

**Escola Tècnica Superior d'Enginyeria
Electrònica i Informàtica La Salle**

Treball Final de Màster

Màster Universitari en Recerca en les TIC i la seva Gestió

Smart Cities: Monitorització dels moviments d'individus en una ciutat

Alumne

Javier Comendero Cuenca

Professor Ponent

Agustín Zaballos Diego

ACTA DE L'EXAMEN DEL TREBALL FI DE MASTER

Reunit el Tribunal qualificador en el dia de la data, l'alumne

D. Javier Comendeiro Cuenca

va exposar el seu Treball de Fi de Màster, el qual va tractar sobre el tema següent:

Smart Cities: Monitorització dels moviments d'individus en una ciutat

Acabada l'exposició i contestades per part de l'alumne les objeccions formulades pels Srs. membres del tribunal, aquest valorà l'esmentat Treball amb la qualificació de

Barcelona,

VOCAL DEL TRIBUNAL

VOCAL DEL TRIBUNAL

PRESIDENT DEL TRIBUNAL

Abstract

Aquest projecte consistirà en analitzar com els individus d'una ciutat es mouen per aquesta i quin és l'estat de la ciutat, amb la finalitat de millorar la salut i la vida dels ciutadans que hi viuen. Primer es farà un estudi teòric sobre quines tecnologies actuals tenim per rastrejar el recorregut dels individus i quins sensors són els més adients per controlar l'estat d'aquesta ciutat. Després es durà a terme la part pràctica del projecte que tractarà tant la construcció del propi sistema de rastreig i de control de la ciutat, com del processament de les dades obtingudes a partir d'aquest.

Este proyecto consistirá en analizar cómo los individuos de una ciudad se mueven por esta y cuál es el estado de la ciudad, con el fin de mejorar la salud y la vida de los ciudadanos que viven. Primero se hará un estudio teórico sobre qué tecnologías actuales tenemos para rastrear el recorrido de los individuos y cuáles sensores son los más adecuados para controlar el estado de esta ciudad. Después se llevará a cabo la parte práctica del proyecto que tratará tanto la construcción del propio sistema de rastreo y de control de la ciudad, como del procesamiento de los datos obtenidos a partir de este.

This project will consist of analyzing how the individuals of a city move through it and what is the state of the city, in order to improve the health and life of the citizens who live. First, a theoretical study will be carried out on what current technologies we have to track the path of individuals and which sensors are the most appropriate to control the state of this city. Then the practical part of the project will be carried out, which deal with both the construction of the cities will own tracking and control system, as well as the processing of the data obtained from it.

Agraïments

Voldria agrair a diverses persones l'ajuda que m'han ofert en la realització d'aquest Treball de Fi de Màster (TFM):

En primer lloc al meu director de TFM, Agustín Zaballos Diego, per haver confiat en mi en la realització d'aquest projecte i per l'ajuda que he rebut en moments de màxima dificultat en el projecte. Moltes gràcies Agustí.

En segon lloc a la meva família, al meu pare, la meva mare i el meu germà, per tot el suport constant i totes les mostres d'afecte que m'han donat. Moltes gràcies de tot cor.

Finalment als meus amics, per tots els bons moments que vam passar junts. La seva proximitat, sincera amistat, i els simpàtics i inoblidables moments que continuem vivint han fet que l'escriptura d'aquest TFM hagi estat molt més suportable en aquests últims mesos. A Àlex Sistac, Àlex Martínez, Adrià Cenzano, Byron Garzón, Adrián Carpallo, Àngel Jiménez, Juan Carlos Martínez, Raúl Godino i Carlos Rodríguez, per aguantar-me i suportar-en moments de màxima tensió i de màxima treball amb el projecte. Moltes gràcies nois, sou els millors!

Contingut

Contingut	5
1 Objectius del treball	8
2 Plantejament del projecte.....	10
3 Introducció	11
3.1 Definició de <i>Smart City</i>	12
3.2 Influència de les TIC en la prestació de serveis millors	13
3.3 Els nous <i>Smart Services</i>	15
3.3.1 Smart mobility	15
3.3.2 Smart environment.....	17
3.3.3 Smart living	17
3.3.4 Smart governance.....	18
3.3.5 Smart economy.....	19
3.3.6 Smart people	19
3.4 Projecte THAMOS: “Towards Healthy smArt MetropolISalthy”	20
4 Projecte realitzat	22
4.1 Descripció del projecte	22
4.2 Tecnologies utilitzades.....	24
4.2.1 Obtenció de coordenades (GPS).....	24
4.2.2 Sistema de comunicació (WiFi).....	29
4.2.3 Sensor de temperatura i humitat	31
4.2.4 Sensor detector de contaminació	33
4.2.5 Sensor de llum	36
4.2.6 Sensor de so.....	37
4.2.7 Arduino Mega 2560	38
5 Muntatge del projecte	42
5.1 Esquema del projecte	42
5.2 Interconnexió del hardware	43
5.3 Programació amb Arduino (components)	45
5.3.1 Diagrama GPS i NodeMCU.....	45
5.3.2 Diagrama Sensor DHT11.....	47
5.3.3 Diagrama Sensor de gas MQ	48

5.3.4	Diagrama Sensor de llum LDR	49
5.3.5	Diagrama Sensor de soroll CN07	50
5.4	Base de dades	51
5.4.1	MariaDB.....	51
5.4.2	MySQL.....	52
5.4.3	MongoDB.....	53
5.4.4	SQLite.....	54
5.4.5	Firestore	54
5.4.6	Base de dades escollida.....	55
5.5	Configuració de la base de dades Firestore.....	57
5.6	Programació amb Arduino (Connexió Hardware-FireBase)	60
5.7	Base de dades Firestore.....	63
5.8	Visualització	65
5.8.1	Mapa amb recorregut	65
6	Proves del projecte.....	68
6.1	Prova del sensor LDR	68
6.2	Prova del sensor CN07	69
6.3	Prova del sensor MQ-7	69
6.4	Prova del sensor DHT11.....	70
6.5	Prova de la geolocalització.....	71
6.6	Prova del projecte sencer	72
7	Estudi econòmic del projecte.....	74
8	Problemes i consideracions del projecte	76
9	Conclusions.....	77
10	Línies de futur	79
11	Bibliografia.....	80
12	Annex.....	83
12.1	Codi Arduino	83
12.1.1	Codi GPS i NodeMCU	83
12.1.2	Codi Sensor DHT11	84
12.1.3	Codi Sensor de gas MQ.....	84
12.1.4	Codi Sensor de llum LDR.....	85

12.1.5	Codi Sensor de soroll CN07	85
12.1.6	Codi Arduino (Conexió Hardware-FireBase).....	86
12.1.7	Codi Arduino (Complert)	88
12.2	Base de dades Firebase.....	91
12.3	Visualització de les dades	92
12.3.1	Codi HTML del mapa.....	92
12.3.2	Codi del script del mapa	93

1 Objectius del treball

Abans de començar a parlar del projecte que hem triat, deixarem fixats els objectius que volem assolir:

En primer lloc, i l'objectiu més important, busquem millorar la salut i la vida de la gent. Directament amb aquest projecte no millorem la vida de ningú, però si que és molt important els resultats que podem obtenir de cara a futurs projectes.

En segon lloc, i relacionat amb el primer, estem molt interessats en els moviments de la gent per la ciutat. És fonamental saber per on circulen bicicletes i vianants per la ciutat, per saber en una ciutat quins són els nuclis més actius i d'aquesta manera obrar en conseqüència, per exemple incrementant en aquestes zones recursos i serveis i d'aquesta manera fer un ús més òptims d'aquests.

En tercer lloc, volem aplicar coneixements tècnics a les ciutats per modernitzar-les. Com enginyers que som, volem buscar solucions aplicant tecnologies a problemes que existeixen actualment.

En quart lloc, estem molt interessats en l'aprenentatge de noves tecnologies que s'utilitzen actualment. Farem una exhaustiva recerca de tecnologies per poder obtenir les dades que necessitem, gestionar-les i finalment visualitzar-les.

En cinquè lloc, i ja que som enginyers electrònics de telecomunicacions, fer una obtenció de dades a partir d'algun sistema amb hardware, és a dir, treballar i provar diferents components electrònics i d'aquesta manera entendre quins són més adients i més útils en aquest projecte.

En sisè lloc, aprendre com funciona un sistema GPS i fer un estudi de les diferents formes d'obtenció de coordenades i veure quina encaixa millor amb el nostre projecte.

En setè lloc, consolidar coneixements d'assignatures que hem treballat durant la carrera sobre els diferents sensors que utilitzarem en aquest projecte (luminositat, temperatura, humitat).

En vuitè lloc, aprofundir en coneixements sobre bases de dades i emmagatzematge de grans quantitats d'informació, ja que en la meua titulació no n'hem treballat molt.

En novè lloc, conèixer el significat d'una Smart City. Com hem comentat anteriorment, les ciutats cada vegada tendeixen més a estar monitoritzades per tecnologies TIC, doncs ens resulta interessant saber què significa que una ciutat és intel·ligent, quins serveis intel·ligents tenen, com poden fer millorar la ciutat, etc.

Finalment, aprendre a desenvolupar un projecte de dimensions considerables, deixant de banda els diferents casos a nivell acadèmic que ens hem trobat durant la nostra formació universitària i portant un projecte que podria aproximar-se als que ens trobaríem en un futur en el món laboral.

2 Plantejament del projecte

A continuació, comentarem com plantejarem aquest projecte. Com podem observar a l'abstract, aquest projecte tractarà la mobilitat de les persones d'una ciutat com Barcelona, a peu i amb bicicleta, a més d'altres paràmetres com la contaminació, temperatura i humitat entre d'altres.

Com hem comentat anteriorment, volem resoldre els aspectes del projecte utilitzant diferents tipus de tecnologies, és a dir, a la llarga el que voldrem serà dotar a la ciutat de sistemes tecnològics per optimitzar els recursos i serveis dels ciutadans millorant així la seva vida. Per això, és molt important introduir el concepte "Smart City" en aquest treball. Explicarem tots els aspectes necessaris per poder comprendre què és, com es divideixen els diferents serveis, cap a on evolucionen, etc. A més tractarem l'origen d'aquest projecte, un projecte relacionat amb Smart Cities anomenat ***Thamos: Towards Healthy smArt MetropolISalthy***.

Una vegades posats en context, començarem a dissenyar el nostre projecte. Fent recerca i analitzant cadascuna de les diferents opcions que tenim, prenent així les eleccions més adients. La part pràctica del projecte la dividirem en 3 parts:

- **Obtenció de dades:** Mitjançant un sistema hardware ens encarregarem d'obtenir totes les dades que volem tractar, des de les pròpies coordenades on es troba el ciutadà fins a tots els sensors que utilitzarem per saber com es troba la ciutat d'estudi.
- **Emmagatzematge de dades:** Una vegada obtingudes les dades com hem comentat anteriorment, les emmagatzemem en una base de dades.
- **Visualització de dades:** Finalment visualitzarem les dades emmagatzemades en la base de dades. Per la mobilitat del individu utilitzarem un mapa per marcar el seu recorregut, i la resta de dades les visualitzarem directament des de la base de dades.

Finalment, farem un estudi econòmic on observarem quin és el pressupost que necessitem per dur a terme el projecte.

3 Introducció

Aquest projecte, no deixa de ser un projecte relacionat amb les *Smart Cities*, ja que en el fons de la qüestió, tenim una ciutat que volem convertir en intel·ligent mitjançant diferents tecnologies. Així doncs, els primers conceptes que haurem d'analitzar seran tots aquells relacionats amb les ciutats intel·ligents. A més, també analitzarem d'on sorgeix la idea d'aquest projecte, explicant en que consisteix el projecte ***Thamos: Towards Healthy smArt MetropolISalthy***.

El concepte *Smart City* sorgeix recolzat en diferents conferències i cimeres en què es tracten els problemes mediambientals i els relacionats amb el clima. Aquest terme té la seva base en l'Informe Brundtland de 1987, on es va parlar per primera vegada de el terme desenvolupament sostenible. *Smart City* i desenvolupament sostenible són dos termes que basen la seva definició la una a l'altra, ja que una ciutat sostenible és aquella que mitjançant l'ús de l'avançada tecnologia actual, es preocupa per reduir costos mediambientals, millorar la qualitat dels seus habitants i ser més dinàmica amb els mateixos o menys recursos. D'altra banda, moltes persones poden pensar que les *Smart Cities* són un repte a llarg termini, però tot el contrari, és una cosa a l'abast de moltes de les ciutats i que per això no ha d'haver impediments per aconseguir aquest reconeixement, sinó tot el contrari, ja que cada vegada és més necessari per la contínua i imparable emigració del camp a la ciutat, saturant aquestes cada vegada més.

D'altra banda, dins de les ciutats, ens trobem els serveis públics urbans. Aquest terme ve a referir-se als anomenats comunament, serveis municipals. Aquests serveis, són realitzats per l'Ajuntament de cada localitat i entre aquests es troben alguns com són el transport, abastament d'aigües, manteniment de zones verdes i paviment, subministraments o recollida d'escombraries.

A l'entorn de la *Smart City*, han d'existir aquests serveis però realitzats d'una manera *Smart*, és a dir, de manera que s'utilitzin els recursos municipals de la millor manera per tal que aquesta ciutat passi a ser molt més eficient que en l'actualitat. D'aquí, cal destacar alguns nous termes encunyats per referir-se als serveis públics d'aquestes noves ciutats. Aquests nous termes (smart mobility, smart people, smart environment, smart governance i smart living) es poden observar en l'estratègia que persegueix la Unió Europea.

Un cop contextualitzat el tema a tractar, s'estableixen els objectius d'aquest Treball de Fi de Màster. Primer de tot, l'objectiu general d'aquest projecte és conèixer i comprendre el concepte de *Smart City* amb la finalitat de realitzar un bon pla per convertir la ciutat de Barcelona en un territori *Smart*.

Una vegada que s'ha fixat aquest, es procedirà a establir diferents objectius específics que responguin a les següents preguntes:

- Què és una *Smart City*?
- Quin paper tenen els serveis públics en les *Smart Cities*?
- Com una *Smart City* pot contribuir a millorar la prestació dels serveis públics?
- Quins beneficis es deriven per la ciutadania?
- Cap a on evolucionen aquestes ciutats intel·ligents?

3.1 Definició de *Smart City*

L'expressió *Smart City* es tradueix directament del anglès com *Ciutat Intel·ligent*. És un concepte emergent, i per tant les seves accepcions en espanyol i en altres idiomes, i fins i tot en el propi idioma anglès, estan subjectes a constant revisió. És també un terme actual, que s'està utilitzant com un concepte de màrqueting en l'àmbit empresarial, en relació a polítiques de desenvolupament, i pel que fa a diverses especialitats i temàtiques.

La *Ciutat Intel·ligent* de vegades també anomenada *Ciutat Eficient* o *Ciutat Súper-eficient*, es refereix a un tipus de desenvolupament urbà basat en la sostenibilitat que és capaç de respondre adequadament a les necessitats bàsiques d'institucions, empreses, i de els mateixos habitants, tant en el pla econòmic, com en els aspectes operatius, socials i ambientals.

Les ciutats intel·ligents, donat el seu origen natural de les *Ciutats Digitals*, es basa en l'ús intens de les Tecnologies de la Informació i Comunicació (TIC) en prestació de serveis públics d'alta qualitat i calidesa, seguretat, productivitat, competitivitat, innovació, emprenedoria, participació, formació i capacitació.

Una ciutat o complex urbà podrà ser qualificat d'intel·ligent en la mesura que les inversions que es realitzin en capital humà (educació permanent, ensenyament inicial, ensenyament mitjà i superior, educació d'adults ...), en aspectes socials, en infraestructures d'energia (electricitat, gas), tecnologies de comunicació (electrònica, Internet) i infraestructures de transport, contemplin i promoguin una qualitat de vida elevada, un desenvolupament econòmic-ambiental durable i sostenible, una governança participativa, una gestió prudent i reflexiva dels recursos naturals, així com un bon aprofitament de el temps dels ciutadans.

Les ciutats modernes, basades en infraestructures eficients i durables d'aigua, electricitat, telecomunicacions, gas, transports, serveis d'urgència i seguretat, equipaments públics, edificacions intel·ligents d'oficines i de residències, etc., s'han d'orientar a millorar el confort dels ciutadans , sent cada vegada més eficaços i brindant nous serveis de qualitat, mentre que es respecten al màxim els aspectes ambientals i l'ús prudent i en declivi dels recursos naturals no renovables.

En efecte, una ciutat o un territori que es consideri intel·ligent es manifesta fonamentalment pel seu caràcter multidimensional i polifacètic, en termes d'actors, pel que fa a dominis clau (transports, energia, educació, salut, residus, vigilància, economia, ...), i en desenvolupament i utilització de tecnologies. Naturalment, els principals actors de el sector privat implicats en un projecte com l'analitzat de ciutat, territori, barri o edifici intel·ligent, són els industrials i empresaris de sectors clau, com ara energia, aigua, transports, i serveis, així com dirigents d'empreses públiques de telecomunicacions i infraestructures, editors, proveïdors de logística de suport a la gestió, així com a l'administració i la consultoria.

3.2 Influència de les TIC en la prestació de serveis millors

Dins de tots els municipis, hi ha una sèrie de serveis públics que són prestats per l'Ajuntament de el lloc. Aquests serveis, tenen un paper molt rellevant en el procés de transformació d'una ciutat convencional fins convertir-se en una urbs intel·ligent. Atès

que aquestes noves ciutats estan íntimament lligades a la tecnologia, els seus serveis públics per tant també ho estaran, de manera que s'ha passat en a penes uns anys, d'unes ciutats amb escàs ús de la tecnologia a aquelles on la tecnologia està present en cada racó de les mateixes. Els serveis públics juntament amb les noves tecnologies són una part fonamental de el desenvolupament de les ciutats. Gràcies a les TIC, molts serveis com els energètics, de transport i d'informació es poden veure molt beneficiats. És important el fet de la rapidesa de l'evolució de les tecnologies, ja que fa tot just deu anys, molta de la tecnologia de què disposem no existia, així que de la mateixa manera, l'actual, es veurà desplaçada en el termini d'una dècada per unes noves tecnologies.

És curiós que tot i estar en una època en què la informatització de la vida és cada vegada més gran, hi ha una quantitat ínfima a Europa d'edificis intel·ligents, tan sols l'1% dels edificis de nova construcció es consideren com a tal. Aquest aspecte de l'alt percentatge d'edificis no intel·ligents és una cosa preocupant, ja que els edificis són uns grans emissors de gasos contaminants a l'atmosfera.

De la mateixa manera, en moltes de les infraestructures ja construïdes i dels serveis amb prou feines es pot percebre la digitalització. Aquest fet és una cosa impactant, ja que els grans aliats de les ciutats en aquest aspecte són els seus ciutadans. Sent els habitants d'una ciutat en molts casos el principal escull a superar, resulta estrany que estant aquests gairebé 24 hores del dia pendants de diferents tecnologies, aquests serveis urbans en moltes ciutats es resisteixin a adoptar aquesta digitalització.

Per adoptar les tecnologies en aquests nous serveis, han sorgit diferents projectes que serviran en un futur proper a millorar l'eficiència de les ciutats. Existeixen certes barreres que impedeixen la implantació de les TIC a les ciutats. En primer terme, tot i existir moltes tecnologies que encara no han estat introduïdes en la societat, en moltes ocasions es vol implantar un pla cap als serveis urbans que inclouen tecnologies encara no desenvolupades o sense la suficient certesa que aquestes funcionin correctament. També, les ciutats han de comprendre que fins i tot podent comprar aquestes tecnologies, han de tenir en compte aquelles que li resultaran més beneficioses, ja que en moltes ocasions, les ciutats adquireixen una tecnologia que resulta no ser adequada per a la seva gestió de serveis, pel que han patit una pèrdua de recursos financers importants que es podria destinar a una altra partida. Una altra de les barreres més importants és la influència de les ciutats sobre alguns serveis que estan en mans d'empreses, ja que aquests s'han privatitzat, com és el cas del transport urbà o el subministrament d'aigua en moltes ciutats. Finalment, una de les grans barreres és el

referent a la privacitat i seguretat de les dades que s'han obtingut. En una època en què les dades significa poder i coneixement, és important salvaguardar i respectar la privacitat, sobretot els referents a les persones.

3.3 Els nous *Smart Services*

La Unió Europea, a través del seu Parlament, va aprovar l'any 2014 les bases que s'asseurien aquests nous serveis, els quals estan dividits en sis àrees que comprenen la mobilitat urbana, els ciutadans, el medi ambient, el govern i els serveis de seguretat, sanitat i salut. A partir d'aquesta desagregació, es creen els següents serveis intel·ligents: smart mobility, smart environment, smart living, smart governance, smart economy i smart people. Aquests termes, encunyats pel Parlament Europeu el 2014, es troben recollits en l'European Smart City Project, establint d'aquesta manera les característiques que ha de tenir tota Smart City per poder ser considerada com a tal i establir la seva estratègia a seguir.

3.3.1 Smart mobility

Smart mobility és com es coneix a la mobilitat intel·ligent. Aquest concepte es refereix a l'ús de mitjans de transport alternatius a l'ús individual del vehicle privat. A la pràctica, Smart mobility pot donar-se de formes diferents, com compartir el cotxe, desplaçar-se en transport públic, els patinets elèctrics, caminar o anar amb bicicleta, entre d'altres.

La necessitat de mobilitat intel·ligent va sorgir de la creixent congestió de trànsit i els seus efectes secundaris relacionats, incloent-hi la contaminació, les morts i el temps perdut. No es pot parlar de Smart mobility sense esmentar a les ciutats intel·ligents, que són les que estan contribuint a augmentar la popularitat d'aquest tipus de solucions, al fer-les possibles en la pràctica.

A mesura que la urbanització i la població de les ciutats han augmentat, el trànsit per carretera s'ha convertit en un gran problema a tot el món. Però construir més carreteres i expandir la capacitat, la primera de les solucions que es van posar en marxa, només va conduir a un augment del trànsit, un cas clàssic de demanda induïda.

Les aplicacions mòbils van ser una altra idea, dissenyada per ajudar a trobar rutes alternatives. No obstant això, com aplicacions en realitat estaven empitjorant el trànsit i augmentant la congestió als carrers laterals.

El concepte de Smart mobility inclou una àmplia gamma de modes de transport: patinets, bicicletes (normals, elèctriques, plegables), autobusos, trens lleugers, metres, tramvies, taxis, vehicles autònoms, caminar ... la llista no deixa de créixer amb l'avantatge que, a diferència del que succeeix amb la majoria de vehicles, que són propietat del conductor principal, amb la majoria d'opcions per al desplaçament dins de l'ecosistema de la mobilitat intel·ligent els usuaris tenen l'opció de posseir o compartir.

Smart mobility va més enllà de les formes alternatives de transport. La mobilitat intel·ligent es basa en els següents principis:

- **Flexibilitat:** les múltiples maneres de transport permeten als viatgers triar quins funcionen millor per a una situació determinada.
- **Eficiència:** el viatge porta a el viatger al seu destí amb una interrupció mínima i en el menor temps possible.
- **Integració:** la ruta completa es planifica porta a porta, independentment dels mitjans de transport que s'utilitzin.
- **Tecnologia neta:** el transport s'allunya dels vehicles que causen contaminació als de zero emissions.
- **Seguretat:** les morts i accidents es redueixen dràsticament.

3.3.2 Smart environment

Les comunitats i les ciutats són entitats que, per la seva dimensió, són capaços de consumir un gran volum de recursos, de generar grans quantitats de residus i d'emetre un gran nombre de gasos relacionats amb l'efecte hivernacle. Una universitat amb un gran campus, com la Universitat d'Alacant, és conscient de la necessitat d'optimitzar els seus recursos, de crear un entorn més verd, més net i més eficient, potenciant projectes i iniciatives capaços d'establir les bases per a la seva reconversió cap a una comunitat més ecològica i smart.

Per això s'estan implantant iniciatives orientades a ser un referent en sostenibilitat, protecció ambiental, racionalització, control d'energia i en la implantació de TI, el principal objectiu serà dotar d'intel·ligència als edificis, parcs i xarxes de sanejament, la qual cosa permeti una interacció entre si i una capacitat de prendre decisions autònomes enfocades a respectar el medi ambient i optimitzar els recursos naturals.

Per aconseguir els objectius desitjats, Smart Environment se centra en els següents àmbits als quals podem dotar d'intel·ligència a partir de solucions TIC:

- **Energia** (Smart Grid): consum i eficiència energètica.
- **Aigua** (Smart Water): control, gestió i optimització de l'aigua.
- **Residus** (Smart Waste): control i sensorització de contenidors. Monitorització de flotes encarregades de la recollida de residus.
- **Medi Ambient** (Smart Green): monitorització de la pol·lució, el soroll, el medi natural i perceptual, eco-edificis sostenibles.

3.3.3 Smart living

Mitjançant smart living, es pretén fer un ús responsable i eficient dels recursos de què disposa una ciutat per millorar la vida i la satisfacció dels seus habitants. D'aquesta definició sorgeixen diferents conceptes, un d'ells és l'edificació sostenible. Dins d'aquest tipus de nova construcció, s'engloben principalment quatre categories; seguretat, comunicacions, suport logístic i automatització de processos. Respecte a la seguretat, hi ha dos objectius. El primer d'ells consisteix a controlar els accessos, de

portes exteriors i interiors, a part de monitoritzar el flux de persones que entren i surten de l'edifici i cap a quines zones de la mateixa s'han desplaçat. Per aconseguir aquest objectiu, es planteja la utilització de targetes d'entrada o de radiofreqüència per localitzar els empleats. El segon objectiu, correspon a la vigilància mitjançant un sistema de vídeo intel·ligent amb detecció facial, visió nocturna i càmeres infraroges. Pel que fa a comunicació, suport logístic i automatització de processos, es tracta d'analitzar les condicions climàtiques dins de l'edifici (humitat, temperatura, qualitat de l'aire) per tal d'obtenir un aire net sense partícules nocives. També, mitjançant l'obtenció de dades, es pretén predir la utilització de l'edifici amb l'objectiu d'optimitzar els seus recursos, com ara llums intel·ligents, ascensors que analitzin la capacitat i demanda dels mateixos o el monitoratge de l'aparcament de l'edifici perquè l'usuari conegui de manera prèvia on pot aparcar.

Al costat de l'edificació sostenible, dins de smart living, es troben els serveis sanitaris, obtenint uns nous serveis intel·ligents destinats a la salut, smart health. Aquest nou terme fa referència a l'ús de diferents eines i dispositius mitjançant els quals es millorarà l'eficiència i la gestió de la sanitat, obtenint com a resultat una major interacció entre ciutadans, sistema de salut i el seu entorn. Amb aquest nou concepte de sanitat intel·ligent, es persegueix construir diferents plataformes de gestió per millorar l'estada dels malalts en hospitals, optimitzar la logística sanitària i addicionalment a tot això, es traslladarà la tecnologia a les llars per monitoritzar a malalts crònics a través de sensors o apps sanitàries.

3.3.4 Smart governance

El vincle entre societat i les TIC és una bretxa cada vegada més ínfima. No obstant això, les eines per a una millor governança a les ciutats estan encara per polir. D'aquesta manera, el futur en aquest àmbit demanda de solucions que impliquin polítiques i accions orientades a la gestió eficient i sostenible dels recursos amb la finalitat última d'afavorir a el benestar del conjunt de la ciutadania.

Els beneficis de l'Smart governance son:

- **Augmentar la participació ciutadana:** Amb les eines TIC, els governs poden comunicar-se directament amb els ciutadans. A més, el mateix govern podria rebre feedback i suggeriments dels programes, polítiques o accions dutes a terme a la ciutat.

- **Accedeix a informació crucial:** De la mateixa manera, els ciutadans podrien accedir a informacions relacionada amb fons, despeses i inversions. Excepte informació que afecti el desenvolupament de la ciutat en si, la resta ha de ser pública a tots els ciutadans.
- **Més democràcia:** Un Smart Governance pot implementar mesures d'identificació i autenticació per prevenir frau electoral.
- **Futur sostenible:** Les anàlisis obtinguts amb l'ajuda de la tecnologia ajudaran a planificar millors polítiques. Totes elles orientes a la conservació de recursos i medi ambient, desenvolupament de la comunitat, seguretat dels ciutadans, millor educació i ocupació i el benestar públic.

3.3.5 Smart economy

En aquest àmbit, hi ha uns elements claus que són el comerç electrònic, els e-business i l'economia col·laborativa. A més d'aquests elements claus, l'economia intel·ligent té una característica principal i és l'increment de l'PIB. Aquest increment es buscarà de diferents maneres, ja sigui atraient inversions de fora de la ciutat, mitjançant l'augment de la despesa dels turistes o un augment el nombre d'ells, o atraient nous habitants. Aquesta economia smart, a part de les característiques prèvies, té certs trets que van en diferents direccions. El primer d'ells té a veure amb l'R + D + I, les ciutats hauran de fomentar la investigació mitjançant les TIC i destinar una partida pressupostària a aquesta, ja que pot oferir solucions que altres stakeholders poden adaptar i crear amb això uns serveis innovadors. Un altre tret important és la de crear una marca de ciutat, és a dir, transformar la ciutat en una marca per atreure amb això aquest augment del PIB que tenen com a objectiu principal les ciutats.

3.3.6 Smart people

Aquest terme té a veure amb l'educació dels ciutadans. Els governs de les ciutats deuen, en cooperació amb altres organitzacions tant territorials com estatals, donar als seus ciutadans una correcta formació en matèries clau per al desenvolupament urbanístic de la ciutat i també en habilitats digitals. Donar formació en matèria digital és molt important en les noves perspectives de ciutat intel·ligent. Com s'exposa en smart governance, els ciutadans van a passar a formar part activa de la ciutat, no només en l'activitat de la mateixa, sinó també en la seva organització. La ciutadania s'unifica d'acord a diferents indicadors de benestar social com són el nivell d'educació,

la diversitat social, el nivell de cosmopolitisme de la ciutat, la creativitat o la participació en la vida pública.

Mitjançant la col·laboració i participació en la vida pública i en les decisions d'aquests municipis, s'aconseguirà prestar serveis que s'adaptin a les demandes de la població. Per aquest motiu, les tecnologies adquireixen un paper protagonista i tenen un gran pes en les ciutats de el futur, ja que a través d'ella, es farà arribar als seus ciutadans diferents informacions, tant administratives com del mig, de manera que aconseguiran millorar la seva qualitat de vida i perquè aquests prenguin decisions sobre diferents aspectes amb la informació rebuda en temps real, s'aconseguirà reduir la bretxa digital entre els nadius digitals i les persones més grans i es propiciarà un pensament creatiu, igualment en relació a aquest últim aspecte, es millorarà la formació dels ciutadans, ja que aquesta passarà a ser un aprenentatge continu al llarg de la vida dels habitants.

3.4 Projecte THAMOS: “Towards Healthy smArt MetropOliSalthy”

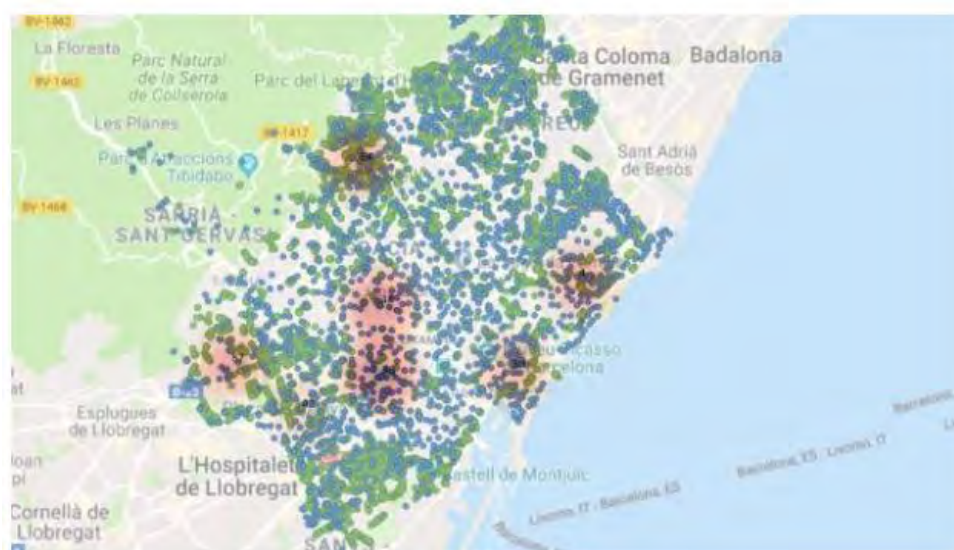
Com hem comentat anteriorment, a continuació explicarem en que consisteix el projecte *Thamos*, projecte del qual nosaltres intentarem solucionar una de les seves fases.

Al campus de La Salle Barcelona han iniciat el projecte de recerca “Towards Healthy smArt MetropOLiS (THAMOS): Data-driven multidisciplinary strategies for pursuing sustainable mobility”. Aquest projecte és una col·laboració transversal pionera entre els grups de recerca d'Enginyeria i Arquitectura (GRITS — Research Group in Internet Technologies & Storage, GRETEL — Research Group in Technology Enhanced Learning, and IARC/IAM — Research Group in Mediterranean Architecture) amb l'objectiu de crear nous estratègies per ajudar a les ciutats contemporànies en la transformació cap a una metròpoli intel·ligent i saludable, mitjançant l'explotació de dades tant de conjunts de dades existents com de sensors de camp.



Imatge 1: Projecte Thamos

Prop de dos terços dels europeus adults no són actius físicament als nivells recomanats per l'Organització Mundial de la Salut. La inactivitat física causa aproximadament 600.000 morts anuals a Europa, amb un cost elevat per a teràpies amb sobrepès i obesitat. La viabilitat i la ciclabilitat han aparegut recentment com a alternatives potents per enfortir els micro teixits urbans, promoure l'ocupació de l'espai, fomentar l'activitat física i contribuir positivament a la dinàmica del regne urbà, que millora la qualitat de vida dels ciutadans. Creiem que l'ajust d'espais urbans per augmentar la viabilitat i la ciclabilitat reduint l'impacte ambiental del transport motoritzat pot fomentar l'activitat física dels ciutadans i, per tant, millorar la qualitat de vida i les condicions de salut de la població.



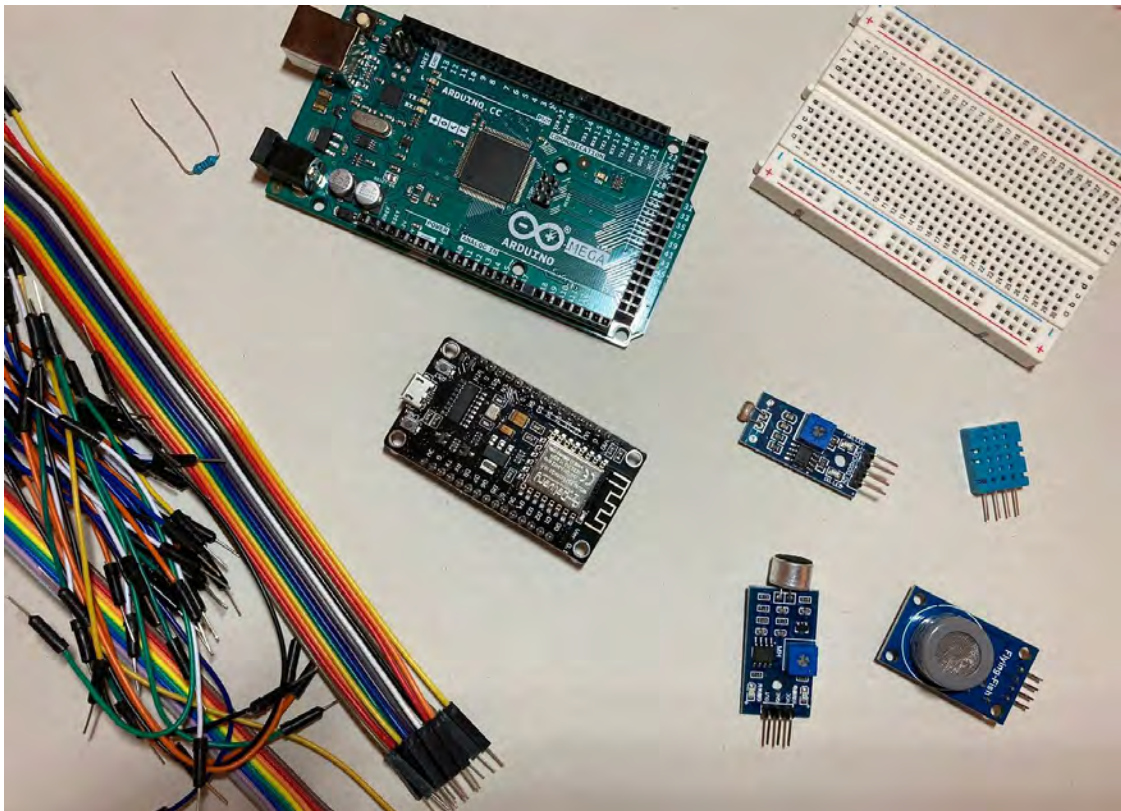
Imatge 2: Mobilitat de la ciutat

4 Projecte realitzat

4.1 Descripció del projecte

Com ja hem comentat en apartats anteriors, el projecte tractarà la mobilitat de les persones d'una ciutat com Barcelona, a peu i amb bicicleta, a més d'altres paràmetres com la contaminació, temperatura i humitat entre d'altres.

El projecte constarà dels elements que observarem en la següent imatge:



Imatge 3: Components del projecte

En aquesta imatge, observem com a element principal el NodeMCU, el qual serà el nostre component clau per resoldre la problemàtica en la que ens trobem. La seva funció serà obtenir les coordenades geogràfiques del individu que volem monitoritzar els seu moviments. També serà l'encarregat de gestionar les dades que rebran els diferents sensors que disposem.

Els sensors que surten a la imatge són:

- Sensor DHT11: Encarregat de rebre la temperatura i humitat del carrer.
- Sensor LDR: Encarregat de rebre la lluminositat del carrer.
- Sensor MQ-07: Encarregat de rebre la contaminació (emissions de CO₂) del carrer.
- Sensor CN07: Encarregat de rebre el nivell de soroll del carrer.

Utilitzarem un Arduino Mega 2560 únicament per utilitzar la font de tensió de 5V, ja que el NodeMCU només té font de 3,3V i molts sensors necessiten 5V per poder funcionar. Sabem que no seria necessari un Arduino Mega 2560 per únicament alimentar alguns sensors a 5V, al apartat de “Problemes i consideracions del projecte” expliquem quin motiu ens ha portat a utilitzar-lo.

Finalment, tenim una Protoboard i cables per aquesta, que seran necessaris per les diferents interconnexions entre el NodeMCU i el Arduino Mega amb els diferents sensors.

En el següent apartat (4.2 Tecnologies utilitzades), tractarem i justificarem cadascuna de les tecnologies hardware escollides en aquest projecte, tant els sensors com les plaques.

El funcionament i la interconnexió dels components, ho tractarem en el apartat número 4 (“Muntatge del projecte”).

4.2 Tecnologies utilitzades

4.2.1 Obtenció de coordenades (GPS)

El sistema de posicionament global, conegut com a GPS és un sistema de navegació per satèl·lit que permet saber amb molta precisió la mateixa situació geogràfica i l'hora de referència amb gran exactitud en gairebé qualsevol lloc de la Terra o en una Òrbita de la Terra. El GPS és un Sistema Global de Navegació per Satèl·lit (GNSS) que permet determinar a tot el món la posició d'una persona, un vehicle o una nau, amb una desviació mitjana de quatre metres.

La situació dels satèl·lits és coneguda pel receptor amb base a les efemèrides (5 elements orbitals), paràmetres que són transmesos pels mateixos satèl·lits. La col·lecció d'efemèrides de tota la constel·lació es completa cada 12 minuts i es guarda en el receptor GPS.

El receptor GPS mesura la seva distància fins als satèl·lits, i fa servir aquesta informació per calcular la seva posició. Aquesta distància es mesura calculant el temps que el senyal triga a arribar al receptor. Conegut aquest temps i basant-se en el fet que el senyal viatja a la velocitat de la llum (llevat d'algunes correccions que s'apliquen), es pot calcular la distància entre el receptor i el satèl·lit.

Cada satèl·lit indica que el receptor es troba en un punt en la superfície de l'esfera, amb centre en el propi satèl·lit i de radi la distància total fins al receptor. En obtenir informació de dos satèl·lits se'ns indica que el receptor es troba sobre la circumferència que és intersecció de les dues esferes.

Si adquirim la mateixa informació d'un tercer satèl·lit notem que la nova esfera només talla la circumferència anterior en dos punts. Un d'ells es pot descartar perquè ofereix una posició absurda. D'aquesta manera ja tindriem la posició en 3-D. No obstant això, atès que el rellotge que incorporen els receptors GPS no està sincronitzat amb els rellotges atòmics dels satèl·lits GPS, els dos punts determinats no són suficients.

Tenint informació d'un quart satèl·lit, eliminem l'inconvenient de la manca de sincronització entre els rellotges dels receptors GPS i els rellotges dels satèl·lits. I és en aquest moment quan el receptor GPS pot determinar una posició 3-D exacta (latitud, longitud i altitud). En no estar sincronitzats els rellotges entre el receptor i els satèl·lits, la intersecció de les quatre esferes amb centre en aquests satèl·lits és un petit volum en comptes de ser un punt. La correcció consisteix a ajustar l'hora del receptor de manera que aquest volum es transformi en un punt.

Una de les tasques que hem de realitzar en aquest projecte és la geolocalització de l'individu en la ciutat, rastrejant així la seva posició en cada moment. Així doncs fent una recerca exhaustiva trobem que els sensors més típics per resoldre aquesta tasca son els següents:

- SIM808 GPS GSM. Aquest mòdul costa al voltant de 50 € i incorpora la tecnologia GSM per enviar informació a través de les xarxes mòbils.
- Kuman SIMCOM SIM900. A l'igual que l'altre mòdul, també incorpora GSM per transmetre dades. El seu preu ronda els 50 €.
- GPS Neo-6 m: el seu preu convida a adquirir un d'ells ja que per només 6 € podem incorporar un d'aquests al nostre projecte. L'inconvenient és que no té connexió GSM.

Per un projecte de les nostres característiques, a priori, qualsevol dels 3 sensors són igual d'útils, tot i que la diferència de preus ens fa decantar per aquest últim, el GPS Neo-6m.



Imatge 4: GPS NEO6MV2

Els dispositius NEO-6 són una família de receptors fabricats per U-Blox, que poden ser connectats amb facilitat a un autòmat o processador com Arduino. Disposen d'interfície de comunicació UART, SPI, DDC (I2C) i USB. Suporten els protocols NMEA, UBX binary i RTCM.

La família de receptors GPS NEO-6 estan dissenyats per tenir una petita grandària, petit cost, i petit consum. La intensitat de corrent necessària és d'uns 37mA en mode de mesurament continu. La precisió que en posició és de 2.5m, en velocitat 0,1M / s i en orientació 0.5º, valors més que acceptables per a un sistema de posicionament GPS.

Com acabem de comentar, per poder treballar amb ell, necessitarem un processador com Arduino, del qual en parlarem més endavant.

Una vegada seleccionat el nostre mòdul GPS, és hora de fer les primeres proves. Utilitzant un codi bàsic, proporcionat pel mateix Arduino, i connectant el nostre mòdul amb la nostre placa Arduino, iniciem l'obtenció de coordenades geogràfiques. El resultat de les proves són molt negatives. El mòdul GPS no és capaç de establir connexió amb els satèl·lits per poder obtenir la geolocalització, així doncs haurem de substituir aquest mòdul per un altre.

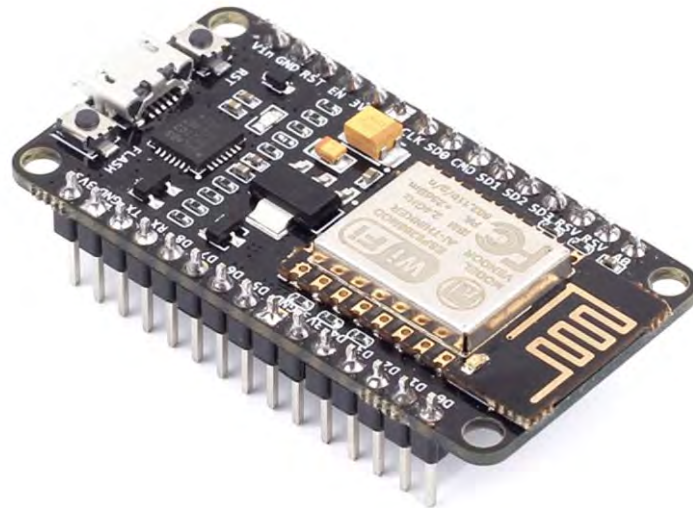
Tenint en compte que els altres mòduls són massa cars per poder dur a terme aquest projecte, el que farem serà buscar una altre forma de poder obtenir les coordenades.

Fent recerca, vam trobar una API de Google Maps, la qual mitjançant una connexió Wifi, et retornava la posició en la que ens trobem. Com més tard necessitarem enviar les dades a un servidor, hem optat finalment per aquesta possibilitat. Així doncs, la nostra forma d'obtenir les coordenades serà mitjançant un mòdul el qual ens permeti accedir a Internet per tal d'accedir a la API de Google Maps i per més tard emmagatzemar dades al servidor. Pel que fa a l'elecció del WiFi en aquest sistema de geolocalització, es tractarà en apartats posteriors.

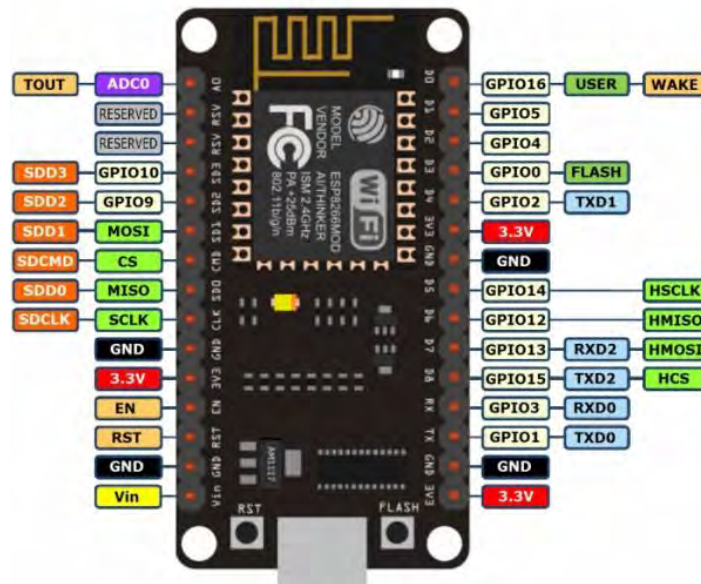
Mirant al mercat, un mòdul que complia aquestes característiques era el mòdul NodeMCU. NodeMCU va ser una de les primeres plaques de desenvolupament amb el microcontrolador ESP8266. Fins llavors aquest xip només estava disponible com plaques ESP-xx com ESP01 o ESP12.

NodeMCU es va popularitzar ràpidament perquè permetia programar aquest microcontrolador d'una manera molt més senzilla que utilitzant el kit de desenvolupament de Espressif. El seu disseny integra l'electrònica necessària. Per utilitzar-la no cal res més que un cable USB i un ordinador. Aquest concepte endollar i llest és el mateix que fan servir les plaques d' Arduino. En aquest projecte utilitzarem una geocalització per WiFi. Tots coneixem la geocalització per GPS. Aquest sistema utilitza satèl·lits per obtenir la ubicació d'un objecte, dispositiu o persona mitjançant la triangulació. També es poden utilitzar tecnologies com GSM, UWM o LPWAN.

Totes aquestes tecnologies requereixen d'un component o shield que ens de la capacitat de poder geocalitzar un dispositiu. Amb la geocalització WiFi aconseguim utilitzar aquest tipus de xarxes per ubicar la nostra placa a més de per connectar-nos a Internet.



Imatge 5: NodeMCU



Imatge 6: Pinout NodeMCU

4.2.2 Sistema de comunicació (WiFi)

Com hem comentat al apartat anterior, el sistema de comunicació que utilitzarem serà WiFi. Abans de continuar amb la tecnologia que utilitzarem en aquest projecte, voldríem justificar el perquè d'aquest sistema de comunicació, ja que aquesta decisió no sembla la més encertada en un projecte d'aquest caire.

El principal problema que tenim és que ens veiem molt limitats amb la tecnologia i els coneixements que disposem, ja que els sistemes que utilitzaríem, en cas que portéssim el projecte THAMOS en marxa, serien uns altres.

Una forma en la que podríem resoldre el problema seria emmagatzemar amb algun tipus de processador totes les dades de les coordenades i dels sensors en una memòria del tipus EEPROM, per exemple, durant tot el recorregut del individu. Més tard, aquestes dades emmagatzemades en aquesta memòria serien enviades via WiFi a la base de dades per posteriorment treballar digitalment amb elles. El principal problema pel qual no podríem posar en pràctica aquesta idea és que aquestes memòries es degraden molt amb el ús, i en el nostre cas voldríem reutilitzar aquest sistema moltes vegades. A part, un altre problema que trobem és l'aparositat de dur a terme el projecte en varies fases, primer una de guardar les dades en la memòria, i després extreure la memòria i fer una connexió a la base de dades, serien masses recursos utilitzats.

Una altre forma en la que podríem resoldre el projecte sense cap tipus de limitacions de coneixements ni tecnologia seria la utilització de xarxes de telefonia mòbil amb unes targetes que només serveixen per dades (no trucades). Per exemple, *Things Mobile* ven una que consumeix 0.1€ per Mb transmès. Res a comparar amb una de prepagament de Movistar, per exemple (0.25 € per SMS). Aquestes companyies utilitzen la infraestructura de les operadores de mòbil de cada país (per exemple, Movistar, Vodafone, Orange, Yoigo a Espanya), de manera que el problema de cobertures gairebé no existeix. Amb una SIM així i un mòdul barat de GSM / GPRS més un Arduino es pot aconseguir connectar i enviar dades a un servidor web on no hi hagi possibilitat de WiFi. Aquesta proposta la trobo força interessant i viable, el problema que per tota la recerca que he fet d'informació d'aquest sistema, aquest treball es podria centrar en només això.

Així doncs, arribem a la opció escollida. La nostra idea serà transmetre en “real time” les coordenades i les dades del sensor a la nostra base de dades mitjançant connexió WiFi.

Wi-Fi és una tecnologia de xarxa local sense fils que permet a un dispositiu electrònic intercanviar dades o connectar amb Internet sigui a 2,4GHz o 5GHz. Hi ha molts dispositius habilitats per utilitzar comunicació WiFi: ordinadors, impressores, videoconsoles, telèfons intel·ligents, càmeres digitals, tablets i reproductors digitals multimèdia, entre d'altres. Aquests es poden connectar a un recurs de xarxa com internet via un punt d'accés WiFi. Aquest punt d'accés WiFi té uns 20 metres en interior fins a alguns centenars en camp obert. La cobertura pot ser dins d'una habitació amb parets i obstacles que bloquegen el senyal, fins a diversos kilòmetres utilitzant punts d'accés intermedis amb antenes direccionals.

WiFi pot ser menys segur que les connexions per cable (Ethernet), ja que l'atacant no necessita ser físicament connectat. Les pàgines web que utilitzen SSL són segures però l'accés wifi sense encriptació és fàcilment detectat pels atacants. A causa d'això les xarxes sense fils han desenvolupat tècniques d'encriptació.

Aquest sistema de comunicació, com hem comentat anteriorment no és el més eficaç de tots, però si que podem resoldre el projecte sempre i quan es compleixi una única condició, que hi hagi connexió WiFi en la zona per on circula l'individu. En el meu cas, inicialment vaig utilitzar una xarxa WiFi d'un local d'uns coneguts meus, però vaig veure que ens limitàvem molt. Llavors, vaig informar-me de les xarxes WiFi de la meua ciutat i vaig trobar diferents àrees on hi havia una xarxa Wifi pública. La majoria de proves que he realitzat, com veurem en apartats posteriors, han estat en aquestes àrees. D'aquesta manera no he tingut cap problema de connexió amb el meu sistema.



Imatge 7: Connexió WiFi pública establerta

4.2.3 Sensor de temperatura i humitat

Dos dels paràmetres més típics per mesurar en un zona, terreny o ciutat són la temperatura i la humitat.

Fent recerca de sensors que compleixin aquestes dues característiques, vam obtenir aquests 2 molt coneguts: El DHT11 i el DHT22.

El DHT11 i el DHT22 són dos models d'una mateixa família de sensors, que permeten realitzar el mesurament simultània de temperatura i humitat. Aquests sensors disposen d'un processador intern que realitza el procés de mesurament, proporcionant el mesurament mitjançant un senyal digital, pel que resulta molt senzill obtenir el mesurament des d'un microprocessador com Arduino.

Tots dos sensors presenten un encapsulat de plàstic similar. Podem distingir dos models pel color de la mateixa. El DHT22 presenta una carcassa blanca, mentre que en el cas del sensor DHT11 l'exterior és blau. De tots dos models, el DHT22 és el germà petit de la família, i compta pitjors característiques tècniques. El DH11 és el model superior però, per contra, té un preu superior.

Les característiques de l'DHT22 són realment escasses, especialment en rang de mesurament i precisió.

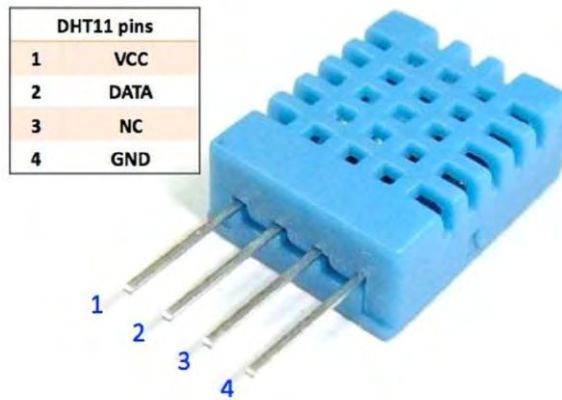
- Mesura de temperatura entre 0 a 50, amb una precisió de 2 ° C
- Mesura d'humitat entre 20 a 80%, amb precisió de el 5%.
- Freqüència de mostreig d'1 mostres per segon (1 Hz)

El DHT22 és un sensor molt limitat que podem usar amb fins de formació, proves, o en projectes que realment no requereixin un mesurament precís. Per contra, el model DHT11 té unes característiques molt més acceptables.

- Mesura de temperatura entre -40 a 125, amb una precisió de 0.5°C
- Mesura d'humitat entre 0 a 100%, amb precisió de l'2-5%.
- Freqüència de mostreig de 2 mostres per segon (2 Hz)

EL DHT11 (sense arribar a ser en absolut un sensor d'alta precisió) té unes característiques acceptables perquè sigui possible emprar-lo en projectes reals de monitoratge o registre, que requereixin una precisió mitjana.

Així doncs com la diferencia de preu no es cap bojeria, triarem el millor dels 2 sensors, el DHT11.



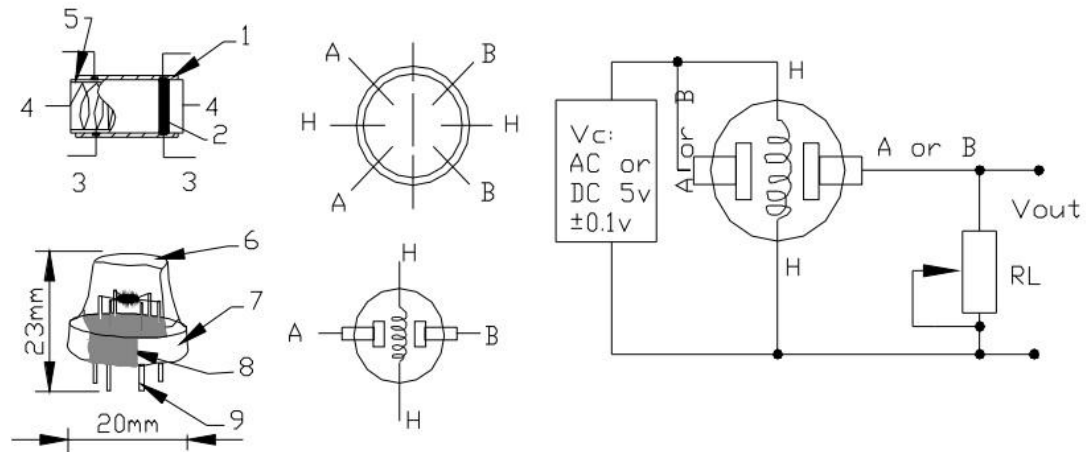
Imatge 8: Sensor DHT11

4.2.4 Sensor detector de contaminació

La contaminació en una ciutat és realment important, així doncs, volem controlar els nivells de contaminació. en el nostre cas de controlarem els nivells de CO, ja que segons diferents estudis que hem observat, és un gas bastant perjudicial per la salut a causa de la combustió incompleta d'alguns combustibles fòssils o biocostumbles.

Els sensors per controlar la presència de gasos són els de la sèrie MQ. Són sensors analògics pel que són fàcils d'implementar amb qualsevol microcontrolador.

Aquests sensors són electroquímics i varien la seva resistència quan s'exposen a determinats gasos, internament posseeix un escalfador encarregat d'augmentar la temperatura interna i amb això el sensor pugui reaccionar amb els gasos provocant un canvi en el valor de la resistència. L'escalfador depenent de el model pot necessitar un voltatge entre 5 i 2 volts, el sensor es comporta com una resistència i necessita una resistència de càrrega (RL) per tancar el circuit i amb aquest fer un divisor de contenció i poder llegir-lo des d'un microcontrolador.



Imatge 9: Esquema sensor MQ

Al mercat, generalment els sensors MQ es troben en mòduls, el que ens simplifica la part de connexions i ens faciliten el seu ús, només n'hi ha prou amb alimentar el mòdul i començar a llegir el sensor, aquests mòduls també tenen una sortida digital la qual internament treballa amb un comparador i amb l'ajuda d'un potenciòmetre podem calibrar el llindar i així poder interpretar la sortida digital com a presència o absència de el gas.

Segons el gas que vulguem detectar la seva presència, haurèm d'escollir un o un altre. A la següent taula deixem relaxat els diferents tipus que existeixen:

Modelo	Sustancias detectadas
MQ-2	Metano, butano, GLP, humo
MQ-3	Alcohol, Etanol, humo
MQ-303A	Alcohol, etanol, humo
MQ-4	Metano, gas natural comprimido (GNP)
MQ-5	Gas natural, GLP
MQ-6	Butano, GLP
MQ-306A	Butano, GLP
MQ-7	Monóxido de carbono
MQ-307A	Monóxido de carbono
MQ-8	Hidrógeno

Taula 1: Sensors de gas MQ

En el nostre cas, al voler controlar els nivells de CO, podríem escollir tan el MQ-7 com el MQ-307A. Com ambdós són iguals, finalment escollim el sensor MQ-7.



Imatge 10: Sensor MQ-7

4.2.5 Sensor de llum

Molt necessari també en una ciutat és saber el nivell de lluminositat que hi ha. Aquest paràmetre pot arribar a ser molt important, en casos, on per exemple un dia que plou molt i es dia però el carrer és fosc degut al cel ennuvolat. Si els fanals estan controlats que s'activen a una certa hora del dia (quan és de nit), en aquest cas el carrer serà fosc i la gent que hi circula pel carrer podria prendre mal. Si fem que els carrers s'il·luminin o no depenent del nivell de lluminositat, podem estalviar recursos i evitar problemes a la salut de la gent.

Per controlar el nivell de lluminositat del carrer, hem investigat i cercat 3 tipus: Sensor LDR, TSL2591 i BH1750.

Observant les característiques de tots 3, els sensors TSL2591 i el sensor BH1750 són molt millors i ofereixen moltes més possibilitats que el sensor simple LDR (fotoresistència). No obstant, el sensor LDR té un preu d'uns 0,73€ i els altres 2 es troben entre 10€ i 12€. Així doncs, com tampoc necessitem molta precisió i moltes possibilitats més enllà de controlar un nivell de foscor, el sensor LDR serà l'escollit.

El sensor LDR o fotoresistor és un component la resistència interna canviarà en funció de la llum percebuda. El seu comportament és el següent:

- Més llum = menor resistència elèctrica
- Menys llum = major resistència elèctrica



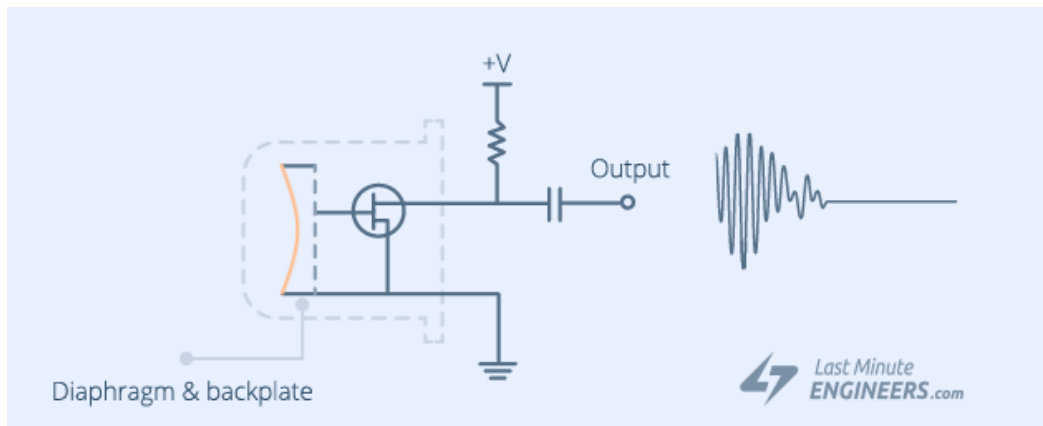
Imatge 11: Sensor LDR

4.2.6 Sensor de so

Finalment, és molt interessant tenir controlat els nivells de soroll en una ciutat, ja que superar els nivells de decibels pot perjudicar l'estat de salut dels ciutadans.

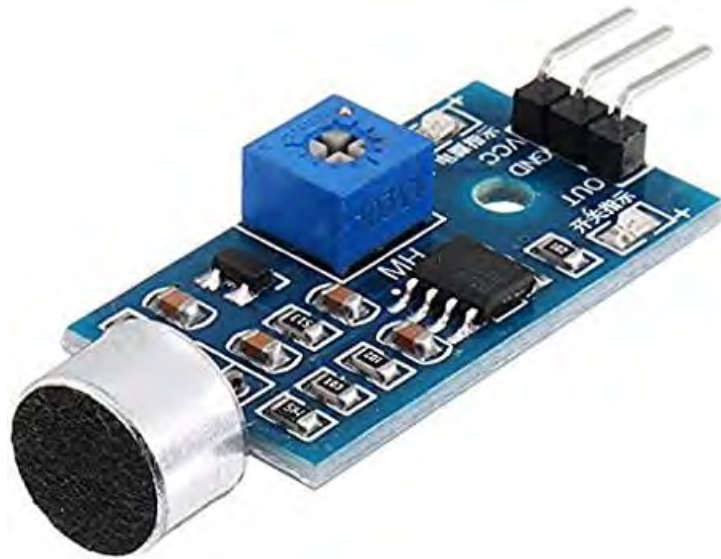
Per monitoritzar els nivells de soroll a la ciutat, segons la recerca que hem realitzat, els sensors més típics són el CN07 i el KY-038. Tots dos són sensors pràcticament iguals. El sensor KY-038, tot i ser més barat que l'altre, és força reduït i per tant si volguéssim que captés sons a més distància necessitaríem un amplificador. El preu dels sensors són aproximadament 0,83€ el KY-038 i aproximadament 1,65€ el CN07. Finalment, ens decantarem per aquest últim, el CN07.

Aquests sensors de so són econòmics, fàcils de connectar i són capaços de detectar sons de veu, clams o cops de porta. Dins del micròfon hi ha el prim diafragma, que en realitat és una placa d'un condensador. La segona placa és la placa posterior, que és propera i paral·lela al diafragma.



Imatge 12: Esquema Sensor de so

Quan parlem al micròfon, les ones sonores creades per la vostra veu impacten contra el diafragma i fan que vibri. Quan el diafragma vibra en resposta al so, la capacitat canvia a mesura que les plaques s'acosten o s'allunyen.



Imatge 13: Sensor CN07

4.2.7 Arduino Mega 2560

Finalment, trobem l'Arduino 2560 en el nostre projecte. Com ja hem comentat anteriorment, la idea inicial del projecte era treballar amb un mòdul Neo6mV2 per l'obtenció de les coordenades del individu, però al no tenir èxit de funcionament, vam optar per altres tecnologies.

El problema que teníem, a continuació, era que els diferents sensors necessitaven una alimentació de 5V i el nostre mòdul NodeMCU només podia alimentar a 3,3V. Així doncs necessitàvem una font d'alimentació a 5V i ens vam plantejar crear una font d'alimentació de 5V (mitjançant piles, per exemple), comprar-ne una o reutilitzar l'Arduino que ja el teníem degut a la primera idea d'obtenció de coordenades mitjançant el mòdul ja mencionat anteriorment.

Finalment, ens vam decantar per aquesta última idea. Així doncs, tot i que en aquest punt només necessitem l'Arduino Mega 2560 com a font d'alimentació a 5V, explicarem que és aquest processador i quines possibilitats té, ja que durant aquest projecte hem estat força temps treballant amb ell.

L'Arduino Mega 2560 és una targeta de desenvolupament de codi obert construïda amb un microcontrolador model Atmega2560 que posseeix pins d'entrades i sortides (I / S), analògiques i digitals. Aquesta targeta és programada en un entorn de desenvolupament que implementa el llenguatge Processing / Wiring. Arduino es pot utilitzar en el desenvolupament d'objectes interactius autònoms o pot comunicar-se a un PC a través del port sèrie (conversió amb USB) utilitzant llenguatges com Flash, Processing, MaxMSP, etc. Les possibilitats de realitzar desenvolupaments basats en Arduino tenen com a límit la imaginació.

L'Arduino Mega té 54 pins d'entrades / sortides digitals (14 de les quals poden ser utilitzades com a sortides PWM), 16 entrades anàlogues, 4 UARTS (ports serial per hardware), vidre oscil·lador de 16MHz, connexió USB, jack d'alimentació, connector ICSP i botó de reset. Arduino Mega incorpora tot el necessari perquè el microcontrolador treballi; simplement connecta'l al teu PC per mitjà d'un cable USB o amb una font d'alimentació externa (9 fins 12VDC). L'Arduino Mega és compatible amb la majoria dels Shields dissenyats per Arduino Duemilanove, Diecimila o UN.



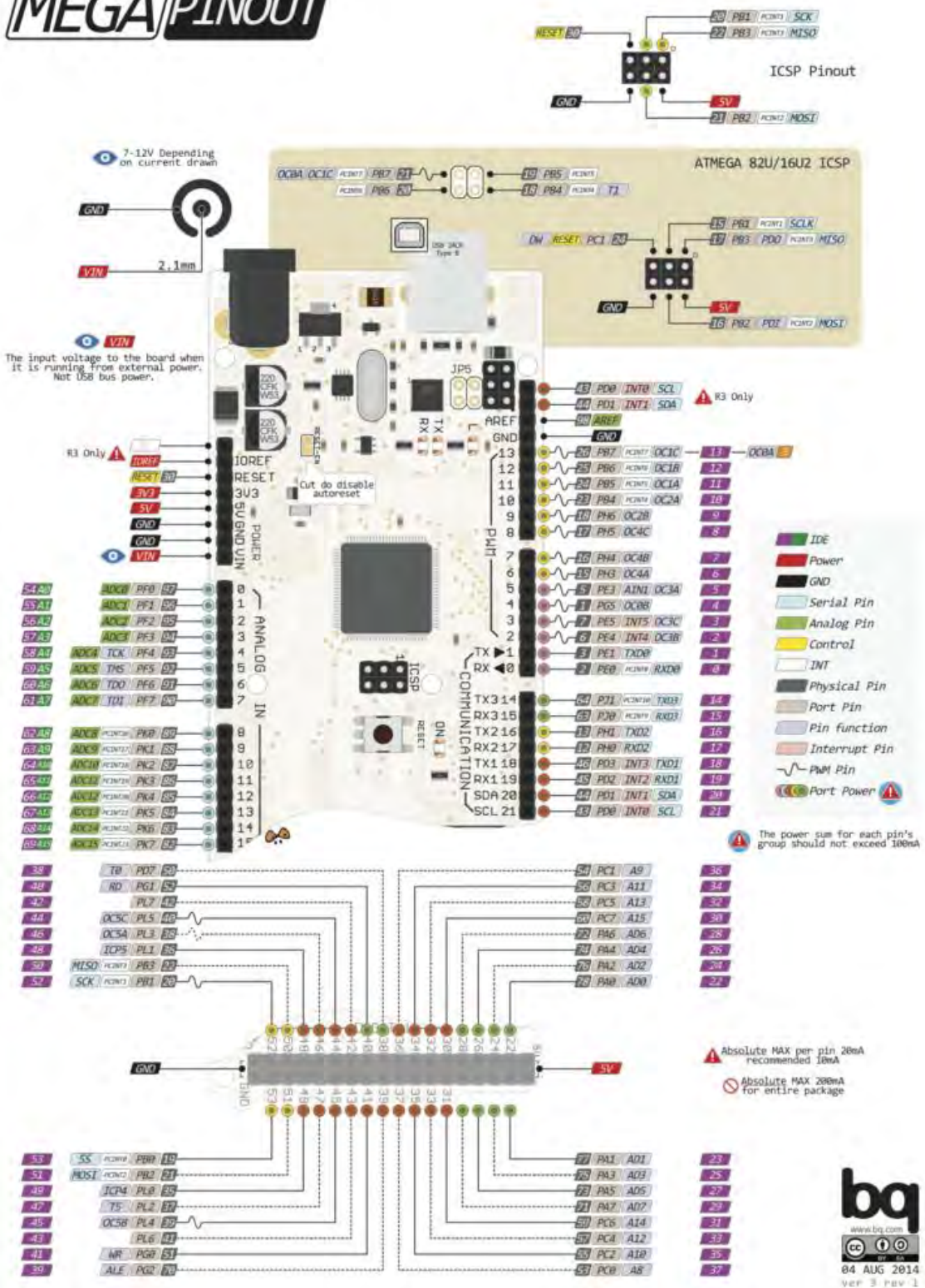
Imatge 14: Arduino Mega 2560

Característiques electròniques de l'Arduino Mega 2560:

- Microcontrolador ATmega2560.
- Voltatge d'entrada de - 7-12V.

- 54 pins digitals d'entrada / Sortida (14 d'ells són sortides PWM).
- 16 entrades anàlogues.
- 256 k de memòria flash.
- Velocitat de el rellotge de 16MHz.
- Dimensions: 102mm x 53mm

MEGA PINOUT

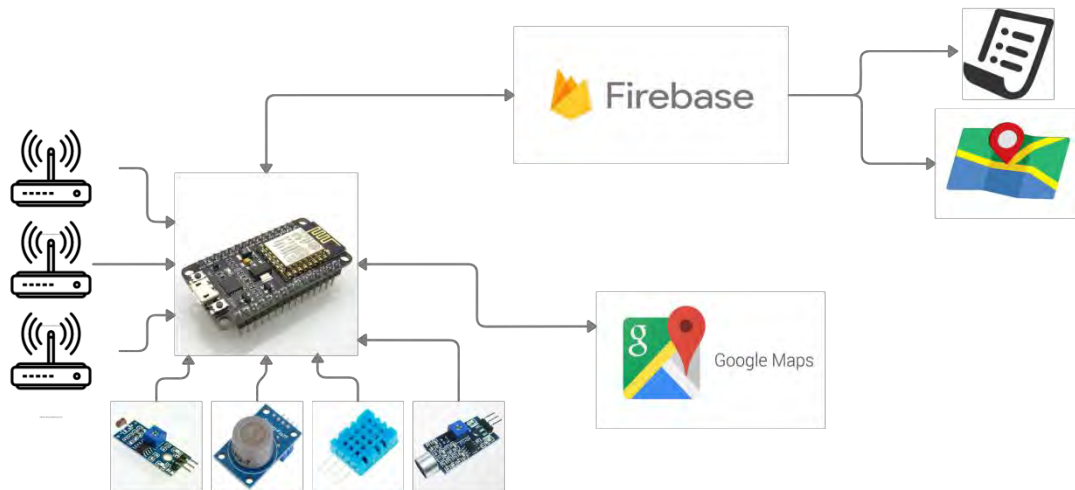


Imatge 15: Pinout Arduino Mega 2560

5 Muntatge del projecte

5.1 Esquema del projecte

Per poder entendre correctament el projecte, a la següent imatge il·lustrem quina línia seguirà i com treballarem amb les dades que obtindrem.

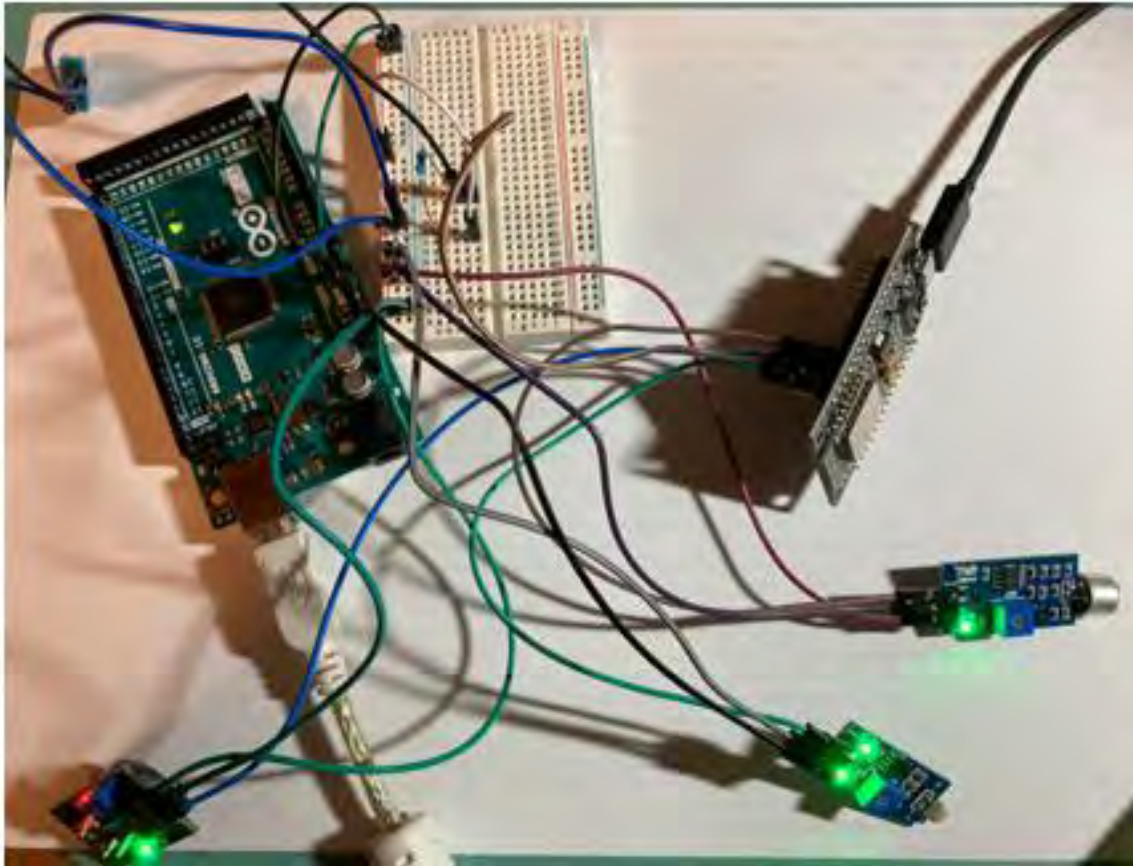


Imatge 16: Esquema del projecte

Com podem observar, el centre del nostre projecte es el NodeMCU, el qual s'encarregarà de gestionar la informació rebuda per tots els sensors (sensor LDR, MQ, DHT11 i CN07), connectar-se mitjançant connexió WiFi a l'Api de Google Maps per poder rebre les coordenades geogràfiques i enviar totes les dades esmentades a la nostre base de dades (Firebase).

A continuació, Firebase emmagatzemarà totes les dades enviades pel NodeMCU i a partir dels nostres arxius mapa i sensors, podrem mostrar el recorregut que ha realitzat el vianant o el ciclista i les dades dels sensors en temps real en el punt on es troba el NodeMCU amb els sensors.

5.2 Interconnexió del hardware



Imatge 17: Interconnexió dels components del projecte

En la imatge anterior, tenim el muntatge complet de la part de hardware del nostre projecte. A continuació comentarem el pinout dels components utilitzats:

NodeMCU:

- D1: Data DHT11
- D2: D0 LDR
- D3: D0 MQ-7
- D4: OUT Noise

Arduino Mega 2560:

- 5V: Protoboard +
- GND: Protoboard –

LDR:

- VCC: Protoboard +
- GND: Protoboard –
- D0: D2 NodeMCU
- A0: NC

DHT11:

- VCC: Protoboard +
- GND: Protoboard –
- D0: D1 NodeMCU
- A0: NC

CN07:

- VCC: Protoboard +
- GND: Protoboard –
- OUT: D4 NodeMCU

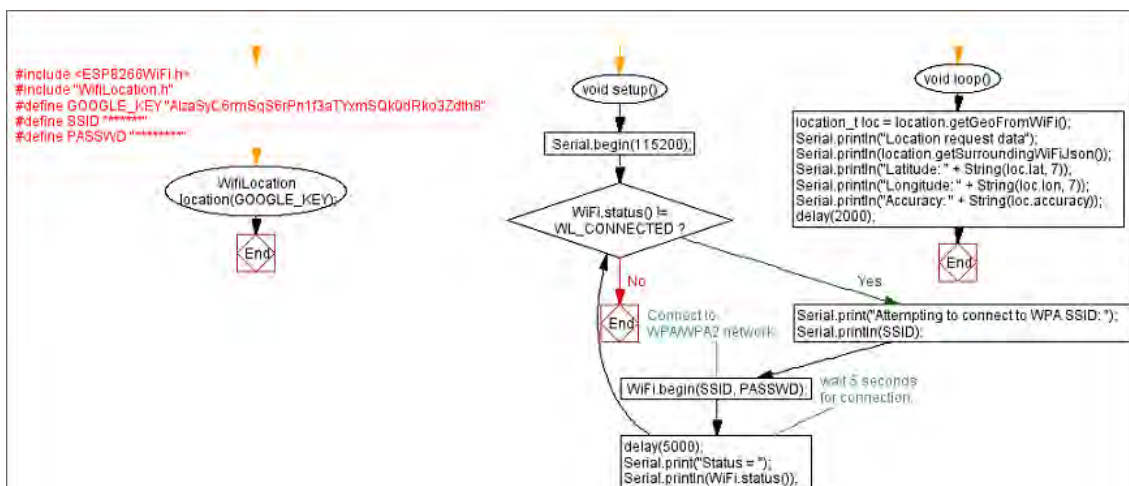
MQ-7:

- VCC: Protoboard +
- GND: Protoboard –
- D0: D3 NodeMCU
- A0: NC

5.3 Programació amb Arduino (components)

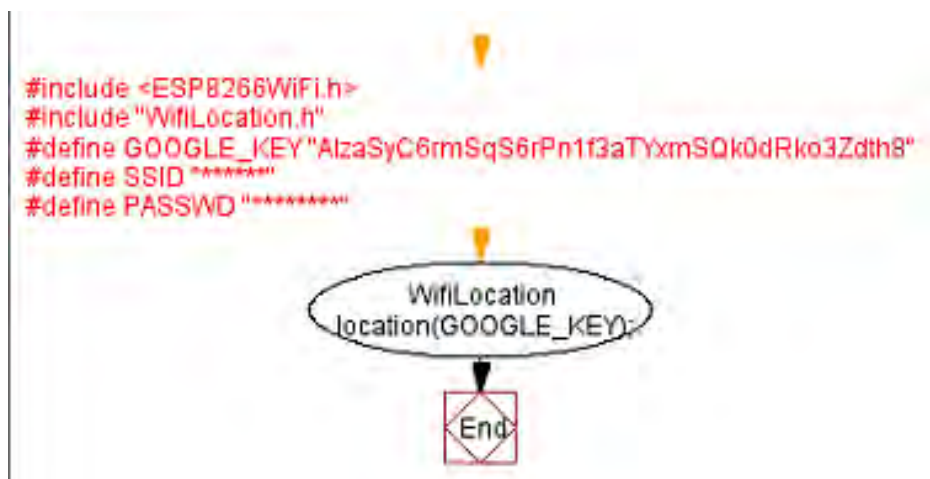
A continuació, tractarem cadascun dels diferents components amb els que treballarem en el nostre projecte utilitzant diferents diagrames de flux.

5.3.1 Diagrama GPS i NodeMCU



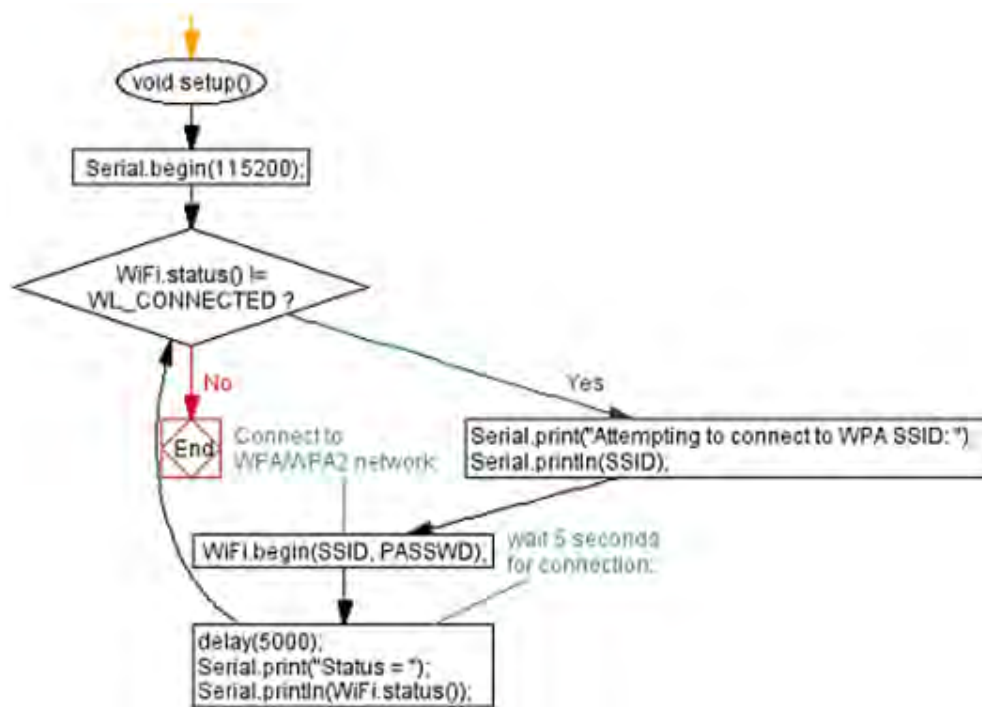
Imatge 18: Codi d'obtenció de coordenades

En aquest diagrama podem observar l'obtenció de les coordenades geogràfiques (tan la latitud, com la longitud) a través de la connexió WiFi del nostre sistema. A continuació, explicarem cadascuna de les parts del diagrama.



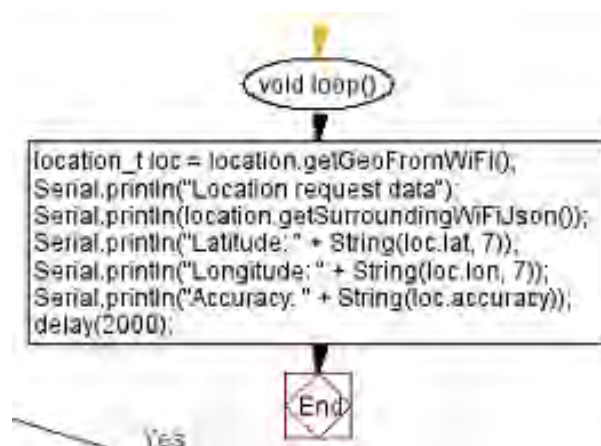
Imatge 19: Declaració de llibreries i constants

Aquí podem observar la configuració de llibreries que utilitzarem per l'obtenció de coordenades. Declarem les llibreries "ESP8266" encarregat de les coordenades i la llibreria "WifiLocation" encarregada d'establir connexió WiFi amb el nostre NodeMCU.



Imatge 20: Establiment de connexió

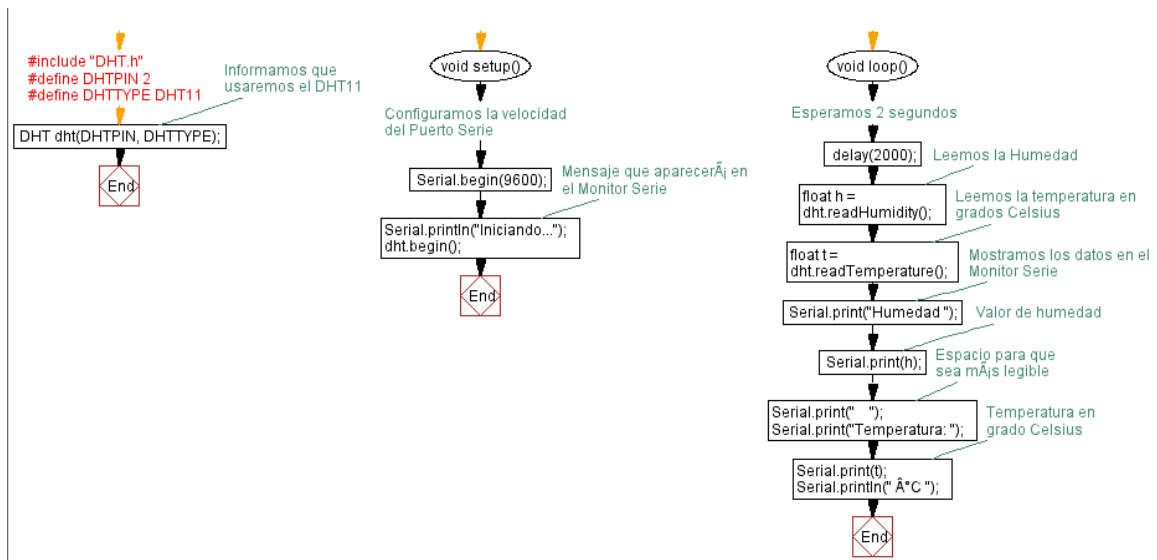
En aquest tros de codi farem provarem la connexió a Internet, mitjançant WiFi per posteriorment extraure les diferents coordenades.



Imatge 21: Crida per obtenir les coordenades

En aquest últim fragment, després d'haver establert una connexió correcta, farem una crida per que ens retornin la latitud, la longitud i la precisió amb la que ens retornen els 2 valors esmentats anteriorment.

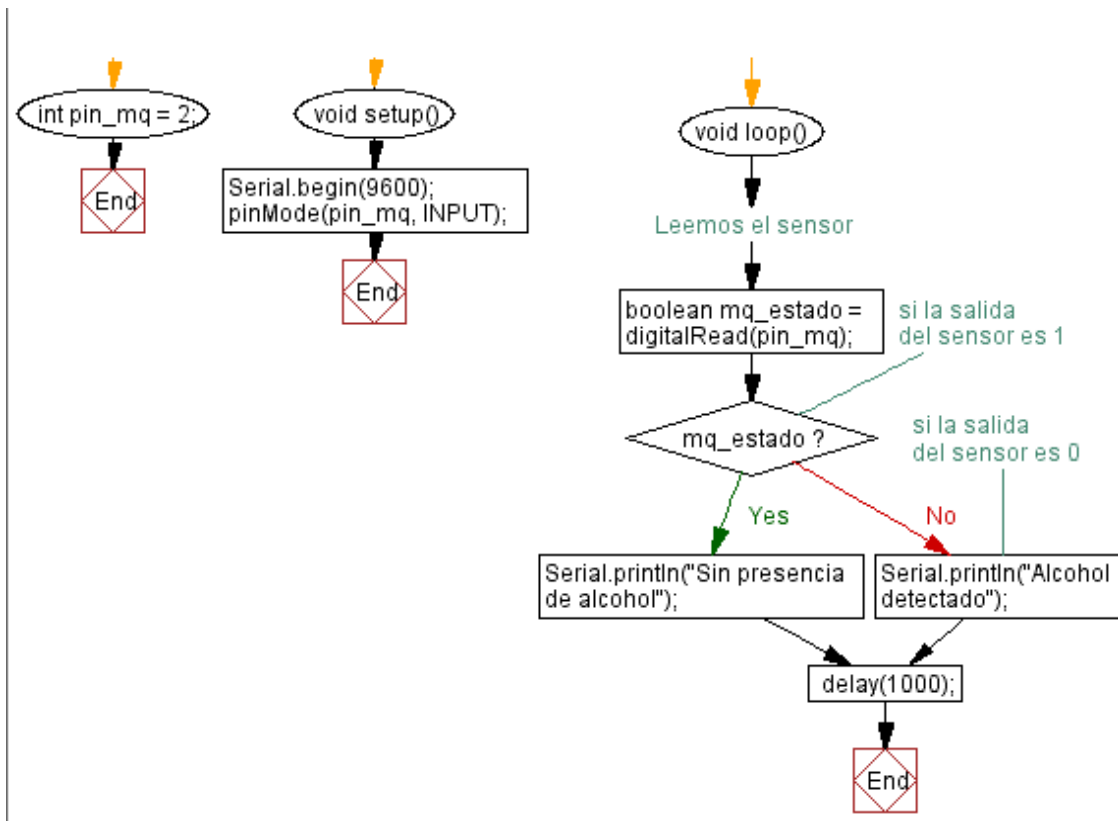
5.3.2 Diagrama Sensor DHT11



Imatge 22: Codi Sensor DHT11

En aquest codi, observem com tractarem les dades del sensor DHT11. Tindrem un pin d'entrada "DHTPIN" el qual el tractarem com una entrada analògica. El que voldrem serà el valor que ens torni d'humitat i de temperatura cada cert temps.

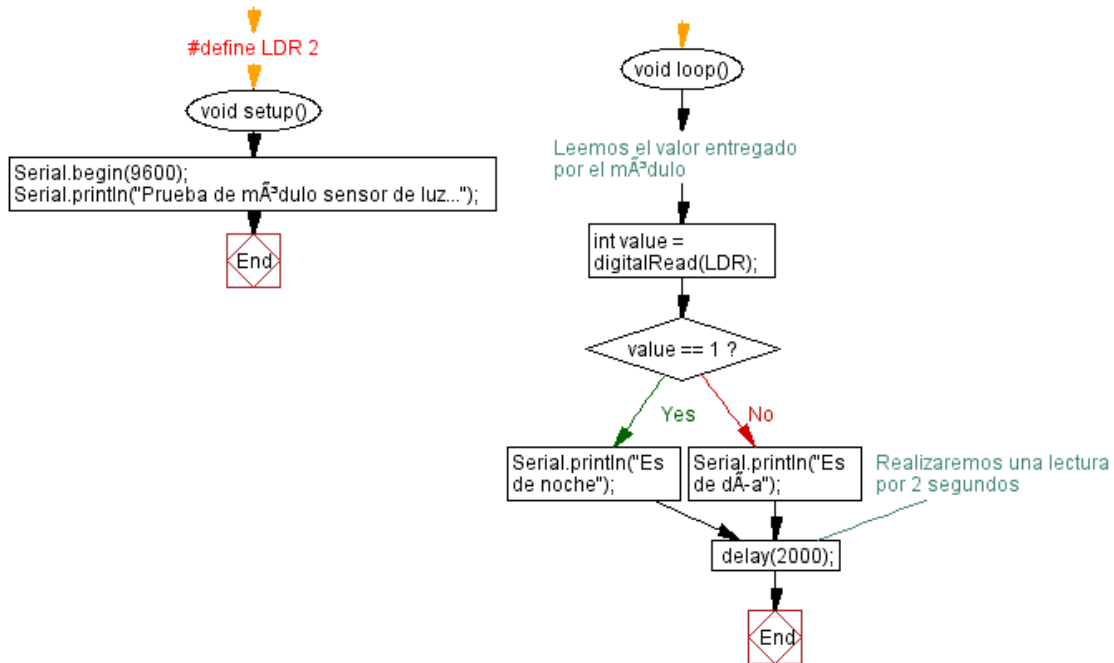
5.3.3 Diagrama Sensor de gas MQ



Imatge 23: Diagrama Sensor MQ-7

En aquest diagrama, observem com tractarem les dades del sensor de gas CO₂. Tindrem un pin d'entrada "pin_mq" el qual el tractarem com una entrada digital, és a dir, només ens interessarà saber si la variable booleana "mq_estado" és "TRUE" o "FALSA" (si hi ha presència d'alcohol o no).

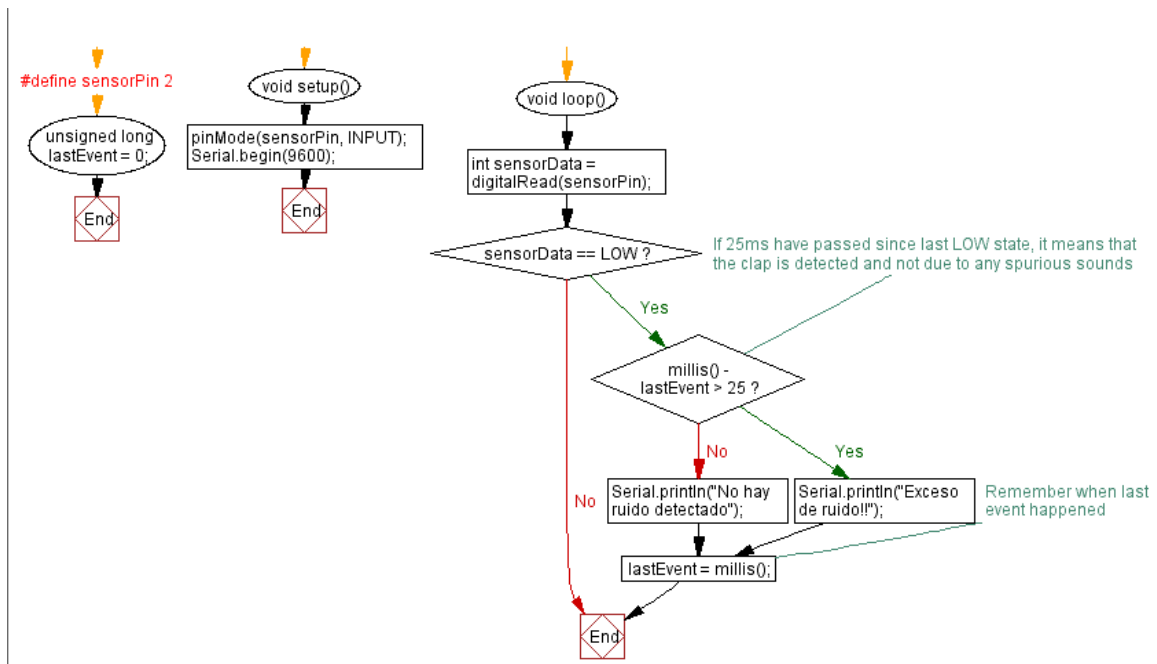
5.3.4 Diagrama Sensor de llum LDR



Imatge 24: Diagrama Sensor LDR

En aquest diagrama observem com tractarem les dades del sensor de llum. Tindrem un pin d'entrada "LDR" el qual el tractarem com una entrada digital, és a dir, només ens interessarà saber si val '1' o '0' (si passa un cert llindar o no).

5.3.5 Diagrama Sensor de soroll CN07



Imatge 25: Diagrama Sensor de CN07

En aquest diagrama observem com tractarem les dades del sensor de soroll. Tindrem un pin d'entrada "sensorPin" el qual el tractarem com una entrada digital, és a dir, només ens interessarà saber si val '1' o '0' (si passa un cert llindar o no). En cas que el valor de la variable "sensorData" sigui '0', començarà un contador per determinar quants sorolls capta el sensor, és a dir, si el sensor queda en '0' menys de 25ms només ens sortirà l'avís una vegada i no tantes vegades com marqui la velocitat del programa.

5.4 Base de dades

Una vegada hem fet l'obtenció de dades per part dels mòduls i dels diferents sensors que hem comentat en apartats anteriors, ara ens toca emmagatzemar totes les dades en una base de dades.

Una base de dades és un conjunt de dades segons una estructura coherent i accessibles des d'un o més programes o aplicacions, de manera que qualsevol d'aquestes dades pot ésser extreta del conjunt i actualitzada, sense afectar ni l'estructura del conjunt ni les altres dades. Ara, ens toca tornar a fer una nova recerca dels diferents tipus de base de dades que existeixen actualment i quina ens anirà millor en el nostre projecte. Les bases de dades que hem trobat són les següents:

5.4.1 MariaDB

Aquesta solució, creada pels desenvolupadors originals de MySQL, és emprada per gegants tecnològics com Wikipedia, Facebook i fins i tot Google. MariaDB és un servidor de base de dades que ofereix funcionalitat de substitució directa per a MySQL. La seguretat és prioritària per als desenvolupadors de MariaDB, que inclouen en cada versió els pegats de seguretat de MySQL i fins i tot els milloren si cal.

Avantatges:

- Alta escalabilitat i fàcil integració.
- Accés en temps Real.
- Funcionalitat central de MySQL (MariaDB és una alternativa a MySQL).
- Motors d'emmagatzematge alternatius, optimitzacions de servidor i pegats.

Desavantatges:

- No té connector de complexitat de contrasenya.
- Sense interfície en memòria cau de memòria.

- Sense rastrejador d'optimització.

5.4.2 MySQL

MySQL porta al mercat des de 1995 i ara és propietat d'Oracle. Encara que és de codi obert, disposa de diverses edicions de pagament que ofereixen funcions addicionals, com georeplicació de clústers i escalat automàtic. Ja que MySQL, escrit tant en C com en C ++, és un estàndard de el sector, és compatible amb pràcticament qualsevol sistema operatiu. Aquesta solució és ideal per a usuaris internacionals, ja que el servidor pot oferir als clients missatges d'error en múltiples idiomes i compta amb suport per a diferents conjunts de caràcters.

Avantatges:

- Verificació basada en host.
- Pot usar-se encara que no hi hagi cap xarxa disponible.
- Sistema flexible de privilegis i contrasenyes.
- Xifrat de seguretat de tot el trànsit de contrasenyes.
- Ofereix el servidor per separat per a entorns amb xarxes de client / servidor.

Desavantatges:

- Els usuaris senten que MySQL ja no és gratuïta i de codi obert.
- Ja no és una solució impulsada per la comunitat.
- Els usuaris no poden solucionar errors ni crear "parches".
- Superada per altres solucions que s'actualitzen amb més freqüència.

5.4.3 MongoDB

MongoDB, fundada el 2007, es coneix com "la base de dades de les idees gegants". Desenvolupada per l'equip que va crear DoubleClick, ShopWiki i Gilt Groupe, MongoDB compta amb el suport de coneguts inversors com Fidelity Investments, The Goldman Sachs Group, Inc i Intel Capital. Des de la seva creació, MongoDB s'ha descarregat 20 milions d'vegades i compta amb més de 1 000 socis dedicats a aquesta solució gratuïta i de codi obert que advoquen per bases de dades i programacions senzilles i naturals.

Avantatges:

- Validació de documents.
- Motor d'emmagatzematge xifrat.
- Utilitzada en aplicacions mòbils, catàlegs de producte i gestió de contingut.
- Aplicacions en temps real.
- Temps reduït entre fallada primari i recuperació.

Desavantatges:

- No és adequada per a aplicacions que necessitin de transaccions complexes.
- No és adequada per substitució directa d'aplicacions heretades.
- Solució jove: el programari canvia i evoluciona ràpidament.

5.4.4 SQLite

SQLite, autoproclamada com "la base de dades més implementada de el món", es va crear el 2000 i ha comptat entre els seus usuaris amb importants empreses com Apple, Facebook, Microsoft i Google. Cada versió es prova minuciosament per garantir la seva fiabilitat. En cas d'errors, els desenvolupadors de SQLite indiquen honestament les potencials desavantatges, oferint en cada versió llistes d'errors i cronologies de canvis en el codi.

Avantatges:

- Sense processos de servidor per separat.
- Format d'arxius multiplataforma.
- Biblioteca compacta: s'executa més ràpid fins i tot amb més memòria.

Desavantatges

- No es recomana pe aplicacions client / servidor, webs amb alt volum, grans conjunts de dades i per alta concurrència.

5.4.5 Firebase

Firebase és una plataforma per al desenvolupament d'aplicacions web i aplicacions mòbils desenvolupada per Google en 2014. És una plataforma situada en el núvol, integrada amb Google Cloud Platform, que utilitza un conjunt d'eines per a la creació i sincronització de projectes que seran dotats d'alta qualitat, fent possible el creixement de el nombre d'usuaris i donant resultat també a l'obtenció de una major monetització.

Avantatges:

- Molt recomanable per a aplicacions que necessitin compartir dades en temps real.
- Les seves funcionalitats, a més de ser variades, es complementen molt bé i es poden gestionar de forma senzilla des d'un únic panell. A més, no cal fer servir

totes aquestes opcions per a l'aplicació, podent triar només aquelles que més ens interessin.

- Facilita l'enviament de notifikacions: són molt senzilles d'implementar i gestionar, a més de ser extremadament útils per mantenir l'atenció dels usuaris.
- Engloba Analytics: especialitzat en determinades mètriques d'aplicacions mòbils i integrat en el panell central de Firebase amb un funcionament molt intuïtiu. Essencial per prendre decisions en diferents fases del procés.
- Suport gratuït via email, independentment de si el desenvolupador utilitza la versió gratuïta o de pagament.
- Ofereix seguretat a l'usuari: amb els certificats SSL.
- Permet als desenvolupadors restar-li atenció a l'backend i a les infraestructures complexes per centrar-se completament en altres aspectes.

Desavantatges:

- El desavantatge més comunament esmentat és el preu. Té un pla gratuït, no obstant això, té limitacions (principalment de nombre d'usuaris simultanis i d'espai d'emmagatzematge), pel que pot resultar necessari contractar una versió de pagament. Com podem veure, podem trobar Firebase gratis, però si necessitem treballar amb això de manera professional, haurem d'acceptar de contractar algun tipus de pagament.

5.4.6 Base de dades escollida

Finalment, després d'haver realitzat aquesta tasca de recerca d'informació i observant tots els pros i contres de cadascuna de les bases de dades tindrem en compte 3 aspectes principalment:

- En aquest projecte no tenim un volum massa gran de dades a emmagatzemar, per tant totes 5 seran igual de vàlides per treballar amb elles, ja que totes suportaran la quantitat de dades a emmagatzemar.

- Hi ha algunes bases de dades que són gratuïtes i d'altres que no. Així doncs, en aquest projecte escollirem una que sigui gratuïta o al menys que en la forma d'utilitzar-la nosaltres ho sigui.
- Buscarem una base de dades fàcil de programar i treballar amb ella. No volem centrar molt el projecte en la base de dades, ja que entenem que no es la principal font d'interès.
- Finalment, necessitem que la base de dades sigui fàcil de comunicar-se amb el nostre sistema Arduino, ja que com hem dit abans no volem que aquesta part ens porti massa mals de cap.

Per escollir la nostra base de dades ens hem basat en els punts que hem comentat anteriorment. Si les condicions de projecte fossin unes altres, la nostra elecció dependria d'uns altres paràmetres.

Així doncs, després d'analitzar cadascuna de les possibilitats anteriors mitjançant els diferents punts que acabem de comentar, triarem Firebase. Firebase és una base de dades molt senzilla de treballar en projectes d'aquests tipus, ja que hem treballat alguna vegada amb ella en el passat. Per projectes d'aquest caire amb la versió gratuïta tenim suficient i les dades que hem d'emmagatzemar en ella no són un volum massa gran.

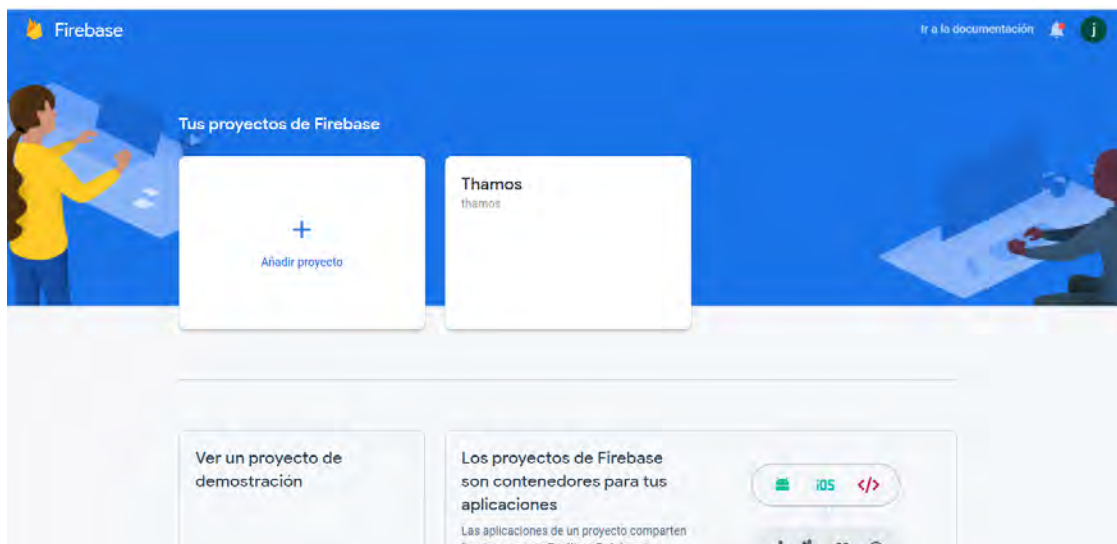
5.5 Configuració de la base de dades Firebase

A continuació, explicarem detalladament la configuració de la base de dades “FireBase”. Per accedir al servei Firebase heu de tenir un compte de Google. Accedim a la pàgina d’inici de Firebase i punxem en IR A LA CONSOLA.



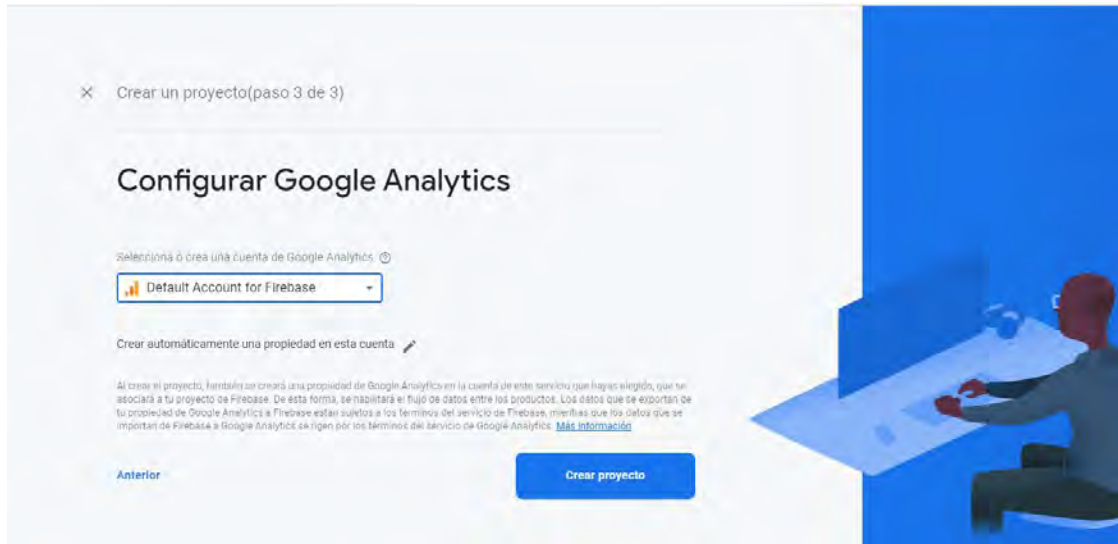
Imatge 26: Accedir a la consola

Això obrirà el tauler de control de Firebase. Un cop hem entrat al tauler de control crearem un nou projecte així que fes click a AÑADIR PROYECTO.



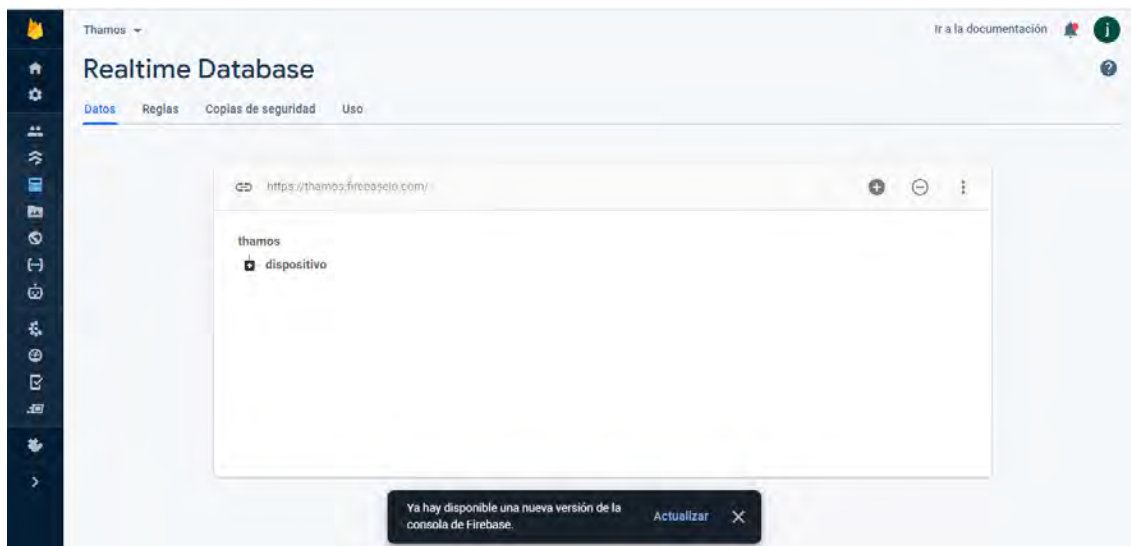
Imatge 27: Afegir un nou projecte

Això obre una nova pantalla perquè omplis les dades del projecte. En el meu cas he posat el nom Thamos, he clickat “Habilitar Google Analytics en este proyecto” i seleccionem “Default Account for FireBase”. Un cop seleccionat fes click a CREAR PROYECTO.



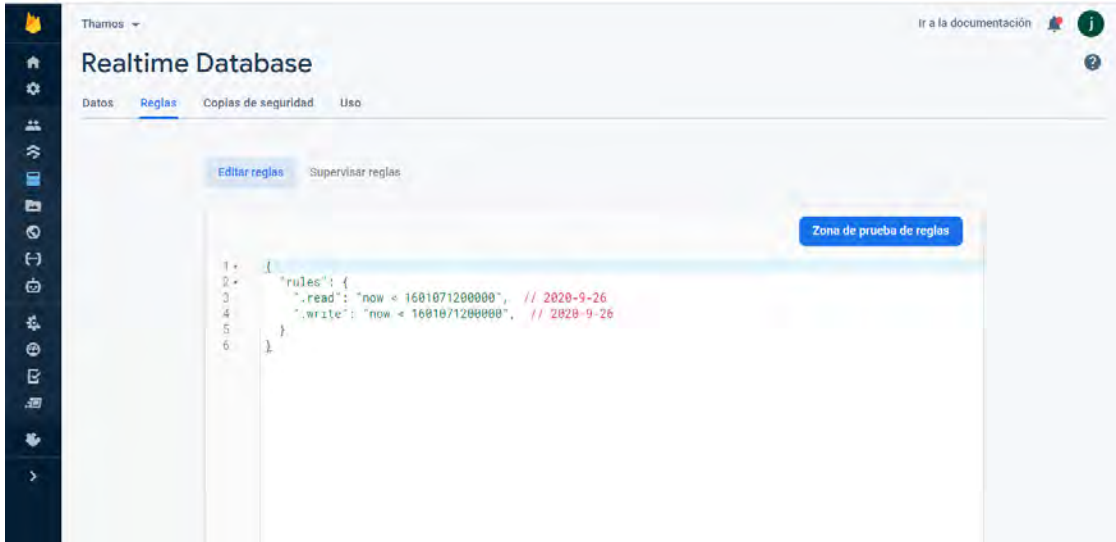
Imatge 28: Configuració del nou projecte

Un cop creat el projecte s'obre el panell de control. Aquí podràs accedir a les diferents apartats de configuració i podràs veure l'identificador únic del teu projecte (ID). Anem a configurar la base de dades, fes click en “Realtime Database”.



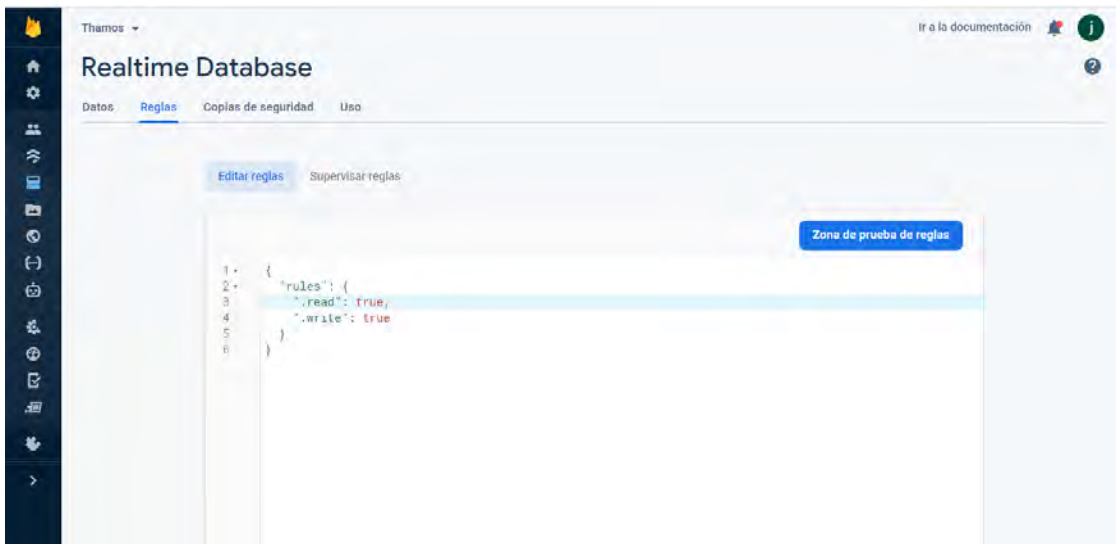
Imatge 29: Base de dades “thamos”

Per defecte, la base de dades té assignades unes regles d'accés que només permet connectar-se a usuaris autenticats a Google. Anem a modificar aquestes regles perquè pugui accedir tothom sense restriccions. Obre la pestanya de regles en el tauler de control de "Realtime Database".



Imatge 30: Regles d'accés per defecte

Hem de posar els dos valors read i write a true (verdader). No t'oblidis a fer clic en PUBLICAR.

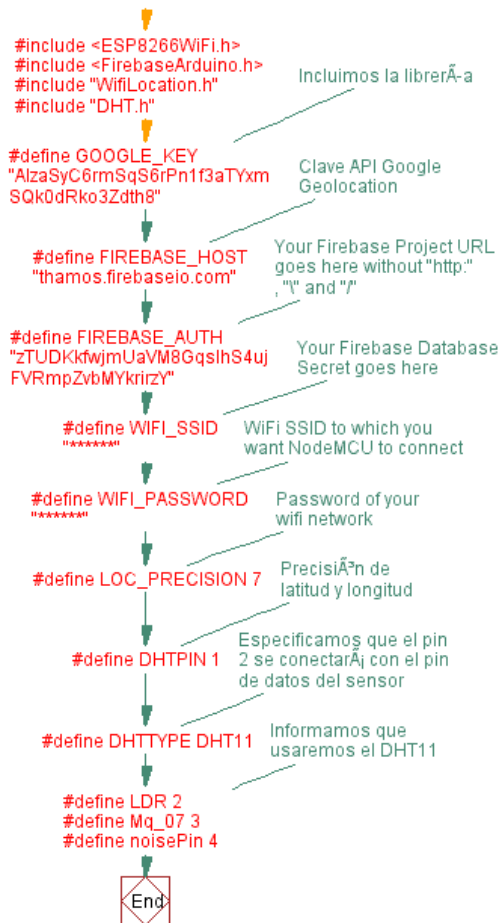


Imatge 31: Regles d'accés modificades correctament

I amb això ja tindriem configurat Firebase per utilitzar-lo amb NodeMCU.

5.6 Programació amb Arduino (Connexió Hardware-Firebase)

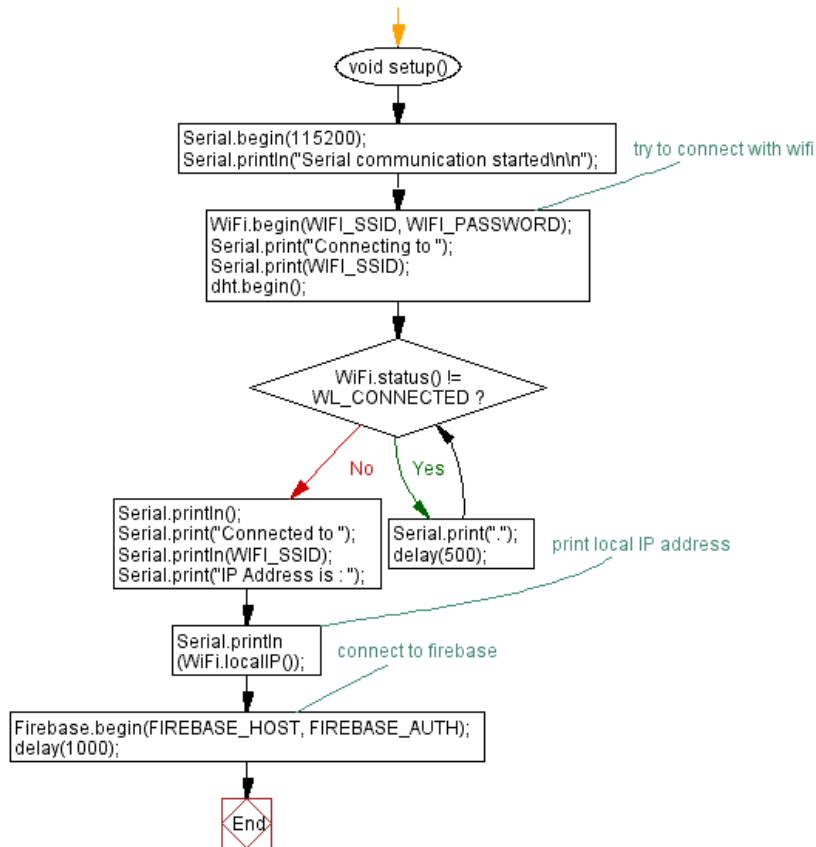
A continuació, analitzarem, mitjançant un diagrama de flux, la connexió d'Arduino amb la base de dades Firebase:



Imatge 32: Definim les llibreries i les constants

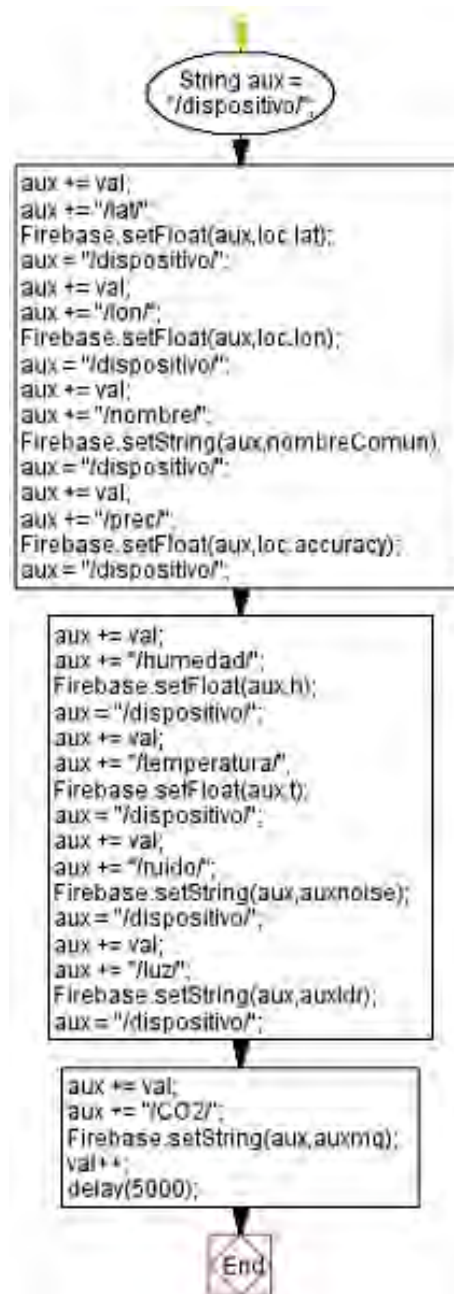
Primer de tot definim les llibreries necessàries per establir la connexió amb el servidor. Les que necessitarem seran la "ESP8266WiFi.h" per establir connexió WiFi mitjançant el NodeMCU i la "FirebaseArduino.h" per establir la connexió amb el servidor Firebase.

A més, podem observar la definició de constants del Host de Firebase, de la clau d'autenticació de Firebase, l'usuari Wifi que utilitzarem i la seva clau (òbviament aquestes dues últimes estan ocultes).



Imatge 33: Establiment de connexió Wifi i amb Firebase

Aquí observem, la forma d'establir connexió WiFi i amb el servidor. A la part superior de la imatge trobem la part de connexió WiFi, on necessitarem simplement l'usuari de la xarxa i la clau i amb la funció "WiFi.begin (SSID, PASSWORD)" establim la connexió. A la part inferior trobem la part de connexió amb Firebase, on necessitarem l'url del host de la base de dades i una autenticació i amb la funció "Firebase.begin (HOST, AUTH)" establim la connexió.



Imatge 34: Enviem dades al servidor Firebase

Finalment, en la imatge anterior podem observar la forma d'enviar informació a la base de dades. La forma es bastant simple, amb la comanda "Firebase.setTIPUS("name", <variable>)" podem enviar dades de forma individual cap a Firebase, on TIPUS correspon al tipus de dades que volem enviar (String, integer, float,...); "name" serà el nom que tindrà la variable a la base de dades; <variable> serà el valor que tindrà la variable que volem enviar a Firebase.

5.7 Base de dades Firebase

Firestore (i, majoritàriament, totes les bases de dades NoSQL), d'altra banda, no tenen aquest tipus de restriccions. La base de dades de Firestore és només un gran objecte JSON on pots emmagatzemar el que vulguis dins. A diferència de SQL, no hi ha esquema per a la base de dades, ni taules, ni columnes, només és una combinació de parells clau / valor.

Així doncs, la nostra taula de posicionament en Firestore es veuria així:

```
{
  "dispositivo": [
    {
      "CO2" : "No hay contaminación",
      "humedad" : 60,
      "lat" : 41.45432,
      "lon" : 2.223855,
      "luz" : "Es de día, no hace falta alumbrado",
      "nombre" : "NodeMCU",
      "prec" : 25,
      "ruido" : "Limite de ruido aceptables",
      "temperatura" : 25
    }
  ]
}
```

Imatge352: JSON de la nostra base de dades

El que podem observar a la imatge és l'estructura (JSON) en que la nostra base de dades guarda la informació.

Primer de tot tindrem la paraula clau "dispositivo" la qual marcarà que totes les dades que volem guardar en aquell moment serà del mateix dispositiu. Això és útil si, per exemple, ara volem desar dades en aquesta mateixa base de dades però d'un altre dispositiu, i així que no quedin barrejats.

Dins del mateix dispositiu, trobem les diferents paraules clau:

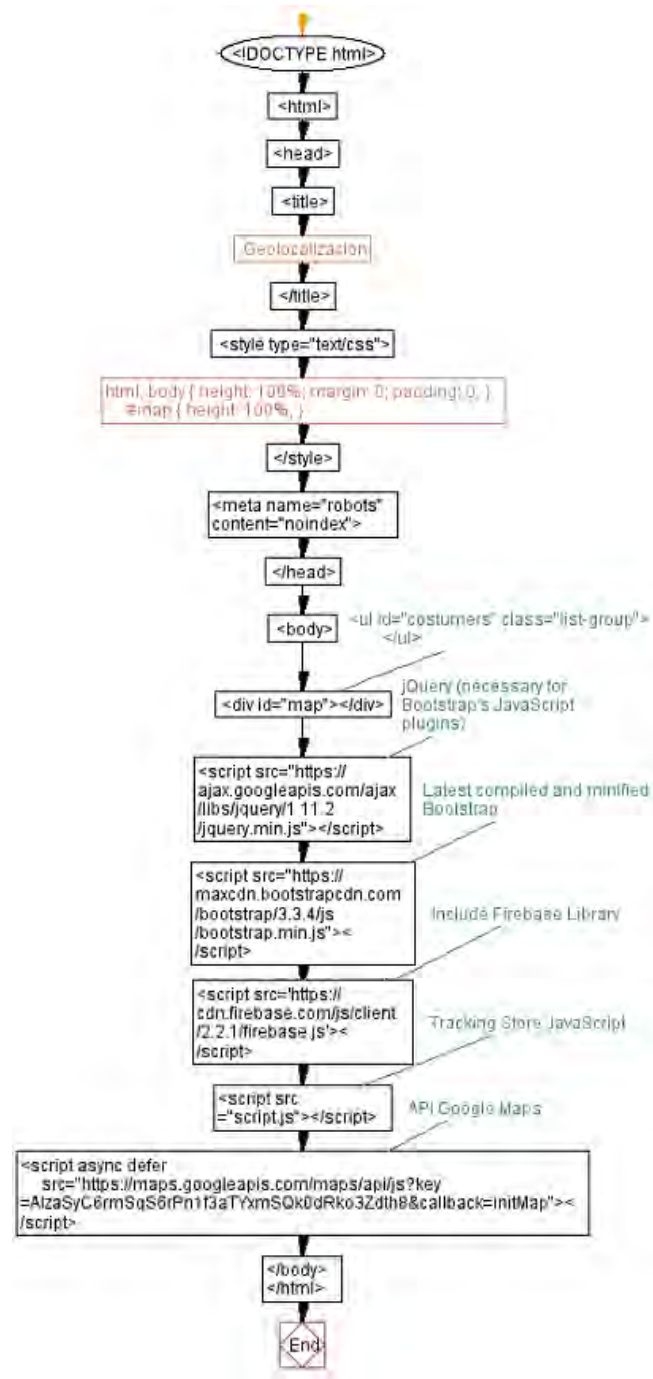
- "CO2": Guardarà un String que comentarà si el nivell de contaminació és acceptable o no.

- “humedad”: Guardarà un float que ens indicarà quin valor d’humitat hi ha en l’ambient.
- “lat”: Guardarà un float que ens indicarà quin valor de latitud té la coordenada on es troba l’individu.
- “lon”: Guardarà un float que ens indicarà quin valor de longitud té la coordenada on es troba l’individu.
- “luz”: Guardarà un String que comentarà si el nivell de lluminositat és acceptable o no.
- “nombre”: Guardarà un String que indicarà quin és el dispositiu que està enviant les dades.
- “prec”: Guardarà un float que ens indicarà quin valor de precisió té les coordenades on es troba l’individu.
- “ruido”: Guardarà un String que comentarà si el nivell de soroll és acceptable o no.
- “temperatura”: Guardarà un float que ens indicarà quin valor d’humitat hi ha en l’ambient.

5.8 Visualització

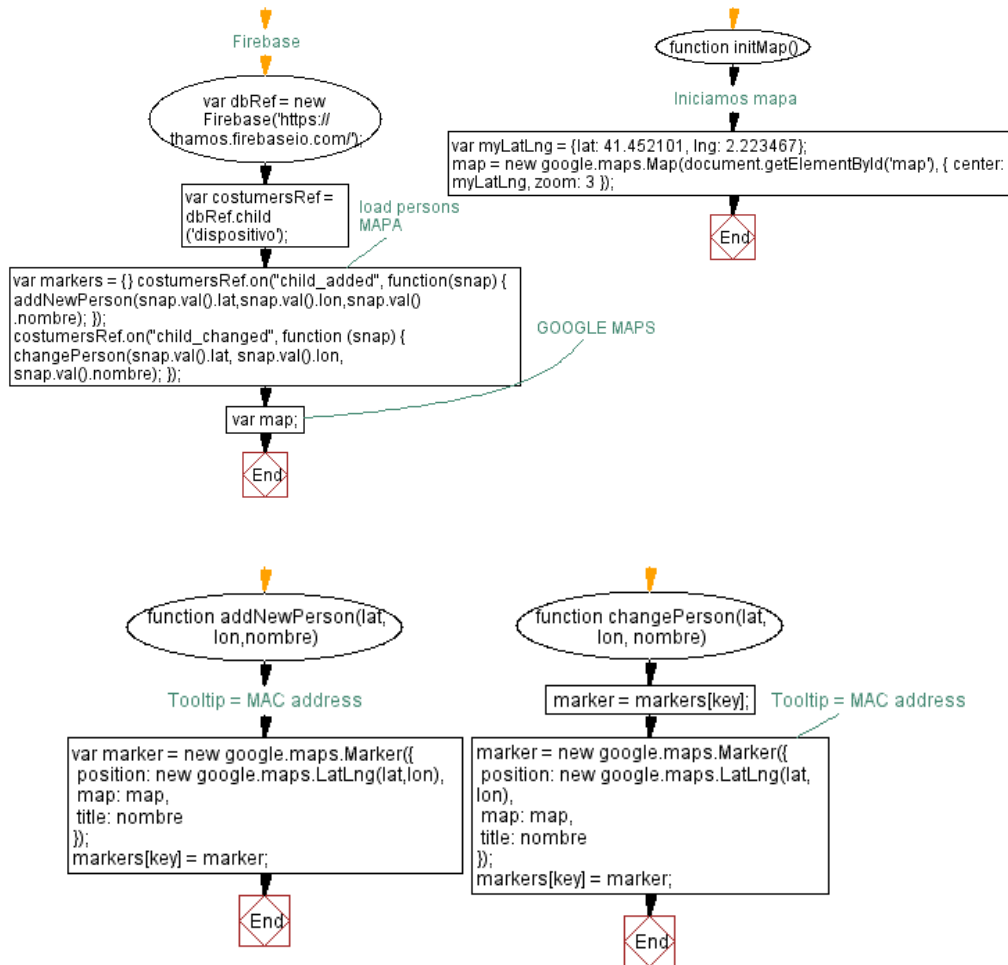
5.8.1 Mapa amb recorregut

Ara ja ha arribat el moment de visualitzar el recorregut en Google Maps. A continuació mostrarem un diagrama que representa el codi HTML per poder representar el mapa:



Imatge 36: Diagrama del codi HTML per representar el mapa

El codi HTML anterior no serveix de res si no creem un arxiu .js on estigui la programació per mostrar les dades al mapa. Aquest fitxer es dirà script.js i ha d'estar en la mateixa carpeta que l'arxiu mapa.html anterior.

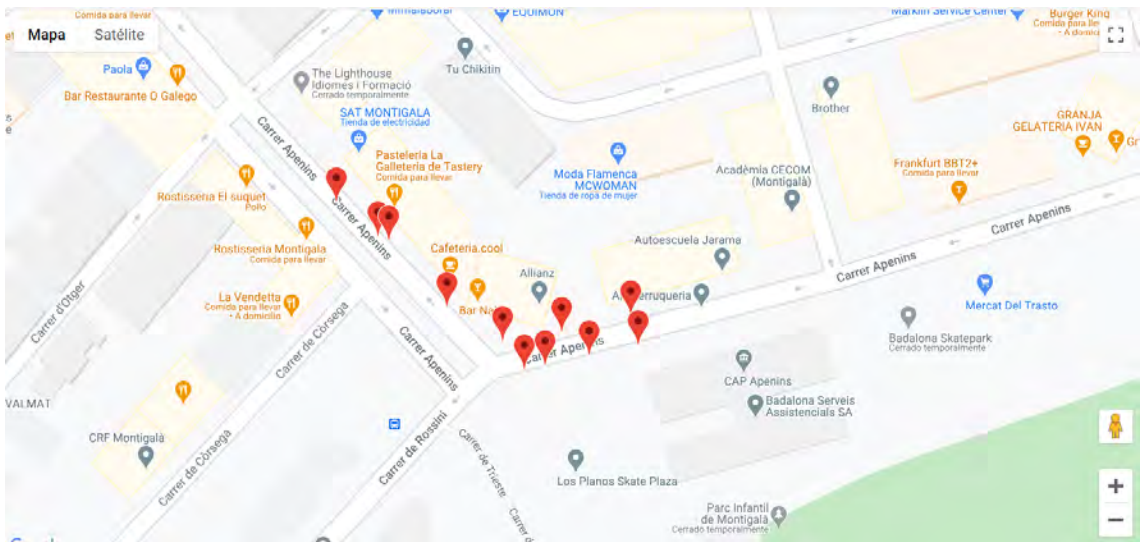


Imatge 37: Diagrama del Script que treballarà amb Firebase

Finalment, obrim l'arxiu mapa.html i ens carrega el mapa amb el trajecte que hem realitzat com podem observar en les següents imatges.



Imatge 38: Mapa que marca el recorregut

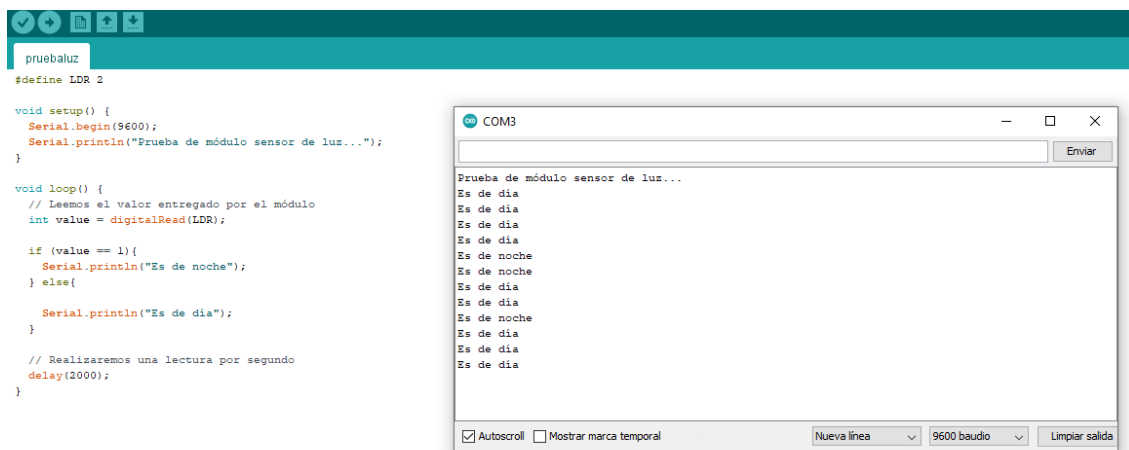


Imatge 39: Mapa que marca el recorregut (2)

6 Proves del projecte

En aquest apartat realitzarem una sèrie de proves de les diferents parts del projecte, per demostrar així el seu correcte funcionament. Les proves dels diferents sensors i del sistema de geolocalització les realitzarem de forma independent, d'aquesta manera podrem analitzar correctament el seu funcionament. Finalment tindrem una prova amb tots els components mostrant el correcte funcionament del projecte.

6.1 Prova del sensor LDR



```
pruebaluz
#define LDR 2

void setup() {
  Serial.begin(9600);
  Serial.println("Prueba de módulo sensor de luz...");
}

void loop() {
  // Leemos el valor entregado por el módulo
  int value = digitalRead(LDR);

  if (value == 1){
    Serial.println("Es de noche");
  } else{
    Serial.println("Es de día");
  }

  // Realizaremos una lectura por segundo
  delay(2000);
}
```

COM3

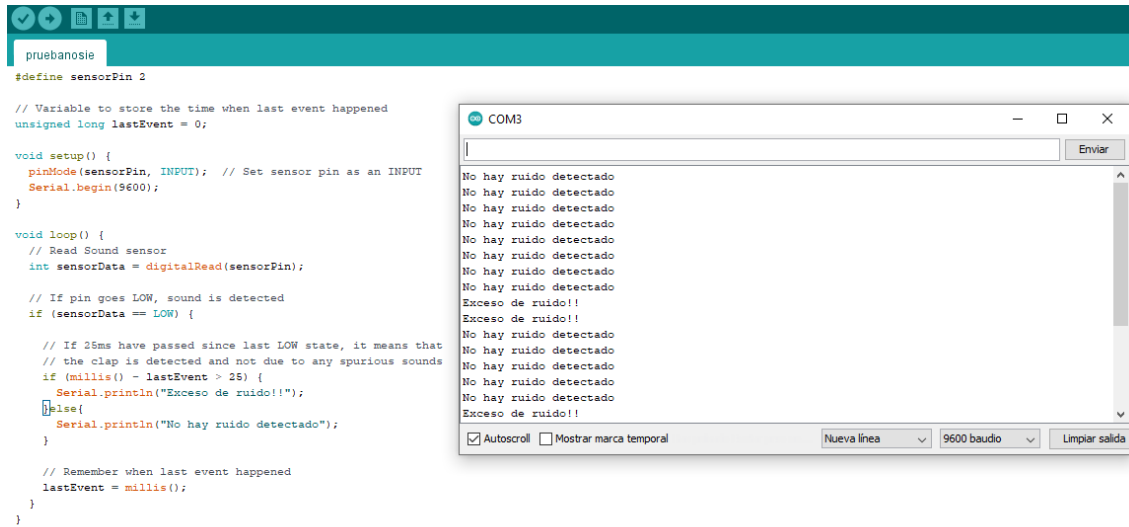
Prueba de módulo sensor de luz...
Es de día
Es de día
Es de día
Es de día
Es de noche
Es de noche
Es de día
Es de día
Es de noche
Es de día
Es de día
Es de día
Es de día

Autoscroll Mostrar marca temporal Nueva línea 9600 baudio Limpiar salida

Imatge 40: Prova de funcionament del sensor LDR

En aquest cas, la prova que hem realitzat ha estat provar el sensor al carrer mentre era de dia (hi havia llum). El que podem veure es com surt cada 2 segons que és de dia, ja que teníem el sensor sense tapar i per tant rebia llum, excepte en un parell de moments que hem tapat el sensor (simulant que es feia fosc) que ens indicava que era de nit.

6.2 Prova del sensor CN07



```
pruebanosis
#define sensorPin 2

// Variable to store the time when last event happened
unsigned long lastEvent = 0;

void setup() {
  pinMode(sensorPin, INPUT); // Set sensor pin as an INPUT
  Serial.begin(9600);
}

void loop() {
  // Read Sound sensor
  int sensorData = digitalRead(sensorPin);

  // If pin goes LOW, sound is detected
  if (sensorData == LOW) {

    // If 25ms have passed since last LOW state, it means that
    // the clap is detected and not due to any spurious sounds
    if (millis() - lastEvent > 25) {
      Serial.println("Exceso de ruido!!");
    } else {
      Serial.println("No hay ruido detectado");
    }

    // Remember when last event happened
    lastEvent = millis();
  }
}
```

COM3

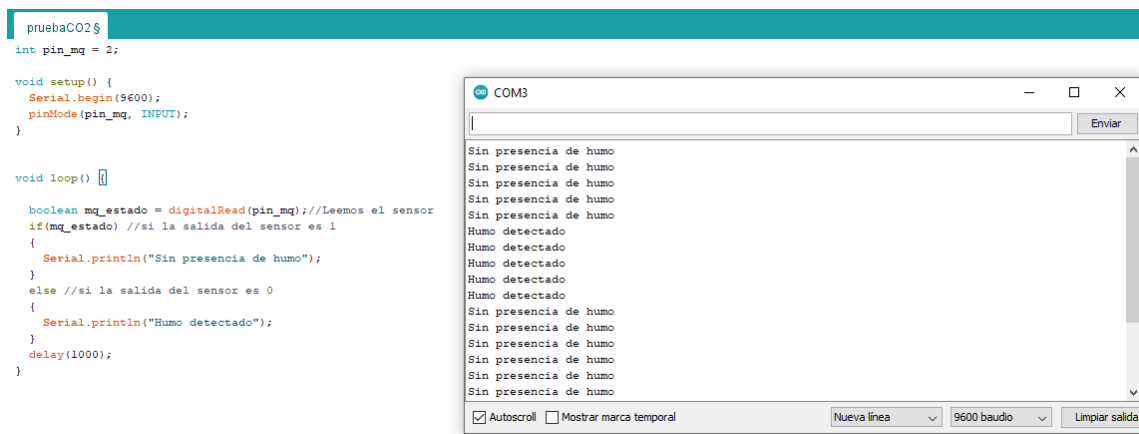
No hay ruido detectado
No hay ruido detectado
No hay ruido detectado
No hay ruido detectado
No hay ruido detectado
No hay ruido detectado
No hay ruido detectado
No hay ruido detectado
No hay ruido detectado
Exceso de ruido!!
Exceso de ruido!!
No hay ruido detectado
No hay ruido detectado
No hay ruido detectado
No hay ruido detectado
No hay ruido detectado
Exceso de ruido!!

Autoscroll Mostrar marca temporal Nueva línea 9600 baudio Limpiar salida

Imatge 41: Prova de funcionament del sensor CN07

En aquest cas, la prova que hem realitzat ha estat provar el sensor de soroll al carrer. Com en aquell moment no hi havia cap soroll que pogués superar el límit fixat, l'hem simulat nosaltres mateixos. El que podem veure es com surt que no hi ha cap tipus de soroll detectat mentre no capta cap soroll, però en el moment que superem el límit fixat, ens detecta que s'ha produït un soroll, tal i com queda indicat.

6.3 Prova del sensor MQ-7



```
pruebaCO2$
int pin_mq = 2;

void setup() {
  Serial.begin(9600);
  pinMode(pin_mq, INPUT);
}

void loop() {
  boolean mq_estado = digitalRead(pin_mq); //Leemos el sensor
  if(mq_estado) //si la salida del sensor es 1
  {
    Serial.println("Sin presencia de humo");
  }
  else //si la salida del sensor es 0
  {
    Serial.println("Humo detectado");
  }
  delay(1000);
}
```

COM3

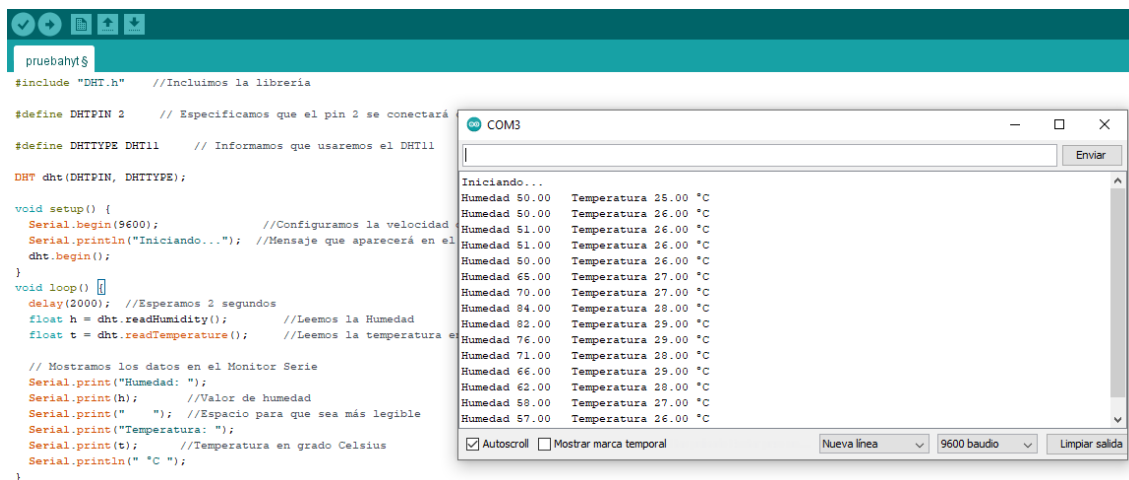
Sin presencia de humo
Sin presencia de humo
Sin presencia de humo
Sin presencia de humo
Sin presencia de humo
Humo detectado
Humo detectado
Humo detectado
Humo detectado
Humo detectado
Sin presencia de humo
Sin presencia de humo
Sin presencia de humo
Sin presencia de humo
Sin presencia de humo
Sin presencia de humo

Autoscroll Mostrar marca temporal Nueva línea 9600 baudio Limpiar salida

Imatge 42: Prova de funcionament del sensor MQ-7

En aquest cas, la prova que hem realitzat ha estat provar el sensor de contaminació (sensor de gas MQ-7) al carrer. Per simular la alta concentració de gas, hem aprofitat una cigarreta en el moment que volíem que el sensor canviés d'estat. El que podem veure es com surt que no hi ha cap tipus de presència de fum mentre no apropem la cigarreta, però en el moment que l'apropem, ens detecta la presència de fum (contaminació), tal i com queda indicat.

6.4 Prova del sensor DHT11



```
#include "DHT.h" //Incluimos la libreria

#define DHTPIN 2 // Especificamos que el pin 2 se conectará

#define DHTTYPE DHT11 // Informamos que usaremos el DHT11

DHT dht(DHTPIN, DHTTYPE);

void setup() {
  Serial.begin(9600); //Configuramos la velocidad
  Serial.println("Iniciando..."); //Mensaje que aparecerá en el
  dht.begin();
}

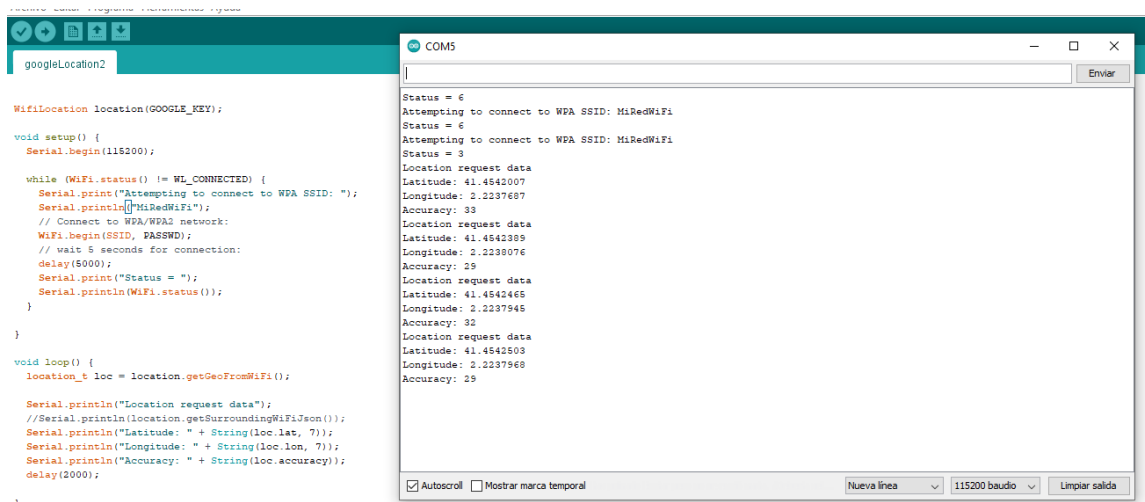
void loop() {
  delay(2000); //Esperamos 2 segundos
  float h = dht.readHumidity(); //Leemos la Humedad
  float t = dht.readTemperature(); //Leemos la temperatura en
  // Mostramos los datos en el Monitor Serie
  Serial.print("Humedad: ");
  Serial.print(h); //Valor de humedad
  Serial.print(" "); //Espacio para que sea más legible
  Serial.print("Temperatura: ");
  Serial.print(t); //Temperatura en grado Celsius
  Serial.println(" °C ");
}
```

Humedad	Temperatura
50.00	25.00 °C
50.00	26.00 °C
51.00	26.00 °C
51.00	26.00 °C
50.00	26.00 °C
65.00	27.00 °C
70.00	27.00 °C
84.00	28.00 °C
82.00	29.00 °C
76.00	29.00 °C
71.00	28.00 °C
66.00	29.00 °C
62.00	28.00 °C
58.00	27.00 °C
57.00	26.00 °C

Imatge 43: Prova de funcionament del sensor DHT11

En aquest cas, la prova que hem realitzat ha estat provar el sensor d'humitat i temperatura DHT11 al carrer. Per simular una pujada d'humitat, hem llençat l'alè a prop del sensor i així observant el que ha passat. Podem observar la pujada d'humitat de forma brusca en el moment que llancem l'alè i el petit increment de temperatura produït per la temperatura calenta del alè. Una vegada deixem de llençar-lo, el sensor torna a estabilitzar-se.

6.5 Prova de la geolocalització



```
googleLocation2

WifiLocation location(GOOGLE_KEY);

void setup() {
  Serial.begin(115200);

  while (WiFi.status() != WL_CONNECTED) {
    Serial.print("Attempting to connect to WPA SSID: ");
    Serial.println("MiRedWiFi");
    // Connect to WPA/WPA2 network:
    WiFi.begin(SSID, PASSWD);
    // wait 5 seconds for connection:
    delay(5000);
    Serial.print("Status = ");
    Serial.println(WiFi.status());
  }
}

void loop() {
  location_t loc = location.getGeoFromWiFi();

  Serial.println("Location request data");
  //Serial.println(location.getSurroundingWiFiJson());
  Serial.println("Latitude: " + String(loc.lat, 7));
  Serial.println("Longitude: " + String(loc.lon, 7));
  Serial.println("Accuracy: " + String(loc.accuracy));
  delay(2000);
}
```

```
COM5

Status = 6
Attempting to connect to WPA SSID: MiRedWiFi
Status = 6
Attempting to connect to WPA SSID: MiRedWiFi
Status = 3
Location request data
Latitude: 41.4542007
Longitude: 2.2237687
Accuracy: 33
Location request data
Latitude: 41.4542389
Longitude: 2.2238076
Accuracy: 29
Location request data
Latitude: 41.4542465
Longitude: 2.2237945
Accuracy: 32
Location request data
Latitude: 41.4542503
Longitude: 2.2237968
Accuracy: 29

 Autoscroll  Mostrar marca temporal Nueva línea 115200 baudio Limpiar salida
```

Imatge 44: Prova de funcionament del NodeMCU

En aquest cas, la prova que hem realitzat ha estat provar la geocalització. Primer de tot el NodeMCU ha d'establir connexió amb la API de Google Maps per rebre la posició en la que ens trobem. Mentre no ha establert connexió podem observar com el "Status" és diferent a 3, que vol dir que ha establert connexió. A partir d'aquí, simplement el sistema cada 2 segons crida a la funció de geocalització i ens va retornant la posició en la que ens trobem.

6.6 Prova del projecte sencer

```
googleLocation4
Serial.println();
Serial.print("Connected to ");
Serial.println("MiRedWiFi");
//Serial.print("IP Address is : ");
//Serial.println(WiFi.localIP());
Firebase.begin(FIREBASE_HOST, FIREBASE_AUTH); // connect to firebase
delay(1000);
}

void loop() {
// Obtenemos la geolocalización WiFi
loc = location.getGeoFromWiFi();
float h = dht.readHumidity(); //Leemos la Humedad
float t = dht.readTemperature(); //Leemos la temperatura en grados Celsius
int value = digitalRead(LDR);
String auxldr;
int sensorData = digitalRead(noisePin);
String auxnoise;
boolean mq_estado = digitalRead(Mq_07);//Leemos el sensor
String auxmq;

// Firebase Error Handling *****
if (Firebase.failed()){
  delay(500);
  Firebase.begin(FIREBASE_HOST, FIREBASE_AUTH);
  Serial.println(Firebase.error());
  Serial.println("Connection to firebase failed. Reconnecting...");
  delay(500);
}

COM5
-----
Connected to MiRedWiFi
Location request data
Latitude: 41.4542313
Longitude: 2.2239257
Accuracy: 28
Humedad 52 Temperatura: 22 °C
Es de noche, hace falta alumbrado
Limite de contaminacion aceptable
Limite de ruido aceptable
-----
Location request data
Latitude: 41.4542313
Longitude: 2.2239133
Accuracy: 29
Humedad 52 Temperatura: 22 °C
Es de noche, hace falta alumbrado
Limite de contaminacion aceptable
Limite de ruido aceptable
-----
Location request data
Latitude: 41.4543152
Longitude: 2.2239481
Autoscroll Mostrar marca temporal Nueva linea 115200 baudio Limpiar salida
```

Imatge 45: Prova de funcionament del codi sencer

Aquesta és la prova del projecte sencer, en la que testejarem la part de codi Arduino, la part de base de dades i la part de visualització en el mapa.

La prova s'ha realitzat a les 22:00h al carrer. A més, en aquest moment no hi ha molt de soroll al carrer i el nivell de contaminació no supera el límit. Així doncs, observem quins són els resultats trobats en aquest instant:

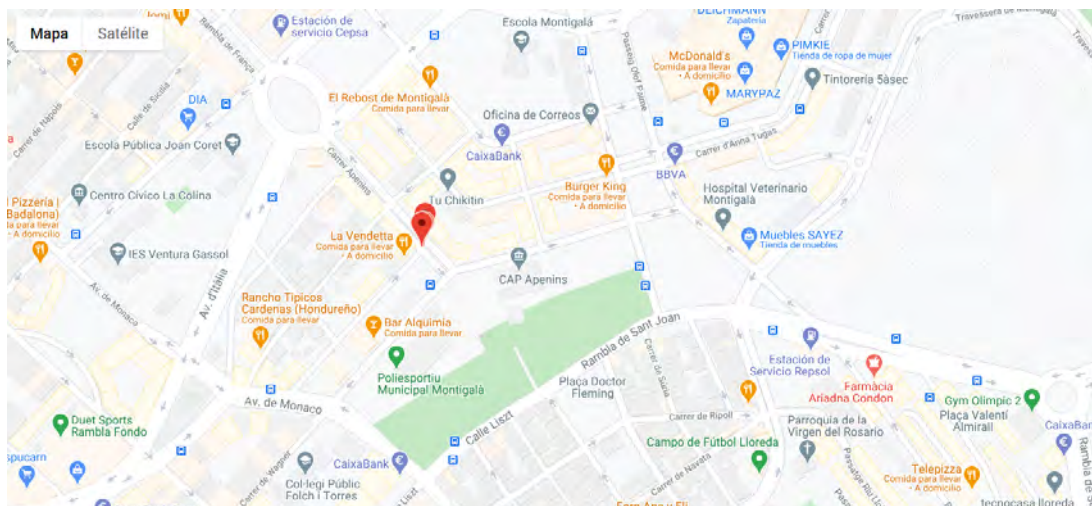
- Temperatura: 22°C.
- Humitat: 52%.
- Lluminositat: És de nit, s'han d'encendre els fanals.
- Contaminació: Límit de concentració acceptable.
- Soroll: Límit de soroll acceptable.

Els valors obtinguts en la prova són coherents amb la realitat al carrer. Anem a observar si s'han guardat correctament aquests valors a la base de dades.



Imatge 46: Prova de funcionament de la base de dades Firebase

Com podem observar en la imatge, a la base de dades els valors que s’han guardat són els mateixos que ens sortien pel monitor del Arduino, és a dir, hem enviat correctament les dades a Firebase. Per últim, visualitzarem en el mapa si la posició que hem obtingut en l’Arduino i s’ha emmagatzemat a Firebase, correspon amb les coordenades on s’han realitzat la prova.
















Imatge 47: Prova de funcionament del mapa

Finalment podem determinar que la prova ha estat un èxit, ja que la posició marcada amb el mapa i la posició on s’ha realitzat la prova, és la mateixa. Ens surten 3 marcadors, ja que la prova ha enregistrat 3 instants.

7 Estudi econòmic del projecte

A continuació, tractarem la inversió econòmica emprada en aquest projecte:

	Imatge	Unitats	Preu (€)
Arduino Mega 2560		1	15,80
NodeMCU		1	4,96
Sensor LDR		1	0,73
Sensor CO2		1	3,39
Sensor DHT11		1	1,60
Sensor Noise		1	1,65
Cable micro-usb	 <small>Micro USB 8mm</small>	1	3,67

Cable USB 2.0 Type A/B		1	2,93
Placa ProtoBoard		1	4,95
Resistors		2	0,15
Cables ProtoBoard mascle/masclle		6	1,65
Cables ProtoBoard mascle/femella		6	1,97
Cables ProtoBoard femella/femella		6	1,87
Preu TOTAL			45,32 €

Taula 2: Taula del pressupost del projecte

Així doncs, descrits i visualitzats tots el components necessaris per poder dur a terme aquest projecte, podem observar com el cost material és d'uns 45,32€. Com a comentari rellevant, l'Arduino Mega 2560 és únicament utilitzat per el subministrament de 5V, ja que molts sensors necessiten una alimentació superior als 3,3V. Al apartat de problemes i consideracions explicarem el perquè del us d'aquest Arduino i no d'una font de tensió (com una pila).

8 Problemes i consideracions del projecte

Aquest projecte ha estat un dels més complexos que he realitzat mai, per una banda per la interconnexió de tants blocs individuals a realitzar (Arduino, firebase, html,...) i per l'altre degut a la resolució individual del projecte, ja que en projectes d'aquest caire sempre hem treballat en grup.

Un dels problemes més importants que he tingut i on he invertit molt temps d'aquest projecte, ha estat en l'obtenció de les coordenades geogràfiques. En un primer moment, la idea era utilitzar un sensor Neo6MV2, el qual mitjançant amb un Arduino Mega 2560, s'encarregaria de retornar-nos les coordenades sense problemes, tal i com vaig estar informant-me. Segons la recerca que vaig fer, aquest component es comunicava amb l'Arduino mitjançant els ports TX i RX de tots dos components i prometia tenir una fàcil sincronització entre els satèl·lits i el component, no obstant no va ser així.

Així doncs vaig optar per l'utilització d'un NodeMCU, el qual obté les dades de les coordenades via WiFi, per tant, si aconseguim establir una connexió WiFi acceptable pel NodeMCU, tindrem les coordenades de la posició en la que ens trobem.

Un altre problema important que vaig tenir, va ser la connexió amb el servidor (base de dades). Al principi volia optar per la idea que em va suggerir el meu tutor del projecte, de treballar amb SENTILO. SENTILO és una plataforma de codi obert patrocinada per l'Ajuntament de Barcelona. Està dissenyada, a més, per adaptar-se a nivell de sensors i actuadors de qualsevol ciutat de manera simple. No obstant, no trobava molta informació i no sabia molt bé com enfocar-ho, així que vaig optar per una altra base de dades (FireBase). Amb FireBase va ser bastant còmode treballar i bastant intuïtiu la forma de treballar, tal i com queda expressat en apartats anteriors, tot i que també m'ha donat bastants problemes.

El problema més important amb la connexió amb FireBase es produeix quan vull emmagatzemar una coordenada geogràfica relativament igual a l'anterior, és a dir, quan ens trobem quietes. El que sorgeix és que el servidor guarda a la base de dades una posició de latitud i longitud de 0. Aquest ha estat l'únic problema que no he pogut resoldre.

9 Conclusions

En el aquest projecte s'ha estudiat la implantació d'un sistema de captació de dades per un vianant o ciclista mitjançant un NodeMCU i uns quants sensors, per analitzar com ens movem per la nostre ciutat.

S'ha d'incidir en primer lloc, abans de res, en la importància de portar les ciutats cap a Smart Cities o ciutats intel·ligents, amb la única finalitat de millorar la vida i la salut dels ciutadans d'aquesta i aprofitar els recursos dels que disposem d'una forma més eficient .

Tot i que la ciutat de Barcelona, és encara una ciutat “jove” en el món de les Smart Cities, comencem a trobar diferents tipus de serveis relacionats amb les TICs que poden marcar l'inici d'una ciutat més intel·ligent, i en definitiva, millor per viure.

Aquest projecte pot marcar l'inici de molts altres projectes, que potser no tenen res a veure amb aquest, però si que puguin utilitzar els resultats d'aquest, ja que saber com es mouran el ciutadans d'una ciutat, ens pot indicar, per exemple, si és necessari tenir més o menys transport públic en una zona (aprofitar recursos) segons el número de ciutadans que circulin (encara que sigui a peu) per aquella zona.

A nivell personal, d'aquest projecte hem complert tots i cadascun dels objectius que ens havíem plantejat inicialment. Hem pogut posar en pràctica molts dels coneixements adquirits durant la nostra formació universitària i d'aquesta manera assentar la base que teníem i reforçar-la amb un projecte real.

Pel que fa la part més electrònica, hem aprofundit molt més en la programació de mòduls Arduino i NodeMCU, i també en la part de sensors, en la qual teníem una base pràcticament teòrica, hem pogut analitzar diferents tipus i saber quins són més útils en cada moment depenent el projecte que estem tractant.

Pel que fa a la part de bases de dades, hem pràcticament après des de 0, com funciona una, com podem fer crides, com gestionar la informació, etc..., per això estem

especialment orgullosos en la forma tan autodidacta que hem pogut resoldre cadascun dels problemes amb els que ens trobat.

Pel que fa a la part Smart Cities, considerem que hem consolidat aquells coneixements que hem vist en assignatures del Master i aprofundit una mica més en aquests temes, ja sigui en la importància que tenen i tindran, la nova forma en que seran gestionades les ciutats, l'optimització de recursos i serveis que tindrem, etc.

Tot i que en aquest projecte no hem pres les millors decisions per elaborar-lo, degut a l'elevada inversió que suposaria els diferents components o tecnologies, hem sabut resoldre el problema que s'ens plantejava inicialment. Òbviament, que amb un pressupost molt més elevat al actual i uns coneixements molt més avançats en certes àrees, el projecte quedaria resolt d'una altre forma.

Econòmicament parlant, aquest no és un projecte excessivament car en un inici. Depenent com vulguem desenvolupar-lo i la infraestructura que vulguem desplegar, podríem tenir un gran i car projecte. Per exemple, en el meu cas per poder marcar el meu recorregut a peu pel carrer, vaig utilitzar una xarxa WiFi d'un local pròxim a casa meva que són coneguts meus, però si volem portar aquest projecte per tota la ciutat potser necessitaríem fer un desplegament d'Acces points i repetidors amb una xarxa pública que abastés tota la ciutat, i per tant s'encariria el projecte molt més.

10 Línies de futur

Aquest projecte es podria dir que s'ha fet pensant en gran part en les línies de futur. Trobo moltes formes d'evolucionar amb aquest projecte.

La primera seria per exemple substituir tot el hardware que he utilitzat amb l'Arduino per material una mica més usable, ja que és bastant incòmode circular pel carrer amb la placa, els components, l'ordinador, etc... El localitzador de posició podria tenir una forma de polsera o rellotge, per exemple, que no molestés a la vianant o ciclista.

Una altre opció seria mitjançant una aplicació mòbil. El mòbil és un element indispensable per tots el ciutadans de la ciutat i és força estrany trobar alguna persona que no en tingui. Per això, amb una aplicació mòbil que s'encarregués de marcar el recorregut que realitzes, utilitzant el micròfon d'aquest com a sensor de soroll i algun dispositiu extern que s'uniria al mòbil per substituir els altres sensors, podríem tenir una altre forma de desenvolupar el projecte.

Com una última forma d'evolucionar el projecte, podria ser fer un estudi per de desenvolupar un sistema de xarxa pública i gratuïta per tots el habitants, amb Access Points i repetidors per tota la ciutat amb la finalitat de poder marcar el recorregut del ciutadà sense que es produeixi cap problema o canvi de connexió de xarxa.

11 Bibliografía

- 2020.[online] Available at: <https://www.youtube.com/watch?v=5ezyekFaVJI&t=232s> [Accessed 8 September 2020].
- 2020.[online] Available at: <https://www.youtube.com/watch?v=nLzC0-VaqDs> [Accessed 9 September 2020].
- Aktar, K., 2020. *How To Connect To An SSL Protected Server With ESP8266(Wificlient)*. [online]
- Theamplituhedron.com. Available at: [https://www.theamplituhedron.com/articles/How-to-connect-to-an-SSL-protected-server-with-ESP8266\(WiFiClient\)/](https://www.theamplituhedron.com/articles/How-to-connect-to-an-SSL-protected-server-with-ESP8266(WiFiClient)/) [Accessed 9 September 2020].
- Alex7tutoriales.blogspot.com. 2020. *ESP8266 Y Firebase - Arduino*. [online] Available at: <https://alex7tutoriales.blogspot.com/2020/03/esp8266-y-firebase-arduino.html> [Accessed 8 September 2020].
- Arduino.cl - Compra tu Arduino en Línea. 2020. *Arduino Mega 2560 / Arduino.Cl - Compra Tu Arduino En Línea*. [online] Available at: <https://arduino.cl/arduino-mega-2560/> [Accessed 9 September 2020].
- Arduinolibraries.info. 2020. *Firestore*. [online] Available at: <https://www.arduinolibraries.info/libraries/firebase-json> [Accessed 9 September 2020].
- Console.firebase.google.com. 2020. *Sign In - Google Accounts*. [online] Available at: <https://console.firebase.google.com/project/thamos/database/thamos/data> [Accessed 8 September 2020].
- Coordenadas-gps.com. 2020. *Crear Su Propio Mapa Personalizado Google*. [online] Available at: <https://www.coordenadas-gps.com/mapa-personalizado> [Accessed 9 September 2020].
- Es.batchgeo.com. 2020. *Batchgeo: Crear Mapas A Partir De Sus Datos*. [online] Available at: <https://es.batchgeo.com> [Accessed 9 September 2020].

- Esp8266-arduino-spanish.readthedocs.io. 2020. *Clase: Client Secure — Documentación De ESP8266 Arduino Core - 2.4.0.* [online] Available at: <https://esp8266-arduino-spanish.readthedocs.io/es/latest/esp8266wifi/client-secure-class.html> [Accessed 9 September 2020].
- firebase, E., 2020. *ESP8266 And Https Connection To Firebase.* [online] Stack Overflow. Available at: <https://stackoverflow.com/questions/58533549/esp8266-and-https-connection-to-firebase> [Accessed 9 September 2020].
- GitHub. 2020. *HTTPS Error · Issue #10 · Gmag11/Wifilocation.* [online] Available at: <https://github.com/gmag11/WifiLocation/issues/10> [Accessed 9 September 2020].
- GitHub. 2020. *Mobizt/Firebase-Arduino-Wifi101.* [online] Available at: <https://github.com/mobizt/Firebase-Arduino-WiFi101> [Accessed 9 September 2020].
- Gsampallo.com. 2020. *Nodemcu Y Micropython: Múltiples Entradas Analógicas – Software / Iot.* [online] Available at: <https://www.gsampallo.com/2019/09/11/nodemcu-y-micropython-multiples-entradas-analogicas/> [Accessed 9 September 2020].
- soy?, ¿., 2020. *Cómo Utilizar El DHT11 Para Medir La Temperatura Y Humedad Con Arduino.* [online] Programar fácil con Arduino. Available at: <https://programarfácil.com/blog/arduino-blog/sensor-dht11-temperatura-humedad-arduino/> [Accessed 9 September 2020].
- 2020. [online] Available at: https://www.youtube.com/watch?v=asVv6VZld_o [Accessed 9 September 2020].
- Arduino Hobby. 2020. *Sensor De Luz Con Arduino (Módulo) - Arduino Hobby.* [online] Available at: <https://www.arduinhobby.com/sensor-de-luz-con-arduino-modulo/> [Accessed 9 September 2020].
- Arduino Project Hub. 2020. *How To Interface Arduino Mega With NEO-6M GPS Module.* [online] Available at: <https://create.arduino.cc/projecthub/ruchir1674/how-to-interface-arduino-mega-with-neo-6m-gps-module-1b7283> [Accessed 9 September 2020].

- Last Minute Engineers. 2020. *In-Depth: Interface Sound Sensor With Arduino And Control Devices With A Clap*. [online] Available at: <https://lastminuteengineers.com/sound-sensor-arduino-tutorial/> [Accessed 9 September 2020].

- Naylampmechatronics.com. 2020. *Tutorial Sensores De Gas MQ2, MQ3, MQ7 Y MQ135*. [online] Available at: https://naylampmechatronics.com/blog/42_Tutorial-sensores-de-gas-MQ2-MQ3-MQ7-y-MQ13.html [Accessed 9 September 2020].

12 Annex

12.1 Codi Arduino

12.1.1 Codi GPS i NodeMCU

```
googleLocation2 $
#include <ESP8266WiFi.h>
#include "WifiLocation.h"
#define GOOGLE_KEY "AIzaSyC6rmSqS6rPnlf3aTYxmSQk0dRko3Zdth8"
#define SSID "*****"
#define PASSWD "*****"
WifiLocation location(GOOGLE_KEY);

void setup() {
  Serial.begin(115200);

  while (WiFi.status() != WL_CONNECTED) {
    Serial.print("Attempting to connect to WPA SSID: ");
    Serial.println(SSID);
    // Connect to WPA/WPA2 network:
    WiFi.begin(SSID, PASSWD);
    // wait 5 seconds for connection:
    delay(5000);
    Serial.print("Status = ");
    Serial.println(WiFi.status());
  }
  location_t loc = location.getGeoFromWiFi();

  Serial.println("Location request data");
  Serial.println(location.getSurroundingWiFiJson());
  Serial.println("Latitude: " + String(loc.lat, 7));
  Serial.println("Longitude: " + String(loc.lon, 7));
  Serial.println("Accuracy: " + String(loc.accuracy));
}

void loop() {
}
```

Imatge 48: Codi d'obtenció de coordenades

12.1.2 Codi Sensor DHT11



```
pruebahyt
#include "DHT.h" //Incluimos la librería

#define DHTPIN 2 // Especificamos que el pin 2 se conectará con el pin de datos del sensor

#define DHTTYPE DHT11 // Informamos que usaremos el DHT11

DHT dht(DHTPIN, DHTTYPE);

void setup() {
  Serial.begin(9600); //Configuramos la velocidad del Puerto Serie
  Serial.println("Iniciando..."); //Mensaje que aparecerá en el Monitor Serie
  dht.begin();
}
void loop() {
  delay(2000); //Esperamos 2 segundos
  float h = dht.readHumidity(); //Leemos la Humedad
  float t = dht.readTemperature(); //Leemos la temperatura en grados Celsius

  // Mostramos los datos en el Monitor Serie
  Serial.print("Humedad ");
  Serial.print(h); //Valor de humedad
  Serial.print(" "); //Espacio para que sea más legible
  Serial.print("Temperatura: ");
  Serial.print(t); //Temperatura en grado Celsius
  Serial.print(" °C ");
}
```

Imatge 49: Codi Sensor DHT11

12.1.3 Codi Sensor de gas MQ



```
pruebaCO2
int pin_mq = 2;

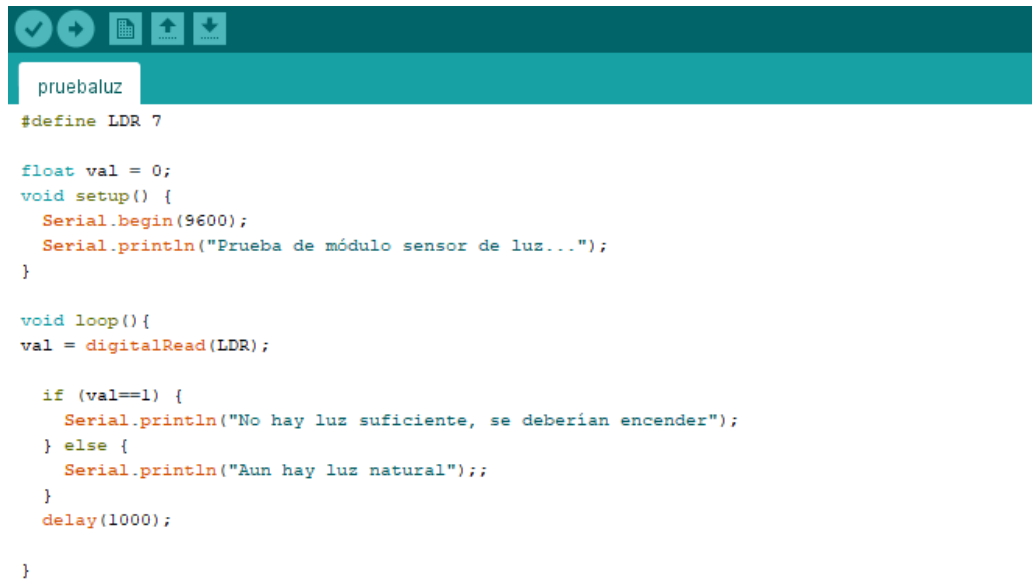
void setup() {
  Serial.begin(9600);
  pinMode(pin_mq, INPUT);
}

void loop() {

  boolean mq_estado = digitalRead(pin_mq); //Leemos el sensor
  if(mq_estado) //si la salida del sensor es 1
  {
    Serial.println("Sin presencia de alcohol");
  }
  else //si la salida del sensor es 0
  {
    Serial.println("Alcohol detectado");
  }
  delay(1000);
}
```

Imatge 50: Codi Sensor MQ-7

12.1.4 Codi Sensor de llum LDR



```
pruebaluz
#define LDR 7

float val = 0;
void setup() {
  Serial.begin(9600);
  Serial.println("Prueba de módulo sensor de luz...");
}

void loop(){
  val = digitalRead(LDR);

  if (val==1) {
    Serial.println("No hay luz suficiente, se deberían encender");
  } else {
    Serial.println("Aun hay luz natural");;
  }
  delay(1000);
}
```

Imatge 51: Codi Sensor LDR

12.1.5 Codi Sensor de soroll CN07



```
pruebanosie $
#define sensorPin 2

// Variable to store the time when last event happened
unsigned long lastEvent = 0;

void setup() {
  pinMode(sensorPin, INPUT); // Set sensor pin as an INPUT
  Serial.begin(9600);
}

void loop() {
  // Read Sound sensor
  int sensorData = digitalRead(sensorPin);

  // If pin goes LOW, sound is detected
  if (sensorData == LOW) {

    // If 25ms have passed since last LOW state, it means that
    // the clap is detected and not due to any spurious sounds
    if (millis() - lastEvent > 25) {
      Serial.println("Exceso de ruido!!");
    }

    // Remember when last event happened
    lastEvent = millis();
  }
}
```

Imatge 52: Codi Sensor de so

12.1.6 Codi Arduino (Conexió Hardware-FireBase)



```
googleLocation3$
#include <ESP8266WiFi.h>
#include <FirebaseArduino.h>
#include "WifiLocation.h"
#include "DHT.h" //Incluïmos la libreria
#define GOOGLE_KEY "AIzaSyC6rmSqS6rPnlf3aTYxmSQk0dRko3Zdth8" // Clave API Google Geolocation
#define FIREBASE_HOST "thamos.firebaseio.com" //Your Firebase Project URL goes here without "http:" , "\" and "/"
#define FIREBASE_AUTH "zTUDKkfwjmUaVM8GqslhS4ujFVRmpZvbMYkrirzY" //Your Firebase Database Secret goes here
#define WIFI_SSID "*****" //WiFi SSID to which you want NodeMCU to connect
#define WIFI_PASSWORD "*****"//Password of your wifi network
#define LOC_PRECISION 7 // Precisión de latitud y longitud
#define DHTPIN 1 // Especificamos que el pin 2 se conectará con el pin de datos del sensor
#define DHTTYPE DHT11 // Informamos que usaremos el DHT11
#define LDR 2
#define Mg_07 3
#define noisePin 4
```

Imatge 53: Definim les llibreries i les constants

```
void setup() {

  Serial.begin(115200);
  Serial.println("Serial communication started\n\n");
  WiFi.begin(WIFI_SSID, WIFI_PASSWORD); //try to connect with wifi
  Serial.print("Connecting to ");
  Serial.print(WIFI_SSID);
  dht.begin();

  while (WiFi.status() != WL_CONNECTED) {
    Serial.print(".");
    delay(500);
  }

  Serial.println();
  Serial.print("Connected to ");
  Serial.println(WIFI_SSID);
  Serial.print("IP Address is : ");
  Serial.println(WiFi.localIP()); //print local IP address
  Firebase.begin(FIREBASE_HOST, FIREBASE_AUTH); // connect to firebase
  delay(1000);
}
```

Imatge 54: Establiment de connexió Wifi i amb Firebase



```
String aux = "/dispositivo/";
aux += val;
aux += "/lat/";
Firebase.setFloat(aux, loc.lat);
aux = "/dispositivo/";
aux += val;
aux += "/lon/";
Firebase.setFloat(aux, loc.lon);
aux = "/dispositivo/";
aux += val;
aux += "/nombre/";
Firebase.setString(aux, nombreComun);
aux = "/dispositivo/";
aux += val;
aux += "/prec/";
Firebase.setFloat(aux, loc.accuracy);
aux = "/dispositivo/";
aux += val;
aux += "/humedad/";
//Falta humedad
Firebase.setFloat(aux, h);
aux = "/dispositivo/";
aux += val;
aux += "/temperatura/";
//Falta temperatura
Firebase.setFloat(aux, t);
aux = "/dispositivo/";
aux += val;
aux += "/ruido/";
Firebase.setString(aux, auxnoise);
aux = "/dispositivo/";
aux += val;
aux += "/luz/";
Firebase.setString(aux, auxldr);
aux = "/dispositivo/";
aux += val;
aux += "/CO2/";
Firebase.setString(aux, auxmq);
```

Imatge 55: Enviem dades al servidor Firebase

12.1.7 Codi Arduino (Complert)

```
googleLocation3$
#include <ESP8266WiFi.h>
#include <FirebaseArduino.h>
#include "WifiLocation.h"
#include "DHT.h" //Incluimos la librería
#define GOOGLE_KEY "AIzaSyC6rmSqS6rPnlf3aTYxmSQk0dRko3Zdth8" // Clave API Google Geolocation
#define FIREBASE_HOST "thamos.firebaseio.com" //Your Firebase Project URL
#define FIREBASE_AUTH "zTUDKkfwjmUaVM8GqslhS4ujFVRmpZvbMYkrirzY" //Your Firebase Database Sec
#define WIFI_SSID "*****" //WiFi SSID to which
#define WIFI_PASSWORD "*****"//Password of your wifi network
#define LOC_PRECISION 7 // Precisión de latitud y longitud
#define DHTPIN 1 // Especificamos que el pin 2 se conectará con el pin de datos del sensor
#define DHTTYPE DHT11 // Informamos que usaremos el DHT11
#define LDR 2
#define Mq_07 3
#define noisePin 4

// Llamada a la API de Google
WifiLocation location(GOOGLE_KEY);
location_t loc; // Estructura de datos que devuelve la librería WifiLocation
DHT dht(DHTPIN, DHTTYPE);

// Variables
int val=0;
String nombreComun = "NodeMCU";
unsigned long lastEvent = 0;

void setup() {

  Serial.begin(115200);
  Serial.println("Serial communication started\n\n");
  WiFi.begin(WIFI_SSID, WIFI_PASSWORD); //try to connect w
  Serial.print("Connecting to ");
  Serial.print(WIFI_SSID);
  dht.begin();

  while (WiFi.status() != WL_CONNECTED) {
    Serial.print(".");
    delay(500);
  }

  Serial.println();
  Serial.print("Connected to ");
  Serial.println(WIFI_SSID);
  Serial.print("IP Address is : ");
  Serial.println(WiFi.localIP()); //print local IP
  Firebase.begin(FIREBASE_HOST, FIREBASE_AUTH); // connect to firebase
  delay(1000);
}

void loop() {
  // Obtenemos la geolocalización WiFi
  loc = location.getGeoFromWiFi();
  float h = dht.readHumidity(); //Leemos la Humedad
  float t = dht.readTemperature(); //Leemos la temperatura en grados Celsius
  int value = digitalRead(LDR);
  String auxldr;
  int sensorData = digitalRead(noisePin);
  String auxnoise;
  boolean mq_estado = digitalRead(Mq_07); //Leemos el sensor
```



```

String auxmq;

// Firebase Error Handling *****
if (Firebase.failed())
{ delay(500);
  Firebase.begin(FIREBASE_HOST, FIREBASE_AUTH);
  Serial.println(Firebase.error());
  Serial.println("Connection to firebase failed. Reconnecting...");
  delay(500);
}

else {

////////////////////////////////////
// Mostramos la información en el monitor serie
Serial.println("Location request data");
Serial.println(location.getSurroundingWiFiJson());
Serial.println("Latitude: " + String(loc.lat, 7));
Serial.println("Longitude: " + String(loc.lon, 7));
Serial.println("Accuracy: " + String(loc.accuracy));
////////////////////////////////////
Serial.print("Humedad ");
Serial.print(h); //Valor de humedad
Serial.print(" "); //Espacio para que sea más legible
Serial.print("Temperatura: ");
Serial.print(t); //Temperatura en grado Celsius
Serial.println(" °C ");
////////////////////////////////////
if (value == 1){
  Serial.println("Es de noche, hace falta alumbrado");

  auxldr ="Es de noche, hace falta alumbrado";
} else{
  Serial.println("Es de dia, no hace falta alumbrado");
  auxldr ="Es de dia, no hace falta alumbrado";
}
////////////////////////////////////
if(mq_estado) //si la salida del sensor es 1
{
  Serial.println("Limite de contaminacion aceptable");
  auxmq ="Limite de contaminacion aceptable";
}
else //si la salida del sensor es 0
{
  Serial.println("Exceso de contaminacion");
  auxmq ="Exceso de contaminacion";
}
////////////////////////////////////
if (sensorData == LOW) {

  // If 25ms have passed since last LOW state, it means that
  // the clap is detected and not due to any spurious sounds
  if (millis() - lastEvent > 25) {
    Serial.println("Superando el limite de ruido");
    auxnoise = "Superando el limite de ruido";
  }else{
    Serial.println("Limite de ruido aceptable");
    auxnoise = "Limite de ruido aceptable";
  }

  // Remember when last event happened

```

```

    lastEvent = millis();
}
////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
String aux = "/dispositivo/";
aux += val;
aux += "/lat/";
Firebase.setFloat(aux, loc.lat);
aux = "/dispositivo/";
aux += val;
aux += "/lon/";
Firebase.setFloat(aux, loc.lon);
aux = "/dispositivo/";
aux += val;
aux += "/nombre/";
Firebase.setString(aux, nombreComun);
aux = "/dispositivo/";
aux += val;
aux += "/prec/";
Firebase.setFloat(aux, loc.accuracy);
aux = "/dispositivo/";
aux += val;
aux += "/humedad/";
//Falta humedad
Firebase.setFloat(aux, h);
aux = "/dispositivo/";
aux += val;
aux += "/temperatura/";
//Falta temperatura
Firebase.setFloat(aux, t);
aux = "/dispositivo/";
aux += val;
aux += "/ruido/";
Firebase.setString(aux, auxnoise);
aux = "/dispositivo/";
aux += val;
aux += "/luz/";
Firebase.setString(aux, auxldr);
aux = "/dispositivo/";
aux += val;
aux += "/CO2/";
Firebase.setString(aux, auxmq);

val++;
delay(5000);
}

}

```

Imatge 56: Codi Arduino complet del projecte

12.2 Base de dades Firebase



Imatge 57: Informació emmagatzemada a Firebase

12.3 Visualització de les dades

12.3.1 Codi HTML del mapa

```
mapa: Bloc de notes
Archivo Edición Formato Ver Ayuda
<!DOCTYPE html>
<html>
<head>
  <title>Geolocalizacion</title>
  <style type="text/css">
    html, body { height: 100%; margin: 0; padding: 0; }
    #map { height: 100%; }
  </style>
  <meta name="robots" content="noindex">
</head>
<body>
  <!--<ul id="costumers" class="list-group">
</ul-->
  <div id="map"></div>
  <!-- jQuery (necessary for Bootstrap's JavaScript plugins) -->
  <script src="https://ajax.googleapis.com/ajax/libs/jquery/1.11.2/jquery.min.js"></script>
  <!-- Latest compiled and minified Bootstrap -->
  <script src="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.4/js/bootstrap.min.js"></script>
  <!-- Include Firebase Library -->
  <script src='https://cdn.firebase.com/js/client/2.2.1/firebase.js'></script>
  <!-- Tracking Store JavaScript -->
  <script src="script.js"></script>
  <!-- API Google Maps -->
  <script async defer
    src="https://maps.googleapis.com/maps/api/js?key=AIzaSyC6rmSqS6rPn1f3aTYxmSQk0dRko3Zdth8&callback=initMap">
  </script>
</body>
</html>
```

Imatge 58: Codi HTML del mapa

12.3.2 Codi del script del mapa

 script: Bloc de notas

Archivo Edición Formato Ver Ayuda

```
//Firebase
var dbRef = new Firebase('https://thamos.firebaseio.com/');
var costumersRef = dbRef.child('dispositivo');
var markers = {}
//load persons
costumersRef.on("child_added", function(snap) {
  // MAPA
  addNewPerson(snap.val().lat,snap.val().lon,snap.val().nombre);
});

costumersRef.on("child_changed", function (snap) {
  changePerson(snap.val().lat, snap.val().lon, snap.val().nombre);
});

/***** GOOGLE MAPS *****/
var map;
function initMap() {
  // Iniciamos mapa
  var myLatLng = {lat: 41.452101, lng: 2.223467};
  map = new google.maps.Map(document.getElementById('map'), {
    center: myLatLng,
    zoom: 3
  });
}

function addNewPerson(lat, lon,nombre){
  var marker = new google.maps.Marker({
    position: new google.maps.LatLng(lat,lon),
    map: map,
    title: nombre // Tooltip = MAC address
  });
  markers[key] = marker;
}

function changePerson(lat, lon, nombre) {
  marker = markers[key];
  marker = new google.maps.Marker({
    position: new google.maps.LatLng(lat, lon),
    map: map,
    title: nombre // Tooltip = MAC address
  });
  markers[key] = marker;
}
```

Imatge 59: Script del mapa