

# CONSTRUCCIÓ D'UN LLUM CONTROLAT A TRAVÉS DE DISCORD AMB Node-RED



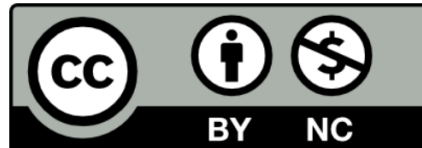
**Nom:** Matias Molina Luquez  
**Tutor:** Jordi Orts  
**Grup:** 2n Batxillerat B  
**Curs acadèmic:** 2022/2023  
Institut Príncep De Viana

## **Construcció d'un llum controlat a través de discord amb Node-Red**

Matias Molina Luquez

1a edició

Gener 2023



Aquest treball està subjecte a la llicència de Reconeixement-NoComercial 3.0 No adaptada Creative Commons. Per veure una còpia de la llicència, visiteu <http://creativecommons.org/licenses/bync/3.0/>.

# Índex

<b>1. INTRODUCCIÓ</b>	<b>4</b>
1.1. Raons d'elecció	4
1.2. Objectius del projecte i estructura	4
<b>2.DISCORD I LA SEVA FACILITAT PER PROGRAMAR</b>	<b>6</b>
2.1.Què és?	6
<b>3.ARDUINO</b>	<b>7</b>
<b>4.NODE-RED</b>	<b>8</b>
4.1.Què és?	8
4.2.Taula amb diferents fluxos	9
<b>5.MQTT</b>	<b>10</b>
5.1.Què és?	10
<b>6.COMPONENTS</b>	<b>11</b>
6.1.Wemos D1 mini	11
6.2.Relay Shield	12
6.3.Button Shield	12
6.4.Buzzer Shield	12
6.5.RGB Shield	13
6.6.Tripler Base	13
<b>7.DISSENY</b>	<b>14</b>
7.1.Diagrama de blocs	14
7.2.Esquemes	15
7.2.1. Relay Shield	15
7.2.3. Buzzer Shield	15
7.3.Disseny de la base de la capsa	16
7.3.1. Disseny tècnic de la base de la làmpada	18
7.4.Disseny de les peces de metacrilat	19
7.4.1. Disseny tècnic de les peces de metacrilat	20
<b>8.SOFTWARE</b>	<b>22</b>
8.1.Arduino i MQTT	23
8.2.Node-RED	27
<b>9. PLANIFICACIÓ</b>	<b>29</b>
<b>10.PRESSUPOST</b>	<b>31</b>
<b>11.CONCLUSIONS</b>	<b>32</b>
11.1.Avaluació del projecte i possibles millores	32
11.2.Problemes	32

---

11.3. Altres aplicacions	33
11.4. Valoració personal	34
<b>AGRAÏMENTS</b>	<b>35</b>
<b>12. ANNEXOS</b>	<b>36</b>
A. ANNEX A. Creació d'un servidor	36
B. ANNEX B. Creació d'un bot	38
C. ANNEX C. Instalar Node-RED en Ubuntu 20.04	41
D. ANNEX D. Instalar MQTT Mosquitto Broker en Ubuntu 20.04	43
E. ANNEX E. Configuració de l'entorn Arduino IDE	45
F. ANNEX F. Plantilla peces de metacrilat a mida real	46
G. ANNEX G. Codi exemple relé	47
H. ANNEX H. Codi exemple polsador	47
I. ANNEX I. Codi exemple brunzidor	48
J. ANNEX J. Codi exemple led RGB	48
K. ANNEX K. Codi Arduino	49
L. ANNEX L. Codi node-red	58
M. ANNEX M. Creació i configuració de fluxos	64
N. ANNEX N. Taula de freqüències de les notes utilitzades (Hz)	66
O. ANNEX O. Codi notes.h	67
<b>13. BIBLIOGRAFIA</b>	<b>68</b>
13.1. Fonts consultades	68
13.2. Programari utilitzat	69

# 1. INTRODUCCIÓ

## 1.1. Raons d'elecció

En l'elecció de l'àmbit del meu treball de recerca, igual que la modalitat de batxillerat, no vaig tenir cap dubte en triar el tecnològic, ja que a més de ser una assignatura que des de petit m'ha interessat molt, enfoca al camp que vull estudiar i treballar en el futur.

La idea d'escollir un microcontrolador que es pogués controlar a través de discord, va sorgir en quart d'ESO, quan entre amics van crear una comunitat d'aquesta, dedicada a videojocs. Llavors, és quan ens vam interessar en la creació dels bots dins d'aquesta plataforma.

Justament, en les classes de tecnologia, estàvem treballant amb l'Arduino UNO, per això se'm va acudir comentar al professor de tecnologia si era possible fer aquesta idea. Qüestió a la qual em va respondre positivament i em va afegir que podria ser un bon treball de recerca.

## 1.2. Objectius del projecte i estructura

Des d'aquell dia, ja tenia clar l'objectiu principal del meu treball, aconseguir la creació d'un tipus de sistema que permet a l'usuari interactuar amb un bot de discord i que aquest pugui enllaçar-se amb un microcontrolador, en el meu cas, el Wemos D1 mini, però encara em faltava trobar una aplicació real.

En donar-li voltes em va passar pel cap fer un sistema per a control de persones majors (amb sensor de caigudes, GPS...), que automàticament avisi en un grup de discord amb els familiars on també puguin interaccionar amb el bot, per obtenir informació de la localització, temperatura... Tot i que a mesura que anava fent el BOTDUINO (Bot- = discord bot, -duino = arduino) em vaig adonar que, encara que discord és molt senzill i fàcil d'utilitzar, aquesta aplicació és utilitzada més aviat per a adolescents, que per a persones adultes i per això vaig voler continuar amb el mateix esquema del projecte, però en un àmbit diferent.

Tenint en compte que els joves que emprenen discord, normalment és més aviat per comunicar-se alhora que juguen a videojocs (per això molts equips competitiu d'esports l'utilitzen) em vaig adonar que hauria d'orientar el meu projecte per aquests usuaris. Per això conversant amb el meu professor vaig decidir-me en fer un circuit senzill amb un RGB i un brunzidor a sota d'una làmpada transparent, feta per fusta, de manera que quan un jugador vulgui avisar als altres, executi una comanda, perquè el LED s'encengui d'un color o d'un altre. Per exemple, si volen connectar-se perquè hi ha una competició podem posar "/partit" i que la làmpada s'encengui de color verd i comenci a sonar una alarma, per tal que tots els participants es connectin a l'hora.

He organitzat aquest projecte en tres parts: una part amb teoria on explico conceptes bàsics per poder dur a terme el meu projecte i del marc pràctic tenim una part de disseny, hardware... i una altra de software.

A l'hora de realitzar el treball em vaig proposar els següents objectius:

1. Instal·lar una plataforma capaç d'enllaçar el meu microcontrolador amb discord.
2. Trobar un protocol que m'ajudi a connectar aquesta plataforma amb el microcontrolador.
3. Trobar un dispositiu de hosting per a aquesta plataforma
4. Configurar la meua placa per actuar d'una manera o altra segons el missatge que rep.
5. Crear una integració de discord adequada per aquest projecte
6. Trobar algun mètode o llibreria per poder fer ús d'aquesta integració.
7. Dissenyar una làmpada a mida per a la Wemos D1 mini
8. Fer la làmpada
9. Realitzar una bona memòria, útil i entenedora per als lectors.

## 2.DISCORD I LA SEVA FACILITAT PER PROGRAMAR

### 2.1.Què és?

Discord s'ha convertit en una plataforma fidel amb servei de missatgeria gratuït entre joves, especialment gamers, la qual permet comunicar-se per xat de veu, text i fins i tot vídeo mitjançant grups, comunitats, xats privats de la temàtica que més interessa els quals pots crear o unir-te dins de l'aplicació.

Aquest interès per la comunitat gamer ha despertat per motius varis, entre ells la seva versatilitat i accessibilitat, ja que és executable tant al navegador com en aplicació de mòbil o ordinador i a més la poca utilització de recursos que implica una millor efectivitat quan s'executa per parlar amb els teus companys mentre jugues a un videojoc.

Com he esmentat anteriorment, he escollit discord per diversos motius, però sobretot perquè ja estava una mica familiaritzat amb la plataforma i la voluntat d'ampliar els meus coneixements i la de millorar l'experiència dels usuaris de la meva comunitat. Malgrat la meva decisió, cal aclarir que no es pogués haver fet amb qualsevol servei de missatgeria, pel fet que Discord ens brinda moltes opcions per tal que els desenvolupadors siguin capaços de construir i integrar serveis personalitzats.

Entre les integracions més famoses i estimades per la comunitat de desenvolupament de Discord estan els bots i les webhooks, on els programadors poden personalitzar-los mitjançant els llenguatges de Node.js o Python, i així ampliar la funcionalitat de l'aplicació.

Aquestes coses es poden fer mitjançant l'API de Discord, una API oberta que serveix per atendre sol·licituds tant de bots com d'integracions d'OAuth2, un marc d'autorització per obtenir un accés limitat dels usuaris en la xarxa, la qual és compatible amb REST, és a dir, permet interactuar a través d'una web o una aplicació externa, cosa que dona unes àmplies opcions per fer projectes IoT.

### 3.ARDUINO

Com he esmentat anterior, la placa que utilitzaré serà una Wemos D1 mini, molt útil per a projectes IoT (Internet Of Things), ja que integra un mòdul wifi que facilita la comunicació amb altres interfícies sense fils.

Aquesta la podem connectar a l'Arduino IDE (Integrated Development Environment), una plataforma basada en software i hardware lliure, és a dir, els usuaris poden modificar el programari, que serveix com a eina per crear aplicacions per les plaques Arduino, mitjançant una petita configuració detallada a l'ANNEX E. Configuració de l'entorn Arduino IDE.

No és cap misteri, que aquesta aplicació tingui tant d'èxit, ja no per flexibilitat i accessibilitat per personalitzar el teu microcontrolador, sinó pel seu senzill llenguatge de programació, C++, molt útils pels que s'inicien en aquest món.

Més endavant, veurem que haurem de connectar la nostra placa a un servidor MQTT, que ens servirà com a intermediari entre el D1 mini i un client de MQTT. Per aconseguir aquest enllaç farem servir la llibreria "PubSubClient" en l'Arduino IDE, que permet al broker rebre i enviar valors de la placa.

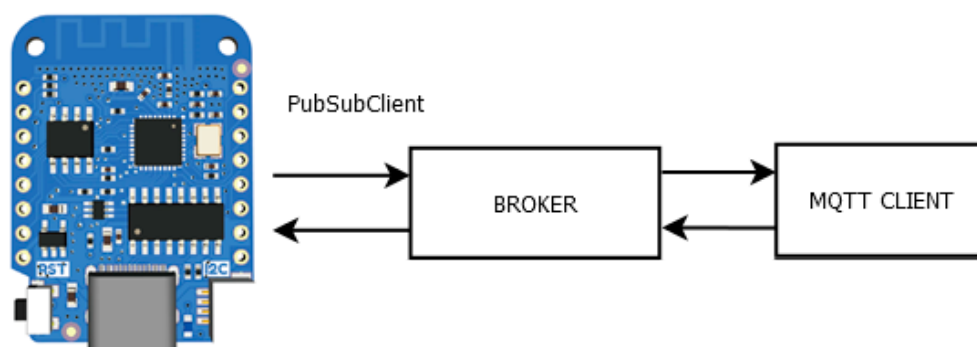


Fig 1: Arduino. Esquema de funcionament de la llibreria "PubSubClient"



## 4.NODE-RED

### 4.1.Què és?

El MQTT client que em referia anteriorment era node-red, una eina de desenvolupament open-source, és a dir, qualsevol usuari té accés al seu codi de programació, creada el 2013 pel grup d'IBM, amb l'objectiu d'integrar un hardware amb altres serveis.

Això fa que sigui una eina útil per a projectes IoT, i per això s'adapta molt al meu projecte, ja que ens permet connectar la nostra placa mitjançant fluxos de serveis a través de protocols MQTT i això processar dades en temps real.

La seva plataforma està basada en Node.js, per tant, podem executar el programari en un hardware barat. En el meu cas, vaig trobar un ordinador Asus A53U per casa, el qual no li donava cap ús i per això vaig aprofitar a instal·lar l'Ubuntu 20.04, tot i que és recomanable usar un raspberry pi.

El que més em va agradar de node-red, és l'accessibilitat de connectar el hardware amb diferents serveis d'internets i APIs, però sobretot per la facilitat alhora de programa, ja que ofereix un tipus de llenguatge visual, a través de fluxos.

Aconseguirem enllaçar amb discord amb la llibreria de node-red "node-red-contrib-discord-advanced", la qual permet interactuar amb discord a través de Discord.js.

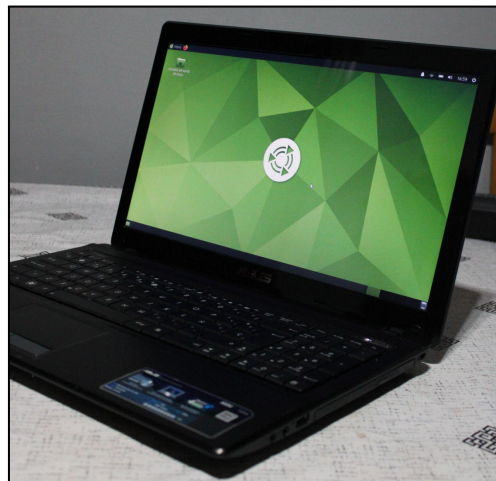

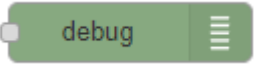

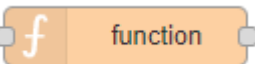
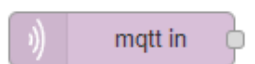
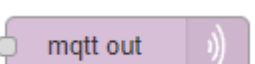
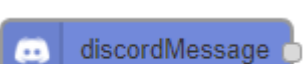

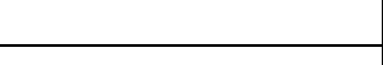
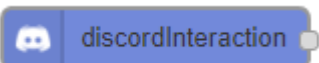


Fig 2: Node-RED. Asus A53U

## 4.2.Taula amb diferents fluxos

A continuació veurem els diferents fluxos bàsics i els de la llibreria de discord.

<u>Nom</u>	<u>Flux</u>	<u>descripció</u>
<b>common</b>		
inject		Activa els fluxos posteriors manualment.
debug		Serveix per mostrar els missatges en la barra lateral.
change		Estableix, canvia o suprimeix propietats d'un missatge.
<b>function</b>		
function		Permet executar missatges amb codi JavaScript.
<b>network</b>		
mqtt in		Rep valors mitjançant MQTT.
mqtt out		Transmet valors mitjançant MQTT.
<b>discord</b>		
discordMessage		Permet rebre el contingut de missatges entrants.
discordMessageManager		Envia, edita o elimina missatges en canals d'un servidor o en xats privats.
discordReactionManager		Identifica les reaccions en un missatge específic.
discordInteraction		Pot rebre comandes, botons o seleccionar interaccions de menú i decidir com respondre a aquestes.

## 5.MQTT

### 5.1.Què és?

Per tal d'intermediar la nostra placa amb node-red, utilitzaré MQTT (**M**essage **Q**ueuing **T**elemetry **T**ransport), un protocol de missatgeria senzill amb un sistema de publicació i subscripció a un *tòpic* específic per rebre missatges.

MQTT igual que node-red és Open Source i és que no és cap casualitat que la integració d'aquest protocol sigui tan senzill en la plataforma, això és degut al fet que tots dos sorgeixen del treball d'IBM. Per això era imprescindible en el projecte, independentment que és el millor protocol per intercanviar dades entre múltiples dispositius IoT.

Un concepte bàsic en MQTT és el *tòpic* o tema, un nom que filtra el missatge entrant o el que es publica del broker, separada amb "/" i està endreçat de més general a més concret, com ara "*casa/cuina/làmpada*", cal recalcar que s'hi diferencien les minúscules i majúscules.

El broker és el responsable de rebre tots els missatges, filtrar-los decidir qui està interessat en ells i després publicar el missatge per a tots els clients subscrits. Entre els més populars i el que nosaltres utilitzarem està *mosquitto broker*.

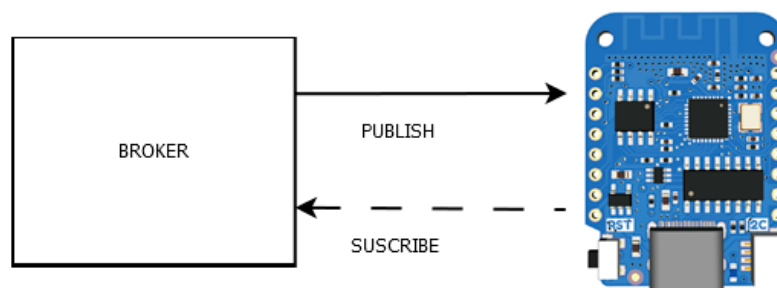


Fig 3: MQTT. Esquema sistema MQTT.

## 6.COMPONENTS

### 6.1.Wemos D1 mini

La placa que utilitzaré serà la “WEMOS D1 MINI”, una placa basada en l’”ESP8266” que inclou un mòdul wifi i les següents característiques.

Microcontrolador	ESP-8266
Voltatge operatiu	3,3 V
Pins digitals	11
Pins d'entrada analògics	1 (3.2V Max)
Velocitat de rellotge	80/160MHz
Flash	4M Bytes
Longitud	34,2 mm
Amplada	25,6 mm
Pes	3 g
Port	micro USB
Programes compatibles	MicroPython, Arduino, nodemcu

A més de l'accessibilitat de la placa en projectes IoT, per la integració d'una antena wifi, l'he triada perquè durant les classes de taller, estàvem fent un projecte semblant al meu on utilitzàvem aquesta. Per tal, d'agafar experiència i poder compartir els meus error i solucionar els de classe.

Tot i que en el meu cas no n'utilitzem cap, cal destacar que aquest microcontrolador inclou port “i2c”, un protocol de comunicació serial que serveix per comunicar un o més *masters* amb un o diversos *slaves*.

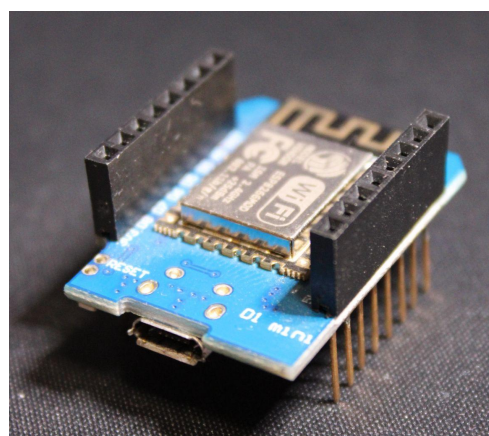


Fig 4: Components. Fotografia de la Wemos D1 mini.

## 6.2.Relay Shield

Aquest mòdul porta un simple relé, un dispositiu electromagnètic que funciona com un interruptor elèctric, el qual permet deixar passar i parar el corrent elèctric dintre d'un circuit elèctric. Pot configurar-se el port entre D1 i D7, en el nostre cas en el D1 i segons si està connectat a N.º o NC els límits de càrrega canvien:

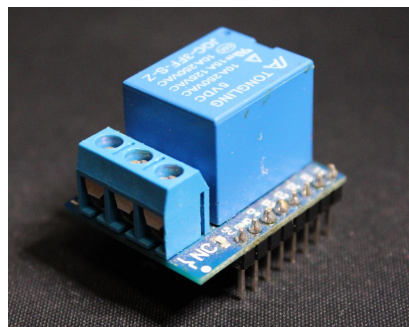


Fig 5: Components. Fotografia del Relé

N.º	5 A (250 V CA/30 V CC), 10 A (125 V CA), MÁX.: 1250 VA/150 W
NC	3A(250VAC/30VDC), MÁX.:750VA/90W

## 6.3.Button Shield

En aquest hi ha integrat un polsador, aparell de comandament que quan està pressionat obtenim *HIGH* i en cas contrari en *LOW*, connectat al port D3.



Fig 6: Components. Fotografia del Polsador

## 6.4.Buzzer Shield

Aquest shield inclou un bronzidor, un tipus de vibrador que produeix un so, connectat en la pota D5. Gràcies a la funció "*analogWriteFreq(freq)*", ens permet fer melodies.

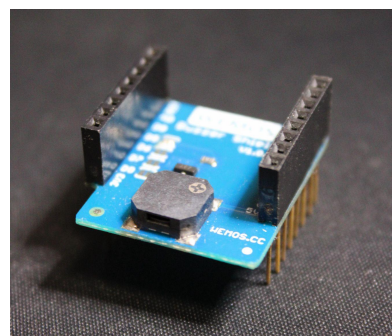


Fig 7: Components. Fotografia del bronzidor.

## 6.5.RGB Shield

L'últim shield tenim el RGB Shield, un mòdul que obté un led RGB connectat al pin D2. La forma més senzilla i per extreure-li més fruit és amb la llibreria “*Adafruit\_NeoPixel*”, que podem instal·lar des del gestor de llibreries de l'Arduino IDE.

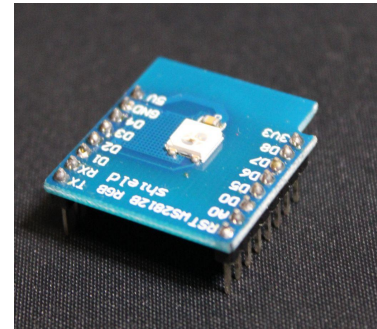


Fig 8: Components. Fotografia del RGB.

## 6.6.Tripler Base

Finalment, tenim un tripler base, el qual hem soldat terminals femelles, que ens serviran per poder enllaçar tots els nostres mòduls i el D1 mini en blocs.

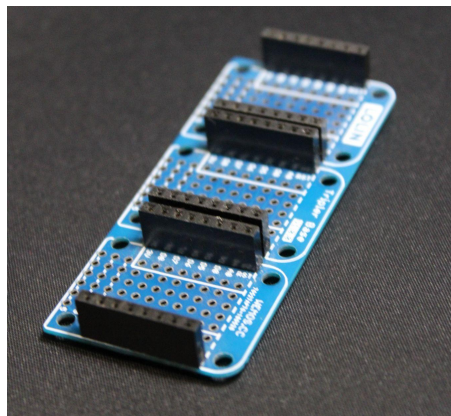


Fig 9: Components. Fotografia del tripler base.

## 7.DISENY

### 7.1.Diagrama de blocs

En aquest diagrama mostra les connexions i les potes utilitzades en total en el nostre microcontrolador:

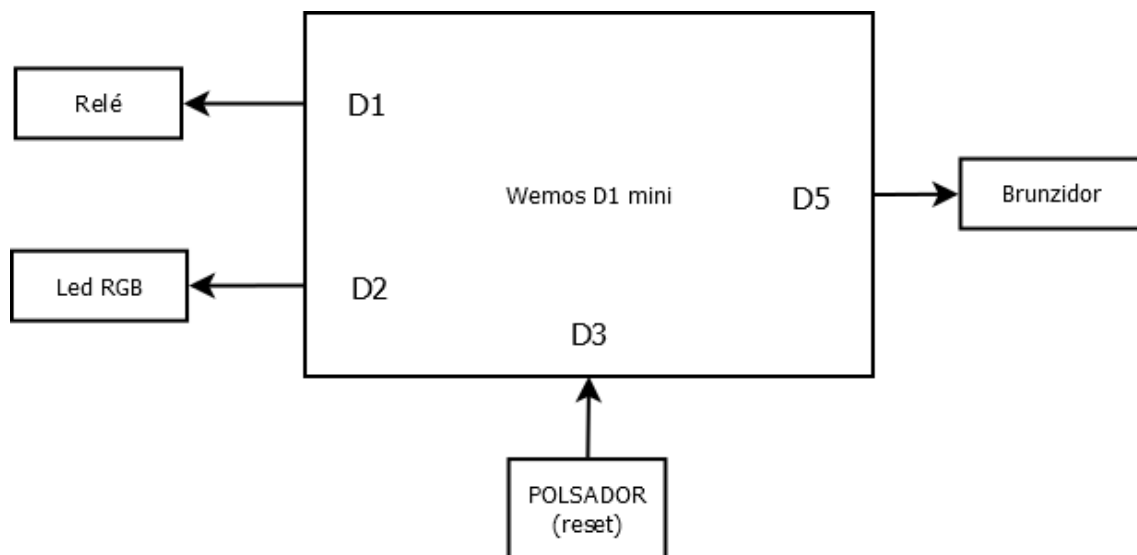


Fig 10: Disseny. Diagrama de blocs del projecte.

Resumidament, tenim una entrada digital:

- **D3:** On connectarem el polsador, que utilitzarem per resetejar la placa.

Tres sortides digitals:

- **D1:** En aquest integrarem el relé, per tal poder fer una millora en un futur i poder connectar un altre component.
- **D2:** En aquest tindrem el component més important per la nostra placa, ja que ens servirà per donar llum i colors a la nostra làmpada.
- **D5:** En aquesta pota, posarem un brunzidor, així podem crear una melodia diferent segons la comanda.

## 7.2.Esquemes

### 7.2.1. Relay Shield

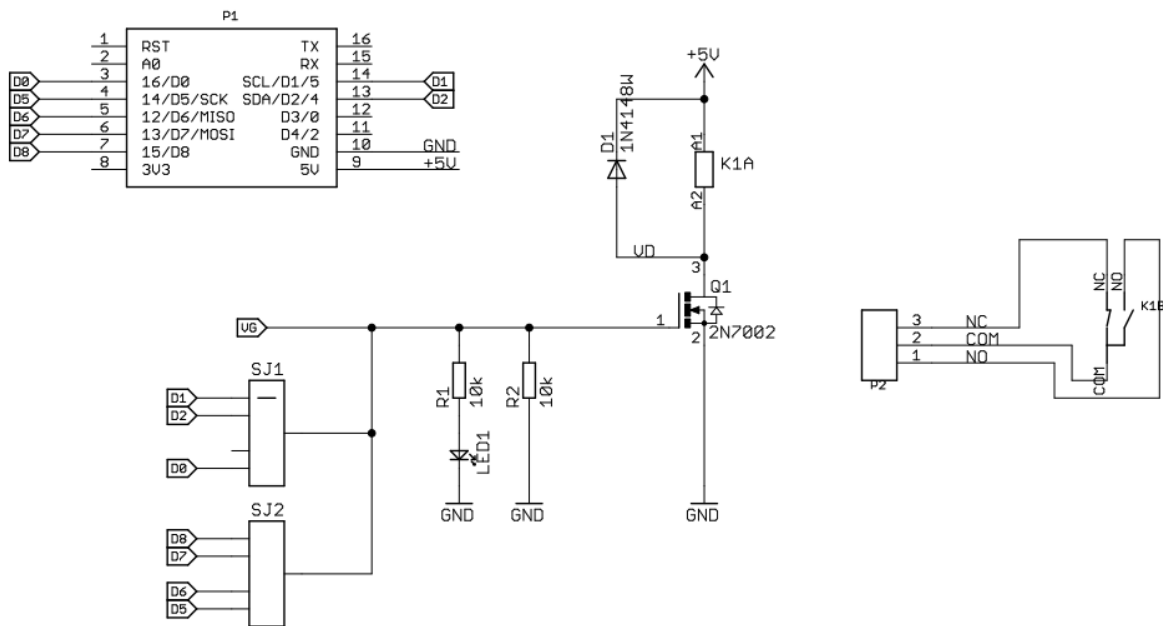


Fig 11: Disseny. Esquema Relay Shield.

### 7.2.3. Buzzer Shield

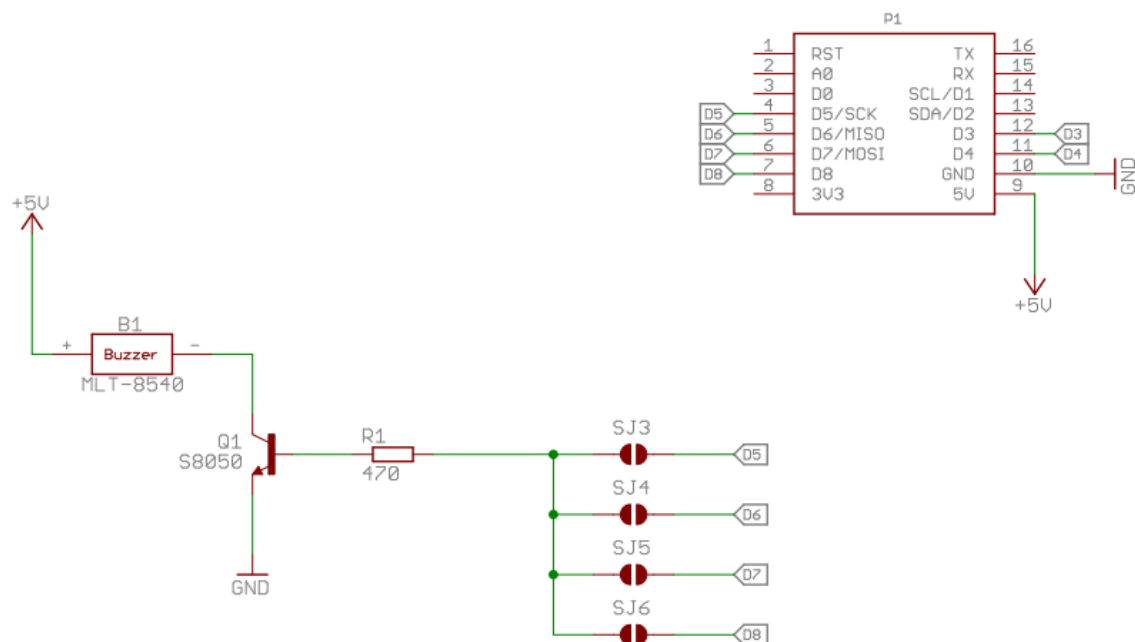


Fig 12: Disseny. Esquema Buzzer Shield.



### 7.3.Disseny de la base de la capsa

Indagant per internet, vaig trobar diversos dissenys, però per la senzillesa i el seu aspecte gaming, em vaig decantar per una base que subjecta a una peça de metacrilat on hi ha un dibuix gravat i s'il·lumina per un led que està situat avall.

Per fer la base, vaig utilitzar una caixa que tenia disponible, i vaig començar a prendre mides perquè quedés una mica d'espai per si en un futur volíem desarmar els shields, ja que després anirà cargolada perquè no es mogui.



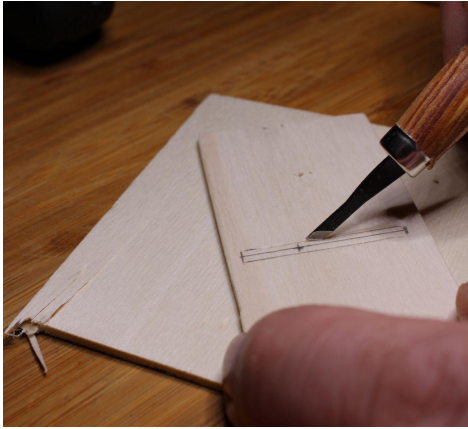
Fig 13: Disseny. Mesurament capsa.

Un cop preses les mides, vaig retallar totes les parts amb ajuda d'una serra de calar i les vaig enganxar amb adhesiu de muntatge i claus.

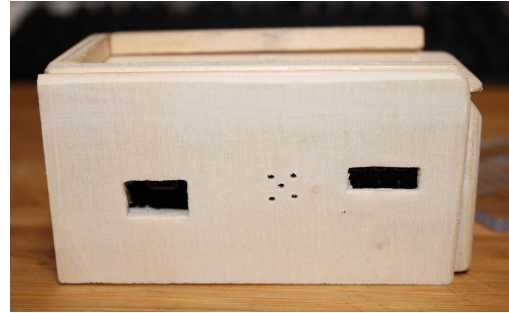


Fig 14: Disseny. Mesurament capsa.

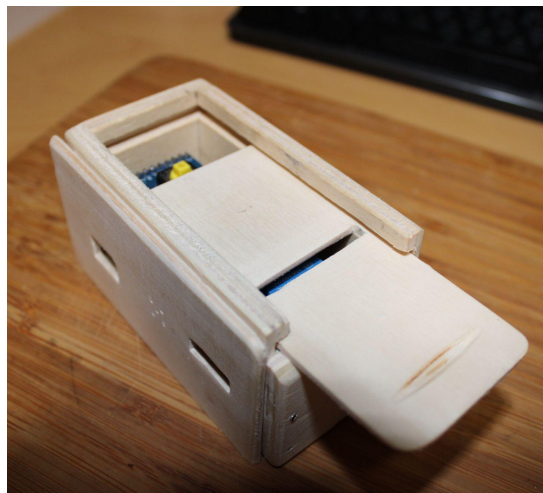
Ja feta la caixa, amb un trepant vaig començar a fer els forats del brunzidor, el cable d'alimentació i el relé, i en el cas de la tapa, el botó de reset. Finalment, els vaig donar un acabat final amb un paper de vidre i amb gúbies, que també m'ajudaré per retallar el forat de la tapa, on anava la peça de metacrilat.



**Fig 15:** Disseny. Tallament de la tapa.



**Fig 16:** Disseny. Forats capsa.



**Fig 17:** Disseny. Base de la làmpada.

<p style="text-align: center;"><b>ALÇAT</b></p> <p style="text-align: center;"><b>Fig 18:</b> Disseny. Alçat de la base de la làmpada.</p>	<p style="text-align: center;"><b>PERFIL</b></p> <p style="text-align: center;"><b>Fig 19:</b> Disseny. Perfil de la base de la làmpada.</p>
<p style="text-align: center;"><b>PLANTA</b></p> <p style="text-align: center;"><b>Fig 20:</b> Disseny. Planta de la base de la làmpada</p>	<p style="text-align: center;"><b>TAPA</b></p> <p style="text-align: center;"><b>Fig 21:</b> Disseny. Planta de la tapa de la base de la làmpada.</p>

#### 7.4. Disseny de les peces de metacrilat

Com que no em decidia entre dos models de peces, en vaig fer dues, perquè es puguin anar canviant quan vulguem, amb una placa de metacrilat que vaig comprar en un basar, i gràcies amb la serra de calar vaig retallar.



Fig 22: Disseny. Tallament de les peces de metacrilat.

Finalment, després de retallar-les i donar-los forma amb un paper de vidre, vaig gravar el dibuix amb una llima rotatòria.



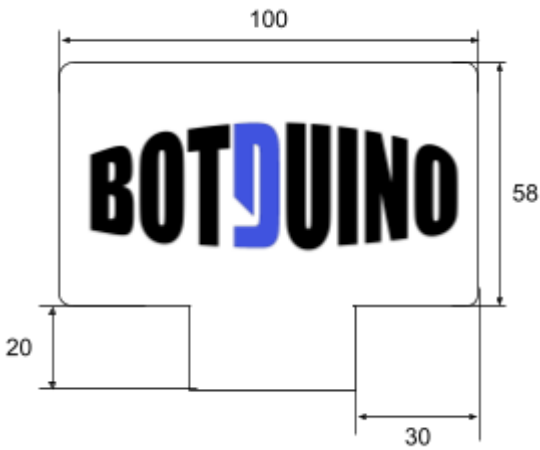


Fig 23: Disseny. Gravat de les peces.



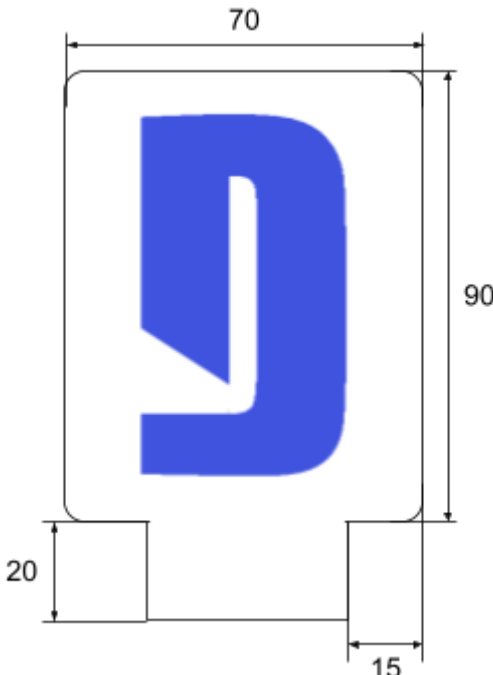

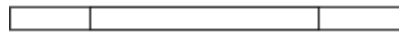
Fig 24: Disseny. Resultat de la làmpada.

### 7.4.1. Disseny tècnic de les peces de metacrilat

- Peça "BOTDUINO"

ALÇAT	PERFIL
 <p>Fig 25: Disseny. Alçat de la peça "Botduino".</p>	 <p>Fig 26: Disseny. Perfil de la peça "Botduino".</p>
PLANTA	
 <p>Fig 27: Disseny.Planta de la peça "Botduino".</p>	

- Peça "D"

ALÇAT	PERFIL
 <p><b>Fig 28:</b> Disseny. Alçat de la peça "D"..</p>	 <p><b>Fig 29:</b> Disseny. Perfil de la peça "D"..</p>
PLANTA	
 <p><b>Fig 30:</b> Planta de la peça "D"..</p>	

\*A l'ANNEX F.Plantilla peces de metacrilat a mida real, podreu imprimir les peces a mida real.



## 8.SOFTWARE

El gran repte d'aquest projecte era la part del programari i trobar la manera d'enllaçar discord amb la nostra placa. Per això, després d'hores d'investigació i múltiples intents amb diversos mètodes diferents escrits en les Conclusions: Problemes vaig encertar amb node-red.

Aquesta idea d'utilitzar node-red me la va proposar el meu tutor de treball de recerca i professor de tecnologia industrial, que com ja estàvem treballant en classe, sabíem que mitjançant MQTT i la biblioteca "PubSubClient" podíem accionar amb el D1 mini desde node-red.

Ara només calia trobar alguna manera que pugui involucrar discord, i és per això que després d'investigar vaig trobar la biblioteca "node-red-contrib-discord-advanced" que a través de discord.js permet interactuar amb discord i, per tant, rebre informació de comandes i enviar missatges rebuts des de node-red, entre altres.

Les comandes que vaig crear:

NOM	OBJECTIU	ACCIÓ
/partit	Avisa als altres jugadors que hi ha partit.	Encén el color blau, el relé i activa la melodia de Pac-Man
/jugar	Notifica als companys per jugar per diversió.	Activa el verd, relé i la melodia de Star Wars.
/escalfar	Alerta als usuaris per escalfar abans de jugar.	Posa el color vermell, el relé i la música de pirates del carib.

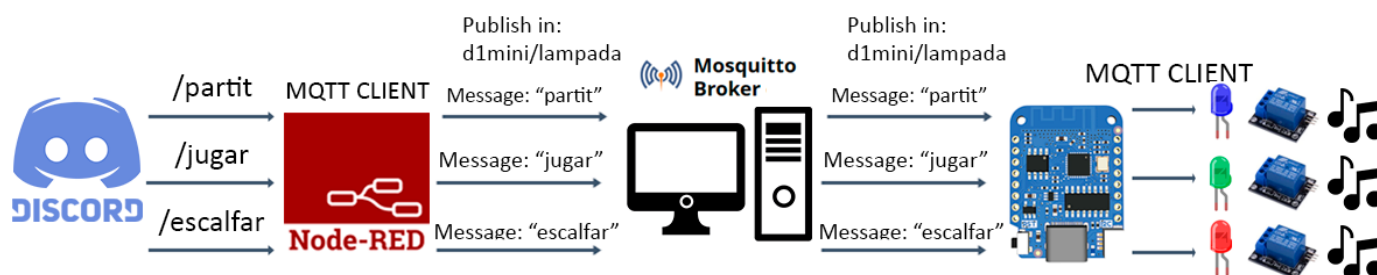


Fig 31: Software. Esquema funcionament del projecte.

## 8.1.Arduino i MQTT

A continuació explicaré el codi Arduino (Codi sencer en l'ANNEX K.Codi Arduino) que he fet servir per connectar la placa amb MQTT, subscriure's al tòpic i com actuar segons el missatge rebut.

1. Incloem les següents llibreries i notes.h (Codi en ANNEX O.Codi notes.h) que haurem de posar en una nova pestanya.

```
#include <ESP8266WiFi.h>
#include <PubSubClient.h>
#include <Adafruit_NeoPixel.h>
#include "notes.h"
```

2. Declarem els pins i configurem la llibreria NeoPixel,

```
const int buzzer = D5;
const int button = D3;
const int relay = D1;
const int RGB = D2;

Adafruit_NeoPixel pixels = Adafruit_NeoPixel(1, RGB, NEO_GRB +
NEO_KHZ800);
```

3. Posem les credencials per connectar-se al server mqtt i el wifi.

```
const char* ssid      = "POSAR_SSID";
const char* password = "POSAR_CONTRASENYA";
const char* mqtt_server = "DIRECCIÓ_IP_SERVER_MQTT";
```

4. Posem les instàncies wifi i client MQTT i creem dues variables globals per emmagatzemar el tòpic i el payload.

```
WiFiClient espClient;
PubSubClient client(espClient);

String _topic;
String _payload;
```

5. Creem una funció setup\_wifi() que ens permetrà connectar-nos al wifi.

```
delay(10);
Serial.println();
Serial.print("Conectant a ");
Serial.println(ssid);
WiFi.begin(ssid, password);
while (WiFi.status() != WL_CONNECTED) {
```



```
    delay(500);
    Serial.print(".");
}
Serial.println("");
Serial.print("Wifi conectat! - ESP Direcció IP: ");
Serial.println(WiFi.localIP());
}
```

6. Funció `reconnect()`, permet connectar-nos i subscriure'ns a un tòpic en el broker i la funció `callback()` que rep els missatges del client.

```
void reconnect() {
  while (!client.connected()) {
    Serial.print("Intentant connexió MQTT ..");
    String clientId = "ESP8266Client-";
    clientId += String(random(0xffff), HEX);
    if (client.connect("ESP8266Client")) {
      Serial.println("Conectat!");
      client.subscribe("dlmini/lampada");
    } else {
      Serial.print("falla, estado=");
      Serial.print(client.state());
      Serial.println("Intentant connectar novament en 5 segons");
      delay(5000);
    }
  }
}

void callback(char* topic, byte* payload, unsigned int length) {
  Serial.print("Ha arribat missatge sobre el topic: ");
  Serial.print(topic);
  Serial.print(". Missatge: ");
  String conc_payload;
  for (int i = 0; i < length; i++) {
    Serial.print((char)payload[i]);
    conc_payload += (char)payload[i];
  }
  Serial.println();
  _topic = topic;
  _payload = conc_payload;
}
```

7. Iniciem el port serial, els pins i establim el servidor MQTT.

```
void setup() {
  pinMode(relay, OUTPUT);
  pixels.begin(); //Iniciem biblioteca NeoPixel
  pinMode(buzzer, OUTPUT);
  digitalWrite(buzzer, LOW);
  analogWriteRange(1023);
  Serial.begin(115200);
  setup_wifi();
  client.setServer(mqtt_server, 1883);
  client.setCallback(callback);
}
```

8. Dins del void loop(), comprovem que estem connectats i reiniciem la placa amb ESP.restart().

```
void loop() {  
  if (!client.connected()) {  
    reconnect();  
  }  
  if(!client.loop())  
    client.connect("ESP8266Client");  
  
  if (digitalRead(button) == LOW){  
    ESP.restart();  
  }  
}
```

9. Finalment, configurem segons el missatge que rep el microcontrolador. Per exemple, si arriba el missatge "jugar"

```
if (_payload == "jugar"){  
  pixels.setPixelColor(0, pixels.Color(0, 255, 0)); //Color verd  
  pixels.show();  
  digitalWrite(relay, HIGH);  
  
  //Melodia de Star Wars  
  tone(buzzer, La, 500);  
  delay(500+50);  
  tone(buzzer, La, 500);  
  delay(500+50);  
  tone(buzzer, La, 500);  
  delay(500+50);  
  tone(buzzer, Fa, 350);  
  delay(350+50);  
  tone(buzzer, DoH, 150);  
  delay(150+50);  
  tone(buzzer, La, 500);  
  delay(500+50);  
  tone(buzzer, Fa, 350);  
  delay(350+50);  
  tone(buzzer, DoH, 150);  
  delay(150+50);  
  tone(buzzer, La, 1000);  
  delay(1000+50);  
  tone(buzzer, MiH, 500);  
  delay(500+50);  
  tone(buzzer, MiH, 500);  
  delay(500+50);  
  tone(buzzer, MiH, 500);  
  delay(500+50);  
  tone(buzzer, FaH, 350);  
  delay(350+50);  
  tone(buzzer, DoH, 150);  
  delay(150+50);  
  tone(buzzer, Sols, 500);  
  delay(500+50);  
  tone(buzzer, Fa, 350);  
  delay(350+50);  
  tone(buzzer, DoH, 150);  
  delay(150+50);  
}
```

```
tone(buzzer, La, 1000);
delay(1000+50);
tone(buzzer, LaH, 500);
delay(500+50);
tone(buzzer, La, 350);
delay(350+50);
tone(buzzer, La, 150);
delay(150+50);
tone(buzzer, LaH, 500);
delay(500+50);
tone(buzzer, SolHs, 250);
delay(250+50);
tone(buzzer, SolH, 250);
delay(250+50);
tone(buzzer, FaHs, 125);
delay(125+50);
tone(buzzer, FaH, 125);
delay(125+50);
tone(buzzer, FaHs, 250);
delay(250+50);
delay(250);
tone(buzzer, Las, 250);
delay(250+50);
tone(buzzer, ReHs, 500);
delay(500+50);

digitalWrite(relay, LOW);
pixels.setPixelColor(0, pixels.Color(0, 0, 0));
pixels.show();
digitalWrite(buzzer, LOW);
_payload = "";
}
```

Per fer les melodies he utilitzat tone(pin, freqüència, temps), més informació de les freqüències en l'ANNEX N.Taula de freqüències de les notes utilitzades (Hz). Aquestes melodies les he trobat en internet, menys la de Pac-Man que ho he fet gràcies a una pàgina per crear l'arxiu midi al codi Arduino.

## 8.2. Node-RED

Encara que aquesta part era la més senzilla, ja que només havia d'aconseguir que arribessin els missatges al tòpic *d1mini/lampada*, vaig dividir aquest treball en dos, pel fet que no estava familiaritzat amb aquest tipus de programació visual.

Llavors, després de configurar la placa amb el codi arduino perquè segons que missatge rebí actuï d'una manera diferent, el primer pas que vaig fer, va ser posar tres fluxos manuals, perquè quan premi envii el missatge configurat:

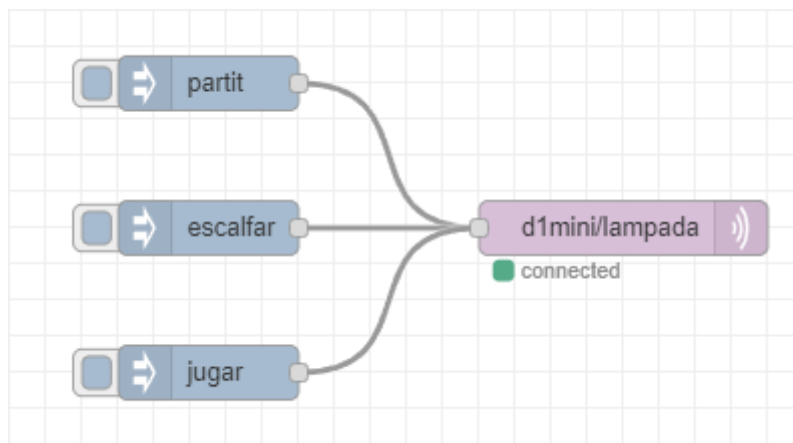


Fig 32: Software. Node-RED 1.

Quan em vaig assegurar que tot funcionava, ara només havia de canviar els fluxos manuals, per comandes de discord. Però abans de continuar necessitarem el generador de comandes per crear aquestes pel bot.

Vam utilitzar Discord Slash Command Builder, on haurem de configurar el nostre bot i afegim les comandes que volem amb una petita descripció. En el meu cas `/partit` i la descripció: "Avisa als teus companys que hi ha partit, posant `/partit`".

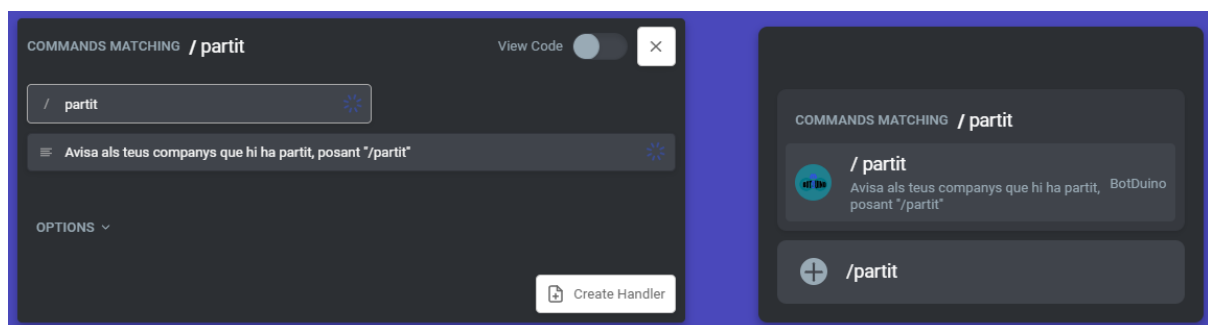


Fig 33: Software. Generador de comandes.

Un cop configurat les comandes i investigant, vaig utilitzar el node "discordInteraction", configurat perquè s'activi quan rebi una comanda, connectat a un flux "change" perquè envii el missatge "partit" al tòpic d1mini/lampada. Finalment, he afegit un node debug i així amb les altres comandes. (Codi exportat en l'ANNEX L.Codi node-red). Tota la creació i configuració dels fluxos els he documentades en l'ANNEX M.Creació i configuració de fluxos.

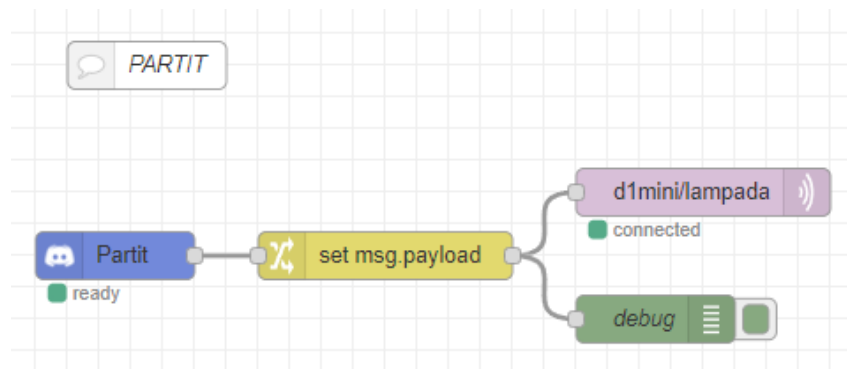


Fig 34: Software. Comanda "/partit".

## 9. PLANIFICACIÓ

A continuació, he creat una taula de la planificació amb el temps, materials i les eines emprats per poder realitzar les operacions. Cal tenir en compte, que en aquesta taula no he documentat les vies que he seguit i no han tingut sortida, per algun error o incompatibilitat, que he detallat a les conclusions.

	Operacions	Materials	Eines	Temps
General	Metodologia del projecte	-	Ordinador	02:00h
	Recerca d'informació	-	Ordinador	01:00h
	Planificació del projecte	-	Llibreta i llapis	00:30h
Preparació de l'entorn	Elaboració D1 mini	Estany D1 mini Relay shield Button shield Buzzer shield RGB shield Tripler base	Soldador d'estany	00:30h
	Instal·lar Arduino i configurar placa	Placa D1 mini	Ordinador Programa Arduino	00:45h
	Instal·lar node-red	-	Ordinador hosting Programari terminal	01:00h
	Instal·lar Mosquitto Broker	-	Ordinador hosting Programari terminal	00:45h
	Crear servidor de discord	-	Ordinador Aplicació Discord	00:30h
	Crear Bot de discord	-	Ordinador Aplicació Discord	00:45h
Marc pràctic	Codi arduino	Placa D1 mini	Ordinador Programa Arduino	05:00h
	Programa node-red	Placa D1 mini	Ordinador hosting Ordinador node-red Programa arduino Programari terminal	03:00h
	Construcció de la làmpada	D1 mini Relay shield Button shield Buzzer shield RGB shield Tripler base Fusta	Peu de rei Regle Llapis Serra de calar Trepant Paper de vidre	02:45h

		placa de metacrilat	Gúbies	
Altres	Esbós del disseny	Paper	Llapis Peu de rei Regle	01:00h
	Documentació i elaboració de la memòria.	-	Ordinador	05:00h
	Comprovació funcionament.	-	-	00:30h
<b>Temps total</b>				28:00h

## 10.PRESSUPOST

Descripció	Proveïdor	Quantitat	Preu unitari	Preu total
Raspberry Pi 4	Amazon	1	189,99€	189,99 €
LOLIN D1 mini	Aliexpress	1	2,30€	2,30€
Relay Shield	Aliexpress	1	0,91€	0,91€
Button Shield	Aliexpress	1	0,65€	0,65€
Buzzer Shield	Aliexpress	1	1,24€	1,24€
RGB Shield	Aliexpress	1	1,13€	1,13€
Tripler Base	Aliexpress	1	0,83€	0,83€
Estany 0,6 mm	Aliexpress	50 g	0,98€/50g	0,98
Placa de metacrilat	Basar	1	8€	8€
<b>Preu final</b>				<b>206,04€*</b>

\*Aquest preu és provisional, ja que en el cas de haver sigut demanat a un enginyer, cal afegir el preu de mà d'obra. En la taula següent, tenint en compte el cost mitjà d'un enginyer, us faré una aproximació:

<b>Preu mà d'obra per hora</b>	15€
<b>Temps emprat</b>	28:00h
<b>Preu de mà d'obra total</b>	420€
<b>Preu sense mà d'obra</b>	206,04€
<b>IVA (+21%):</b>	137,47€
<b>Preu total:</b>	<b>763,51€</b>

Encara que aquest valor s'assembla més al preu final, és important saber que per abaratir aquest preu, normalment és disminueix el temps, comprant a l'engròs i més tècniques. A més cal tenir en compte, que per a un grup de jugadors, només caldria una raspberry pi, i el preu per làmpada seria molt més baix.



## 11.CONCLUSIONS

### 11.1.Avaluació del projecte i possibles millores

Malgrat alguns inconvenients a l'inici, hem aconseguit acabar el nostre projecte i, per tant, la majoria de les nostres metes proposades inicialment, les hem pogudes superar:

- ☒ He instal·lat node-red per enllaçar el meu microcontrolador amb discord.
- ☒ He utilitzar el protocol MQTT, per connectar node-red amb el D1 mini.
- ☒ Tot això instal·lat en un ordinador vell.
- ☒ He fet un codi arduino perquè actuï segons el missatge que rep del *broker*.
- ☒ Vaig crear un bot de discord
- ☒ Vaig trobar una biblioteca per poder fer ús del bot en node-red.
- ☒ He fet un disseny a mida per la nostra placa.
- ☒ He sigut capaç de fer la làmpada.
- ☒ Finalment, ho he documentat en aquesta memòria.

Però, tot i haver superat els nostres objectius, en aquestes altures i a mesura que anava fent el projecte he anat pensant en algunes millores. Entre la més important trobo l'aplicació del meu projecte, ja que tot i ser original i innovadora, penso que potser no és tan útil que com una alarma antirobatori o per a control de persones majors, encara que l'opció més realista segons els usuaris de discord ha sigut aquesta. També penso que tot i que podria haver afegit més funcions al nostre bot de discord, com publicar valors de temperatura cada interval de temps.

Pel que fa al disseny, continuaria amb la idea d'il·luminar la placa de metacrilat, tot i que com a millora, primer posaria més LEDs per augmentar la lluminositat i les activaria a ritme de la melodia. Finalment, per donar-li més professionalitat al producte, faria la base a mida en impressora 3D.

### 11.2.Problemes

La part més graciosa d'aquest projecte, no era tant l'aplicació que li donem, sinó trobar la manera de poder rebre o enviar ordres a un microcontrolador mitjançant

una aplicació de text, en el nostre cas, discord. Per això, ja sabia que anava a ser un repte aconseguir-ho i era conscient que em toparia amb alguns problemes.

Aquesta part del projecte, va passar per diverses versions abans d'escollir el mètode actual. Entre elles primer vaig cercar la manera de fer-ho amb MQTT, però investigant, només vaig trobar la manera de comunicar-me amb les webhooks. Les quals ens limiten a enviar missatges de valors, és a dir, temperatures, humitat... Però no rebre cap missatge, i enviar-lo a la placa.

Llavors, és quan investigant vaig trobar dues maneres: la primera va ser la de fer un bot de discord amb node.js i construir un API REST usant express.js perquè aquest es comuniqui amb el D1 mini, i la segona era fer un bot de discord i utilitzar la llibreria Johnny-Five, una plataforma que corre en un servidor node.js i serveix per integrar l'arduino amb javascript, cosa que dona una amplia possibilitats per a projectes IoT.

Entre aquestes dues possibilitats, només em vaig adentrar a la de Johnny-Five, encara que posteriorment, quan ja tenia el projecte avançat, em vaig topar amb un error. Per fer el bot, necessitava instal·lar el paquet discord.js el qual exigia una versió de node.js nova. En canvi, per instal·lar Johnny-Five, necessitem una versió de node.js una mica més antiga, ja que aquestes contenen un paquet en *node\_modules* de serialport específic.

### 11.3. Altres aplicacions

Sincerament, crec que una de les característiques del meu projecte, és que pot tenir diverses aplicacions. De fet, com he anomenat, si no fos perquè volia donar-li més realisme i una utilitat diferent dels estàndards, el meu projecte inicialment anava a ser un sistema d'avís per a persones majors. A continuació us deixo alguns exemples més:

- **Alarma antirobatori:** Que podria incloure un interruptor magnètic, un sensor de moviment o una càmera i que informi amb discord.

- **Estació meteorològica:** la qual amb sensors de temperatura, il·luminació i altres pugui enviar els valors en un missatge de discord, cada x temps o bé quan escriguis una comanda.
- **Alarma de fum:** Que avisi amb un missatge quan un detector de flama o fum detecti foc o gas.
- **Time lapse:** una tècnica de fotografia la qual consisteix a fer una foto cada x temps, que podria enviar-les a un servidor de discord.
- **Control per a plantes:** Una altra aplicació la podríem utilitzar per veure la humitat de la planta i mitjançant una comanda accionar un sistema de reg, en cas que estiguem de vacances.
- **Sistema de domòtica:** La qual pot servir per automatitzar la casa, mitjançant unes comandes, poder pujar o baixar les persianes, controlar les llums... En aquest aspecte hi ha una plataforma "*Home assistant*" la qual permet utilitzar node-red i connectar amb dispositius intel·ligents mitjançant wifi.

#### 11.4. Valoració personal

Tot i tots els disgustos que m'he emportat cada vegada que un mètode no em funcionava, aquest projecte m'ha agradat molt, sobretot aquells moments de satisfacció cada vegada que aconseguia progressar un pas.

A més durant tot aquest temps, aquest treball m'ha convençut del meu interès en aquest àmbit i sobretot m'ha fet ampliar els meus coneixements, especialment en ampliar els meus coneixements de programació en projectes IoT, C++ i node-red, i m'ha ajudat professionalment, com ara organitzar-me per fer un treball d'aquestes dimensions o fer-me adonar que sempre n'hi ha sortides a tots els problemes, sinó que només cal dedicar-li temps.

## AGRAÏMENTS

Vull agrair a totes les persones que sense la seva presència i ajuda no hagués sigut possible aquest treball, sobretot al meu tutor del treball de recerca, Jordi Orts, no només per brindar-me material quan ho he necessitat sinó també pels seus consells, la facilitació d'informació i per revisar la meva memòria.

També m'agradaria donar les gràcies a la meva família, que són els que m'han donat suport durant tot aquest temps que he estat fent aquest projecte, especialment en ajudar-me a fer la làmpada.

## 12.ANNEXOS

### A. ANNEX A.Creació d'un servidor

Per tal d'intercanviar idees o interessos amb altres usuaris, discord ens ofereix crear un lloc amb diferents canals de veu i text, anomenats servidors. Abans de fer el bot, haurem de continuar uns passos per tal de crear un servidor que ens faciliti la interacció amb el bot.

1. Un cop en l'aplicació, haurem d'anar a la barra de l'esquerra i pressionar el símbol "+" de "Añadir un servidor".

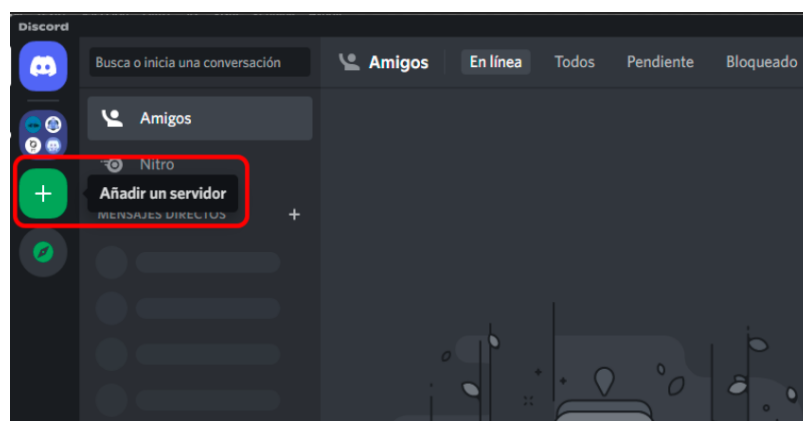


Fig 35: Discord. Crear servidor 1

2. S'obriran dues finestres, una on podrem escollir entre unir-te a un servidor o crear-ne un i l'altra si el servidor és d'amics o per crear una comunitat. Farem clic a "Crear un servidor" i l'altra la podrem ometre.

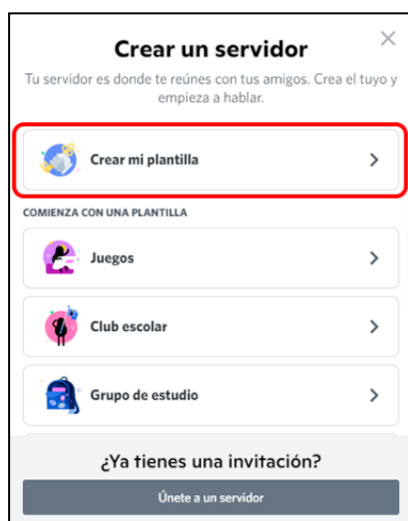


Fig 36: Discord. Crear servidor 2.1



Fig 37: Discord. Crear servidor 2.2

3. Llavors, sortirà una opció per posar el nom i pujar una icona.

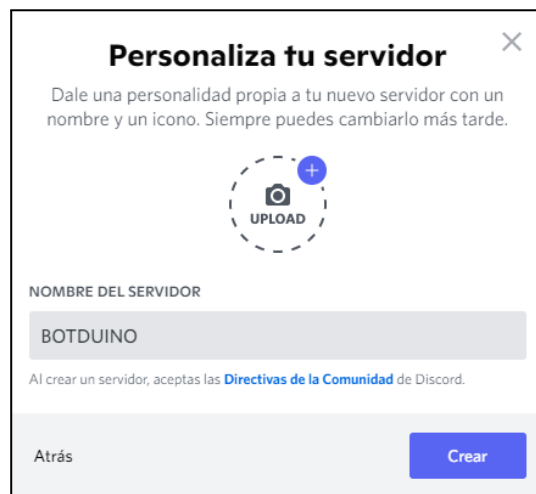


Fig 38: Discord. Crear servidor 3

4. Finalment, podrem afegir canals de text o veu, i diferents categories per endreçar a aquests, o bé editar les predeterminades. Per exemple, vaig posar un canal de proves en la categoria "Projecte".



Fig 39: Discord. Crear servidor 4.1

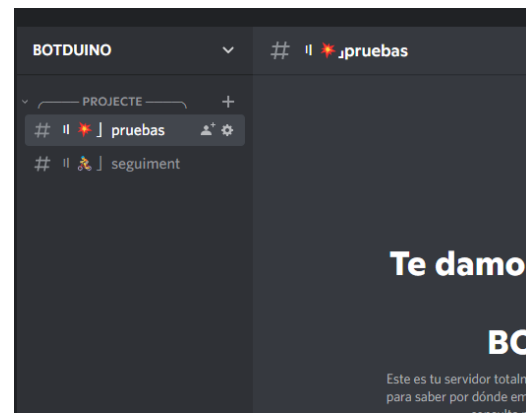


Fig 40: Discord. Crear servidor 4.2

## B. ANNEX B.Creació d'un bot

Per comunicar la nostra placa amb discord ens era indispensable trobar una integració automàtica dins de la plataforma. En cerca d'una d'aquestes vaig trobar les webhooks, un sistema que ens facilita la comunicació entre apps mitjançant un intercanvi de dades entre aplicacions web. Tot i que, malauradament només podia publicar les dades obtingudes del D1 mini i no podia rebre comandes per enviar-les a aquest.

Per això em vaig haver de decantar als bots, una integració molt utilitzada en l'aplicació, ja que poden complir de forma independent i automàtica moltes funcions, on els usuaris aprofiten per millorar la funcionalitat i l'experiència en els servidors.

Qualsevol usuari pot crear-ne un gratuïtament, seguint aquestes instruccions.

1. Abans de crear un bot, cal activar l'opció de "Modo desarrollador" dins de "Ajustes de usuario">"Avanzado"

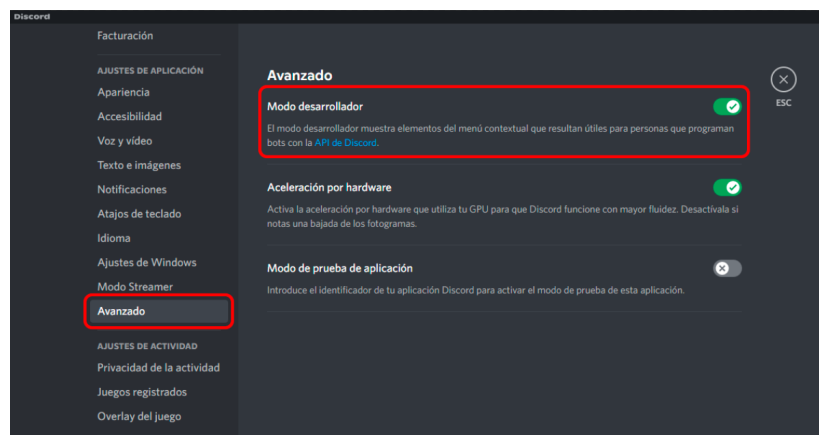


Fig 41: Discord. Crear bot 1

2. Un cop fet, ens dirigirem al portal de desarrolladores de discord en internet i farem clic a "New Application".

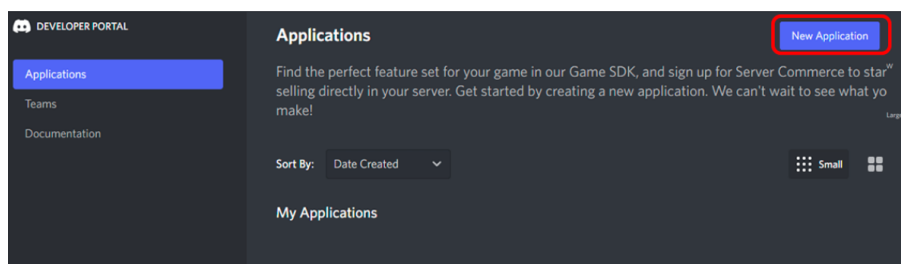


Fig 42: Discord. Crear bot 2

3. Després d'això només ens faltaria posar el nom del bot, en el meu cas posaré "BOTDUINO", i acceptar els termes de servei i la política.

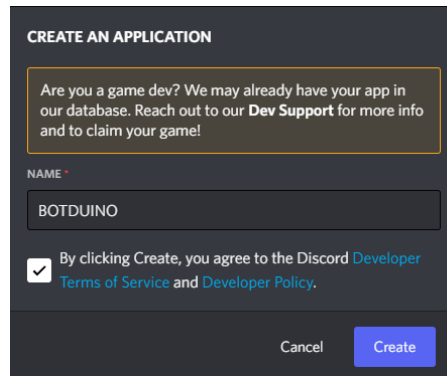


Fig 43: Discord. Crear bot 3

4. Per crear un bot en l'aplicació anirem a "Bot">"Add Bot".

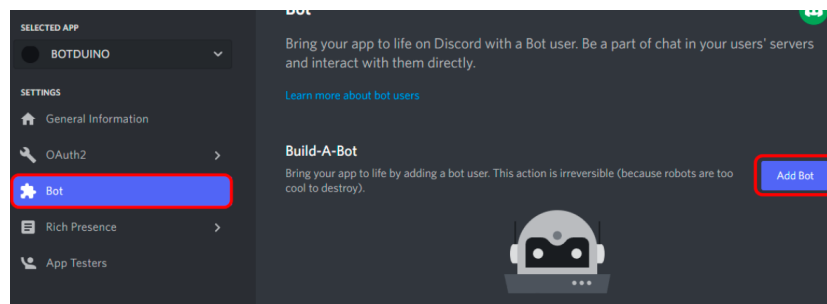


Fig 44: Discord. Crear bot 4

5. Per tal, d'unir el teu bot al servidor, només haurem d'anar a "OAuth 2">"URL Generator">"bot" i donar-li permisos d'administrador.

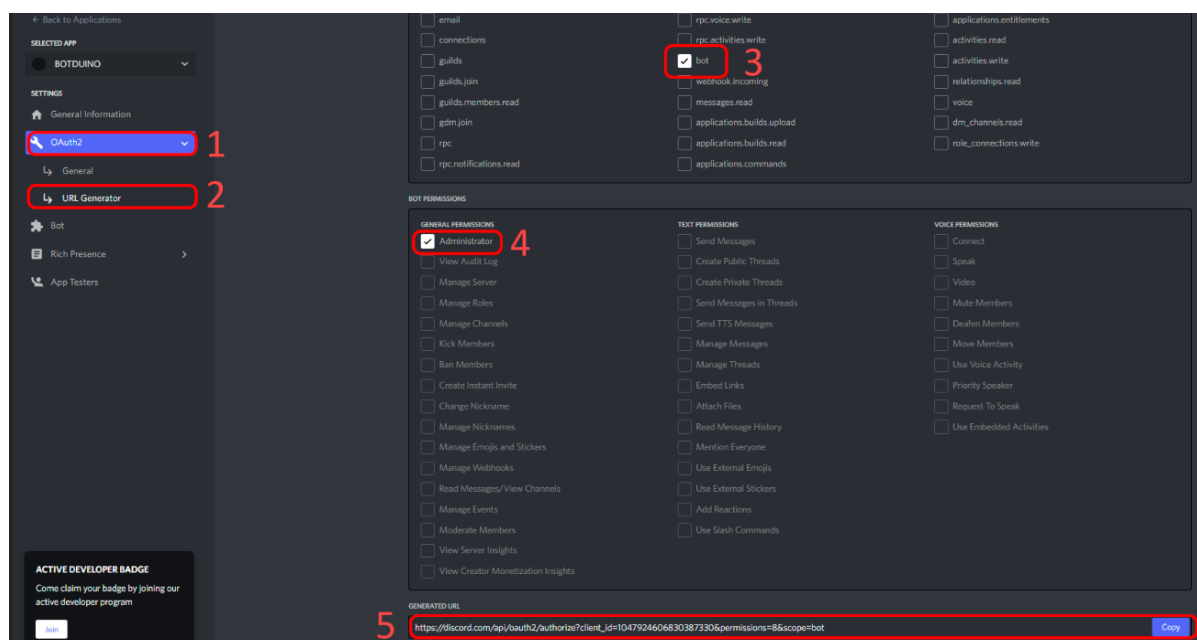


Fig 45: Discord. Crear bot 5



6. Finalment, entrem a l'enllaç i seleccionem el servidor on volem invitar al bot.

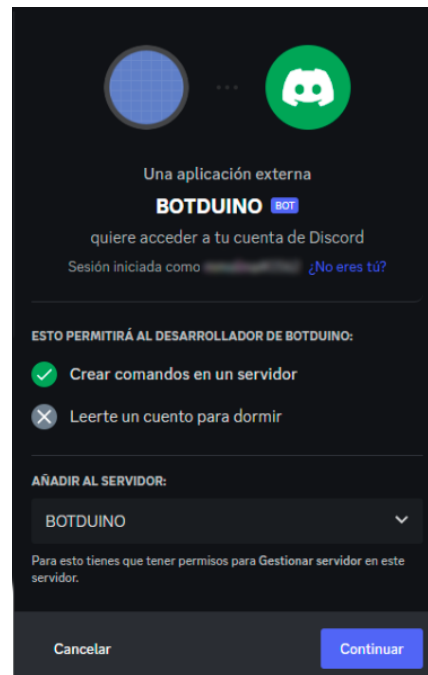


Fig 46: Discord. Crear bot 5

### C. ANNEX C.Instalar Node-RED en Ubuntu 20.04

Un cop obert la terminal amb la combinació “*ctrl + alt + t*” cal posar aquestes comandes en ordre:

1. Primer actualitzem el servidor:

```
sudo apt update && sudo apt upgrade -y
```

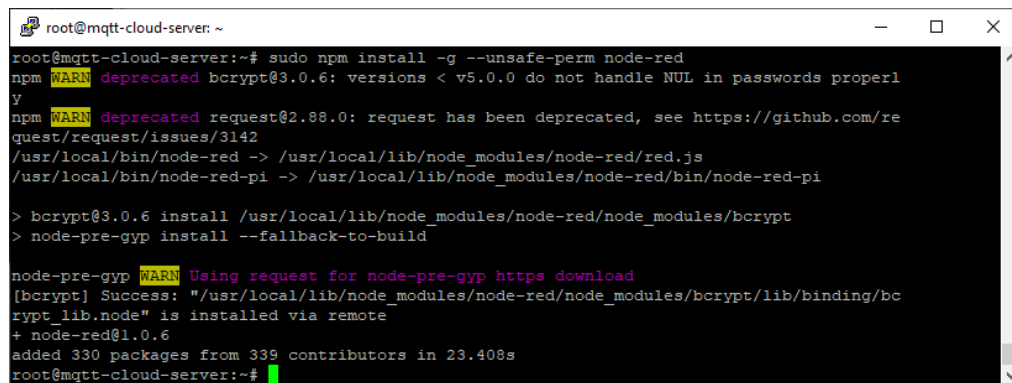
2. Abans d'instal·lar node-red, instal·lem npm:

```
sudo apt install npm -y
```

3. Instal·lem node-red:

```
sudo npm install -g --unsafe-perm node-red
```

Us hauria d'aparèixer una cosa semblant a aquesta:



```
root@mqtt-cloud-server: ~  
root@mqtt-cloud-server:~# sudo npm install -g --unsafe-perm node-red  
npm WARN deprecated bcrypt@3.0.6: versions < v5.0.0 do not handle NUL in passwords properly  
npm WARN deprecated request@2.88.0: request has been deprecated, see https://github.com/request/request/issues/3142  
/usr/local/bin/node-red -> /usr/local/lib/node_modules/node-red/red.js  
/usr/local/bin/node-red-pi -> /usr/local/lib/node_modules/node-red/bin/node-red-pi  
  
> bcrypt@3.0.6 install /usr/local/lib/node_modules/node-red/node_modules/bcrypt  
> node-pre-gyp install --fallback-to-build  
  
node-pre-gyp WARN Using request for node-pre-gyp https download  
[bcrypt] Success: "/usr/local/lib/node_modules/node-red/node_modules/bcrypt/lib/binding/bcrypt_lib.node" is installed via remote  
+ node-red@1.0.6  
added 330 packages from 339 contributors in 23.408s  
root@mqtt-cloud-server:~#
```

Fig 47: Annex. Captura de la consola després d'instal·lar node-red.

4. Si mai heu oberts els ports, per tema de jocs. Potser hauria d'obrir el port 1880 (port que utilitza node-red per defecte):

```
sudo ufw allow 1880
```

5. Finalment executem

```
node-red start
```

A continuació, ens hauria de sortir una cosa amb aparença semblant:

```
matu@matu-K050: ~  
Archivo Editar Ver Buscar Terminal Ayuda  
- /usr/local/lib/node_modules/node-red/red.js  
4 Dec 11:34:39 - [warn] -----  
4 Dec 11:34:39 - [info] Settings file   : /home/matu/.node-red/settings.js  
4 Dec 11:34:39 - [info] Context store  : 'default' [module=memory]  
4 Dec 11:34:39 - [info] User directory : /home/matu/.node-red  
4 Dec 11:34:39 - [warn] Projects disabled : editorTheme.projects.enabled=false  
4 Dec 11:34:39 - [info] Flows file      : /home/matu/.node-red/start  
4 Dec 11:34:39 - [warn] -----  
  
Your flow credentials file is encrypted using a system-generated key.  
  
If the system-generated key is lost for any reason, your credentials  
file will not be recoverable, you will have to delete it and re-enter  
your credentials.  
  
You should set your own key using the 'credentialSecret' option in  
your settings file. Node-RED will then re-encrypt your credentials  
file using your chosen key the next time you deploy a change.  
-----  
4 Dec 11:34:39 - [info] Server now running at http://127.0.0.1:1880/  
4 Dec 11:34:39 - [info] Starting flows  
4 Dec 11:34:40 - [info] Started flows
```

Fig 48: Annex. Captura de la consola després d'executar node-red.

Per accedir al panel de node-red, només caldrà cercar “<http://127.0.0.1:1800/>” en l'ordinador on s'està executant. En el meu cas, com que l'ordinador anava lent, vaig treballar amb un altre equip des de la meua xarxa domèstica cercant “<http://direccioIP:1880/>” normalment “<http://192.168.1.XX:1880/>”

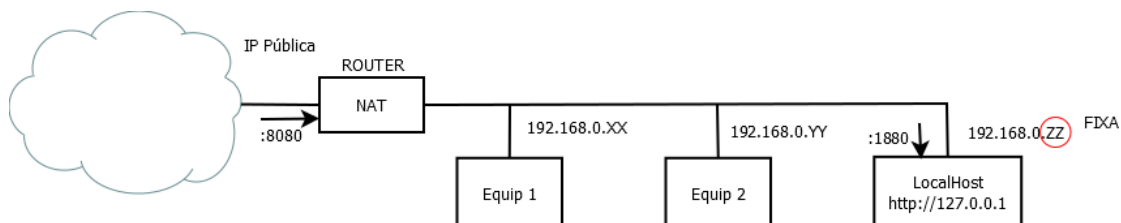


Fig X: Annex. Esquema red domestica.

En aquest esquema, ens resumeix la nostra xarxa domèstica en el port 8080, on per accedir al localhost del port 1880 des d'un altre equip, haurem d'editar l'ID host. En el cas de la imatge “ZZ”.

## D. ANNEX D.Instalar MQTT Mosquitto Broker en Ubuntu 20.04

Amb la terminal oberta, continuem les següents comandes:

1. Agregem els repositoris de Mosquitto dev:

```
sudo apt-add-repository ppa:mosquitto-dev/mosquitto-ppa
```

2. Actualitzem els repositoris de linux:

```
sudo apt-get update
```

3. Instal·lem Mosquitto Broker:

```
sudo apt-get install mosquitto
```

4. Instal·lem les llibreries de desenvolupament

```
sudo apt-get install libmosquitto-dev
```

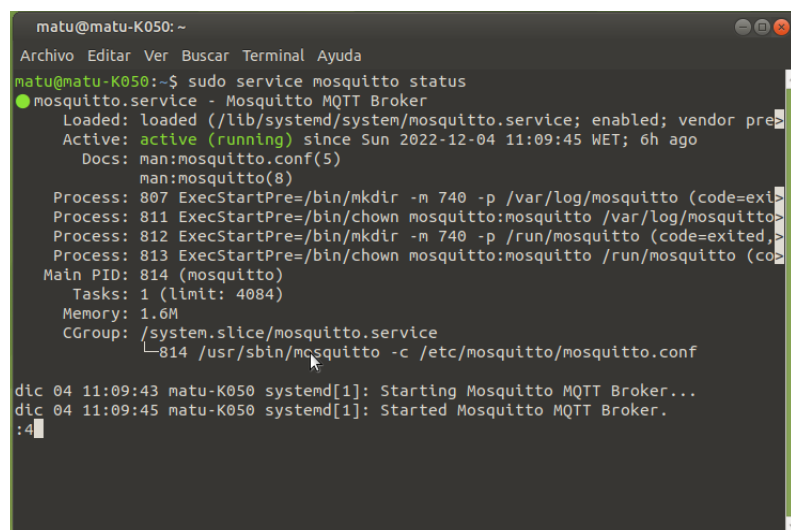
5. Instaleu el client MQTT

```
sudo apt-get install mosquitto-clients
```

6. Finalment, per comprovar si s'està executant:

```
sudo service mosquitto status
```

Si l'instal·lació ha estat correcta hauria de sortir:



```
matu@matu-K050: ~  
Archivo Editar Ver Buscar Terminal Ayuda  
matu@matu-K050:~$ sudo service mosquitto status  
● mosquitto.service - Mosquitto MQTT Broker  
   Loaded: loaded (/lib/systemd/system/mosquitto.service; enabled; vendor pre  
   Active: active (running) since Sun 2022-12-04 11:09:45 WET; 6h ago  
     Docs: man:mosquitto.conf(5)  
           man:mosquitto(8)  
   Process: 807 ExecStartPre=/bin/mkdir -m 740 -p /var/log/mosquitto (code=exi  
   Process: 811 ExecStartPre=/bin/chown mosquitto:mosquitto /var/log/mosquitto  
   Process: 812 ExecStartPre=/bin/mkdir -m 740 -p /run/mosquitto (code=exited,  
   Process: 813 ExecStartPre=/bin/chown mosquitto:mosquitto /run/mosquitto (co  
   Main PID: 814 (mosquitto)  
     Tasks: 1 (limit: 4084)  
    Memory: 1.6M  
    CGroup: /system.slice/mosquitto.service  
            └─814 /usr/sbin/mosquitto -c /etc/mosquitto/mosquitto.conf  
  
dic 04 11:09:43 matu-K050 systemd[1]: Starting Mosquitto MQTT Broker...  
dic 04 11:09:45 matu-K050 systemd[1]: Started Mosquitto MQTT Broker.  
:4
```

Fig 49: Annex. Captura de la consola després de veure l'estat de mosquitto.

Per acabar de configurar el nostre servidor:

1. Obrim l'archiu "*mosquitto.conf*"

```
sudo nano /etc/mosquitto/mosquitto.conf
```

2. Añadim les següents línies al final de l'archiu, com la captura.

```
listener 1883
allow_anonymous true
```

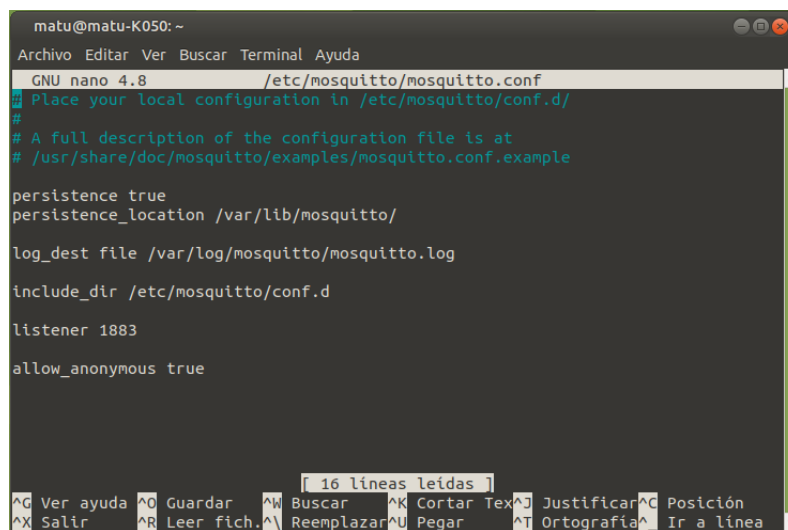


Fig 50: Annex. Captura de la consola de l'archiu "*mosquitto.conf*".

3. Desem i reiniciem el servidor:

```
sudo systemctl restart mosquitto
```

## E. ANNEX E. Configuració de l'entorn Arduino IDE

Després d'instal·lar Arduino, haurem de fer una petita configuració per utilitzar la nostra placa Wemos.

1. Enganxem l'enllaç “[https://dl.espressif.com/dl/package\\_esp32\\_index.json](https://dl.espressif.com/dl/package_esp32_index.json)” en la secció “Archivo”>”Preferencias”

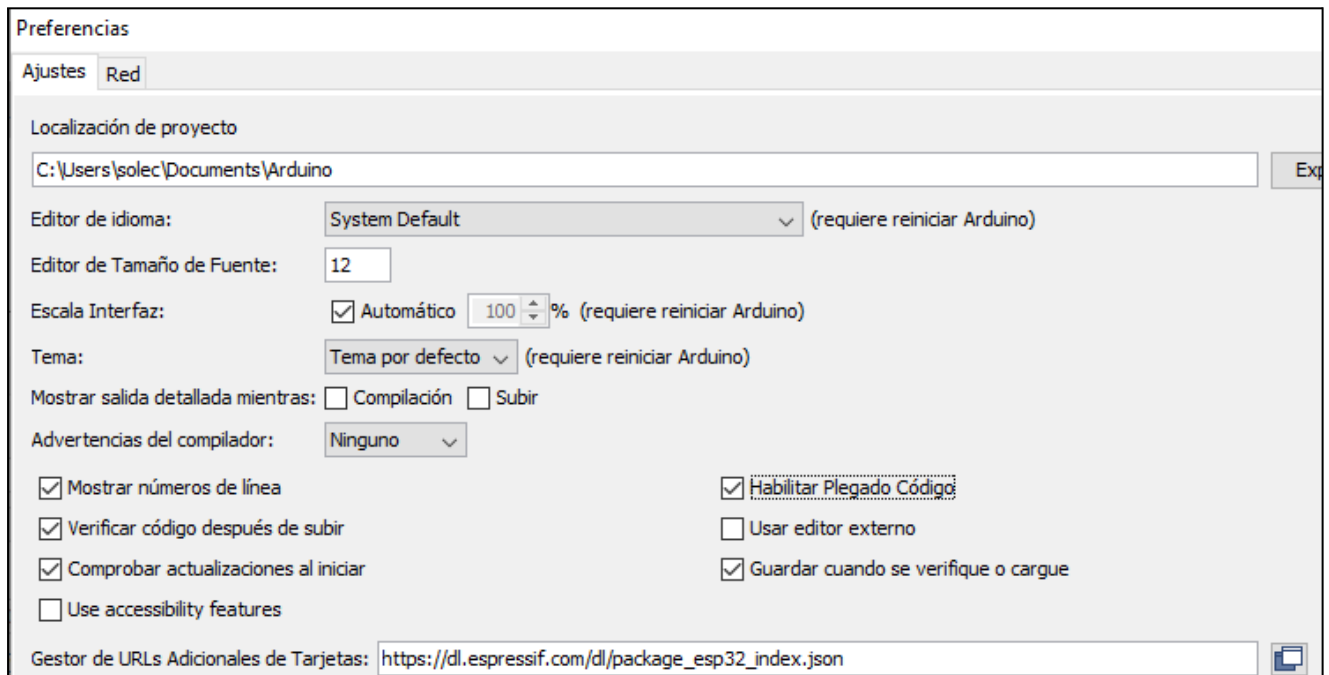


Fig 51: Annex. Configuració de l'entorn Arduino IDE 1.

2. Un cop fet això, instal·lem dins de “Herramientas”>”Placa”>”Gestor de tarjetas” i instal·lem “esp32”

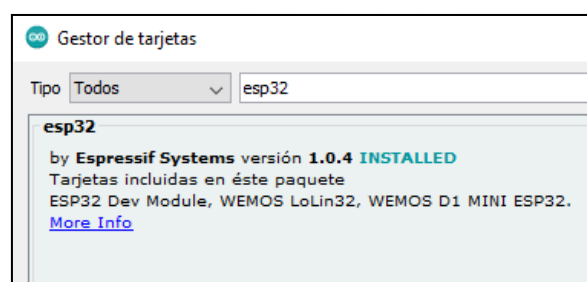


Fig 52: Annex. Configuració de l'entorn Arduino IDE 2.

3. Finalment, seleccionem el nom de la nostra placa dins de “Herramientas” i el port que utilitzarem. En el nostre cas “LOLIN(WEMOS) D1 R2 & mini”

**F. ANNEX F.Plantilla peces de metacrilat a mida real**

- Peça “BOTDUINO”

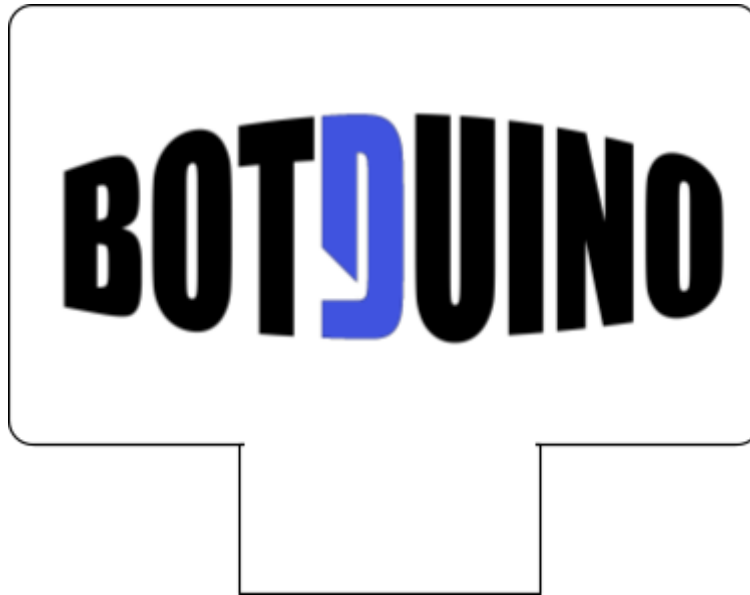


Fig 53: Software. Node-RED 1.

- Peça “D”

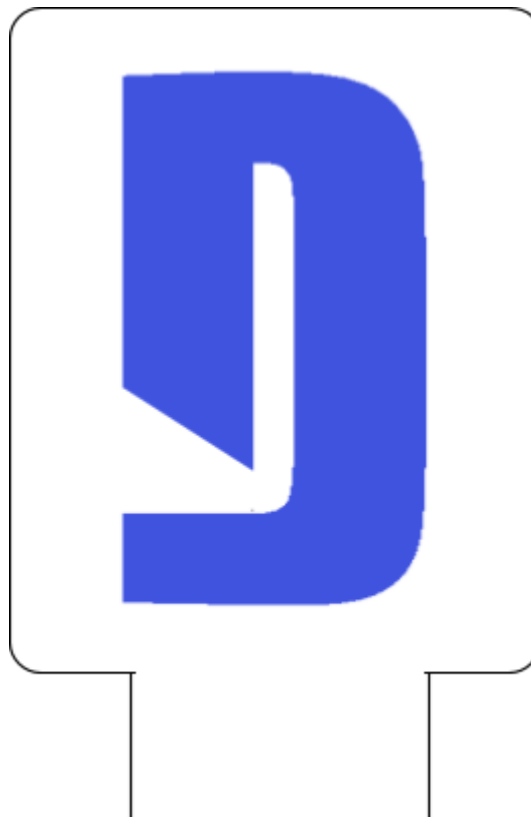


Fig 54: Software. Node-RED 1.

## G. ANNEX G.Codi exemple relé

En aquest exemple encén el relé durant dos segons i l'apaga en el mateix temps repetidament.

```
const int relayPin = D1;
const long interval = 2000; // pause for two seconds

void setup() {
  pinMode(relayPin, OUTPUT);
}

void loop() {
  digitalWrite(relayPin, HIGH); // turn on relay with voltage HIGH
  delay(interval);             // pause
  digitalWrite(relayPin, LOW);  // turn off relay with voltage LOW
  delay(interval);             // pause
}
```

## H. ANNEX H.Codi exemple polsador

En aquest exemple quan es presiona el polsador, el LED s'encén.

```
const int buttonPin = D3;
const int ledPin = BUILTIN_LED;

int buttonState = 0;

void setup() {
  pinMode(buttonPin, INPUT);
  pinMode(ledPin, OUTPUT);

  // set initial state, LED off
  digitalWrite(ledPin, buttonState);
}

void loop() {
  // read button state, HIGH when pressed, LOW when not
  buttonState = digitalRead(buttonPin);

  // if the push button pressed, switch on the LED
  if (buttonState == HIGH) {
    digitalWrite(ledPin, HIGH); // LED on
  } else {
    digitalWrite(ledPin, LOW); // LED off
  }
}
```



## I. ANNEX I.Codi exemple brunzidor

En aquest exemple farà una melodia durant uns segons, després estarà en silenci cinc segons i així repetidament:

```
int buzzer=D5; //Buzzer control port, default D5
void setup() {
  pinMode(buzzer, OUTPUT);
  digitalWrite(buzzer, LOW);
}
void loop() {
  for (int i=0;i<5;i++){
    analogWriteFreq(988); // 988 Hz, nota Do
    analogWrite(buzzer, 512); // ona simètrica
    delay(700); // 0,7 s durada
    analogWrite(buzzer, 0); // silenci
    delay(50); // 0,05 s
    analogWriteFreq(587); // 587 Hz, nota Si
    analogWrite(buzzer, 512);
    delay(700);
    analogWrite(buzzer, 0);
    delay(50);
  }
  pinMode(buzzer, OUTPUT);
  digitalWrite(buzzer, LOW); // silenci
  delay(5000);
}
```

## J. ANNEX J.Codi exemple led RGB

En aquest exemple el LED, va canviant de color .

```
#include <Adafruit_NeoPixel.h>

#define PIN    D4
#define LED_NUM 7

Adafruit_NeoPixel leds = Adafruit_NeoPixel(LED_NUM, PIN, NEO_GRB +
NEO_KHZ800);

void setup() {
  leds.begin(); // This initializes the NeoPixel library.
}

void led_set(uint8 R, uint8 G, uint8 B) {
  for (int i = 0; i < LED_NUM; i++) {
    leds.setPixelColor(i, leds.Color(R, G, B));
    leds.show();
    delay(50);
  }
}

void loop() {
```

```
    led_set(10, 0, 0); //red
    led_set(0, 0, 0);

    led_set(0, 10, 0); //green
    led_set(0, 0, 0);

    led_set(0, 0, 10); //blue
    led_set(0, 0, 0);

}

}
```

## K. ANNEX K.Codi Arduino

```
#include <ESP8266WiFi.h>
#include <PubSubClient.h>
//https://www.arduino.cc/reference/en/libraries/pubsubclient/
#include <Adafruit_NeoPixel.h>
//https://github.com/adafruit/Adafruit_NeoPixel
#include "notes.h"

const int buzzer = D5;
const int button = D3;
const int relay = D1;
const int RGB = D2;

//Configurem la llibreria dient, el nombre de píxels i el pin.
//Atenció: Les tires de NeoPixel més antigues és possible que hagueu de
canviar el tercer paràmetre: consulteu la prova de femella!
Adafruit_NeoPixel pixels = Adafruit_NeoPixel(1, RGB, NEO_GRB +
NEO_KHZ800);

//Canvieu les credencials segons el seu encaminador i el seu agent MQTT
const char* ssid      = "POSAR_SSID";
const char* password = "POSAR_CONTRASENYA";
const char* mqtt_server = "DIRECCIÓ_IP_SERVER_MQTT";

//Inicialitza l'espClient. Hauríeu de canviar el nom de l'espClient si
teniu diversos ESP en execució al vostre sistema domòtic
WiFiClient espClient;
PubSubClient client(espClient);

String _topic;
String _payload;

void setup_wifi() {
    delay(10);
    Serial.println();
    Serial.print("Conectant a ");
    Serial.println(ssid);
    WiFi.begin(ssid, password);
    while (WiFi.status() != WL_CONNECTED) {
        delay(500);
        Serial.print(".");
    }
    Serial.println("");
}
```

```
Serial.print("Wifi conectat! - ESP Direcció IP: ");
Serial.println(WiFi.localIP());
}

void reconnect() {
  while (!client.connected()) {
    Serial.print("Intentant connexió MQTT ..");
    String clientId = "ESP8266Client-";
    clientId += String(random(0xffff), HEX);
    if (client.connect("ESP8266Client")) {
      Serial.println("Conectat!");
      client.subscribe("dlmini/lampada");
    } else {
      Serial.print("falla, estado=");
      Serial.print(client.state());
      Serial.println("Intentant connectar novament en 5 segons");
      delay(5000);
    }
  }
}

void callback(char* topic, byte* payload, unsigned int length) {
  Serial.print("Ha arribat missatge sobre el topic: ");
  Serial.print(topic);
  Serial.print(". Missatge: ");
  String conc_payload_;
  for (int i = 0; i < length; i++) {
    Serial.print((char)payload[i]);
    conc_payload_ += (char)payload[i];
  }
  Serial.println();
  _topic = topic;
  _payload = conc_payload_;
}

void setup() {
  pinMode(relay, OUTPUT);
  pixels.begin(); //Iniciem biblioteca NeoPixel
  pinMode(buzzer, OUTPUT);
  digitalWrite(buzzer, LOW);
  analogWriteRange(1023);
  Serial.begin(115200);
  setup_wifi();
  client.setServer(mqtt_server, 1883);
  client.setCallback(callback);
}

void loop() {
  if (!client.connected()) {
    reconnect();
  }
  if(!client.loop())
    client.connect("ESP8266Client");

  //Si pressionem el polsador, reiniciem el wifi
  if (digitalRead(button) == LOW){
    ESP.restart();
  }

  //Si arriba el missatge "partit"
```

```
if (_payload == "partit"){
  pixels.setPixelColor(0, pixels.Color(0, 0, 255)); //Color Blau
  pixels.show(); //Actualitza el color
  digitalWrite(relay, HIGH); //Encendre el Relé

  //Melodia de Pac-Man
  tone(buzzer, Si);
  delay(150);
  noTone(buzzer);
  tone(buzzer, SiH);
  delay(150);
  noTone(buzzer);
  tone(buzzer, FaHs);
  delay(150);
  noTone(buzzer);
  tone(buzzer, ReHs);
  delay(150);
  noTone(buzzer);
  tone(buzzer, SiH);
  delay(150);
  noTone(buzzer);
  tone(buzzer, FaHs);
  delay(150);
  noTone(buzzer);
  tone(buzzer, ReHs);
  delay(450);
  noTone(buzzer);
  tone(buzzer, 523);
  delay(150);
  noTone(buzzer);
  tone(buzzer, 1047);
  delay(150);
  noTone(buzzer);
  tone(buzzer, SolH);
  delay(150);
  noTone(buzzer);
  tone(buzzer, MiH);
  delay(150);
  noTone(buzzer);
  tone(buzzer, 1047);
  delay(150);
  noTone(buzzer);
  tone(buzzer, SolH);
  delay(150);
  noTone(buzzer);
  tone(buzzer, MiH);
  delay(212);
  noTone(buzzer);
  tone(buzzer, Si);
  delay(150);
  noTone(buzzer);
  tone(buzzer, SiH);
  delay(150);
  noTone(buzzer);
  tone(buzzer, FaHs);
  delay(150);
  noTone(buzzer);
  tone(buzzer, ReHs);
  delay(150);
  noTone(buzzer);
```

```
tone(buzzer, SiH);
delay(150);
noTone(buzzer);
tone(buzzer, FaHs);
delay(150);
noTone(buzzer);
tone(buzzer, ReHs);
delay(212);
noTone(buzzer);
tone(buzzer, ReHs);
delay(150);
noTone(buzzer);
tone(buzzer, MiH);
delay(150);
noTone(buzzer);
tone(buzzer, FaH);
delay(150);
noTone(buzzer);
tone(buzzer, FaH);
delay(150);
noTone(buzzer);
tone(buzzer, FaHs);
delay(150);
noTone(buzzer);
tone(buzzer, SolH);
delay(150);
noTone(buzzer);
tone(buzzer, SolH);
delay(150);
noTone(buzzer);
tone(buzzer, 831);
delay(150);
noTone(buzzer);
tone(buzzer, LaH);
delay(150);
noTone(buzzer);
tone(buzzer, SiH);
delay(300);
tone(buzzer, Si);
delay(150);
noTone(buzzer);
tone(buzzer, SiH);
delay(150);
noTone(buzzer);
tone(buzzer, FaHs);
delay(150);
noTone(buzzer);
tone(buzzer, ReHs);
delay(150);
noTone(buzzer);
tone(buzzer, SiH);
delay(150);
noTone(buzzer);
tone(buzzer, FaHs);
delay(150);
noTone(buzzer);
tone(buzzer, ReHs);
delay(450);
noTone(buzzer);
tone(buzzer, 523);
```

```
delay(150);  
noTone(buzzer);  
tone(buzzer, 1047);  
delay(150);  
noTone(buzzer);  
tone(buzzer, SolH);  
delay(150);  
noTone(buzzer);  
tone(buzzer, MiH);  
delay(150);  
noTone(buzzer);  
tone(buzzer, 1047);  
delay(150);  
noTone(buzzer);  
tone(buzzer, SolH);  
delay(150);  
noTone(buzzer);  
tone(buzzer, MiH);  
delay(450);  
noTone(buzzer);  
tone(buzzer, Si);  
delay(150);  
noTone(buzzer);  
tone(buzzer, SiH);  
delay(150);  
noTone(buzzer);  
tone(buzzer, FaHs);  
delay(150);  
noTone(buzzer);  
tone(buzzer, ReHs);  
delay(150);  
noTone(buzzer);  
tone(buzzer, SiH);  
delay(150);  
noTone(buzzer);  
tone(buzzer, FaHs);  
delay(150);  
noTone(buzzer);  
tone(buzzer, ReHs);  
delay(212);  
noTone(buzzer);  
tone(buzzer, ReHs);  
delay(150);  
noTone(buzzer);  
tone(buzzer, MiH);  
delay(150);  
noTone(buzzer);  
tone(buzzer, FaH);  
delay(150);  
noTone(buzzer);  
tone(buzzer, FaH);  
delay(150);  
noTone(buzzer);  
tone(buzzer, FaHs);  
delay(150);  
noTone(buzzer);  
tone(buzzer, SolH);  
delay(150);  
noTone(buzzer);  
tone(buzzer, SolH);
```

```
delay(150);
noTone(buzzer);
tone(buzzer, 831);
delay(150);
noTone(buzzer);
tone(buzzer, LaH);
delay(150);
noTone(buzzer);
tone(buzzer, SiH);

digitalWrite(relay, LOW);
pixels.setPixelColor(0, pixels.Color(0, 0, 0));
pixels.show();
digitalWrite(buzzer, LOW);
_payload = "";
}

//Si arriba el missatge "jugar"
if (_payload == "jugar"){
  pixels.setPixelColor(0, pixels.Color(0, 255, 0)); //Color verd
  pixels.show();
  digitalWrite(relay, HIGH);

  //Melodia de Star Wars
  tone(buzzer, La, 500);
  delay(500+50);
  tone(buzzer, La, 500);
  delay(500+50);
  tone(buzzer, La, 500);
  delay(500+50);
  tone(buzzer, Fa, 350);
  delay(350+50);
  tone(buzzer, DoH, 150);
  delay(150+50);
  tone(buzzer, La, 500);
  delay(500+50);
  tone(buzzer, Fa, 350);
  delay(350+50);
  tone(buzzer, DoH, 150);
  delay(150+50);
  tone(buzzer, La, 1000);
  delay(1000+50);
  tone(buzzer, MiH, 500);
  delay(500+50);
  tone(buzzer, MiH, 500);
  delay(500+50);
  tone(buzzer, MiH, 500);
  delay(500+50);
  tone(buzzer, FaH, 350);
  delay(350+50);
  tone(buzzer, DoH, 150);
  delay(150+50);
  tone(buzzer, Sols, 500);
  delay(500+50);
  tone(buzzer, Fa, 350);
  delay(350+50);
  tone(buzzer, DoH, 150);
  delay(150+50);
  tone(buzzer, La, 1000);
  delay(1000+50);
```

```
tone(buzzer, LaH, 500);
delay(500+50);
tone(buzzer, La, 350);
delay(350+50);
tone(buzzer, La, 150);
delay(150+50);
tone(buzzer, LaH, 500);
delay(500+50);
tone(buzzer, SolHs, 250);
delay(250+50);
tone(buzzer, SolH, 250);
delay(250+50);
tone(buzzer, FaHs, 125);
delay(125+50);
tone(buzzer, FaH, 125);
delay(125+50);
tone(buzzer, FaHs, 250);
delay(250+50);
delay(250);
tone(buzzer, Las, 250);
delay(250+50);
tone(buzzer, ReHs, 500);
delay(500+50);

digitalWrite(relay, LOW);
pixels.setPixelColor(0, pixels.Color(0, 0, 0));
pixels.show();
digitalWrite(buzzer, LOW);
_payload = "";
}

//Si arriba el missatge "escalfar"
if (_payload == "escalfar"){
  pixels.setPixelColor(0, pixels.Color(255, 0, 0)); //Color vermell
  pixels.show();
  digitalWrite(relay, HIGH);

//Melodia pirates del carib
tone(buzzer,Re,100);
delay(100);
tone(buzzer,Re,200);
delay(200);
tone(buzzer,Re,100);
delay(100);
tone(buzzer,Re,200);
delay(200);
tone(buzzer,Re,100);
delay(100);
tone(buzzer,Re,100);
delay(100);
tone(buzzer,Re,100);
delay(100);
tone(buzzer,Re,100);
delay(100);
tone(buzzer,Re,200);
delay(200);
tone(buzzer,Re,100);
delay(100);
tone(buzzer,Re,200);
delay(200);
```



```
tone(buzzer,Re,100);  
delay(100);  
tone(buzzer,Re,200);  
delay(200);  
tone(buzzer,Re,100);  
delay(100);  
tone(buzzer,Re,100);  
delay(100);  
tone(buzzer,Re,100);  
delay(100);  
tone(buzzer,Re,100);  
delay(100);  
tone(buzzer,Re,200);  
delay(200);  
tone(buzzer,Re,100);  
delay(100);  
tone(buzzer,Re,200);  
delay(200);  
tone(buzzer,Re,100);  
delay(100);  
tone(buzzer,Re,200);  
delay(200);  
tone(buzzer,Re,100);  
delay(100);  
tone(buzzer,Re,100);  
delay(100);  
tone(buzzer,La,100);  
delay(100);  
tone(buzzer,DoH,100);  
delay(100);  
tone(buzzer,ReH,100);  
delay(200);  
tone(buzzer,ReH,100);  
delay(200);  
tone(buzzer,ReH,100);  
delay(100);  
tone(buzzer,MiH,100);  
delay(100);  
tone(buzzer,FaH,100);  
delay(200);  
tone(buzzer,FaH,100);  
delay(200);  
tone(buzzer,FaH,100);  
delay(100);  
tone(buzzer,784,100);  
delay(100);  
tone(buzzer,MiH,100);  
delay(200);  
tone(buzzer,MiH,100);  
delay(200);  
tone(buzzer,ReH,100);  
delay(100);  
tone(buzzer,DoH,100);  
delay(100);  
tone(buzzer,DoH,100);  
delay(100);  
tone(buzzer,ReH,100);  
delay(300);  
tone(buzzer,La,100);  
delay(100);
```

```
tone(buzzer,DoH,100);  
delay(100);  
tone(buzzer,ReH,100);  
delay(200);  
tone(buzzer,ReH,100);  
delay(200);  
tone(buzzer,ReH,100);  
delay(100);  
tone(buzzer,MiH,100);  
delay(100);  
tone(buzzer,FaH,100);  
delay(200);  
tone(buzzer,FaH,100);  
delay(200);  
tone(buzzer,FaH,100);  
delay(100);  
tone(buzzer,784,100);  
delay(100);  
tone(buzzer,MiH,100);  
delay(200);  
tone(buzzer,MiH,100);  
delay(200);  
tone(buzzer,ReH,100);  
delay(100);  
tone(buzzer,DoH,100);  
delay(100);  
tone(buzzer,ReH,100);  
delay(400);  
tone(buzzer,La,100);  
delay(100);  
tone(buzzer,DoH,100);  
delay(100);  
tone(buzzer,ReH,100);  
delay(200);  
tone(buzzer,ReH,100);  
delay(200);  
tone(buzzer,ReH,100);  
delay(100);  
tone(buzzer,FaH,100);  
delay(100);  
tone(buzzer,784,100);  
delay(200);  
tone(buzzer,784,100);  
delay(200);  
tone(buzzer,784,100);  
delay(100);  
tone(buzzer,LaH,100);  
delay(100);  
tone(buzzer,932,100);  
delay(200);  
tone(buzzer,932,100);  
delay(200);  
tone(buzzer,LaH,100);  
delay(100);  
tone(buzzer,784,100);  
delay(100);  
tone(buzzer,LaH,100);  
delay(100);  
tone(buzzer,ReH,100);  
delay(300);
```

```
tone(buzzer,ReH,100);
delay(100);
tone(buzzer,MiH,100);
delay(100);
tone(buzzer,FaH,100);
delay(200);
tone(buzzer,FaH,100);
delay(200);
tone(buzzer,784,100);
delay(200);
tone(buzzer,LaH,100);
delay(100);
tone(buzzer,ReH,100);
delay(300);
tone(buzzer,ReH,100);
delay(100);
tone(buzzer,FaH,100);
delay(100);
tone(buzzer,MiH,100);
delay(200);
tone(buzzer,MiH,100);
delay(200);
tone(buzzer,FaH,100);
delay(100);
tone(buzzer,ReH,100);
delay(100);
tone(buzzer,MiH,100);
delay(400);

digitalWrite(relay, LOW);
pixels.setPixelColor(0, pixels.Color(0, 0, 0));
pixels.show();
digitalWrite(buzzer, LOW);
_payload = "";

}

}
```

## L. ANNEX L.Codi node-red

```
[
  {
    "id": "a1ca39e0415da0d9",
    "type": "tab",
    "label": "BOTDUINO",
    "disabled": false,
    "info": "",
    "env": []
  },
  {
    "id": "1085b288f8eb3e5d",
    "type": "comment",
    "z": "a1ca39e0415da0d9",
    "name": "PARTIT",
    "info": "",
    "x": 170,
```

```

        "y": 80,
        "wires": []
    },
    {
        "id": "56d512d8ae20e9f0",
        "type": "discordInteraction",
        "z": "alca39e0415da0d9",
        "name": "Partit",
        "token": "",
        "interactionType": "command",
        "custom_id": "partit",
        "commandResponse": "Avisant que hi ha partit als altres
jugadors...",
        "interactionObject": false,
        "ephemeral": false,
        "x": 150,
        "y": 200,
        "wires": [
            [
                "e8e570b1db1c079a"
            ]
        ]
    },
    {
        "id": "e8e570b1db1c079a",
        "type": "change",
        "z": "alca39e0415da0d9",
        "name": "",
        "rules": [
            {
                "t": "set",
                "p": "payload",
                "pt": "msg",
                "to": "partit",
                "tot": "str"
            }
        ],
        "action": "",
        "property": "",
        "from": "",
        "to": "",
        "reg": false,
        "x": 320,
        "y": 200,
        "wires": [
            [
                "c0c7d6dd1f6aacd3",
                "5930f4aebd3c8b25"
            ]
        ]
    },
    {
        "id": "c0c7d6dd1f6aacd3",
        "type": "mqtt out",
        "z": "alca39e0415da0d9",
        "name": "",
        "topic": "dlmini/lampada",
        "qos": "",
        "retain": "",
        "respTopic": "",

```

```

        "contentType": "",
        "userProps": "",
        "correl": "",
        "expiry": "",
        "broker": "10e78a89.5b4fd5",
        "x": 520,
        "y": 160,
        "wires": []
    },
    {
        "id": "5930f4aebd3c8b25",
        "type": "debug",
        "z": "alca39e0415da0d9",
        "name": "debug",
        "active": true,
        "tosidebar": true,
        "console": false,
        "tostatus": false,
        "complete": "payload",
        "targetType": "msg",
        "statusVal": "",
        "statusType": "auto",
        "x": 490,
        "y": 240,
        "wires": []
    },
    {
        "id": "336004816ad270f0",
        "type": "comment",
        "z": "alca39e0415da0d9",
        "name": "ESCALFAR",
        "info": "",
        "x": 190,
        "y": 340,
        "wires": []
    },
    {
        "id": "9ca3996fb0b3c88d",
        "type": "discordInteraction",
        "z": "alca39e0415da0d9",
        "name": "Escalfar",
        "token": "",
        "interactionType": "command",
        "custom_id": "escalfar",
        "commandResponse": "Avisant d'escalfar als altres jugadors...",
        "interactionObject": false,
        "ephemeral": false,
        "x": 160,
        "y": 460,
        "wires": [
            [
                "5552470dabb647e3"
            ]
        ]
    },
    {
        "id": "5552470dabb647e3",
        "type": "change",
        "z": "alca39e0415da0d9",
        "name": "",

```

```
    "rules": [
      {
        "t": "set",
        "p": "payload",
        "pt": "msg",
        "to": "escalfar",
        "tot": "str"
      }
    ],
    "action": "",
    "property": "",
    "from": "",
    "to": "",
    "reg": false,
    "x": 320,
    "y": 460,
    "wires": [
      [
        "006ae157af233471",
        "566b106e8c057a72"
      ]
    ]
  },
  {
    "id": "006ae157af233471",
    "type": "mqtt out",
    "z": "alca39e0415da0d9",
    "name": "",
    "topic": "dlmini/lampada",
    "qos": "",
    "retain": "",
    "respTopic": "",
    "contentType": "",
    "userProps": "",
    "correl": "",
    "expiry": "",
    "broker": "10e78a89.5b4fd5",
    "x": 520,
    "y": 420,
    "wires": []
  },
  {
    "id": "566b106e8c057a72",
    "type": "debug",
    "z": "alca39e0415da0d9",
    "name": "debug",
    "active": true,
    "tosidebar": true,
    "console": false,
    "tostatus": false,
    "complete": "payload",
    "targetType": "msg",
    "statusVal": "",
    "statusType": "auto",
    "x": 490,
    "y": 500,
    "wires": []
  },
  {
    "id": "9fb6f95ada39cb1d",
```

```
    "type": "comment",
    "z": "alca39e0415da0d9",
    "name": "JUGAR",
    "info": "",
    "x": 170,
    "y": 580,
    "wires": []
  },
  {
    "id": "0db3218e35d63bd4",
    "type": "discordInteraction",
    "z": "alca39e0415da0d9",
    "name": "Jugar",
    "token": "",
    "interactionType": "command",
    "custom_id": "jugar",
    "commandResponse": "Avisant per jugar als altres jugadors...",
    "interactionObject": false,
    "ephemeral": false,
    "x": 150,
    "y": 700,
    "wires": [
      [
        "d3bc76d727b1ac95"
      ]
    ]
  },
  {
    "id": "d3bc76d727b1ac95",
    "type": "change",
    "z": "alca39e0415da0d9",
    "name": "",
    "rules": [
      {
        "t": "set",
        "p": "payload",
        "pt": "msg",
        "to": "jugar",
        "tot": "str"
      }
    ],
    "action": "",
    "property": "",
    "from": "",
    "to": "",
    "reg": false,
    "x": 320,
    "y": 700,
    "wires": [
      [
        "7654553fd588a7e1",
        "09575dbc65c0412a"
      ]
    ]
  },
  {
    "id": "7654553fd588a7e1",
    "type": "mqtt out",
    "z": "alca39e0415da0d9",
    "name": "",
```

```
    "topic": "dlmini/lampada",
    "qos": "",
    "retain": "",
    "respTopic": "",
    "contentType": "",
    "userProps": "",
    "correl": "",
    "expiry": "",
    "broker": "10e78a89.5b4fd5",
    "x": 520,
    "y": 660,
    "wires": []
  },
  {
    "id": "09575dbc65c0412a",
    "type": "debug",
    "z": "a1ca39e0415da0d9",
    "name": "debug",
    "active": true,
    "tosidebar": true,
    "console": false,
    "tostatus": false,
    "complete": "payload",
    "targetType": "msg",
    "statusVal": "",
    "statusType": "auto",
    "x": 490,
    "y": 740,
    "wires": []
  },
  {
    "id": "10e78a89.5b4fd5",
    "type": "mqtt-broker",
    "name": "",
    "broker": "localhost",
    "port": "1883",
    "clientId": "",
    "usetls": false,
    "compatmode": true,
    "keepalive": "60",
    "cleansession": true,
    "birthTopic": "",
    "birthQos": "0",
    "birthPayload": "",
    "closeTopic": "",
    "closeQos": "0",
    "closePayload": "",
    "willTopic": "",
    "willQos": "0",
    "willPayload": ""
  }
]
```



## M. ANNEX M.Creació i configuració de fluxos

- Fluxos *inject* (partit, escalfar, jugar):

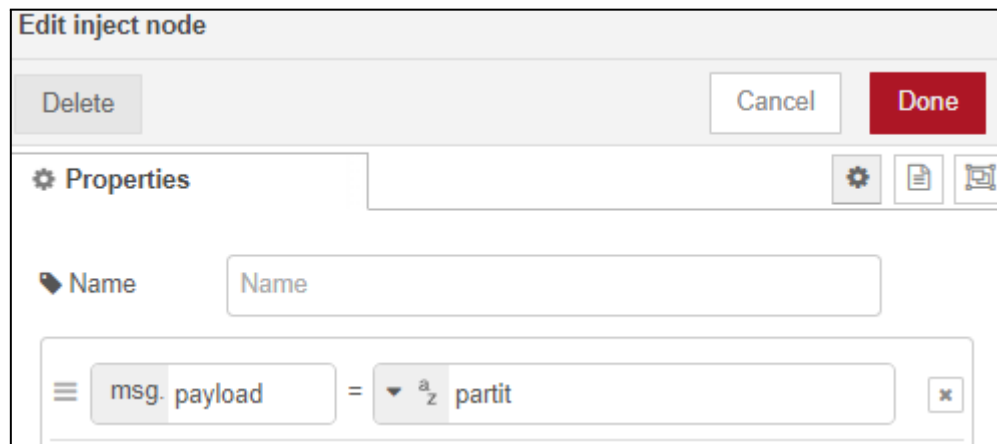


Fig 55: Software. Node-RED 1.

- Flux *mqtt out* (d1mini/lampada):

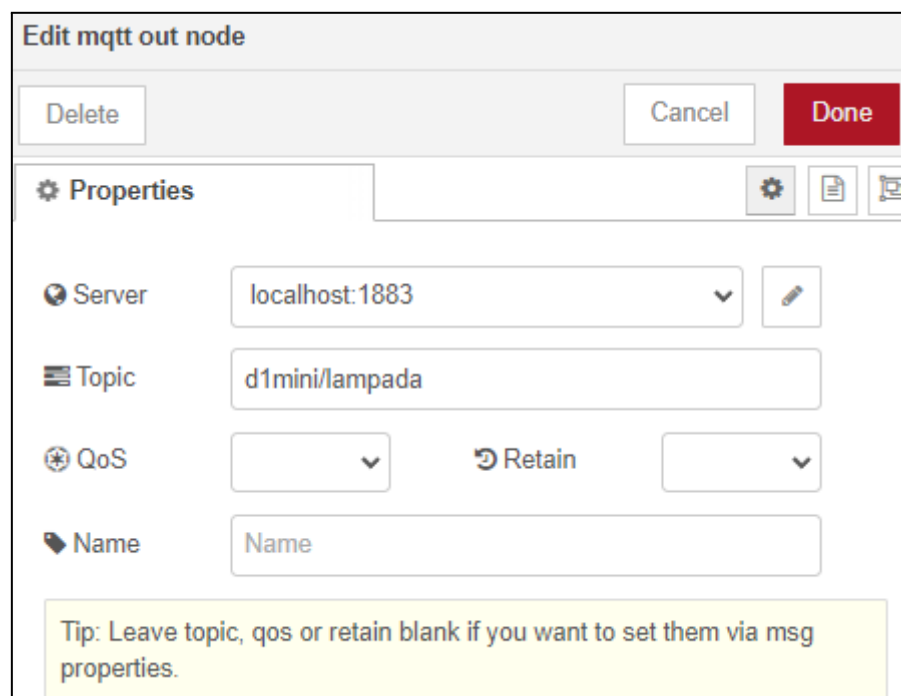


Fig 56: Software. Node-RED 1.

- Flux *discordInteraction* (partit, escalfar, jugar):

**Edit discordInteraction node**

Delete Cancel Done

**Properties**

token BotDuino

Name Partit

Interaction type Command

Command names partit  
(leave blank to not filter by custom\_id/commandName, or separated by ',' for more than one custom\_id/commandName)

Command Response text Avisant que hi ha partit als altres jugadors...

☐ Ephemeral

☐ Inject interaction object

Fig 57: Software. Node-RED 1.

- Flux change (set msg.payload):

**Edit change node**

Delete Cancel Done

**Properties**

Name Name

**Rules**

Set msg.payload

to the value a\_z partit

Fig 58: Software. Node-RED 1.

**N. ANNEX N.Taula de freqüències de les notes utilitzades (Hz)**

Octava:	0	1	2	3	4	5	6	7	8
Do	16	33	65	131	261	523	1047	2093	4186
Do#	17	35	69	139	277	554	1109	2217	4435
Re	18	37	73	147	294	587	1175	2349	4699
Re#	19	39	78	156	311	622	1245	2489	4978
Mi	21	41	82	165	329	659	1319	2637	5274
Fa	22	44	87	175	349	698	1397	2794	5588
Fa#	23	46	92	185	370	740	1480	2960	5920
Sol	24	49	98	196	391	784	1568	3136	6272
Sol#	26	52	104	208	415	830	1661	3520	6645
La	28	55	110	220	440	880	1760	3729	7040
La#	29	58	117	233	466	932	1865	3951	7459
Si	31	62	123	246	494	988	1976	3941	7902

\*Els quadres de color, són les freqüències utilitzades en aquest projecte.

**O. ANNEX O.Codi notes.h**

```
/*  
Octava 5 amb "H" (High)  
Octava 6 amb "Hh"  
Notes sostingudes amb "s"  
*/  
  
int Do=261;  
int DoH=523;  
int DoHh=1047;  
int DoHs=554;  
int Re=294;  
int ReH=587;  
int ReHs=622;  
int Mi=329;  
int MiH=659;  
int Fa=349;  
int FaH=698;  
int FaHs=740;  
int Sol=391;  
int SolH=784;  
int Sols=415;  
int SolHs=830;  
int La=440;  
int LaH=880;  
int Las=466;  
int LaHs=932;  
int Si=494;  
int SiH=988;
```

# 13.BIBLIOGRAFIA

## 13.1.Fonts consultades

- Discord:
  - Que és discord i com crear servidors: <https://www.xataka.com/basics/primeros-pasos-discord-que-como-registrarte-como-crear-acceder-a-sus-servidores>
  - Crear aplicació i que és Discord API: [https://fusebit.io/blog/discord-rest-api/?utm\\_source=www.google.com&utm\\_medium=referral&utm\\_campaign=none](https://fusebit.io/blog/discord-rest-api/?utm_source=www.google.com&utm_medium=referral&utm_campaign=none)
- Arduino:
  - Què és arduino i com funciona: <https://www.xataka.com/basics/que-arduino-como-funciona-que-puedes-hacer-uno>
- Node-RED:
  - Que és node-red: <https://blog.gruposinelec.com/actualidad/que-es-node-red-y-para-que-sirve/>
  - fluxes: <https://nodered.org/docs/user-guide/nodes>
  - node-red-contrib-discord-advanced: <https://flows.nodered.org/node/node-red-contrib-discord-advanced>
- MQTT:
  - Què és MQTT: <https://randomnerdtutorials.com/what-is-mqtt-and-how-it-works/>
  - Llegir MQTT amb node-xarxa: <https://www.promotec.net/probando-mas-nodos-en-nodered/>
- Components i esquemes:
  - [https://www.wemos.cc/en/latest/d1\\_mini\\_shield/index.html](https://www.wemos.cc/en/latest/d1_mini_shield/index.html)
- Software:
  - melodia Star Wars: <https://inputmakers.com/componentes/melodia-de-star-wars-con-arduino-y-zumbador>
  - archiu midi Pac-Man: <https://www.khinsider.com/midi/arcade/pac-man>
  - Com resetear ESP: <https://circuits4you.com/2017/12/31/software-reset-esp8266/>
  - midi to arduino: <https://arduinomidi.netlify.app/>
- Pressupost:
  - raspberry pi 4: [https://www.amazon.es/STUUC-Raspberry-Actualizada-Ventilador-Interruptor/dp/B0B23VQ1YS/ref=sr\\_1\\_2\\_sspa?\\_\\_mk\\_es\\_ES=%C3%85M%C3%85%C5%BD%C3%95%C3%91&crd=16T41U8HZHJGZ&keywords=raspberrypi+4&qid=1670258088&srefix=raspberrypi+4%2Caps%2C119&sr=8-2-spons&sp\\_csd=d2lkZ2V0TmFtZT1zcF9hdGY&psc=1](https://www.amazon.es/STUUC-Raspberry-Actualizada-Ventilador-Interruptor/dp/B0B23VQ1YS/ref=sr_1_2_sspa?__mk_es_ES=%C3%85M%C3%85%C5%BD%C3%95%C3%91&crd=16T41U8HZHJGZ&keywords=raspberrypi+4&qid=1670258088&srefix=raspberrypi+4%2Caps%2C119&sr=8-2-spons&sp_csd=d2lkZ2V0TmFtZT1zcF9hdGY&psc=1)
  - tripler base: [https://es.aliexpress.com/item/1005003531822622.html?spm=a2q0o.order\\_list.order\\_list\\_main.11.7d2f194dKvTHCn&gatewayAdapt=glo2esp](https://es.aliexpress.com/item/1005003531822622.html?spm=a2q0o.order_list.order_list_main.11.7d2f194dKvTHCn&gatewayAdapt=glo2esp)
  - lolin d1 mini: [https://es.aliexpress.com/item/1005004508626229.html?spm=a2q0o.order\\_list.order\\_list\\_main.17.7d2f194dKvTHCn&gatewayAdapt=glo2esp](https://es.aliexpress.com/item/1005004508626229.html?spm=a2q0o.order_list.order_list_main.17.7d2f194dKvTHCn&gatewayAdapt=glo2esp)
  - buzzer shield: [https://es.aliexpress.com/item/1005004898475292.html?spm=a2q0o.productlist.0.0.4cb44ef6eY43W5&algo\\_pvid=7cc01520-8f2b-411e-922e-8b2f6bd01b68&aem\\_p4p\\_detail=202212071126474140802657958300002108617&algo\\_exp\\_id=7cc01520-8f2b-411e-922e-8b2f6bd01b68-0&pdp\\_ext\\_f=%7B%22sku\\_id%22%3A%2212000030941831214%22%7D&pdp\\_npi=2%40dis%21EUR%211.78%211.24%21%21%21%21%21%402100bb5116704412071463696ea795%2112000030941831214%21sea&curPageLogUid=f0ayPw44BLXV&ad\\_pvid=202212071126474140802657958300002108617\\_1&ad\\_pvid=20221207112647414080265795830002108617\\_1](https://es.aliexpress.com/item/1005004898475292.html?spm=a2q0o.productlist.0.0.4cb44ef6eY43W5&algo_pvid=7cc01520-8f2b-411e-922e-8b2f6bd01b68&aem_p4p_detail=202212071126474140802657958300002108617&algo_exp_id=7cc01520-8f2b-411e-922e-8b2f6bd01b68-0&pdp_ext_f=%7B%22sku_id%22%3A%2212000030941831214%22%7D&pdp_npi=2%40dis%21EUR%211.78%211.24%21%21%21%21%21%402100bb5116704412071463696ea795%2112000030941831214%21sea&curPageLogUid=f0ayPw44BLXV&ad_pvid=202212071126474140802657958300002108617_1&ad_pvid=20221207112647414080265795830002108617_1)
  - RGB shield: [https://es.aliexpress.com/item/32892154533.html?spm=a2q0o.order\\_list.order\\_list\\_main.4.7d2f194dKvTHCn&gatewayAdapt=glo2esp](https://es.aliexpress.com/item/32892154533.html?spm=a2q0o.order_list.order_list_main.4.7d2f194dKvTHCn&gatewayAdapt=glo2esp)
  - altres shields: [https://es.aliexpress.com/item/1005004508626229.html?spm=a2q0o.order\\_list.order\\_list\\_main.17.7d2f194dKvTHCn&gatewayAdapt=glo2esp](https://es.aliexpress.com/item/1005004508626229.html?spm=a2q0o.order_list.order_list_main.17.7d2f194dKvTHCn&gatewayAdapt=glo2esp)
  - Placa de metacrilat: <https://www.amazon.es/MW-Materials-World-Metacrilato-transparente/dp/B00PM2LZ4G>
  - estany 0,6 mm 50g: [https://es.aliexpress.com/item/32986265878.html?spm=a2q0o.productlist.0.0.52576a3frGWJOK&algo\\_pvid=29a5cc7a-9c64-4c34-83cb-9c08c9b1dcb6&aem\\_p4p\\_detail=202212071127282062584763459350002131643&algo\\_exp\\_id=29a5cc7a-9c64-4c34-83cb-9c08c9b1dcb6-4&pdp\\_ext\\_f=%7B%22sku\\_id%22%3A%2210000013513509048%22%7D&pdp\\_npi=2%40dis%21EUR%216.03%210.98%21%21%21%21%21%402100bde316704412483492475e7c76%2110000013513509048%21sea&curPageLogUid=qyBFxJnFtacy&ad\\_pvid=202212071127282062584763459350002131643\\_5&ad\\_pvid=202212071127282062584763459350002131643\\_5](https://es.aliexpress.com/item/32986265878.html?spm=a2q0o.productlist.0.0.52576a3frGWJOK&algo_pvid=29a5cc7a-9c64-4c34-83cb-9c08c9b1dcb6&aem_p4p_detail=202212071127282062584763459350002131643&algo_exp_id=29a5cc7a-9c64-4c34-83cb-9c08c9b1dcb6-4&pdp_ext_f=%7B%22sku_id%22%3A%2210000013513509048%22%7D&pdp_npi=2%40dis%21EUR%216.03%210.98%21%21%21%21%21%402100bde316704412483492475e7c76%2110000013513509048%21sea&curPageLogUid=qyBFxJnFtacy&ad_pvid=202212071127282062584763459350002131643_5&ad_pvid=202212071127282062584763459350002131643_5)

- Annex:

- crear servidor: <https://www.ionos.es/digitalguide/servidores/know-how/server-de-discord/>
- crear bot: <https://www.ionos.es/digitalguide/servidores/know-how/discord-bot/>
- instalar node-red: <https://randomnerdtutorials.com/access-node-red-dashboard-anywhere-digital-ocean/#more-97743>
- instalar mqtt: <https://randomnerdtutorials.com/how-to-install-mosquitto-broker-on-raspberry-pi/>
- configurar arduino: <https://www.profetolocka.com.ar/2020/05/02/programando-el-esp-8266-con-el-arduino-ide/>
- introducció openSCAD: <https://sites.google.com/a/iepegasoviana.cat/impressio-3d/disseny-amb-el-programari-openscad>
- taula de freqüències: <https://www.ciudadpentagrama.com/2020/01/tabla-frecuencias-notas-musicales.html>

### 13.2.Programari utilitzat

<b>Diagrames i esquemes</b>	<a href="#">Dia Diagram Editor</a>
<b>Programar la placa</b>	<a href="#">Arduino IDE</a>
<b>Plataforma de comunicació</b>	<a href="#">Discord</a>
<b>Enllaç placa-plataforma de comunicació</b>	<a href="#">Node-RED</a>
<b>Documentar memòria</b>	<a href="#">Documents de Google</a>
<b>Passar arxius midi per arduino</b>	<a href="#">arduinomidi.netlify</a>
<b>Disseny</b>	<a href="#">Adobe Photoshop</a>