñado para ser conectado directamente a un microcontrolador a través del bus de datos. La información recolectada de presión y temperatura son compensados por los valores de calibración previamente incorporados en la E²PROM⁷ del BMP180.

⁷E²PROM stands for electrically erasable programmable read-only memory and is a type of non-volatile memory Memoria electricamente programable, imborrable, no volátil, de solo lectura.



El BMP180 consiste en un sensor piezo-resistivo, un conversor A/D⁸, una unidad de control con E²PROM y una interfaz I²C⁹. El BMP180 proporciona el valor no compensado de la presión atmosférica relativa al lugar donde se encuentra y la temperatura.

En la E²PROM se han almacenado 176 bits de los datos de calibración del sensor. Esto se utiliza para compensar offset, dependencia de la temperatura y otros parámetros del sensor a la hora de reflejar los datos obtenidos.

En la Tabla 4.4 se puede observar el formato de datos que estipula el fabricante de este sensor para su funcionamiento.

Formato de los datos	
UP = Pressure Data 16 -19 bits	TD = Temperature Data 16 bits

Tabla 4.4: Formato de datos para el sensor BMP180 de Bosch.

Principio de funcionamiento

El microcontrolador envía una secuencia de inicio para dar comienzo a una medición de la presión o la temperatura. Después de un tiempo de conversión, el valor del resultado (UP o UT, respectivamente) se puede leer a través de la interfaz I²C. Para el cálculo de la temperatura en [°C] y la presión en [hPa], los datos de calibración tienen que ser utilizados. Estas constantes se pueden leer desde la E²PROM través de la interfaz I²C en la inicialización del software.

La frecuencia de muestreo se puede aumentar hasta 128 muestras por segundo (modo estándar) para la medición dinámica. En este caso es suficiente para medir la temperatura sólo una vez por segundo y utilizar este valor para todas las mediciones de presión durante el mismo período.

⁸A/D Comúnmente se denomina deéstanamnera al conversor Analógico/Digital.

⁹I²C (del inglés *Inter-Integrated Circuit*) es un bus serie de datos desarrollado en 1982 por Philips Semiconductors. Se utiliza principalmente internamente para la comunicación entre diferentes partes de un circuito, por ejemplo, entre un controlador y circuitos periféricos integrados.

Diagrama de estado para la lectura

A continuación, en la Figura 4.7 se puede observar el diagrama de flujo que rige el funcionamiento de este sensor descrito por el fabricante en su hoja de datos.

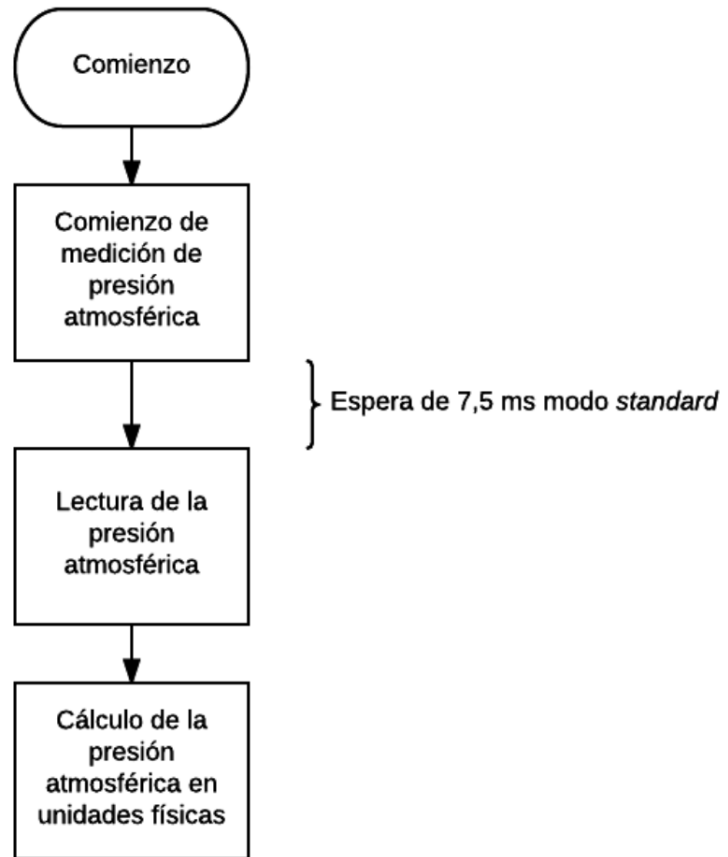


Figura 4.7: Proceso de lectura de la presión atmosférica efectuado por el sensor BMP180 de Bosch según especifica el fabricante en su hoja de datos.

4.4.3. Polvo ambiental

Este sensor de polvo GP2Y1010AU0F es un dispositivo diseñado para detectar el polvo doméstico, humo de cigarrillo, etc. Fue proyectado para ser utilizado en el monitoreo automático de las condiciones atmosféricas para el ambiente que lo rodea en aplicaciones como purificador de aire.

Ejemplo de conectividad a un microcontrolador

En la Figura 4.8 puede observarse la forma que el fabricante recomienda conectar el sensor de polvo modelo GP2Y1010 a un microcontrolador. Para este proyecto se siguió exactamente este ejemplo con la finalidad de respetar las especificaciones.

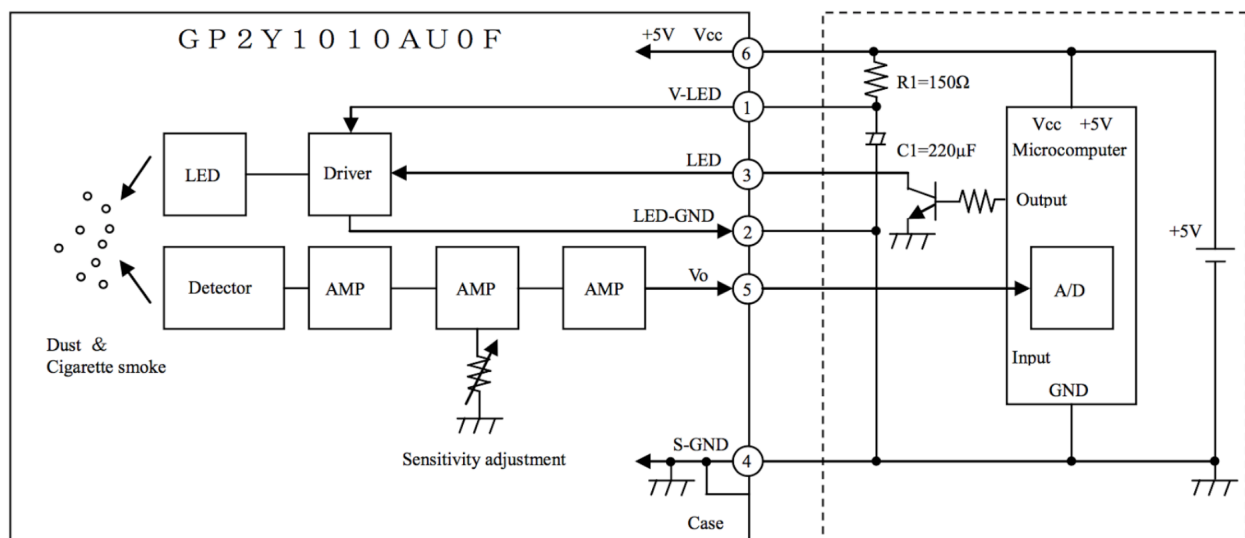


Figura 4.8: Conexión tipo para el sensor GP2Y1010 a un microcontrolador.

Principio de funcionamiento

La luz procedente del LED es visto con una lente y una ranura como se muestra en la Figura 4.9. También para el detector de luz (fotodiodo), una lente y una hendidura se coloca delante de él para cortar luz de perturbación y para detectar el reflejo de luz (al detectar el polvo) de manera eficiente. El área en la que los dos haces de luz cruzan el eje óptico es el área de detección del dispositivo.

La Figura 4.10a muestra lo que sucede en el interior del dispositivo cuando no existe el polvo y la Figura 4.10b muestra cuando existe polvo. El dispositivo genera una salida de tensión, incluso cuando no se está detectando polvo. Esta tensión de salida cuando no se detecta polvo es igual a V_{oc} como se especifica en la hoja de datos.

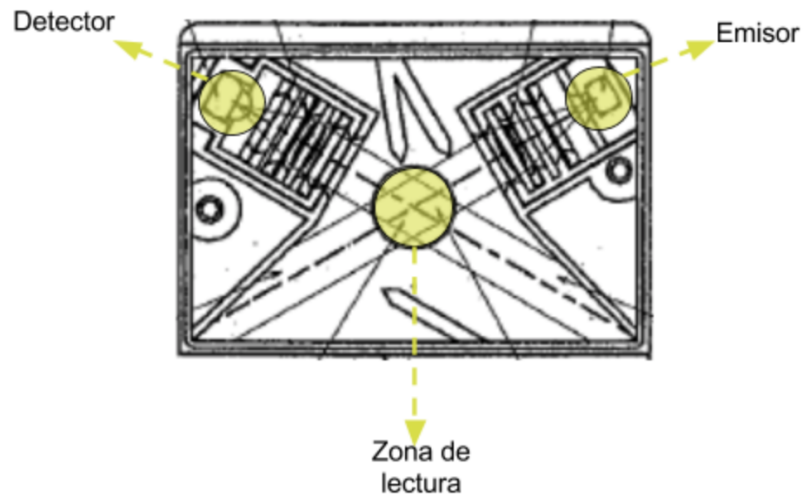


Figura 4.9: Interior del sensor GP2Y1010.

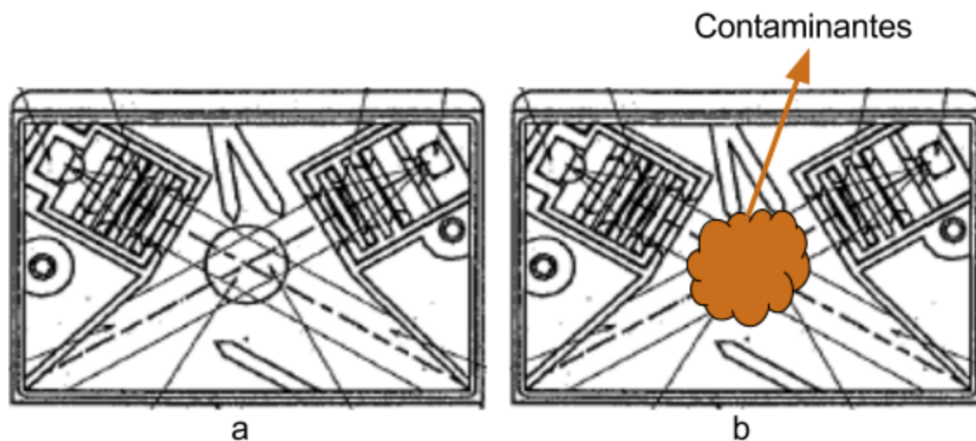


Figura 4.10: Interior del dispositivo (a) sin la presencia de contaminantes y (b) con la presencia de contaminantes.

La Figura 4.10b muestra cómo funciona el dispositivo cuando existe polvo y/o el humo de cigarrillo en el interior del mismo. En este caso, el dispositivo detecta la luz reflejada por el polvo y/o una partícula del humo del cigarrillo. Cuando esto sucede el artefacto actúa en proporción a la cantidad de la luz detectada, entonces, el dispositivo hace que la salida de tensión analógica varíe.

A continuación en la Figura 4.11 se puede observar como el valor de tensión de salida depende directamente de la densidad de las partículas que están siendo captadas.

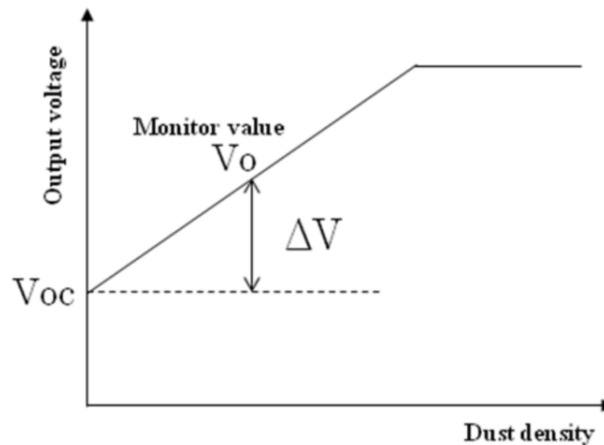


Figura 4.11: Respuesta del sensor GP2Y1010, relación entre tensión de salida V_{oc} y la densidad del polvo captado en el ambiente.

Manejo de la señal de salida

Tiempo de muestreo

El proceso de muestreo llevado a cabo por este sensor transcurre en un periodo de 10 ms como se puede apreciar a continuación en la Figura 4.12.

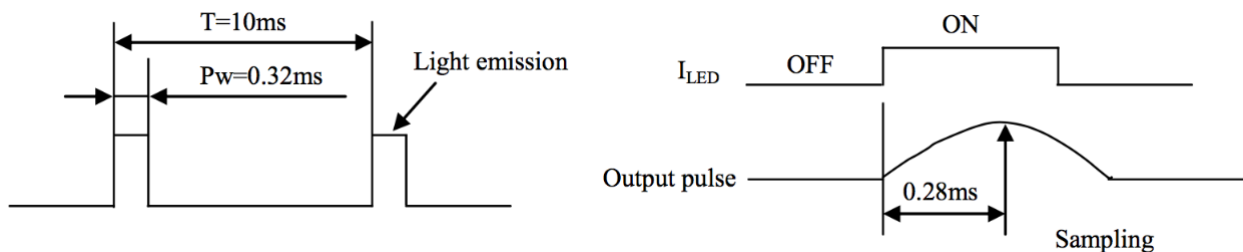


Figura 4.12: Períodos para la toma de datos correspondientes al sensor GP2Y1010.

El tiempo en que el led emite luz para poder capturar la concentración de polvo en el ambiente es de tan solo 0,32 [ms]. Luego el período se completa en 10 [ms], después de transcurrido este tiempo, el dispositivo se encuentra capaz de realizar una nueva medición.



4.4.4. Gases

Los sensores de gases MQ-135 y MQ-7 están especialmente diseñados para capturar la presencia de determinados gases del ambiente que los rodea.

Principio de funcionamiento

Cuando el gas en cuestión es captado por el sensor a través de la parte posterior de la membrana porosa por el electrodo, éste comienza el proceso de sensado. Es aquí cuando se produce una reacción electroquímica que da paso a una corriente eléctrica que circula a través del circuito externo. Además de medir, amplificar y realizar otras funciones de procesamiento de señales, el circuito externo mantiene la tensión en el sensor entre los electrodos de trabajo.

La tensión de salida analógica del sensor varía en presencia del gas para el cual fue construido y calibrado en forma proporcional a la concentración capturada.

Capítulo 5

Implementación

En esta sección se describe el armado del prototipo, los *test cases* realizados sobre el mismo y el código encargado de gobernar al sistema.

5.1. Ensamble de las partes

La placa de desarrollo Arduino UNO posee puertos de E/S, tanto analógicas como digitales previamente establecidos, otorgando al usuario la posibilidad de interactuar con el microcontrolador. Para este trabajo se conectaron los diferentes sensores a través de estos puertos de E/S, utilizando entradas analógicas e interfaces seriales para la adquisición de los datos.

En primera instancia y luego de validar la viabilidad técnica de cada uno de los componentes involucrados se procedió a interconectar el sensor de humedad y temperatura digital DHT11 con la placa de desarrollo Arduino UNO, como se aprecia en la Figura 5.1



Figura 5.1: Conexión lógica entre el sensor DHT11 y la placa Arduino UNO.

Se dispuso de que el sensor DHT11 fuese el primero en formar parte del prototipo ya que tanto la interconexión con el microcontrolador como el desarrollo del programa que gobierna al mismo, se encuentra entre los más simples y directos de implementar.

Junto con el proceso de incorporación del sensor de humedad y temperatura a la placa Arduino UNO se desarrolló e implementó la base de datos y el *front-end* en Raspberry Pi.

Una vez capitalizado el hecho de poder tener los datos provenientes del sensor en el microcontrolador se trabajó en la transmisión de esa información desde la placa Arduino UNO hacia Raspberry Pi mediante un programa construido para tal fin.

Habiendo logrado el camino completo que recorrerá la información entre las variables físicas a medir y el front-end provisto por Grafana, se procedió a incorporar el resto de los sensores bajo la misma filosofía adaptando los programas en *Southbound* y *Northbound* para que se adecuen a los nuevos sensores.

En las Figuras 5.2-5.5 se puede observar en forma aislada la relación lógica que existe entre cada uno de los sensores y la placa de desarrollo Arduino UNO en lo que conforma el *Southbound* del proyecto.

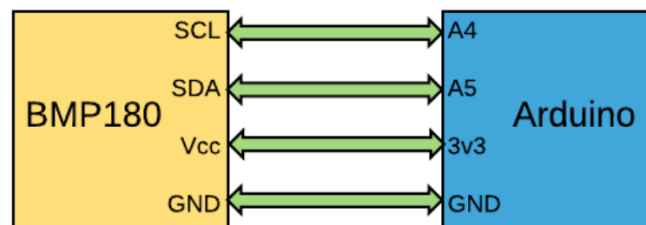


Figura 5.2: Conexión lógica entre el sensor BMP180 y la placa Arduino UNO.

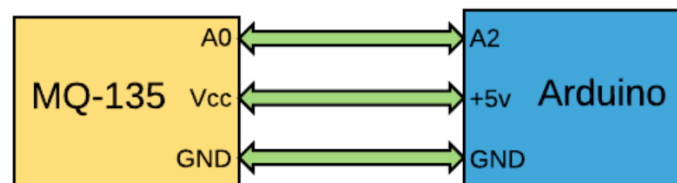


Figura 5.3: Conexión lógica entre el sensor MQ-135 y la placa Arduino UNO.

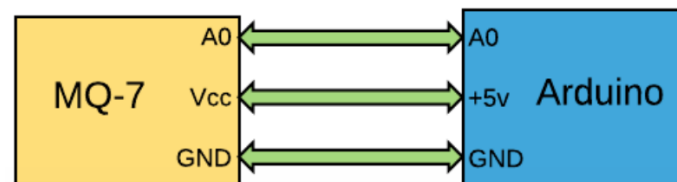


Figura 5.4: Conexión lógica entre el sensor MQ-7 y la placa Arduino UNO.

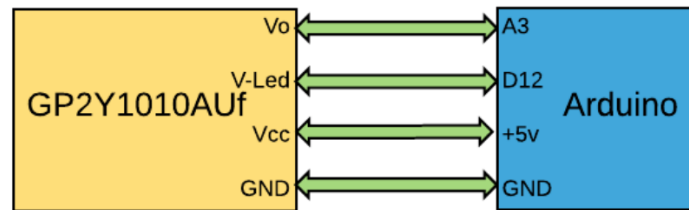


Figura 5.5: Conexión lógica entre el sensor GP2Y1010AUf y la placa Arduino UNO.

Una vez evaluados por separado cada uno de los sensores con la placa Arduino UNO en forma aislada, se procedió a la integración del conjunto.

En este punto del proceso se cuenta con todos los sensores conectados a la placa de desarrollo Arduino UNO conformando de esta manera el área lógica *Southbound*.

En la Figura 5.6 se puede apreciar el camino lógico que recorre la información adquirida por los sensores hasta llegar al *front-end* desde una vista general de alto nivel.

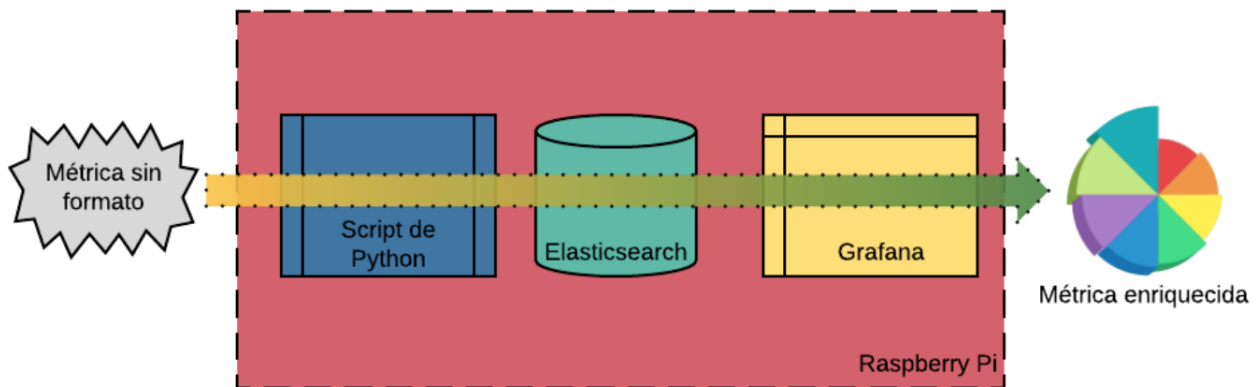


Figura 5.6: Camino lógico de la información a través de Raspberry Pi.

5.2. Prototipo

La confección de este dispositivo fue llevada a cabo en su totalidad. Mediante la utilización de un gabinete estanco, se ubicaron todos los componentes electrónicos dentro del mismo para salvaguardar su vida útil. A continuación, se observan en las Fig 5.7-5.10 detalles del prototipo en funcionamiento.



Figura 5.7: Gabinete cerrado.

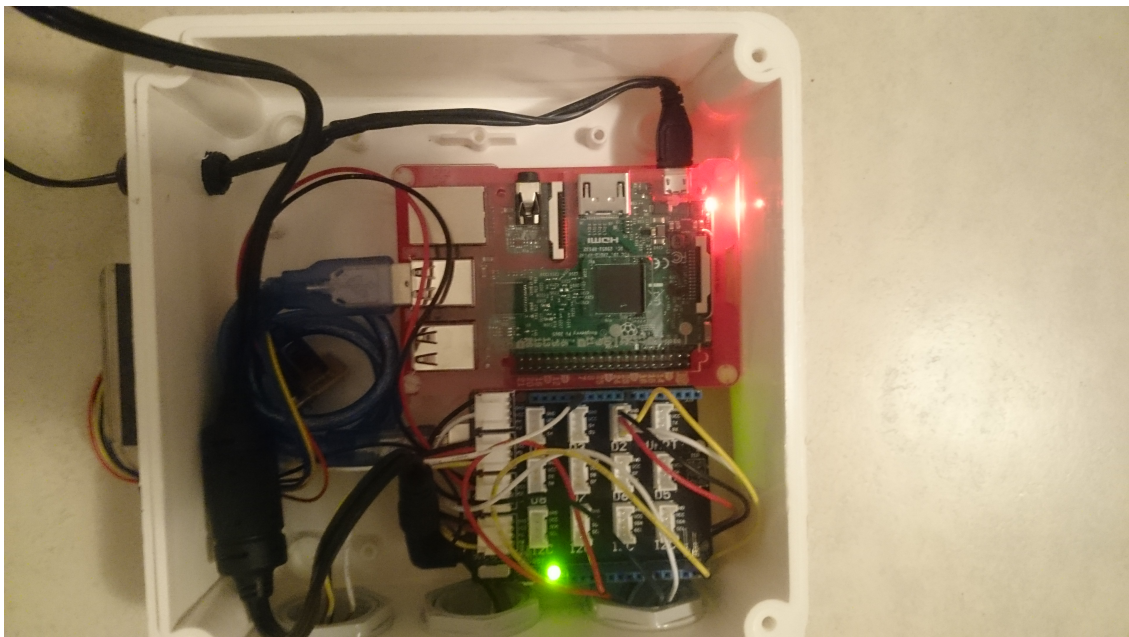


Figura 5.8: Interior del gabinete.

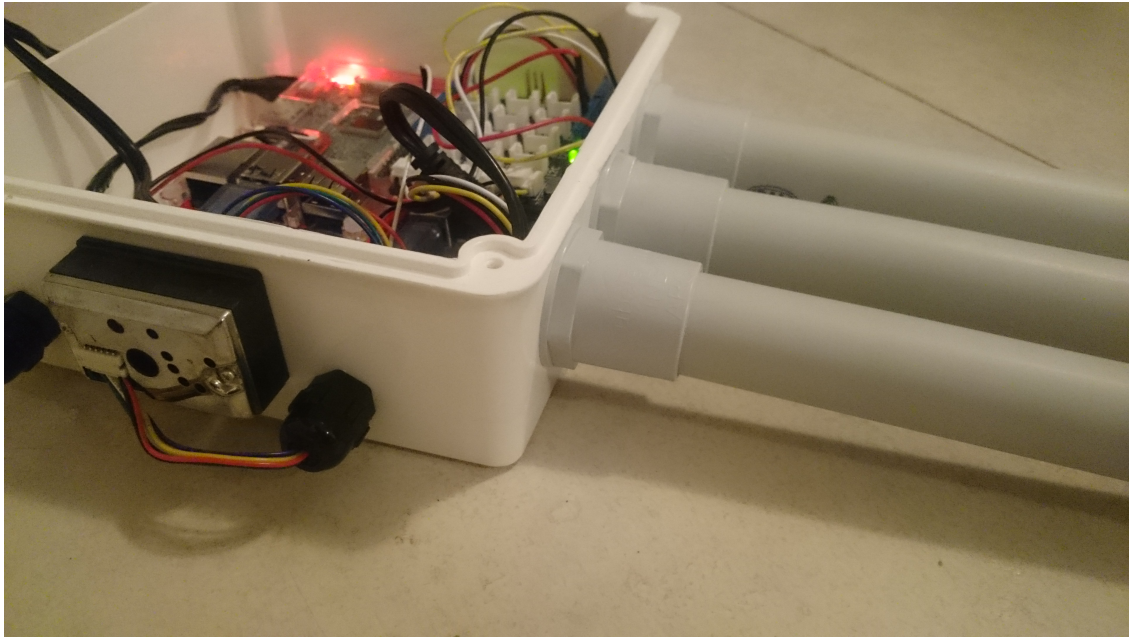


Figura 5.9: Vista lateral del gabinete.

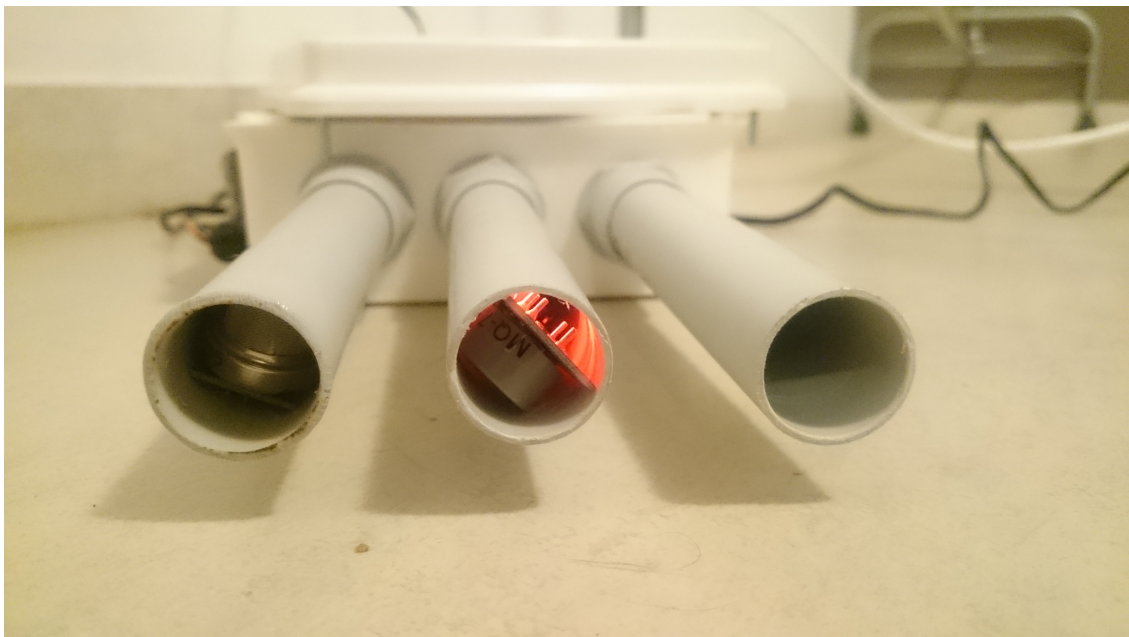


Figura 5.10: Vista inferior de los sensores.

En la última figura, Fig 5.10, puede apreciarse la ubicación estratégica de los sensores en el interior de los tubos inferiores, otorgándole a los mismos una protección ideal frente al clima.

5.3. Mediciones

Para consolidar las funcionalidades del prototipo, se llevaron a cabo mediciones sobre el dispositivo descrito anteriormente. Como resultado de esta evaluación, se observan en las Figuras 5.11-5.12 explícitamente los resultados obtenidos por los sensores en un período de treinta minutos.

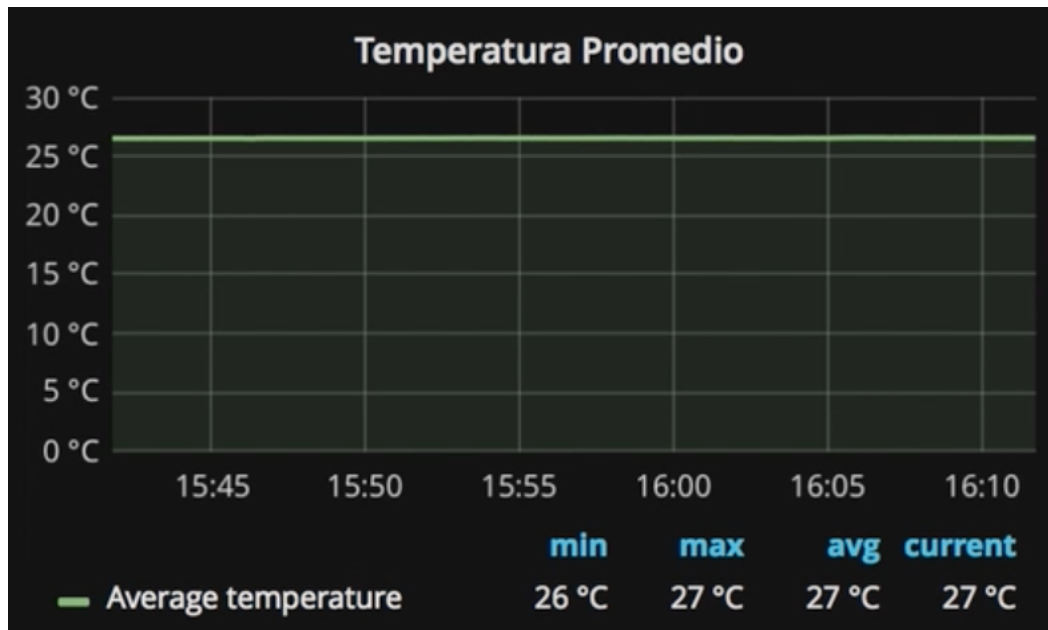


Figura 5.11: Histórico de temperatura promedio.

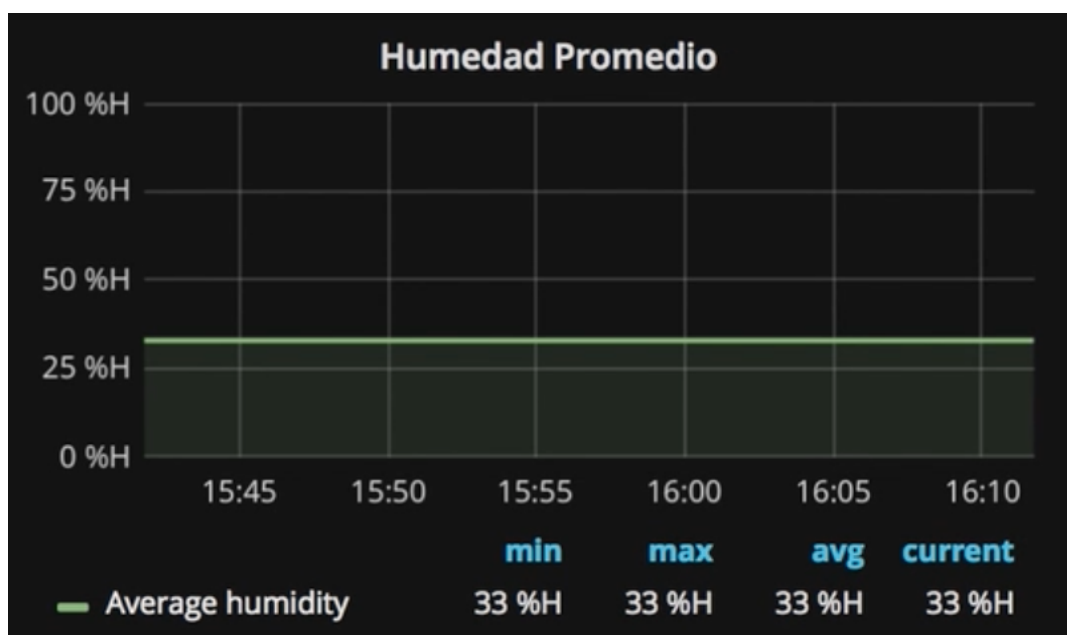


Figura 5.12: Histórico de humedad promedio.

VARIABLES como la temperatura y la humedad del ambiente son relativamente estables y no cambian tan fácilmente como se observa en las Figuras 5.11-5.12. En cambio, por el contrario, parámetros como el polvillo y la concentración de gases poseen velocidades de cambio mayores para el mismo período de tiempo, tal cual se observa en las Figuras 5.13-5.14



Figura 5.13: Histórico de polvillo promedio disperso en el ambiente.

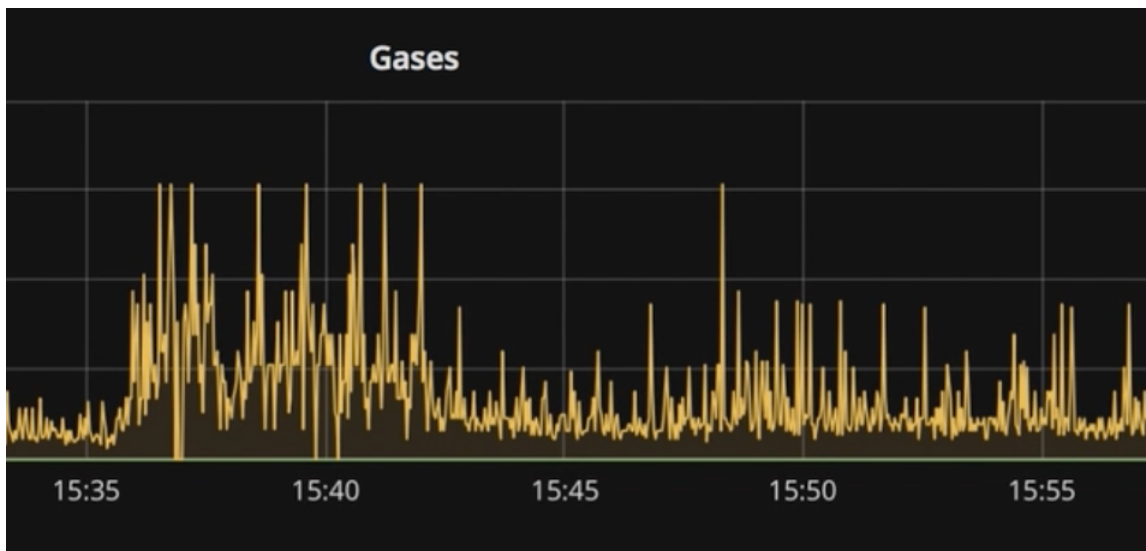


Figura 5.14: Histórico de la concentración de gases.



5.4. *Test Cases*

Un *test case* (caso de prueba en español), en ingeniería de software, es un conjunto de condiciones bajo las cuales se pretende determinar si una aplicación, sistema de software o una de sus características está funcionando como se estableció originalmente.

El mecanismo para determinar si un programa de software o sistema ha pasado o no una prueba de este tipo es conocido como un oráculo de pruebas. En algunas configuraciones, un oráculo podría ser, por ejemplo, un caso de requerimiento o la respuesta del sistema a cierta condición a la cual es sometido.

Normalmente, los *test cases* son utilizados para determinar que un programa de software o el sistema se considera suficientemente robusto y listo para ser lanzado.

En este proyecto se realizaron dos tipos de test cases al prototipo: de conectividad y de funcionalidad. A continuación se describen con mayor detalle cada uno de ellos.

5.4.1. De conectividad

Estos *test cases* pueden dividirse en dos clases bien diferenciadas respecto al tipo de señal y protocolo que se está evaluado.

Por un lado están las pruebas de señales, analógicas y digitales, que pretenden asegurar el conexionado entre los sensores y el resto del hardware a las que se denominan de tipo-1 y por otro están las realizadas sobre la conectividad a nivel lógico de la base de datos con el *front-end* y el usuario final denominadas de tipo-2.

Tipo-1 Tomando en consideración las recomendaciones hechas por los fabricantes y las notas capturadas en el sitio web oficial de Arduino¹ denominado *Playground*, donde se exponen ejemplos, librerías, consejos, etc., de como interactuar con la placa Arduino UNO y los posibles periféricos, se llevaron adelante pruebas de conectividad tal como se observa a continuación en la Figura 5.15.

¹<http://playground.arduino.cc/Main/HomePage>

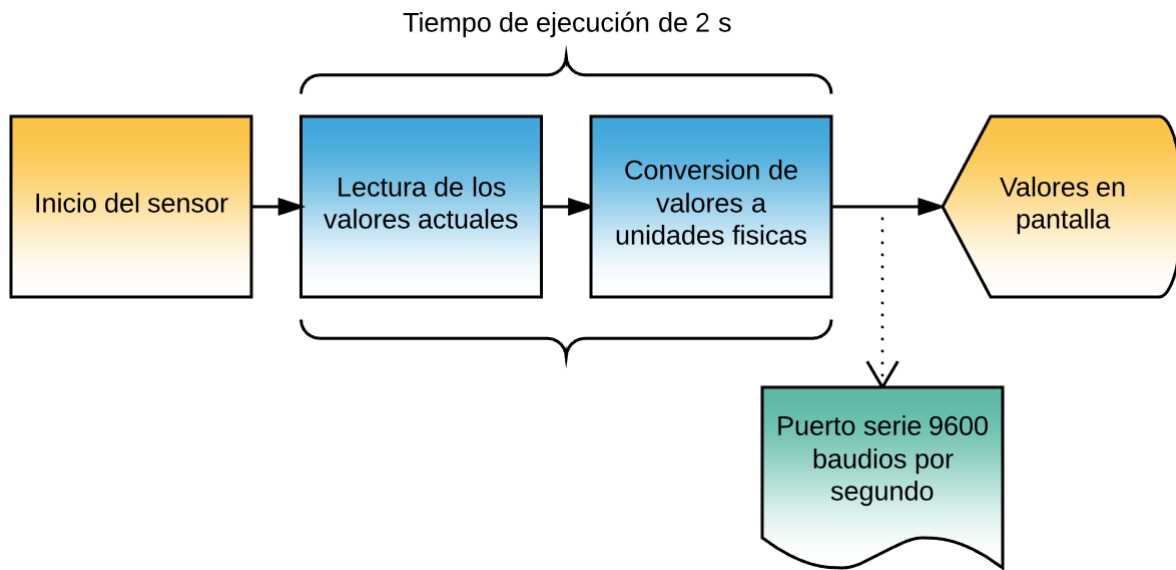


Figura 5.15: Proceso de sensado para DHT11.

Una vez que se logró tener los datos de las condiciones atmosféricas capturadas por los sensores presentes en el emulador de puerto serie provisto por Arduino IDE, se procedió a ejecutar el *script* en Raspberry Pi para que éste finalmente lo importase en el índice de Elasticsearch.

Por último, observando los registros presentes en el índice de Elasticsearch, se verificó que se incrementaban los registros añadidos al mismo en forma correcta, dando por finalizada la prueba.

Tipo-2 Para las pruebas de este tipo, el ensayo de la misma se basó en utilizar herramientas internas propias de Elasticsearch y Grafana que garantizan la conectividad de las partes.

En el Código 5.1 y 5.2 se observan un extractos de la configuración interna de ambas herramientas denotando que es la correcta.



Código 5.1: Extracto del archivo de configuración de Elasticsearch (/etc/elasticsearch/elasticsearch.yml).

```
#                               Network
#
# Set the bind address to a specific IP (IPv4 or IPv6):
#
network.host: _wlan0_
#
# Set a custom port for HTTP:
#
http.port: 9200
#
# For more information, see the documentation at:
# <http://www.elastic.co/guide/en/elasticsearch/reference/current/modules-network.html>
#
```

Código 5.2: Extracto del archivo de configuración de Grafana (/etc/grafana/grafana.ini).

```
#                               Server
[server]
# Protocol (http or https)
;protocol = http

# The ip address to bind to, empty will bind to all interfaces
;http_addr =

# The http port to use
;http_port = 3000

# The public facing domain name used to access grafana from a browser
;domain = localhost
```



Finalmente, para poder asegurar que el *datasource* de Elasticsearch está correctamente configurado en Grafana, se utilizó a la propia herramienta de *front-end* para corroborarlo. En la Figura 5.16 se observa a continuación una impresión de pantalla.

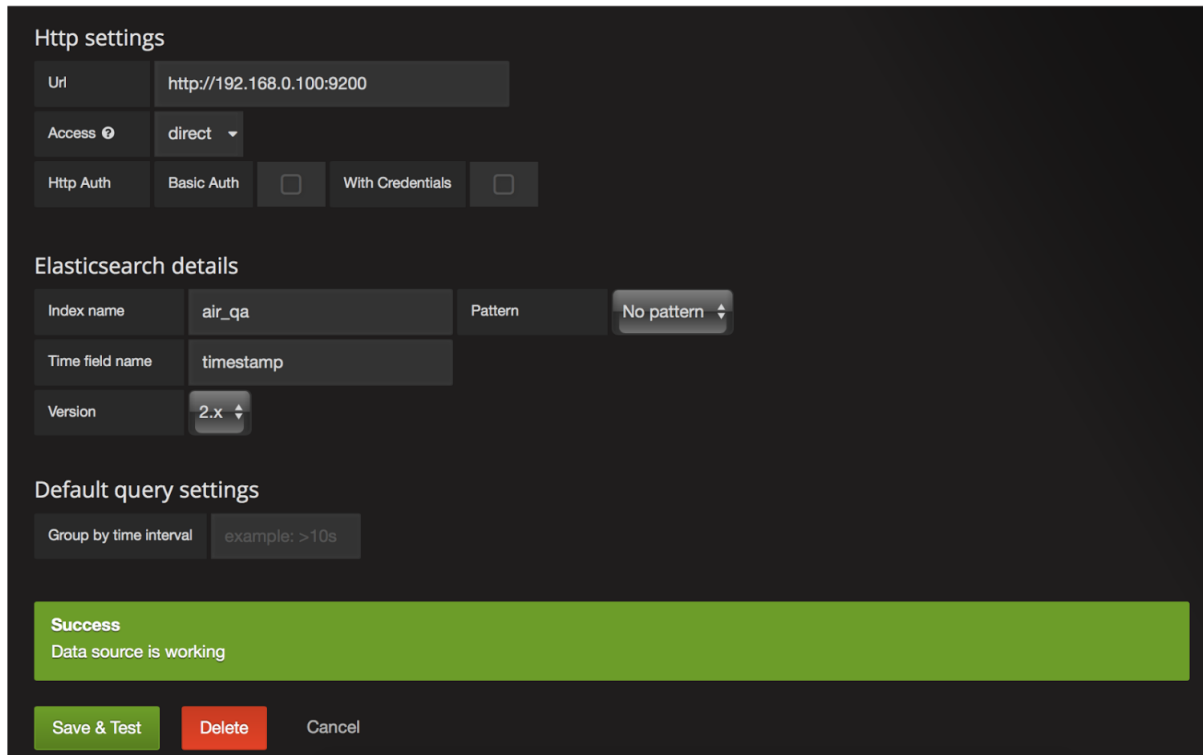


Figura 5.16: Impresión de pantalla de Grafana, configuración de *datasources*.



5.4.2. De funcionalidad

Una vez aceptados y concluidos los test de conectividad garantizando el correcto acople de las partes, se procedió a realizar los tests de funcionalidad.

Este tipo de pruebas consiste básicamente en determinar qué tanto se asemeja el desempeño del dispositivo y su capacidad de poder cumplir con el objetivo de capturar las variaciones de parámetros en el ambiente que lo rodea.

La evaluación de desempeño del dispositivo para poder capturar en forma eficiente valores de temperatura, humedad y presión atmosférica fue contrastado con instrumental diseñado para tal fin, como es el caso de termómetros ambientales, medidores de humedad ambiental y barómetros.

En lo que respecta a monóxido y dióxido de carbono, lamentablemente no se cuenta con un entorno controlado para poder validar estas mediciones más allá de lo que se observa a simple vista cuando el prototipo se encuentra al aire libre respecto a cuando lo hace en una habitación encerrada por largos periodos de tiempo.

Para obtener datos fehacientes en la concentración de monóxido y dióxido de carbono, se siguió a rigurosamente un proceso de calibración indirecta mediante regresión lineal y un modelo matemático específico impulsado por Davide Gironi². Este proceso de calibración indirecta está tácitamente adherido al proyecto mediante el uso de librerías propias.

²Modelo de regresión lineal y calibración de sensores MQ-XX
<http://davidegironi.blogspot.com.ar/2014/01/cheap-co2-meter-using-mq135-sensor-with.html#.WCdhLaOZMb0>



Capítulo 6

Resultados

Tras la implementación, prueba, puesta a punto y mejora del prototipo, los resultados obtenidos son satisfactorios.

En cuanto al cumplimiento de los objetivos principales de este trabajo, se logró cuantificar con éxito la mayoría de los parámetros propuestos, tal es el caso de la humedad, temperatura, presión atmosférica absoluta y relativa, concentración de monóxido de carbono y el dióxido de carbono.

Por otro lado, no fue posible manipular y aislar gases tales como el dióxido de azufre u óxidos de nitrógeno, para luego poder capturar valores significativos de los mismos. El dióxido de azufre es un gas incoloro con un característico olor asfixiante y muy tóxico, y los óxidos de nitrógeno se presentan bajo combustiones controladas mezclando oxígeno y nitrógeno. En ambos casos, las condiciones de los experimentos y herramientas involucradas necesarias para evaluarlos excedían las capacidades del proyecto.

Desde otra perspectiva, el prototipo final dista de los objetivos originales del proyecto únicamente en su capacidad de detectar cierto tipo de gases, pero en líneas generales, cumplió con las expectativas respetando rigurosamente el diseño propuesto desde un comienzo.

Entre los aspectos más relevantes de este proyecto se encuentra la tarea de interconexión de los sensores a la placa Arduino UNO. Quizás el punto más dificultoso fue lograr que las señales analógicas y digitales provenientes de los diferentes dispositivos se adapten a variables físicas cuantificables y calificables. Para poder abordar este problema se hizo uso íntegro de la potencialidad y flexibilidad de la programación mediante el lenguaje de *scripting* Python.

La información recolectada por este prototipo junto al gran potencial de *IoT* para el acceso y el procesamiento de la misma, tienen una importancia sobresaliente en el impulso y desarrollo de ciudades ecológicamente activas y pujantes en su progreso.



Capítulo 7

Conclusiones

Se puede destacar a partir de todo este proceso de desarrollo y evaluación de un sistema de medición de la calidad del aire, la importancia del uso de la tecnología asociada a la *IoT* para tomar conciencia sobre el medio ambiente y las condiciones climáticas presentes donde un sistema de estas características esté siendo utilizado.

Con la construcción de este prototipo en serie y la instalación de los mismos en lugares esenciales de una ciudad, podría conformarse así una red neuronal artificial con estos equipos incrementado las prestaciones del mismo. De esta manera, el sistema no solamente mostraría datos históricos y actuales de los parámetros que definen la calidad del aire de una ciudad, sino que también podría realizar la predicción de determinados eventos que permitirían tomar decisiones más acertadas sobre las actividades en el área.

Respecto al trabajo pendiente de esta propuesta, queda por incorporar en este dispositivo el uso de tecnología celular y/o Wi-Fi para acceder a Internet permitiendo la movilidad del equipo. Por otro lado, también resulta de interés incluir un sistema de alimentación y carga de baterías, por ejemplo, utilizando paneles solares para que el sistema de medición pueda funcionar ante eventuales cortes de energía o ser instalado en lugares donde no se tenga acceso a la red de suministro eléctrico.



Capítulo 8

Bibliografía

Bibliografía Principal

- *MQ7* - Hoja de datos del sensor MQ-7 [online], Zhengzhou Winsen Electronics Technology, 2014. Disponible: www.winsensor.com (documento adjunto).
- *GP2y1010au0f* Hoja de datos del sensor GP2y1010au0f [online], Sharp, 2010. Disponible: <http://www.sharp-world.com> (documento adjunto).
- *MQ135* Hoja de datos del módulo con sensor MQ135 [online], Waveshare, 2008. Disponible: www.waveshare.com (documento adjunto).
- *DHT11* Hoja de datos del sensor DHT11 [online], Microbot, 2014. Disponible: www.microbot.it (documento adjunto).
- *Tanenbaum* A. Tanenbaum y D. Wetherall, Redes de Computadoras, 2012.
- *Arduino* Plataforma Arduino: www.arduino.cc
- *Reyes* F. Reyes Cortés y J. Cid Monjaraz, Arduino, Aplicaciones en Robótica, Mecatrónica e Ingenierías, 2015.
- *Raspi* Raspberry Pi: www.raspberrypi.org
- *Normativa* Ley Nacional N° 20248 [online]. Disponible: www2.medioambiente.gov.ar (documento adjunto).
- *IoT* El Internet de las Cosas [online], Fundación de la Innovación Bankinter, 2011. Disponible: www.fundacionbankinter.org

Bibliografía complementaria

- *G. Galeano*, Programación de Sistemas Embebidos en C, 2009.
- *W. Stallings*, Comunicaciones y Redes de Computadoras, 2004.



Recursos bibliográficos online

- Google Scholar: <https://scholar.google.es>
- IEEE Xplore: <http://ieeexplore.ieee.org>
- CiteSeerX: <http://citeseerx.ist.psu.edu>



Anexos