

PROJETE FINAL

CONTROL REMOT DE FANALS



JESUÏTES El Clot
Escola del Clot

Sara Puig Cabruja

DAM 2

Curs 2019-2020

“The best way to predict the future is to implement it.”
— David Heinemeier Hansson

Índex

| | |
|--|----|
| Resum..... | 4 |
| Resumen..... | 4 |
| Abstract..... | 4 |
| 1 Estudi de viabilitat..... | 5 |
| 1.1 Establiment de l'abast del sistema..... | 5 |
| 1.2 Estudi de la situació actual..... | 5 |
| 1.3 Definició dels requisits del sistema..... | 6 |
| Requisits funcionals..... | 6 |
| Requisits tècnics..... | 7 |
| 1.4 Estudi de les alternatives de solució..... | 7 |
| 1.5 Valoració de les alternatives..... | 9 |
| 1.6 Selecció de la solució..... | 10 |
| 2 Anàlisi del sistema..... | 11 |
| 2.1 Definició del sistema..... | 11 |
| 2.2 Definició d'interfícies d'usuari..... | 11 |
| 2.3 Especificació del pla de proves..... | 12 |
| 3 Disseny del sistema..... | 19 |
| 3.1 Arquitectura del sistema..... | 19 |
| 3.2 Revisió de casos d'ús..... | 20 |
| 3.3 Anàlisi paradigma Orientat a Objectes..... | 21 |
| 3.4 Disseny paradigma Orientat a Objectes..... | 21 |
| 3.5 Persistència de dades: Anàlisi i disseny de bases de dades..... | 25 |
| 4 Desenvolupament..... | 28 |
| 4.1 Planificació de les activitats de desenvolupament i integració de sistema..... | 28 |
| 4.2 Desenvolupament..... | 29 |
| 5 Implantació..... | 30 |
| 5.1 Formació..... | 30 |
| 5.2 Implantació del sistema..... | 30 |
| 6 Manteniment i versions futures..... | 31 |
| Pressupost..... | 32 |
| Manual d'ús..... | 33 |
| Informació específica, enllaços..... | 41 |

Resum

Aquest projecte desenvolupa una solució IoT per a poder controlar remotament els fanals d'una plaça mitjançant una aplicació multiplataforma. Aquesta ens permetrà fer canvis en els colors de la il·luminació de la plaça fanal a fanal o configurant diverses agrupacions. L'aplicació s'ha desenvolupat amb Qt i desa la informació en una base de dades MySQL. La solució inclou també el desenvolupament de microprogramari amb Arduino IDE, per al maquinari al qual estaran connectats els leds dels fanals.

Resumen

Este proyecto desarrolla una solución IoT que permite el control remoto de las farolas de una plaza mediante una aplicación multiplataforma. La aplicación nos permitirá cambiar los colores que iluminan la plaza farola a farola o mediante agrupaciones. La aplicación se ha desarrollado con Qt y guarda la información en una base de datos MySQL. La solución incluye también el desarrollo de firmware con Arduino IDE, para el hardware al que estarán conectados los leds de las farolas.

Abstract

This project develops an IoT solution that allows remote control of street lamps on a square through a cross-platform application. This application lets you change the colors that illuminate the square from each street light or to a whole grup of them. The application has been developed with Qt and stores its information into a MySQL database. This solution also includes a firmware, developed using Arduino IDE, that will be placed to the hardware that manages the LED's build in the street lights.

1 Estudi de viabilitat

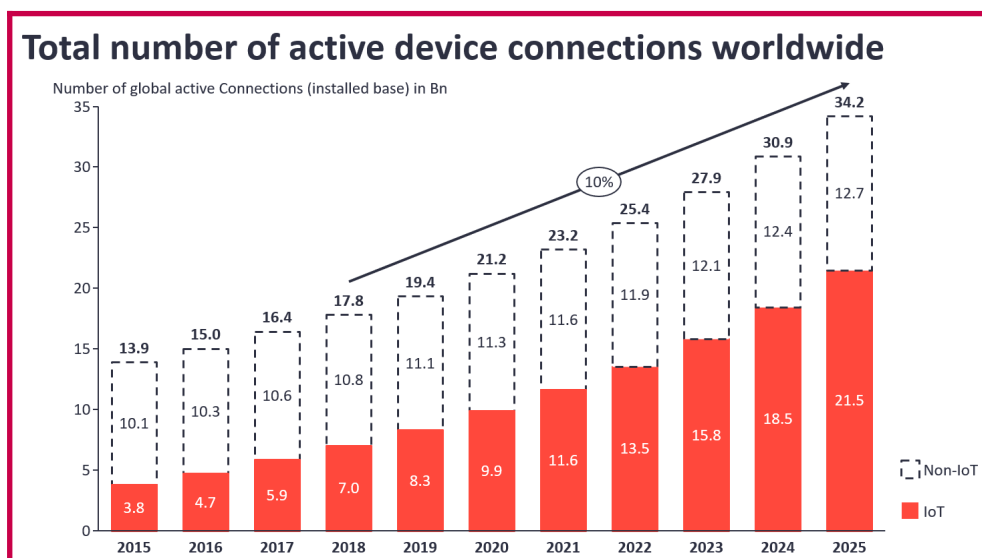
L'estudi de viabilitat ens permetrà saber si aquest projecte prosperarà i tindrà bona acollida entre el públic, tot i que d'entrada ja és una petició per part d'un client, no podem ometre'l. L'objectiu d'aquest és facilitar la presa de decisions a l'hora d'implementar la millor solució possible davant les necessitats expressades pel client.

1.1 Establiment de l'abast del sistema

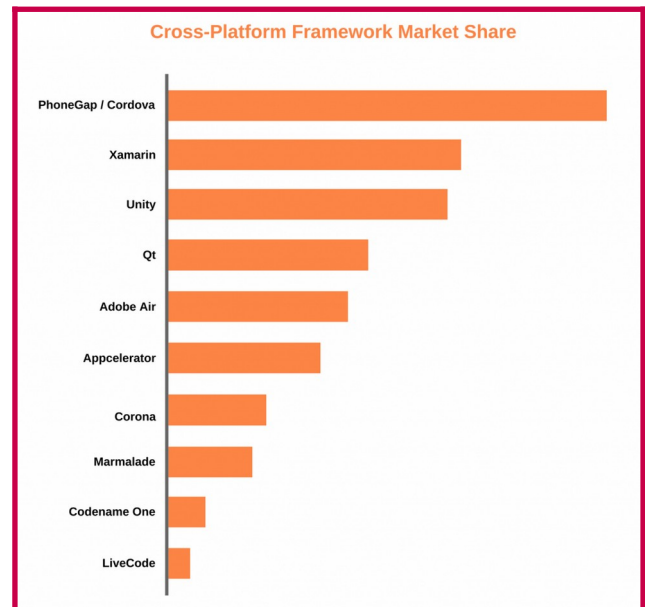
Aquest projecte s'ha desenvolupat amb programari lliure en tant que ha sigut possible. Consta d'una aplicació multiplataforma, d'un programa executat com daemon en un servidor virtual privat (VPS) per accedir a la base de dades, d'una base de dades en un VPS on emmagatzemar la informació i finalment un programa per a les plaques que controlen els fanals.

1.2 Estudi de la situació actual

Fa uns anys va sorgir la necessitat de tenir diverses coses de la vida quotidiana connectades per tal de poder compartir informació entre elles, aquesta necessitat segueix creixent avui en dia. Durant els últims anys, IoT s'ha convertit en una de les tecnologies més importants. Aquesta tecnologia, que combina el món físic i el món digital, ens permet una comunicació fluida entre persones, processos i coses amb una intervenció humana mínima. El mercat d'Internet de les coses (IoT) es troba en ple creixement i expansió gràcies a l'alta demanda d'aplicacions i solucions derivades d'aquesta necessita.



Les diferents plataformes per a utilitzar aplicacions es troben en una espiral de desenvolupament i canvis constants, fet que potència la venda i l'èxit d'aplicacions multiplataforma. Actualment podem trobar diverses eines per a la realització d'aquestes aplicacions, en són exemples React Native, Ionic, Qt i Unity, entre d'altres.



1.3 Definició dels requisits del sistema

Requisits funcionals

El que se'ns demana és tot el programari necessari per a poder controlar remotament els fanals d'una plaça. Els principals requisits funcionals són els següents:

RQF1 – L'aplicació ha de permetre la configuració de la plaça.

RQF2 – L'aplicació ha de permetre la configuració dels fanals.

RQF3 – L'aplicació ha de permetre la configuració dels grups.

RQF4 – L'aplicació ha de permetre el canvi de color dels leds, segons els grups configurats o individualment.

RQF5 – L'aplicació ha de permetre que la plaça tingui més d'una configuració de grups.

RQF6 – L'aplicació ha de permetre visualitzar un mapa de la plaça.

RQF7 – El microprogramari de les plaques ha de permetre el canvi color dels leds corresponent als canvis realitzats des de l'aplicació.

Requisits tècnics

Aquest projecte és un projecte real pensat per a una plaça de Sabadell coneguda com a Vapor Cosidó. Els requisits tècnics per a poder utilitzar aquesta aplicació són els següents:

RQT1 – Disposar d'un dispositiu Android (API 26), tot i que també hi ha la versió per escriptori.

RQT2 – Disposar de plaques ESP32 connectades als fanals.

RQT3 – Disposar de leds RGBW als fanals.

RQT4 – Disposar de connexió a internet (TCP/IP). Tant per a les plaques com per al dispositiu on es troba l'aplicació.

A causa de les dificultats sorgides de la COVID-19 el RQT2 i RQT3 s'han substituït per una emulació utilitzant un conjunt d'ESP8266 que inclouen un led tricolor RGB. Es tornarà al pla original quan sigui possible.

1.4 Estudi de les alternatives de solució

Per a l'**aplicació multiplataforma** les diferents alternatives estudiades són les següents:

Iònic: Eina relativament jove basada en HTML5, CSS i JavaScript, lliure i de codi obert. En ser una eina recent pot ser que hi hagi mòduls o biblioteques existents en altres eines que no estiguin desenvolupats per aquesta solució. No obstant això, és una solució híbrida amb un bon rendiment sempre que no utilitzi grans quantitats de recursos.

Qt: Eina lliure i de codi obert, tot i que ofereix llicències per a la integració en productes privats, basada en C++ tot i que es pot enllaçar amb altres llenguatges com QML. Funciona en les principals plataformes i té un ampli suport. La documentació que aporta l'API és molt completa i permet connexions a diverses bases de dades, permet comunicació via MQTT (Message Queuing Telemetry Transport), permet l'ús d'XML, així com la gestió de fils.

Unity: Eina propietària que necessita una llicència que inicialment pot ser gratuïta. Basada en C# i JavaScript. Proporciona un entorn i un motor gràfic per crear principalment videojocs.

Per a la **base de dades** les diferents alternatives estudiades són les següents:

MySQL: Sistema de gestió de base de dades relacionals, de codi obert i lliure, tot i que ofereix llicències per a la integració en productes privatis. Utilitza el llenguatge SQL, és reconeguda per permetre multi-fils i multiusuaris, a més a més de la velocitat en executar consultes.

InfluxDB: Sistema gestor de base de dades de sèries horàries de codi obert i lliure. Reconeguda per a l'emmagatzematge ràpid i d'alta recuperació de dades de sèries horàries. Molt útil en sectors com l'emmagatzematge de dades de sensors d'Internet de les coses i les representacions en temps real.

MongoDB: Sistema gestor de base de dades orientada a documents, escalable i d'alt rendiment, de codi obert i lliure.

Neo4J: Sistema gestor de base de dades orientades a grafs, de codi obert i lliure, amb llicència per a integracions en productes privatis.

Per al **microprogramari de la placa** les diferents alternatives estudiades són les següents:

Arduino IDE: Aplicació multiplataforma que s'utilitza per escriure i carregar programes en plaques Arduino però també es pot usar en plaques d'altres proveïdors. Llenguatge de programació C i C++.

PlatformIO: Entorn de desenvolupament integrat per a la programació de microcontroladors i sistemes encastats basat en VSCode.

MicroPython: Implementació del llenguatge de programació Python 3 que inclou un petit subconjunt de la biblioteca estàndard de Python i està optimitzat per funcionar amb microcontroladors i en entorns restringits. Els codis no són tan ràpids com en C o C++ i poden ocupar més memòria.

Per al programa executat com a **daemon** les alternatives estudiades són les següents:

Python: Llenguatge d'alt nivell que busca llegibilitat i permet expressar conceptes en poques línies de codi.

C: Llenguatge de mig nivell però amb moltes característiques de baix nivell. Disposa de les estructures típiques dels llenguatges d'alt nivell però, també disposa de construccions del llenguatge que permeten un control a molt baix nivell.

Java: Llenguatge interpretat i orientat a objectes.

1.5 Valoració de les alternatives

Pel que fa a l'aplicació les solucions estudiades eren Ionic, Qt i Unity. Qt ofereix la possibilitat de compilar per moltes més plataformes que Ionic. Qt ofereix una implementació de drag&drop més senzilla que Unity. Durant aquest segon curs de DAM he treballat amb Unity al M08 UF3 amb el propòsit de fer un videojoc. En canvi, he treballat amb Qt en diversos mòduls per realitzar comunicacions MQTT entre altres aplicacions de IoT. Qt és una eina que em resulta més familiar i que domino més per a presentar solucions que es basen a enviar missatges entre dispositius.

Pel que fa a la base de dades, analitzant una mica més a fons el projecte, s'hauran de desar dades de la plaça, dels fanals, de les agrupacions de fanals, de la posició de fanals, entre d'altres. Les dades a desar estan relacionades entre si, per tant necessitem una base de dades relacional.

Pel que fa al microprogramari de les plaques em resulta més familiar l'Arduino IDE eina amb la qual ja he treballat durant els dos anys d'aquest grau. Descarto el MicroPython perquè els microprogramaris no són tan ràpids i ocupen més memòria.

Pel que fa al daemon es necessita una solució senzilla que inclogui facilitats a l'hora de realitzar comunicació MQTT i facilitats a l'hora d'executar consultes a la base de dades relacional. Python proporciona aquestes eines igual que Java. Tanmateix, és molt més senzill i ràpid de desenvolupar en Python.

1.6 Selecció de la solució

Segons la valoració de l'apartat anterior, les decisions preses han sigut desenvolupar l'aplicació mitjançant Qt, desenvolupar el programa en python, desenvolupar el codi per la placa mitjançant Arduino IDE i desar les dades en MySQL. També cal tenir en compte que són eines que el professor ponent domina.

2 Anàlisi del sistema

L'anàlisi del sistema ens permet especificar més a fons com serà el sistema escollit com a solució gràcies a l'estudi de viabilitat.

2.1 Definició del sistema

Després de la valoració i selecció de la solució podem dir que el sistema està definit de la següent manera:

Aplicació: Desenvolupada amb l'eina Qt, utilitzant QML per al frontend i utilitzant C++ per al backend. Es fa ús la biblioteca QMqtt.

Microprogramari per la placa: Desenvolupat amb l'eina Arduino IDE, utilitzant C i C++. S'utilitzen les següents biblioteques: EEPROM, ESP8266WiFi, PubSubClient, Wire i ArduinoJson.

Daemon: Desenvolupat utilitzant Python, es deixarà en marxa en un VPS per tal de poder fer les consultes a la base de dades. S'utilitzen les biblioteques: mysql, json i paho.

Base de dades: Gestionada per MySQL, desenvolupada utilitzant SQL.

La comunicació entre l'aplicació i el microprogramari de la placa es fa mitjançant WiFi i el protocol MQTT mitjançant un broker al núvol. La comunicació entre l'aplicació i el daemon també es realitza amb WiFi i el protocol MQTT mitjançant un broker al núvol. Finalment, la comunicació entre el daemon i la base de dades es realitza de forma directa mitjançant un connector de MySQL. Gairebé tots els missatges transportats utilitzen el format Json.

2.2 Definició d'interfícies d'usuari

La interfície d'usuari per aquest projecte és l'aplicació tot i que les compilacions només s'han fet per a escriptori i Android, Qt permet més opcions de compilat. Com ja he comentat anteriorment, el frontend està desenvolupat amb C++ fent ús de la biblioteca MQTT i el backend està desenvolupat amb QML.

Aquesta aplicació està composta per una pantalla de benvinguda, una configuració de la plaça, els fanals i els grups, posteriorment recuperable de la base de dades i per una selecció de modes i colors. Podem veure el seu funcionament a l'apartat de Manual.

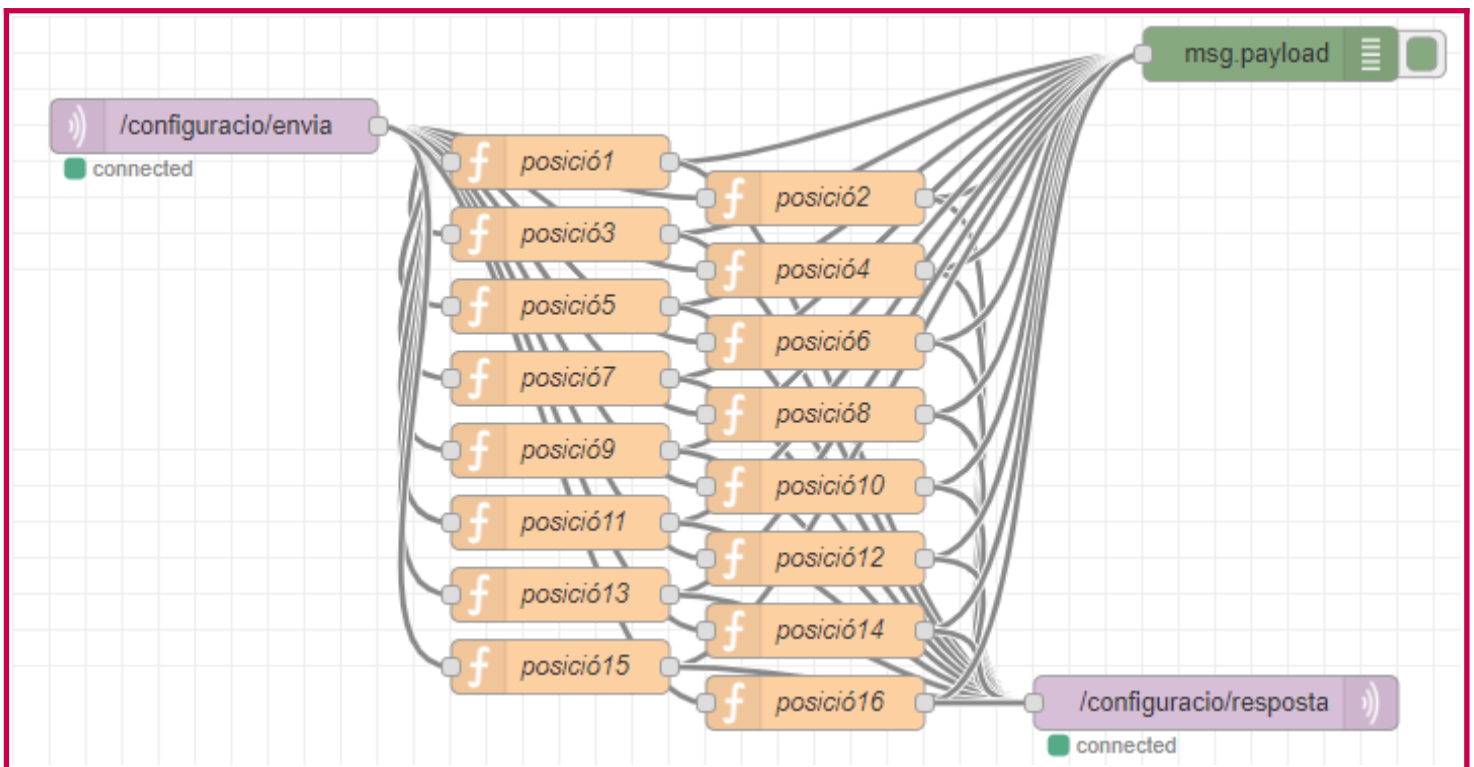
2.3 Especificació del pla de proves

Per aquest projecte s'han realitzat quatre proves fonamentals, a part de les unitàries i d'integració de mòduls de l'aplicació.

Per a realitzar les proves s'ha utilitzat el Node-RED, eina de programari que permet connectar dispositius físics de IoT. També permet fer simulacions.

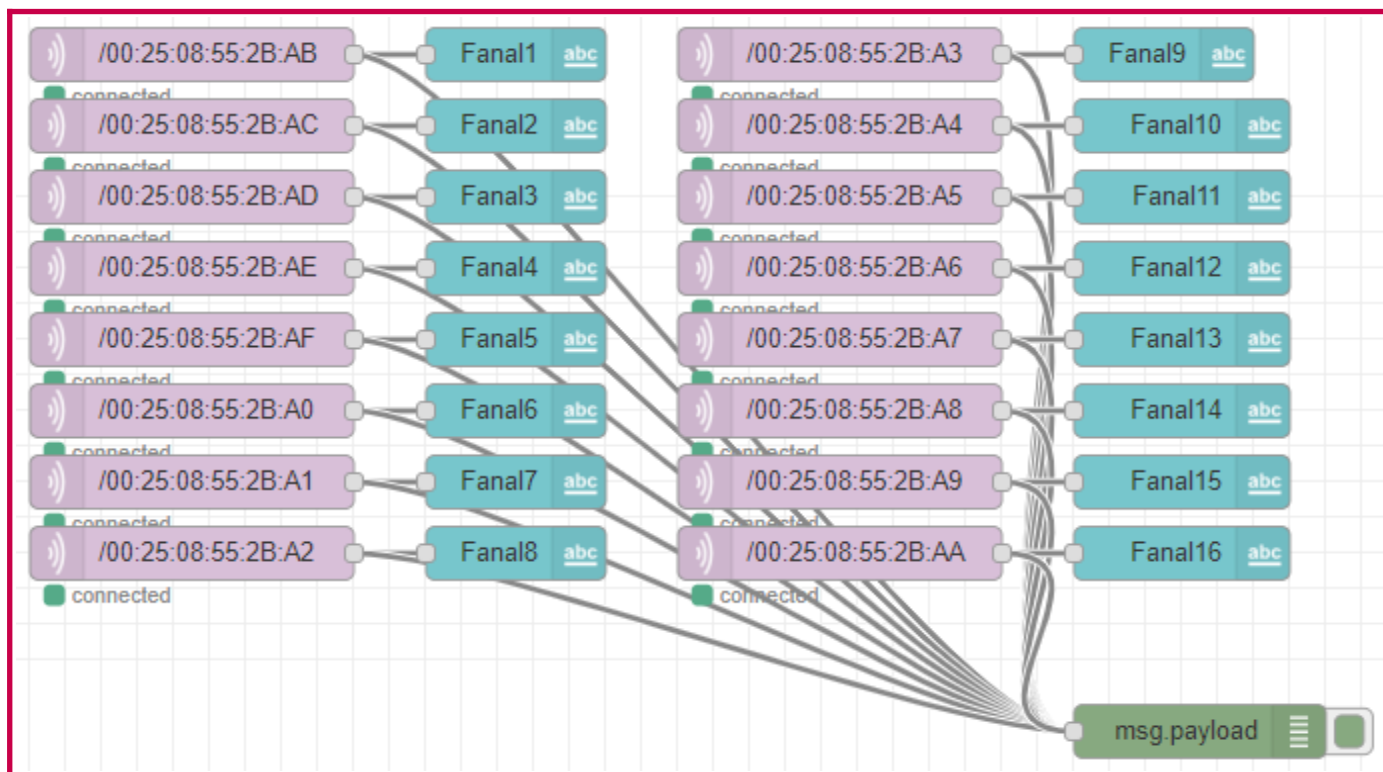
La primera prova s'ha realitzat de l'aplicació contra el Node-RED:

S'han simulat setze plaques des del Node-RED que es comporten com s'haurien de comportar les plaques reals.



En aquesta part el que simulem és l'arribada de la informació per part de l'aplicació i la posterior enviada de les adreces mac per a la configuració dels fanals.

Seguidament simulem la rebuda d'informació per part de les adreces mac, per fer l'assignació del color a cada un d'ells. En aquesta simulació l'enviament del color es feia amb format hexadecimal, tot i que finalment funciona amb format un format Json que conté el RGBW.

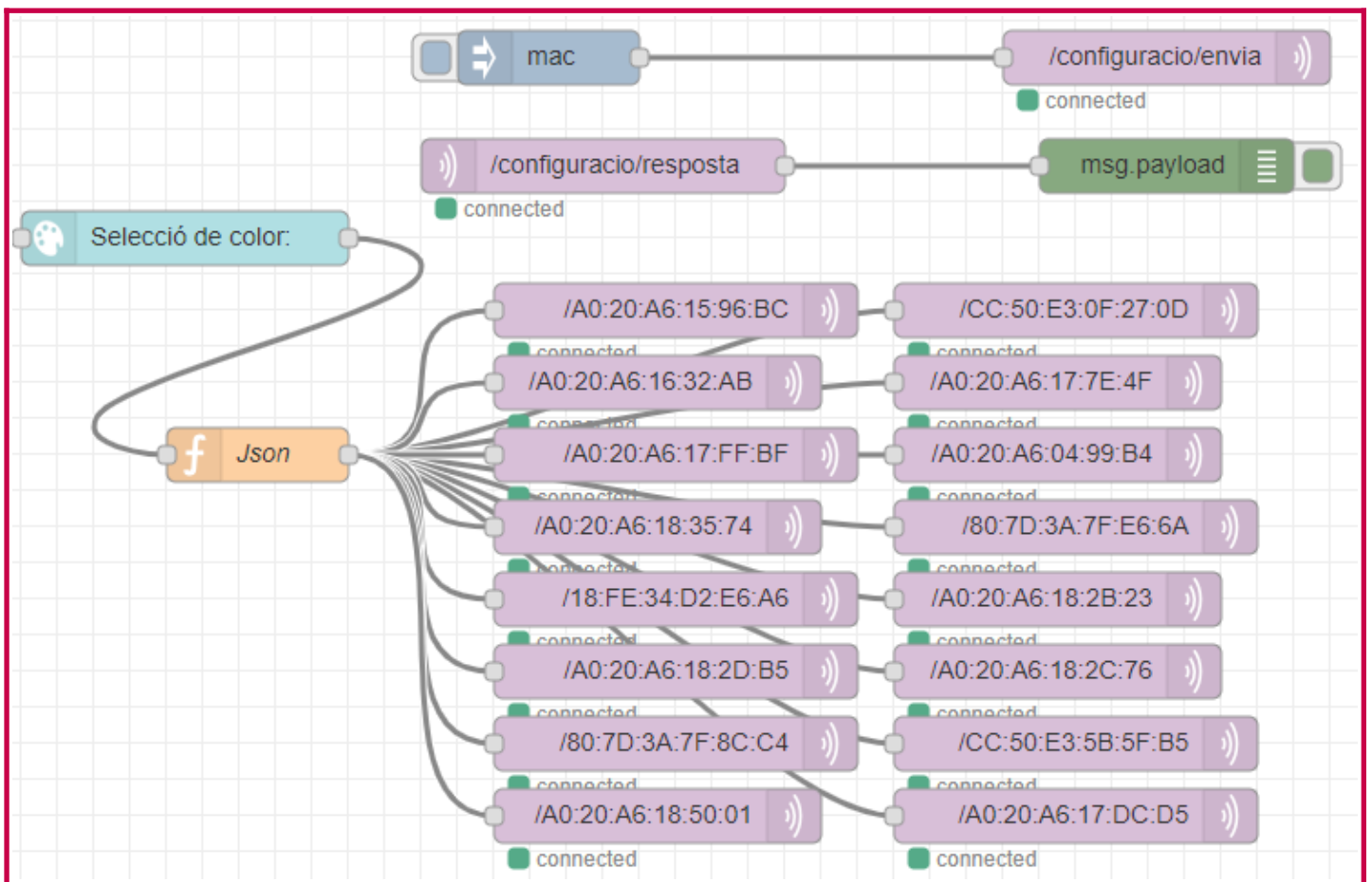


Podem veure la representació dels leds a la part gràfica del Node-RED:

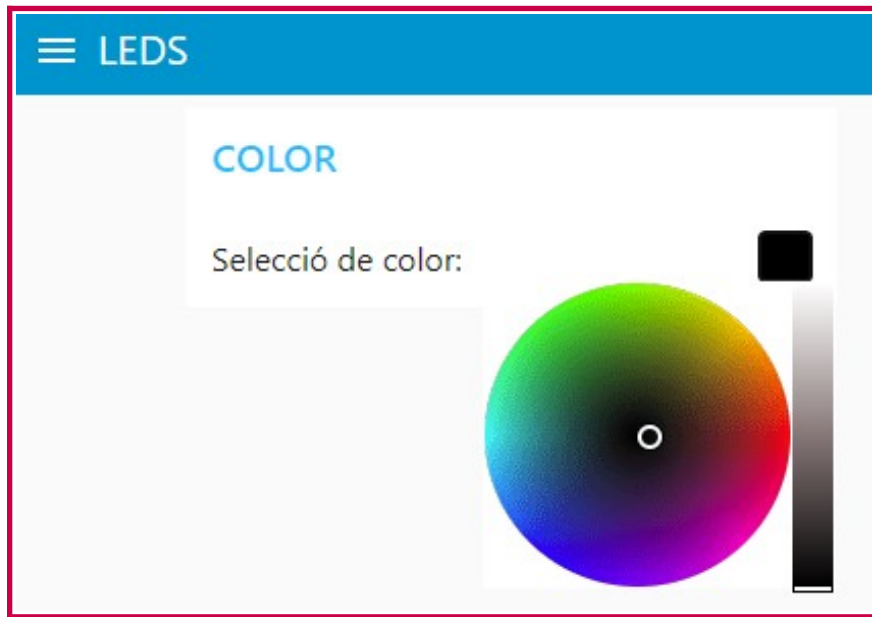
| Fanals | | | |
|---------|---|---------|---|
| Fanals1 | | Fanals2 | |
| Fanal1 | ● | Fanal9 | ● |
| Fanal2 | ● | Fanal10 | ● |
| Fanal3 | ● | Fanal11 | ● |
| Fanal4 | ● | Fanal12 | ● |
| Fanal5 | ● | Fanal13 | ● |
| Fanal6 | ● | Fanal14 | ● |
| Fanal7 | ● | Fanal15 | ● |
| Fanal8 | ● | Fanal16 | ● |

La següent prova que s'ha realitzat ha sigut de la placa contra el Node-RED:

En aquest cas se simula el comportament de l'aplicació per a veure que les plaques responen correctament. Es publica el missatge per tal de rebre les adreces mac i després es configuren cada un dels temes per a cada fanal. Posteriorment s'envia un color per veure el seu funcionament correcte, en aquesta prova ja es treballava amb Json en lloc d'utilitzar el format hexadecimal per al color.



Podem veure a la part gràfica la selecció del color.



Les següents dues proves són ja amb tot el sistema en marxa, una per veure que realment l'aplicació controla les dades i l'altre per veure que tenim accés a les dades emmagatzemades a la base de dades. Aquestes quatre proves les podem veure en els següents enllaços:

Prova1: <https://www.youtube.com/watch?v=j9oEDyuStMc>

Prova2: <https://www.youtube.com/watch?v=-s6f6lt7YWM>

Prova3: <https://www.youtube.com/watch?v=-dBa7hXUzBU>

Prova4: <https://www.youtube.com/watch?v=sJFOTUgb2vQ>

3 Disseny del sistema

En aquest punt del projecte definirem l'arquitectura del maquinari i del programari per a satisfer els requeriments.

3.1 Arquitectura del sistema

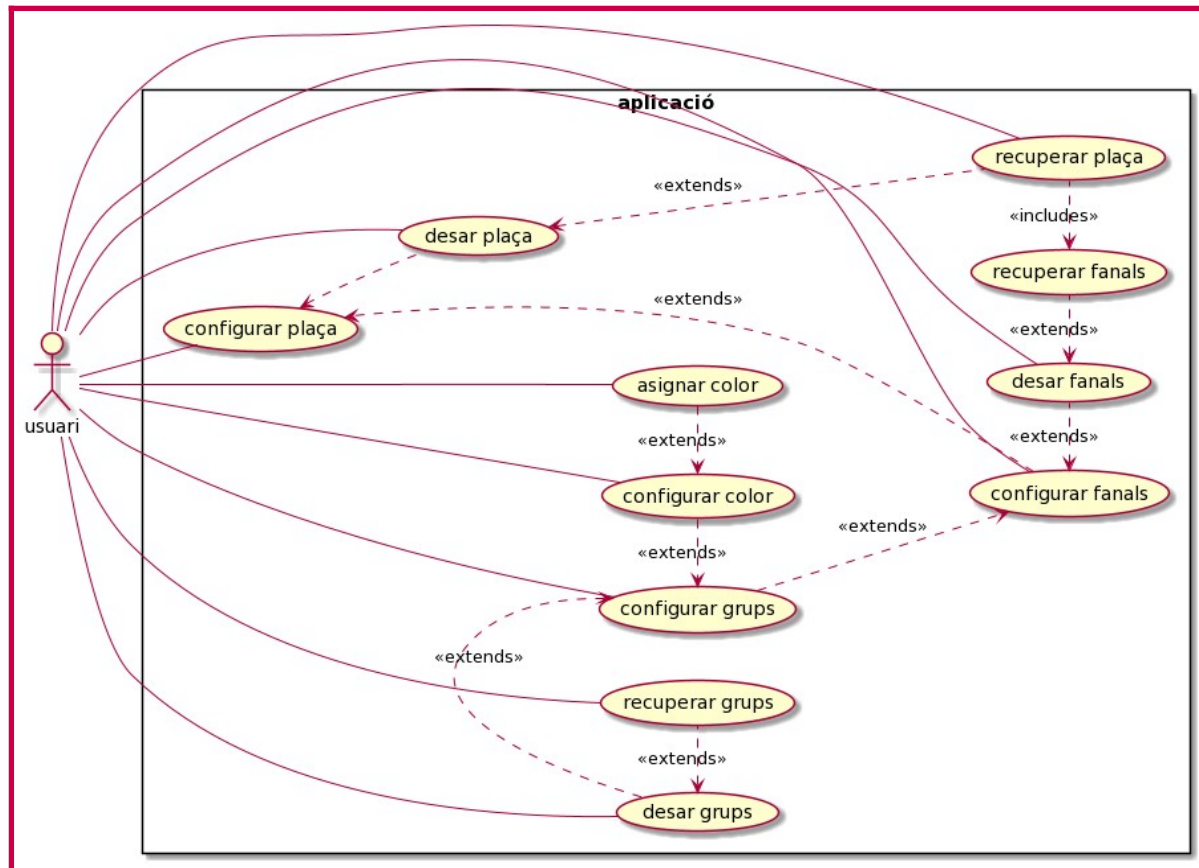
En aquest diagrama es mostra l'arquitectura del sistema que he definit anteriorment. Podem veure que el flux de dades va des de l'aplicació al microprogramari de les plaques o a la base de dades passant per al daemon o a l'inrevés, ja que és bidireccional. El daemon és l'encarregat de rebre les peticions per part de l'aplicació, gestionar aquestes peticions a la base de dades i retornar la informació a l'aplicació.

A la part dreta trobem una bifurcació, a la part superior d'aquesta bifurcació es representa el pla original de controlar els fanals amb l'ESP32 i a la part inferior l'emulació que ha permès desenvolupar la COVID-19, amb l'ESP8266 amb el led tricolor incorporat.



3.2 Revisió de casos d'ús

El següent diagrama mostra els casos d'ús que contempla l'aplicació, aquests casos d'ús fan referència als requisits funcionals demanats.



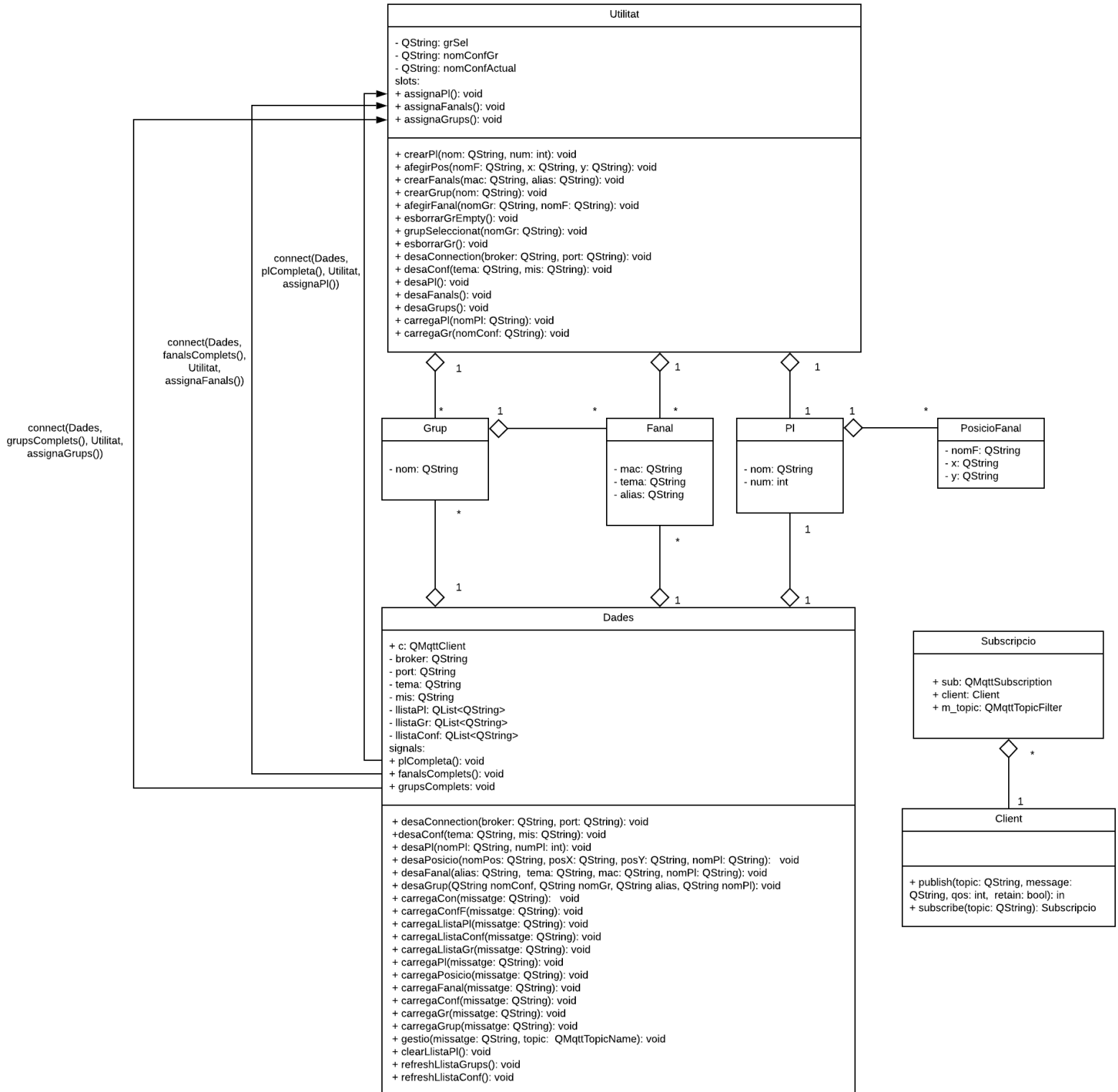
Com podem veure, tots els requisits demanats queden implementats.

3.3 Anàlisi paradigma Orientat a Objectes

L'aplicació desenvolupada es troba dins el paradigma orientat a objectes. Dins d'aquest utilitza una arquitectura de Model-Vista-Controlador. El model és la part de l'aplicació desenvolupada en C++, que s'encarrega de representar les dades. La vista és la part de QML que visualitza l'usuari de l'aplicació. El controlador és una classe de C++ la qual he anomenat Utilitat, que s'encarrega de fer d'intermediari entre la vista i el model, mitjançant peticions i devolucions de dades. En aquest cas no es donen les dades directament del model a la vista, sempre passen pel controlador.

3.4 Disseny paradigma Orientat a Objectes

Podem veure el disseny del paradigma orientat a objectes amb el següent diagrama de classes:



3.5 Persistència de dades: Anàlisi i disseny de bases de dades

La persistència de dades s'ha fet utilitzant una base de dades MySQL, ja que les taules importants estaven relacionades entre si.

A la base de dades volem guardar les dades de la connexió Mqtt per tal de no haver de posar-les cada vegada. Passa el mateix amb les dades de tema i missatge per tant també es volen desar a la base de dades. Aquestes taules són independents.

Es vol desar de la plaça el nom i el nombre de fanals. Una plaça conté posicions, fanals i grups. De la posició es vol desar del valor x, y i el nom de la posició. Dels fanals es vol desar l'àlies, el tema i l'adreça mac. Dels grups es vol desar el nom de la configuració, el nom del grup i el nom del fanal que hi pertany. Els noms de les places no es poden repetir. Cada fanal pertany a una plaça però els àlies es poden repetir en diferents places. Cada grup pertany a una plaça però els noms dels grups i de la configuració es poden repetir per a diferents places. Cada posició pertany a una plaça però els noms de les posicions es poden repetir en diferents places.

Diagrama entitat-relació:

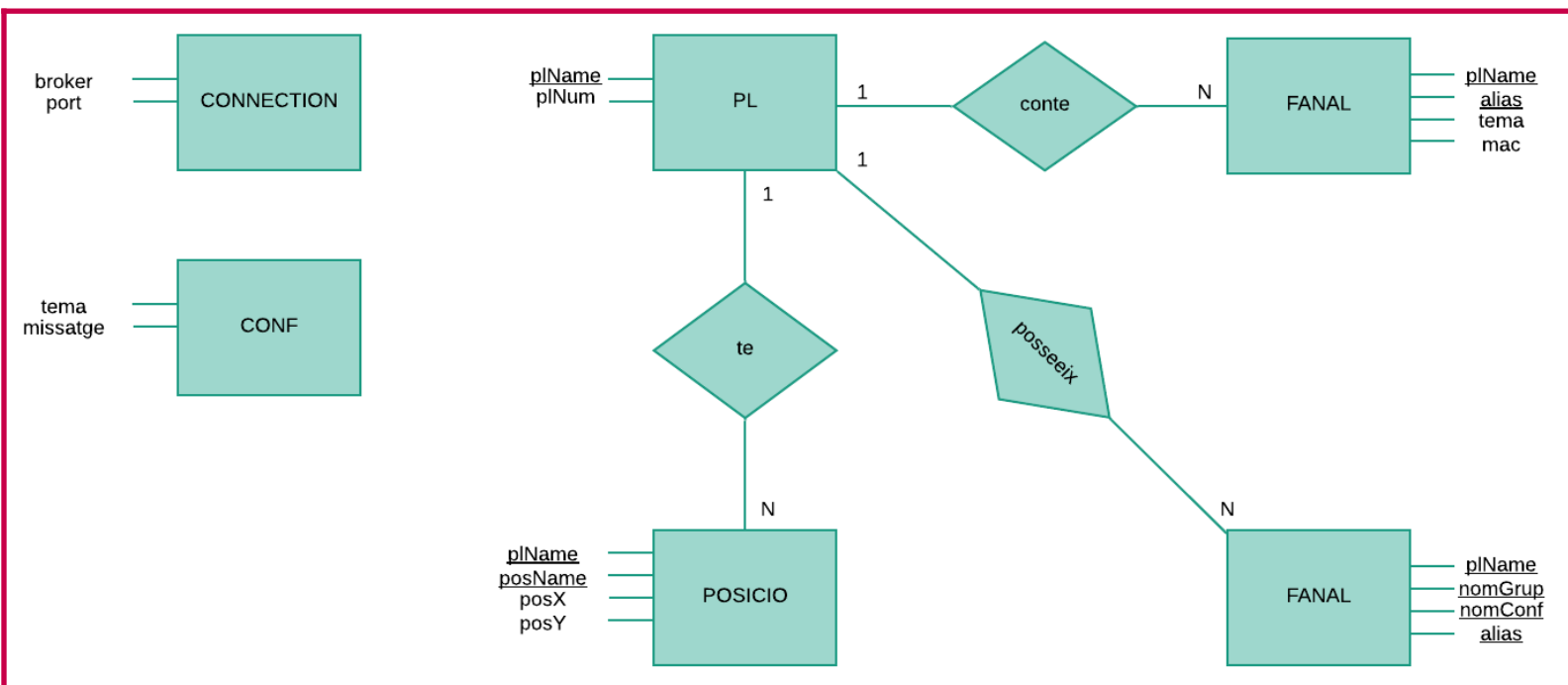
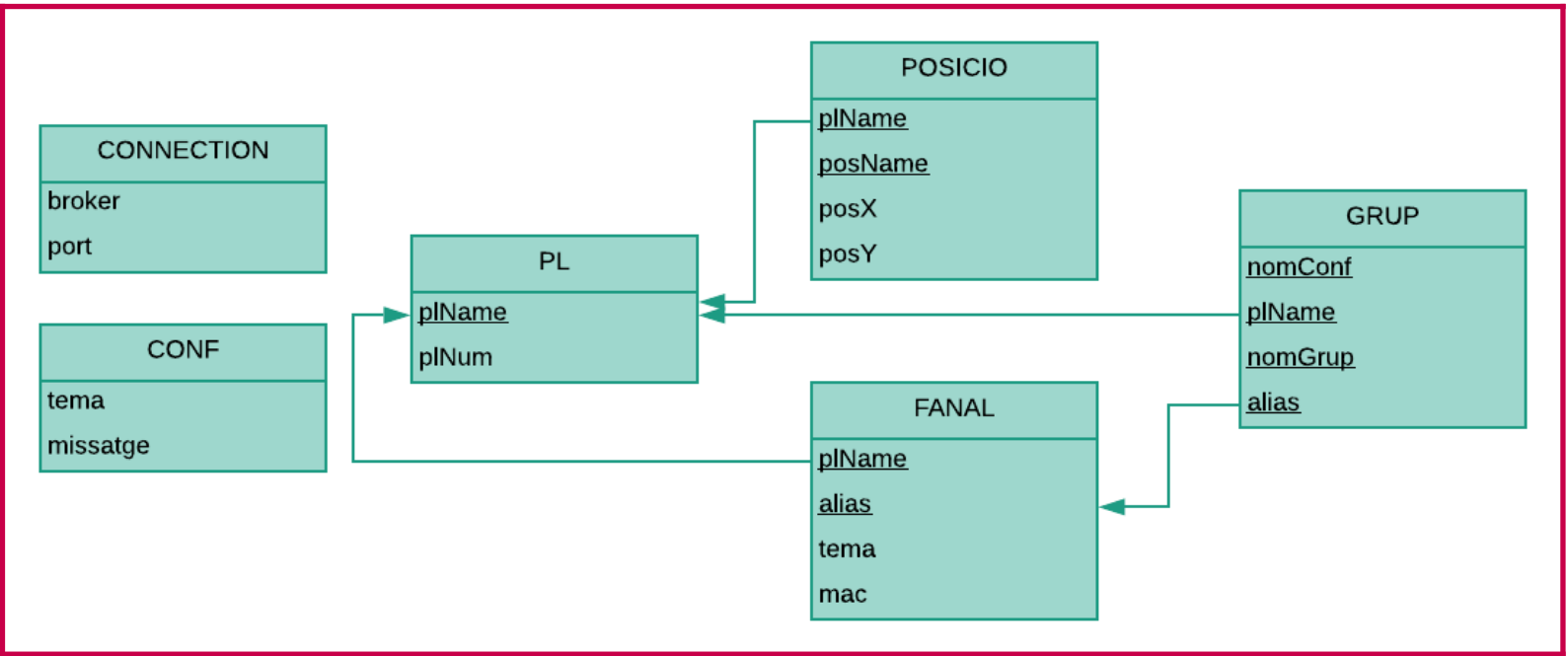


Diagrama relacional:

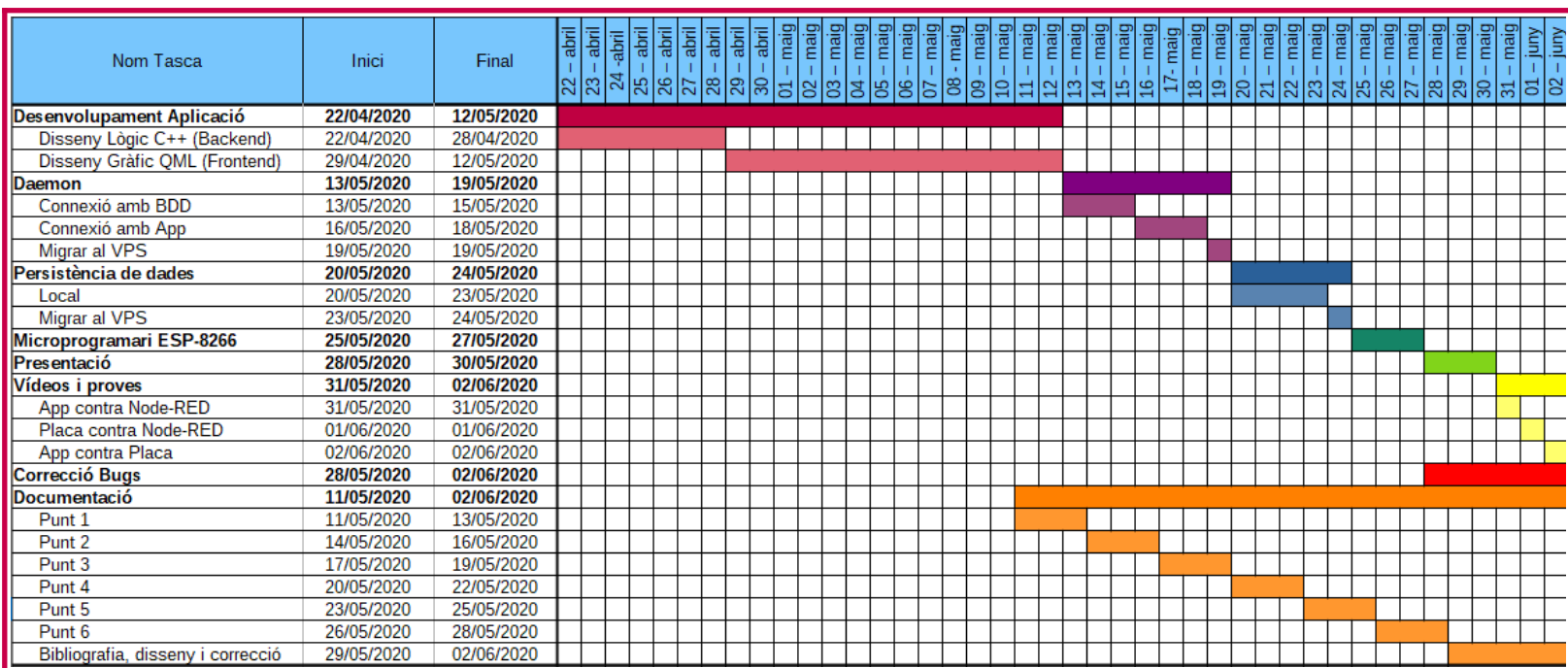


4 Desenvolupament

Aquest projecte s'ha desenvolupat amb quaranta-dos dies, podem veure el procés de desenvolupament a continuació.

4.1 Planificació de les activitats de desenvolupament i integració de sistema

Diagrama de Gantt on podem veure aquesta planificació:



4.2 Desenvolupament

Aquest projecte s'ha iniciat amb dues proves de concepte. La primera es basa en una aplicació bàsica per a la modificació d'un sol ESP8266.

La segona prova de concepte es basa en un control remot dos leds connectats a una Raspberry Pi mitjançant un túnel SSH. Es poden fer fotografies amb la càmera web, com podem veure a la imatge.



Seguidament s'ha desenvolupat l'aplicació per a controlar ESP32 i leds de quatre colors RGBW, però en veure que el material no arribava a temps s'ha pres la decisió de fer una emulació amb ESP8266, que només té leds tricolor. Tot i que l'aplicació està preparada per a quan arribi el material que es pugui fer el canvi.

En acabar l'aplicació s'ha dut a terme la persistència de dades a la base de dades escollida. En veure que MySQL no permetia l'accés directe remotament amb facilitat, s'ha optat per a desenvolupar en python un daemon, d'aquesta manera s'accedeix remotament al daemon que és qui fa l'accés a la base de dades i retorna la informació necessària. Primer s'han fet proves en local i després s'ha fet la implementació al VPS.

A continuació, s'ha acabat d'enllestir el microprogramari per a les ESP8266 que s'havia utilitzat per a la primera prova de concepte.

A mesura que s'han anat acabant els diferents mòduls s'han integrat mitjançant proves unitàries i d'integració.

Finalment s'ha obtingut una aplicació que compleix amb tots els requeriments proposats.

5 Implantació

La implantació es durà a terme quan arribi el material. Passant d'ESP8266 a ESP32 mantenint els temes de subscripció i publicació. L'aplicació, la base de dades i el daemon es mantindran. Només s'haurà de migrar el microprogramari perquè funcioni amb les noves condicions.

5.1 Formació

Està planejat desenvolupar una pàgina d'estil Viquipèdia que incorpori els vídeos i el manual. A més a més es vol afegir alguna explicació d'alguna part del software.

5.2 Implantació del sistema

Aquest projecte encara no està implantat però la seva implantació constarà de la fabricació de les plaques amb proves a fàbrica per saber que funciona tot correctament. Seguidament es farà la instal·lació dels drivers i la instal·lació del microprogramari a les plaques, el muntatge dels diferents components físics com els leds, els dissipadors i les lents. A continuació, el muntatge dels fanals als màstils i la posada en marxa del sistema. Finalment es faran proves per veure que tot funciona correctament. La implantació serà similar a la que va fer Siarq a Sant Pere de Ribes, que podem veure a la imatge.



6 Manteniment i versions futures

La primera versió a realitzar és deixar l'emulació amb ESP8266 per a realitzar una migració de microprogramari per a l'ESP32. Les principals diferències entre l'emulació amb ESP8266 i la realitat amb ESP32, que l'ESP32 utilitza el BUS-I2C per controlar un xip que fa de potenciòmetre analògic per *dimeritzar* (control del nivell de llum per cada un dels quatre canals, RGBW) i utilitzarà controladors de nivells de llum de MeanWell controlats pel xip amb el potenciòmetre analògic per al funcionament dels leds.

La següent versió i pot ser una de les més importants és generar certificats per a passar de Mqtt a Mqtts i fer la comunicació segura.

Una versió interessant de fer seria que si les plaques perden la connexió o l'alimentació, quan tornin a posar-se en marxa sàpiguen quin era l'últim color que tenien, sempre que no sigui en mode espectacle. Per tant, s'hauria de guardar el color a la memòria flaix (SPIFFS) si aquest està en qualsevol altre mode i no canvia en un període de temps considerable, ja que la memòria flaix té un nombre limitat d'escriptures, i il·limitat de lectures; i no se'n pot abusar. Permet de cent mil a un milió d'escriptures.

Pressupost

En aquest pressupost no es contemplen les hores civils de muntatge, ni els màstils dels fanals, ja que en aquest cas es poden reutilitzar els que ja estan instal·lats, si no es poguessin utilitzar, s'haurien d'afegir al pressupost.

| MATERIAL LEDS QUANTITAT | NOM | # PROJECTORS | POTÈNCIA | TOTAL (W) | TOTAL LEDS x PROJECTOR | TOTAL LEDS INSTAL·LATS |
|----------------------------|-----------|--------------|-------------------|-----------|---------------------------|---------------------------|
| 3 | FUL 12 | 4 | 31W | 124 | 12 | 36 |
| 11 | FUL 9 (A) | 3 | 31W | 93 | 9 | 99 |
| 1 | FUL 9 (B) | 3 | 86W (1) + 31W (2) | 148 | 15 | 15 |
| 1 | FUL 5 | 2 | 31W | 62 | 6 | 6 |

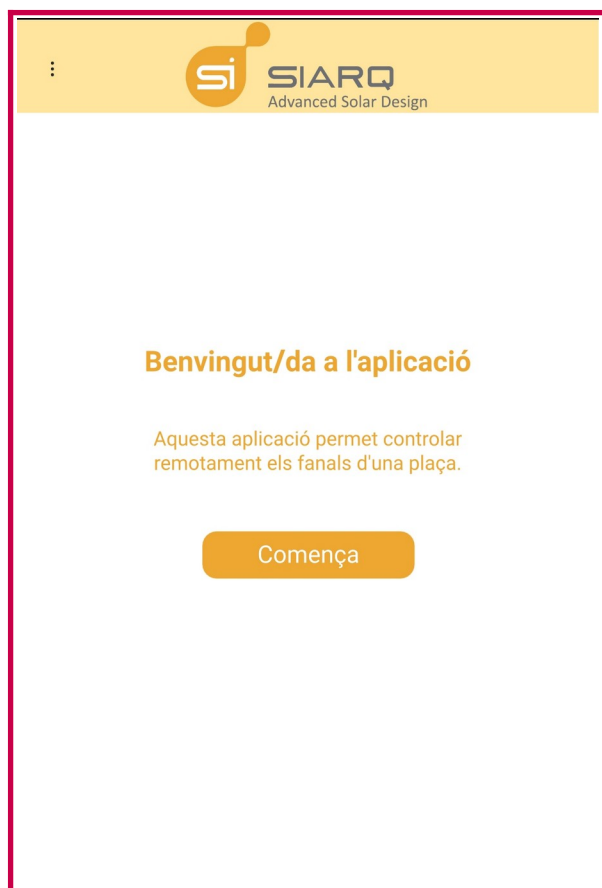
TOTAL 156
Preu Unitari 8,00 €
Preu Total 1.248,00 €

| ALTRES MATERIALS | NOM | PREU UNITAT | TOTAL |
|------------------|---|-------------|------------|
| 42 | Lent Òptica | 11,90 € | 499,80 € |
| 42 | Dissipador d'Alumini | 10,00 € | 420,00 € |
| 64 | LED Driver Mean Well ODCL-45-350 | 25,00 € | 1.600,00 € |
| 16 | Placa ESP32 | 3,00 € | 48,00 € |
| 42 | Carcasa mòdul LED | 171,00 € | 7.182,00 € |
| 16 | Cable connexions LEDS | 35,00 € | 560,00 € |
| 16 | Cable connexions elèctriques | 35,00 € | 560,00 € |
| 99 | Hores ma d'obra | 35,00 € | 3.465,00 € |
| 1 | Router Comfast 300 Mbs 802.11AC per exteriors | 45,90 € | 45,90 € |

TOTAL ALTRES MATERIALS 14.380,70 €
TOTAL LEDS + ALTRES MATERIALS 15.628,70 €

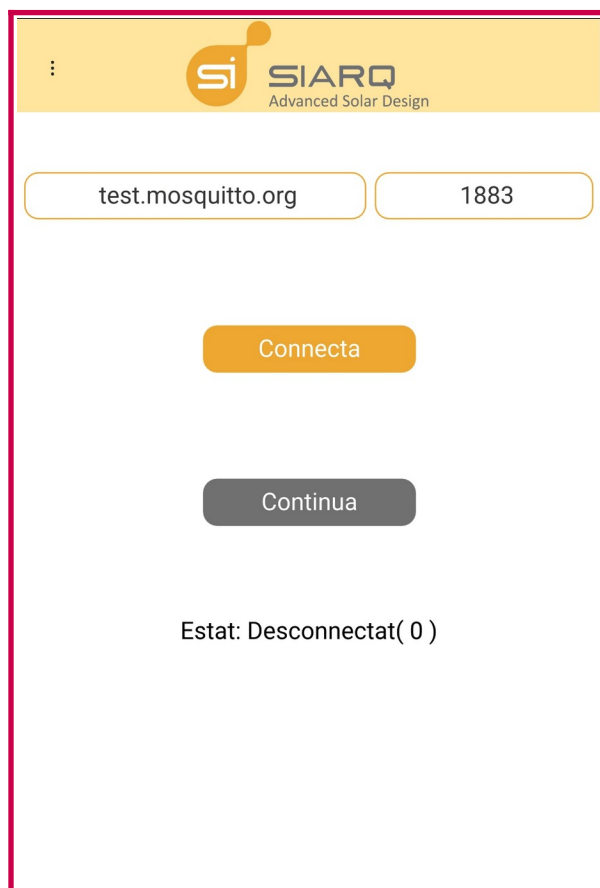
Manual d'ús

Només hi ha un manual d'ús per a tota mena d'usuari, explicant la utilització de l'aplicació:



Pantalla de benvinguda:

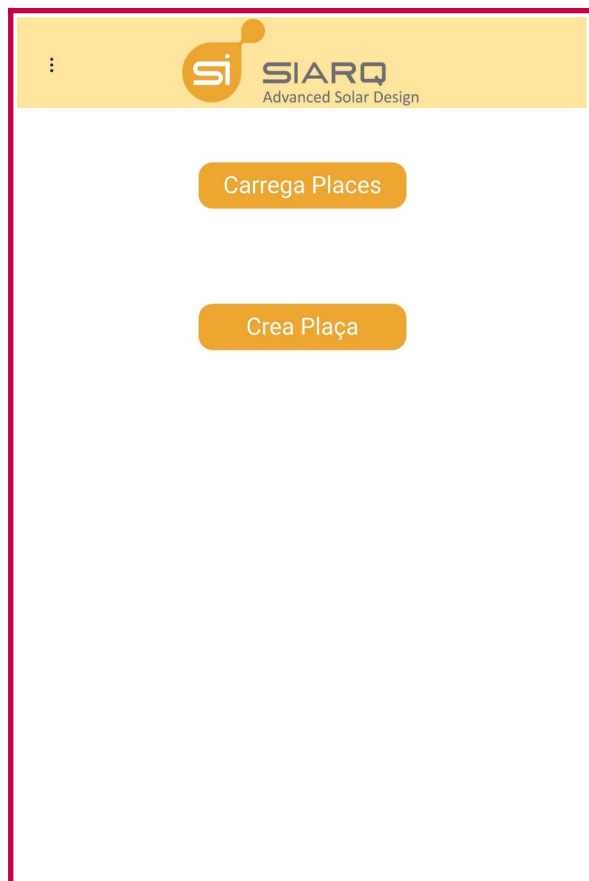
Prémer el botó Comença.



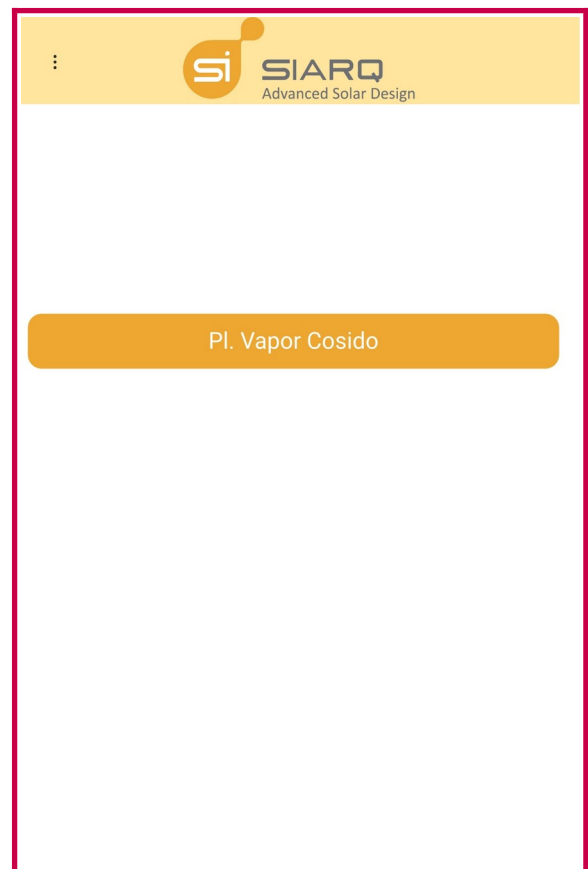
Pantalla de connexió:

Prémer el botó Connecta, quan s'hagi connectat s'habilitarà el botó Continua.

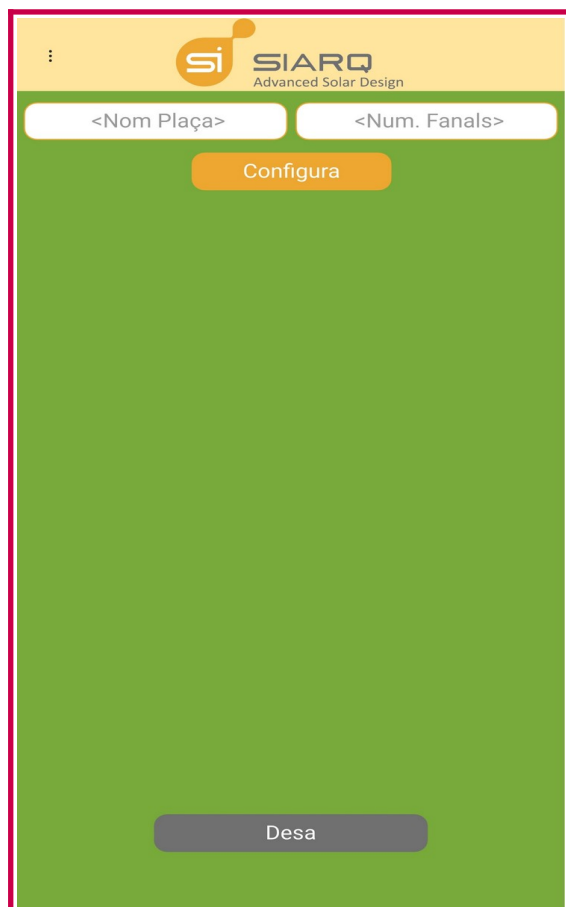
Prémer el botó Continua.



Pantalla que permet dues opcions.



Primera opció: Pantalla que ens deixa visualitzar i carregar les places desades a la base de dades.

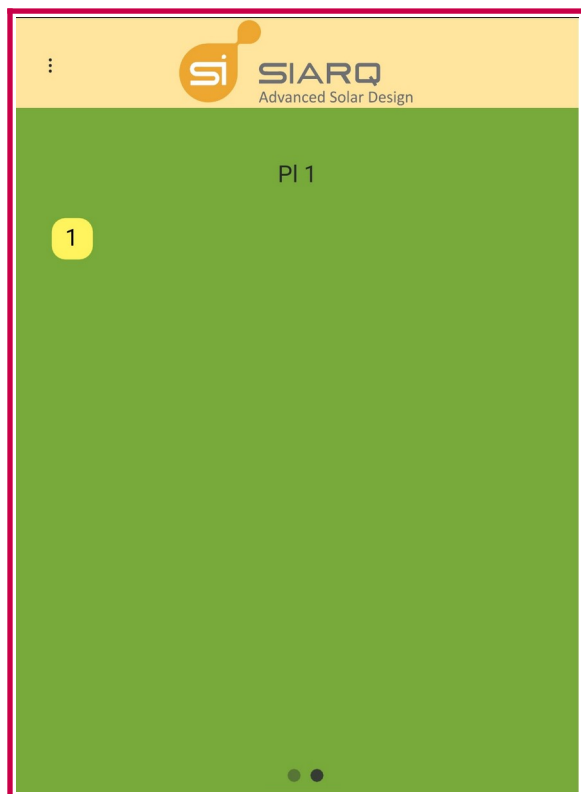


Segona opció: Pantalla que permet la configuració de la plaça, posant el nom, el nombre de fanals i prement el botó Configura, apareixeran els fanals en dues files als laterals, fent drag&drop els col·loquem en la posició desitjada. Premem el botó Desa. Veure el vídeo de configuració plaça.

Pantalla que apareix després de crear la plaça.

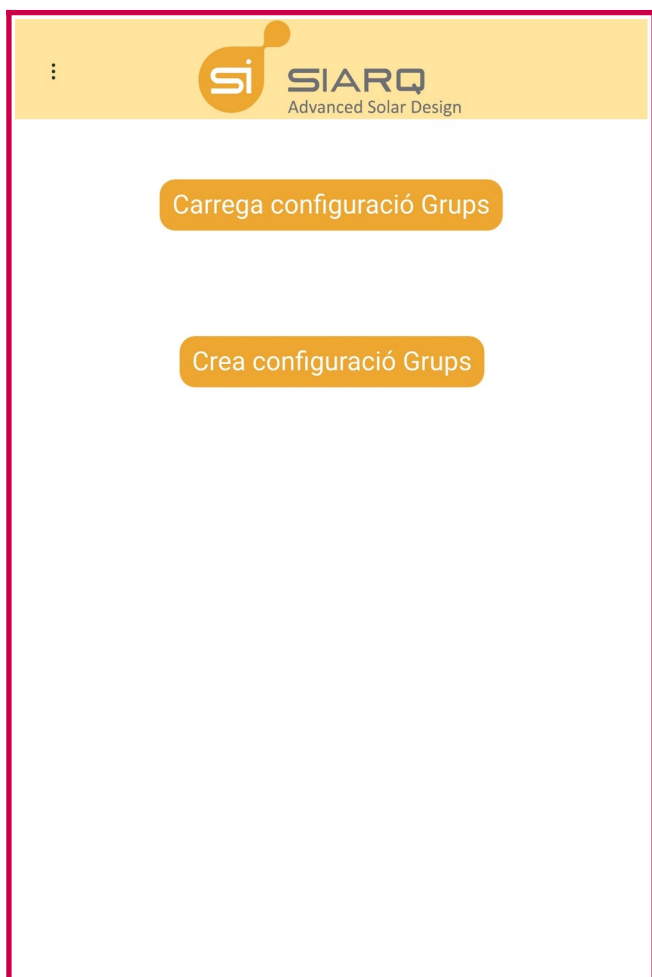
Si premem el botó Publica, apareixeran les adreces mac i els alies per als fanals. El botó Encén permetrà encendre el led del fanal seleccionat i apagar-lo per fer la corresponent assignació dels fanals i les posicions.

Veure el vídeo de configuració fanals.

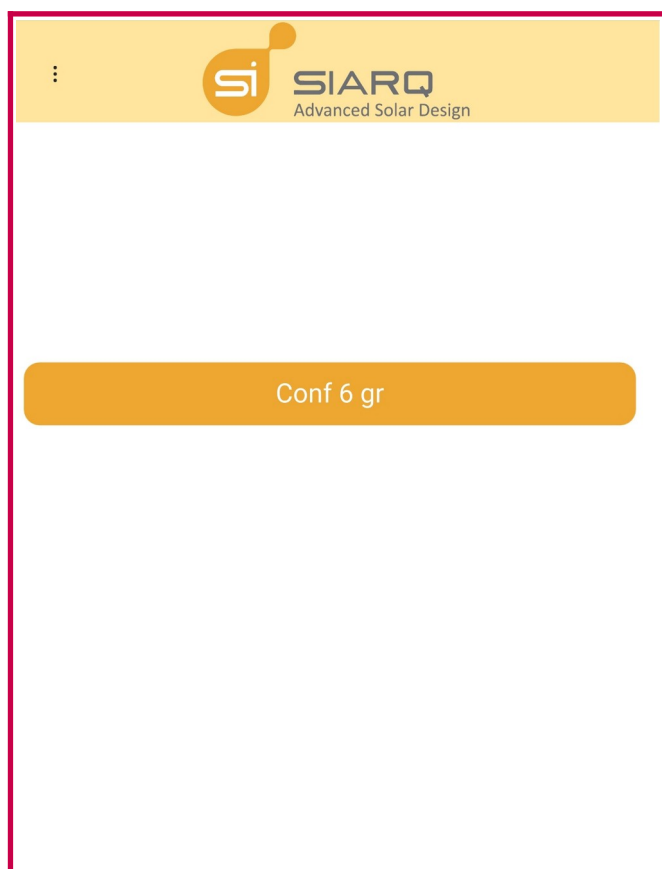


Aquesta pantalla la trobem desplaçant l'anterior a l'esquerra, ens permet consultat la posició de cada fanal a la plaça per fer una assignació correcta d'aquests.

Per exemple, si la plaça té vint fanals quan els configuri davant de la plaça i n'encengui un podré veure consultat el mapa, de quina posició es tracta i assignar l'alties a la posició corresponent.



Per als dos camins anteriors arribem a aquesta pantalla amb dues opcions.



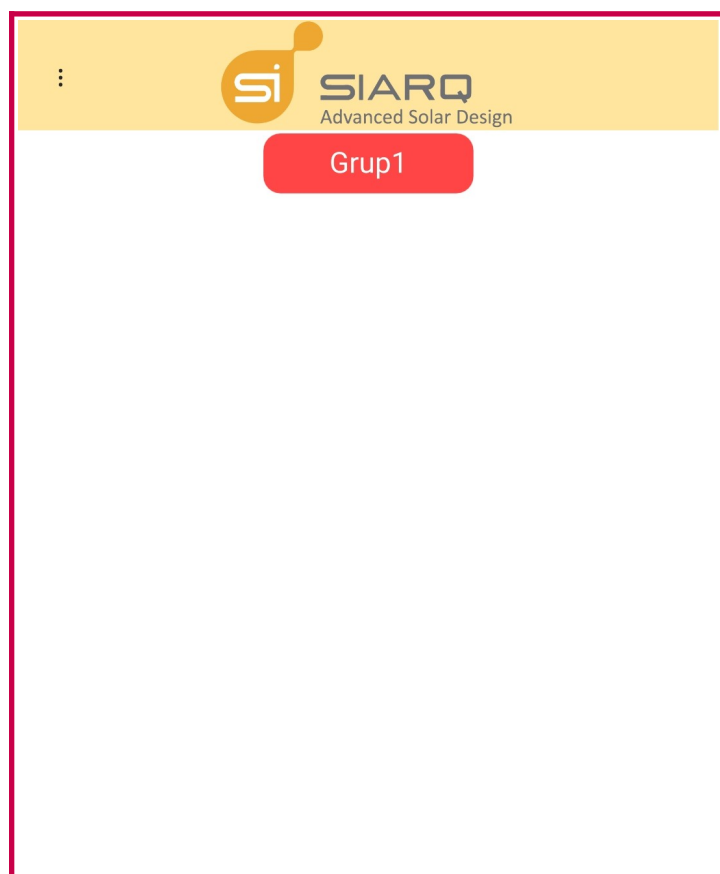
Primera opció: Pantalla que ens deixa a visualitzar i carregar les dades de la configuració de grups desades a la base de dades.



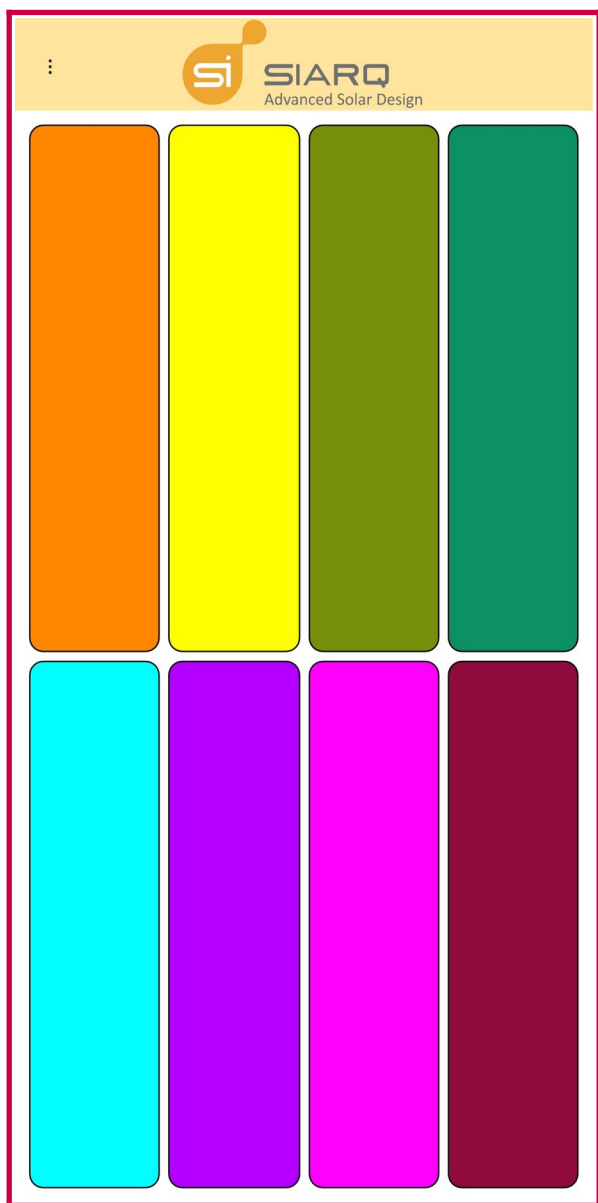
Segona opció: Pantalla que permet configurar els grups. També es pot desplaçant cap a l'esquerra per veure el mapa a l'hora d'agrupar els fanals. S'ha de posar un nom a la configuració, l'assignació de cada fanal a cada grup es pot consultar desplegant assignació. Veure vídeo de configuració grups.



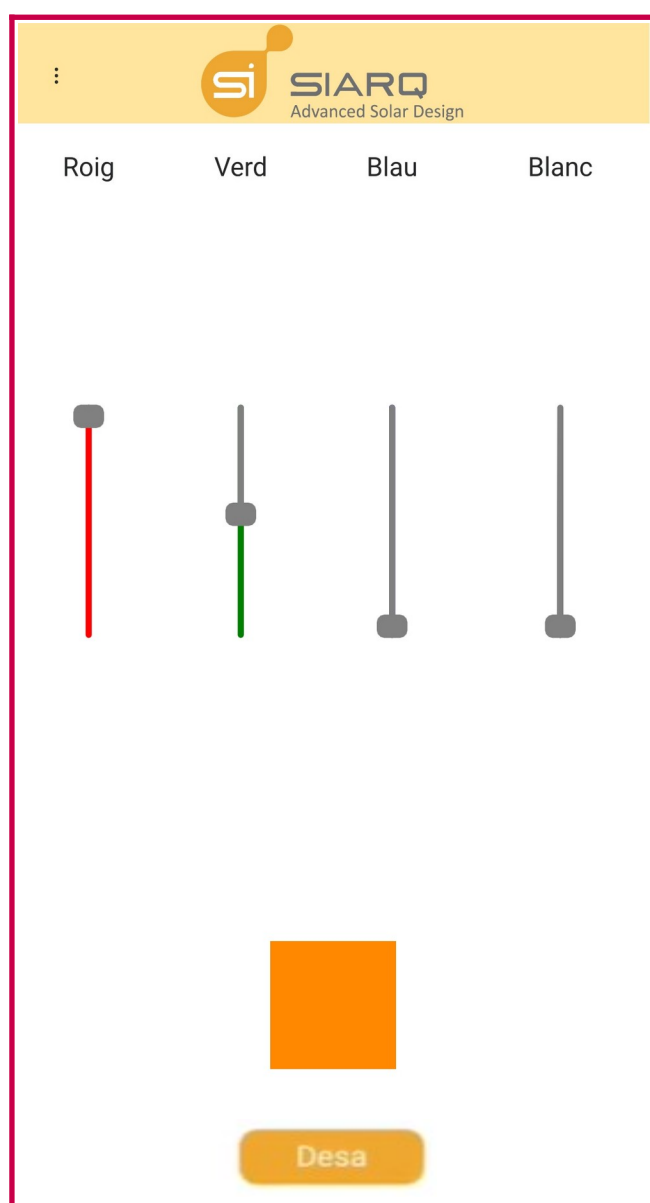
Per als dos camins anteriors arribem a aquesta pantalla, si premem qualsevol botó els fanals es posaran en el Mode desitjat. Veure el vídeo de configuració color.



Pantalla que apareix en prémer el MODE tria color. Aquesta pantalla ens permet modificar el grup al qual li volem modificar el color.



Pantalla que apareix en seleccionar un grup.



Pantalla que apareix en seleccionar un color.

Podem escollir el color desitjat. En fer-ho, apareixerà el color seleccionat a tots els fanals del grup.

Per a visualitzar millor aquesta explicació hi ha quatre vídeos de configuració que trobareu en els següents enllaços:

Configuració plaça: <https://www.youtube.com/watch?v=eDZWc9kQX00>

Configuració fanals: <https://www.youtube.com/watch?v=eqS8T5aqnGU>

Configuració grups: <https://www.youtube.com/watch?v=H8OYBFM7oAg>

Configuració colors: <https://www.youtube.com/watch?v=4FRPinFZqZE>

Informació específica, enllaços

- [Qt](#)
- [MySQL](#)
- <https://www.w3schools.com/>
- [Informació de la plaça](#)
- [Codi aplicació](#)
- [Codis Node-RED](#)
- [Codi ESP8266](#)
- [Codi daemon](#)