



# SmartTemp

**Alejandro Escobar Fuertes**  
Grau en Enginyeria Informàtica  
05.663 TFG - Arduino

**Pere Tuset Peiró**  
**Antoni Morell Pérez**

10 de Juny de 2018



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

© (Alejandro Escobar Fuertes)

Reservats tots els drets. Està prohibit la reproducció total o parcial d'aquesta obra per qualsevol mitjà o procediment, compresos la impressió, la reprografia, el microfilm, el tractament informàtic o qualsevol altre sistema, així com la distribució d'exemplars mitjançant lloguer i préstec, sense l'autorització escrita de l'autor o dels límits que autoritzi la Llei de Propietat Intel·lectual.

*“A la meva dona Silvia i a la meva filla Aina, gràcies a elles soc qui soc i només puc expressar el meu sincer agraïment per animar-me a continuar durant aquesta etapa acadèmica que avui finalitza. Sense elles no hauria sigut possible”*

## FITXA DEL TREBALL FINAL

<b>Títol del treball:</b>	<i>SmartTemp</i>
<b>Nom de l'autor:</b>	<i>Alejandro Escobar Fuertes</i>
<b>Nom del consultor/a:</b>	<i>Pere Tuset Peiró</i>
<b>Nom del PRA:</b>	<i>Antoni Morell Pérez</i>
<b>Data de lliurament (mm/aaaa):</b>	<i>06/2018</i>
<b>Titulació o programa:</b>	<i>Grau en Enginyeria Informàtica</i>
<b>Àrea del Treball Final:</b>	<i>TFG - ARDUINO</i>
<b>Idioma del treball:</b>	<i>Català</i>
<b>Paraules clau</b>	<i>Arduino, MQTT, ESP32</i>
<p><b>Resum del Treball (màxim 250 paraules):</b> <i>Es tracta d'una aplicació desenvolupada en Arduino, fent ús de la placa ESP32 amb una arquitectura client-servidor MQTT i un portal Web com a client MQTT com a interfície gràfica per l'usuari.</i></p> <p><i>Per un lloc tenim el hardware escollit com la placa ESP32, de l'empresa Espressif. S'ha escollit aquesta placa, perquè té Wifi inclosa a més de ser de baix consum. A més una pantalla de tinta electrònica de Waveshare, 1,5 polsades i un sensor de temperatura digital ds18b20. Posteriorment, s'ha inclòs un rellotge RTC per tenir la data i hora actual correcte, i s'ha reutilitzat una bateria vella de mòbil per dotar al sistema de la possibilitat de ser portable.</i></p> <p><i>Per una altra banda, s'ha creat un servidor virtual fent ús de la plataforma oferta per Amazon AWS, on s'ha instal·lat un sistema operatiu Linux Ubuntu i posteriorment s'ha instal·lat un servidor i client MQTT. Aquest servidor rep les peticions de connexió dels clients MQTT, per establir aquesta connexió s'ha de fer per un port TCP especificat prèviament i indicant l'usuari i la contrasenya creats i configurats prèviament al sistema.</i></p> <p><i>Finalment, s'ha creat una pàgina web, que s'adapta gràficament a les diverses plataformes que es poden utilitzar, com pot ser un ordinador o un mòbil. Aquesta pàgina es un client MQTT també, com el node ESP32.</i></p>	

**Abstract (in English, 250 words or less):** *Is an application based on Arduino, using the ESP32 microcontroller with a client-server architecture using the MQTT protocol and Web portal as MQTT client like graphic interface for end users.*

*In one hand, we have the selected hardware like the ESP32 board, from Espressif company. This module has been chosen because it has Wifi built-in and is low power module. Besides, the selected display was a 1,5 inches ink-display of Waveshare and a digital temperature sensor ds18b20. On top of that, it was added a RTC clock to get the current date and time, and we have reused an old mobile battery becoming the system in portable.*

*In the other hand, it has been configured a virtual server on Amazon AWS service available, where it has been installed Linux Ubuntu operating system and afterwards, it has been configured a MQTT server – client. This server receives the MQTT client connection requests, to establish this connection it has to be done using a specific TCP port configured previously, and using a username and password configured before in the system.*

*Finally, it has been developed a responsive webpage, it adapts graphically at different devices used, like computers or mobile phones. This webpage is another MQTT client, like the ESP32 is.*

## Tabla de contenido

<b>1. Introducció.....</b>	<b>2</b>
1.1. Descripció del projecte .....	2
1.2. Objectius del projecte.....	4
1.3. Planificació .....	6
1.3.1. Llista tasques .....	6
1.3.2. Diagrama de Gantt PAC 1.....	8
1.3.3. Diagrama de Gantt PAC 2.....	8
1.3.4. Diagrama de Gantt PAC 3.....	8
1.4. Viabilitat .....	8
1.4.1. Detall del cost del hardware pel prototip Arduino.....	9
1.4.2. Cost manteniment Plataforma Cloud AWS.....	9
1.4.3. Cost manteniment pàgina web.....	10
1.4.4. Estimació cost del desenvolupament del projecte.....	10
1.4.5. Estimació del cost de manteniment mensual .....	11
<b>2. Detall del hardware .....</b>	<b>12</b>
2.1. Microcontrolador.....	13
2.2. E-ink display.....	14
2.3. Descripció del funcionament del sistema desenvolupat.....	15
2.3.1. Reinici del sistema .....	15
2.3.2. Connexió amb servidor MQTT .....	17
2.3.3. Descripció Menús.....	18
2.3.4. Bucle principal .....	22
<b>3. Plataforma Cloud.....</b>	<b>23</b>
3.1. Detall protocol MQTT .....	23
3.1.1. Què es MQTT?.....	23
3.1.2. Comunicació client - servidor mqtt.....	23
3.2. Descripció de la solució escollida .....	24
3.3. Temps d'implementació .....	26
3.4. Configuració del broker MQTT.....	27
3.4.1. Com funciona el servidor MQTT .....	29
<b>4. Pàgina web.....</b>	<b>30</b>
4.1. Descripció general .....	30
4.2. Visió general web .....	31
4.2.1. Mapa del lloc web .....	31
4.2.2. Pagina Inici (login) .....	32
4.2.3. Pàgina principal.....	33
4.2.4. Ajustos bàsics .....	34
4.2.5. Ajustos avançats.....	36

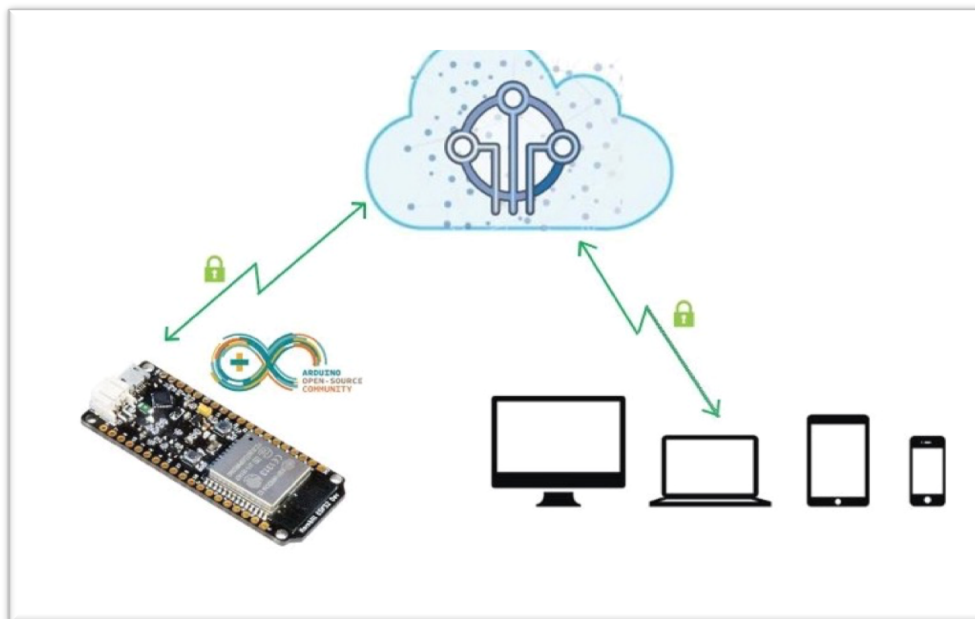
4.2.6.	Programació horària .....	37
4.2.7.	Tancar sessió .....	38
<b>5.</b>	<b><i>Detall del software</i></b> .....	<b>39</b>
5.1.	Software programació ESP32.....	39
5.2.	Software disseny pantalles .....	39
5.3.	Software creació fitxers .C per la pantalla .....	39
5.4.	Disseny pàgina web .....	41
5.5.	Esquemes elèctrics.....	41
5.6.	Diagrames de flux .....	42
<b>6.</b>	<b><i>Conclusions</i></b> .....	<b>43</b>
6.1.	Implementació.....	43
6.2.	Possibles millores .....	43
6.3.	Valoració personal.....	44
<b>7.</b>	<b><i>Bibliografia</i></b> .....	<b>46</b>
<b>8.</b>	<b><i>Annexos</i></b> .....	<b>48</b>
8.1.	Preparació IDE d'Arduino.....	48
8.2.	Llibreries utilitzades.....	50
8.3.	Esquemes elèctrics.....	51

# 1. Introducció

## 1.1. Descripció del projecte

El projecte SmartTemp, consisteix en un termòstat digital com el que tenim actualment a totes les cases, però amb connexió Wifi, amb una interfície gràfica on l'usuari pot programar,-lo consultar històrics, temps de funcionament, etc. En aquest projecte, l'arquitectura constarà d'un node amb connexió Wifi integrada de baix consum energètic, que controlarà amb un contacte lliure de potencial el control de la caldera de la calefacció. A més a més d'un display tindrà una connexió amb un servei de cloud IoT broker MQTT<sup>1</sup>, aquest servidor MQTT, s'implementarà en una plataforma Linux Ubuntu mitjançant els servidors virtuals que ofereix Amazon AWS.

Amb la tecnologia que es desenvoluparà el projecte es pot fer escalable tant en nombre de nodes, com en serveis a mostrar per cada node. A més que la plataforma Cloud IoT seria la base per anar implementant serveis domòtics per cada casa.



L'arduino escollit es de la empresa Espressif i el model es el ESP32. Es un dispositiu de baix consum, amb connexió Wifi integrada i Bluetooth. El sensor de temperatura DS18B20, ja que el DTH11 tenia un error de 2°C i per tant no era recomanable, a més a més disposarà d'un RTC per tenir control del temps i del dia de la setmana. Per finalitzar disposarà d'una pantalla e-ink de 1,54 inches, també de baix consum. El control de la caldera es farà mitjançant un relé OMRON on mitjançant un contacte lliure de potencial controlarà la caldera actual, per tant l'usuari no necessita cap calder especial, pot adaptar els sistema actual sense problemes.

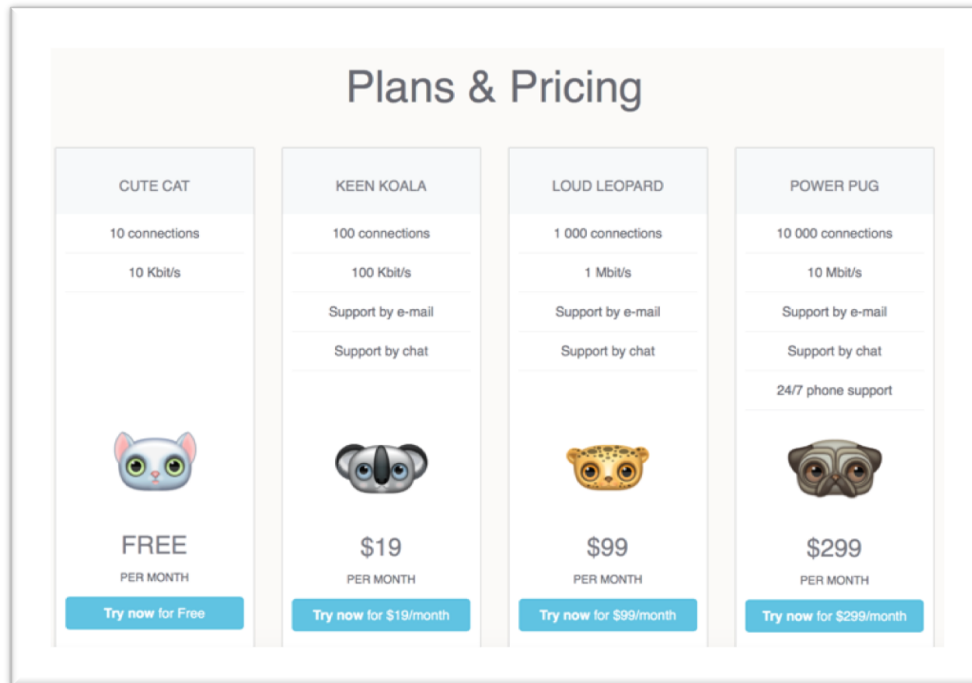
Inicialment havia escollit la plataforma CloudMQTT, ja que disposava d'un pla gratuït, i que ens donava la possibilitat d'escalabilitat. Però, durant el desenvolupament de la PAC3 les





<sup>1</sup> MQTT: Messaging Queuing Telemetry Transport.



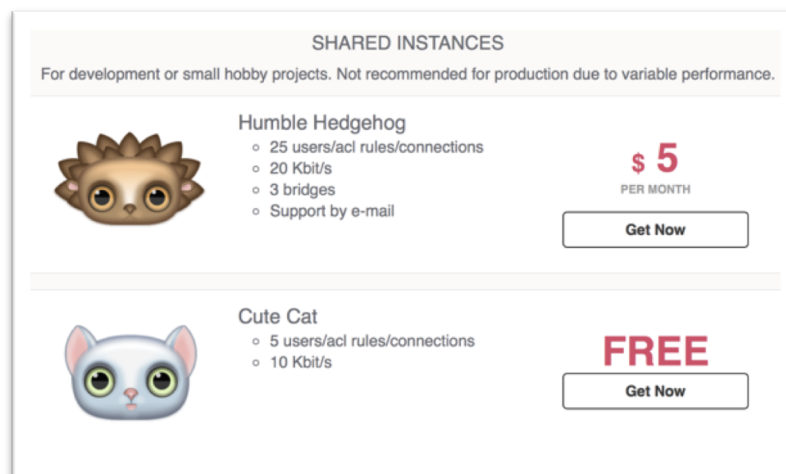
condicions del pla gratuït varen canviar i va passar de 10 connexions a només 5, això afectava a la funcionalitat del producte ideat inicialment.

Condicions inicials:





CUTE CAT	KEEN KOALA	LOUD LEOPARD	POWER PUG
10 connections	100 connections	1 000 connections	10 000 connections
10 Kbit/s	100 Kbit/s	1 Mbit/s	10 Mbit/s
	Support by e-mail	Support by e-mail	Support by e-mail
	Support by chat	Support by chat	Support by chat
			24/7 phone support
			
<b>FREE</b>	<b>\$19</b>	<b>\$99</b>	<b>\$299</b>
PER MONTH	PER MONTH	PER MONTH	PER MONTH
<a href="#">Try now for Free</a>	<a href="#">Try now for \$19/month</a>	<a href="#">Try now for \$99/month</a>	<a href="#">Try now for \$299/month</a>

Condicions actuals:



**SHARED INSTANCES**  
For development or small hobby projects. Not recommended for production due to variable performance.

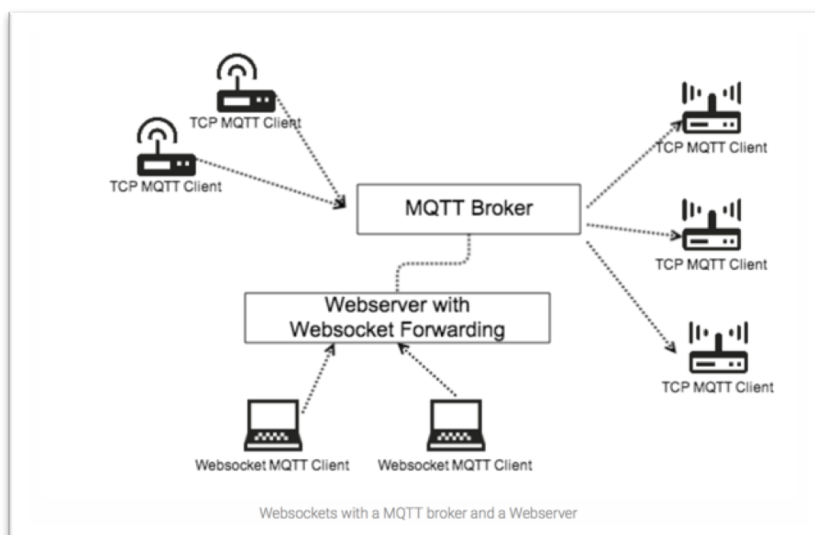
	<b>Humble Hedgehog</b> <ul style="list-style-type: none"> <li>25 users/acl rules/connections</li> <li>20 Kbit/s</li> <li>3 bridges</li> <li>Support by e-mail</li> </ul>	<b>\$ 5</b> PER MONTH <a href="#">Get Now</a>
	<b>Cute Cat</b> <ul style="list-style-type: none"> <li>5 users/acl rules/connections</li> <li>10 Kbit/s</li> </ul>	<b>FREE</b> <a href="#">Get Now</a>

Veient aquest canvi de condicions, es va optar per explorar la via de crear el nostre propi cloud MQTT server, utilitzant els serveis cloud que hi ha disponibles, com son Amazon AWS, Google Cloud i Microsoft Azure. Finalment l'escollit va ser Amazon AWS, ja que es una plataforma que faig un ús regular.

Aquesta plataforma també ens donaria la possibilitat de fer el hosting de la pàgina web, encara que actualment el domini aescobaf-smarttemp.online està gestionat per l'empresa

hostinger.com, ja que la pàgina web estava en desenvolupament quan es va trobar la limitació de la plataforma CloudMQTT.

Per tant l'estructura definitiva del sistema seria, un node que seria un client MQTT publicant i subscrit a una sèrie de tòpics configurats, que es detallaran més endavant. El servidor o broker MQTT i una pàgina web, també client MQTT, on l'usuari pot accedir-ne a veure dades del sistema i dona la possibilitat de configurar el sistema. A continuació es pot veure un esquema gràfic del sistema.



El termòstat, tindrà quatre estats de funcionament, Apagat – Manual dia – Manual Nit – Automàtic.

En els modes manuals, l'usuari seleccionarà la temperatura desitjada i el dispositiu anirà regulant la temperatura controlant el relé incorporat. La temperatura del mode manual dia es diferent al de manual nit.

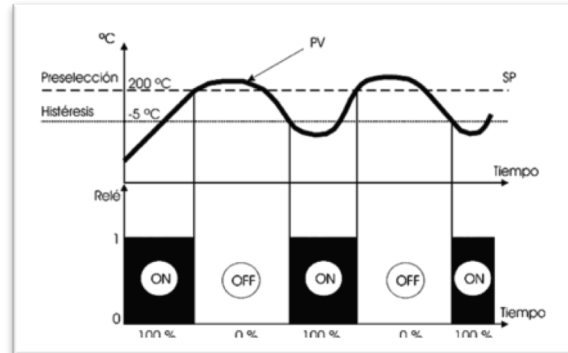
En el mode Automàtic, pot funcionar amb connexió Wifi o sense, en cas de problemes de xarxa. En el cas de que no hi hagi connexió a la xarxa, funcionarà amb els últims valors configurats. La forma de configurar la temperatura segons el dia i l'hora serà des del servei cloud, mitjançant la pàgina web desenvolupada. Remotament, es pot comprovar l'estat actual del sistema i poder activar o desactivar-lo remotament, així com ajustar l'offset del sensor, el valor d'histèresis i canviar el mode de treball.

## 1.2. Objectius del projecte

La principal finalitat del projecte desenvolupat, es el control de la temperatura d'una casa, el sistema seria la base inicial per a un sistema de domòtica integral, escalable amb connexió des d'Internet per saber l'estat de cada node automatitzat i implementat.

El producte desenvolupat per aquest projecte, es tracta d'un node que pot treballar standalone o connectat a la xarxa, on l'usuari marca una temperatura desitjada i el sistema activa o desactiva la caldera en funció a la temperatura actual. No es tracta d'un sistema de regulació PID, es un sistema de control binari, es a dir , connectat o desconnectat, però per





la regulació que es necessita a una casa es suficient. S'ha afegit un valor d'histéresis<sup>2</sup> per evitar que el sistema estigui activant i desactivant la caldera constantment. A la següent imatge es pot veure el concepte de forma gràfica.



L'usuari, pot activar o desactivar el sistema remotament, mitjançant la pàgina web creada, a més de canviar els paràmetres del sistema. Fins i tot configurar la temperatura desitjada hora per hora de dilluns a diumenge.

Els serveis que es podria oferir seria des de la venda dels dispositius, el servei de cloud amb manteniment del servei de supervisió remota, etc.

En el mercat comencen a sortir productes similars, oferts per les empreses de subministrament energètic, com per exemple :

	<b>Momit Home Thermostat BMHTP - Kit de inicio (Termostato +...</b>	€ 96,98 <small>129,95</small>	amazon	MÁS INFO
	<b>tado° Termostato Inteligente (producto adicional) - control inteligente de la...</b>	€ 128,34	amazon	MÁS INFO
	<b>Momit Smart MOMITSTB - Termostato inteligente para controlar la climatización...</b>	€ 138,19 <small>200,00</small>	amazon	MÁS INFO
	<b>Honeywell Lyric T6 - Termostato programable inteligente Wifi cableado</b>	€ 160,62 <small>199,00</small>	amazon	MÁS INFO

L'avantatge del nostre producte, es que es escalable i es una plataforma integral de domòtica per la casa, no només un termòstat intel·ligent. A més de poder oferir una quota mensual o anual per el manteniment del tele servei o accés remot, de servei de realitzar històrics, gràfics, etc.

<sup>2</sup> Histéresis: es la temperatura on el sistema s'activarà o es desactivarà. A més valor d'histéresis menys accionaments es donaràn.

### 1.3. Planificació

La planificació del treball es divideix en 3 fases diferenciades, que corresponen amb les diferents PACs que s'han d'entregar.

La primera fase, que correspon amb la PAC2, es centra en el disseny del hardware d'Arduino, realització del programa i tests, sempre en mode aïllat, sense connexió amb la pàgina web i el servidor MQTT.

La segona fase, que correspon amb la PAC3, es tracta de la interconnexió del dispositiu Arduino, amb el servidor MQTT i la pàgina web. En la planificació original, com s'anava a fer ús de la plataforma cloudMQTT, i després les condicions gratuïtes varen canviar, això va afectar a la planificació original i es va haver d'adaptar la planificació per poder entregar el projecte a la data d'entrega de la PAC3. Per tant, es va tenir que dedicar 30 hores extres, per poder implementar la solució del servidor virtual i del servidor MQTT. Finalment, es va fer un test global del sistema,

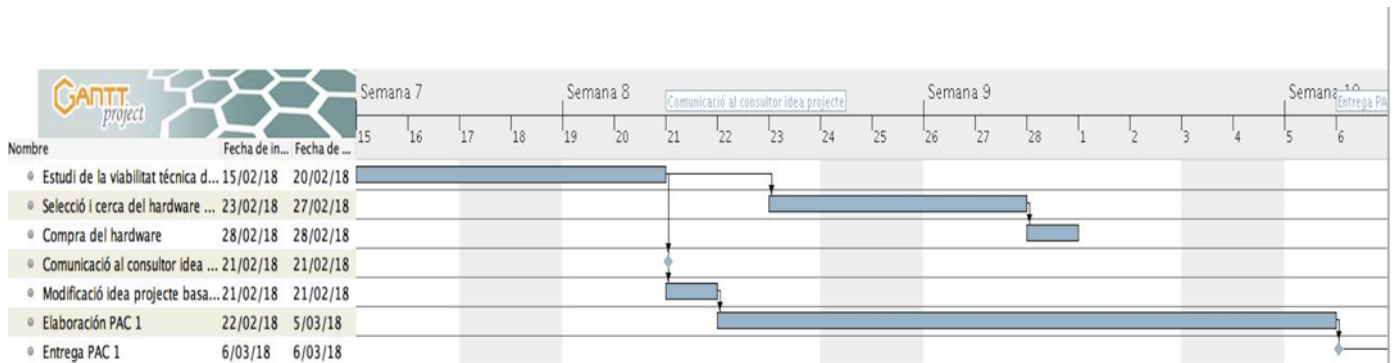
L'última fase, es tracta de la creació de la documentació i presentació final del projecte.

#### 1.3.1. Llista tasques

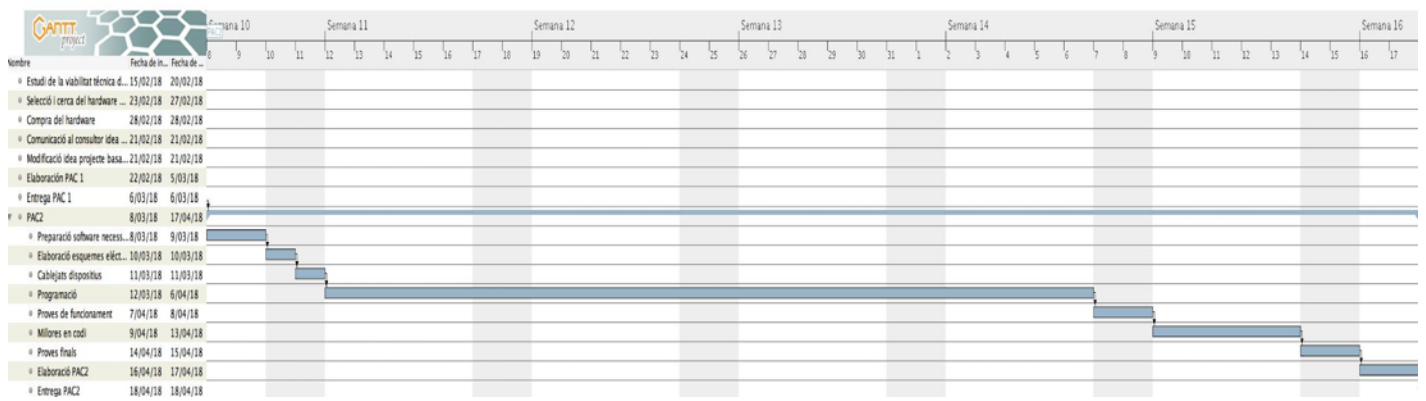
ID	Nombre	Data d'Inici	Data de fi	Duració	Progrés	Costo	Coordinador	Antecessors	Número de esquema	Recursos	Pàgina web	Notes
1	Estudi de la viabilitat tècnica del projecte	15/02/18	20/02/18	6	0	0			1			
2	Selecció i cerca del hardware necessari	23/02/18	27/02/18	5	0	0		1-FS=P2D	2			
37	Compra del hardware	28/02/18	28/02/18	1	0	0		2	3			
5	Comunicació al consultor idea projecte	21/02/18	21/02/18	0	0	0		1	4			
10	Modificació idea projecte basat en el feedback rebut	21/02/18	21/02/18	1	0	0		5	5			
14	Elaboració PAC 1	22/02/18	5/03/18	12	0	0		10	6			
17	Entrega PAC 1	6/03/18	6/03/18	0	0	0		14	7			
21	PAC2	8/03/18	17/04/18	41	0	0		17-FS=P2D	8			
44	Preparació software necessari	8/03/18	9/03/18	2	0	0			8.1			
54	Elaboració esquemes elèctrics	10/03/18	10/03/18	1	0	0		44	8.2			
55	Cablejats dispositius	11/03/18	11/03/18	1	0	0		54	8.3			
60	Programació	12/03/18	6/04/18	26	0	0		55	8.4			

65	Proves de funcionament	7/04/18	8/04/18	2	0	0		60	8.5			
79	Millores en codi	9/04/18	13/04/18	5	0	0		65	8.6			
85	Proves finals	14/04/18	15/04/18	2	0	0		79	8.7			
98	Elaboració PAC2	16/04/18	17/04/18	2	0	0		85	8.8			
103	Entrega PAC2	18/04/18	18/04/18	0	0	0		98	8.9			
46	PAC3	19/04/18	23/05/18	35	0	0		103	9			
52	Selecció plataforma IoT	19/04/18	21/04/18	3	0	0			9.1			
110	Configuració plataforma	22/04/18	26/04/18	5	0	0		52	9.2			
116	Configuració xarxa	27/04/18	28/04/18	2	0	0		110	9.3			
121	Programació interfície usuari	29/04/18	13/05/18	15	0	0		116	9.4			
125	Proves de funcionament	14/05/18	14/05/18	1	0	0		121	9.5			
132	Millores en la interfície	15/05/18	19/05/18	5	0	0		125	9.6			
148	Prova del sistema definitiu	20/05/18	20/05/18	1	0	0		132	9.7			
152	Elaboració PAC3	21/05/18	23/05/18	3	0	0		148	9.8			
156	Entrega PAC3	24/05/18	24/05/18	0	0	0		152	9.9			
160	Memòria del treball	24/05/18	10/06/18	18	0	0		156	10			
191	Elaboració memòria	24/05/18	10/06/18	18	0	0			10.1			
197	Entrega memòria	11/06/18	11/06/18	0	0	0		191	10.2			
201	Presentació treball	11/06/18	17/06/18	7	0	0		197	11			
205	Elaboració presentació	11/06/18	17/06/18	7	0	0			11.1			
209	Entrega presentació	18/06/18	18/06/18	0	0	0		205	11.2			
213	Tribunal	19/06/18	19/06/18	1	0	0		209-FS=P1D	12			
220	Fi del tribunal	20/06/18	20/06/18	0	0	0		213	13			

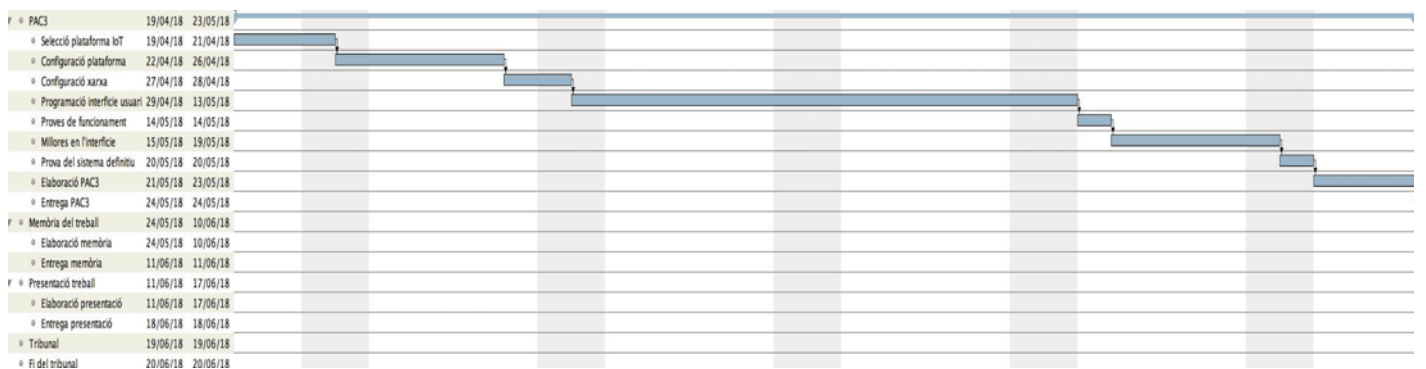
### 1.3.2. Diagrama de Gantt PAC 1



### 1.3.3. Diagrama de Gantt PAC 2



### 1.3.4. Diagrama de Gantt PAC 3



## 1.4. Viabilitat

Encara ens trobem en un esta inicial del projecte, però l'estimació del cost en hardware seria:

### 1.4.1. Detall del cost del hardware pel prototip Arduino

Descripció	Quantitat	Preu unitari	Preu
ESP32 (microcontrolador)	1	5,99€	5,99€
Sensor temperatura DS18B20	1	1€	1€
E-ink Display	1	16,55€	16,55€
Relé OMRON SSR G3MB-202P	1	1,40€	1,40€
Varis (polsadors, cables, LED's, etc)	1	5€	5€
Porta piles	1	1,5€	1,5€
Pila 9V	1	3€	3€
Carcassa	1	10€	10€
<b>Total</b>			<b>44,44€</b>

### 1.4.2. Cost manteniment Plataforma Cloud AWS

El cost per hora del servei de la plataforma cloud MQTT seria de \$0.0132 x hora, per la funcionalitat que es necessita seria suficient, ja que només necessitem intercanvi de dades, res de potencia de computació, gràfic o similar. Per tant el cost mensual seria de 0,0132 x 24 x 31 dies , \$ 9,82 /mes. A això s'ha de sumar el cost de l'assignació de una adreça IP fixa que es de \$0.010 per GB. Tenint en comte que el volum de dades es de l'ordre de KB, es un cost sense cap impacte.

Linux	RHEL	SLES	Windows	Windows con SQL Standard	Windows con SQL Web
Windows con SQL Enterprise	Linux con SQL Standard	Linux con SQL Web	Linux con SQL Enterprise		

Región:	UE (Londres) ▾
---------	----------------

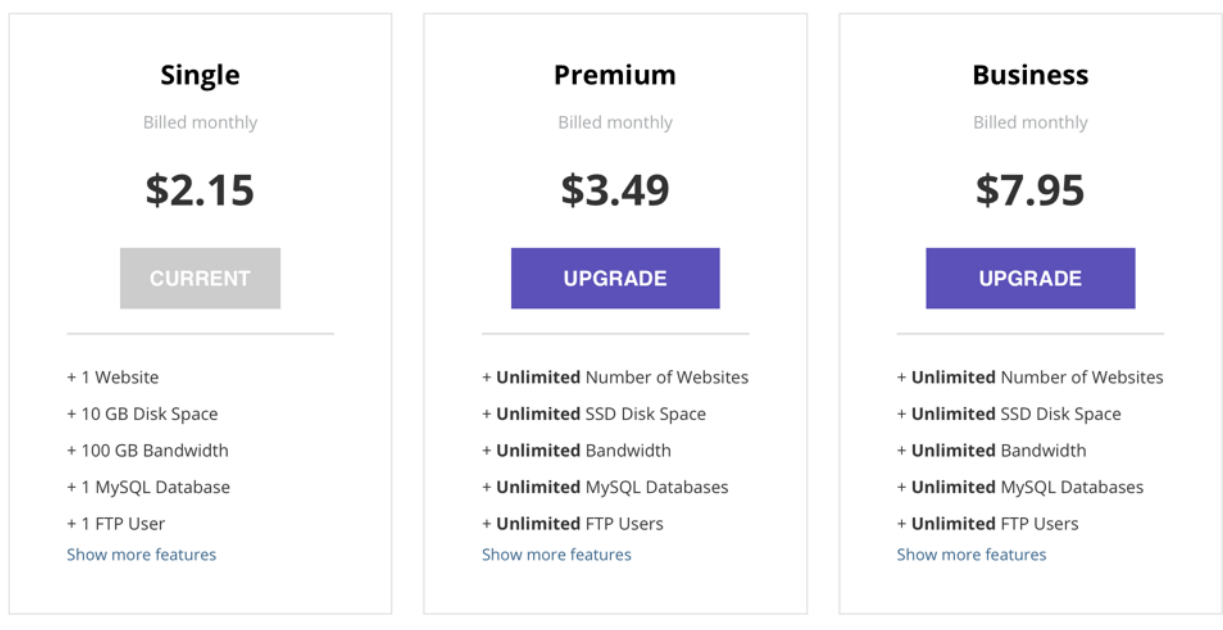
  

	vCPU	ECU	Memoria (GiB)	Almacenamiento de la instancia (GB)	Uso de Linux/UNIX
<b>Uso general – Generación actual</b>					
t2.nano	1	Variable	0.5	Solo EBS	\$0.0066 por hora
t2.micro	1	Variable	1	Solo EBS	\$0.0132 por hora
t2.small	1	Variable	2	Solo EBS	\$0.026 por hora
t2.medium	2	Variable	4	Solo EBS	\$0.052 por hora

### 1.4.3. Cost manteniment pàgina web

El cost de manteniment del domini es de \$19.99 per any i el servei de hosting amb l'empresa [www.hostinger.com](http://www.hostinger.com) es de \$3.45 per mes. En total tindriem un cost mensual de \$5.11 per mes.

◀ UPGRADE - AESCOBARF-SMARTTEMP.ONLINE ▶ WEB HOSTING



Plan	Price (Monthly)	Status	Features
Single	\$2.15	CURRENT	+ 1 Website + 10 GB Disk Space + 100 GB Bandwidth + 1 MySQL Database + 1 FTP User
Premium	\$3.49	UPGRADE	+ Unlimited Number of Websites + Unlimited SSD Disk Space + Unlimited Bandwidth + Unlimited MySQL Databases + Unlimited FTP Users
Business	\$7.95	UPGRADE	+ Unlimited Number of Websites + Unlimited SSD Disk Space + Unlimited Bandwidth + Unlimited MySQL Databases + Unlimited FTP Users

### 1.4.4. Estimació cost del desenvolupament del projecte

Descripció	Quantitat	Preu unitari	Preu
Planificació, selecció materials, etc	10h	24€/h	240€
Assemblatge	2h	24€/h	48€
Programació ESP32	15h	24€/h	360€
Programació plataforma IoT	30h	24€/h	770€
Elaboració Pàgina Web	10h	240€/h	240€
Testeig i millores	10h	24€/h	240€
Elaboració documentació	10h	24€/h	240€
Instal·lació i configuració	2h	24€/h	48€
<b>Total</b>			<b>2088€</b>



#### 1.4.5. Estimació del cost de manteniment mensual

Descripció	Cost mensual
Plataforma cloud MQTT (AWS)	9.82€
IP publica	0.01€ x GB
Hosting Web + domini	5.11€
<b>Total</b>	<b>14,94€</b>

A part d'aquests costos de la plataforma cloud MQTT i Web, s'han de tenir en compte el servei postvenda, lloguer oficina, línia telefònica, etc.

Descripció	Quantitat	Preu unitari	Cost mensual
Personal tècnic atenció al client	2 persones	2000€ bruts /any	4000€
Lloguer oficina	1	500€	500€
Línia telefònica + Fibra	1	100€	100€
Material informàtic	2	100€	200€
Energia	1	100€	100€
Altres	1	100€	100€
<b>Total</b>			<b>5000€</b>

L'empresa Viesgo, segons varies notícies a diferents medis, ha instal·lat 350,000 unitats als Països Baixos i Bèlgica, per tant, si aquests països tenen una població de 28 milions de persones, correspon aproximadament al 1,25% de la població. Si el nostre mercat fos Espanya i es mantingués el mateix percentatge de vendes, tenint en compte que la població d'Espanya es de 46 milions i restéssim els 13 milions de persones que viuen en risc de pobresa, segons <http://www.rtve.es/noticias/20171016/casi-13-millones-personas-sufren-pobreza-exclusion-social-espana/1628613.shtml>, tenim mercat potencial dunes 400.000 unitats.

Si veiem que en el mercat els termòstats intel·ligents, tenen un preu que varien des dels 90€ fins els 160€, i el nostre cost de disseny es de 2088€ + 44,44€ x unitat de hardware, si venguéssim 40 unitats a un preu de 92,89€ cobriríem les despeses inicials.

Per tal de garantir la viabilitat del servei post venta, si considerem que un càrrec de 15€ es un preu competitiu, s'hauria de tenir una base instal·lada de 333 dispositius.

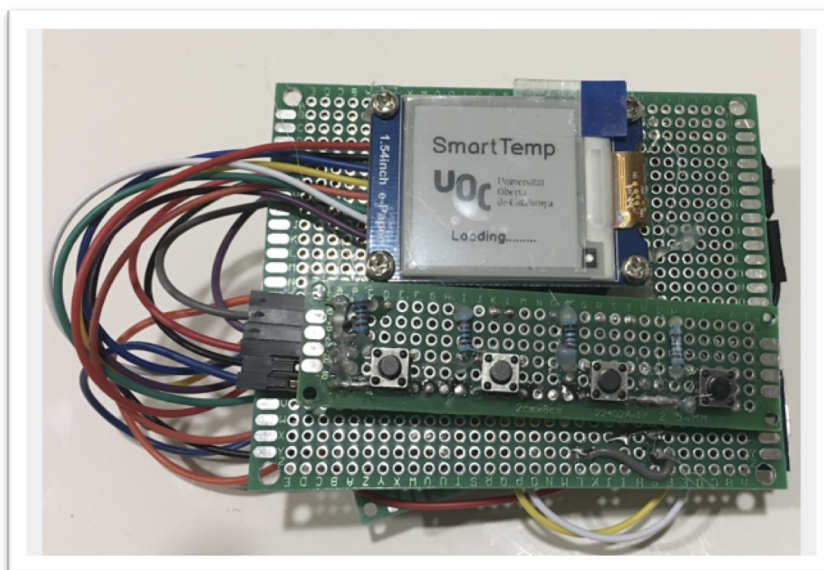
Amb unes vendes potencials a nivell nacional de 400,000 unitats, podríem dir que la potencial facturació del projecte seria de 37milions d'euros en concepte de venda de hardware, sense tenir en compte la possibilitat d'oferir més servei que els termòstats.

Els punts febles del negoci, seria l'alta competència per part d'altres marques, provinents principalment del sud-est asiàtic, màrqueting necessari, així com la necessitat d'aliances amb empreses distribuïdores d'energia per tenir accés a la seva massa d'abonats.

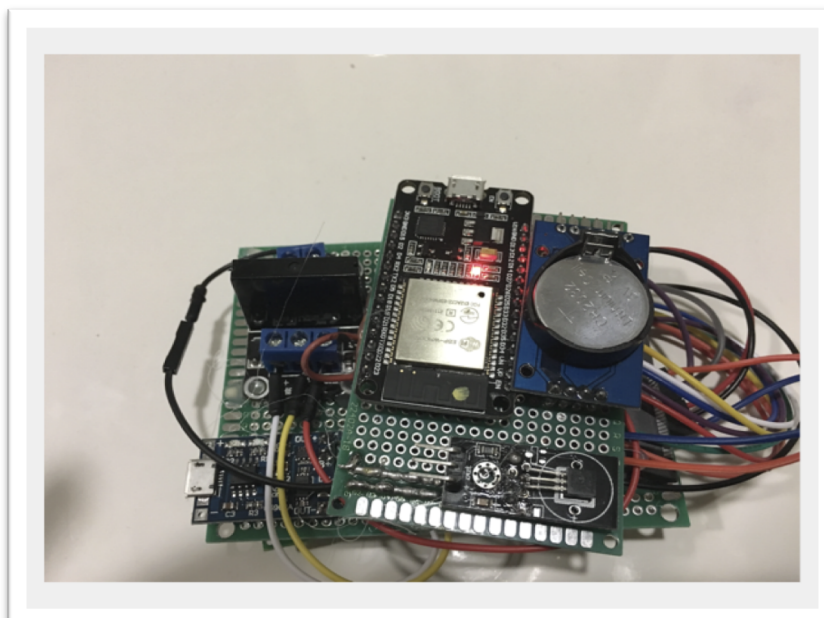
## 2. Detall del hardware

El sistema actual es el següent:

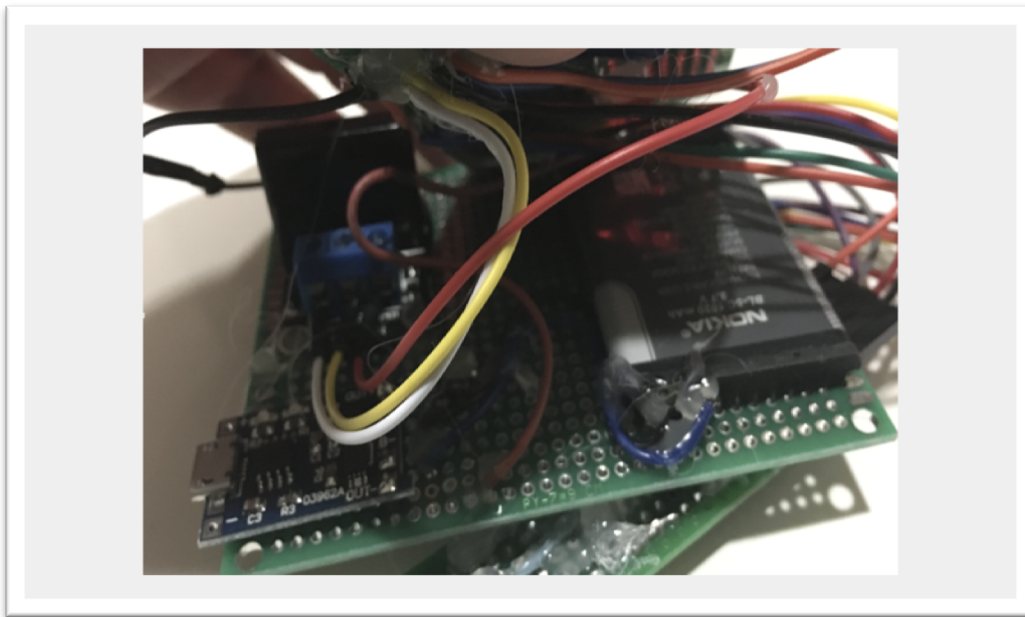
Part frontal, on veiem la pantalla e-ink i els pulsadors,



En la part del darrere, veiem el rellotge RTC, el sensor de temperatura i el ESP32.



En aquesta imatge veiem una bateria de mòbil adaptada amb el mòdul de carrega utilitzat. Per tant el sistema pot treballar sense estar endollat, pot ser portàtil.



## 2.1. Microcontrolador

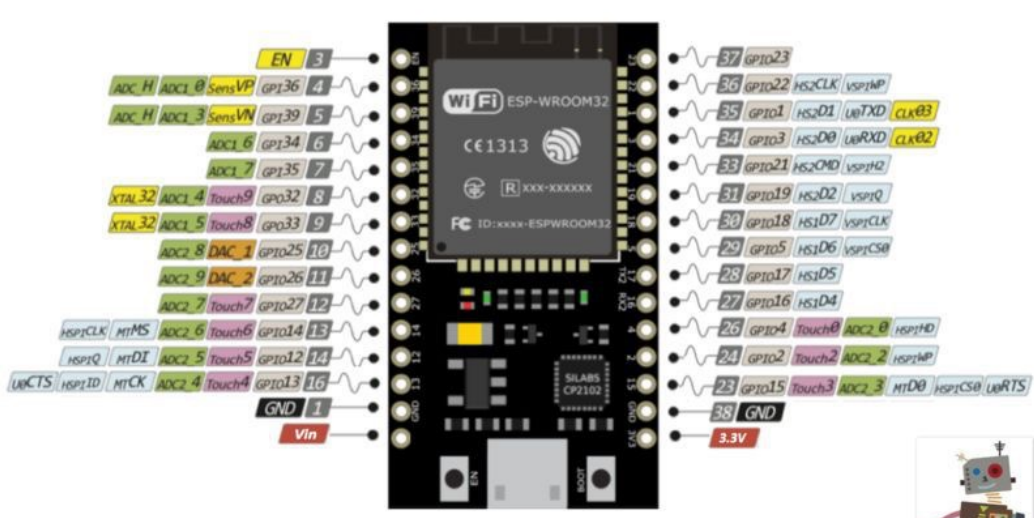
El microcontrolador que hem utilitzat ha sigut el ESP32 de l'empresa Espressif. Aquest hardware no es compatible directament amb la plataforma d'Arduino, però es pot adaptar seguint tutorials trobats per internet.

# ESP32

## PINOUT


ESP32 DEVELOPMENT BOARD DUAL CORE ESP-32 & ESP-32S BOARD

- Power
- GND
- Serial Pin
- Analog Pin
- Control
- Physical Pin
- Port Pin
- Touch Pin
- DAC Pin
- PWM Pin



The diagram shows the ESP32 development board with pins labeled on both sides. The left side labels include: EN (3), ADC\_H ADC1\_8 SensVP GPI36 (4), ADC\_H ADC1\_3 SensVN GPI39 (5), ADC1\_6 GPI34 (6), ADC1\_7 GPI35 (7), XTAL\_32 ADC1\_4 Touch9 GPO32 (8), XTAL\_32 ADC1\_5 Touch8 GPO33 (9), ADC2\_8 DAC\_1 GPI25 (10), ADC2\_9 DAC\_2 GPI26 (11), ADC2\_7 Touch7 GPI27 (12), HSP1CLK M1MS ADC2\_6 Touch6 GPI14 (13), HSP1Q M1DI ADC1\_5 Touch5 GPI12 (14), U1CTS HSP1ID M1CK ADC2\_4 Touch4 GPI13 (15), GND (1), and Vin. The right side labels include: 37 GPIO23, 36 GPIO22 HS2CLK VSP1M, 35 GPIO1 HS2D1 U1TXD CLK03, 34 GPIO3 HS2D0 U1RXD CLK02, 33 GPIO21 HS2CMD VSP1M2, 31 GPIO19 HS2D2 VSP1Q, 30 GPIO18 HS1D7 VSP1CLK, 29 GPIO5 HS1D6 VSP1CS0, 28 GPIO17 HS1D5, 27 GPIO16 HS1D4, 26 GPIO4 Touch8 ADC2\_8 HSP1M0, 24 GPIO2 Touch2 ADC2\_2 HSP1M, 23 GPIO15 Touch3 ADC2\_3 M1D0 HSP1CS0 U1RTS, and 38 GND. A 3.3V pin is also indicated.

Freely adapted by <https://MJRoBot.org>



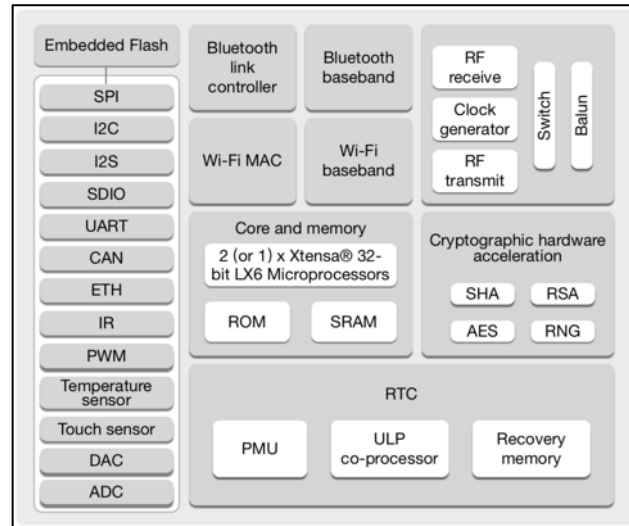
MJRoBot.org

Entre altres característiques tècniques importants destaquem que aquest microcontrolador té Wifi i Bluetooth integrats, memòria EEPROM i RTC inclòs, a més de ser de baix consum. En la pàgina del fabricant espressif

<https://www.espressif.com/en/products/hardware/esp32/overview> , es pot trobar tota la

informació tècnica i comercial d'aquest dispositiu. A continuació en els annexos es pot consultar les especificacions tècniques, més rellevants.

En la següent imatge podem veure el diagrama de blocs intern del ESP32.



## 2.2. E-ink display

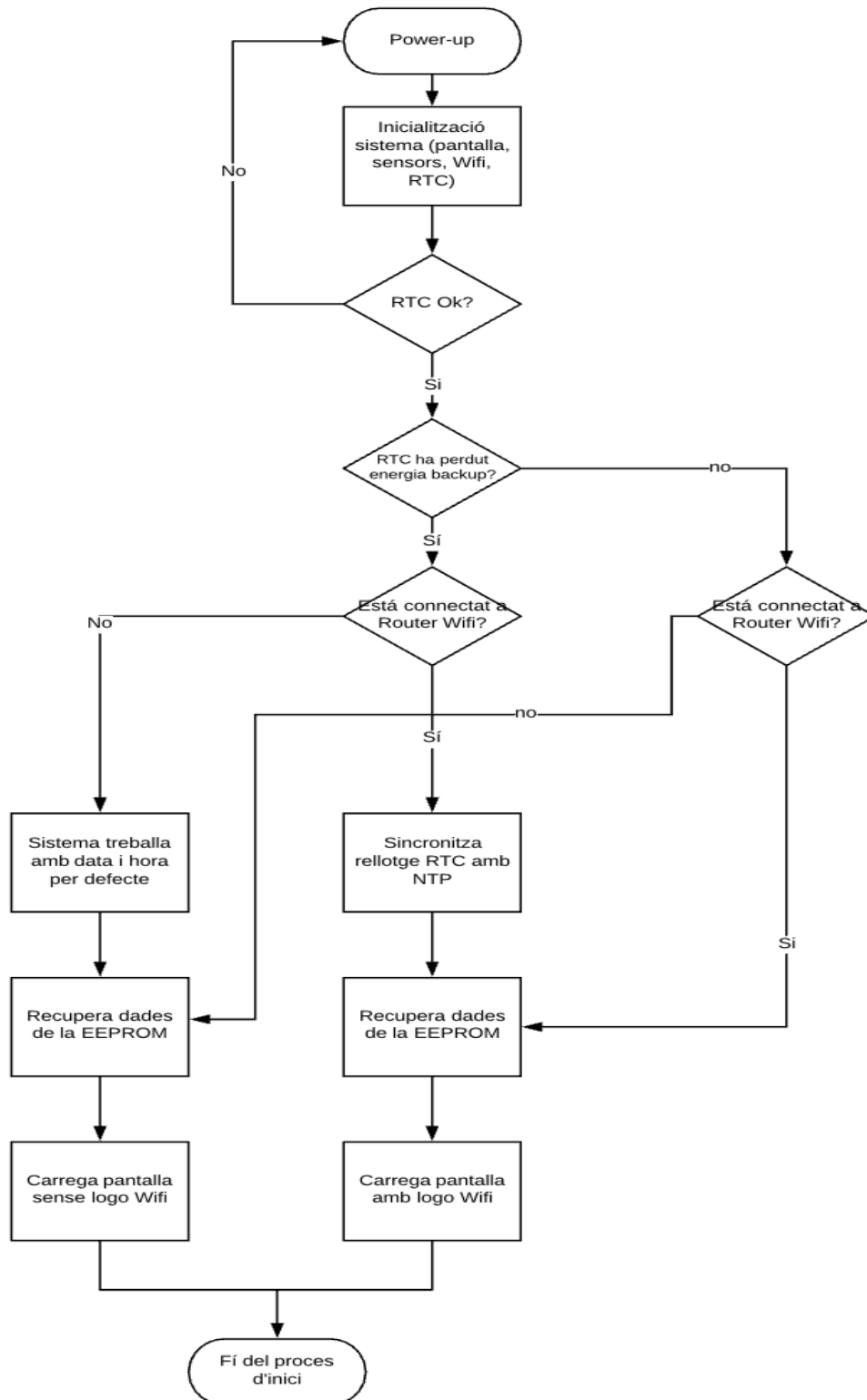
La pantalla escollida pel projecte es una pantalla de tecnologia de tinta electrònica (e-ink), ja que es de baix consum, i l'objectiu del projecte es desenvolupar un sistema de molt baix consum elèctric. El principi de funcionament d'aquesta tecnologia es crear esferes molt petites, on els pigments de color estan suspesos en un oli transparent i es mouen depenent la carga electrolítica. El sistema no necessita de il·luminació de fons, i encara que estigui en un punt on el sol li pugui donar directe la pantalla seguirà sent visible.

Tota la informació tècnica es pot trobar al següent link

[https://www.waveshare.com/wiki/1.54inch\\_e-Paper\\_Module#Working\\_principle](https://www.waveshare.com/wiki/1.54inch_e-Paper_Module#Working_principle) .

## 2.3. Descripció del funcionament del sistema desenvolupat

### 2.3.1. Reinici del sistema



El sistema durant la re-inicialització, primer, arranca el port sèrie per poder obrir la consola del IDE d'arduino i poder veure la sortida del programa pel port sèrie.

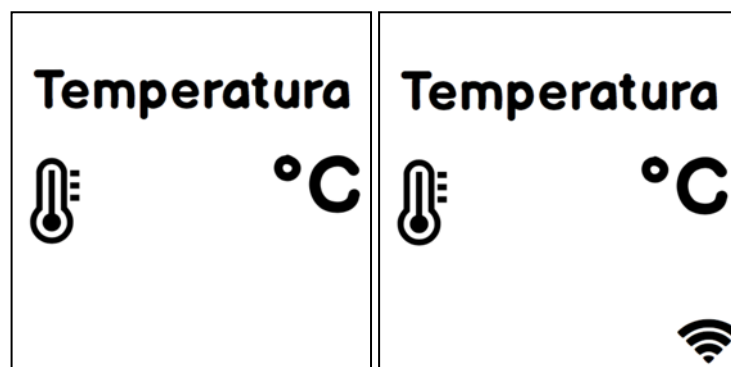
Tot seguit inicialitza la pantalla, els sensors, i mostra la pantalla d'inici.



Posteriorment, inicialitza la connexió Wifi, i aplica un delay de 3 segons per donar temps a obrir la consola.

Inicialitza els pulsadors i la sortida digital on es connecta el relé. Tot seguit inicialitza el rellotge RTC extern, si no troba el rellotge, treu el missatge de RTC<sup>3</sup> no trobat i reinicia el sistema. Per tant, el sistema entraria en un bucle de reinici si no troba el rellotge RTC, ja que es primordial pel funcionament del sistema. Si el sistema detecta el mòdul, llavors comprova si el RTC ha perdut l'energia de backup, la pila, si s'ha perdut l'energia de la bateria llavors crida a la funció creada per sincronitzar el RTC amb el servidor NTP<sup>4</sup>. Per tant per fer aquesta sincronització el sistema ha d'estar connectat al router. Si no està connectat a cap xarxa Wifi, llavors el sistema treballarà amb l'hora i data que té el RTC per defecte 01/01/2000 i l'hora 00:00.

Una vegada el sistema ja té una hora i data correcte, inicialitza la EEPROM<sup>5</sup> interna, i recupera les dades emmagatzemades i les guarda en la matriu de dades de treball. Si per lo contrari no troba la EEPROM, llavors el sistema es reinicialitzaria com ha fet amb el RTC. Per últim, una vegada ha recuperat les dades de la EEPROM, carrega la pantalla principal. Hi ha dues pantalles principals, una per quan el sistema està connectat a la Wifi i l'altre per quan no està connectat a cap xarxa Wifi.



<sup>3</sup> RTC: Real Time Clock

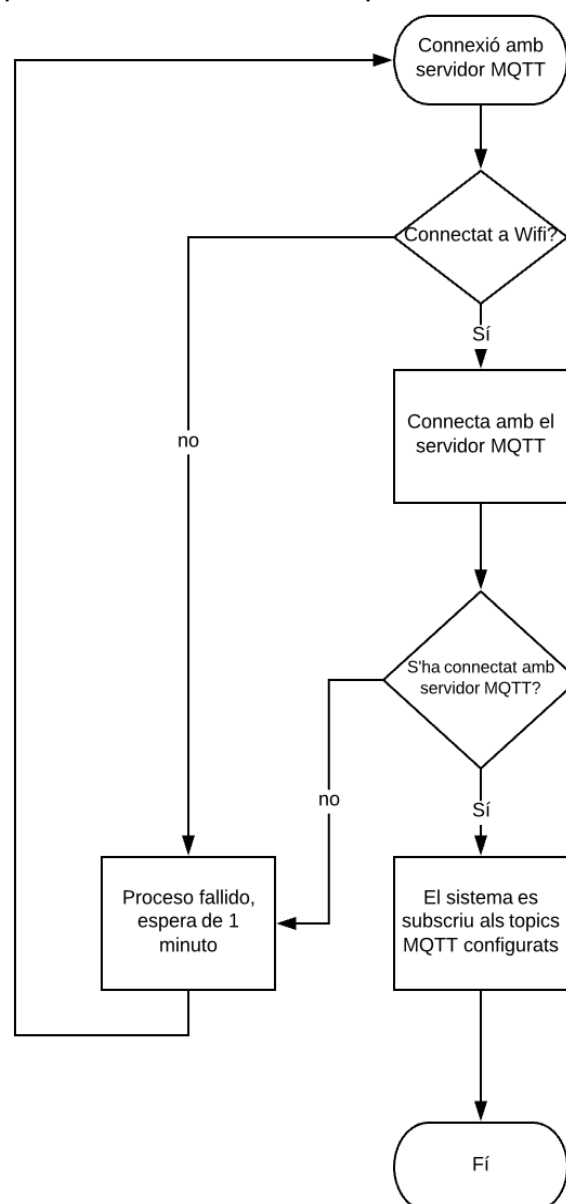
<sup>4</sup> NTP: Network time Protocol

<sup>5</sup> EEPROM: Electrically Erasable Programmable Read Only Memory

### 2.3.2. Connexió amb servidor MQTT

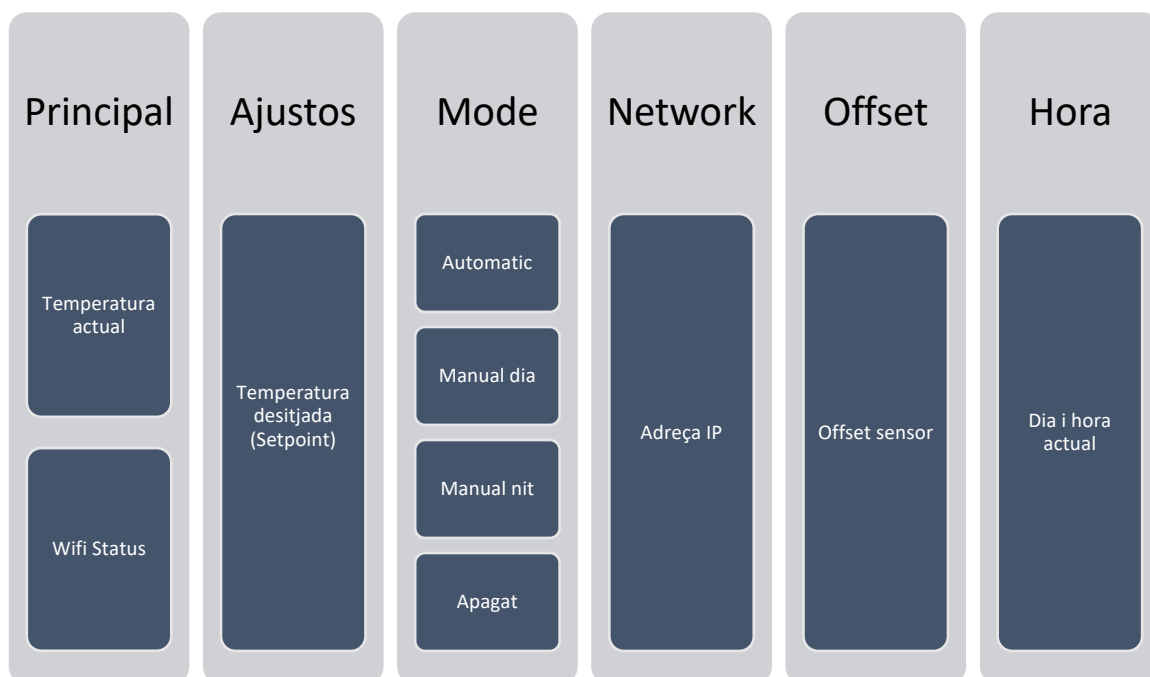
Si el sistema està connectat a la xarxa Wifi, intenta establir connexió amb el servidor MQTT, si ho aconsegueix, publica i es subscriu als tòpics establerts, en cas que no es comuniqui amb el servidor MQTT, llavors el sistema treballa sense connexió al servidor, encara que després d'un minut torna a intentar-ho. Aquest temps es pot augmentar, però per fer les proves es va determinar en un minut.

A continuació es detalla el diagrama de flux de connexió amb el servidor MQTT, es una tasca que s'executa en paral·lel a la resta de tasques.



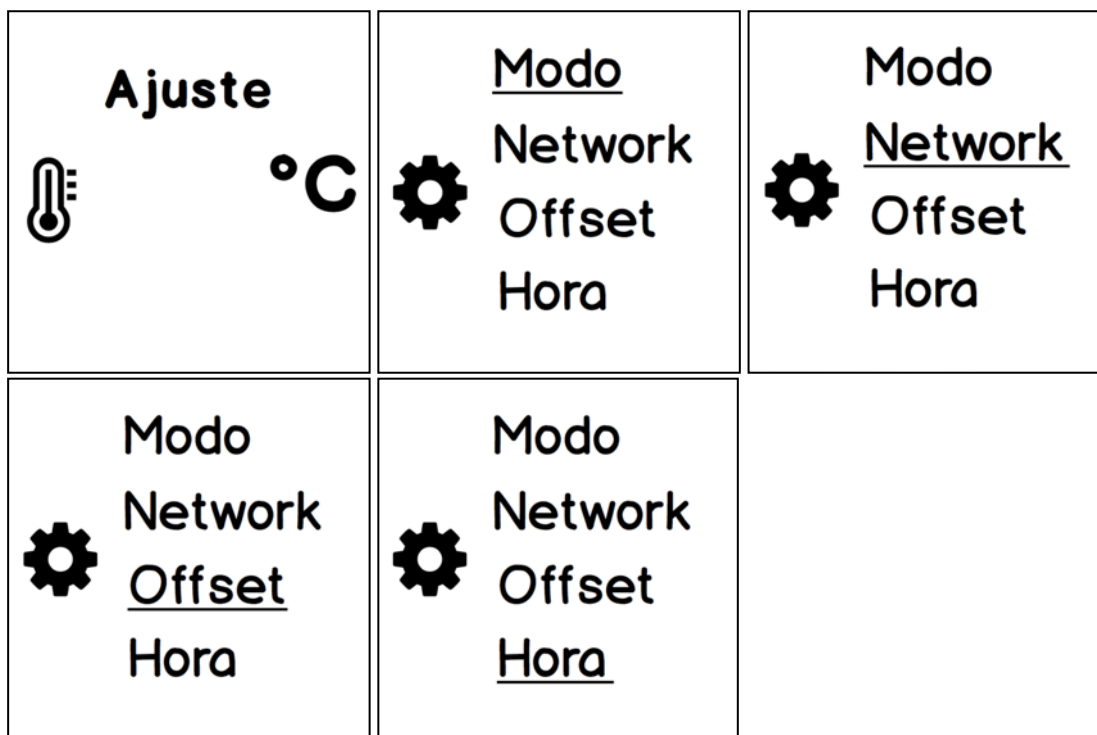
### 2.3.3. Descripció Menús

El sistema consta de 4 polsadors, que son (Menú, +, -, OK), amb el polsador Menú es navega pels menús del sistema, a continuació es pot veure un esquema lògic amb els menús i la informació que mostra a l'usuari.



A continuació, podem veure la representació gràfica dels menús en el display del sistema.





Per accedir-hi a qualsevol d'aquests menús, s'ha de prémer el polsador OK. Si no es polsa cap polsador durant 15 segons el sistema retorna automàticament a la pantalla inicial, una altre forma de tornar a la pantalla inicial es polsant Menú i OK a la vegada.

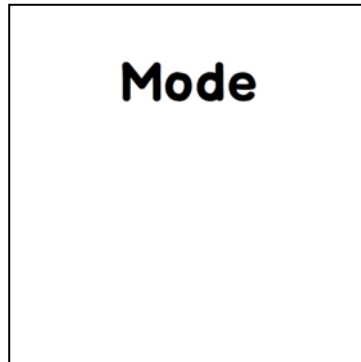
En el menú Ajust, es on l'usuari selecciona la temperatura desitjada que el sistema ha de mantenir, es el setpoint. Per canviar la temperatura s'ha de fer amb el polsadors (+ i -), per defecte incrementa o redueix el valor actual de temperatura desitjada en +/- 0.1 graus, si volem canviar la base s'ha de fer polsant el polsador Ok, llavors passa a +/- 1.0 grau, si tornem a polsar llavor tornem a la base 0.1 graus. Per acceptar el valor no es necessari polsar OK o cap altre botó, el sistema agafa l'últim valor i actualitza el sistema.

Si el sistema es troba en automàtic, llavors el sistema modifica el valor de l'array de treball, aquesta array es un array bi-dimensional de 7 x 24, que son el 7 dies de la setmana i 24 hores de cada dia. Si el sistema està en manual, la temperatura es modifica en la variable de dia o nit segons escollim.

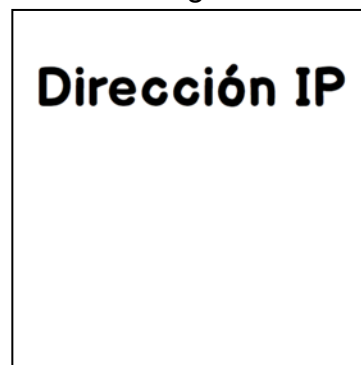
Els submenús que hi ha son els següents:

El menú Mode, es on l'usuari selecciona el mode de funcionament (Automàtic, Manual dia, Manual nit i Apagat).

El menú network, mostra la IP actual, si el sistema no està connectat llavors mostra 0.0.0.0 .

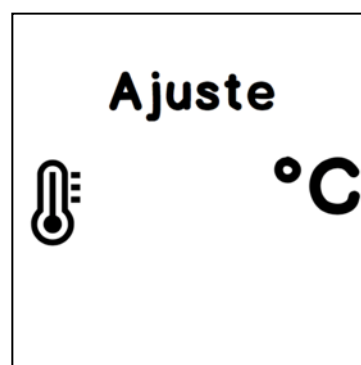


El dispositiu, està configurat per obtenir la adreça IP per DHCP, i així facilitar l'experiència d'usuari, per tant el nostre router ha de tenir configurada aquesta funcionalitat. No es disposa de la possibilitat de assignació de IP manual, per evitar la possibilitat de col·lisions entre dispositius de la xarxa, així com de configuracions errònies, per part de l'usuari.



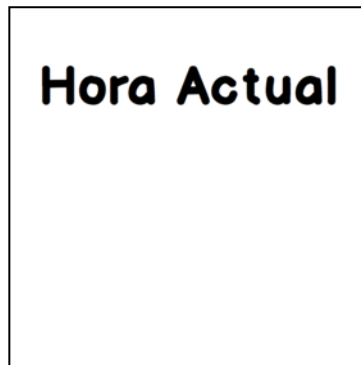
Si volem connectar el sistema o fer una reconexió a la xarxa, hem de prémer OK, el sistema llavors intenta connectar a la xarxa local configurada.

El menú Offset, es on l'usuari configura una temperatura per corregir la lectura del sensor instal·lat.



El funcionament en aquest menú es exactament el mateix que en el menú on ajustem la temperatura desitjada.

Per últim, el menú hora, on l'usuari pot veure el dia i l'hora actual, també si premem OK, el RTC es sincronitza amb el servidor NTP, en cas d'estar connectat amb la Wifi.



En tots els submenús, si premem el polsador Menú, tornem al menú principal.

El sistema sincronitza el RTC amb el NTP tots els diumenges a les 3 o a les 2 de la nit, depenent el mes on es troba, per tal de poder fer el canvi d'hora automàtic a la tardor i a la primavera. A part cada dia a las 4 del matí guarda les dades de treball a la EEPROM, per memoritzar les dades, en cas de reiniciar el sistema.



## 3. Plataforma Cloud

### 3.1. Detall protocol MQTT

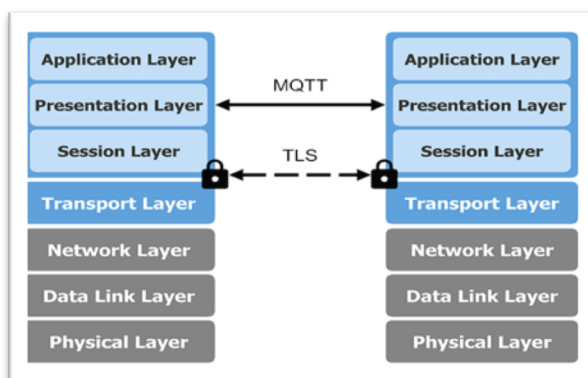
#### 3.1.1. Què es MQTT?

MQTT<sup>6</sup> es un protocol de missatgeria, Standard ISO, molt simple i lleuger, que es basa en publicació / subscripció. Es va dissenyar per dispositius que es troben en xarxes poc fiables amb alta latència i una amplada de banda molt limitada. Els principis d'aquest protocol era minimitzar l'amplada de banda i els recursos dels dispositius, però garantint la fiabilitat i en un grau la garantia de recepció de la informació. Aquests principis fan d'aquest un protocol ideal per la comunicació màquina-a-màquina (M2M) o per al món d'internet de les coses.

Aquest protocol va ser inventat al 1999, per Dr. Andy Stanford-Clark de IBM i Arlen nipper d'Arcom (ara Eurotech).

MQTT té uns ports definit que son el TCP/IP 1883, reservat amb IANA<sup>7</sup> per fer ús amb MQTT i el TCP/IP 8883 enregistrat per ser utilitzat per MQTT sobre SSL.

Com es pot veure a la següent imatge, el protocol MQTT es troba en les capes superiors del model OSI, per tant permet una codificació amb TLS.



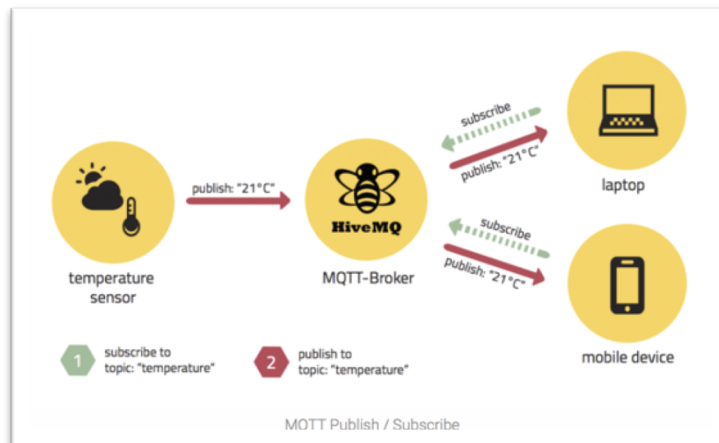
#### 3.1.2. Comunicació client - servidor mqtt

Una vegada el concepte de bróker o servidor MQTT està definit, passem a detallar breument el funcionament de la nostra arquitectura.

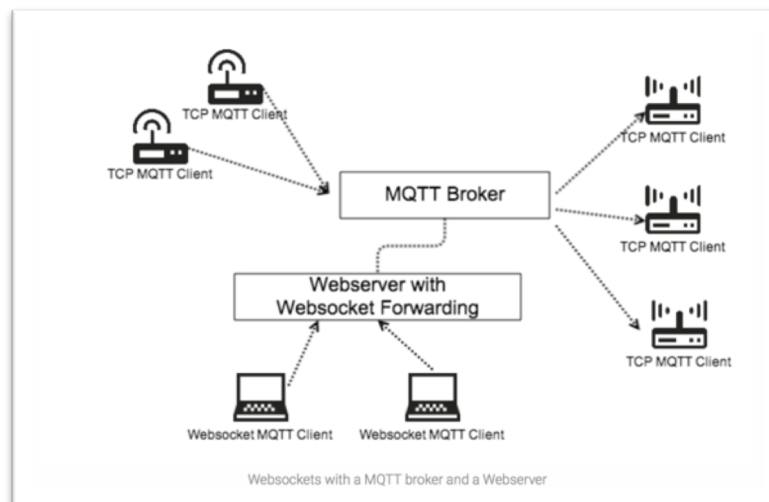
Com s'ha detallat en l'apartat anterior, la comunicació client-servidor es mitjançant el mètode publicació/subscripció, en la següent imatge es pot veure gràficament.

<sup>6</sup> MQTT: Messaging Queuing Telemetry Transport.

<sup>7</sup> IANA: es el coordinador global de els DNS, adreces IP i altres recursos de protocols d'internet.  
<http://www.iana.org>



En la següent imatge podem observar una arquitectura de varis clients MQTT que es comuniquen amb el servidor MQTT mitjançant els ports TCP/IP 1883 o 8883 com hem detallat anteriorment, i aquest servidor es comunica mitjançant Websockets amb uns altres clients MQTT, en aquest projecte s'han escollit els ports 9001 i 8083, ja que no hi han ports reservats per a una comunicació mitjançant sockets. El protocol de Websockets ja està integrat en tots els navegadors Web. Només es necessita una llibreria JavaScript, en aquest cas s'ha utilitzat Paho MQTT, es pot trobar detall de la llibreria en el següent link <https://www.eclipse.org/paho/clients/js/>.



### 3.2. Descripció de la solució escollida

Després d'explorar les funcionalitats de la plataforma, es va optar per crear una instància de Linux Ubuntu, amb les característiques que hi havia disponibles de forma gratuïta.



res de potencia de computació, gràfic o similar. Per tant el cost mensual seria de  $0,0132 \times 24 \times 31$  dies, \$ 9,82 /mes. A això s'ha de sumar el cost de l'assignació de una adreça IP fixa que es de \$0.010 per GB. Tenint en comte que el volum de dades es de l'ordre de KB, es un cost sense cap impacte.

Linux					
RHEL	SLES	Windows	Windows con SQL Standard	Windows con SQL Web	
Windows con SQL Enterprise		Linux con SQL Standard	Linux con SQL Web	Linux con SQL Enterprise	
Región:	UE (Londres)				
vCPU	ECU	Memoria (GiB)	Almacenamiento de la instancia (GB)	Uso de Linux/UNIX	
<b>Uso general – Generación actual</b>					
t2.nano	1	Variable	0.5	Solo EBS	\$0.0066 por hora
t2.micro	1	Variable	1	Solo EBS	\$0.0132 por hora
t2.small	1	Variable	2	Solo EBS	\$0.026 por hora
t2.medium	2	Variable	4	Solo EBS	\$0.052 por hora

### 3.3. Temps d'implementació

Aquest canvi de plataforma, ha afectat greument a la planificació inicial, ja que s'han destinat moltes hores a la elecció de la nova solució, valorant funcionalitat, cost, impacte en temps, etc. Malgrat aquest impacte, es valorar que els beneficis d'aquesta nova implementació superaven el factors negatius, ja que el desenvolupament inicial, només tindrà un impacte econòmic a curt termini, en canvi serà inexistent a llarg termini.

Finalment s'han destinat 30 hores només a la configuració del servidor MQTT. Aquesta canvi s'ha detallat en una nova versió del diagrama de Gantt i en l'estudi econòmic de viabilitat del producte.

- Estudi plataforma Cloud ----- 5 hores
- Creació Instancies ----- 5 hores
- Configuració del sistema operatiu ----- 2 hores
- Estudi del broker MQTT ----- 3 hores
- Instal·lació del broker MQTT ----- 1 hora
- Configuració bàsica broker MQTT ----- 8 hores
- Tests inicials ----- 4 hores
- Configuració seguretat tòpics ----- 2 hora



### 3.4. Configuració del broker MQTT

La configuració del broker o servidor MQTT, resideix en el fitxer mosquitto.conf, en aquest cas s'ha renombrat el fitxer a test.conf.

```

test.conf x
# Config file for mosquitto
#
# See mosquitto.conf(5) for more information.
#
# Default values are shown, uncomment to change.
#
# Use the # character to indicate a comment, but only if it is the
# very first character on the line.

# =====
# General configuration
# =====

# =====
# Puertos y protocolos
# =====

listener 1883
listener 8883
listener 9001
protocol websockets
listener 8083
protocol websockets
# =====
# Certificados SSL
# =====

#cafile /etc/moquitto/certs/ca.crt
#certfile /etc/mosquitto/certs/server.crt
#keyfile /etc/mosquitto/certs/server.key

#tls_version tlsv1.1
#require_certificate true

# =====
# Seguridad
# =====
allow_anonymous false
password_file /etc/mosquitto/passwd
  
```

Com es pot veure la part dels certificats SSL, està des-habilitada, ja que s'ha considerat deixar-ho com possibles millores a implementar en el futur. Al igual que les llistes d'accés ACL i l'arquitectura bridge entre servidors MQTT, anteriorment descrita.

Per últim, la part de seguretat, s'ha forçat als clients MQTT a enviar l'usuari i el password per poder publicar o subscriure's a un tòpic.

Es va crear un fitxer de passwords, anomenat passwd amb la contrasenya encriptada, a continuació es pot observar com es:

```

passwd x
aescobarf:$6$GtY11yTwfWVSPB/p$iyZjTAFIUXLD+8EkLZzUace8GWE09VeNTrSdhaqUYc2o95sD6p1LOEGPMB/x9oXLQth9sKiw+FF0iDFEzhkp1g==
smar ttemp:$6$F8pN1bkC4YS6a6BI$QCb0R0ywxZADDHhtXd70PIEaHxd6npMnS0bWHzg f30Ym4pVr1MLT6xP1g9Aa30/EFk/VCfMZL4HG2Tqpysgd/Q==
  
```

No s'han implementat les regles ACL<sup>8</sup>, ja que no son necessàries per al projecte, encara que si es fa escalable o el broker MQTT es fa accessible per altres clients, seria recomanable. Així com la configuració de bridge amb altres servidors MQTT, tampoc necessaris per aquest projecte.

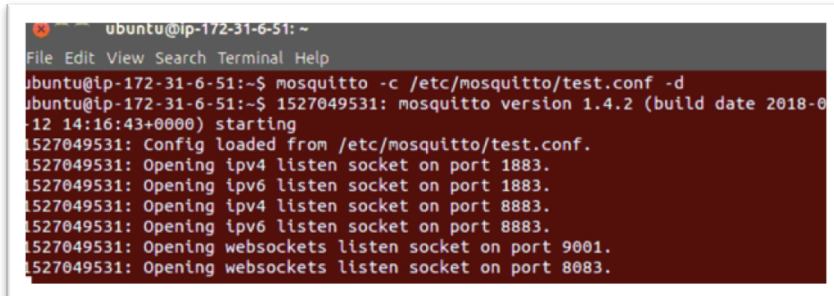
La configuració de bridge es tractaria de una comunicació entre dos brokers MQTT, per tant es per tenir un MQTT server local i aquest comunica amb el server remot, això obligaria al client a tenir i mantenir el broker MQTT local per tenir accés remot, es un opció que es pot deixar per futures implementacions.

---

<sup>8</sup> ACL: Llistes de control d'accés, s'utilitzen quan hi ha més d'un usuari i/o dispositius, i es determina quin nivell d'accés tenen, per exemple llegir, escriure o tot dos, a nivell de tòpic o a nivell fins i tot de dispositiu.

### 3.4.1. Com funciona el servidor MQTT

Per arrencar el servidor MQTT al servidor Linux, es fa mitjançant la següent comanda:  
`mosquitto -c /etc/mosquitto/test.conf -d`



```

ubuntu@ip-172-31-6-51: ~
File Edit View Search Terminal Help
ubuntu@ip-172-31-6-51:~$ mosquitto -c /etc/mosquitto/test.conf -d
ubuntu@ip-172-31-6-51:~$ 1527049531: mosquitto version 1.4.2 (build date 2018-0
12 14:16:43+0000) starting
1527049531: Config loaded from /etc/mosquitto/test.conf.
1527049531: Opening ipv4 listen socket on port 1883.
1527049531: Opening ipv6 listen socket on port 1883.
1527049531: Opening ipv4 listen socket on port 8883.
1527049531: Opening ipv6 listen socket on port 8883.
1527049531: Opening websockets listen socket on port 9001.
1527049531: Opening websockets listen socket on port 8083.
  
```

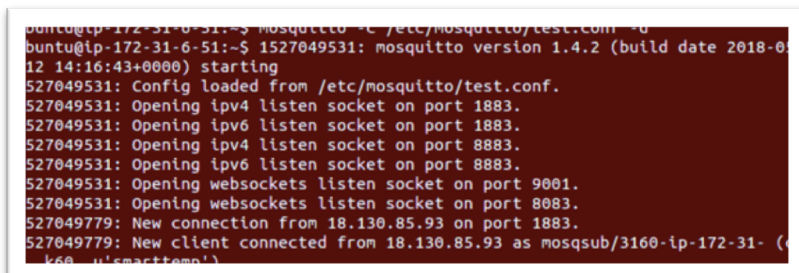
A la imatge podem veure com el servidor MQTT està arrencat, escoltant els ports MQTT 1883 i 8883 i els ports websockets 9001 i 8083, per la connexió de la web.

Arranquem un client amb la següent ordre per estar subscrit a tots els tòpics que es publicuin que pertanyin al usuari smarttemp, el path d'aquests tòpics es el següent:  
`/smarttemp/#`

```

ubuntu@ip-172-31-6-51:~$ mosquitto_sub -t "/smarttemp/#" -v -h "18.130.85.93" -u
"smarttemp" -P "12345" -p 1883
  
```

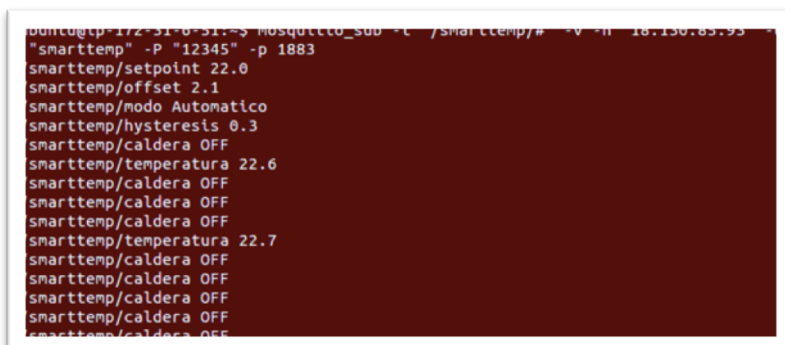
A la imatge veiem la comanda `mosquitto_sub` amb el tòpic, la adreça IP del servidor l'usuari i la contrasenya anteriorment definits al fitxer `passwd`. Una vegada executat la comanda, podem veure al servidor com hi ha un client connectat.



```

ubuntu@ip-172-31-6-51:~$ mosquitto -c /etc/mosquitto/test.conf -d
ubuntu@ip-172-31-6-51:~$ 1527049531: mosquitto version 1.4.2 (build date 2018-0
12 14:16:43+0000) starting
1527049531: Config loaded from /etc/mosquitto/test.conf.
1527049531: Opening ipv4 listen socket on port 1883.
1527049531: Opening ipv6 listen socket on port 1883.
1527049531: Opening ipv4 listen socket on port 8883.
1527049531: Opening ipv6 listen socket on port 8883.
1527049531: Opening websockets listen socket on port 9001.
1527049531: Opening websockets listen socket on port 8083.
1527049779: New connection from 18.130.85.93 on port 1883.
1527049779: New client connected from 18.130.85.93 as mosqsub/3160-ip-172-31-
k60 - u'smarttemp'
  
```

A continuació el client comença a enregistrar les dades que l'arduino va publicant.



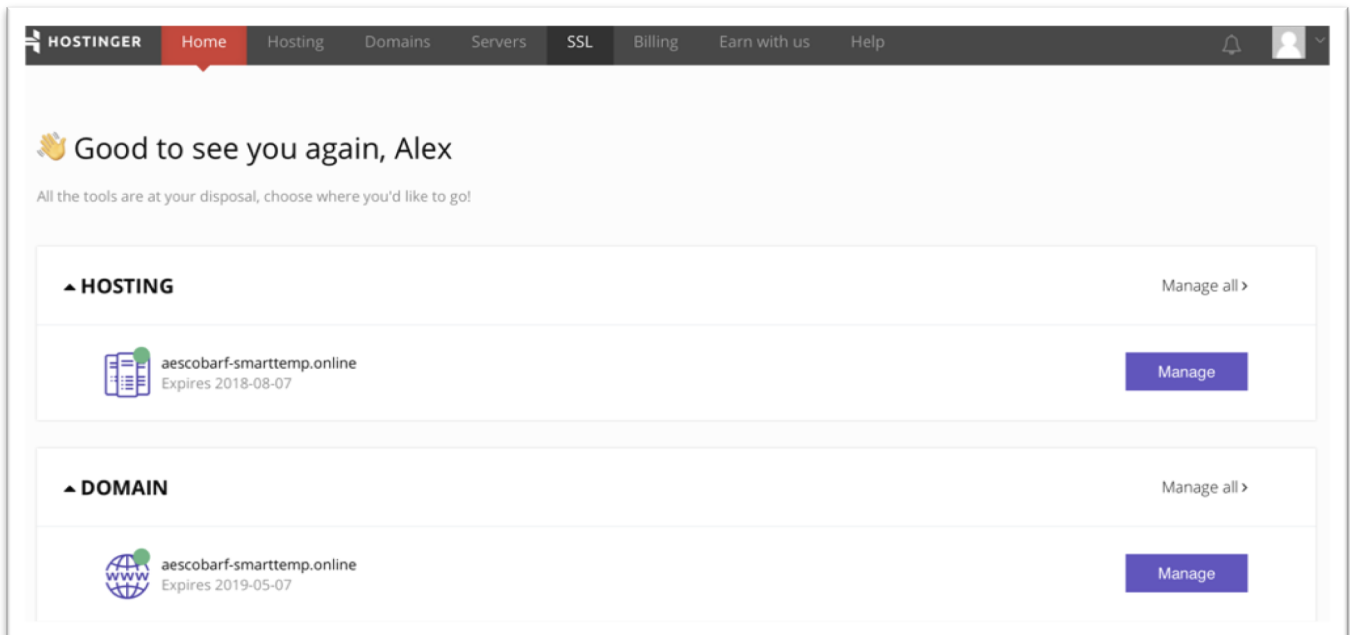
```

ubuntu@ip-172-31-6-51:~$ mosquitto_sub -t "/smarttemp/#" -v -h "18.130.85.93"
"smarttemp" -P "12345" -p 1883
smarttemp/setpoint 22.0
smarttemp/offset 2.1
smarttemp/modo Automatico
smarttemp/hysteresis 0.3
smarttemp/caldera OFF
smarttemp/temperatura 22.6
smarttemp/caldera OFF
smarttemp/caldera OFF
smarttemp/caldera OFF
smarttemp/temperatura 22.7
smarttemp/caldera OFF
smarttemp/caldera OFF
smarttemp/caldera OFF
smarttemp/caldera OFF
  
```

## 4. Pàgina web

### 4.1. Descripció general

Per la creació de la pàgina web primerament es va escollir el proveïdor de hosting, en aquest cas inicialment es va fer mitjançant [www.hostinger.com](http://www.hostinger.com). El domini escollit va ser [aescobarf-smarttemp.online](http://aescobarf-smarttemp.online).



Encara que després de fer la implementació de la plataforma cloud a Amazon AWS, es migraria el hosting a AWS S3.

La pagina web s'ha creat a partir de codi HTML i CSS per el format gràfic d'aquesta. El sistema de ingrés de l'usuari mitjançant usuari i contrasenya, s'ha implementat mitjançant PHP, amb interacció amb MySQL on s'ha declarat l'usuari i el password corresponent. Com podem veure a la següent imatge es tracta d'una base de dades molt senzilla.

	MySQL Database	MySQL User	MySQL Host	Disk Usage, MB
+	u685262745_db	u685262745_admin	mysql.hostinger.com	0.00



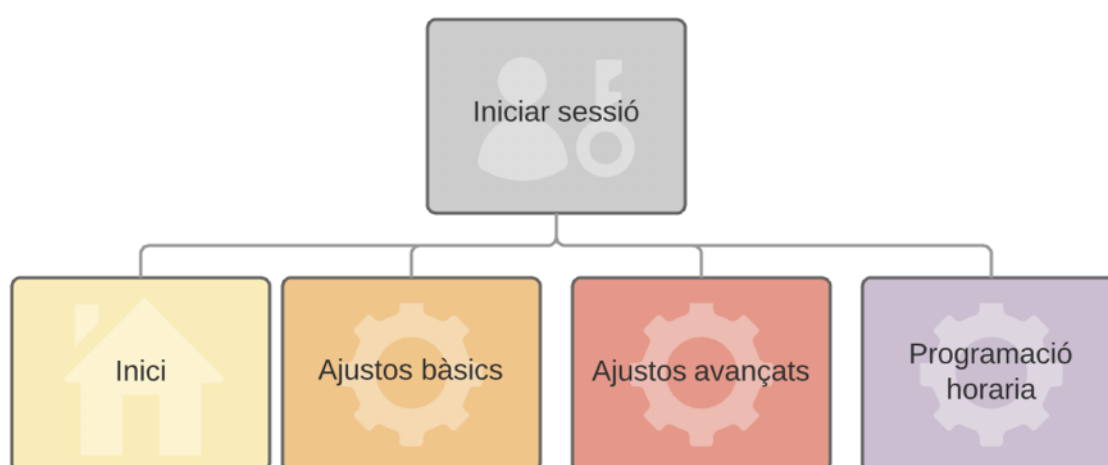
Com es pot observar, s'ha fet que l'usuari sigui clau i per tant ha de ser únic no es pot repetir, encara que en un futur si escau es podria modificar la base de dades en funció a les

necessitats. La interacció amb la base de dades ens donaria la possibilitat de fer un històric de les temperatures mesurades i així poder fer gràfiques, etc. Aquesta funcionalitat no ha sigut implementada en aquesta versió inicial.

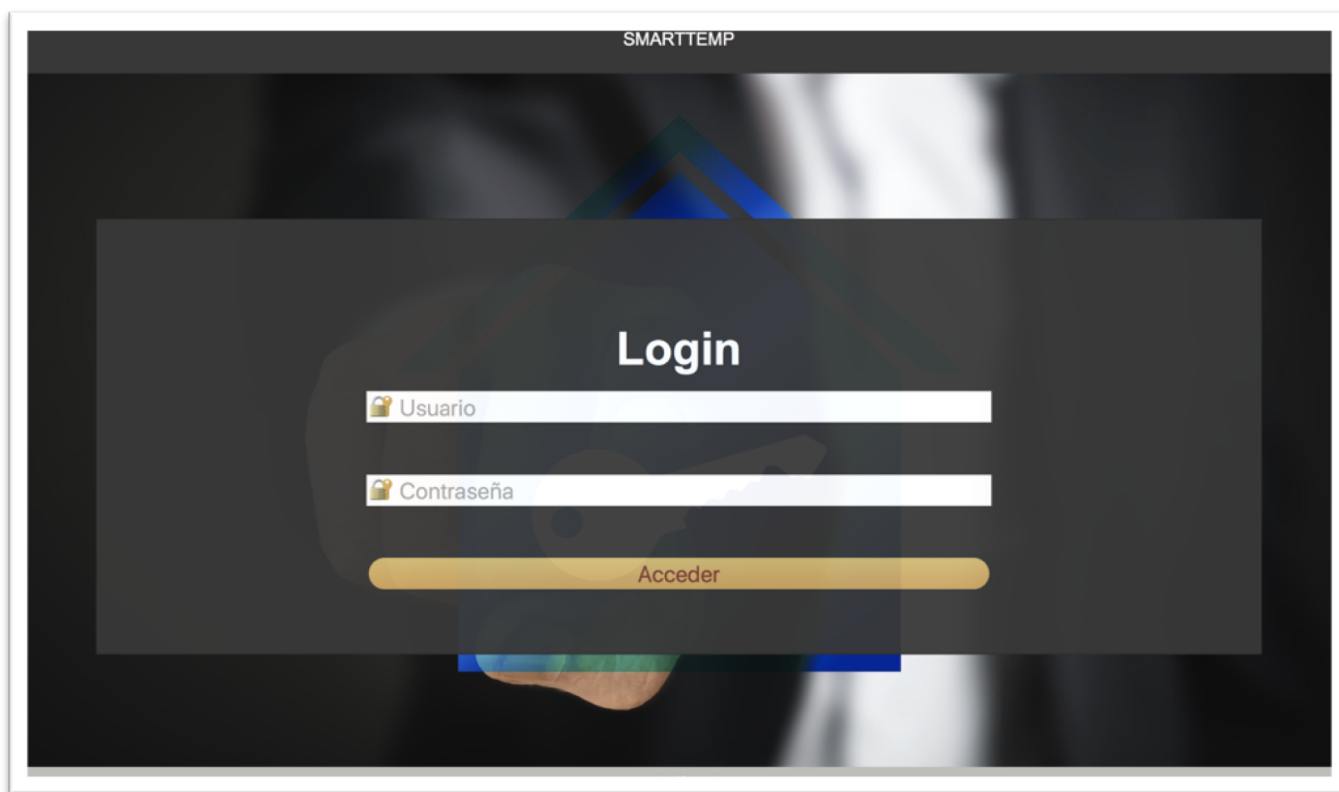
La part de la comunicació de la web amb el servidor MQTT s'ha fet mitjançant JavaScript, fent ús de la llibreria Paho MQTT client, les pàgines webs apunten a aquesta adreça <https://cdnjs.cloudflare.com/ajax/libs/paho-mqtt/1.0.1/mqttws31.js>

## 4.2. Visió general web

### 4.2.1. Mapa del lloc web

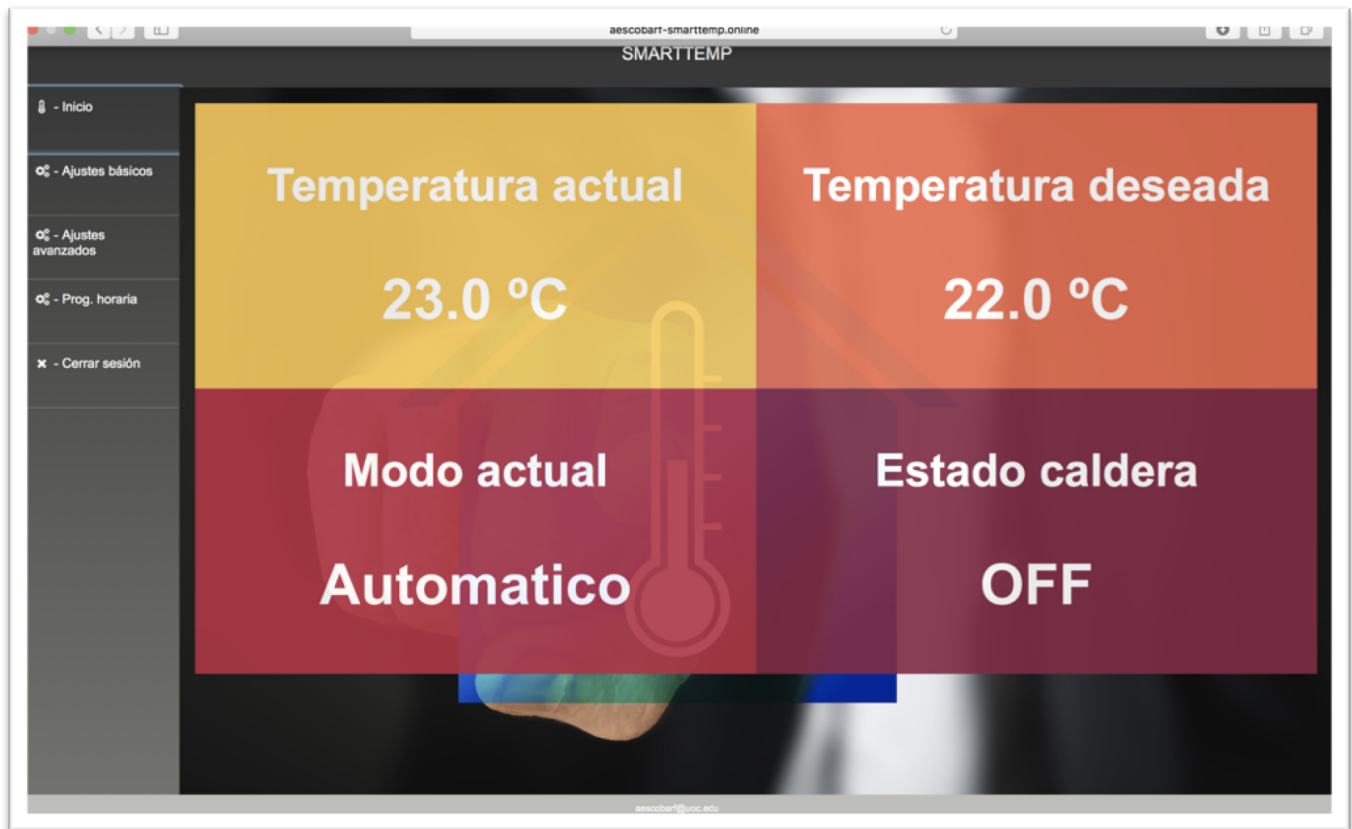


## 4.2.2. Pagina Inici (login)



Aquí introduïrem l'usuari i la contrasenya. Una vegada es premi Accedir, llavors el sistema buscarà l'usuari i el password a la base de dades i si es correcte donarà accés a la pagina principal, si es incorrecte no donarà accés. Si un usuari intenta accedir-hi a una de les pagines secundaries sense identificar-se, el sistema o detecta i el direccionaria a aquesta pàgina d'inici.

### 4.2.3. Pàgina principal



En aquesta pàgina inicial, l'usuari pot visualitzar les dades més importants del sistema, com poden ser la temperatura actual, temperatura desitjada, el mode de funcionament i l'estat de la caldera. Es per tant només de visualització. Per canviar qualsevol dada s'ha d'accedir a ajustos bàsic , avançats o programació horària.

El sistema una vegada carrega la pàgina, envia la comanda al brokerMQTT mitjançant el tòpic /smarttemp/peticion amb el valor 1, el node client, en aquest cas l'arduino que està subscrit, detecta el valor 1 i retorna el valor del setpoint programat, l'offset el mode de funcionament, el valor d'histèresis, l'estat de la caldera i la temperatura ideal. El format del tòpic sempre es el mateix, /usuari /+ comanda

```

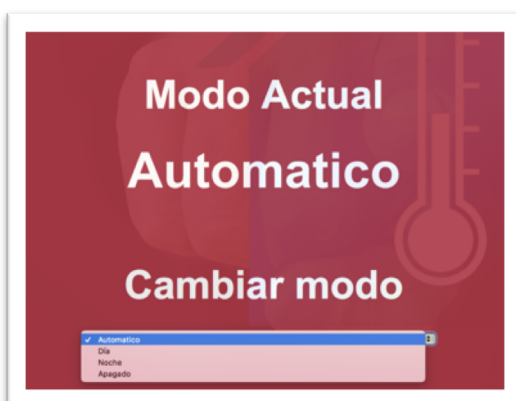
/smarttemp/peticion 1
/smarttemp/setpoint 22.0
/smarttemp/offset 2.1
/smarttemp/modo Automatico
/smarttemp/hysteresis 0.3
/smarttemp/caldera OFF
/smarttemp/temperatura 22.6
  
```

#### 4.2.4. Ajustos bàsics



En aquesta pàgina Ajustos bàsics, l'usuari pot canviar el setpoint (temperatura desitjada) i el mode actual de treball.

El setpoint es canvia només variant el valor, una vegada acceptem el canvi el sistema envia la comanda a l'arduino perquè l'actualitzi amb el nou valor. Exactament el mateix funcionament amb el mode de funcionament, només que el sistema retorna el nou mode de treball per mostrar-lo visualment i el setpoint associat a aquest mètode de treball.







Els valors de la temperatura i l'estat de la caldera, es van actualitzant regularment com en totes les pàgines on hi són.

Com a la pàgina inicial, quan es carrega la pàgina, s'envia una comanda com la següent:

```
/smarttemp/peticion 2
/smarttemp/temperatura 24.2
/smarttemp/setpoint 22.0
/smarttemp/modo Automatico
/smarttemp/caldera OFF
```

Si l'usuari canvia el valor de mode de funcionament, llavors el sistema enviaria la comanda amb el següent format: /smarttemp/setup/modo i el valor del nou mode de funcionament.

Una vegada l'arduino detecta la ordre de setup i mode, canvia el mode de funcionament i retorna a la pàgina web el nou valor de setpoint i el mode de funcionament.

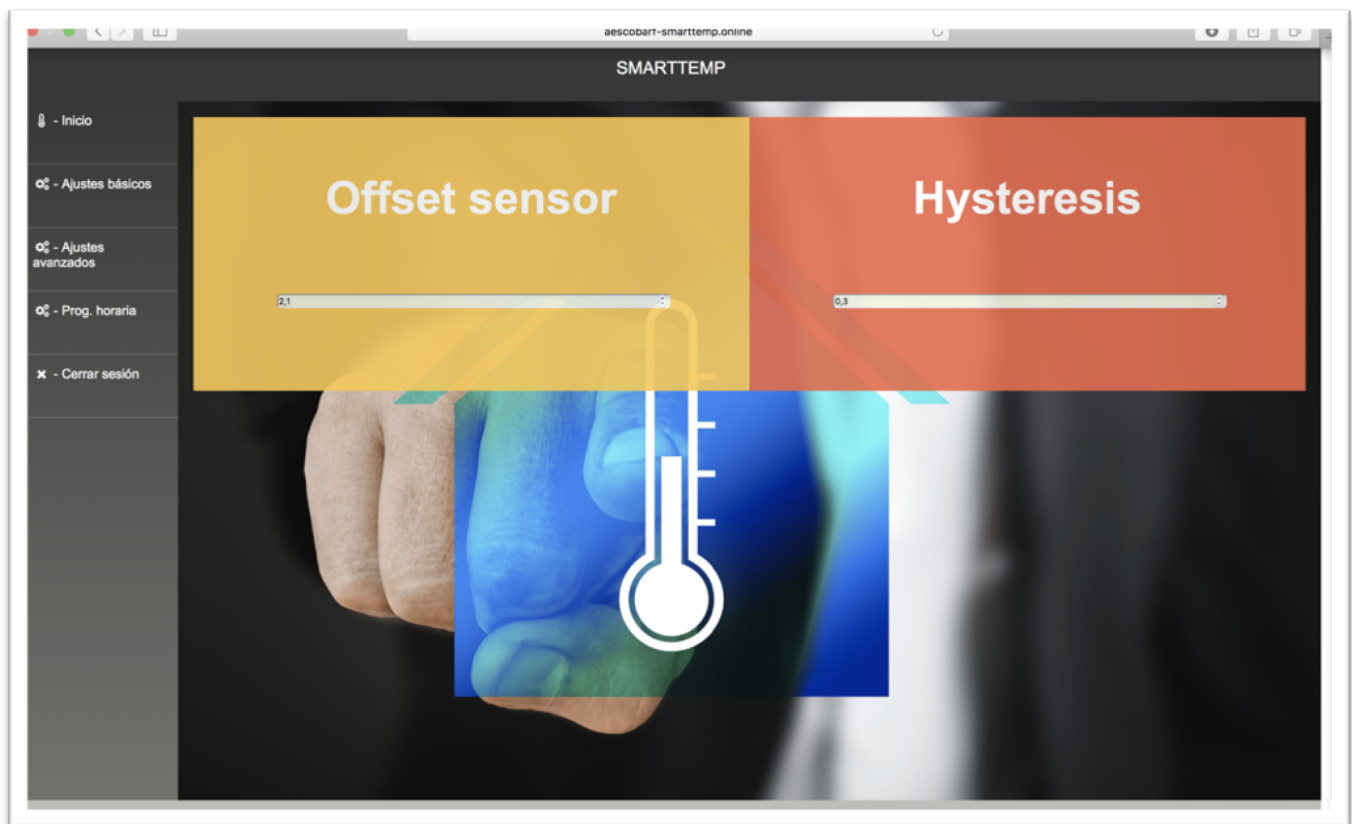
```
/smarttemp/setup/modo 1
/smarttemp/setpoint 21.1
/smarttemp/modo Manual dia
```

Si en lloc de canviar el mode de treball, l'usuari canvia el setpoint, llavors el sistema només envia la comanda de setup del setpoint, ja que el nou valor ja el mostra la web i per tant estalviem l'enviament de dades per la xarxa. El tòpic aquesta vegada seria

```
/smarttemp/setup/setpoint i el valor.
```

```
/smarttemp/setup/setpoint 20
```

## 4.2.5. Ajustos avançats



En aquesta pàgina, ajustos avançats, es poden configurar dos valors molt importants per el bon funcionament del sistema, el valor d'offset per poder calibrar el sensor, ja que depenent on s'instal·li el sistema, pot variar la temperatura de mesura. L'altre valor es la histèresis, es un valor que evita que la caldera estigui constantment encenent i apagant-se. Per exemple, si l'usuari configura 0,3, valor per defecte, el sistema configura:

valor d'activació de la caldera = setpoint – histèresis

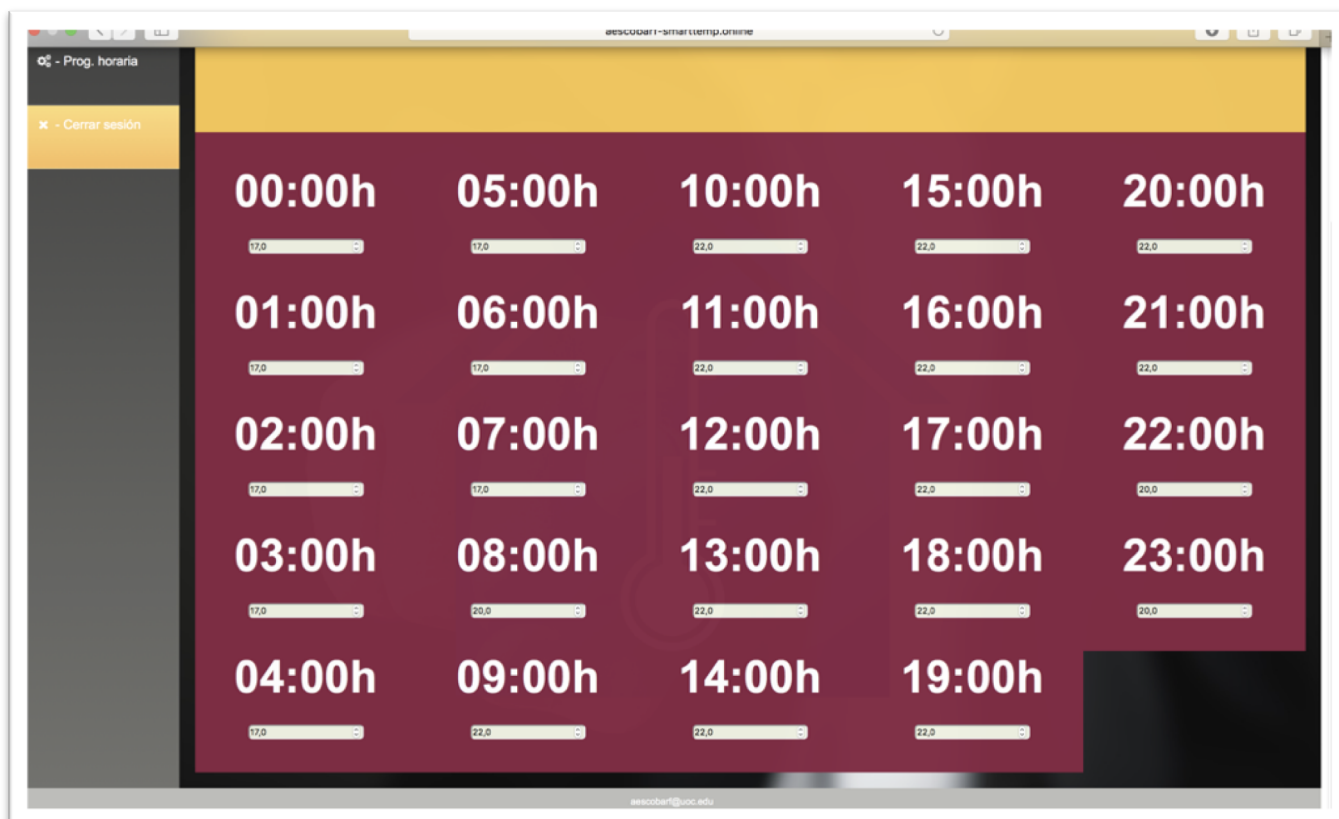
valor de desactivació de la caldera = setpoint + histèresis

Quan aquest a pàgina es carrega, com s'ha detallat anteriorment, envia la següent ordre:

```
/smarttemp/peticion 3
/smarttemp/offset 2.1
/smarttemp/hysteresis 0.3
```

Per tant el sistema indica a l'arduino que ha de carregar les dades de la pàgina 3, que son l'offset i el valor d'histèresis. El sistema d'actualització d'aquestes dades es exactament el mateix que el setpoint.

### 4.2.6. Programació horària



En aquesta pàgina, programació horària, l'usuari pot configurar hora per hora el setpoint de cada dia de la setmana de Diumenge a Dissabte. El funcionament d'aquesta pàgina es el

següent, quan es carrega s'envia l'ordre `/smarttemp/setup/programación` + el numero corresponent al dia que es vol modificar, per defecte la primera vegada sempre es carrega el diumenge, per tant envia el valor 0. Els valors dels dies de la setmana serien (Diumenge = 0, Dilluns = 1, Dimarts = 2, Dimecres = 3, Dijous = 4, Divendres = 5, Dissabte = 6).

El sistema retorna el valor del setpoint per cada hora del dia escollit, amb el següent format `/smarttemp/programación/XX` (valor de la hora que va de 0 a 23) i el valor del setpoint. S'ha escollit aquest format per simplificar el codi de la web.

Exemple del intercanvi de informació:

```

/smarttemp/setup/programacion 0
/smarttemp/programacion/0 17.0
/smarttemp/programacion/1 17.0
/smarttemp/programacion/2 17.0
/smarttemp/programacion/3 17.0
/smarttemp/programacion/4 17.0
/smarttemp/programacion/5 17.0
/smarttemp/programacion/6 17.0
/smarttemp/programacion/7 17.0
/smarttemp/programacion/8 20.0
/smarttemp/programacion/9 22.0
/smarttemp/programacion/10 22.0
/smarttemp/programacion/11 22.0
/smarttemp/programacion/12 22.0
/smarttemp/programacion/13 22.0
/smarttemp/programacion/14 22.0
/smarttemp/programacion/15 22.0
/smarttemp/programacion/16 22.0
/smarttemp/programacion/17 22.0
/smarttemp/programacion/18 22.0
/smarttemp/programacion/19 22.0
/smarttemp/programacion/20 22.0
/smarttemp/programacion/21 22.0
/smarttemp/programacion/22 20.0
/smarttemp/programacion/23 20.0
  
```

Si l'usuari canvia un valor qualsevol de setpoint per a una hora en concret, la web envia la següent ordre:

```

/smarttemp/setup/planificacion 00018.0
  
```

El format de l'ordre es el tòpic `/smarttemp/setup/planificacion` i a continuació s'envia el numero del dia + l'hora + el valor a modificar. L'arduino accedeix al valor corresponent de la seva matriu de setpoints i el modifica.

#### 4.2.7. Tancar sessió

Per finalitzar, si l'usuari prem l'opció de tancar sessió del menú de navegació, el sistema desloga l'usuari i retorna a l'usuari a la pantalla d'inici per a que introdueixi de nou l'usuari i la contrasenya.

## 5. Detall del software

### 5.1. Software programació ESP32

El codi del microcontrolador ESP32, s'ha desenvolupat mitjançant la plataforma IDE d'arduino Standard. Com el microcontrolador ESP32 no es d'arduino, el software no està adaptat per detectar i poder compilar el programa per aquesta placa. Per tant, s'han de fer unes passes per poder adaptar el software, incloent tot el repositori necessari per això. El procediment necessari per fer aquesta adaptació està descrit al Annex.

D'una forma resumida, ens hem de baixar de la xarxa, el programa Python, l'hem d'instal·lar al nostre ordinador, per després descarregar-nos les llibreries de ESP32 i mitjançant python, adaptar aquestes llibreries d'Expressif a Arduino, Per últim agafem les carpetes creades durant aquest procés i les copiem a les subcarpetes del IDE d'arduino.

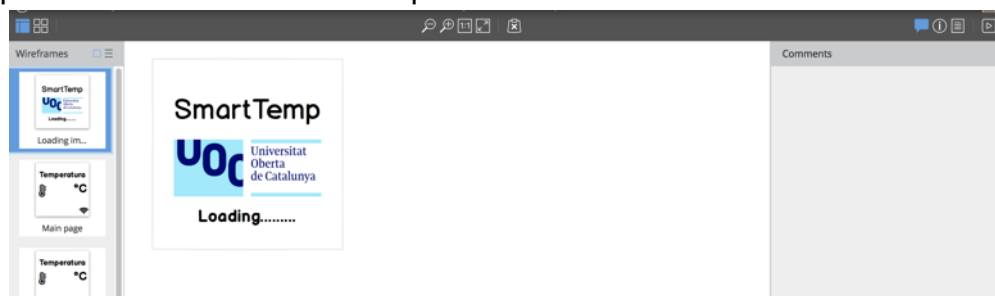
Després arrenquem el IDE d'arduino, i a les plaques disponibles ja apareixerà el ESP32. Tal i com he dit anteriorment, el procediment està detallat amb links i gràficament, pas per pas, a l'annex.

### 5.2. Software disseny pantalles

El programa utilitzat per desenvolupar les pantalles, ha sigut l'eina online balsamiq <https://balsamiq.cloud/s1gdlbk/projects>, es un portal Web on es poden crear wireframes, al següent link es pot trobar més informació i un vídeo on es pot veure els seus serveis <https://docs.balsamiq.com/cloud/intro/>.

Inicialment es pot crear un projecte de forma gratuïta, encara que als 30 dies la versió trial expira, i s'ha de contractar el servei.

Amb aquesta eina, he creat totes les pantalles, ja que et dona la possibilitat de fer us d'icones, i poder creacions amb moltes possibilitats. Es molt intuïtiu i senzill.



### 5.3. Software creació fitxers .C per la pantalla

En el següent link [https://www.waveshare.com/wiki/1.54inch\\_e-Paper\\_Module](https://www.waveshare.com/wiki/1.54inch_e-Paper_Module) trobem tota la informació necessària per cablejar, crear pantalles, exemple de codi per l'arduino, etc.

El software per la creació dels fitxers .C es Image2Lcd.exe i es pot baixar del següent link <https://www.waveshare.com/wiki/File:Image2Lcd.7z> . Aquest programa només es compatible amb Windows.

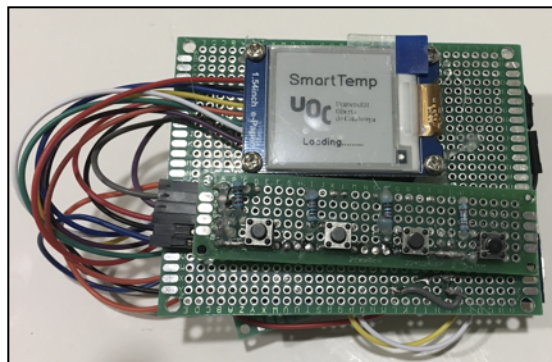
Una vegada tenim la imatge en format .bmp monocrom, ja que aquesta pantalla utilitzada es blanc i negre, la carreguem en aquesta aplicació.

Seleccionem el mètode d'escaneig vertical o horitzontal, aquest mètode dependrà de l'orientació de la pantalla en el muntatge final, ja que d'això dependrà com es mostrarà la imatge. Una altre cosa important es la dimensió de la imatge en nombre de píxels, en aquest cas la pantalla es de 200 x 200.

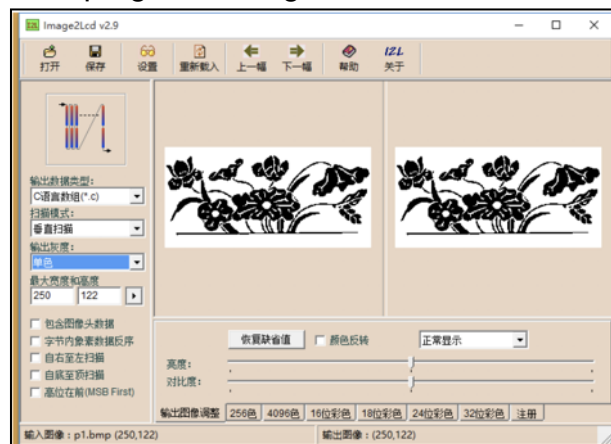
Per últim, hem de decidir si la volem mostrar amb el fons en blanc o en negre, si volem la imatge amb el fons en blan, llavors hem d'invertir la imatge. Com en aquest exemple:



Invertint els colors, la pantalla mostrarà el fons en blanc i el text en negre, com podem veure a la següent imatge:



A continuació un exemple del programa Image2Lcd.exe.



Una vegada ja està importat en la aplicació, el guardaríem i es crearia un fitxer .C, hauríem de copiar el contingut i enganxar-lo en un fitxer \*.h, que es el que l'arduino crida per carregar la imatge.

### 5.4. Disseny pàgina web

Per el codi de la pàgina web s'ha utilitzat el software Sublime Text, es tracta d'un programa compatible per Windows i Mac, amb una versió gratuïta totalment funcional.

### 5.5. Esquemes elèctrics

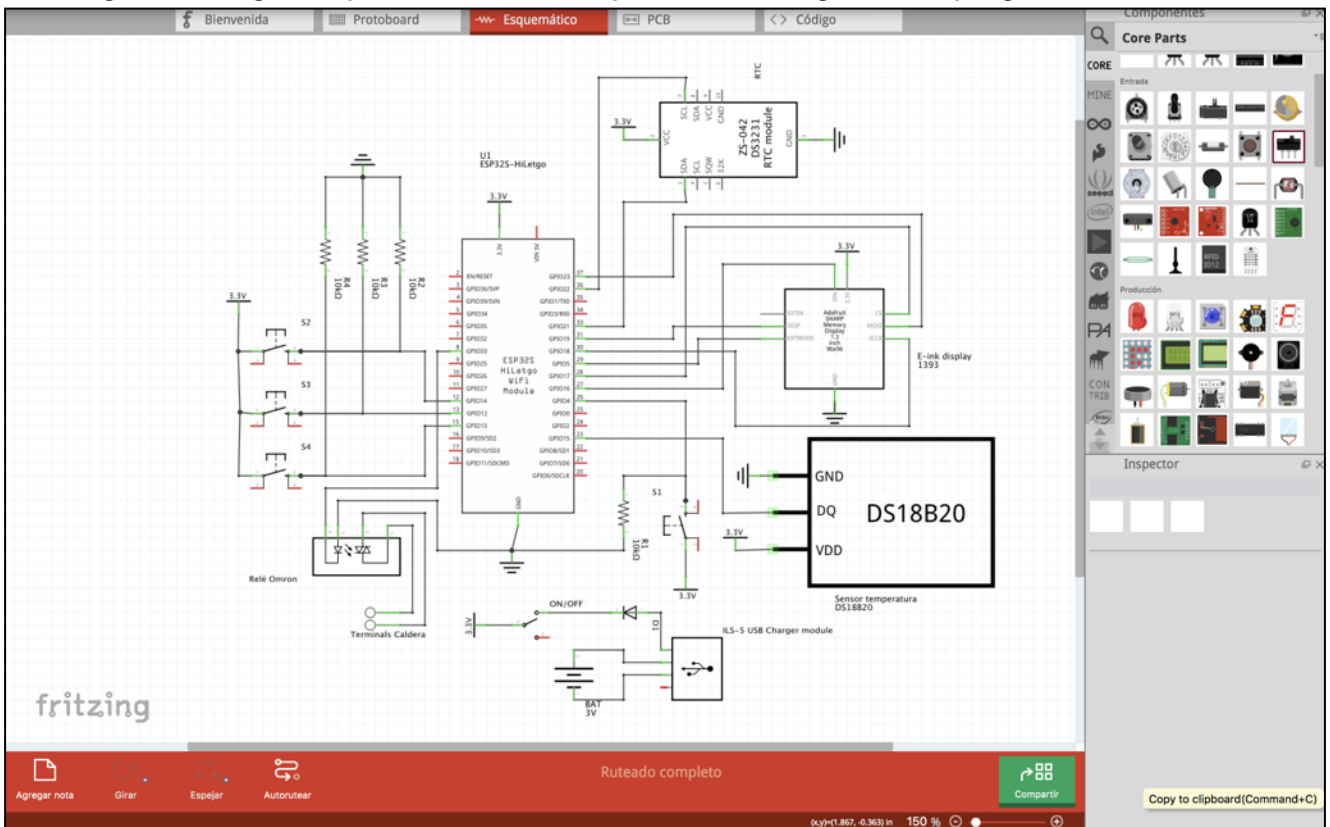
El programa utilitzat per fer els esquemes elèctrics es el Fritzing en la seva versió 0.9.3b, compatible amb Windows, Linux i OSX. Es pot descarregar des del següent link

<http://fritzing.org/home/>

Es un programa, on tots el fabricants d'elements electrònics per Arduino, creen les seves llibreries per poder importar els seus dispositius en forma gràfica, i així poder crear l'esquema elèctric del circuit. Es una eina totalment gratuïta.

Un problema trobat amb aquest software, es que algunes llibreries no estan actualitzades i fallen en l'escala que es mostren. Per exemple, en aquest disseny el sensor de temperatura DS18B20, com es pot veure té una escala molt gran, en comparació amb la resta de dispositius. Es un problema reportat en els fòrums del programa.

A la següent imatge, es pot veure un exemple del entorn gràfic del programa:



## **5.6. Diagrames de flux**

Tots els diagrames de flux, s'han realitzat amb la eina online de la pàgina web <https://www.lucidchart.com> .



## 6. Conclusions

En aquest apartat es comenten les conclusions una vegada finalitzat el projecte, a més de proposar les possibles millores i per últim es va fer una valoració personal del projecte elaborat.

### 6.1. Implementació

Una vegada el projecte s'ha finalitzat satisfactòriament, es detalla que s'han implementat la funcionalitats establertes inicialment.

Se ha integrat un sensor de temperatura a una placa d'Espressif ESP32, a més d'una pantalla de tinta electrònica, un rellotge RTC, un relé Omron, i una sèrie de polsadors per poder navegar pels menús creats. A més a més, s'ha instal·lat una bateria reutilitzada d'un mòbil antic Nokia, amb un mòdul de carrega.

Es va escollir la placa ESP32, perquè disposava de connexió Wifi integrada, i per tant no es necessita la configuració de cap mòdul extern Wifi.

S'ha configurat un servidor virtual, fent ús del servei de Cloud Computing d'Amazon AWS, s'ha associat una adreça IP fixe, per tal de comunicar sense problemes amb el node d'Arduino, i per últim, s'ha instal·lat un servidor i client MQTT en aquest servidor. Inicialment, no es contemplava aquesta necessitat, ja que la pàgina CloudMqtt, ja donava el servei gratuït, però durant la implementació es va detectar una limitació que evitava que el projecte pogués dur-se a terme amb èxit.

Una vegada el servidor broker estava funcionant i testejat. Es va implementar la llibreria PUBSubClient a Arduino, per tal de poder publicar i subscriure's al servidor MQTT.

Finalment, es va crear una pàgina web mitjançant HTML, CSS i PHP, a més d'implementar la llibreria Paho MQTT de JavaScript, per tal que la pàgina web sigues un client MQTT més, però mitjançant una comunicació per Websockets.

El sistema de identificació de l'usuari a la Web, es va dur a terme amb una base de dades MySQL, on la web feia una petició de comprovació del usuari introduït i després amb la verificació de la contrasenya, si era tot correcte el sistema donava accés a l'usuari.

### 6.2. Possibles millores

Una vegada s'estava immers en el desenvolupament del projecte, s'han anat trobant diverses millores del sistema. Algunes d'aquestes millores s'han pogut implementar però unes d'altres no s'han pogut per diferents motius principalment per manca de temps, per tant es detallen com a possibles millores en cas de voler desenvolupar el projecte per treure'l comercialment.

En relació al hardware d'Arduino, s'hauria de crear una placa integrada per miniaturitzar encara més el disseny, a més de aïllar el sensor de temperatura, ja que la placa ESP32 es calenta força i fa que el sensor no mesuri correctament la temperatura ambient. S'ha millorat, instal·lant un dissipador al ESP32, però seria recomanable instal·lar el sensor en una altra part del sistema. La pantalla, en un principi es va decidir per una de tinta electrònica, però seria millor una de LCD, per motius de il·luminació si la habitació està sense llum, encara que no es un problema important. Per últim, si es vol comercialitzar seria la creació de una carcassa de plàstic adaptada al sistema, actualment, com es tracta d'un prototip no s'ha creat cap tipus de carcassa.

Configuració de la adreça IP manualment, ara només es pot configurar mitjançant DHCP.

Configuració de la connectivitat mitjançant Bluetooth, per a dispositius mòbils.

En relació al servidor – client MQTT les possibles millores serien la creació de ACLs en el servidor MQTT i la configuració de la encriptació SSL/TLS per els clients MQTT connectats, tant l'Arduino com la pàgina web.

Vincular una base de dades al servidor MQTT, per tal de tenir històrics, per després poder analitzar les dades.

Creació d'una aplicació per android i IOS, per tal de no tenir que accedir-hi mitjançant una pàgina Web.

### **6.3. Valoració personal**

En trets generals, la valoració personal es molt positiva. La realització d'un projecte d'aquest tipus com a projecte de final de grau ja sigut molt gratificant, ja que a priori, mai havia treballat amb cap dels elements del projecte. Es la meua primera interacció amb Arduino, MQTT inclús HTML, CSS, JavaScript. Encara que l'esforç que he tingut que dedicar ha sigut força important, degut a la falta de coneixements en aquestes àrees, considero que el projecte resultant es força bo.

Per una part, m'ha permès conèixer el potencial de hardware del tipus Arduino, el protocol MQTT, el qual ha sigut una descoberta molt interessant, ja que el meu entorn laboral relacionat amb la automatització està molt relacionat, amb l'adquisició de dades dels diferents dispositius distribuïts per les màquines, amb un sistema SCADA. Crec que el protocol MQTT es força interessant per aquest propòsit.

Per una altra part, m'ha fet interessar-me per l'entorn de desenvolupament de pàgines web, ja que mai havia tingut contacte amb aquests llenguatges, ja que estan molt allunyats del meu entorn laboral.

Per últim, he pogut conèixer les possibilitats de la domòtica, així com el mercat que l'envolta. Ja que el meu concepte d'aquest entorn era força diferent al que tinc ara. Considero que es un mercat laboral interessant, per explotar.

Per totes aquestes conclusions, estic molt satisfet d'haver escollit aquesta àrea pel treball de fi de grau, i també amb el resultat obtingut, encara que com sempre, sempre es pot millorar.

## 7. Bibliografia

- [1] Arduino.cc (2018). Getting started. [online] Available at: <https://www.arduino.cc/en/Guide/HomePage> [Accessed Feb. 2018].
- [2] Randomnerdtutorials.com (2018). Installing the ESP32 board in Arduino IDE. [online] Available at: <http://randomnerdtutorials.com/installing-the-esp32-board-in-arduino-ide-windows-instructions/> [Accessed Feb. 2018].
- [3] Esp32.com (2018). ESP32 Forum. [online]. Available at: <https://esp32.com> [Accessed Mar. 2018].
- [4] Espressif Systems (2018). ESP32 Technical Reference Manual [online] Available at: [https://www.espressif.com/sites/default/files/documentation/esp32\\_technical\\_reference\\_manual\\_en.pdf](https://www.espressif.com/sites/default/files/documentation/esp32_technical_reference_manual_en.pdf) [Accessed Mar. 2018].
- [5] Sparkfun.com (2018). DS18B20 datasheet. [online] Available at: <https://cdn.sparkfun.com/datasheets/Sensors/Temp/DS18B20.pdf> [Accessed Feb. 2018].
- [6] Cloudmqtt.com (2018). CloudMQTT. [online] Available at: <https://www.cloudmqtt.com> [Accessed Feb. 2018].
- [7] Waveshare.com (2018). Working principle. [online] Available at: [https://www.waveshare.com/wiki/1.54inch\\_e-Paper\\_Module#Working\\_principle](https://www.waveshare.com/wiki/1.54inch_e-Paper_Module#Working_principle) [Accessed Mar. 2018].
- [8] Programarfácil.com (2018). Reloj con Arduino, cómo controlar los tiempos con un RTC. [online] Available at: <https://programarfácil.com/blog/arduino-blog/reloj-con-arduino-rtc/> [Accessed Mar. 2018].

- [9]** Balsamiq.com (2018). Introduction to Balsamiq Cloud. [online] Available at: <https://docs.balsamiq.com/cloud/intro/> [Accessed Mar. 2018].
- [10]** Mqtt.org (2018) MQTT- News. [online] Available at: <http://MQTT.org/> [Accessed Apr. 2018].
- [11]** Wikipedia.org (2018). MQTT- Informació general. [online] Available at: <https://en.wikipedia.org/wiki/MQTT> [Accessed Apr. 2018].
- [12]** Mosquitto.org (2018). Eclipse Mosquitto. An open source MQTT broker. [online] Available at: <https://Mosquitto.org/> [Accessed Apr. 2018].
- [13]** Eclipse.org (2018). Eclipse paho project. [online] Available at: <https://www.eclipse.org/paho/> [Accessed May 2018].
- [14]** w3.org (2018). HTML i CSS. [online] Available at: <https://www.w3.org/standards/webdesign/htmlcss> [Accessed Apr. 2018].
- [15]** w3schools.com (2018). CSS tutorial. [online] Available at: <https://www.w3schools.com/Css/> [Accessed Apr. 2018].
- [16]** w3schools.com (2018). Html tutorial. [online] Available at: <https://www.w3schools.com/Html/> [Accessed Apr. 2018].
- [17]** Aws.amazon.com (2018). Amazon AWS. [online] Available at: <https://aws.amazon.com/es/> [Accessed Apr. 2018].
- [18]** Fritzing.com (2108). Fritzing download. [online] Available at: <http://fritzing.org/home/> [Accessed Mar. 2018].
- [19]** Lucidchart.com (2018). Lucidchart [online]. Available at: <https://www.lucidchart.com> [Accessed Jun. 2018].

## 8. Annexos

### 8.1. Preparació IDE d'Arduino

El software per la programació del ESP32, es l'IDE d'arduino, al no ser un microcontrolador d'Arduino, s'ha de preparar L'IDE per poder compilar el programa i per poder comunicar amb el dispositiu.

El procediment per fer-ho està detallat al següent link:

<http://randomnerdtutorials.com/installing-the-esp32-board-in-arduino-ide-windows-instructions/>

Instal·lem Python al nostre ordinador:

```
aesco@Portatil_Asus MINGW32 ~
$ python -m pip install --upgrade pip && \
> pip install pyserial
Requirement already up-to-date: pip in c:\python27\lib\site-packages
Requirement already satisfied: pyserial in c:\python27\lib\site-packages
```

Una vegada Python està instal·lat, creem un directori per descarregar tot el paquet del ESP32 necessari per la seva posterior instal·lació.

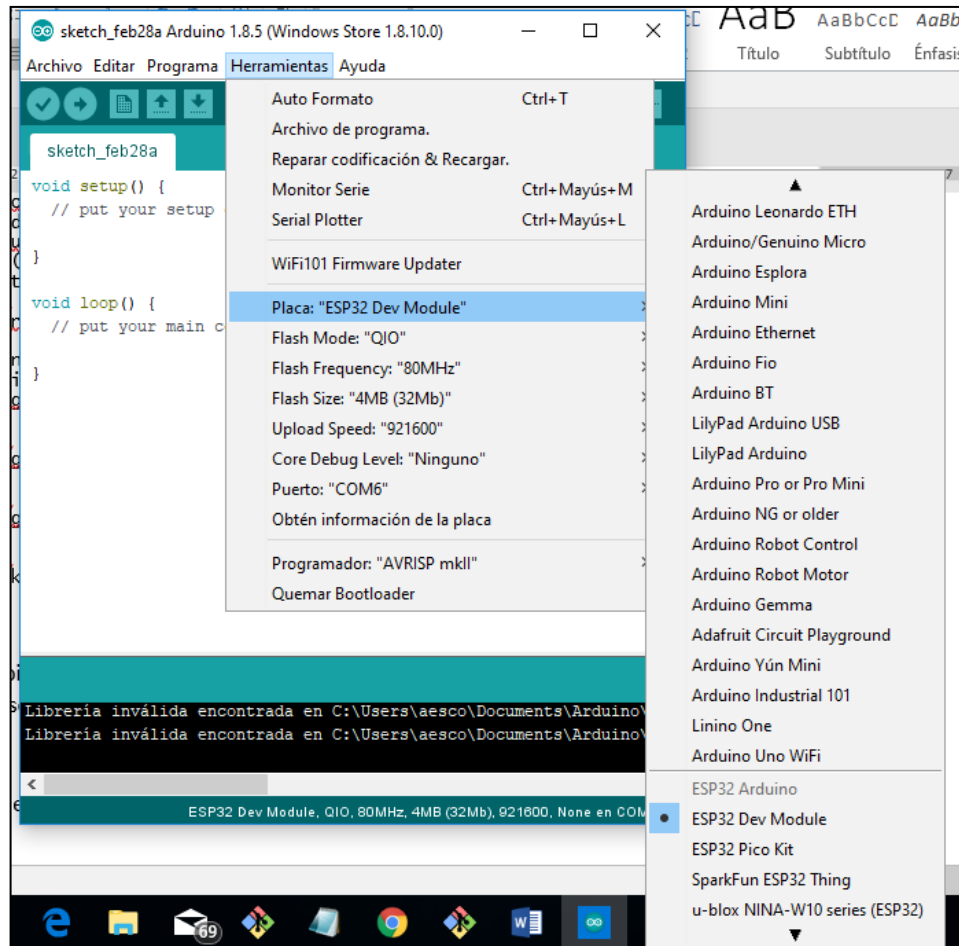
```
aesco@Portatil_Asus MINGW32 ~
$ mkdir -p ~/Documents/Arduino/hardware/espressif && \
> git clone https://github.com/espressif/arduino-esp32.git esp32 && \
> cd esp32/tools/ && \
> python get.py
Cloning into 'esp32'...
remote: Counting objects: 7155, done.
remote: Total 7155 (delta 0), reused 0 (delta 0), pack-reused 7154
Receiving objects: 100% (7155/7155), 104.88 MiB | 419.00 KiB/s, done.
Resolving deltas: 100% (4099/4099), done.
Checking out files: 100% (1287/1287), done.
Traceback (most recent call last):
  File "get.py", line 25, in <module>
    import requests
ImportError: No module named requests
$ get.exe
System: Windows, Info: windows-10-10.0.15063
Platform: i686-mingw32
Downloading xtensa-esp32-elf-win32-1.22.0-80-g6c4433a-5.2.0.zip
Done
Extracting xtensa-esp32-elf-win32-1.22.0-80-g6c4433a-5.2.0.zip
Downloading esptool-4dab24e-windows.zip
Done
Extracting esptool-4dab24e-windows.zip
Downloading mkspiffs-0.2.1-windows.zip
Done
Extracting mkspiffs-0.2.1-windows.zip
Renaming mkspiffs-0.2.1-windows/ to mkspiffs
Done
```

Després copiem la carpeta:

C:\Users\aesco\esp32 a C:\Users\aesco\Documents\Arduino\hardware\espressif

I es reinicia l'IDE d'arduino.

Una vegada es reinicia el software, en el menú herramientas, ja apareix la placa ESP32, i es pot comunicar sense problemes.



## 8.2. Llibreries utilitzades

A continuació es detallen les llibreries utilitzades en el programa del ESP32.

Les llibreries per el display de tinta electrònica son les següents, es poden baixar del següent link (<https://github.com/ZinggJM/GxEPD>)

```
<GxEPD.h>
<GxGDEP015OC1/GxGDEP015OC1.cpp>
<GxIO/GxIO_SPI/GxIO_SPI.cpp>
<GxIO/GxIO.cpp>
```

La següent llibreria indicada, encara que està relacionada amb el funcionament de la pantalla, es on es guarden les pantalles dissenyades, mitjançant el software Image2Lcd.exe, descrit en seccions anteriors.

```
<BitmapGraphics.h>
```

Les següents dos llibreries son per la comunicació i presentació de la temperatura amb el sensor ds18b20, totes dues estan per defecte al IDE d'Arduino, però es poden baixar del següent link (<https://github.com/PaulStoffregen/OneWire>).

```
<OneWire.h>
<DallasTemperature.h>
```

Les següents llibreries son ja incloses en el IDE d'Arduino encara, que es detalla el link on es poden descarregar.

```
<WiFi.h>           https://github.com/arduino/Arduino/tree/master/libraries/WiFi
<NTPtimeESP.h>    https://github.com/Sensorslot/NTPtimeESP
<Wire.h>          https://github.com/esp8266/Arduino/tree/master/libraries/Wire
<RTCLib.h>        https://github.com/adafruit/RTCLib
<EEPROM.h>        https://github.com/esp8266/Arduino/tree/master/libraries/EEPROM
```

Aquestes dues llibreries son per la gestió d'interrupccions i del watchdog, es poden descarregar del següent link (<https://esp-idf.readthedocs.io/en/v1.0/api/wdts.html>).

```
<esp_task_wdt.h>
<esp_int_wdt.h>
```

Aquesta darrera llibreria es necessària per la publicació i subscripció de dades en el protocol MQTT.

```
<PubSubClient.h> https://github.com/knolleary/pubsubclient/releases/tag/v2.6
```



### 8.3. Esquemes elèctrics

