



Adquisició i gestió de la contaminació a les ciutats mitjançant Arduino i Sentilo

Albert Robles Montoro

Grau de Tecnologies de la Telecomunicació

Àrea de treball: Arduino

Nom Consultor/a: Antoni Morell Pérez

Nom Professor/a responsable de l'assignatura: Pere Tuset Peiró

Data: 11/06/2017



Aquesta obra està subjecta a una llicència de [Reconeixement-NoComercial-SenseObraDerivada 3.0 Espanya de Creative Commons](https://creativecommons.org/licenses/by-nc-nd/3.0/es/)

FITXA DEL TREBALL FINAL

Títol del treball:	<i>Adquisició i gestió de la contaminació a les ciutats mitjançant Arduino i Sentilo</i>
Nom de l'autor:	<i>Albert Robles Montoro</i>
Nom del consultor/a:	<i>Antoni Morell Pérez</i>
Nom del PRA:	<i>Pere Tuset Peiró</i>
Data de lliurament (mm/aaaa):	<i>06/2017</i>
Titulació o programa:	<i>Pla d'estudis de l'estudiant</i>
Àrea del Treball Final:	<i>Arduino</i>
Idioma del treball:	<i>Català</i>
Paraules clau	<i>Arduino, sensors ambientals, smart city</i>
Resum del Treball (màxim 250 paraules):	
<p>Les smart cities s'han convertit en una necessitat degut a la gran densitat de població als nuclis urbans i a la quantitat de recursos consumits per ells. Es fan necessàries unes ciutats on s'implanti una gestió eficient i sostenible dels recursos energètics i humans, de la mobilitat, de la economia i la política, tenint un impacte positiu a la qualitat de vida i al benestar dels ciutadans a partir de les tecnologies de la informació.</p> <p>Un dels principal eixos de les smart cities és la gestió de recursos i la pol·lució, degut a que influeix directament en el benestar i la salut dels ciutadans. A data del 2016, a Europa moren prematurament 432.000 persones a l'any degut a la contaminació, unes 32.000 a Espanya, segons les dades del informe anual de qualitat de l'aire de 2015 de l'Agència Europea de Medi Ambient (AEMA) [1].</p> <p>Aquest projecte tracta de la adquisició de les dades de pol·lució a partir d'una xarxa de sensors i la realització d'un mapa de contaminació de l'aire, acústica i lumínica a les ciutats així com la utilització de les dades per a regulació del transit mitjançant panells informatius.</p> <p>El sistema consta de sensors basats en l'arquitectura Arduino i la plataforma Sentilo que és una plataforma d'interconnexió de sensors i actuadors de codi obert impulsada per l'ajuntament de Barcelona al 2012 i que està dissenyada per gestionar l'arquitectura Smart City de qualsevol ciutat que tingui com a objectiu la transversalitat i interoperabilitat.</p> <p>El sistema ens permetrà connectar els sensors a la plataforma, monitoritzar les dades a un servidor web i prendre decisions basades en les dades adquirides.</p>	

Abstract (in English, 250 words or less):

Smart cities have become a necessity due to the high population density in urban areas and the amount of resources consumed by them. Are necessary cities where implemented an efficient and sustainable management of energy and human resources, mobility, the economy and politics, having a positive impact on citizens' life's quality and welfare based on information technologies.

One of the main axes of smart cities is the management of resources and pollution, because it directly affects the health and welfare of citizens. As of 2016, in Europe die prematurely 432,000 people a year due to pollution, 32,000 people in Spain, according to the "Annual Air Quality-2015 report" by the European Environment Agency (EEA) [1].

This project is about the acquisition of pollution data from a network of sensors, conducting a map of air, noise and light pollution in cities and the use of these data for regulation of transit through informative panels.

The system is based on sensors installed on Arduino architecture and the platform Sentilo which is an open source platform for networking of sensors and actuators set up by the Barcelona City Council in 2012. Sentilo is designed to manage the Smart City architecture of any city that aims at mainstreaming and interoperability.

The system allows to connect the sensors, display the data through a web server and make decisions based on data.

Índex

1.	Introducció	4
1.1	Motivació.....	4
1.2	Objectius.....	6
1.3	Planificació de treball.....	6
1.4	Organització del document	7
2.	Estat de l'art	8
2.1	El projecte Word Air Quality Index	8
2.2	Projecte Smart Citizen.	8
2.3	Plataformes d'administració de sensors.....	9
2.3.1	Projecte Sentilo.....	10
2.3.2	Sofia 2.....	12
2.4	Libelium	13
3.	Plataforma Sentilo	13
3.1	Allotjament.....	14
3.2	Estructura	15
3.3	Administració Sentilo	16
3.3.1	API REST de Sentilo	16
3.3.2	Catàleg Web: Configuració xarxa de sensors a la plataforma.....	18
3.3.3	Alertes	20
4.	Arduino	21
4.1	Estructura	21
4.2	Components i actuadors.....	22
5.	Sensors	23
5.1	Micròfon	23
5.2	Sensor il·luminació	24
5.3	Qualitat de l'aire	25
5.4	Temperatura i humitat.....	27
6.	Desenvolupament del projecte.....	27
6.1	Catàleg Web	27
6.2	Prototip Arduino Components	33
6.2.1	Adquisició de dades.....	34
6.2.2	Model Ethernet.....	35
6.2.3	Model Wifi.....	37
6.2.4	Mecanitzat	40
6.2.5	Alimentació:	42
6.3	Actuadors	44
6.3.1	Tractament Alertes al Actuator.....	44
6.3.2	Publicació missatges per LCD.....	46
7.	Proves funcionament.....	46
8.	Viabilitat.....	49

8.1 Viabilitat tècnica.....	49
8.2 Viabilitat econòmica:	50
8.2.1 Plataforma Sentilo	51
8.2.2 Payback.	51
8 Conclusions	53
8.2 Possibles millores.....	54
8.3 Autoavaluació personal	54
Bibliografia	56
ANNEX	58
A1. Accés a Thingtia.cloud	58
A2. Càlculs adquisició de dades i calibració	59
A3. Llista de comandes AT pel mòdul ESP8266-01	64
A5. Enllaços Datasheets	65

Lista d'imatges

<i>Imatge 1. Efectes del soroll a la salut humana. Font http://contaminacionacustica.net</i>	5
<i>Imatge 2. Publicació de dades de qualitat de l'aire a Barcelona a aqi.cn.org</i>	8
<i>Imatge 3. Publicació de dades d'un sensor a la plataforma Smart Citizen</i>	9
<i>Imatge 4. Visualització de les dades d'un sensor a la plataforma Sentilo Barcelona</i>	10
<i>Imatge 5. Arquitectura Plataforma Sentilo. Font Sentilo.io</i>	11
<i>Imatge 6. Arquitectura funcional Sofia2. Font sofia2.com</i>	12
<i>Imatge 7. Pàgina inici Plataforma Sentilo utilitzada pel projecte</i>	15
<i>Imatge 8. Estructura Plataforma Sentilo. Font Sentilo.io</i>	15
<i>Imatge 9. Entorn gràfic de l'aplicació Catalog Web</i>	18
<i>Imatge 10. Informació de les estadístiques de Sentilo Terrassa</i>	19
<i>Imatge 11. Vista universal amb les dades d'un sensor de Sentilo Terrassa</i>	20
<i>Imatge 12. Historial de dades captades per un sensor de Sentilo Terrassa</i>	20
<i>Imatge 13. Arduino Mega 2560. Font: https://forum.arduino.cc/</i>	21
<i>Imatge 14. Sparkfun BOB-09964. Font: https://www.sparkfun.com</i>	24
<i>Imatge 15. LDR GL5528</i>	24
<i>Imatge 16. Mòdul sensor il·luminació TSL2561</i>	24
<i>Imatge 17. Corba il·luminació-resistència GL5528</i>	25
<i>Imatge 18. Sensor Sharp GP2Y1010AUOF</i>	25
<i>Imatge 19. Sensor NO₂ MICS 2714</i>	26
<i>Imatge 20. Mòdul MQ-135 i corba característica dels diferents agents</i>	26
<i>Imatge 21. Sensor temperatura i humitat DTH22</i>	27
<i>Imatge 22. Estructura catalog Web</i>	28
<i>Imatge 23. Shield de sensors</i>	33
<i>Imatge 24. Connexió i esquema dels sensors i l'Arduino</i>	34
<i>Imatge 25. Component model Ethernet</i>	35
<i>Imatge 26. Ethernet Shield 2 de Arduino</i>	36
<i>Imatge 27. Component model WiFi</i>	37
<i>Imatge 28. ESP8266-01</i>	38
<i>Imatge 29. Diagrama de blocs del ESP8266</i>	38
<i>Imatge 30. Disseny 3D del mecanitzat</i>	41
<i>Imatge 31. Component model WiFi amb el mecanitzat</i>	42
<i>Imatge 32. YwRobot Power MB V2</i>	42
<i>Imatge 33- Esquema circuit alimentació 3,3V</i>	43
<i>Imatge 34. Actuador</i>	44
<i>Imatge 35. Component WiFi ubicat a un fanal</i>	49

Lista de Taules

<i>Tabla 1. Expressions per avaluar alertes internes a Sentilo. Font Sentilo.io</i>	21
<i>Tabla 2. Llibreria sentiloClient</i>	37
<i>Tabla 3. Comandes AT per ESP8266</i>	39
<i>Tabla 4. Consum Component model WiFi</i>	43
<i>Tabla 5. Funcions llibreria Ethernet</i>	45
<i>Tabla 6. Funcions llibreria LCD</i>	46
<i>Tabla 7. Exemple de pressupost de sistema autònom</i>	50

1. Introducció

1.1 Motivació

La gran densitat de població als nuclis urbans i a la quantitat de recursos consumits a ells fan necessàries unes ciutats intel·ligents amb una gestió eficient i sostenible dels recursos energètics i humans, de mobilitat, economia i política.

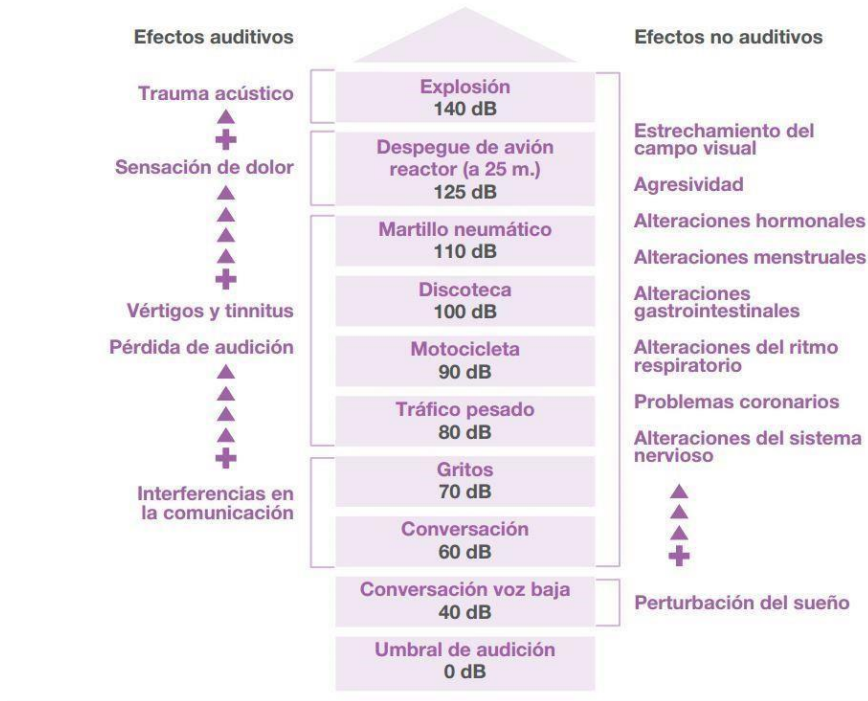
Un dels principals problemes dels nuclis urbans és la contaminació, tant de l'aire, com lumínica i sonora. A més, està d'actualitat degut a les mesures preses per alguns ajuntaments com el de Barcelona i Madrid que han restringit la velocitat o el transit als seus nuclis urbans basant-se en les dades de contaminació tal com marca la Directiva Europea 2008/50/CE del Parlament Europeu i el Consell, on es defineixen i es fixen els objectius de qualitat de l'aire ambient a fi d'evitar, prevenir o reduir els efectes nocius per a la salut humana i el medi ambient.

La pol·lució influeix directament en el benestar i la salut dels ciutadans i segons informe anual de qualitat de l'aire de 2015 de la Agència Europea de Medi Ambient (AEMA) a Europa van morir prematurament 432.000 persones durant l'any 2012 degut a la contaminació, unes 25.000 a Espanya. Entorn al 90% de la població urbana de la Unió Europea està exposada a concentracions de contaminants atmosfèrics a nivells alts considerats nocius per a la salut. Per exemple, es calcula que les partícules fines (PM2.5) presents a l'atmosfera redueixen l'esperança de vida en la Unió Europea en més de vuit mesos. [1]

La preocupació per la pol·lució ha augmentat a la societat i unit a la demanda dels ciutadans de polítiques de transparència es justifica la creació d'un servei de mapes de contaminació de les ciutats amb dades en temps real i que permeti als ciutadans conèixer totes les dades de contaminació a les que s'exposen accedint a un lloc web.

Encara que la contaminació atmosfèrica sembla ser la més perjudicial per a la població, no s'han de deixar de banda altres factors com la contaminació acústica i lumínica que també influeixen en el benestar dels ciutadans. La contaminació acústica té conseqüències com trastorns auditius, interferències a la comunicació, pertorbació de la son, estrès, irritabilitat, disminució del rendiment i de la concentració, agressivitat, cefalees, alteració de la pressió arterial o depressió del sistema immunològic [2].

Ruido y salud



Imatge 1. Efectes del soroll a la salut humana. Font <http://contaminacionacustica.net>

En el cas de la contaminació lumínica, que és la que més desapercibuda ha passat per la societat, amenaça amb eliminar el segon cicle astronòmic, es a dir la nit, cosa que afecta tant al medi ambient com a éssers vius i la salut humana [3]. Precisament son els països desenvolupats i les grans ciutats els grans afectats per la contaminació lumínica.

És per això que a aquest treball es tracten, a més de la contaminació ambiental, la contaminació sonora i la lumínica a través de sensors de so i llum connectats a la plataforma web.

Amb els sistema es podran, a partir de les dades obtingudes pels diferents sensors, prendre decisions sobre l'accés dels vehicles a determinades zones de la ciutat, recomanant rutes alternatives o restringint el transit. Seran els panells informatius els encarregats de mostrar les alertes, recomanacions o restriccions als ciutadans.

Per la realització del projecte s'ha decidit treballar amb la plataforma Sentilo, una plataforma impulsada per l'Ajuntament de Barcelona al 2012 de codi obert, que permet la interconnexió entre sensors i actuadors i que està sent fortament impulsada per la diputació de Barcelona. Mitjançant la plataforma Sentilo es poden rebre les dades dels sensors, emmagatzemar-les, publicar-les i crear alertes segons unes regles configurables.

La plataforma Sentilo fortament impulsada per la diputació de Barcelona i guardonada amb el premi "Open Awards 2016" com a plataforma més innovadora desenvolupada amb programari lliure. A més a més, la seva arquitectura ha estat adoptada per City Protocol Society [4] com a arquitectura Smart city de referencia degut a que Sentilo està dissenyada com una plataforma creuada amb l'objectiu de compartir informació entre sistemes heterogenis i disposant d'una fàcil integració de noves aplicacions. Es pot dir que Sentilo encaixa en l'arquitectura smart city de qualsevol ciutat.

1.2 Objectius

El treball busca facilitar l'accés a les dades de contaminació atmosfèrica, sonora i lumínica als ciutadans a partir d'una plataforma web. Així mateix es vol utilitzar la mateixa plataforma per poder crear alarmes i missatges que es visualitzaran a panells informatius i que podran mostrar alarmes i suggerir rutes alternatives, restriccions de transit a carrers o zones determinades de la ciutat.

Es poden enumerar els següents objectius principals per ordre de prioritat:

1. Implementació d'un sistema encastat format per una placa Arduino i els sensors de contaminació atmosfèrica, sonora i lumínica realitzant la configuració i programació de la placa.
2. Adquisició de les dades i comunicació del sistema encastat amb la plataforma Sentilo per la posterior publicació de les dades a la web.
3. Creació del catàleg de components i sensors al Sentilo Catalog a partir del panell d'administració i de la API (application programming interface) REST (Representational State Transfer) que Sentilo proporciona.
4. Publicació de les dades obtingudes pels sensors a la plataforma web.
5. Creació de les alarmes de contaminació superats uns llindars específics a partir d'un servidor de subscripció/publicació.
6. Subscripció dels Arduino a les publicacions de les alarmes de Sentilo.
7. Processament als Arduino de les alarmes rebudes i publicació de missatges de informació o restricció als panells informatius segons el tipus d'alarma.

Una vegada aconseguits els objectius principals podem enumerar altres objectius amb menys prioritat, també per ordre, que son:

1. Implementació de la comunicació mitjançant WiFi.
2. Estudi de la viabilitat d'utilització de Arduino UNO o d'altres més econòmics en lloc de Arduino Mega.
3. Creació d'una shield amb tots els sensors utilitzats.
4. Càlcul i implementació de l'alimentació autònoma dels sistema a partir de bateries.
5. Mecanització del prototip del sistema encastat.

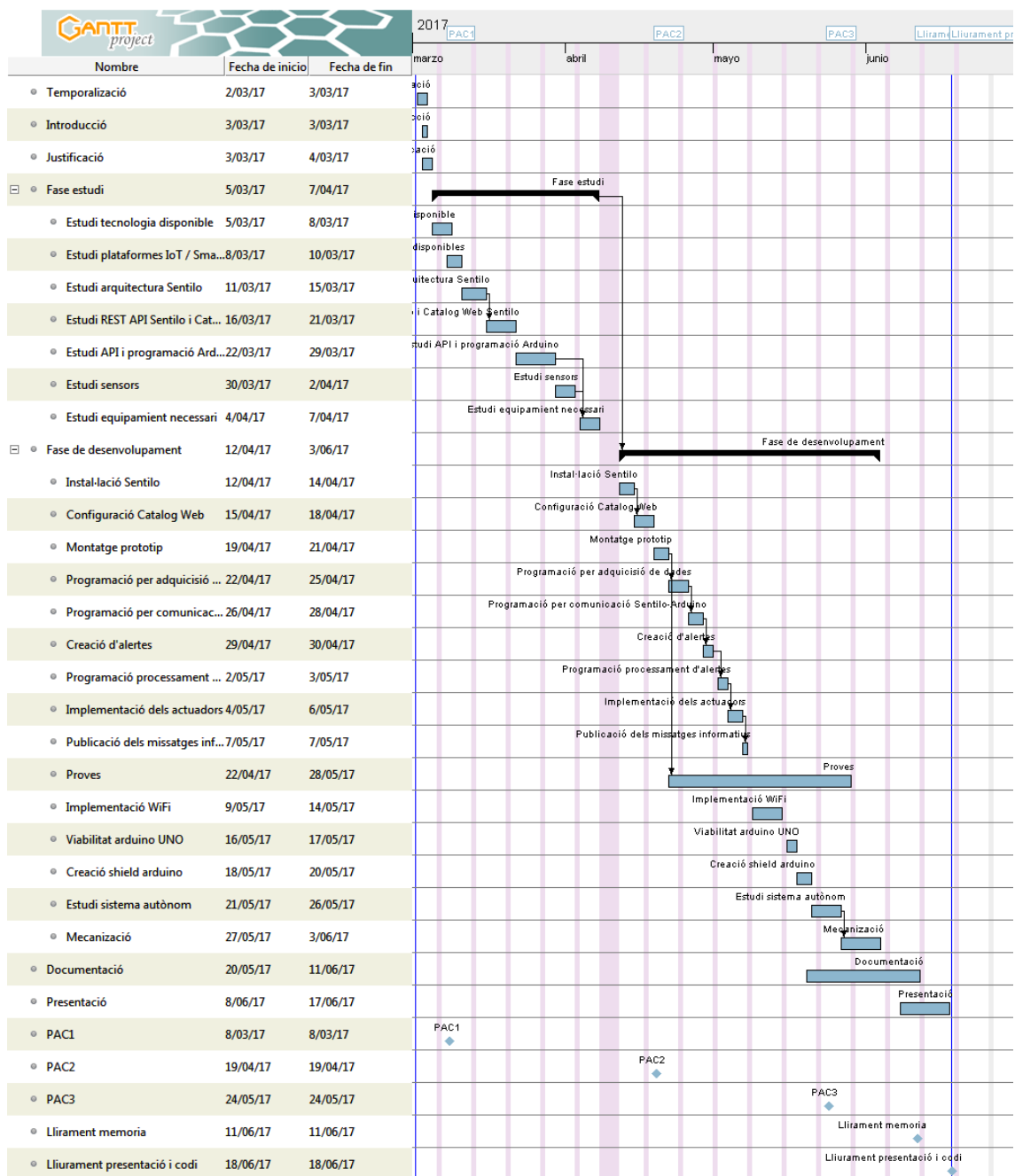
1.3 Planificació de treball

Per la planificació del treball s'han calculat les hores de cada tasca i s'han repartit tenint en compte una dedicació de 17 hores setmanals.

També s'ha tingut en compte que els dilluns han sigut el dies festius donat que és el dia que no hi ha disponibilitat per realitzar tasques del TFG i que del dia 8 al 11 d'Abril s'han considerat vacances. Al diagrama de Gantt s'han marcat els dies festius indicant els dilluns com cap de setmana i del 8 al 11 d'Abril com festius.

Per realitzar la planificació s'han tingut en compte els coneixements previs i la manca de coneixements de programació d'Arduino i de la REST API de Sentilo.

El projecte es divideix en dos parts, durant la primera part es realitzarà una fase d'estudi i posteriorment una fase de desenvolupament. A la següent imatge es pot observar la planificació del projecte complerts, amb totes les fases i les dates de lliuraments de les PAC, memòria i presentacions:



Durant el projecte s'ha realitzat un seguiment de la planificació sense desviacions greus. L'únic incident ha estat la falta de subministrament del sensor de son que es va resoldre en un període de 5 dies amb la compra d'un altre sensor a un altre distribuïdor.

1.4 Organització del document

El treball consta de quatre parts diferenciades, la primera és una introducció amb la justificació del projecte i els objectius a aconseguir.

Una segona part tracta de l'estudi i anàlisis de la documentació on s'inclou l'estat de l'art i l'estudi de la documentació de Sentilo, Arduino i els sensors a utilitzar.

Una vegada realitzat l'estudi s'inicia el desenvolupament del projecte que consta de la creació i programació del prototip així com la configuració de la plataforma Sentilo per poder enviar les dades, publicar-les i crear les alarmes.

Finalment s'exposen els resultats i les conclusions obtingudes.

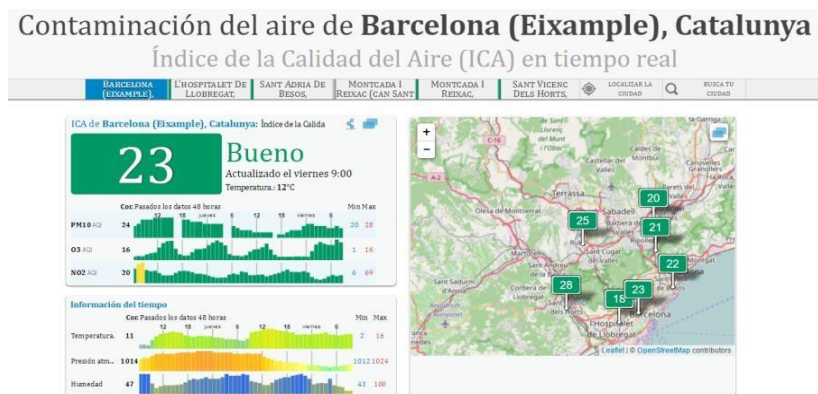
2. Estat de l'art

A l'actualitat algunes ciutats com Madrid publiquen dades de pol·lució de l'aire de la ciutat al seu Open Data per poder descarregar-lo en format de text [5] i existeixen dues alternatives per poder visualitzar les dades sobre un mapa que son a través del projecte Word Air Quality Index [6] i la plataforma Smart Citizen [7].

2.1 El projecte Word Air Quality Index

El projecte Word Air Quality Index és un projecte de caràcter social començat al 2007, amb la missió de promoure el coneixement de contaminació de l'aire i proporcionar informació unificada sobre la qualitat de l'aire per a tot el món, a partir de la publicació de les dades de qualitat de l'aire en temps real.

El projecte a data de Desembre de 2016 està mostrant de manera transparent la informació sobre la qualitat de l'aire per més de 70 països, cobrint més de 9000 estacions a 600 grans ciutats a través del llocs web aqicn.org i waqi.info.



Imatge 2. Publicació de dades de qualitat de l'aire a Barcelona a aqi.cn.org

2.2 Projecte Smart Citizen.

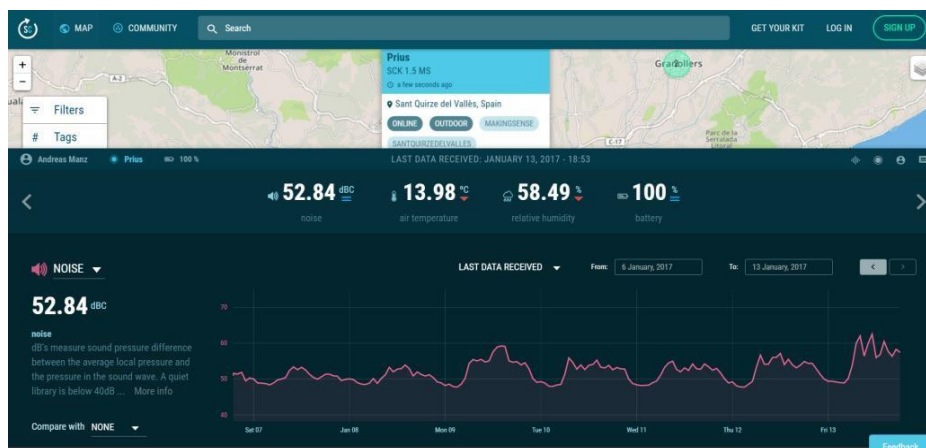
Va néixer dins de Fab Lab Barcelona i l'institut d'Arquitectura Avançada de Catalunya, tots dos centres enfocats en l'impacte de les noves tecnologies en les diferents escales de l'habitat humà, al 2012 gracies a la finançament a través del micromecenatge.

El projecte consta d'una plataforma, uns sensors i una aplicació mòbil i es basa en la geolocalització, Internet, hardware y software gratuït per a la captura i intercanvi de dades. Es busca generar processos participatius del ciutadans fent servir la plataforma com a node

per a la contribució d'indicadors que en aquest cas son les dades de composició de l'aire, humitat, temperatura, so i il·luminació a partir del hardware Smart Citizen Kit.

El Smart Citizen Kit és un hardware compost per un sensor, una tarja de processament de dades i una bateria, mecanitzats a un equip tancat. La versió 1.1 es basa en la següents components hardware [8]:

- Una shield de Arduino amb una CPU ATMEGA32U4.
- Un mòdul de WiFi encastat, WIFly RN-131G.
- Un sensor MICS4514, que és sensor de CO₂ i NO₂ .
- Un sensor de temperatura i humitat, el SHT21que inclou els dos sensors.
- El fotodíode BH1730FVC.
- Un sensor de soroll POM-3044P-R.
- Bus I2C que permet afegir altres sensors.



Imatge 3. Publicació de dades d'un sensor a la plataforma Smart Citizen

2.3 Plataformes d'administració de sensors

Existeixen dues filosofies diferents amb perspectives oposades en el camp de les plataformes de gestió de sensors. D'una banda, hi ha plataformes comercials propietàries que ofereixen enfocaments verticals amb solucions completes des dels sensors fins el software i actuadors, que no permeten la interoperabilitat amb altres plataformes.

Per altra banda, estan les plataformes obertes que permeten connexions heterogènies, amb sensors de múltiples proveïdors i que permeten una escalabilitat horitzontal de sensors i d'aplicacions. Aïllant les aplicacions que es desenvolupen per explotar la informació generada de la capa de sensors desplegats per recollir i transmetre aquesta informació s'aconsegueix evitar solucions verticals compartint informació entre sistemes heterogenis. Entre aquestes plataformes trobem Sentilo [9] i Sofia[10].

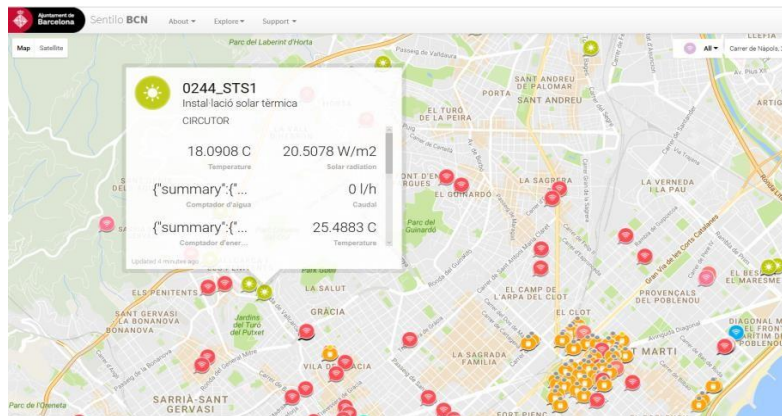
2.3.1 Projecte Sentilo

Sentilo és el resultat d'un projecte concebut per l'Ajuntament de Barcelona al 2012 per definir l'estratègia i les accions necessàries per aconseguir un posicionament global de Barcelona com un referent en l'àmbit de les ciutats intel·ligents.

És una plataforma lliure i de codi obert d'interconnexió entre sensors i actuadors dissenyada per encaixar en l'arquitectura Smart City de qualsevol ciutat que busqui una fàcil interoperabilitat i gran apertura que va guanyar el premi "Open Awards 2016" [11] com a plataforma més innovadora desenvolupada amb programari lliure.

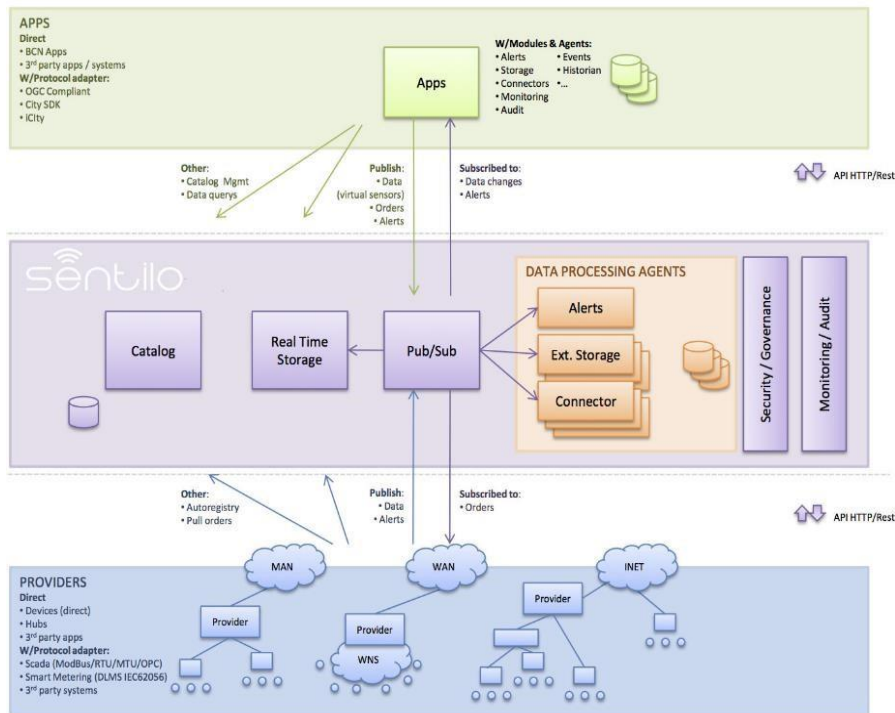
L'arquitectura de Sentilo disposa de les següents parts:

- Un front-end per al processament de missatges, amb una API REST
- Una consola d'administració web per configurar el sistema i gestionar el catàleg
- Una base de dades de memòria
- Una base de dades NoSQL, per tal d'aconseguir un sistema més flexible i escalable
- Un visor universal a partir d'un lloc web on publicar les dades.



Imatge 4. Visualització de les dades d'un sensor a la plataforma Sentilo Barcelona

- Un mòdul d'estadístiques que registra i mostra els indicadors bàsics de rendiment de la plataforma
- Un conjunt d'agents per a l'explotació de les bases de dades i per processar les alarmes internes.



Imatge 5. Arquitectura Plataforma Sentilo. Font Sentilo.io

Des del 2013 Sentilo s'ha implantat e integrat amb varies solucions per Smart Cities, les més destacades i es van presentar al SCEWC 2016 (Smart City Expo World Congres) a Barcelona i son les implementades a Barcelona, Terrassa, Reus i Sant Feliu de Llobregat, encara que s'estan realitzant proves a altres ciutats, entre elles Dubai i Kobe.

- **Barcelona**

Barcelona està utilitzant Sentilo com el middleware que interconnecta les xarxes de sensors i actuadors desplegats en tota la ciutat amb les aplicacions que els supervisen i controlen. A data del 2016 es registren fins 3.200.000 registres al dia, amb pics de 2000 registres per segon i l'estructura consta de 1800 sensors, en total uns 14000 paràmetres que tractar.

El principal servei que dona la plataforma és l'administració de consum i generació d'energia en edificis públics, però hi ha altres projectes per realitza lectures de comptadors d'aigua i recollida d'escombraries.

- **Diputació de Barcelona**

La diputació de Barcelona per desenvolupar el seu model de Smart Region ofereix el Sentilo com a SaaS (Software as a Service) fins a 311 municipis per a que desenvolupin els seus projectes de smart city amb Sentilo i ofereix ajudes de fins 10000€ per aquest projectes.

- **Terrassa**

El sistema consta de 1764 sensors amb una mitja de 4,8 registres per segon des de que es va posar en marxa al 2014.

Sobre la plataforma Sentilo Terrassa, s'ha implementat en primera instància la integració de les dades d'estacions meteorològiques, de servei de recollida de residus urbans, de gestió energètica d'equipaments municipals, de gestió de flotes de neteja viària i el sistema de reg de parcs de la ciutat.

- **Reus**

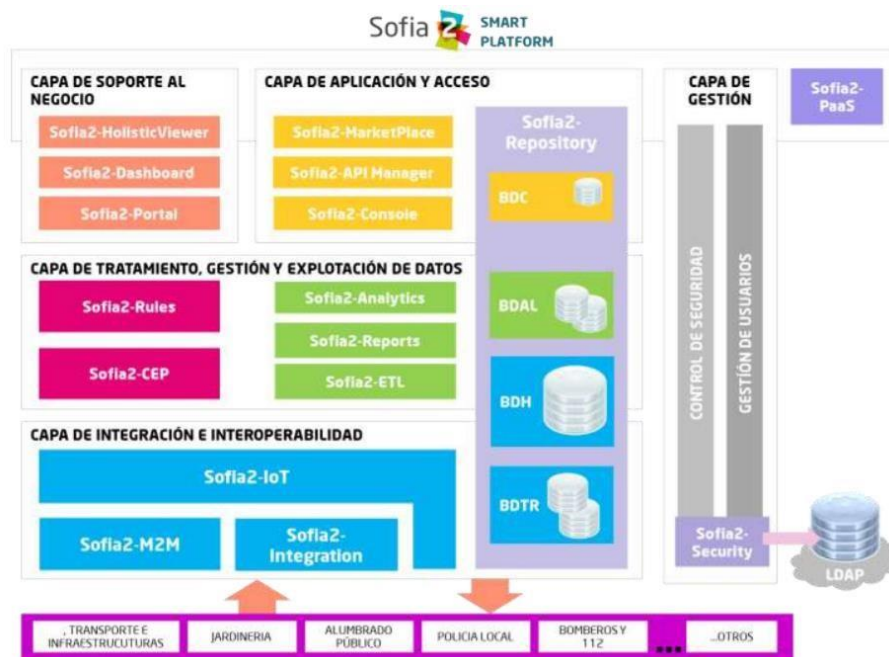
A Reus i Sant Feliu de Llobregat s'ha integrat la plataforma de Sentilo amb un sistema expert, Skynet, al qual s'ha programat tota la intel·ligència artificial i capacitat de decisió, per tal d'automatitzar la gestió d'infraestructures de smart city d'una manera ràpida i flexible.

A Reus les aplicacions que s'han desenvolupat son de gestió de transit i de aparcaments intel·ligents a zones comercials, a parades de Taxi i a places de carrega de vehicles elèctrics. A les parades de taxi a més s'utilitzen Beacons per la identificació i autenticació dels vehicles.

2.3.2 Sofia 2

Sofia 2 [10] també és un middleware de codi obert que ofereix interoperabilitat entre múltiples dispositius i sistemes, en aquest cas desenvolupat per Indra a partir de Sofia.

La seva arquitectura funcional es pot veure a la següent imatge [12]:



Imatge 6. Arquitectura funcional Sofia2. Font sofia2.com

Les característiques que ofereix Sofia2 son:

- Interfícies Big Data .
- Bases de dades.
- Integració amb back-ends per l'administració.
- Una API REST amb la utilització de l'estàndard JSON.
- Consola web d'administració.

Sofia2 ofereix solucions integrades completes, que normalment tenen un temps i cost d'implantació elevat, també ofereix una instància gratuïta al núvol anomenada Sofia2 CloudLab [13] per a realitzar proves i estudis per a col·laboradors.

Un projecte desenvolupat amb Sofia2 es Smart Coruña [14], Plataforma IOT (Internet of things) de la Coruña. Que dona serveis com els de localització d'autobusos i el temps real d'arribada a les parades i el visor de BiciCoruña on es poden consultar el nombre de bicicletes disponibles i els aparcaments disponibles per a bicicletes de la seva xarxa.

2.4 Libelium

Libelium és una empresa espanyola que dissenya i comercialitza equips per connectar sensors al núvol, basant-se en l'arquitectura d'Internet de les coses. I han participat en projectes de diferents àmbits com el de la medicina, la qualitat de l'aire a les mines d'or i la radiació nuclear entre d'altres [15].

És compatible amb Sentilo i s'han utilitzat en conjunt a alguns projectes com el de control de reg.

Entre els seus productes es troben el dispositiu waspnote i el Meshlium.

Waspnote [16] integra una placa amb una CPU ATmega1281 i una memòria flash de 128Kb, una placa amb els sensors entre les quals es poden escollir entre sensors de pol·lució, de presència, de radiació, de pàrquing, entre d'altres [17]. Per al cas que ens ocupa el més interessant és el Smart Cities PRO, que integra 20 sensors diferents, entre els que podem trobar els de contaminació atmosfèrica com CO2, NO2 i partícules, so i il·luminació.

Permet utilitzar diferents tecnologies de comunicació, en concret permet:

- Per llarg abast: 4G / 3G / GPRS / GPRS+GPS / LoRaWAN / LoRa / Sigfox / 868 MHz / 900 MHz
- Per mitjà abast: ZigBee / 802.15.4 / DigiMesh / WiFi
- Per curt abast: RFID/NFC / Bluetooth 2.1 / BLE

Meshlium és una passarel·la per la connexió de a xarxa de sensors 868 MHz / 900 MHz 802.15.4 al núvol via ethernet i 4G / 3G / GPRS / GSM.

3. Plataforma Sentilo

Els motius per escollir Sentilo son, com s'ha comentat prèviament, que evita solucions verticals permetent desenvolupar noves funcionalitats de forma modular i facilita la compartició d'informació entre diferents aplicacions. Al ser una plataforma dissenyada per treballar amb sistemes heterogenis, redueix la dependència de tecnologies específiques, productes o proveïdors i permet créixer conforme a les necessitats sense estar condicionades de base per la planificació inicial. La plataforma Sofia és una altre alternativa, però la versió gratuïta és per a col·laboradors i a més no disposa de totes les funcions.

Sentilo està composta de varis serveis que li donen la funcionalitat de interconnectador entre sensors i actuadors. Els serveis que el componen son bases de dades NoSQL i SQL, serveis Web i un servidor REDIS de publicació/subscripció.

Per la comunicació amb la plataforma Sentilo s'utilitzarà l'aplicació web i la aplicació API (application programming interface) REST (Representational State Transfer) que incorpora la plataforma.

A través de la plataforma web es crearan els proveïdors i els components necessaris per crear la xarxa de sensors i la mateixa plataforma aplicació web es publicaran les dades adquirides pels sensors.

Mitjançant la API REST de Sentilo es configuraran i subscriuran els sensors, s'enviaran les dades adquirides i es crearan les alarmes. En aquest punt cal destacar que els continguts dels missatges per utilitzar la API REST es realitzarà en format JSON.

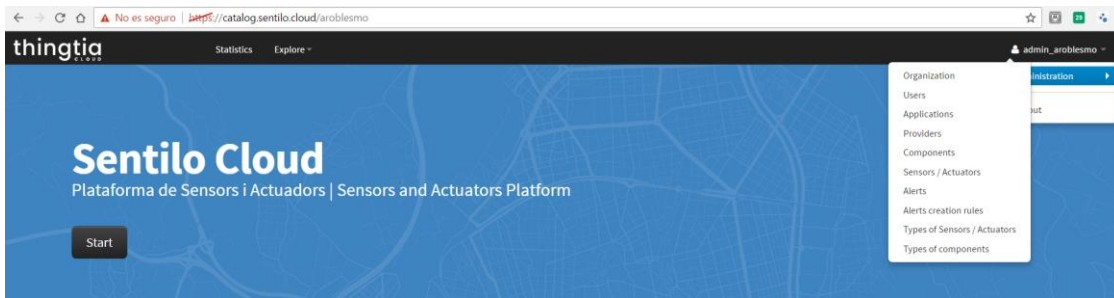
3.1 Allotjament

Per l'allotjament de la plataforma durant la realització del projecte s'han estudiat 3 possibilitats:

- Realitzar una instal·lació de tots els serveis de la plataforma a un servidor propi.
- Utilitzar la màquina virtual que Sentilo proporciona a un servidor propi amb el sistema operatiu VmWare ESXi.
- Utilitzar el servei SAAS de Sentilo que ofereix Thingtia Cloud. En concret la instància gratuïta Free2use, que s'ofereix per una primera aproximació doncs la major restricció és que permet 1000 observacions per dia i ens limitaria a l'hora d'augmentar el nombre de sensors.

Després de realitzar proves d'accés a la plataforma, fent l'allotjament a un servidor domèstic propi fent ús de la màquina virtual i a una instància proporcionada per Thingtia Cloud, s'ha vist que la fiabilitat, velocitat i capacitat de la instància Thingtia Cloud és major que la que el servidor domèstic podia proporcionar i per tant s'ha decidit utilitzar la instància de Sentilo proporcionada per Thingtia Cloud. Per contra, la decisió fa que es vegi limitat el control dels serveis i sobretot la personalització de l'entorn gràfic de la plataforma.

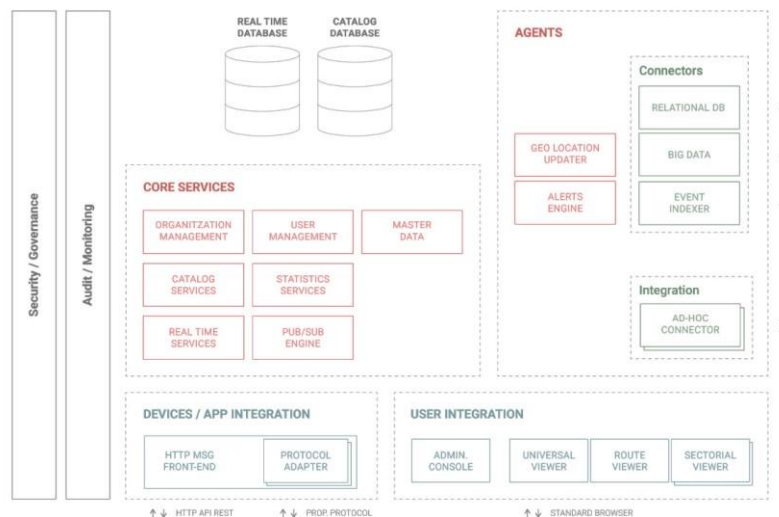
Per tant, per la realització del projecte s'ha utilitzat la màquina virtual proporcionada per Sentilo dins d'una xarxa local per a la realització de proves prèvies i per l'estudi de les funcionalitats de la Plataforma i una instància gratuïta del servei SAAS de Sentilo que ofereix Thingtia Cloud per la implementació final del projecte, l'adreça del qual és <https://catalog.sentilo.cloud/aroblesmo> .



Imatge 7. Pàgina inici Plataforma Sentilo utilitzada pel projecte

3.2 Estructura

La estructura de la plataforma Sentilo està formada per varis components que proporcionen els serveis.



Imatge 8. Estructura Plataforma Sentilo. Font Sentilo.io

Per poder funcionar la plataforma es val del següent components:

Servidor de publicació/subscripció per la comunicació amb els sensors i actuadors i poder subscriure els actuadors a les alertes.

Està basat en Redis, un motor de base de dades no relacional en memòria, basat en emmagatzemament en taules de tipus diccionari de clau/valor i dissenyat per actuar com a servidor d'estructures de dades. Com guarda les dades en memòria l'accés a les dades és més ràpid, però no te persistència en cas de fallida.

Per obtenir persistència guarda periòdicament les dades a disc, per recuperacions d'avant caigudes. A més a més disposa de una bitàcora de persistència on es guarden la llista de totes les comandes que han sigut executades.

Redis aporta també la funcionalitat de sistema de publicació i subscripció a coles de missatges.

La subscripció permet escoltar un canal i la publicació permet enviar missatges pel canal. El format utilitzat per la plataforma per enviar els missatges serà JSON (JavaScript Object Notation).

JSON: és un estàndard obert basat en text dissenyat per a intercanvi de dades llegible per humans. Amb JSON es representen estructures de dades simples i llistes associatives, anomenades objectes, es descriu al RFC:4627 [18].

Exemple d'un objecte JSON:

```
{  "precision": "zip",
  "Latitude": 37.7668,
  "Longitude": -122.3959,
  "Address": "",
  "City": "SAN FRANCISCO", }
```

Base de Dades NoSQL (Not only Structured Query Language) Mongo DB, per tenir una base de dades més eficient i elàstica.

MongoDB és un programari de codi obert, per a la creació i gestió de bases de dades orientada a documents, escalable, d'alt rendiment i lliure d'esquema programada en C++. El estar orientada a documents permet que moltes aplicacions puguin gestionar les dades de manera més natural, és el sistema de base de dades més popular de tipus NoSQL.

Gestor base de dades MySQL, sistema de gestió de bases de dades relacionals, que utilitza el llenguatge SQL. A la plataforma s'utilitza per poder exportar les dades dels esdeveniments a bases de dades relacionals externes.

Servidor Web Tomcat per allotjar l'aplicació Web Catalog, proporciona un entorn per al codi Java a executar en cooperació amb un servidor web autònom que permet un alt nivell de trànsit i alta disponibilitat.

3.3 Administració Sentilo

Les dues aplicacions que s'utilitzaran per la gestió de la plataforma son el API REST i el Catalog Web. La gestió consistirà en l'administració dels components hardware com sensors i actuadors, així com en la subscripció i publicació de alarmes.

3.3.1 API REST de Sentilo

La comunicació dels elements externs com els sensors, actuadors o d'altres aplicacions amb la plataforma ha de ser a través del protocol HTTP i es fa a través de la seva API REST [19].

REST és una arquitectura de programari pensada per sistemes distribuïts basats en hipermèdia. Utilitza XML, JSON, HTML i HTTP per gestionar els recursos mitjançant peticions HTTP. Tots els recursos estan identificats per URIs úniques que s'utilitzaran a les capçaleres HTTP per realitzar les accions, de manera que es poden crear, llegir, editar i eliminar recursos realitzant peticions HTTP. Al nostre cas els recursos seran sensors, actuadors i subscripcions a les publicacions de les alertes de la plataforma Sentilo y s'utilitzarà l'estàndard JSON per l'enviament de les estructures de dades.

Mitjançant la API es poden realitzar les següents funcions:

- Registrar aplicacions/mòduls i proveïdors/sensors al catàleg de la plataforma.
- Publicar, recuperar i esborrar dades, per enviar informació des de els sensors a les aplicacions o fer-ne peticions per obtenir-la.
- Publicar, recuperar i esborrar alarmes.
- Subscriure aplicacions/mòduls y sensors als esdeveniments de les alarmes, recuperar les subscripcions o cancel·lar-les.
- Publicar, recuperar i esborrar ordres, per posteriorment enviar ordres des de les aplicacions als sensors.

Per exemple, per publicar una nova alarma es pot fer a través d'un client REST, enviant una petició http PUT a `http://sentiloalbert.com/alarm/43` amb el cos del missatge amb les dades en format JSON:

```
{"message": "Límit de temperatura excedit: 42"}
```

Protocol HTTP

Hypertext Transfer Protocol (HTTP) es el protocol de comunicació que permet les transferències d'informació a la World Wide Web, defineix la sintaxi i la semàntica que utilitzen els elements software de l'arquitectura web per comunicar-se.

És un protocol que segueix l'esquema de petició-resposta entre un client i un servidor.

Per la realització del projecte és important conèixer l'estructura de les peticions i les respostes HTTP, ja que es crearan els missatges HTTP en text pla.

Els missatges segueixen la següent estructura:

- Lineal inicial: Si és una petició indica el mètode de petició seguit de la URL del recurs i la versió HTTP que suporta el client.
Si és una resposta, indica la versió del HTTP seguit del codi de resposta que indica que ha passat amb la petició.
Termina amb retorn de carro i salt de línia.
- Capçaleres: son metadades. Finalitzen amb una línia en blanc.
- Cos del missatge: Es opcional, incorpora les dades que s'intercanvien entre el client i el servidor, com per exemple les dades en format JSON.

Els codis de resposta indiquen que ha passat amb la petició. Els codis que utilitza la API REST de Sentilo son:

Error Code	HTTP	Descripció
200	Success	Petició acceptada i processada correctament.
4xx	Client Error	Error a la petició (Format incorrecte, falten paràmetres,...)

401	Unauthorized	Petició no autoritzada: credencials invàlids o buides
403	Forbidden	Acció de la petició no autoritzada.
5xx	Server Error	Error al processar la petició.

Mètodes HTTP

- GET: Sol·licita una representació d'un recurs específic.
- HEAD: demana una resposta idèntica a la de una petició GET, però sense el cos de la resposta.
- POST: S'utilitza per enviar dades de un recurs específic.
- PUT: Actualitza les representacions actuals d'un recurs.
- DELETE: Esborra un recurs específic.
- CONNECT: estableix un túnel fins al servidor identificat pel recurs
- TRACE: Realitza una prova de bucle de retorn de missatge.
- PATCH: s'utilitza per aplicar modificacions parcials a un recurs.

La API REST de Sentilo suporta el mètodes GET, PUT, POST i DELETE.

3.3.2 Catalog Web: Configuració xarxa de sensors a la plataforma

Per a la configuració de la plataforma s'utilitzarà l'aplicació web Catalog que proporciona Sentilo que permet administrar i monitoritzar els recursos de la plataforma i la seva activitat.



Imatge 9. Entorn gràfic de l'aplicació Catalog Web

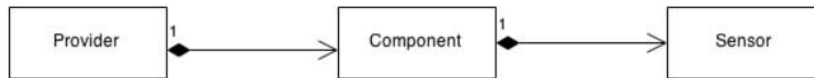
És necessari conèixer l'estructura proveïdors, components i sensors de Sentilo per poder gestionar els recursos:

- Aplicació/mòdul: entitat que consumeix les dades processades per la plataforma. Poden rebre dades dels proveïdors i sensors i enviar-los ordes, també poden realitzar subscripcions als esdeveniments.
- Proveïdor: entitat que representa un grup de components permeten comunicar-se amb Sentilo enviant dades i rebent comandes. Poden fer registres a la plataforma, subscriure's als esdeveniments i publicar dades.

- **Components:** Al context de Sentilo, els components tenen un significat especial, que no estan vinculats a la API REST, es a dir, els components no estan obligats a publicar o llegir dades. S'utilitzen els components en el catàleg per agrupar sensors compartint un conjunt de propietats com pot ser ubicació, proveïdor, energia, connectivitat.

Al projecte els components s'utilitzen per agrupar sensors per ubicació, creant un component per a cada dispositiu amb els sensors de qualitat d'aire, temperatura, humitat, so i il·luminació, es a dir, cada component està format per un Arduino i un sensor de cada tipus.

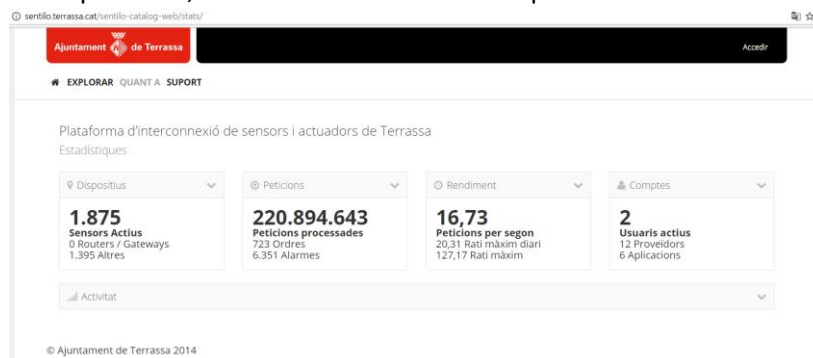
Tots els sensors han d'estar relacionats amb un component seguint les següents relacions:



- **Sensors:** Els sensors es podran crear amb el Catalog Web o directament mitjançant la API REST. De la mateixa manera que els proveïdors, poden fer registres a la plataforma, subscriure's als esdeveniments i publicar dades.

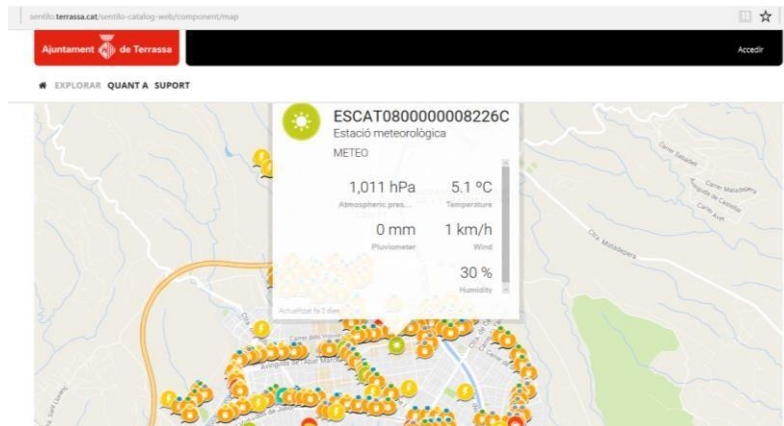
Una altra aplicació del Catalog Web és la publicació de les dades, que es poden trobar tres formats d'informació:

- **Estadístiques:** es poden veure el nombre de sensors donats d'alta a la plataforma i el nombre de peticions, així com el rendiment de la plataforma.



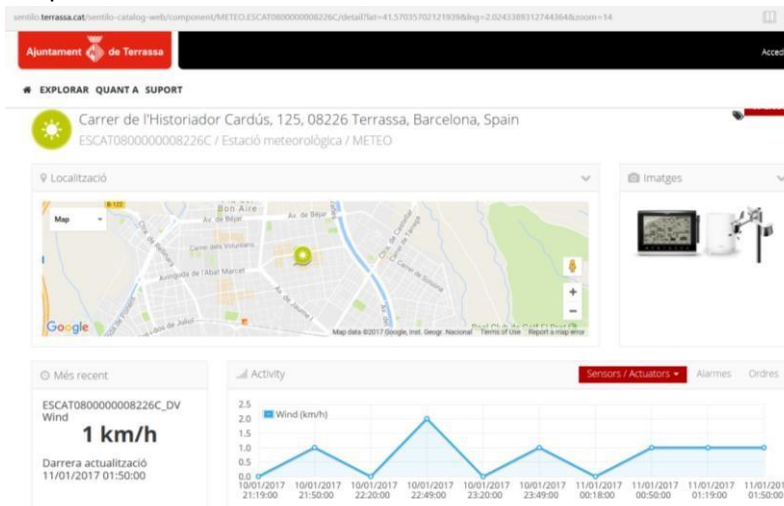
Imatge 10. Informació de les estadístiques de Sentilo Terrassa

- **Vista universal:** Mapa amb les dades de Google Maps amb la posició de cada component, clicant sobre el component podem veure la última dada capturada pels sensors.



Imatge 11. Vista universal amb les dades d'un sensor de Sentilo Terrassa

- Historial de dades captades per un component o sensor: entrat al component es poden seleccionar els sensors que el componen per veure l'historial de dades captades pel sensor.



Imatge 12. Historial de dades captades per un sensor de Sentilo Terrassa

3.3.3 Alertes

A l'administrador del Catàleg Web o mitjançant l'API es poden crear les alarmes internes mitjançant regles que es defineixen a través d'expressions simples. A la següent taula es mostren les expressions simples que la plataforma es capaç d'avaluar:

Id	Description	Expression value
GT	Greater than <expression>	Any numerical value
GTE	Greater than or equal <expression>	Any numerical value
LT	Less than <expression>	Any numerical value
LTE	Less than or equal <expression>	Any numerical value
EQ	Equal <expression>	Any value
CHANGE	Any change	Not apply here
CHANGE_DELTA	Any variation greater to delta <expression>	Any numerical value between 0 and 100
FROZEN	No data received in <expression> minutes	Any numerical value

Tabla 1. Expressions per avaluar alertes internes a Sentilo. Font Sentilo.io .

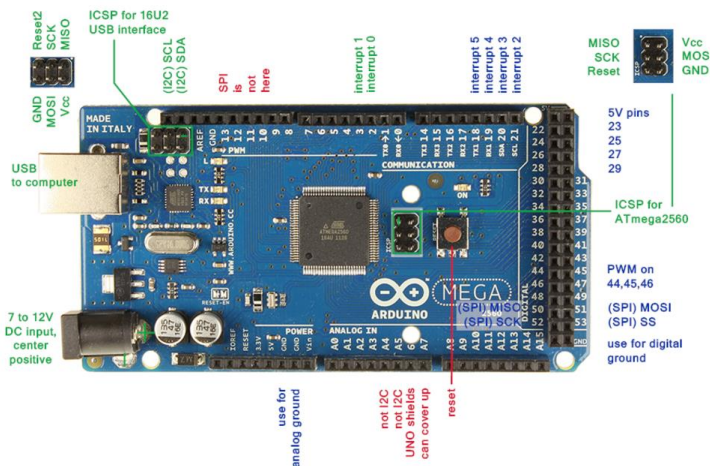
Aquestes alertes es poden associar a un proveïdor, component o sensor determinat.

Es poden crear alertes per a grups de sensors, agregant una regla d'alerta per a tot un conjunt de sensors.

4. Arduino

4.1 Estructura

Per crear cada component s'utilitzarà una placa Arduino, el model Mega 2560, que és la que recomana la documentació tècnica de Sentilo [20] per tenir major memòria de programació, 256KB, que el model UNO, 32KB.



Imatge 13. Arduino Mega 2560. Font: <https://forum.arduino.cc/>

Les característiques tècniques d'Arduino Mega 2560 son:

- Microcontrolador: ATmega2560
- Memòria flash: 256 KB
- SRAM: 8 KB

- EEPROM: 4 KB
- Velocitat de rellotge: 16 MHz
- Tensió de alimentació: 5V
- Tensió d'entrada recomanada: 7-12V
- Pins digitals: 54 (14 amb PWM)
- Entrades analògiques: 16
- Corrent màxima per pin: 40 mA
- Corrent màxima pel pin 3.3V: 50 mA

La raó de no utilitzar el Smart Citizen kit del projecte és que la placa utilitzada al kit és una Arduino Leonardo [21], amb el xip ATMEGA32U4, que posseeix 32 KB de memòria, el mateix motiu pel qual no s'utilitza el model yún [22] que incorpora connectivitat WiFi però que també disposa de 32 KB de memòria per programació al utilitzar el xip ATMEGA32U4.

4.2 Components i actuadors

La comunicació dels components i actuadors amb la plataforma s'ha realitzat en primera instància mitjançant Ethernet ja que les llibreries de Sentilo estan creades per la Shield Ethernet i posteriorment s'ha implementat la comunicació dels components mitjançant la tecnologia inalàmbrica WiFi 802.11 sense utilitzar les llibreries de Sentilo.

Les funcions dels Arduinos utilitzats als components i actuadors son:

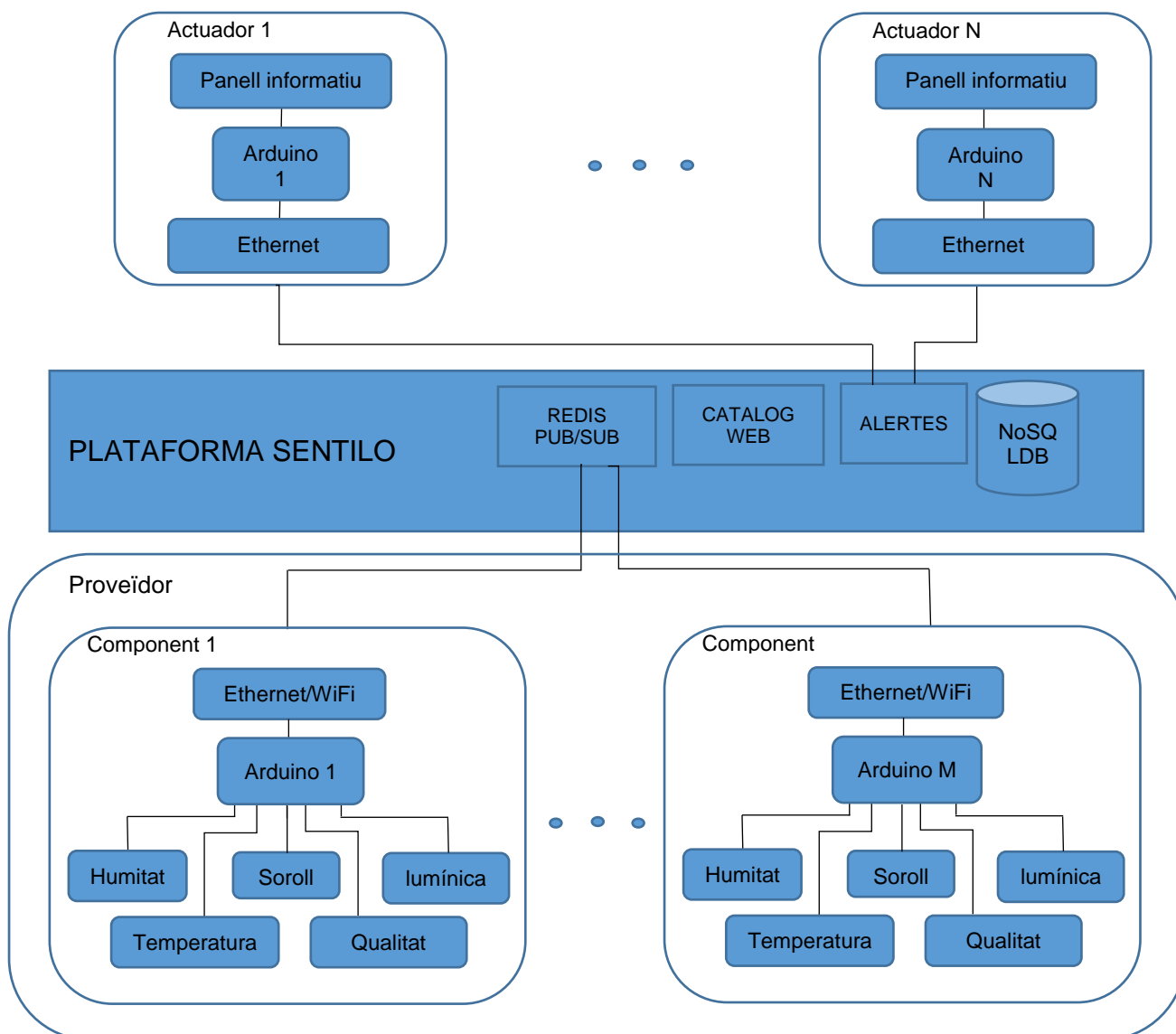
- Registre dels sensors a la plataforma Sentilo
- Enviament de les dades adquirides pels sensors a la plataforma Sentilo
- Subscripció a les alertes
- Processament de les alertes rebudes
- Publicació dels missatges als panells informatius

Cada punt d'adquisició de dades està format per un component on els sensors a utilitzar, i per tant connectat a cada Arduino inclòs, son:

- Sensor de temperatura i humitat.
- Sensor de qualitat de l'aire.
- Contaminació lumínica: Fotodíode
- Soroll: sensor amb micròfon

Per implementar els actuadors també s'utilitzaran Arduino Mega, que mitjançant la subscripció als esdeveniments rebran les alertes i les processaran. El processat de les alertes permetrà mostrar als panells informatius un determinat missatge informatiu o restrictiu segons el cas. Per al prototipatge s'han utilitzat displays LCD connectats a l'Arduino.

L'esquema final de la realització del projecte queda de la següent manera:



5. Sensors

Mitjançant els sensors es vol monitoritzar la contaminació acústica, la temperatura, la humitat, la qualitat de l'aire i la contaminació lumínica.

Per l'elecció dels sensors s'han tingut en compte la facilitat d'interconnexió amb Arduino, el cost econòmic, la precisió i els rangs de funcionament.

5.1 Micròfon

Com s'ha comentat la monitorització de la contaminació acústica és un dels principals objectius del projecte. Degut a l'impacte que té a la vida de les persones.

Es planteja la monitorització del soroll a través de micròfons, que són capaços de captar el so al seu voltant i transformar-ho en informació, en aquest cas voltatge a partir del qual es calcularà el valor de so en decibels.

Per la realització del projecte s'han tingut en compte dues opcions, el micròfon KY-038 i el Sparkfun BOB-09964.

El senyal de sortida del micròfon no és prou alta com per poder mesurar el nivell directament i es necessari amplificar-la. L'avantatge del model Sparkfun BOB-09964 sobre el mòdul KY-038 és que és un micròfon preamplificat i per tant en el seu cas es pot connectar directament a Arduino sense la necessitat d'un amplificador afegit.

L'amplificador del mòdul és un OPA344, té un guany de 20dB i voltatges d'operació de 2,7V a 5,5V.



Imatge 14. Sparkfun BOB-09964. Font: <https://www.sparkfun.com>

5.2 Sensor il·luminació

Per la contaminació lumínica no es requereix de gran precisió, doncs l'objectiu es detectar zones a les quals durant la nit la il·luminació es considera excessiva o zones a les quals es pot optimitzar la il·luminació mitjançant sistemes d'enlluernament intel·ligent.

Després de sondejar el mercat, es van escollir dos possibles sensors, una fotoresistència GL5528 i el sensor de il·luminació TSL2561 .



Imatge 15. LDR GL5528



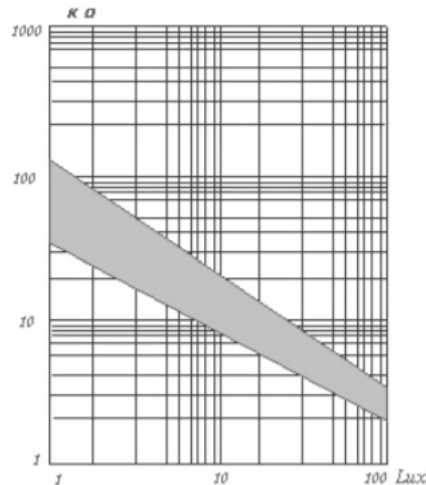
Imatge 16. Mòdul sensor il·luminació TSL2561

Degut a la baixa precisió necessària i al preu s'ha escollit la fotoresistència, que després d'un procés de calibració amb un luxòmetre ens permetrà obtenir un valor aproximat de la il·luminació.

El funcionament de les fotoresistències es basa en que a la seva composició entren materials que posseeixen pocs electrons lliures quan es troben en condicions de fosc,

però el nombre d'electrons lliures augmenta de forma considerable quan s'il·lumina el dispositiu. En conseqüència, la conductivitat creix i per tant la resistència disminueix.

A continuació es mostra la corba de resposta il·luminació-resistència de la fotoresistència GL5528 on es pot observar que la seva precisió es baixa. Per una resistència de 10K Ω la il·luminació està entre 80 i 110 lux.



Imatge 17. Corba il·luminació-resistència GL5528

5.3 Qualitat de l'aire

Per l'estudi de la qualitat de l'aire s'han tingut en compte els sensors de PM10, els de NO₂, els de CO₂ i els de CO.

PM10

Durant l'estudi dels sensors de PM10 disponibles al mercat, es va detectar la poca oferta i el seu alt cost.

A primera instància el model escollit va ser el GP2Y1010AUOF, un sensor de qualitat de l'aire òptic de la companyia SHARP, amb un preu de 15€, dissenyat per detectar partícules de pols i utilitzat sobretot a aplicacions interiors com aires condicionats.



Imatge 18. Sensor Sharp GP2Y1010AUOF

El fet que detecti pols i fum fa que detecti partícules més petites a 10 μ m i que no sigui realment un sensor de PM10. Això unit al pressupost i que les seves dimensions (46.0 x 30.0 x 17.6 mm) son superiors a tot el conjunt de la resta d'equipament va fer descartar el seu ús.

NO₂

L'oferta de sensors que mesuren la variable NO₂, diòxid de nitrogen, és molt escassa i cara. La opció que es va trobar van ser el MICS 2714, un sensor al qual la seva resistència varia en funció de la concentració de NO₂ al que s'exposa. El seu preu de 21€ el va descartar.



Imatge 19. Sensor NO₂ MICS 2714

CO₂

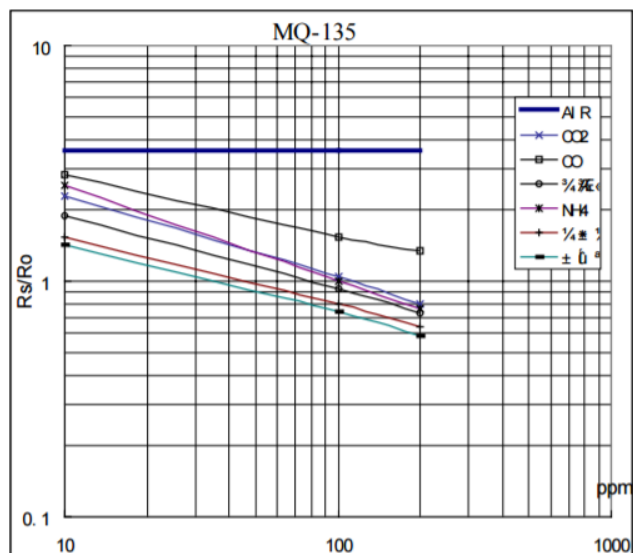
Una vegada estudiat el mercat de sensors de CO₂, es pot observar que el seu preu no es menor que el dels sensors nombrats anteriorment.

Els estudiats van ser els sensors:

Sensor de CO₂ MG811 amb un preu de 45\$

Sensor de CO₂ MH-Z14A amb un preu de 33\$

Finalment, prioritant el preu, s'ha optat per utilitzar un sensor de qualitat d'aire que detecta diferents agents a l'aire, el MQ-135, que detecta NH₃, alcohol, benzé, CO₂ i CO. La seva corba característica es mostra a continuació.



Imatge 20. Mòdul MQ-135 i corba característica dels diferents agents

Es pot observar que aquest sensor es sensible en similar proporció als gasos mencionats i d'aquesta manera podem determinar si l'aire es net. Tot i així el CO₂, dels gasos que

detecta el sensor, és el més abundant a l'atmosfera amb uns 400ppm aproximadament. Això significa que en una atmosfera normal el sensor detecta principalment CO₂.

A aquest projecte s'ha realitzat una estimació del CO₂ per obtenir una equació a partir d'una regressió a partir de la lectura obtinguda del sensor MQ135 i la seva corba característica.

5.4 Temperatura i humitat

Per mesura la temperatura i la humitat s'ha utilitzat el sensor DTH22, que permet mesurar tots dos paràmetres i existeix la seva pròpia llibreria per Arduino que facilita la mesura.



Imatge 21. Sensor temperatura i humitat DTH22

Les especificacions del DTH22 son:

- Mesura de temperatura entre -40 a 125, amb una precisió de 0.5°C
- Mesura d'humitat entre 0 a 100%, con precisió del 2-5%.
- Mostreig de 2 mostres por segon (0.5 Hz)

6. Desenvolupament del projecte

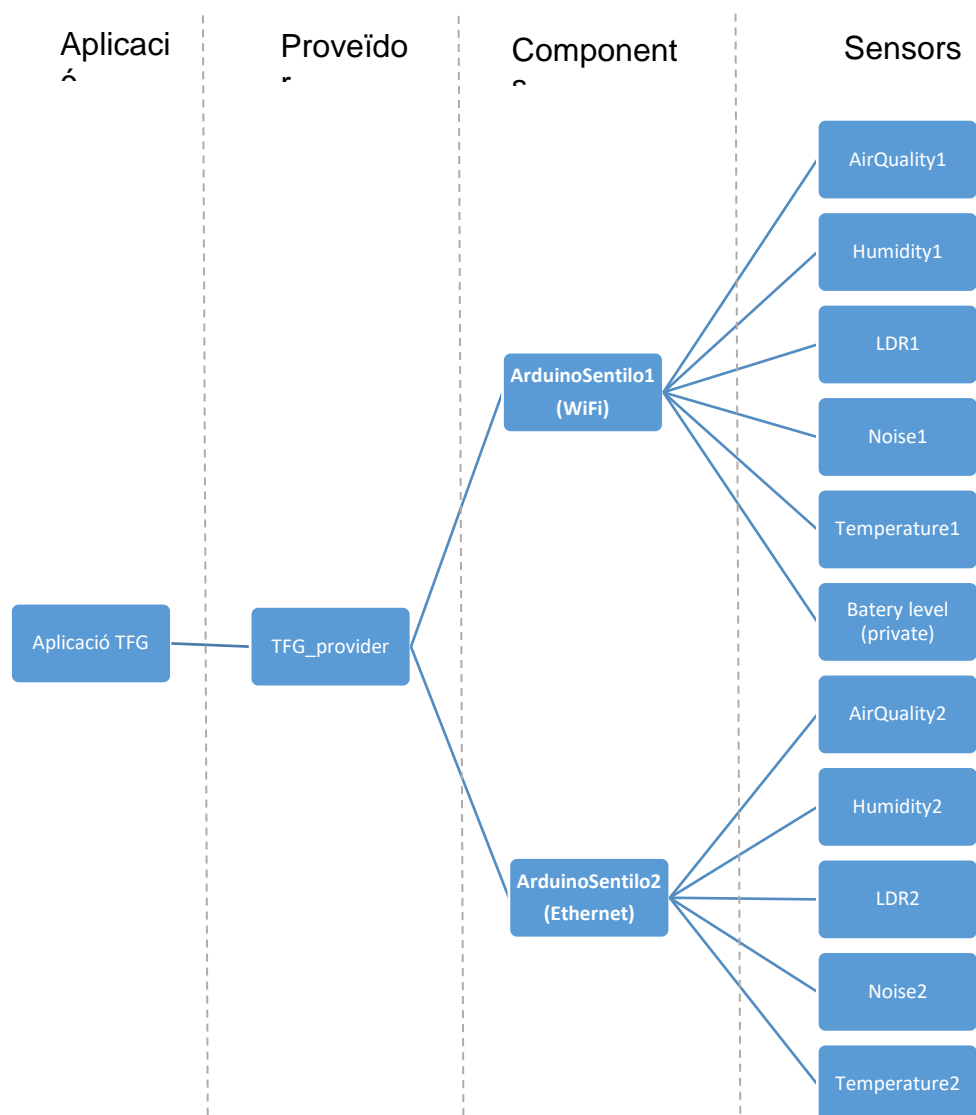
Els passos seguits han estat:

1. Configuració del catàleg Web
2. Muntatge dels components
3. Muntatge dels actuadors
4. Processament de les alarmes per part dels actuadors

6.1 Catàleg Web

La configuració del catàleg web consta de la creació de tots els elements de l'estructura del catàleg, la creació de les alertes i alarmes i la subscripció dels actuadors a les diferents alarmes a través del servidor de publicació/subscripció de la plataforma.

Al següent esquema es poden observar tots els elements que conformen l'estructura del Catàleg Web:



Imatge 22. Estructura catalog Web

Per la configuració del Catalog Web s'ha creat una aplicació anomenada TFG, a la qual se li assigna un Authorization Token que proporciona seguretat als diferents serveis de la API REST, sent necessari enviar aquest Authorization Token a la capçalera de les peticions HTTP amb paràmetre anomenat IDENTITY_KEY.

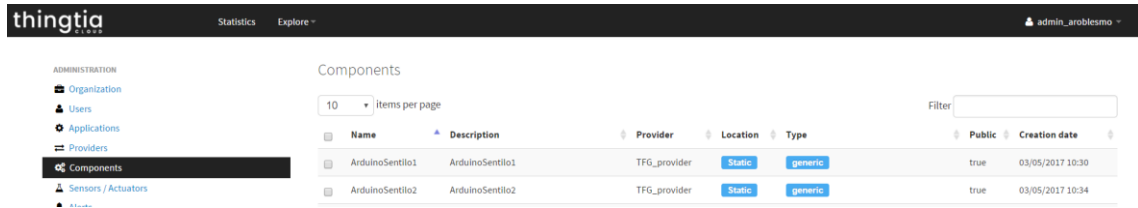
Per tant a totes les peticions HTTP que es realitzen a la API REST s'ha d'incloure el camp IDENTITY_KEY amb el valor del Authorization Token de l'aplicació TFG.

S'ha creat el proveïdor TFG_provider per agrupar tots els components creats amb el mateix objectiu de mesura de contaminació. A aquest TFG_provider se li han assignat permisos d'administració sobre l'aplicació TFG per poder publicar, gestionar i demanar informació de l'aplicació des de la API REST.

Components

A catàleg Web s'han configurat dos components, cadascun d'ells amb els seus 5 sensors.

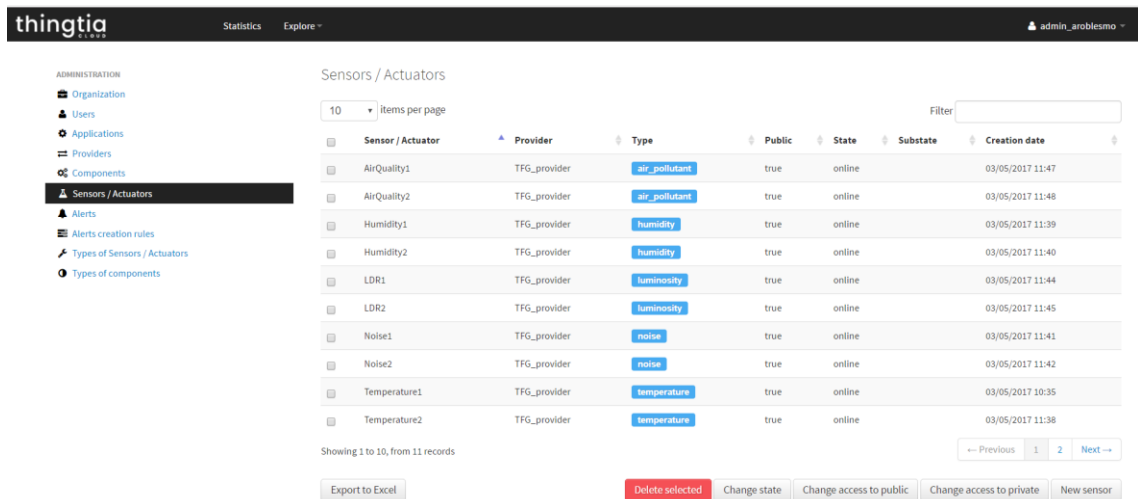
Vista del Catàleg Web:



Name	Description	Provider	Location	Type	Public	Creation date
ArduinoSentilo1	ArduinoSentilo1	TFG_provider	Static	generic	true	03/05/2017 10:30
ArduinoSentilo2	ArduinoSentilo2	TFG_provider	Static	generic	true	03/05/2017 10:34

Sensors

Vista dels sensors creats a Catalog Web:



Sensor / Actuator	Provider	Type	Public	State	Substate	Creation date
AirQuality1	TFG_provider	air_pollutant	true	online		03/05/2017 11:47
AirQuality2	TFG_provider	air_pollutant	true	online		03/05/2017 11:48
Humidity1	TFG_provider	humidity	true	online		03/05/2017 11:39
Humidity2	TFG_provider	humidity	true	online		03/05/2017 11:40
LDR1	TFG_provider	luminosity	true	online		03/05/2017 11:44
LDR2	TFG_provider	luminosity	true	online		03/05/2017 11:45
Noise1	TFG_provider	noise	true	online		03/05/2017 11:41
Noise2	TFG_provider	noise	true	online		03/05/2017 11:42
Temperature1	TFG_provider	temperature	true	online		03/05/2017 10:35
Temperature2	TFG_provider	temperature	true	online		03/05/2017 11:38

Alertes

S'han configurat les alertes corresponents als nivells de soroll i qualitat de l'aire. Les alertes, com s'ha vist a l'apartat 5.3.3, es poden crear per un valor igual o superior a un altre, però no per un valor entre dos llindars. A més, si es creen dos alertes sobre un mateix paràmetre es notifica la més restrictiva, no les dues.

Per tant s'ha creat una sola alerta per a cada sensor de soroll i qualitat de l'aire i son els actuadors els que mitjançant la subscripció a les alarmes reben les notificacions, processen la informació i divideixen les alertes de so i qualitat de l'aire en tres nivells cadascuna.

Identifier	Typology	Trigger type	Active	Creation date
a1	Internal	G T(40)	true	20/03/2017 18:34
a2	Internal	G T(80)	true	23/05/2017 19:54
quality_air1	Internal	G T(500)	true	23/05/2017 20:01
quality_air2	Internal	G T(500)	true	23/05/2017 20:02
s1	Internal	G T(70)	true	23/05/2017 21:31
sound_level1	Internal	G T(70)	true	24/05/2017 16:51
sound_level2	Internal	G T(70)	true	23/05/2017 20:00

El detall de la configuració de les alertes disposa del següent format:

Identifier:
 Name:
 Description:
 Active:
 Typology:
 Provider:
 Component:
 Sensor id / Actuator:
 Expression
 Trigger type:
 Expression to evaluate:

Alarmes

Les alarmes han d'estar relacionades amb una alerta creada prèviament, s'ha creat una alarma per a cada una de les alertes configurades.

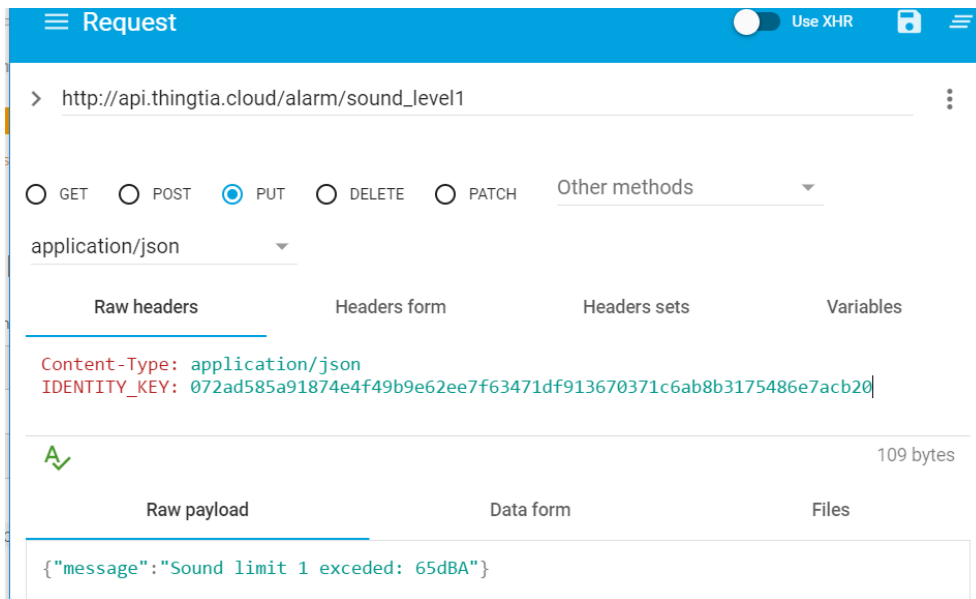
No es poden crear mitjançant el Catalog Web, han de ser creades amb la API REST mitjançant una petició HTTP PUT a la API REST amb el format:

`http://api.thingtia.cloud/alarm/<nom_alerta_relacionada>`

amb el missatge que es vol visualitzar al contingut en format JSON, per exemple:

`{"message": "Sound limit exceeded: 65 dBA"}`

Per fer-ho s'ha utilitzat l'aplicació Advanced REST Client.



Subscripció

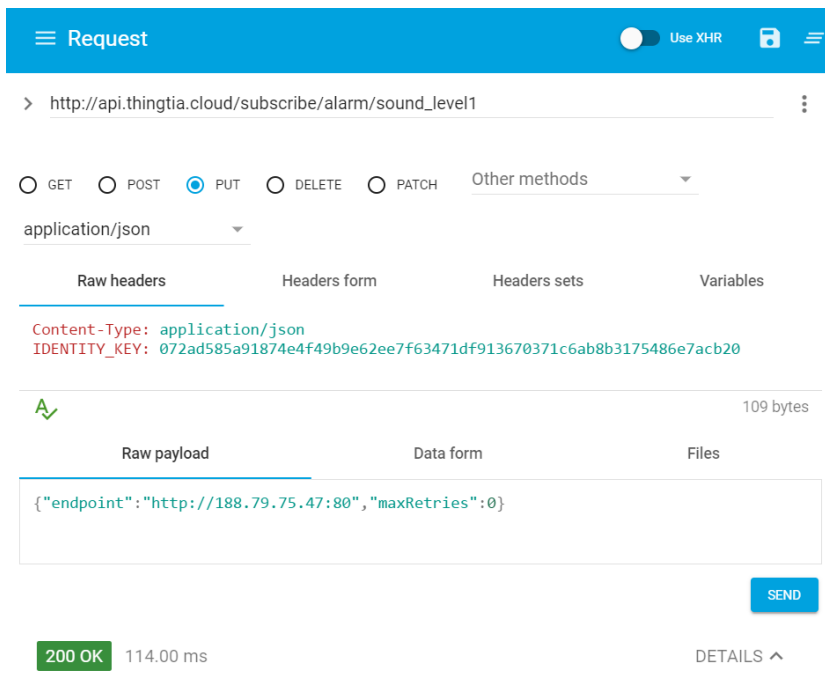
Per a que els Actuadors rebin les alarmes existeixen dues possibilitats, una seria anar fent consultes (polling) de peticions HTTP a la plataforma Sentilo per saber si alguna alarma ha canviat els seu estat i l'altra, la escollida, utilitzar el servidor de Publicació/Subscripció de la plataforma i subscriure als actuadors a les alarmes convenients per a que cada vegada que existeixi aquesta alarma se li notifiqui.

Per realitzar la subscripció s'ha de fer a través de la API REST amb una petició amb el format:

http://api.thingtia.cloud/subscribe/alarm/<nom_alerta_relacionada>

Amb l'adreça i el port habilitat a l'actuador, en aquest cas la IP 188.79.75.40 al que se li ha d'enviar les notifikacions al missatge en format JSON.

{"endpoint": "http://188.79.75.47:80"}



Per veure les subscripcions ho poden fer enviant una petició GET a la REST API amb el següent format:

`http://api.thingtia.cloud/subscribe`

La resposta serà :

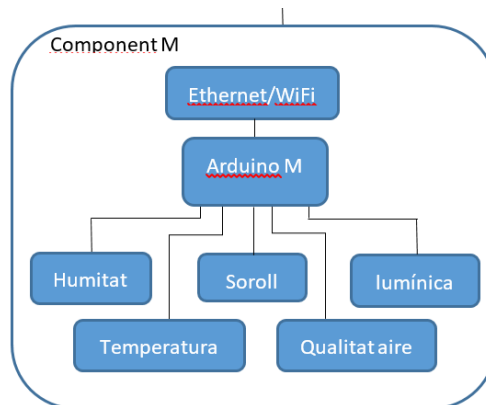
```
{
  - "subscriptions": [Array[5]
    -0: {
      "endpoint": "http://188.79.75.47:80",
      "type": "ALARM",
      "alert": "sound_level2",
      "maxRetries": 3,
      "retryDelay": 5
    },
    -1: {
      "endpoint": "http://188.79.75.47:80",
      "type": "ALARM",
      "alert": "sound_level1",
      "maxRetries": 3,
      "retryDelay": 5
    },
    -2: {
      "endpoint": "http://188.79.75.47:80",
      "type": "ALARM",
      "alert": "quality_air1",
      "maxRetries": 3,
      "retryDelay": 5
    },
    -3: {
      "endpoint": "http://188.79.75.47:80",
      "type": "ALARM",
      "alert": "quality_air2",
      "maxRetries": 3,
      "retryDelay": 5
    },
    -4: {
      "endpoint": "http://188.79.75.47:80",
      "type": "ALARM",
      "alert": "a1",
      "maxRetries": 3,
      "retryDelay": 5
    }
  ],
}
```

6.2 Prototip Arduino Components

Per la realització del projecte s'han realitzat dos prototips amb Arduino diferenciats. Per una banda, els components i per una altra banda els actuadors.

La funció dels components es captar les dades a través dels sensors i enviar les dades a la plataforma Sentilo. Cada component està format per un Arduino Mega, una shield amb els 5 sensors especificats a l'apartat anterior i una xip o shield de telecomunicacions.

La seva estructura queda de la següent manera:

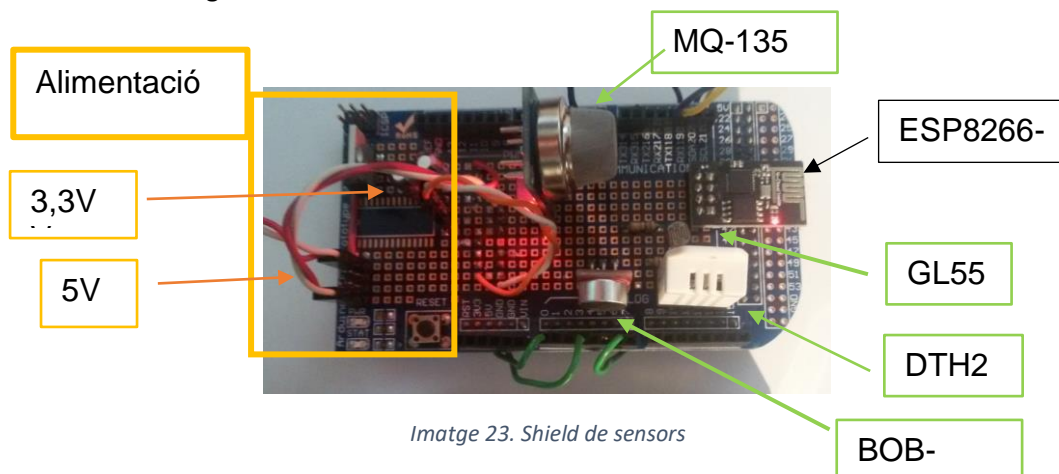


S'han desenvolupat dos models:

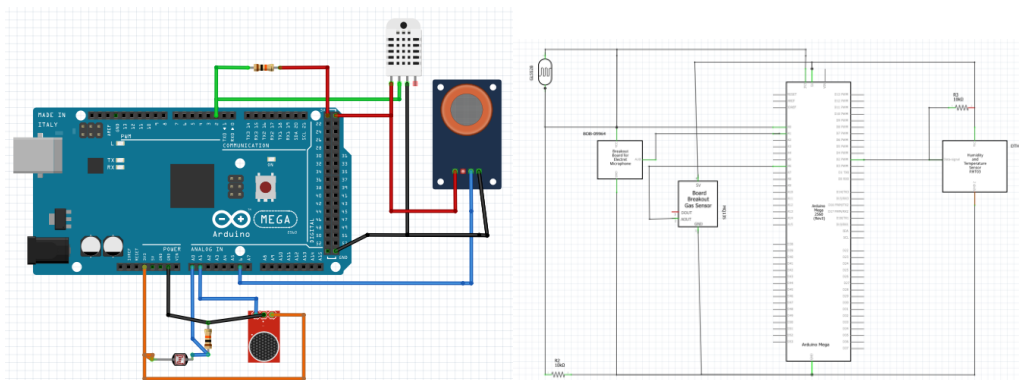
- Model amb shield Ethernet: utilitza la llibreria que Sentilo proporciona amb els mètodes i les funcions necessàries per la comunicació, incloent les peticions HTTP.
- Model amb xip Wifi: no utilitza la llibreria que Sentilo proporciona i s'ha implementat la programació de l'enviament de peticions HTTP PUT per la publicació de les dades i la recepció de les seves respostes.

En aquest cas el xip WiFi està inclòs a la mateixa shield dels sensors.

Destacar que els dos model poden utilitzar la mateixa shield de sensors creada. La placa queda distribuïda de la següent manera:



L'esquema de la connexió dels sensors queda de la següent manera:



Imatge 24. Connexió i esquema dels sensors i l'Arduino

6.2.1 Adquisició de dades

Sensor de temperatura i humitat DTH22

El sensor DTH22 disposa de la seva pròpia llibreria que proporciona els mètodes per obtenir els valors de temperatura en graus Celsius i els valors d'humitat en %. Els mètodes són:

- `dht.readHumidity();`
- `dht.readTemperature();`

Sensor d'il·luminació. Fotoresistència GL5528

Els LDR varien la seva resistència en funció de la quantitat de LUX en forma exponencial, que es converteix en lineal a l'escala logarítmica.

S'han calculat els lux a partir dels valors de la resistència de fosc, la resistència a la llum a 10 lux i el paràmetre α que proporciona el fabricant al datasheet.

La fórmula utilitzada ha estat: $lux = (L * A * 10) / B * Rc * (1024 - L)$; on L és la lectura amb el Arduino i el 1024 correspon als 10 bits del convertidor A/D del Arduino.

No obstant degut al comportament potencial les petites diferències en termes de resistència suposen grans variacions a la mesura de lux i per tant també s'ha realitzat una calibració utilitzant el luxòmetre CHAUVIN ARNOUX C.A 813 light-meter.

El valors de les mesures i el procediment es poden consultar al l'annex.

Sensor de so.

El sensor de so proporciona una sortida dependent de la pressió SPL.

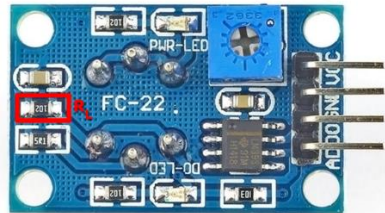
El so consta de pressions i depressions, de manera que la forma de mesurar els dB s'ha de realitzar tenint en compte el offset del dispositiu i calculant el valor eficaç utilitzant una finestra de un segon.

Posteriorment, s'ha aplicat un factor de correcció realitzant una calibració amb el sonòmetre de classe 2 Proskit MT-4008 una vegada el dispositiu es troba dins de la caixa final.

El valors de les mesures i el procediment es poden consultar a l'annex.

Sensor de qualitat de l'aire MQ-135.

El seu funcionament es basa en que internament posseeix un escalfador encarregat d'augmentar la temperatura interna i amb això el sensor pugui reaccionar als gasos NH₃, NO_x, alcohol, bencé, CO₂ i CO provocant un canvi de valor en la resistència. Es necessària una resistència de carrega per tancar el circuit i fer un divisor de tensió. El mòdul utilitzat ja disposa de la resistència de carrega i té un valor de 1KΩ.



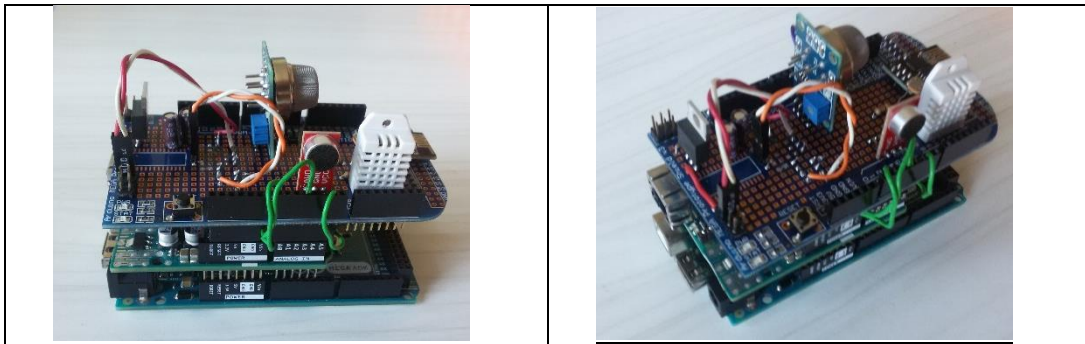
Per a que la sortida sigui estable es necessari un tems d'escalfament de 24 hores.

Per realitzar la calibració s'ha realitzat una regressió de la corba que proporciona el datasheet per obtenir la equació que relaciona la resistència mesurada amb el valor en ppm.

Per fer-ho s'ha utilitzat la llibreria pel MQ135 de G.Krocker i la seva funció `get.ppm()`

El procediment de càlcul es pot consultar a l'Annex.

6.2.2 Model Ethernet



Imatge 25. Component model Ethernet

Per la implementació del model amb ethernet s'ha utilitzat la Arduino Shield Ethernet 2 per a la qual s'ha desenvolupat la llibreria de Sentilo. S'ha utilitzat per tant la llibreria Ethernet.h i SentiloClient.h.

EthernetShield:



Imatge 26. Ethernet Shield 2 de Arduino

Les característiques principals de la EthernetShield 2 para Arduino son:

- Velocitat de connexió de 1/100Mb
- Lector de targeta MicroSD
- Connector Ethernet RJ45
- Alimentació a 5V i possibilitat d'alimentació mitjançant POE
- Connexió amb Arduino mitjançant un port sèrie.

Programació:

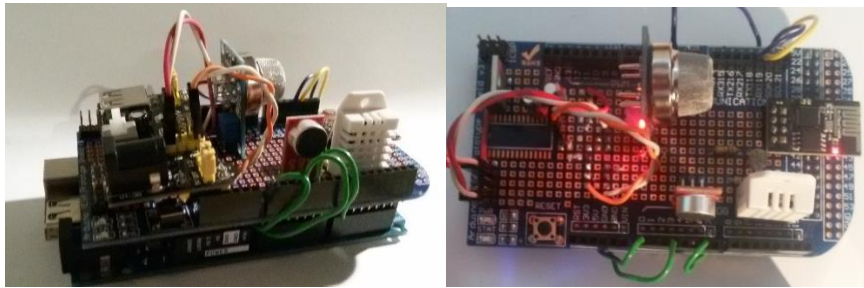
De la llibreria SentiloClient que fa crides a la llibreria Ethernet s'han utilitzat els següent mètodes i funcions:

Mètodes/funcions	Comentari
<pre>SentiloClient = SentiloClient(ip,port)</pre>	<p>Constructor per crear un objecte de la classe sentiloClient. L'objecte SentiloClient està format pels atributs privats host, port, nombre de capçaleres i tipus de contingut, en aquest cas application/json i pels atributs públics referents als sensors on s'inclou dades com el tipus de sensor, el component al que pertany la localització o la unitat de mesura y les observacions.</p>
<pre>sentiloClient.dhcp();</pre>	<p>Assigna una adreça MAC al client.</p>
<pre>setupSetiloClient();</pre>	<p>Mitjançant la llibreria de Ethernet configura connexió de xarxa, obtenint una adreça IP.</p>
<pre>SentiloClient::Observation;</pre>	<p>S'encarrega de crear l'observació i preparar la petició HTTP creant el contingut del missatge en format json. Ho fa mitjançant el codi:</p> <pre>String message = "{\"observations\":[{"; message += "\"value\":\","+String(observation.v alue)+"\""; message += ",\"timestamp\":\","+String String(observation.timestamp)+"\""; message += "}]}"</pre>

<pre> statusCode = sentiloClient.publishObservation(providerId, sensorId1, observation, apiKey, response); </pre>	<p>Primer prepara la capçalera de la petició HTTP cridant a la funció request, que afegeix la ruta, la Autoritzation Token per poder fer la petició i l'objecte observació que s'ha crea mitjançant el SentiloClient.Observation.</p> <p>Ho fa mitjançant el codi:</p> <pre> write(method); write(" "); write(path); write(" HTTP/1.1\r\n"); for(int i=0; i<num_headers; i++){ write(headers[i]); write("\r\n"); } write("Host: "); write(host); write("\r\n"); write("Connection: close\r\n"); </pre> <p>Posteriorment realitza la petició HTTP PUT cridant a la funció sentiloClient.put(path, body, response)</p>
--	---

Tabla 2. Llibreria sentiloClient

6.2.3 Model Wifi

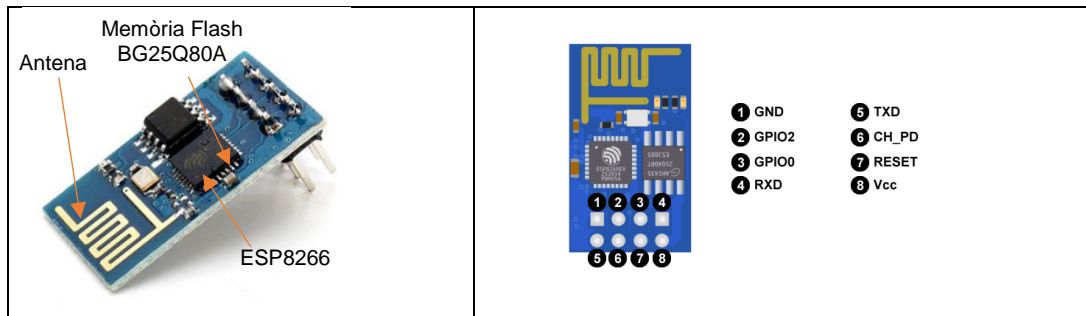


Imatge 27. Component model WiFi

El model amb WiFi era un altre dels objectius del projecte. Per realitzar-ho s'ha utilitzat un mòdul ESP8266-01.

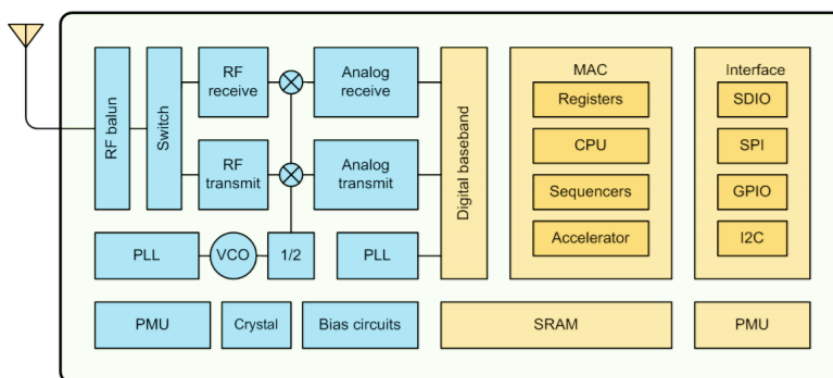
ESP8266

Mòdul que permet la connectivitat WiFi 802.11 b/g/n a baix cost i amb un espai reduït. Integra la pila de protocol HTP/IP, suporta WPA i WPA2, inclou els seu propi processador de 32 bits i memòria ROM, RAM i flash, interfícies de comunicació Sèrie i I2C (Integrated Circuits bus) i té disponibles dos pins GPIO (General Porpouse Input Output) digitals on es poden connectar diferent sensors i actuadors.



Imatge 28. ESP8266-01

A més, disposa de tota la electrònica necessària per la comunicació WiFi, com és l'antena, el balun de RF, el mòduls de transmissió i recepció amb els amplificadors de baix soroll i els filtres.



Imatge 29. Diagrama de blocs del ESP8266

Les seves característiques principals son:

- Compatible amb WiFi b/g/n
- Potència de transmissió b/g/n: 20 /27 /14 dBm
- Sensibilitat b/g/n : -91 / -75 / -71 dBm
- CPU Tensilica L 106 32 bits
- Voltatge d'operació entre 3V i 3,6V
- Corrent d'operació 80mA
- Suporta IPv4 i els protocols TCP/UDP/HTTP/FTP
- Pot treballa en mode accés point o estació Wifi

El mòdul es pot utilitzant programant la seva pròpia memòria flash o mitjançant sistemes externs que el governin a partir d'un processador, en aquest cas s'utilitza Arduino per controlar mitjançant les comandes AT via port sèrie el mòdul WiFi.

Programació:

Per realitzar la programació del model amb WiFi no s'ha utilitzat cap llibreria específica pel mòdul WiFi ni per la creació de les peticions HTTP. S'ha realitzat la programació gestionant el

ESP8266 enviant les comandes necessàries des del Arduino pel port Serial 1 que correspon als pins 18 i 19 al Arduino Mega i creant els datagrames de les peticions HTTP mitjançant strings.

A l'annex es poden observar les comandes AT del dispositiu, a continuació es descriuen les només les utilitzades a la programació:

Comanda	Comentari
AT	Retorna OK si el ESP s'ha iniciat correctament.
AT+CWJAP="SSID", "Password"	Per realitzar la unió a un punt d'accés WiFi.
AT+CWMODE=1	Amb el valor 1 s'especifica que treballi en mode estació WiFi.
AT+ CIPMUX=0	Amb el valor 0 s'especifica que només es permet una connexió.
AT+CIPSTART="TCP", "api.thingtia.cloud",80	Obre una connexió TCP amb l'adreça api.thingtia.cloud en el port 80.
AT+CIPSEND="longitud"	Abans d'enviar un paquet s'ha d'especificar la longitud del mateix. Posteriorment al rebre > es pot enviar la petició HTTP en format text.
AT+CIPCLOSE	

Tabla 3. Comandes AT per ESP8266

Peticions HTTP:

Per enviar missatges i publicar-los al Sentilo es necessari realitzar una petició HTTP PUT.

Al model WiFi per aprofitar l'energia s'envien les dades de forma conjunta a la mateixa. Per crear el datagrama s'ha creat la capçalera i la cadena de caràcters amb les dades en format JSON.

Cada 30 segons s'envien les dades dels sensors de so, d'il·luminació i el nivell de bateria i cada 15 minuts s'envien també les de qualitat de l'aire, temperatura i humitat, que son valors més estables en el temps.

El format que han de seguir el missatges és el següent:

Capçalera

```
PUT /data/TFG_provider/ HTTP/1.1
Host: api.thingtia.cloud
content-length: 129
user-agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/58.0.3029.110 Safari/537.36
accept: application/json
accept-encoding: gzip, deflate
accept-language: en-US,en;q=0.8
Content-Type: application/json
IDENTITY_KEY: 072ad585a91874e4f49b9e62ee7f63471df913670371c6ab8b3175486e7acb20
```

```
{"sensors":[{"sensor":"Temperature2","observations":[{"value":"28.00"}]},{"sensor":"Humidity2","observations":[{"value":"34"}]}}
```

Esperando respuesta...

Contingut

```
+IPD,161:HTTP/1.1 200 OK
Server: nginx/1.4.6 (Ubuntu)
```

Capçalera:

A la capçalera els paràmetres destacats són el tipus de petició, l'adreça a la que es realitza, la longitud de les dades, el tipus d'aplicació i el format del contingut. Destacar que, com s'ha comentat a l'apartat Catalog Web, s'ha d'incloure el IDENTITY_KEY amb el valor del Authorization Token de l'aplicació TFG.

```
322
323 //Creamos el encabezado de la petición PUT http
324 String peticiónHTTP= "PUT /data/TFG_provider/";
325     peticiónHTTP=peticiónHTTP+" HTTP/1.1\r\n";
326     peticiónHTTP=peticiónHTTP+"Host: api.thingtia.cloud\r\n";
327     peticiónHTTP=peticiónHTTP+"content-length: "+String(sensors.length())+"\r\n";
328     peticiónHTTP=peticiónHTTP+"user-agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/58.0.3029.110 Safari/537.36\r\n";
329     peticiónHTTP=peticiónHTTP+"accept: application/json\r\n";
330     peticiónHTTP=peticiónHTTP+"accept-encoding: gzip, deflate\r\n";
331     peticiónHTTP=peticiónHTTP+"accept-language: en-US,en;q=0.8\r\n";
332     peticiónHTTP=peticiónHTTP+"Content-Type: application/json\r\n";
333     peticiónHTTP=peticiónHTTP+"IDENTITY_KEY: 072ad585a91874e4f49b9e62ee7f63471df913670371c6ab8b3175486e7acb20\r\n\r\n";
334
```

Contingut del missatge per enviar les dades de tots els sensors en format JSON:

```
315     sensors="{\"sensors\": [{\"sensor\": \"\"+sensorTempId+\"\", \"observations\": [{\"value\": \"\"+String(t)+\"\"}],\"}, //sensor temperatura
316     sensors=sensors+\"\"+\"sensorHumid+\"\", \"observations\": [{\"value\": \"\"+String(h)+\"\"}],\"}, //sensors humitat
317     sensors=sensors+\"\"+\"sensorLuxId+\"\", \"observations\": [{\"value\": \"\"+String(illum3)+\"\"}],\"}, //sensors LDR
318     sensors=sensors+\"\"+\"sensorCO2Id+\"\", \"observations\": [{\"value\": \"\"+String(CO2ppm)+\"\"}],\"}, //sensors M0135
319     sensors=sensors+\"\"+\"sensorNoiseId+\"\", \"observations\": [{\"value\": \"\"+String(dBA)+\"\"}],\"}"; //sensors so
```

Posteriorment s'obre la comunicació mitjançant AT+CIPSTAT, s'envia la petició indicant prèviament la longitud mitjançant AT+CIPSEND i s'espera la resposta a la petició:

```
Serial1.println("AT+CIPSTART=TCP,\"api.thingtia.cloud\",80);
if(Serial1.find("CONNECT")){
    Serial.println();
    Serial.println();
    Serial.println("Conectat a thingtia");
    Serial.println();
}
else{
    Serial.println("No s'ha pogut connectar a thingtia");
}
//Enviam la mida en caràcters de la petició http:
Serial1.print("AT+CIPSEND=");
Serial1.println(peticiónHTTP.length());
//esperem a ">" per enviar la petició http
if(Serial1.find(">")) // ">" indica que podem enviar la petició http
{
    Serial.println("Enviando HTTP . . .");
    Serial1.println(peticiónHTTP);
    if( Serial1.find("SEND OK"))
    {
        Serial.println("Petició HTTP enviada:");
        Serial.println();
        Serial.println(peticiónHTTP);
        Serial.println("Esperant resposta ...");
    }
}
```

6.2.4 Mecanitzat

Per la realització del mecanitzat s'ha realitzat el disseny en 3D del component model WiFi utilitzant el software Geomagic. [23]

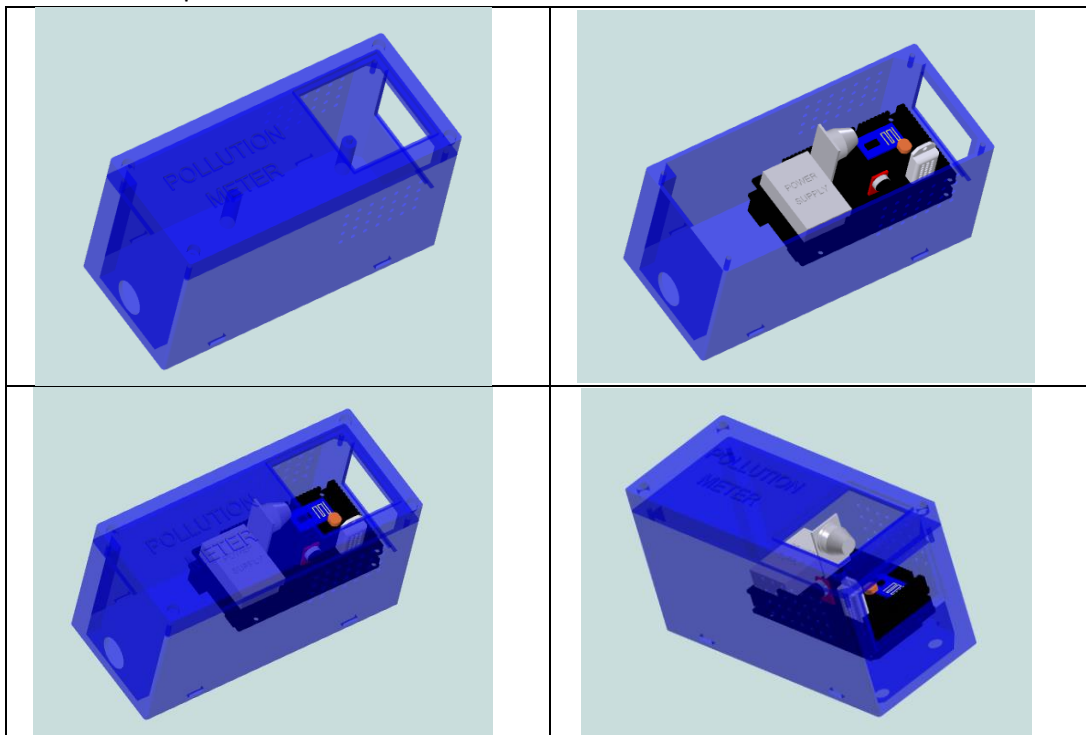
El mecanitzat s'ha realitzat en dues parts:

- Una tapa superior amb una finestra per a que entri la llum solar i poder captar la quantitat d'il·luminació i els forats corresponents per poder cargolar amb la part inferior

- Una part inferior on s'ha realitzat una altra finestra per tenir una major il·luminació i evitar ombres al LDR.
Es disposa de 4 suports verticals on fixar l'Arduino i ubicar sota d'ell la bateria.
S'han afegit una sèrie de forats per crear un corrent d'aire pel detector de qualitat d'aire i facilitar la detecció de so, dues obertures per on passar les brides per agafar-la al suport i un passacables a la part inferior per poder realitzar la carrega de la bateria

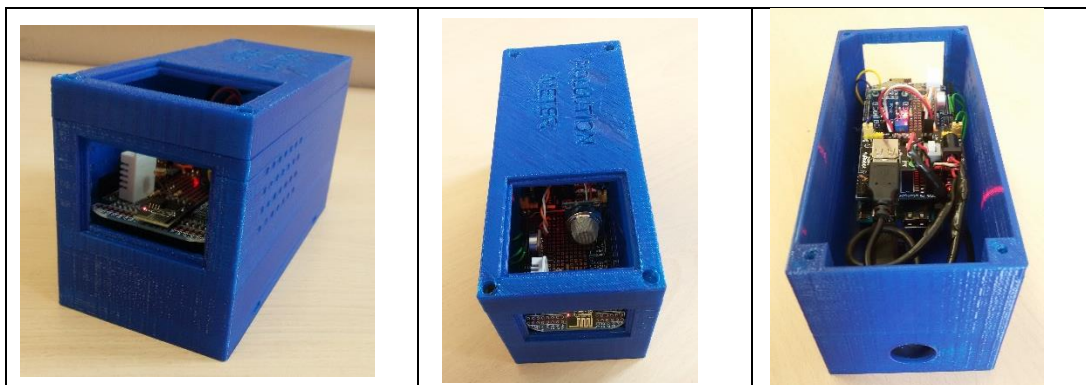
Les dues finestres van tancades amb vidre plàstic de 2mm i el passacables va obert sempre que la posició sigui en vertical, per posició en horitzontal cal afegir un prensastopa per evitar l'entrada d'aigua en cas de pluja.

Vistes de les parts del mecanitzat:



Imatge 30. Disseny 3D del mecanitzat

Pel prototip s'ha realitzat la impressió 3D de les dues peces, quedant de la següent manera:





Imatge 31. Component model WiFi amb el mecanitzat

6.2.5 Alimentació:

Per al model amb WiFi trobem que es necessita alimentació a 3,3V per al ESP8266 i una alimentació de 5V o superior segons l'alimentació de l'Arduino que es vulgui utilitzar, tenint en compte que el sensor MQ135 necessita de 5V d'alimentació.

S'afegeix que el Arduino Mega no es capaç de proporcionar els 200mA que necessita el ESP8266 per arrencar, i per tant es necessari afegir una font d'alimentació per alimentar-lo i que s'aprofita també per alimentar tots els sensors.

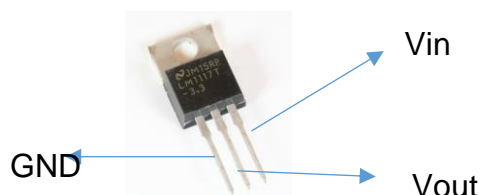
En primera instancia s'ha decidit utilitzar el mòdul d'alimentació YwRobot Power MB V2 que permet dos voltatges de sortida, un de 5V i un altre de 3,3V, capaç de proporcionar fins 700mA.



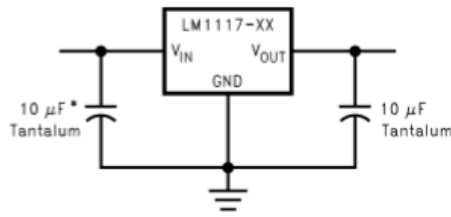
Imatge 32. YwRobot Power MB V2

Mitjançant un power bank usb s'alimenta el mòdul d'alimentació i l'Arduino, tots dos pel port usb que proporciona al Arduino un corrent regulat i estable a 5V.

Al disposar d'un corrent estable de 5V s'ha canviat l'alimentació per un regulador lineal de voltatge que proporciona a la sortida 3,3V i un màxim de 800mA, el LM1117. D'aquesta manera disminuir el cost i el nombre de components.



A continuació es mostra l'esquema:



Imatge 33- Esquema circuit alimentació 3,3V

La bateria USB permet la seva carrega de múltiples formes, ja sigui amb un carregador per dispositius electrònics o una placa fotovoltaica més un regulador a 5 volts. Per a tal efecte s'ha deixat un "forat" a la part inferior per a poder passar els cables d'alimentació des de una placa solar o des de un carregador quan sigui necessari.

Respecte a l'autonomia el dispositiu està pensat per ser carregat mitjançant panells solars i aconseguir una autonomia mínima de 2 dies (48 hores)

Per conèixer els consums del sistema s'han tingut en compte els datasheet del elements per escollir les fonts d'alimentació i posteriorment s'han mesurat els consums reals de les diferents parts:

Element	Consum (mA)
Arduino Mega en execució	73,4 mA
Sensors a 5V: MQ-135 i DTH22	120mA
Sensors + ESP8266 connectats a 3,3V	60,6mA
	254mA

Tabla 4. Consum Component model WiFi

Tots els dispositius alimentats a 3,3V consumeixen 60mA en repòs i 80mA quan s'està transmetent informació mitjançant el ESP8266. L'enviament de dades és cada 30 segons i es triga menys d'un segon, que seria un 3,3% del temps. Per tant el consum mitjà aproximat serà de $60 \cdot 0,97 + 80 \cdot 0,03 = 58,2 + 2,4 = 60,6\text{mA}$

En el cas de desitjar una autonomia de 48 hores, la capacitat mínima de la bateria es $254 \text{ mA} \cdot 48\text{h} = 12192 \text{ mAh}$.

Al utilitzar un power bank la seva carrega es fa mitjançant un port miniUSB a 5V de corrent continua, que es pot realitzar des de un transformador amb sortida de 5V o des de una placa solar amb un regulador amb una sortida de 5V.

Una altra opció es alimentar el conjunt mitjançant bateries de Polimer de Liti, que segons el nombre de celes proporcionen entre 3,7V els de una cel·la fins als 12,6V els de tres cel·les.

En aquest cas es va descartar aquesta opció per ser la més cara i no disposar de pressupost, però es tracta de l'opció més encertada a llarg termini perquè la seva densitat de energia és superior a la resta d'alternatives i, en cas d'alimentar-la amb un panell solar, amb ella es podria augmentar l'autonomia del dispositiu.

Càlcul temps de carrega a partir d'un panell solar

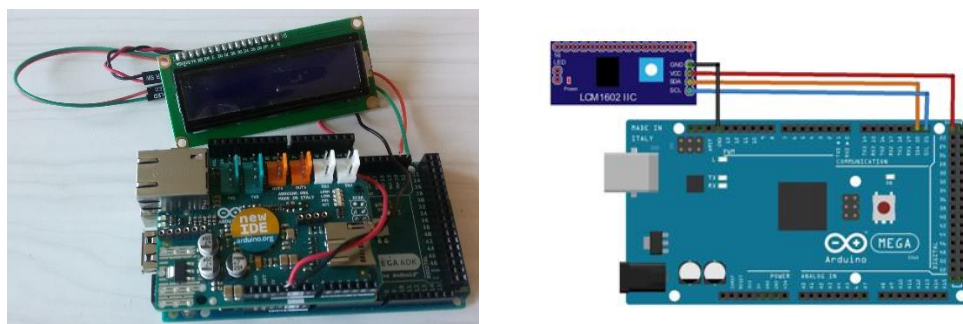
Amb un panell de 20W de potència a Catalunya [Atlas de radiació solar] al mes amb menys radiació solar de l'any, Desembre, obtenim un hsp (hores de pic solar) de 2,48 amb un angle d'inclinació òptim de 55o.

Això indica que utilitzant bateries que permetin una carrega de fins a 1,5A, per un panell de 25W amb una corrent màxima de 1,43A es necessitarien 8,5 hores per carregar una bateria de 12Ah, que al Desembre corresponen a $8,5/2,48 = 3,4$ dies. Al Julio, el més amb un hsp més alt a Barcelona, 5,76 hores, es necessitarien 1,4 dies per realitzar la carrega.

Per panells de 60W es necessitarien bateries que permetin una carrega com a mínim de 3,5A i a Barcelona al pitjor cas, al Desembre, es trigaria 3,48 hores en carregar-se completament. Per tant es necessitaria de mitja 1,4 dies per carregar-se al Desembre i 0,6 dies per carregar-se completament al Juliol.

6.3 Actuadors

En aquest cas cada actuator està compost d'un Arduino Mega, una Shield Ethernet 2 i una pantalla LCD de 16x2. La seva funció, una vegada subscrit a les alarmes, és la de rebre les alarmes, processar les peticions POST i segons els tipus d'alarma mostrar per pantalla el missatge corresponent.



Imatge 34. Actuator

6.3.1 Tractament Alertes al Actuator

La decisió del tipus de missatge es pren a l'Arduino una vegada processades les alarmes, no a la plataforma Web.

Per implementar els actuadors s'ha configurat un servidor web a l'Arduino que està contínuament esperant rebre peticions. Per fer-ho s'utilitza la llibreria Ethernet i les següents funcions:

Mètodes/funcions	Comentari
EthernetServer server(port);	Crea l'objecte Servidor i accepta peticions pel port especificat.
Ethernet.begin(mac, ip);	Configura la connexió ethernet.
server.begin();	Inicialitza el servidor per a que estigui disponible per rebre peticions.

Client.read()	Llegeix caràcter a caràcter el rebut des del client.
Client.println(text)	Envia el text al client
Client.connected()	Retorna un 1 si hi ha un client connectat
Client.available()	Retorna un 1 si el client està disponible
Client.stop()	Tanca la connexió amb el client

Tabla 5. Funcions llibreria Ethernet

La plataforma Sentilo quan detecta una alarma, si hi ha algun subscriptor, envia una petició HTTP POST a la IP del subscriptor amb les dades de l'alarma. A la següent imatge podem veure un exemple de la petició POST que es realitza on es pot observar tant la capçalera com el contingut del missatge en format JSON:

```
POST / HTTP/1.1
Content-Length: 363
Content-Type: application/json; charset=UTF-8
Host: 188.79.75.47:80
Connection: Keep-Alive
User-Agent: Apache-HttpClient/4.2.2 (java 1.5)
{"message": "Alarm a1: value 55 from the sensor temp verifies the restriction: Greater than 40", "timestamp": "23/05/2017T16:38"}
```

Una vegada que el servidor web rep una petició HTTP, el codi del Arduino verifica que es una petició POST i si ho és, cerca l'alarma i el valor que ha originat la petició. Al cas de la imatge és l'alarma a1 amb un valor de 55.

Si a la plataforma Sentilo s'intenten crear diferents rangs per un mateix sensor sempre alerta del més restrictiu, per tant son als propis actuadors on s'especifiquen els diferents rangs d'alarma per a un mateix valor mitjançant comparacions i d'aquesta manera decidir quin missatge es mostra per pantalla.

Posteriorment es respon amb un 200 OK a la petició per a que Sentilo no faci més reintents d'enviament.

Els nivells creats han estat:

Per l'alarma de so:

- Entre 70 i 80 dBA
- Entre 80 i 90 dBA
- Més de 90 dBA

Per l'alarma de qualitat de l'aire:

- Entre 500 i 550 ppm
- Entre 550 i 600 ppm
- Més de 600 ppm

Pel processament de les peticions s'han utilitzat les funcions de substring() que permet extreure una part concreta d'un text i S.indexOf(text) que retorna la posició de la primera coincidència amb "text" a un determinat string. D'aquesta manera es busquen les alarmes a les que corresponen les peticions HTTP POST rebudes.

En aquest cas els actuadors no van mecanitzats, ja que l'Arduino i les pantalles no tenen perquè estar a la mateixa ubicació. La idea és que els Arduino estiguin dins del registre d'alimentació de les pantalles o dins dels registres de semàfors.

6.3.2 Publicació missatges per LCD

Per la comunicació amb la pantalla s'utilitza el bus I2C (Inter-Integrated Circuit), per fer-ho s'ha utilitzat la llibreria LiquidCrystal_I2C.

L'adreça de la pantalla utilitzada per a poder comunicar-se amb l'Arduino per I2C és la 0x27, definint diferents adreces per diferents dispositius es poden comunicar amb l'Arduino pel mateix bus I2C.

Les funcions utilitzades han estat:

Mètodes/funcions	Comentari
lcd.print("text")	Mostra els missatges per pantalla
lcd.setCursor(x,y)	Col·loca el cursor a la posició desitjada
lcd.clear();	Esborra la pantalla
lcd.setBacklight(HIGH/LOW);	Encén i apaga la il·luminació de la pantalla

Tabla 6. Funcions llibreria LCD

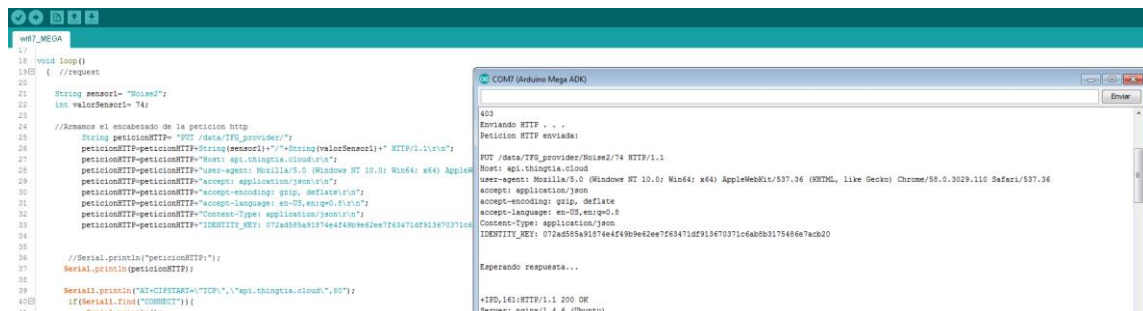
7. Proves funcionament.

Les proves de funcionament realitzades son:

- Comunicació WiFi entre Arduino i Sentilo
- Publicació dades a Sentilo enviant-les des de els components
- Avaluació de les alertes i alarmes creades
- Recepció per part dels subscriptors de les notificacions de les alarmes.
- Processament de les alarmes i mostra de missatges a través del LCD.

Comunicació WiFi

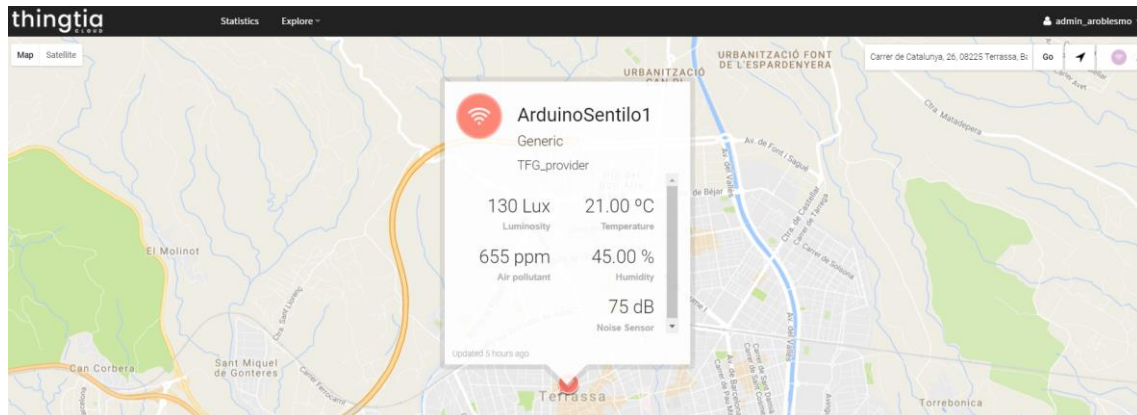
S'ha comprova el correcte funcionament de la comunicació WiFi amb Sentilo, mitjançant la creació dels datagrames i l'obtenció de les respostes per part de Sentilo.



Publicació de les dades a la Plataforma Sentilo.

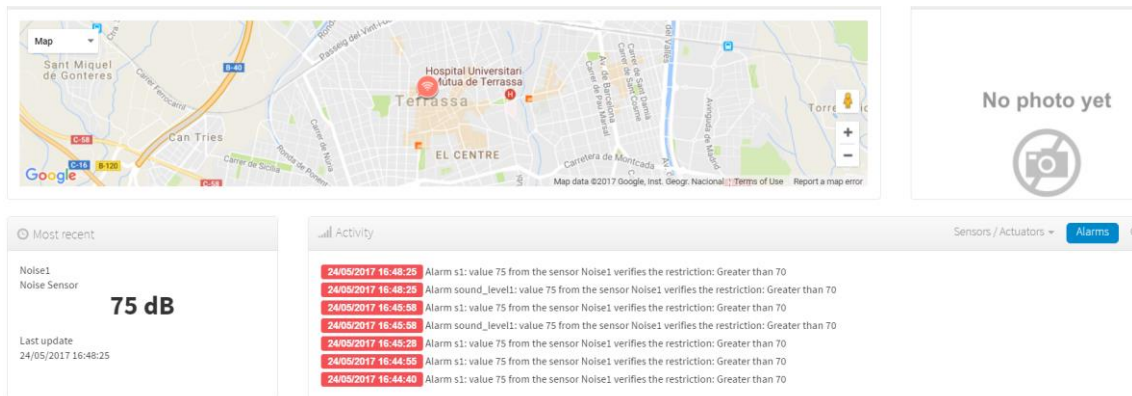
S'han realitzat proves de mesura i publicació de les dades a Sentilo tant amb el model Ethernet com amb el model WiFi, totes dues amb resultats satisfactoris.

A la següent imatge es veu una publicació de tots els sensors a Sentilo.



Avaluació de les alertes i alarmes

També s'han realitzat proves amb el funcionament de les alarmes i les alertes a Sentilo. S'ha forçat una entrada de so de 75dB i l'alarma està configurada per avaluar si el nivell és superior a 70dB. Com el valor es superior a la imatge es poden veure les alarmes relacionades amb el sensor Noise1 detectades per Sentilo.



Subscripció a les Alarmes

S'ha comprovat el correcte funcionament dels actuadors i la pantalla LCD. A continuació una alarma rebuda per tenir 99,1dB al sensor de so 1.

A la imatge es pot observar que es rep la petició HTTP POST referent a la notificació de la alarma. Com es compleix la restricció de ser superior a 70 i que el propi actuator processa la informació i mostra el missatge que és superior a 90dBA.

```

ArduinoPAC1_4_dB
109 // Reset the display
110 //I2C.clear();
111 //I2C.setBacklight(DON);//Backlight OFF
112 //delay(500);
113
114
115
116
117 // listen for incoming clients
118 EthernetClient client = server.available();
119 if (client) {
120   Serial.println("New client");
121   // an http request ends with a blank line
122   boolean currentLineIsBlank = true;
123   int empty_line_count=0;
124   String req_str=""; //String per guardar la petició HTTP
125
server is at 192.168.1.177
New client
Enviando respuesta
POST / HTTP/1.1
Content-Length: 399
Content-Type: application/json; charset=UTF-8
Host: 192.168.1.177
Connection: Keep-Alive
User-Agent: Apache-HttpClient/4.2.2 (java 1.5)
{"message":"Alarm sound_level: value 99.1 from the sensor Noise1 verifies the restriction: Greater than 70","timestamp":"11/06/2017 02:34:35","topic":"/alarm/sound_level","type":"ALARM","sensor":
La alarma es: sound_level
El valor es: 99.1
99
Alarma sonido superior a 90 dB
client disconnected

```

Processament de les alarmes i mostra de missatges a través del LCD

Com es compleix la restricció de ser superior a 70 a la imatge superior es veu que el propi actuador processa la informació i mostra el missatge que és superior a 90dBA.

Com a resultat s'obté el missatge corresponent a la pantalla LCD:



A la plataforma s'observa l'alarma per ser el valor superior a 70dB

Proves a exteriors

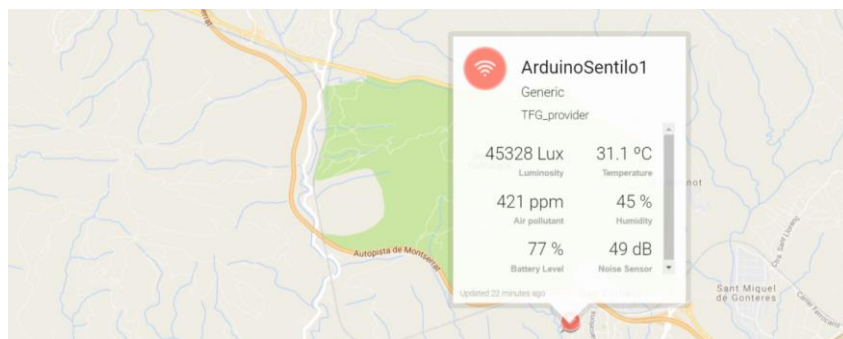
S'han realitzat proves en exteriors amb el model WiFi i s'ha avaluat la cobertura, la precisió de les dades i la correcta comunicació amb Sentilo.

Les proves s'han realitzat a la població de Viladecavalls a la ubicació amb coordenades 41° 33' 28" N, 1° 57' 21". La distància entre el punt d'accés WiFi i el dispositiu és de 22 metres, i només hi ha un envà entre ells.



Imatge 35. Component WiFi ubicat a un fanal

La comunicació entre el punt d'accés i el sistema ha estat satisfactori i les mesures obtingudes i publicades a Sentilo han estat:



Les lectures de so i il·luminació amb el sonòmetre i el luxòmetre en aquesta situació han estat de 52,9 dB i 75800lux.

La desviació en contaminació acústica de quasi 4 dB és acceptable per l'aplicació. La desviació en la il·luminació és molt gran degut a que la calibració s'ha realitzant amb més mostres de valors entre 0 i 1000 lux. Això ha estat així perquè l'objectiu del LDR és saber quan durant la nit hi ha zones amb una il·luminació excessiva, i per tant els valors que interessen estan entre el 0 lux i els 500 lux.

8. Viabilitat

El principal mercat al que va destinat el projecte és als ajuntaments que volen tenir una ciutat intel·ligent i que es preocupen per la qualitat de vida dels seus ciutadans així com de disminuir tot tipus de contaminació al municipi.

Actualment la gran majoria dels municipis de la província de Barcelona estan posant en marxa projectes relacionats amb Smart City i aquest és projecte interessant pels que sofreixen problemes de contaminació degut a la densitat demogràfica o a la densitat de vehicles de motor.

8.1 Viabilitat tècnica

Tant la creació de sistemes encastats per l'adquisició de dades mitjançant Arduino i sensors com el funcionament de la plataforma Sentilo estan contrastats i s'estan utilitzant en diversos

projectes. Els mateixos desenvolupadors de Sentilo han creat una llibreria específica per utilitzar Arduino com a client de Sentilo.

Respecte a la comunicació, la gran majoria de municipis disposen de xarxes WiFi a espais públics tant edificis com a alguns parcs de les ciutats que poden ser utilitzades per la implementació de la xarxa de sensors sense haver de fer una nova inversió en xarxes. Existeixen d'altres municipis que haurien d'augmentar el nombre de punts d'accés o canviar les antenes per augmentar la cobertura de les xarxes existents.

8.2 Viabilitat econòmica:

El pressupost dels sistemes encastats de mesura, variarà en funció dels tipus de sensors a utilitzar i de si ha de ser autònom o no. Per un sistema autònom l'objectiu és que el pressupost estigui entre els 125 y 150€ + IVA . A mesura que la qualitat dels sensors augmenti el pressupost creixerà, sobretot degut als sensors de qualitat de l'aire. Els valors que es poden observar a la taula inferior responen al sistema autònom muntat durant el projecte més una alimentació amb una placa solar de 25W.

Descripció	Referència	Preu
Arduino Mega	Arduino MEGA 2560 rev3	34,50
Mòdul WiFi	ESP8266-01	3,90
Sensor so	Sparkfun BOB-09964	7,60
Fotodíode	LDR	0,40
Sensor qualitat aire	MQ135	4,30
Sensor temperatura i humitat	DHT22	5,80
Alimentació	LM1117T 3,3	1,79
Bateria	AUKEY Powerbank Portàtil 12000mAh	19,90
Regulador	ARDUINO SOLAR CHARGER SHIELD V2.2	14,87
Placa solar	ECO-WORTHY 12 Volt 25 Watt Solar	60
	TOTAL	153,06+ IVA

Tabla 7. Exemple de pressupost de sistema autònom



El nombre de sistemes encastats per adquirir les dades serà diferent segons l'àrea a cobrir i les característiques de cada ciutat.

8.2.1 Plataforma Sentilo

Amb la plataforma Sentilo hi ha la possibilitat de treballar de tres formes diferents, dues d'elles amb servidors propis una tercera mitjançant un SaaS que ofereix l'empresa Thingtia.

Uns dels principals avantatges de Sentilo és el gran impuls que la Diputació de Barcelona li està donant, ha finançat el desenvolupament de la modalitat multi-tenant i junt l'empresa Thingtia la posa a disposició dels 311 municipis de la província de Barcelona de en forma de SaaS de manera gratuïta. A més a més, dona ajudes de fins a 10.000€ als ajuntaments de municipis de més de 20000 habitants per al desenvolupament del programari que permeti publicar o subscriure dades a Sentilo prèvia presentació de projecte.

Com el projecte està destinat a ajuntaments, si aquest son de la província de Barcelona, es considera que la instància i l'allotjament de la plataforma Sentilo als ajuntaments no els hi suposa cap cost econòmic.

En cas de no ser de la província de Barcelona el preu de Thingtia per al sistema seria de 99€ mensuals amb un suport web bàsic i de 228€ mensuals per un suport web i 20 hores de suport telefònic.

8.2.2 Payback.

Es considera que la inversió inicial son 300h de treball del TFG més el cost del material que es valoren en 4000€ nets. El flux de caixa depèn de les unitats venudes i el benefici brut de cada una de les unitats venudes és de 30€.

El benefici de cada unitat s'ha calculat afegint un marge del 20% al seu preu.

Si al preu del sistema encastat ,que és d'uns 150€, li afegim un 20% de marge, per recuperar la inversió de 300 hores de treball, cada un el comercialitzaríem per 180€. Si només ens dediquéssim vendre equips necessitaríem vendre uns 400 sistemes encastats per recuperar la inversió inicial i començar a obtenir beneficis.

EL benefici net de cada equip venut serà aproximadament de 30€ menys impost de societats (35%) i menys IRPF, que finalment serà d'uns 15€ nets.

Encara que el cost dels materials disminuirà considerablement al augmentar la producció, inicialment no s'ha tingut en compte aquest paràmetre degut a que el cost de mà d'obra per implementar el sistema creixerà degut a que la mà d'obra de muntatge del primer prototip està inclosa en les 300 hores de projecte i no al cost del prototip. El cost d'energia, lloguer d'instal·lacions i personal auxiliar tampoc està inclòs al prototip. Per tant, s'ha preferit utilitzar un escenari pessimista a l'hora de realitzar els càlculs de retorn d'inversió i el resultat és que es necessitarien vendre 400 equips per recuperar la inversió i començar a tenir beneficis si només ens dediquéssim a vendre equips.

En aquest cas activitat no és només vendre equips sinó que és la realització de projectes per Smart Cities basades en Arduino i Sentilo.

Per fer els càlculs del payback s'han considerat projectes base amb 30 equips.

Les hores totals dedicades al projecte seran 130 hores, que corresponen a 1 hora per muntatge, configuració i proves de cada equip més 100 hores de projecte sempre que s'utilitzi un SaaS ja implementat.

Respecte als ingressos i les despeses es tindran:

- El preu hora es facturarà a 45€ bruts.
- El preu hora treballador es pagarà al treballador a 12€ nets, que correspondrien 1920€ mensual nets per a una jornada completa.
- El cost hora del treballador a l'empresa una vegada sumats IRPF i Seguretat Social serà de 21€, que a l'empresa li costaran 3360€ en cas de ser una jornada completa.

El benefici brut del projecte seran les hores de projecte per la diferencia entre la facturació per hora i el cost de cada hora més el benefici brut de cada hardware dels equips.

Seran $130h \times 24€ + 30\text{equips} \times 30€ = 3120+900 = 4020 €$

A aquest benefici se li ha de restar l'Impost de Societats (35%) i el IRPF que hem considerat del 20%. Tenim llavors que:

$4020 - 35\% = 2613€$

$2613 - 20\% = 2090,40€$

Com a conclusió es pot veure que el flux de caixa per a un projecte amb 30 equips és de 2090,4€ i serien necessaris 3 projectes com l'estudiat per recuperar la inversió inicial i començar a tenir beneficis.

8 Conclusions

En aquest apartat es comenten les conclusions finals obtingudes i si s'han assolit els objectius fixats al principi del projecte. També s'inclou un anàlisi amb possibles millores i una autoavaluació personal que descriu els punts crítics o que més problemes han donat durant el desenvolupament

9.1 Conclusió del projecte

Els objectius inicials del projecte es dividien en dos parts:

- Facilitar l'accés a les dades de contaminació atmosfèrica, sonora i lumínica als ciutadans a partir d'una plataforma web.
- Utilitzar les dades de contaminació i la mateixa plataforma web per crear alarmes i missatges que es visualitzin a panells informatius.

Les dues parts tenen una part de hardware i una part de software, la part de hardware es basa a la creació del sistema encastat amb tots els sensors i les comunicacions i la part de software en el tractament de les dades mesurades, la seva calibració i la programació de les comunicacions.

La informació trobada sobre la plataforma Sentilo ha estat una base fonamental per començar a treballar amb la comunicació amb l'equip i la plataforma mitjançant Ethernet. La documentació sobre la API REST ha suposat una base fonamental per entendre el funcionament i posteriorment realitzar les peticions HTTP amb les rutes i formats adequats al model amb WiFi.

Tant amb el model Ethernet utilitzant la llibreria proporcionada per Sentilo com al model WiFi a la que no s'utilitza cap llibreria específica per crear les peticions HTTP s'ha aconseguit transmetre i publicar les dades dels sensors a la plataforma. Per tant els objectius de la primera part es consideren complerts, arribant a realitzar tots els objectius que no es consideraven prioritaris al iniciar el projecte, com per exemple la mecanització del prototip del sistema encastat.

La segona part es basava en dur a terme el desenvolupament d'un sketch que una vegada subscrit a les notificacions de les alertes, fos capaç de rebre les notificacions, processar-les i mostrar diferents missatges a una pantalla LCD segons uns paràmetres especificats.

La major dificultat es troba al processament de les peticions, identificant-les i reconeixent el valor que ha fet saltar una determinada alarma.

En aquesta segona part és imprescindible una correcta configuració a la plataforma de les alertes, les alarmes i les subscripcions mitjançant peticions HTTP a la seva API REST. Per facilitar la execució d'aquestes tasques s'ha utilitzat el software Advanced REST Client de Google.

A aquesta part s'ha realitzat amb el nivell de so i el nivell de qualitat d'aire, especificant 3 nivells per a cadascun d'ells i s'ha comprovat el correcte funcionament de la mateixa, per tant també es consideren complerts els objectius.

8.2 Possibles millores

Durant el projecte s'ha prioritzat el cost econòmic i la part de comunicació entre els sistemes per sobre calibració i la qualitat dels sensors.

Utilitzant sensors de més qualitat la precisió i exactitud augmentaria i la calibració dels sensors seria més efectiva.

Per la creació de les alarmes es considera que la precisió es correcta, però no per la publicació de les dades, ja que les dades publicades han de ser menys sesgades.

El sensor MQ135 es sensible a múltiples gasos i per tant la lectura dels ppm de CO2 serien reals si es considera que només hi ha CO2 a l'aire i no hi ha cap de la resta d'agents als que el MQ135 és sensible.

El canvi per un MICS-2714 o un MICS-4514 i detectar NO2 o CO seria una millora considerable a canvi d'un augment del cost de 20€. El canvi també beneficiarà a la autonomia, doncs el consum del sensors MQ-135 (120mA) suposa gairebé la meitat del consum del sistema complet i el MICS-2714 consumeix segons el seu datasheet 26mA i el MICS-4514 entre 26 i 32 mA. El consum total del sistema passaria de 273mA a 179mA i la capacitat necessària per a 48 hores d'autonomia es disminuiria un 30% fins els 8592mAh.

8.3 Autoavaluació personal

A continuació es descriuen els problemes que han anat apareixent durant el desenvolupament d'aquest projecte:

- Problema al intentar publicar dades a la instància de Sentilo de Thingtia.cloud: Després de treballar amb la màquina virtual que proporciona Sentilo es va decidir treballar a una instància gratuïta de Thingtia.cloud. Al enviar les dades per publicar-les a Sentilo thingtia.cloud refusava la petició. El problema era que l'adreça a la que feia la petició no era la correcta i al fer el registre a Thingtia no havia rebut a quina adreça es trobava la API REST. Mitjançant mail des de l'equip de Thingtia em van donar la solució en un parell de dies, l'adreça correcta era la <http://api.thingtia.cloud> i el port el 80.
- Retard en l'enviament del sensor de so: L'enviament va tenir un retràs de dues setmanes, es va demanar un altre sensor de so amb transport urgent per no haver de modificar la planificació.
- Dificultat per trobar equips de mesura per poder realitzar les calibracions dels sensors: Per realitzar les calibracions es van estudiar dues opcions, fer-ho amb aplicacions per smartphones o buscar equips de mesura específics. Es va decidir buscar els equips de mesura específics i es van tenir dificultats per trobar-los. Finalment es van realitzar les calibracions amb un luxòmetre Chauvin Arnoux 813 i amb un sonòmetre Proskit MT-4008 cedits per l'escola Salesians de Terrassa. Per nivell de CO2 no es va aconseguir cap equip de mesura.
- Problema amb Arduino UNO no originals: En fer l'estudi de model d'Arduino a utilitzar en cada cas, es van fer proves amb Arduino UNO, als quals al arribar al 60% de la memòria els sketch no s'executaven correctament, en principi es pensava que era un defecte de programació, però al canviar al Arduino Mega els problemes es solucionaven.
- Dificultats amb les llibreries: Per al LCD s'han trobat varies llibreries disponibles, algunes de les quals no funcionaven correctament amb el LCD adquirit, només apareixia la primera lletra de cada línia i per imprimir una paraula sencera s'havia de fer imprimint lletra a lletra i

movent el cursor. Després de provar varies llibreries es va trobar una que funcionava correctament. Segons la informació trobada el problema era la compatibilitat de la versió del IDE de Arduino, la llibreria i la placa del LCD.

- Dificultat en utilitzar el WiFi ESP8266-01: Tot i que existeixen llibreries per facilitar la utilització del mòdul, després de l'experiència amb el LCD es va optar per utilitzar el ESP8266 directament amb comandes AT sense la utilització de cap llibreria. Això obliga a crear amb les comandes AT les connexions, els paquets a enviar i les peticions i respostes HTTP.
- Degut a la falta de subministrament a data del lliurament de la memòria no s'han tapat les finestres del equip amb el vidre plàstic de 2mm per evitar l'entrada d'aigua.

Personalment ha estat un projecte motivador que ha resultat tot un repte per la escassa formació en Arduino i la seva programació.

Gràcies als coneixements de xarxes la part de programació de peticions HTTP ha resultat més senzilla, però per una altra banda, degut a la desconexió de la programació d'Arduino s'ha hagut de dedicar més temps a l'estudi de les llibreries, a la creació de les connexions i a la adquisició de dades per aconseguir els objectius.

S'han ampliat els coneixements que es tenien sobre la matèria i s'ha aconseguit donar més facilitats per aconseguir fer públiques les dades de pol·lució de les ciutats.

El fet de ser un projecte amb un producte final real ha ajudat a definir els objectius i a motivar sobretot a la part final del projecte.

Bibliografia

- [1] “*Air quality in Europe - 2015 report*”, pag 44,
<http://www.eea.europa.eu/publications/air-quality-in-europe-2015#tab-data-references>, últim accés Març 2017.
- [2] “*Efectos de la contaminación acústica sobre la salud*”,
<http://contaminacionacustica.net/efectos-de-la-contaminacion-acustica-sobre-la-salud/> , últim accés 15 Gener 2017.
- [3] “*Efectos de la contaminación lumínica*”,
<http://www.vidasostenible.org/informes/efectos-de-la-contaminacion-luminica/> ,últim accés Març 2017
- [4] J. Cirera, A. Marin, O. Hellin, J. Budi, A. Rippon, F. Giralt, “*Open Sensor Platform*” , November 2016.
- [5] “*Calidad del aire: Datos en tiempo real*”,
<http://datos.madrid.es/portal/site/egob/menuitem.c05c1f754a33a9fbe4b2e4b284f1a5a0/?vgnextoid=41e01e007c9db410VgnVCM2000000c205a0aRCRD&vgnnextchannel=374512b9ace9f310VgnVCM100000171f5a0aRCRD> , últim accés Març 2017
- [6] <https://waqi.info/>, última accés Gener 2017.
- [7] <https://smartcitizen.me/kits/>, últim accés Març 2017.
- [8] <https://docs.smartcitizen.me/#/start/hardware>, últim accés Març 2017.
- [9] <http://www.sentilo.io/wordpress/>, últim accés Març 2017.
- [10] <http://sofia2.com/>, últim accés Març 2017.
- [11] “*Ganadores Open Awards 2016 – OpenExpo 2016*”,
<http://www.openexpo.es/ganadores-open-awards-2016-openexpo/> , últim accés Març 2017.

- [12] “*Coruña Smart City*”, <http://sofia2.com/docs/Sofia2-Soluciones.pdf> últim accés Març 2017
- [13] “*Modelos de implantación Sofia2*”, <http://sofia2.com/sofia2incloud.html>, últim accés Març 2017.
- [14] <http://www.coruna.es/smartcoruna>, últim accés Març 2017
- [15] <http://www.libelium.com/resources/case-studies/>, últim accés Març 2017
- [16] <http://www.libelium.com/products/waspmote/overview/>, últim accés Març 2017
- [17] <http://www.libelium.com/products/waspmote/sensors/>, últim accés Març 2017
- [18] “*RFC 4627. The application/json Media Type for JavaScript Object Notation*”, Juliol 2006, <https://tools.ietf.org/html/rfc4627>, últim accés Març 2017.
- [19] <http://www.sentilo.io/xwiki/bin/view/APIDocs/Overview>, últim accés Març 2017.
- [20] <http://www.sentilo.io/xwiki/bin/view/Sentilo.Community.Tutorials/Arduino+Client>, últim accés Març 2017.
- [21] <https://www.arduino.cc/en/Main/ArduinoBoardLeonardo>, últim accés Març 2017.
- [22] <https://www.arduino.cc/en/Main/ArduinoBoardYun>, últim accés Març 2017.
- [23] <http://www.geomagic.com/en/>, últim accés Juny 2017

ANNEX

A1. Accés a Thingtia.cloud

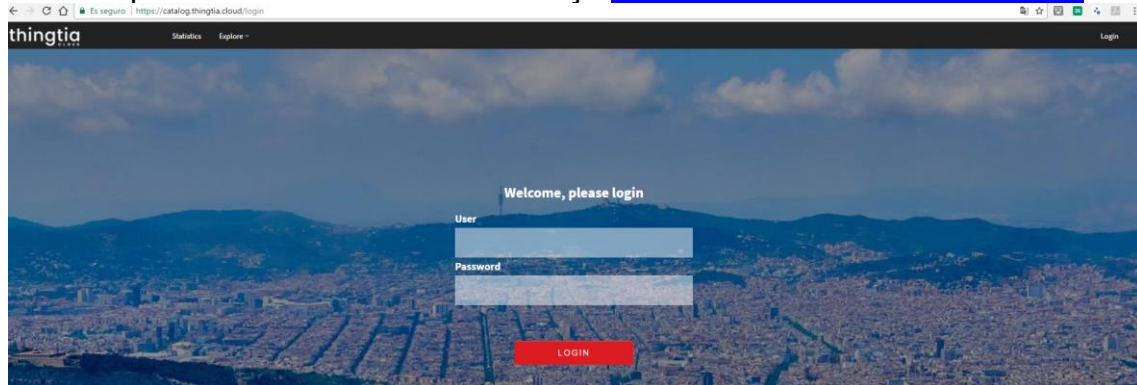
Amb la instància gratuïta de Thingtia per visualitzar les dades és necessari entrar a la plataforma amb un user i un password.

S'ha creat el següent usuari:

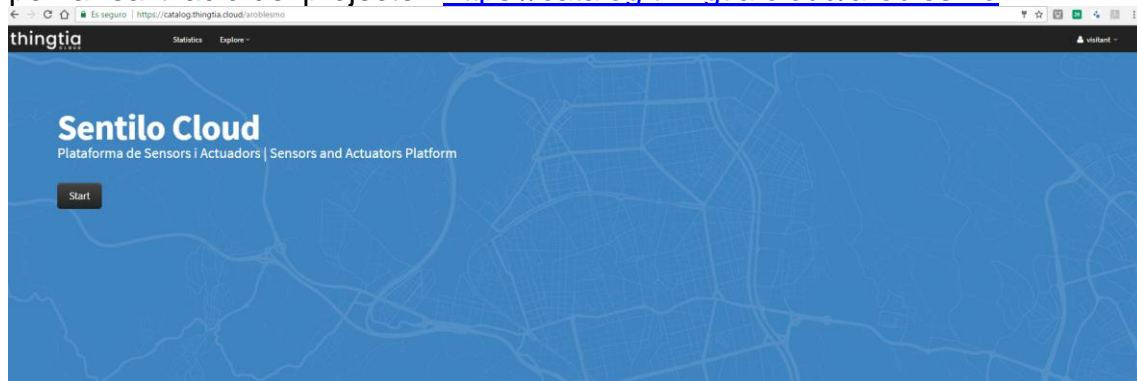
User : visitant

Password: Abc.123

Per tant per accedir s'ha d'anar a l'adreça <https://catalog.thingtia.cloud/login>



Introduir el usuari i el password per accedir a la instància de Sentilo utilitzada per la realització del projecte: <https://catalog.thingtia.cloud/aroblesmo>



Clicant sobre Start es poden visualitzar les últimes dades publicades.

A2. Càlculs adquisició de dades i calibració

A1.1 Sensor d'il·luminació. Fotoresistència GL5528

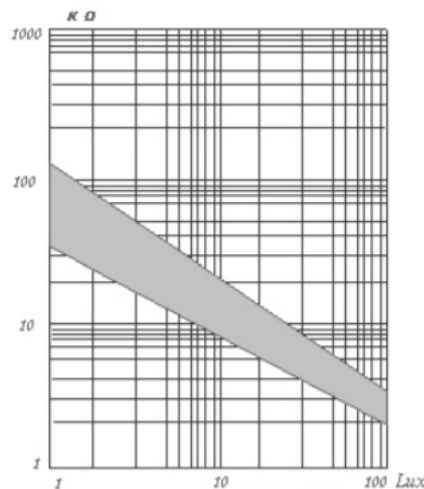
Els LDR varien la seva resistència en funció de la quantitat de LUX en forma exponencial. La dependència entre la resistència i la il·luminació es de la forma:

$$I/I_0 = (R/R_0)^{-\alpha}$$

Sent R_0 la resistència a una intensitat I_0 , totes dues conegudes.

α és la pendent de la gràfica logarítmica, o la pèrdua de resistència per cada dècada.

La dependència es converteix en lineal si s'utilitza una escala logarítmica, tal com es pot veure a la gràfica de dependència entre la resistència i la il·luminació del LDR GL5528 utilitzat:



Per calcular els lux, ho podem fer a partir dels valors de resistència de foscor, la resistència a la llum a 10lux la α que proporciona el fabricant. Al cas del GL5528 els valors utilitzats són:

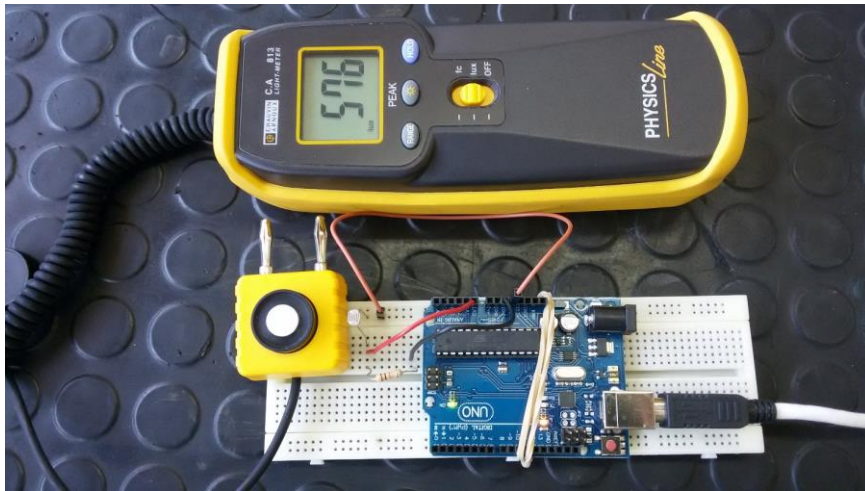
A = 500; //Resistència en oscuridad en KΩ

B = 10; //Resistència a la luz (10 Lux) en KΩ

Rc = 10; //Resistència calibración en KΩ que forma el divisor de tensió amb la LDR

La fórmula utilitzada ha estat: $\text{lux} = (L \cdot A \cdot 10) / (B \cdot R_c \cdot (1024 - L))$; on L és la lectura amb el Arduino i el 1024 correspon als 10 bits del convertidor A/D del Arduino.

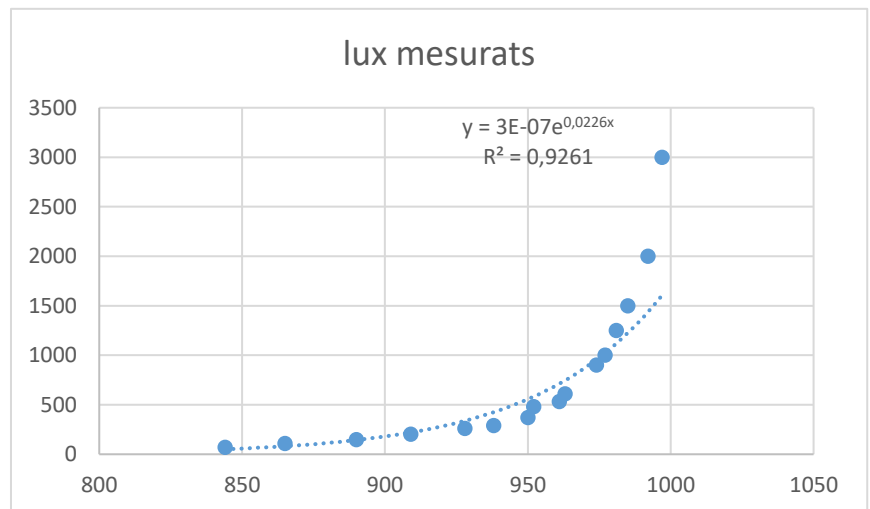
No obstant degut al comportament potencial les petites diferències en termes de resistència suposen grans variacions a la mesura de lux i per tant també s'ha realitzat una calibració. La calibració s'ha realitzat utilitzant el luxòmetre CHAUVIN ARNOUX C.A 813 light-meter.



S'han pres les mesures amb el luxòmetre i amb el Arduino amb el sensor. S'ha realitzat una taula i amb els valors i s'ha fet una regressió per trobar la equació a utilitzar a la conversió.

Els valors de les mesures es mostren a la següent taula:

Mesura Arduino	lux mesurats
997	3000
992	2000
985	1500
981	1250
977	1000
974	900
963	610
961	533
952	482
950	370
938	291
928	260
909	204
890	147
865	110
844	69
75	38
45	0
28	0

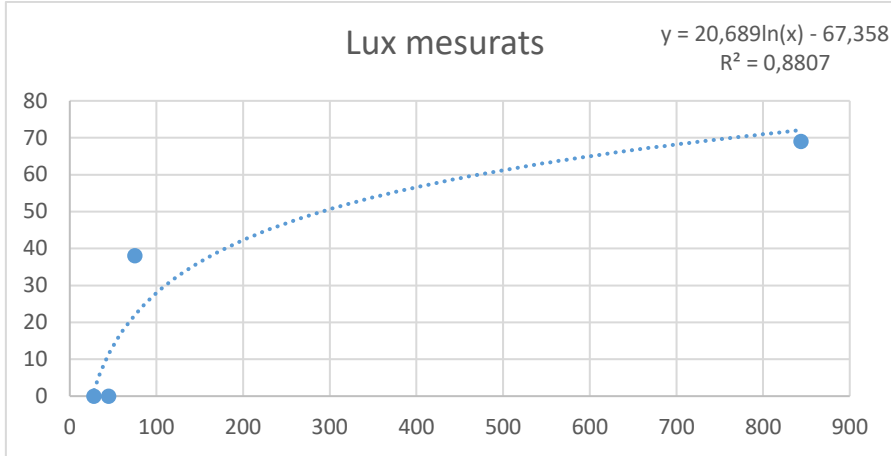


Es pot observar que com el que es vol detectar son zones que durant la nit estiguin il·luminades en excés, s'han pres més mesures per valors inferiors a 1000lux.

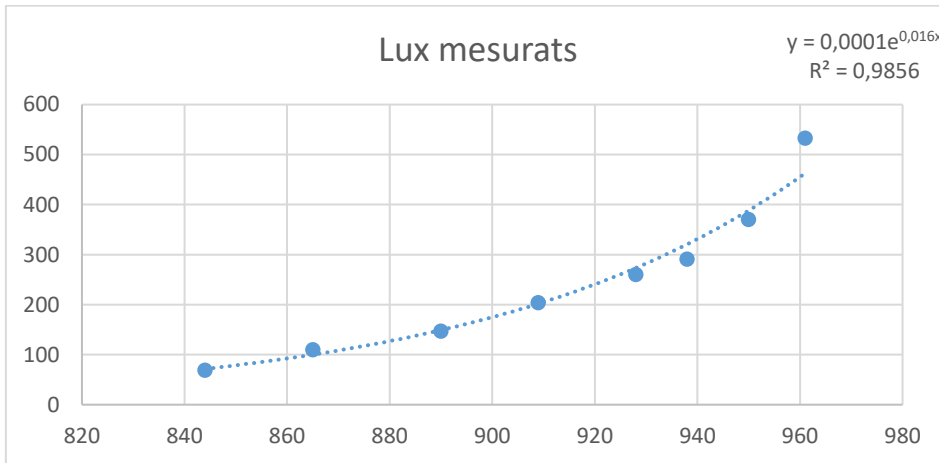
S'observa a la gràfica de la regressió realitzada que amb les dades no es podien linealitzar totes en un sol tram i s'ha decidit realitzar quatre trams diferenciats.

S'han dividit els trams en funció de les mesures obtingudes amb l'Arduino i s'ha realitzat una regressió per a cada un dels trams. Les gràfiques i equacions resultants es poden veure a continuació:

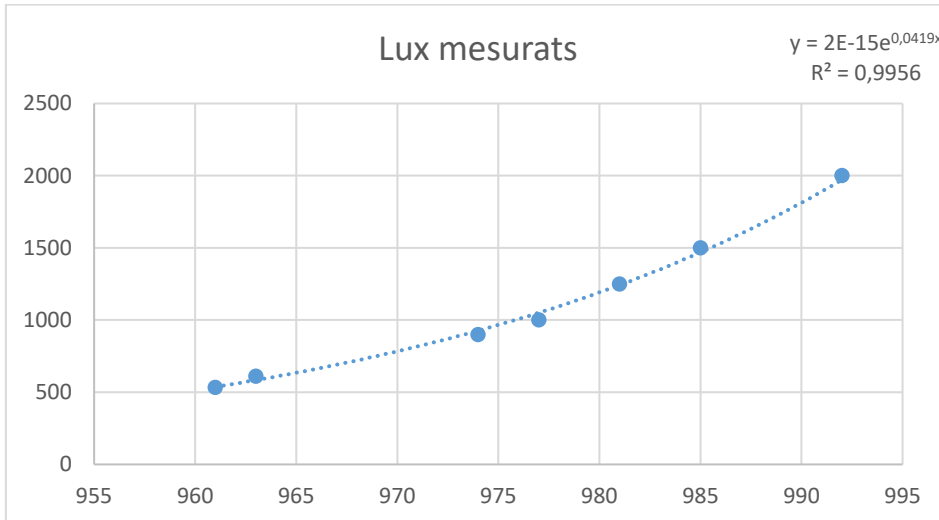
Valors inferiors a 844



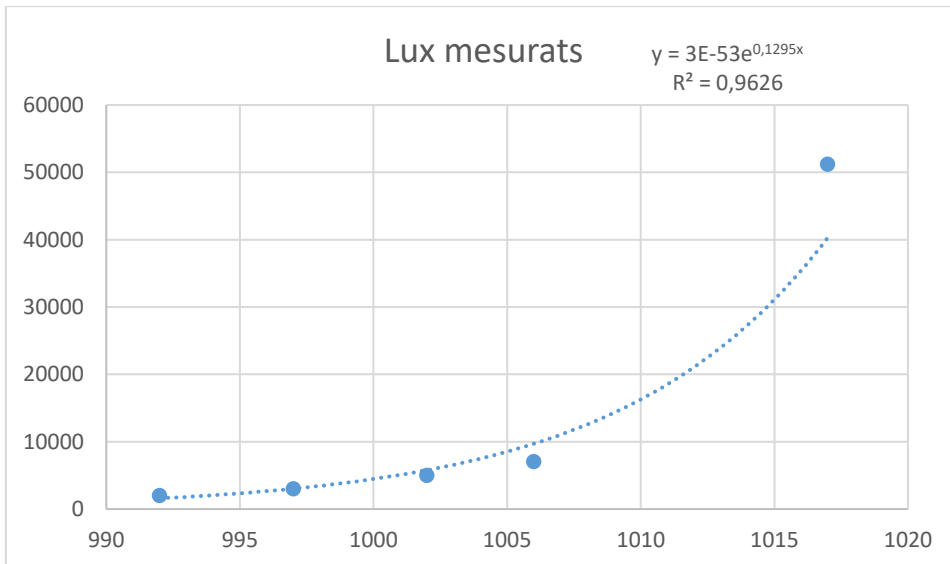
Valors entre 844 i 960



Entre valors de 960 i 992



Entre valors de 992 i 1017



Sensor de so.

El sensor de so proporciona una sortida depenent de la pressió SPL.

El so consta de pressions i depressions, de manera que la forma de mesurar els dB s'ha de realitzar tenint en compte el offset del dispositiu i calculant el valor eficaç.

Per calcular l'offset del dispositiu al setup es deixa durant 4 segons mesurant i posteriorment es fa la mitja de totes les mostres preses.

Per calcular el valor eficaç de les mostres s'eleva al quadrat cada una de les mostres es sumen entre elles i es calcula l'arrel quadrada. S'ha realitzat amb una finestra de 1 segon de durada. La formula és:

$$rms = \sqrt{v_1^2 + v_2^2 + \dots + v_n^2}$$

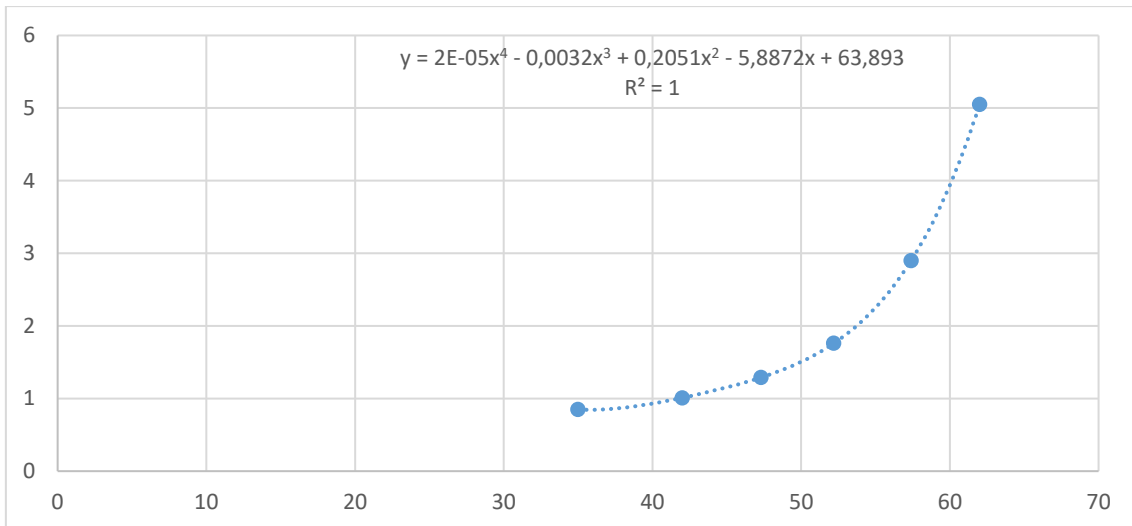
Posteriorment, s'aplica un factor de correcció realitzant una calibració amb el sonòmetre de classe 2 Proskit MT-4008 una vegada el dispositiu es troba dins de la caixa final.



Els valors de les mesures es mostren a la següent taula:

sonòmetre dBA	mesura rms
62	5,05
57,4	2,9
52,2	1,76
47,3	1,29
42	1,01
35	0,85

En aquest cas la equació que més s'aproxima és una polinòmica de grau 3:



Sensor de qualitat de l'aire MQ-135.

És un sensor electroquímic que varia la seva resistència quan s'exposa a determinats gasos, en concret a NH₃, NO_x, alcohol, bencé, CO₂ i CO, sent sensible en similar proporció a tots ells. El seu funcionament es basa en que internament posseeix un escalfador encarregat d'augmentar la temperatura interna i amb això el sensor pugui reaccionar als gasos provocant un canvi de valor en la resistència. Es necessària una resistència de carrega per tancar el circuit i fer un divisor de tensió. El mòdul utilitzat ja disposa de la resistència de carrega i té un valor de 1KΩ.



Per a que la sortida sigui estable es necessari un tems d'escalfament de 24 hores.

Per poder realitzar la calibració s'ha d'utilitzar la corba que proporciona el datasheet i realitzar una regressió per obtenir la formula que té el format:

$$\text{ppm} = a \cdot (R_s/R_0)^{-b}$$

El resultat és $a = 116.6020682$ i $b = 2.769034857$

Una vegada obtinguts els paràmetres a i b a partir de la regressió el introduïrem a la llibreria pel MQ135, que al nostre cas és la llibreria pel MQ-135 de G.Krocker [link a la llibreria]

Per realitzar el procediment de calibració s'ha de mesurar el valor de R_0 . Per fer-ho s'ha de deixar escalfar el sensor durant 24 hores i posteriorment fer la lectura del valor de R_0 a un ambient net, una temperatura de 20°C i una humitat de 35%.

Per fer-ho s'utilitza la funció `gasSensor.getRZero();`

Pels càlculs de la calibració s'utilitzen els ppm a l'atmosfera, que correspon a 409ppm segons la web <https://www.co2.earth/>, per aquest motiu és important realitzar la calibració en un ambient net.

Una vegada calibrat ja es poden prendre mesures mitjançant la funció `gas.Sensor.getPPM()` de la llibreria.

A3. Llista de comandes AT pel mòdul ESP8266-01

ESP8266 AT Command Set

Function	AT Command	Response
Working	AT	OK
Restart	AT+RST	OK [System Ready, Vendor:www.ai-thinker.com]
Firmware version	AT+GMR	AT+GMR 0018000902 OK
List Access Points	AT+CWLAP	AT+CWLAP +CWLAP:(4,"RocheFortSurLac",-38,"70:62:b8:6f:6d:58",1) +CWLAP:(4,"LiliPad2.4",-83,"f8:7b:8c:1e:7c:6d",1) OK
Join Access Point	AT+CWJAP? AT+CWJAP="SSID","Password"	Query AT+CWJAP? +CWJAP:"RocheFortSurLac" OK
Quit Access Point	AT+CWQAP=? AT+CWQAP	Query OK
Get IP Address	AT+CIFSR	AT+CIFSR 192.168.0.105 OK
Set Parameters of Access Point	AT+ CWSAP? AT+ CWSAP= <ssid>,<pwd>,<chl>,<ecn>	Query ssid, pwd chl = channel, ecn = encryption
WiFi Mode	AT+CWMODE? AT+CWMODE=1 AT+CWMODE=2 AT+CWMODE=3	Query STA AP BOTH
Set up TCP or UDP connection	AT+CIPSTART=? (CIPMUX=0) AT+CIPSTART = <type>,<addr>,<port> (CIPMUX=1) AT+CIPSTART= <id><type>,<addr>,<port>	Query id = 0-4, type = TCP/UDP, addr = IP address, port= port
TCP/UDP Connections	AT+ CIPMUX? AT+ CIPMUX=0 AT+ CIPMUX=1	Query Single Multiple
Check join devices' IP	AT+CWLIF	
TCP/IP Connection Status	AT+CIPSTATUS	AT+CIPSTATUS? no this fun
Send TCP/IP data	(CIPMUX=0) AT+CIPSEND=<length>; (CIPMUX=1) AT+CIPSEND= <id>,<length>	
Close TCP / UDP connection	AT+CIPCLOSE=<id> or AT+CIPCLOSE	
Set as server	AT+ CIPSERVER= <mode>[,<port>]	mode 0 to close server mode; mode 1 to open; port = port
Set the server timeout	AT+CIPSTO? AT+CIPSTO=<time>	Query <time>0~28800 in seconds
Baud Rate*	AT+CIOBAUD? Supported: 9600, 19200, 38400, 74880, 115200, 230400, 460800, 921600	Query AT+CIOBAUD? +CIOBAUD:9600 OK
Check IP address	AT+CIFSR	AT+CIFSR 192.168.0.106 OK
Firmware Upgrade (from Cloud)	AT+CIUPDATE	1. +CIPUPDATE:1 found server 2. +CIPUPDATE:2 connect server 3. +CIPUPDATE:3 got edition 4. +CIPUPDATE:4 start update
Received data	+IPD	(CIPMUX=0): + IPD, <len>: (CIPMUX=1): + IPD, <id>,<len>: <data>
Watchdog Enable*	AT+CSYSWDTENABLE	Watchdog, auto restart when program errors occur: enable
Watchdog Disable*	AT+CSYSWDTDISABLE	Watchdog, auto restart when program errors occur: disable

Font: <http://www.pridopia.co.uk/>

A5. Enllaços Datasheets

GL5528:

<http://akizukidenshi.com/download/ds/senba/GL55%20Series%20Photoresistor.pdf>

LM1117T 3.3 : <http://www.ti.com/lit/ds/symlink/lm1117.pdf>

DHT22

<https://www.sparkfun.com/datasheets/Sensors/Temperature/DHT22.pdf>

MQ135: <https://www.olimex.com/Products/Components/Sensors/SNS-MQ135/resources/SNS-MQ135.pdf>

OPA344: <http://www.ti.com/lit/ds/symlink/opa344.pdf>

Micròfon: <https://cdn.sparkfun.com/datasheets/Sensors/Sound/CEM-C9745JAD462P2.54R.pdf>

MICS

2714:

https://www.sgxsensortech.com/content/uploads/2014/08/1107_Datasheet-MiCS-2714.pdf

ESP8266-01 : <http://download.arduino.org/products/UNOWIFI/0A-ESP8266-Datasheet-EN-v4.3.pdf>