

Lluís Fabregat Riesco

Disseny i implementació d'una casa domòtica controlada mitjançant una pàgina web

TREBALL DE FI DE GRAU

Dirigit per: David Carpintero Pleguezuelos

Grau d'Enginyeria Electrònica Industrial i Automàtica



**ESCOLA TÈCNICA SUPERIOR
D'ENGINYERIA
Universitat Rovira i Virgili**



Tarragona

2024

Agraïments

Als meus pares, Xavi i Alicia, per donar-me l'oportunitat de créixer en un entorn d'amor, respecte i esforç i per ensenyar-me que la vida no és fàcil però que val la pena lluitar cada dia.

Espero en algun moment de la meua vida poder retornar una mil·lèsima part de tot allò que m'han donat i fet per mi.

Índex

1. MEMÒRIA.....	7
1.1. Introducció.....	8
1.2. Motivació i justificació.....	8
1.3. Abast.....	9
1.4. Objectius generals.....	10
1.5. Terminologia i simbologia.....	11
1.6. Domòtica	12
1.6.1. Definició	12
1.6.2. Àrees de la domòtica	13
1.6.2.1. Automatització de la llar.....	13
1.6.2.2. Seguretat	16
1.6.2.3. Gestió energètica	17
1.6.2.4. Benestar	18
1.6.3. Nivells de domotització	19
1.6.4. Normativa i certificació	20
1.7. Disseny de la casa domòtica.....	22
1.7.1. Descripció general del sistema	22
1.7.2. Plànols i imatges.....	24
1.7.3. Microcontrolador ESP32	32
1.7.4. Hardware	33
1.7.4.1. Il·luminació LED.....	34
1.7.4.2. Servomotor	36
1.7.4.3. Sensor DHT22	38
1.7.4.4. Sensor d'infrarojos passiu	40
1.7.4.5. Sensor detector de flama.....	41
1.7.4.6. Brunzidor actiu	43
1.7.4.7. Bomba hidràulica.....	45
1.7.4.8. Fotoresistor	47
1.7.4.9. Taula resum.....	48
1.7.5. Software.....	50
1.7.5.1. Entorn de programació	50
1.7.5.1.1. Arduino IDE	50
1.7.5.1.2. Visual Studio Code.....	51
1.7.5.2. Firebase Realtime Database	52
1.7.5.3. Codi per a la pàgina web	53
1.7.5.3.1. HTML.....	54

1.7.5.3.2.	CSS	71
1.7.5.3.3.	JavaScript	72
1.7.5.4.	Codi per al microcontrolador ESP32.....	90
1.7.6.	Cost de construcció del prototip	114
1.7.7.	Aplicació en un entorn real.....	115
1.7.8.	Extensions i millores del projecte.....	118
1.8.	Conclusions	119
2.	ANNEXOS	122
2.1.	Taula de càlcul del nivell domòtic	123
2.2.	Codi HTML	126
2.3.	Codi d'estil CSS del formulari d'inici de sessió.....	138
2.4.	Codi d'estil CSS de la barra lateral de navegació	140
2.5.	Codi d'estil CSS del cos principal de la pàgina	143
2.6.	Codi JavaScript d'autenticació	149
2.7.	Codi JavaScript generador de gràfics	150
2.8.	Codi JavaScript principal	153
2.9.	Codi Arduino per al microcontrolador de sensors	171
2.10.	Codi Arduino per al microcontrolador de motors.....	177
2.11.	Codi Arduino per al microcontrolador d'il·luminació.....	186
3.	PLÀNOLS	192
3.1.	Plànol de la parcel·la (tancaments exteriors, jardí i habitatge).....	193
3.2.	Plànol acotat de la parcel·la.....	194
3.3.	Detall acotat de l'habitatge	195
3.4.	Emplaçament i connexió dels dispositius domòtics	196
4.	AMIDAMENTS	197
5.	PRESSUPOST.....	199
5.1.	Pressupost instal·lació elèctrica.....	200
5.2.	Resum del pressupost	206
	Referències	207

Índex de figures

Figura 1. Visió general de la domòtica	13
Figura 2. Exemples de protocols de comunicació	15
Figura 3. Exemple de la taula de càlcul del nivell de domotització	19
Figura 4. Model 3D: porta exterior i jardí davanter	25
Figura 5. Model 3D: piscina i jardí lateral	25
Figura 6. Model 3D: garatge i jardí lateral	26
Figura 7. Model 3D: façana davantera	26
Figura 8. Model 3D: façana lateral dreta.....	26
Figura 9. Model 3D: façana lateral esquerra	27
Figura 10. Model 3D: façana darrera	27
Figura 11. Model 3D: vista en planta de l'interior	28
Figura 12. Model 3D: rebedor	28
Figura 13. Model 3D: menjador (vista 1).....	29
Figura 14. Model 3D: menjador (vista 2).....	29
Figura 15. Model 3D: bany	29
Figura 16. Model 3D: habitació.....	30
Figura 17. Model 3D: cuina (vista 1)	30
Figura 18. Model 3D: cuina (vista 2)	31
Figura 19. Model 3D: sala de neteja i emmagatzemament	31
Figura 20. Microcontrolador ESP-WROOM-32	32
Figura 21. Distribució de pins del ESP-WROOM-32	33
Figura 22. Tira LED RGB WS2812B.....	34
Figura 23. Pins tira LED NeoPixel.....	35
Figura 24. Servomotor.....	36
Figura 25. Pins d'un servomotor	37
Figura 26. Polsos de control de la posició d'un servomotor	37
Figura 27. Polsos de control de la rotació d'un servomotor continu.....	37
Figura 28. Sensor DHT22	38
Figura 29. Sensor PIR	40
Figura 30. Pins del sensor PIR	40
Figura 31. Sensor detector de flama	42
Figura 32. Pins d'un sensor de flama	42

Figura 33. Brunzidor actiu.....	44
Figura 34. Pins d'un brunzidor.....	44
Figura 35. Bomba hidràulica.....	45
Figura 36. Esquema de connexió del conjunt bomba-relé.....	46
Figura 37. Fotoresistor.....	47
Figura 38. Esquema del fotoresistor.....	47
Figura 39. Icona Arduino IDE.....	50
Figura 40. Icona Visual Studio Code.....	51
Figura 41. Icona Firebase Realtime Database.....	52
Figura 42. Diagrama dels diferents codis que componen la pàgina web.....	53
Figura 43. Diagrama estructural del codi HTML.....	54
Figura 44. Formulari d'inici de sessió.....	56
Figura 45. Seccions de la pàgina principal de la web.....	58
Figura 46. Informació d'usuari i tancament de sessió.....	59
Figura 47. Quadre de control de l'alarma (connectada i informant d'intrusió).....	61
Figura 48. Quadre de visualització de la mesura del clima.....	62
Figura 49. Quadre de detecció d'incendis.....	62
Figura 50. Quadre de control dels accessos a l'habitatge.....	63
Figura 51. Quadre de control de la temperatura.....	64
Figura 52. Quadres de control de la il·luminació manual (esquerra) i automàtica (dreta). 65	
Figura 53. Quadre de control de la depuradora.....	67
Figura 54. Quadre de control de la persiana.....	68
Figura 55. Quadre amb el gràfic de temperatura.....	69
Figura 56. Diagrama estructural del codi JavaScript.....	72

Índex de taules

Taula 1. Requisits dels nivells de domotització.....	20
Taula 2. Criteris d'elecció de la il·luminació LED RGB WS2812B.....	35
Taula 3. Puntuació criteris d'elecció de la il·luminació amb bombetes LED.....	36
Taula 4. Puntuació criteris d'elecció del servomotor continu.....	38
Taula 5. Puntuació criteris d'elecció del motor pas a pas.....	38
Taula 6. Puntuació criteris d'elecció del sensor de temperatura i humitat.....	39
Taula 7. Puntuació criteris d'elecció del sensor PIR.....	41
Taula 8. Puntuació criteris d'elecció del sensor detector de flama.....	43
Taula 9. Puntuació criteris d'elecció del sensor detector de fum.....	43
Taula 10. Puntuació criteris d'elecció del bronzidor actiu.....	44
Taula 11. Puntuació criteris d'elecció d'un bronzidor passiu.....	45
Taula 12. Puntuació criteris d'elecció de la bomba hidràulica.....	46
Taula 13. Puntuació criteris d'elecció del fotoresistor.....	48
Taula 14. Puntuació criteris d'elecció d'un sensor de llum crepuscular.....	48
Taula 15. Taula resum dels dispositius de hardware.....	49
Taula 16. Costos associats als components i dispositius utilitzats pel prototip.....	114
Taula 17. Costos associats als materials utilitzats pel prototip.....	115

1. MEMÒRIA

1.1. Introducció

En l'era actual, la tecnologia ha envaït gairebé tots els aspectes de la nostra vida quotidiana, transformant la manera com interactuem amb el nostre entorn i els objectes que ens envolten. Un dels àmbits en què aquesta transformació és més significativa és en el camp de la domòtica, una disciplina que busca integrar la tecnologia en els sistemes de control i gestió dels nostres habitatges per millorar la comoditat, la seguretat i l'eficiència energètica.

La domòtica ofereix la possibilitat de crear entorns habitables intel·ligents, en els quals els dispositius electrònics es comuniquen entre ells i es controlen de manera centralitzada per adaptar-se a les necessitats dels usuaris. Aquest fet obre la porta a una àmplia gamma de possibilitats, des de la programació d'il·luminació i temperatura fins al control remot de dispositius com ara persianes, electrodomèstics i sistemes de seguretat.

En aquest context, el present treball aborda el disseny i la implementació d'una casa automàtica controlada mitjançant un servidor web al qual l'usuari hi podrà accedir des de d'allà on es trobi. Aquesta solució combina els avenços en tecnologia de la informació i les comunicacions amb els principis de la domòtica per crear un entorn residencial totalment connectat i personalitzable.

A través d'aquesta investigació, explorarem les diferents tecnologies i protocols que permeten la comunicació entre dispositius, així com els principals components d'un sistema de domòtica i les seves funcionalitats. Analitzarem també els reptes i les oportunitats associades a la implementació d'un sistema de control residencial basat en un servidor web, incloent aspectes com la seguretat, la fiabilitat i la interoperabilitat dels dispositius.

Finalment, mitjançant el desenvolupament i la prova d'un prototip funcional, es demostrarà la viabilitat i els avantatges d'aquesta solució en el context de la vida quotidiana moderna.

1.2. Motivació i justificació

La motivació per emprendre aquest treball és alimentada per una passió compartida entre el meu pare i jo per les noves tecnologies, així com per un desig de millorar la qualitat de vida de la meva família a través de la innovació i la integració intel·ligent dels sistemes domèstics. Des de sempre, he mostrat un fort interès per les últimes tendències tecnològiques i les seves aplicacions pràctiques en la vida quotidiana. Aquesta curiositat constant i la voluntat de mantenir-me a l'avantguarda de la innovació han servit com a impulsors per explorar les possibilitats que ofereix la domòtica.

A més, la perspectiva d'implementar un sistema de casa automàtica controlada mitjançant una pàgina web a la meua llar en un futur m'ha inspirat a investigar i comprendre més a fons els fonaments d'aquesta tecnologia. L'oportunitat de crear un entorn residencial que no només sigui més eficient des del punt de vista energètic sinó també més convenient, segur i personalitzat, és un objectiu que em motiva profundament.

A través d'aquest treball, es busca donar una resposta pràctica a aquesta motivació, explorant la viabilitat i els beneficis d'una casa automàtica. La justificació d'aquesta investigació radica en la creixent importància de la domòtica com a solució per abordar els reptes moderns relacionats amb la gestió eficient dels recursos, la comoditat dels usuaris i la seguretat dels habitatges.

A través de la investigació i la implementació pràctica d'aquest sistema, es busca demostrar que la domòtica no és només una opció accessible i viable, sinó també una solució altament beneficiosa per a la vida quotidiana moderna.

1.3. Abast

El present treball té com a objectiu principal el disseny i la implementació d'una casa domòtica controlada mitjançant una pàgina web, amb capacitat d'accés remot per part de l'usuari. Per aconseguir aquest objectiu, s'establiran diverses fases i tasques descrites a continuació:

- Es realitzarà una revisió exhaustiva de la literatura relacionada amb la domòtica, les àrees en les qual s'aplica, els dispositius d'ús domòtic, la legislació i normativa aplicada, els protocols de comunicació entre els dispositius i el servidor web i altres temes pertinents. Aquesta revisió servirà de fonament teòric per al posterior disseny i implementació del sistema.
- S'establiran els components necessaris pel disseny del sistema com per exemple els dispositius d'il·luminació, els sensors i actuadors. A més, es definirà la funcionalitat de cada un d'aquests i la ubicació que ocuparan en el futur prototip.
- Es procedirà a la programació en llenguatge C del microcontrolador ESP32 per tal de comunicar tots els dispositius amb la base de dades en temps real i el servidor. Simultàniament, per tal d'anar comprovant que la programació i sincronització dels dispositius es l'esperada, es realitzarà la programació de la web mitjançant els llenguatges HTML, CSS i JavaScript.
- Es dissenyarà un petit prototip funcional per tal d'integrar tots els dispositius domòtics programats amb l'objectiu de demostrar totes les interaccions possibles entre la interfície de la pàgina web i l'usuari amb l'ambient domèstic.

1.4. Objectius generals

A través dels objectius que es descriuen a continuació, es pretén establir les bases necessàries per al desenvolupament d'un sistema eficient, innovador i adaptable a les necessitats dels usuaris, proporcionant una experiència d'ús intuïtiva i satisfactòria.

1. Integració del microcontrolador ESP32:

- Utilització del microcontrolador ESP32 com a base per a la gestió eficient dels dispositius domòtics, assegurant la compatibilitat i la connectivitat necessàries per al correcte funcionament.

2. Programació en l'entorn Arduino IDE:

- Desenvolupament del codi del microcontrolador utilitzant el llenguatge C a través de l'entorn Arduino IDE, garantint un accés fiable a les llibreries existents.

3. Creació d'una interfície web interactiva:

- Desenvolupament d'una pàgina web mitjançant llenguatges com HTML, CSS i JavaScript, accessible des de qualsevol dispositiu amb connexió a Internet, mitjançant la qual l'usuari pugui controlar de manera intuïtiva els diferents dispositius domòtics, com ara la il·luminació i les portes entre altres funcionalitats.

4. Disseny i personalització de la interfície web:

- Creació d'una interfície web atractiva i fàcil d'utilitzar mitjançant la personalització dels diferents estils, proporcionant una experiència d'usuari satisfactòria i adaptada a les necessitats.

5. Utilització de Firebase Realtime Database:

- Implementació de Firebase Realtime Database de Google per gestionar les dades dels dispositius de manera eficient en temps real, garantint una sincronització ràpida i fiable entre la pàgina web i els dispositius físics.

6. Incorporació de varietat en els dispositius:

- Utilització de gran diversitat d'elements per tal d'abordar totes les àrees possibles en les quals la domòtica té una utilitat en un ambient domèstic.

7. Construcció d'un prototip funcional de la casa:

- Implementació d'un petit prototip de la casa domòtica per demostrar el correcte funcionament del sistema i totes les interaccions possibles que l'usuari pot realitzar amb la pàgina web dissenyada.

A través d'aquests objectius, s'assegura una aproximació completa i detallada al desenvolupament del projecte, abordant tant els aspectes tècnics com les necessitats i expectatives dels usuaris.

1.5. Terminologia i simbologia

- **Arduino IDE:** entorn de desenvolupament integrat (IDE) utilitzat per programar i carregar codi a les plaques Arduino i altres microcontroladors compatibles.
- **CSS:** en anglès, *Cascading Style Sheets*. És un llenguatge de fulls d'estil utilitzat per controlar l'aparença i l'estil de les pàgines web escrites amb HTML.
- **Domòtica:** conjunt de sistemes capaços d'automatitzar les diferents instal·lacions d'un habitatge o edificació de qualsevol tipus, aportant serveis de gestió energètica, seguretat, benestar...
- **Firestore Realtime Database:** servei de base de dades en temps real proporcionat per Google.
- **GND:** en anglès, *Ground*. Fa referència al punt de terra d'un circuit, generalment definit com a 0 V. És el conductor que actua com a retorn per al corrent elèctric d'un circuit, proporcionant una referència comuna per a tots els components.
- **Hardware:** part física i tangible d'un sistema informàtic format per un conjunt de components electrònics i mecànics connectats entre si.
- **HTML:** en anglès, *Hyper Text Markup Language*. És el llenguatge de marcatge estàndard per crear pàgines web.
- **IoT:** en anglès, *Internet of Things*. Fa referència a la interconnexió de dispositius físics a través d'Internet, permetent la recollida i intercanvi de dades per automatitzar processos.
- **JavaScript:** llenguatge de programació utilitzat en el desenvolupament web per afegir interactivitat i dinamisme.
- **PWM:** en anglès, *Pulse Width Modulation*. La modulació per amplada de pols és una tècnica utilitzada en electrònica per controlar la potència d'un senyal digital mitjançant la variació del temps de cicle.
- **Script:** en informàtica, seqüència de comandaments escrits en un llenguatge específic que s'utilitza per automatitzar tasques o realitzar operacions en un sistema informàtic.
- **Sistema embedded:** o embegut, és un sistema de computació basat en un microcontrolador dissenyat per realitzar una o algunes tasques dedicades en temps real.
- **Software:** conjunt de programes, instruccions i dades que permeten a un sistema informàtic realitzar tasques específiques.
- **VCC:** abreviació que representa la tensió d'alimentació positiva d'un dispositiu electrònic.

1.6. Domòtica

La domòtica, una disciplina que combina la tecnologia de la informació, la comunicació i l'automatització, ha emergit com una solució innovadora per millorar la gestió i el control dels sistemes residencials.

En aquest capítol, s'exploren àmpliament els fonaments de la domòtica començant per la seva definició. A continuació, s'analitza com aquesta ciència s'ha convertit en un paradigma que busca introduir la tecnologia en les llars per millorar el benestar, la seguretat i l'eficiència energètica entre altres. Seguidament, es discuteixen els diferents nivells de domotització existents, que proporcionen una visió escalonada de la integració de la tecnologia en els habitatges, des de solucions bàsiques fins a sistemes totalment automatitzats i intel·ligents. Finalment, s'aborda la normativa i la certificació en aquest camp, destacant la importància de la conformitat amb els estàndards i les regulacions per garantir la seguretat dels sistemes.

1.6.1. Definició

Etimològicament, el significat de domòtica prové del llatí i del grec. Es creu que és el resultat de la composició formada per les paraules *domus*, que significa casa en llatí, i *autònom*, que significa “que es governa a si mateix” en grec.

Segons el *Diccionari Essencial de la Llengua Catalana* de l'Institut d'Estudis Catalans [1], la domòtica és la disciplina que s'ocupa de la concepció i de l'aplicació d'automatismes en les instal·lacions dels habitatges.

En termes generals, la domòtica es el conjunt de tecnologies aplicades al control i l'automatització intel·ligent d'un habitatge amb l'objectiu de realitzar una gestió eficient de l'ús d'energia i comunicacions i d'aportar seguretat, benestar i confort als residents. Aquests sistemes poden estar integrats mitjançant xarxes interiors o exteriors de comunicació, cablejades o *wireless* (sense fil) i el control de les quals gaudeix de certa ubicuïtat ja que permet una comunicació directa entre usuari i sistema tant des de dins com de fora de la llar.

Dintre de l'àmplia gamma de sistemes domòtics i tecnologies que es poden introduir cal destacar els següents:

- Sistemes d'il·luminació interiors i exteriors.
- Sistemes de control automàtic de persianes, cortines i tendals.
- Sistemes de regulació i control de la climatització.
- Sistemes de control d'encesa i apagada d'electrodomèstics.
- Sistemes de seguretat.
- Sistemes de control de reg.

1.6.2. Àrees de la domòtica

En aquest apartat, s'exploren les quatre principals àrees de la domòtica que abasten un ampli ventall d'aplicacions i funcionalitats destinades a millorar la qualitat de vida dels usuaris i l'eficiència dels sistemes residencials. A través de l'automatització de dispositius i processos, la seguretat, l'eficiència energètica i el benestar, la domòtica busca transformar els habitatges en entorns intel·ligents i adaptatius.

Mitjançant l'anàlisi d'aquestes quatre àrees clau, es posa de manifest i demostra el gran potencial que té aquesta tecnologia per transformar els espais residencials en llocs més segurs, eficients, còmodes i saludables.



Figura 1. Visió general de la domòtica. Font: [2]

1.6.2.1. Automatització de la llar

Aquesta vessant de la domòtica és, amb gran certesa, l'àrea més important ja que implica la integració de dispositius electrònics, electrodomèstics i sistemes de control per automatitzar i gestionar els diferents processos i funcions que fan de l'habitatge un espai més còmode. A més, gràcies als grans avenços tecnològics que acompanyen l'última dècada, aquesta àrea ofereix una gamma molt àmplia de possibilitats per millorar la comoditat, eficiència i seguretat dels residents.

L'automatització permet programar, controlar i monitoritzar l'estat dels diferents dispositius i processos de manera remota i centralitzada a través d'una sèrie de protocols de comunicació. En funció de diferents factors com la distància, la velocitat a la qual es vol transmetre o la interoperabilitat entre dispositius, la connexió es realitza amb una tecnologia o una altra ja que cada una disposa de una sèrie de característiques i avantatges determinats.

Degut a la gran expansió de la indústria en els últims anys i la gran competència dintre del sector domòtic, les empreses es veuen forçades a desenvolupar i comercialitzar els seus propis productes que es comuniquen amb el seu propi *software*.

Així doncs, fent referència als protocols de comunicació es distingeix entre els estàndards propietaris o tancats i els estàndards oberts:

- **Protocols tancats:** són aquells en què el fabricant es reserva el dret a la modificació i comercialització de manera que tan sols ell pot realitzar actualitzacions i millores. Aquest fet implica de forma directa que els dispositius domòtics a incorporar siguin del mateix fabricant ja que és necessari que “parlin” el mateix idioma.

Aquesta metodologia protegeix els drets del fabricant, però limita l'aparició de millores i evolucions als sistemes domòtics, de manera que, a mesura que els sistemes amb protocol obert es van desenvolupant, van guanyant quota de mercat respecte sistemes de protocol propietari.

Un bon exemple de protocol de comunicació tancat és Konnex (KNX). Actualment, és un dels més implementats a Europa i compta amb el gran avantatge de que pel simple fet que un sistema tingui el segell KNX, implica que compleix amb les regulacions i normatives domòtiques a nivell europeu i mundial.

- **Protocols oberts:** són aquells que estan disponibles públicament i que són definits i desenvolupats entre diverses companyies amb l'objectiu d'unificar criteris. En aquest tipus de sistemes no existeixen patents en quant al protocol, de tal forma que qualsevol fabricant pot desenvolupar productes que portin implícit aquest estàndard. Així mateix, els dispositius de diferents fabricants es poden comunicar entre si i es poden controlar a través d'una plataforma, la qual cosa afavoreix la interoperabilitat i la compatibilitat.

Alguns dels exemples més coneguts i utilitzats en quant a estàndards oberts són els següents:

- **Wi-Fi:** abreviatura de l'anglès, *wireless fidelity*, és un dels protocols més comuns per a la connexió de dispositius en xarxes domèstiques ja que ofereix una connectivitat sense fil robusta i d'alta velocitat. Els dispositius domèstics intel·ligents que utilitzen Wi-Fi poden connectar-se directament a l'enrutador i a Internet, la qual cosa permet realitzar un control remot a través d'aplicacions mòbils o pàgines web.
- **Bluetooth:** és un protocol sense fil àmpliament utilitzat en domòtica per connectar dispositius de poc abast com per exemple altaveus o panys intel·ligents. Permetent la comunicació directa sense la necessitat d'una xarxa Wi-Fi, és recomanable per sistemes domòtics on la distància no és un factor crític.
- **Zigbee:** és un estàndard de comunicació sense fil dissenyat específicament per aplicacions domòtiques i IoT. Utilitza una xarxa de malla per connectar múltiples dispositius entre ells amb un coordinador central. Zigbee és conegut pel baix consum d'energia i la capacitat per crear xarxes ampliables que poden cobrir àrees grans.

- **X10:** és un dels primers protocols utilitzats en la domòtica i utilitza la infraestructura elèctrica existent en l'habitatge per transmetre senyals de control a través de la xarxa elèctrica. D'aquesta forma, no és necessari instal·lar nou cablejat i els costos de la instal·lació es redueixen significativament. Tot i que no és tant modern com els anteriors, encara s'utilitza en algunes aplicacions domòtiques.



Figura 2. Exemples de protocols de comunicació

Un altre factor clau en l'automatització de la llar és l'aparició de l'IoT ja que permet la interconnexió de dispositius i objectes a través d'Internet, permetent així que recopilin, intercanvien i actuen sobre les dades rebudes de forma remota. En el context de la domòtica, l'IoT facilita la creació d'entorns residencials intel·ligents al permetre que els dispositius es comuniquin conjunta i centralitzadament. Aquest mètode, permet als usuaris controlar i monitoritzar les seves llars des de qualsevol lloc mitjançant l'ús d'aplicacions o pàgines web.

Per acabar, es mostren els exemples més comuns a l'hora d'automatitzar processos en una llar:

- **Control de la il·luminació:** programar l'encesa i l'apagada de determinades bombetes en funció de l'hora del dia, de la quantitat de llum que entra de l'exterior o de la presència o no de persones.
- **Control de la temperatura:** automatitzar l'encesa i l'apagada de la calefacció o aire condicionat en funció de la temperatura i humitat de l'habitable per tal de crear un ambient tèrmic estable i confortable.
- **Control de persianes i tendals:** igualment que amb la il·luminació, programar la baixada o pujada de determinades persianes en funció de l'hora del dia o automatitzar els tendals del jardí per tal que s'obrin o es tanquin en determinades circumstàncies climatològiques.
- **Control de reg:** automatitzar l'encesa o l'apagada dels dispositius de reg en funció de l'hora, de l'època de l'any o de la humitat de la terra per mantenir el jardí en bones condicions.

- **Control de l'alarma:** programar l'activació o desactivació de l'alarma durant la nit respectant les rutines de son dels usuaris, és a dir, activant-la lleugerament després de l'hora que acostuma a anar a dormir i desactivant-la lleugerament abans de l'hora que acostuma a llevar-se.

En resum, en el món de la domòtica, la connexió de dispositius, els protocols de comunicació i l'IoT tenen un paper crucial en permetre la creació de llars intel·ligents i l'automatització de diverses funcions i processos. Aquestes tecnologies faciliten la comunicació entre dispositius, permetent la gestió i el control remot per millorar la comoditat, la seguretat i l'eficiència energètica de la llar.

1.6.2.2. Seguretat

La seguretat en la domòtica es refereix a la prevenció de riscos físics i la protecció contra amenaces tant internes com externes. Aquesta definició implica la implementació de mesures per salvaguardar la integritat dels ocupants de la llar, així com la protecció de la propietat i els actius de la llar contra robatoris, incendis, inundacions i altres perills.

Inicialment, cal diferenciar entre els diferents tipus de seguretat ja que depenent de si el sistema és de protecció interior o exterior, el tipus de dispositiu a incorporar varia.

- **Seguretat interior:** fa referència a tots els dispositius que proporcionen una protecció als béns i actius de l'interior mitjançant actuacions immediates en cas de perill o intrusió. Entre aquests es troben càmeres de vigilància, sensors de moviment, panys intel·ligents, simuladors de presència, detectors de fum, sensors de monòxid de carboni i de gasos tòxics o sensors de nivell d'aigua.
- **Seguretat exterior:** fa referència a tots els dispositius que tenen com a objectiu detectar moviments de presències desconegudes en les zones properes a la llar. Entre aquests es troben càmeres de vigilància, alarmes, sensors de moviment i soroll, sistemes de barreres de raigs infrarojos, contactes magnètics per finestres, portes i panys...

Un dels principals avantatges és que mitjançant l'automatització i centralització de tots els dispositius instal·lats per millorar la seguretat de la llar, l'usuari pot monitoritzar en temps real l'estat del seu domicili a través d'una aplicació o interfície web. Per exemple, poden rebre notificacions instantànies sobre esdeveniments sospitosos, veure càmeres de vigilància en directe i activar o desactivar sistemes de seguretat des del seu telèfon.

També cal destacar que gràcies a la incorporació d'aquests sistemes de seguretat, es poden detectar possibles perills potencials i alertar sobre ells. Un clar exemple és la detecció de fuites de gas o incendis mitjançant detectors de fum i monòxid de carboni connectats a la xarxa domòtica, permetent així una resposta més ràpida enfront el perill i reduint el risc danys superiors.

En conclusió, la seguretat a la domòtica és fonamental per prevenir riscos físics i protegir tant els ocupants de la llar com la propietat mateixa contra diverses amenaces.

1.6.2.3. Gestió energètica

La gestió energètica és un aspecte fonamental en l'àmbit de la domòtica ja que permet optimitzar el consum d'energia a la llar, contribuint així de forma directa a un significativ estalvi en les factures i reduint la petjada que deixem impresa en el medi ambient. En un context on l'eficiència energètica i la sostenibilitat cada vegada són més importants, l'aplicació de tecnologies intel·ligents en la gestió de l'energia ofereix una solució innovadora i efectiva.

Degut a l'augment dels costos associats a l'energia, el consum d'aquesta a les llars suposa una de les despeses més grans a què s'enfronten les famílies diàriament. Segons *Red Eléctrica Española* (REE) en l'Informe del Sistema Elèctric Espanyol (2020), el consum elèctric mitjà d'una llar ronda els 3900 kWh anuals [3]. Si a aquest se li afegeix el consum del gas i de l'aigua, la despesa que els usuaris d'una llar afronten anualment és important. Afortunadament, aquesta despesa es pot veure reduïda gràcies a la instal·lació de sistemes domòtics enfocats a millorar l'eficiència energètica.

El principal avantatge d'un sistema domòtic és que es pot monitoritzar el consum energètic i gestionar de forma eficaç els subministraments, ajudant en l'estalvi d'energia. Es calcula que una casa intel·ligent pot arribar a consumir entre un 25% i un 40% menys respecte un habitatge tradicional.

Una de les principals formes en què la domòtica contribueix a l'eficiència es mitjançant l'aprofitament dels recursos naturals com la llum i la radiació solar. Gràcies a aquests, per exemple es pot adaptar la potència de les bombetes a una determinada situació lumínica per aprofitar la llum del sol o desplegar els tendals automàticament quan la radiació solar és excessiva.

A continuació, es mostren les aplicacions més destacades que es poden introduir en una instal·lació domòtica per tal d'aconseguir aquesta gestió eficient dels recursos energètics:

- **Gestió energètica centralitzada:** consisteix en l'ús d'energia exclusivament en aquells moments on és necessari. Mitjançant la monitorització del consum i la interacció intel·ligent entre sensors, dispositius i temporitzadors, el sistema domòtic contribueix a l'estalvi d'energia.
- **Obertura i tancament de finestres:** mitjançant l'ús de contactes magnètics o panys intel·ligents en finestres i portes, el sistema domòtic les pot obrir o tancar en funció de la temperatura programada a l'interior d'una determinada estància de la llar i la temperatura a l'exterior. Aquestes accions permeten estalviar un consum d'energia significativ ja que es substitueix l'ús d'electrodomèstics de climatització durant determinats intervals de temps per la climatització natural dels habitacles.
- **Control automàtic de la il·luminació:** els detectors de presència i moviment ajuden a encendre la llum de forma automàtica únicament quan es necessari. Si durant un determinat interval de temps no es detecta cap moviment, la llum s'apaga. A més, l'estalvi encara pot ser més significatiu amb la implementació de sensors de lluminositat ja que d'aquesta forma es possible controlar els kW de consum i tan sols s'utilitza la intensitat necessària. S'ha demostrat que amb aquestes accions la factura de la llum es pot veure reduïda fins a un 30% [4].

- **Interacció entre persianes i climatització:** permet integrar el calor produït per la radiació solar de forma intel·ligent amb la regulació de la temperatura. En cas que durant el dia el sol es trobi en una orientació favorable durant els mesos d'hivern, les persianes s'obren i l'ús de la calefacció disminueix significativament. De forma contrària, durant els mesos d'estiu les persianes es baixen automàticament, reduint així la potència consumida per l'aire condicionat. Aquests automatismes es tradueixen en una reducció en les factures energètiques d'entre un 15 i un 25% [4].

Amb un sistema domòtic incorporat, les dades de consum d'electricitat, gas, aigua o un altre combustible estan monitoritzades contínuament. Mitjançant l'ús de bases de dades, el sistema domòtic registra i guarda les dades de funcionament i consum, de tal forma que l'usuari pot consultar l'eficiència energètica gràcies als diagrames que s'hi poden obtenir. Amb l'anàlisi d'aquests, es poden observar les tendències al llarg de l'any i canviar paràmetres de programació en cas que fos necessari per tenir una major eficiència i estalvi.

En conclusió, la gestió energètica mitjançant la domòtica es presenta com una eina de valor incalculable per promoure l'eficiència energètica, l'estalvi econòmic i la sostenibilitat ambiental a les llars modernes.

1.6.2.4. Benestar

El benestar en la domòtica no es limita a la automatització i gestió eficient dels recursos del llar. Utilitza tecnologies intel·ligents per millorar la qualitat de vida dels ocupants, creant entorns més segurs, saludables i confortables. Es calcula que, l'any 2050, aproximadament 2.000 milions de persones al món tindran més de 60 anys [5] de tal forma que la tecnologia domòtica i l'edificació intel·ligent assumirà un paper crucial en el benestar diari.

Amb solucions molt pràctiques orientades a augmentar la comoditat, l'usuari pot adaptar l'habitatge al seu gust sense haver d'esforçar-se amb exemples com els mencionats a continuació:

- **Adaptació lumínica de l'entorn:** mitjançant l'ús de bombetes intel·ligents controlades per veu o dispositius mòbils, els usuaris poden ajustar la llum segons necessitats, preferències i horaris.
- **Adaptació tèrmica de l'entorn:** Mitjançant sistemes de climatització intel·ligents o control de persianes automatitzats, s'estableixen temperatures ideals a cada moment, optimitzant l'ús energètic i garantint el confort.

En personalitzar i adaptar l'entorn de la llar a les preferències i necessitats individuals, la domòtica no només proporciona més confort, sinó que també ofereix solucions que promouen l'estalvi energètic i l'autonomia dels usuaris tal i com s'ha vist anteriorment.

1.6.3. Nivells de domotització

S'entén per nivell de domotització o nivell domòtic, el nivell assignat a una instal·lació domòtica com a resultat de la ponderació dels dispositius existents i les aplicacions domòtiques cobertes.

De forma conjunta, l'Associació Espanyola de Domòtica i Immòtica (CEDOM) i L'Associació Espanyola de Normalització i Certificació (AENOR) van dissenyar la taula de càlcul del nivell domòtic ¹ amb què s'avaluen els dispositius i les aplicacions domòtiques amb què compta una instal·lació, per tal de definir en quin grau de domotització es troben. Cada dispositiu que s'incorpora al nostre habitatge compta amb una puntuació assignada i mitjançant la suma total de les puntuacions, s'estableix el nivell de domotització corresponent a la instal·lació.

L'estructura de la taula es divideix en 4 grans blocs: aplicacions domòtiques, dispositius, condicions a complir i ponderacions. Inicialment, en la primera columna, es diferencien mitjançant l'ús de diferents colors les diverses aplicacions domòtiques a avaluar. Entre aquestes, es troben aplicacions com les alarmes d'intrusió, el control de persianes, el control d'il·luminació, de clima...

Seguidament, cada aplicació diferencia els dispositius a instal·lar en el seu àmbit. Per exemple, en l'aplicació domòtica d'alarmes tècniques, els diferents dispositius que s'hi troben són els detectors d'inundació, de concentració de gas, d'incendis, etc. Per acabar, s'hi troben dues columnes que s'entenen de forma conjunta. La primera indica el nombre de dispositius instal·lats o la condició a complir per tal d'obtenir una determinada puntuació, que s'indica en la columna adjacent.

A continuació, a mode d'exemple es pot observar un dels apartats de la taula corresponent a l'aplicació domòtica d'alarmes d'intrusió amb els diferents dispositius a avaluar i les respectives puntuacions:

Aplicación domótica	Dispositivos	Columna de referencia		Usuario
		N.º dispositivos o condición a cumplir	Puntuación	Puntuación
Alarma de intrusión	Detectores de presencia	2	1	
		1 cada 20 m ²	2	
		1 por estancia	3	
	Teclado codificado, llave electrónica o equivalente	1	1	
		No	0	
	Sirena interior	Si	1	
		No	0	
	Contactos de ventanas y/o impactos	En puntos de fácil acceso	1	
		En todas las ventanas	2	
	Sistemas de mantenimiento de alimentación en caso de fallo de suministro eléctrico	No	0	
		Si	2	
	Módulo de habla/escucha, destinado a la escucha en caso de alarma. También se admite cualquier tipo de control que permita conocer si realmente existe un intruso (cámaras web...)	No	0	
		Si	3	
Sistema conectable con central de alarmas	No	0		
	Si	3		

Figura 3. Exemple de la taula de càlcul del nivell de domotització

¹ Consultar Annex 1. Taula nivells de domotització per veure amb detall les diferents aplicacions que s'avaluen i les ponderacions obtingudes en funció dels dispositius instal·lats.

A partir de la taula completa, la ponderació total es pot distingir en tres diferents nivells:

- **Nivell 1:** són instal·lacions amb un nivell mínim de dispositius i/o aplicacions domòtiques. La suma dels pesos ponderats dels dispositius inclosos en la instal·lació ha de ser com a mínim de 13, sempre que al seu torn cobreixi al menys 3 aplicacions. És a dir, aquests 13 punts s’han d’aconseguir amb dispositius repartits entre, almenys, 3 aplicacions diferents que es distingeixen amb diferents colors a la taula.
- **Nivell 2:** són instal·lacions amb un nivell intermedi de dispositius i/o aplicacions domòtiques. En aquest cas, la suma de punts ha de ser com a mínim de 30, sempre i quant es cobreixin 3 aplicacions.
- **Nivell 3:** són instal·lacions amb un nivell alt de dispositius i/o aplicacions domòtiques. Per assolir aquest nivell, la suma de punts ha de ser de 45 com a mínim, sempre i quant es reparteixin entre 6 aplicacions.

	Nivel 1	Nivel 2	Nivel 3
Grado domotización	Mínimo	Medio	Alto
Suma mínima ponderada	13	30	45
Funcionalidades mínimas a incluir	3	3	6

Taula 1. Requisits dels nivells de domotització

Finalment, un cop s’ha calculat el nivell domòtic d’una determinada instal·lació, es pot procedir a la tramitació del certificat d’instal·lacions domòtiques, un document basat en la especificació AENOR EA0026: “Instal·lacions de sistemes domòtics en habitatges. Prescripcions generals d’instal·lació i avaluació” [6].

1.6.4. Normativa i certificació

En el context de la domòtica, la normativa i certificació juguen un paper crucial per garantir la seguretat, la interoperabilitat i la qualitat dels sistemes i dispositius domòtics. El creixement de l’ús de tecnologies intel·ligents a la llar ha generat la necessitat d’establir estàndards i regulacions que encaminin el disseny, la implementació i l’ús d’aquests sistemes, així com la certificació de productes que compleixin aquests estàndards.

Tot i que actualment encara no hi ha una llei que unifiqui tots els criteris necessaris per a les instal·lacions domòtiques, s’ha desenvolupat una norma que pretén ser referent en aquest camp.

Elaborada per un comitè d'AENOR per a la normalització i aprovada al 2013 a nivell europeu, l'EA0026 fa referència a "Instal·lacions de sistemes domòtics en habitatges. Prescripcions generals d'instal·lació i avaluació". Aquesta normativa especifica els requisits mínims que ha de complir qualsevol instal·lació domòtica, a més de contemplar la legislació vigent aplicable a aquest camp.

Cal destacar també que s'ha convertit en el document de referència de la futura legislació europea UNE-EN 50491: "Requisits generals per a sistemes electrònics en habitatges i edificis (HBES) i sistemes d'automatització i control d'edificis (BACS)" [7], que es troba en fase d'elaboració i aprovació.

A continuació, s'exposen la resta de normes tècniques aplicades a la domòtica que cal tenir en consideració:

- **EN 50090 "Home and Building Electronic Systems (HBES)" protocol Konnex** [8]

És un estàndard europeu per a comunicacions tancades entre sistemes electrònics domèstics. Aquesta normativa cobreix qualsevol combinació de dispositius electrònics vinculats entre si mitjançant una xarxa de transmissió digital, amb l'objectiu de proporcionar un control de processos automatitzats per a aplicacions domèstiques com per exemple el control del clima, de la il·luminació o de la gestió energètica.

- **EN/ISO 16484 "Building Automation and Control Systems (BACS)"** [9]

És una norma internacional que especifica les característiques del software i les funcions que s'utilitzen en els sistemes d'automatització i control d'habitatges, així com un mètode per a la documentació del disseny.

- **Normatives UNE-EN 50491 per a Sistemes Electrònics d'Habitatges i Edificis (HBES) i Sistemes d'Automatització i Control d'Edificis (BACS)** [7]

Formades per un conjunt de normatives englobades sota l'epígraf UNE-EN 50491 "Requisits generals per a HBES i BACS" i que s'han de complir a l'instal·lar aquest tipus de tecnologies. A continuació, es destaquen els capítols més rellevants:

- EN 50491-2. Condicions ambientals
- EN 50491-3. Requisits de seguretat elèctrica
- EN 50491-4-1. Requisits generals de seguretat funcional per a productes destinats a ser integrats en HBES i BACS
- EN 50491-5-1. Requisits de compatibilitat electromagnètica (CEM)
- UNE-EN 50491-6-1. Instal·lacions HBES
- UNE-EN 50491-11. Mesurament intel·ligent

En quant a la certificació d'instal·lacions a nivell nacional, es disposa de la normativa EA 2006 creada per AENOR. Aquesta permet certificar una instal·lació domòtica seguint la classificació de tres nivells anteriorment vista. Es considerarà un sistema domòtic tot aquell que assoleixi un nivell 1 que correspon a un nivell bàsic d'automatització.

Actualment a Espanya, entitats com AENOR o AFME són les principals encarregades d'oferir sistemes per avaluar i certificar el grau de domotització dels diferents dispositius d'una instal·lació. Entre aquests sistemes, el més destacat és, tal i com s'ha vist, la taula de nivells de domotització d'una instal·lació creada per la CEDOM, que permet determinar fins quin punt la gestió i automatització d'un habitatge és intel·ligent.

1.7. Disseny de la casa domòtica

1.7.1. Descripció general del sistema

El disseny del sistema domòtic proposat es fonamenta en la gestió automatitzada i centralitzada de diversos aspectes clau dins un entorn residencial. Aquest, és el resultat de la interacció conjunta entre els quatre grans blocs que el conformen: els microcontroladors, els dispositius domòtics, la pàgina web i la base de dades.

En primer lloc, s'ha escollit un microcontrolador de la família dels ESP32 com a component central del sistema, les característiques del qual s'especifiquen en el posterior apartat. Aquest actua com a intermediari entre els dispositius físics incorporats i la plataforma de control basada en una interfície web accessible tant localment com a distància.

El següent gran bloc que conforma el sistema domòtic es tracta de la pàgina web. Aquesta s'ha dissenyat per tal que l'usuari gestioni i pugui tenir un control directe dels dispositius i de l'estat de la seva residència de forma senzilla i còmoda. S'ha dissenyat per tal que sigui accessible tant de forma local com a distància, de tal forma que l'usuari pugui monitoritzar l'estat des de qualsevol localització. De forma local, el sistema incorpora una sèrie de pantalles fixes situades en diferents punts estratègics de la casa on es mostra la pàgina web i es pot realitzar la conseqüent gestió de forma còmoda i eficient.

La base de dades en temps real també adquireix un paper fonamental ja que en aquesta és on s'emmagatzemen tots els valors referents a les mesures dels sensors i l'estat de cada dispositiu. A més, la pròpia pàgina web adquireix els valors de la *database* per tal de ser mostrats en pantalla posteriorment. En el sistema domòtic dissenyat, s'utilitza la *Firebase Realtime Database* de Google per realitzar aquesta tasca d'emmagatzemament en temps real.

El darrer gran bloc que conforma el sistema domòtic és el conjunt de dispositius físics que fan de la domòtica una realitat. En el disseny realitzat, es prioritza la gestió d'aquells elements residencials que es consideren més imprescindibles i tenint en consideració en tot moment els quatre àmbits on té presència aquesta ciència: automatització, seguretat, eficiència energètica i benestar.

A continuació, es mostren detalladament totes les funcionalitats i interaccions que incorpora el sistema domòtic dissenyat:

- **Gestió de la il·luminació:** mitjançant la incorporació d'il·luminació LED, es controla la llum en els diferents espais de l'entorn residencial, tant a l'interior com a l'exterior de la casa. Per una banda, en aquelles àrees de l'habitatge on la il·luminació és un factor crític com per exemple el rebedor, la cuina o el bany, s'incorporen sensors de presència per tal d'activar automàticament les llum en cas que hi hagi moviment a l'interior d'aquestes.

D'altra banda, en aquelles àrees on l'usuari pot o no requerir d'il·luminació com per exemple el menjador, el jardí o la piscina, es realitza un control manual de l'encesa i l'apagada, sempre a través de les interfícies ubicades al llarg de l'habitatge. Finalment, en tots els casos es pot realitzar un control de la intensitat lumínica i del color per tal d'adaptar l'espai a les necessitats de l'usuari en tot moment i augmentar el grau de confort.

- **Gestió de la persianes:** mitjançant la instal·lació d'un sensor que monitoritza la intensitat lumínica a l'interior de l'habitatge, es controla la posició de les persianes. Per una banda, quan el sensor detecta una intensitat elevada de llum, les persianes es pugen per tal d'aprofitar la llum solar i la seva radiació que ajuda a escalfar l'interior. De forma contrària, quan arriben les hores del dia en què comença a caure el sol i per tant, la intensitat lumínica, les persianes es baixen automàticament. Tot i això, l'usuari també pot gestionar la posició de les persianes de forma manual en funció de les seves necessitats en cada moment.

Aquesta funcionalitat compleix gran part dels objectius marcats per la domòtica ja que es tracta d'una tasca automatitzada, aporta benestar a l'usuari ja que no ho ha de realitzar manualment i contribueix en l'eficiència energètica amb un estalvi en el consum d'energia per il·luminar i escalfar l'habitatge.

- **Gestió de la climatització:** mitjançant la instal·lació a l'interior de l'habitatge d'un sensor de temperatura i humitat, es realitza la monitorització del clima. Aquest sensor s'incorpora en la zona del menjador ja que es tracta de la zona amb una superfície més gran i per tant, amb un control del clima de major dificultat. Mitjançant la pàgina web o les pantalles de la casa, l'usuari pot tenir un coneixement en temps real del clima i una visió amb més perspectiva gràcies a les gràfiques creades que mostren l'evolució de la temperatura i humitat amb el pas del temps.

Seguidament, l'usuari pot ajustar el valor de temperatura a les seves necessitats, de tal forma que el dispositiu de climatització s'accionarà o s'apagarà en funció de la temperatura de l'ambient i la desitjada per tal de mantenir un ambient estable adaptat a les peticions de l'usuari. Per tant, l'objectiu de la gestió de la climatització és doble ja que pretén mantenir un nivell alt de benestar amb la personalització de la temperatura i a la vegada, suposa un estalvi energètic ja que tan sols es posa en funcionament el dispositiu de climatització (aire condicionat) quan la temperatura ambiental no s'ajusta a la seleccionada.

- **Gestió de la piscina:** en aquest cas en què el sistema domòtic s'incorpora a una casa que disposa d'una piscina, es realitza un control d'aquesta. Primerament, s'incorpora il·luminació LED a l'interior, la gestió de la qual s'ha vist anteriorment, per tal de permetre la personalització d'aquesta en funció de les necessitats de l'usuari.

A més, s'automatitza l'encesa i apagada de la depuradora per tal de mantenir l'aigua en les millors condicions possibles. Mitjançant un rellotge de temps real, es programa el motor de la depuradora per tal que realitzi el procés tan sols durant els moments especificats, estalviant així un consum significatiu d'energia i automatitzant el procés de tal forma que l'usuari no s'ha de fer càrrec.

- **Gestió dels accessos:** una important funcionalitat dintre de l'àmbit de la seguretat és la de gestionar els accessos i l'estat de les portes. Gràcies al sistema dissenyat, l'usuari pot bloquejar o desbloquejar els panys electrònics instal·lats en els diferents accessos en funció de les seves necessitats. Més concretament, es pot controlar tant la porta que dona accés al recinte residencial com la porta que accedeix a l'interior de l'habitatge.
- **Gestió de l'alarma contra intrusions:** per reforçar la seguretat en l'entorn domèstic, s'implementa un sistema de detecció d'intrusions. El sistema està constituït per una sèrie de sensors de presència instal·lats pel perímetre i per l'interior i una alarma sonora. A través de la pàgina web l'usuari el pot activar o desactivar i si quan aquest es troba activat els sensors detecten una presència o moviment, l'alarma sonora s'activa per alertar tant als usuaris si es troben a l'interior com als possibles intrusos. Finalment, si salta l'alarma també s'indica en la pàgina web sobre la presència d'intrusos.
- **Detecció d'incendis:** l'última funcionalitat implementada per millorar la seguretat del sistema domòtic es tracta de la detecció d'incendis. Gràcies a la instal·lació de sensors detector de foc en les zones més susceptibles de l'habitatge com poden ser la cuina i el menjador, es realitza un monitoreig. En cas que qualsevol dels sensors detecti la presència de flames, una alarma sonora s'activa i s'alerta a l'usuari en la interfície web.

Finalment, un cop vistes totes les funcionalitats implementades en el sistema dissenyat, és important tenir en consideració que aquest sistema domòtic s'enfoca en una implementació a petita escala. Tot i això, el projecte és fàcilment escalable per abastar la gestió d'un nombre superior de dispositius i la realització de tasques amb un grau de sofisticació superior amb l'objectiu de realitzar una aplicació domòtica més realista i a gran escala.

1.7.2. Plànols i imatges

En aquest apartat es presenten els plànols utilitzats per realitzar el prototip i algunes imatges d'un projecte realitzat en 3 dimensions de la casa en què s'incorpora el sistema domòtic.

En referència als plànols, s'han realitzat un total de quatre amb l'objectiu de tenir una visió global de la parcel·la de l'habitatge, de les mesures necessàries per realitzar la construcció del prototip i de la disposició dels diferents elements que componen el sistema per realitzar tota la instal·lació de forma correcta i ordenada.

Els plànols realitzats són els següents i es poden consultar en P:

1. Plànol de la parcel·la (tancaments exteriors, jardí i habitatge)
2. Plànol acotat de la parcel·la
3. Detall acotat de l'habitatge

4. Emplaçament i connexió dels dispositius domòtics

Seguidament, es mostren un seguit d'imatges extretes del projecte en 3 dimensions. Aquest s'ha realitzat mitjançant la versió gratuïta de l'aplicació en línia *HomeByMe* [10]. El projecte inclou tota la parcel·la, mostrant així de forma més realista la zona del jardí, la piscina i l'exterior de l'habitatge.

En primer lloc es mostren les imatges referents a l'exterior de l'habitatge, és a dir, la zona del jardí i de la piscina, el garatge, l'accés a l'interior de la parcel·la, els tancaments i les quatre façanes que presenta.



Figura 4. Model 3D: porta exterior i jardí davanter



Figura 5. Model 3D: piscina i jardí lateral



Figura 6. Model 3D: garatge i jardí lateral



Figura 7. Model 3D: façana davantera



Figura 8. Model 3D: façana lateral dreta



Figura 9. Model 3D: façana lateral esquerra



Figura 10. Model 3D: façana darrera

Finalment, un cop visualitzada la totalitat de la parcel·la, es mostren les imatges corresponents a l'interior de l'habitatge dissenyat. Aquest consta d'una entrada amb un rebedor, un ampli menjador centrat en la il·luminació natural i les vistes a l'exterior, dues habitacions, un bany, una cuina i una estància d'emmagatzematge i neteja.

Les imatges extretes corresponen a una vista en planta per tenir una visió global de la distribució de les diferents zones i seguidament es mostra en detall cada una d'aquestes.



Figura 11. Model 3D: vista en planta de l'interior



Figura 12. Model 3D: rebedor



Figura 13. Model 3D: menjador (vista 1)



Figura 14. Model 3D: menjador (vista 2)



Figura 15. Model 3D: bany



Figura 16. Model 3D: habitació



Figura 17. Model 3D: cuina (vista 1)



Figura 18. Model 3D: cuina (vista 2)



Figura 19. Model 3D: sala de neteja i emmagatzement

1.7.3. Microcontrolador ESP32

El microcontrolador ESP32, desenvolupat per l'empresa multinacional *Espressif Systems*, és un microcontrolador de baix cost i alt rendiment dissenyat específicament per a realitzar tasques en sistemes *embedded* i en aplicacions que comporten l'ús de l'IoT. En els darrers anys, ha guanyat força popularitat gràcies a la seva potència de processament, connectivitat Wi-Fi i Bluetooth integrada, baix consum d'energia i àmplia gamma de perifèrics com per exemple ports ADC, UART, SPI, etc.



Figura 20. Microcontrolador ESP-WROOM-32

Per al desenvolupament d'aquest treball, més concretament s'utilitza el mòdul ESP-WROOM-32. A continuació, a partir de dades extretes del propi full de característiques [11], es detallen alguns dels atributs que fan aquest microcontrolador tan popular:

- **Doble nucli de processament:** integra dos nuclis de processament Tensilica Xtensa de 32 bits amb una freqüència de rellotge ajustable entre 80 i 240 MHz, que permeten executar múltiples tasques de forma simultània i millorar el rendiment en aplicacions que requereixen un processament intensiu.
- **Connectivitat Wi-Fi i Bluetooth:** compta amb connectivitat Wi-Fi 802.11 b/g/n, Bluetooth 4.2 i Bluetooth LE integrat, el que facilita la connexió a xarxes sense fil i la comunicació amb altres dispositius compatibles amb el protocol Bluetooth.

Es pot connectar fàcilment a una xarxa Wi-Fi per accedir a Internet o inclús crear la seva pròpia xarxa per tal que altres microcontroladors o perifèrics s'hi connectin. Aquesta funcionalitat és fonamental per a aplicacions IoT o projectes domòtics, ja que possibilita la comunicació entre diversitat dispositius.

- **Baix consum d'energia:** està dissenyat per optimitzar el consum d'energia, fet que el converteix amb un microcontrolador molt adequat per a aplicacions alimentades amb bateries. També disposa de diferents modes de baix consum que redueixen el corrent a valors inferiors a 5 μ A, permetent així prolongar la durada de les bateries.
- **Àmplia varietat de perifèrics:** ofereix una àmplia gamma de perifèrics integrats tant d'entrada com de sortida. Entre aquests, cal destacar que incorpora ports GPIO, UART, SPI, I2C, ADC, DAC, PWM, etc. Gràcies a aquesta varietat, l'ESP32 facilita la connexió i el control de múltiples sensors, actuadors i dispositius externs.

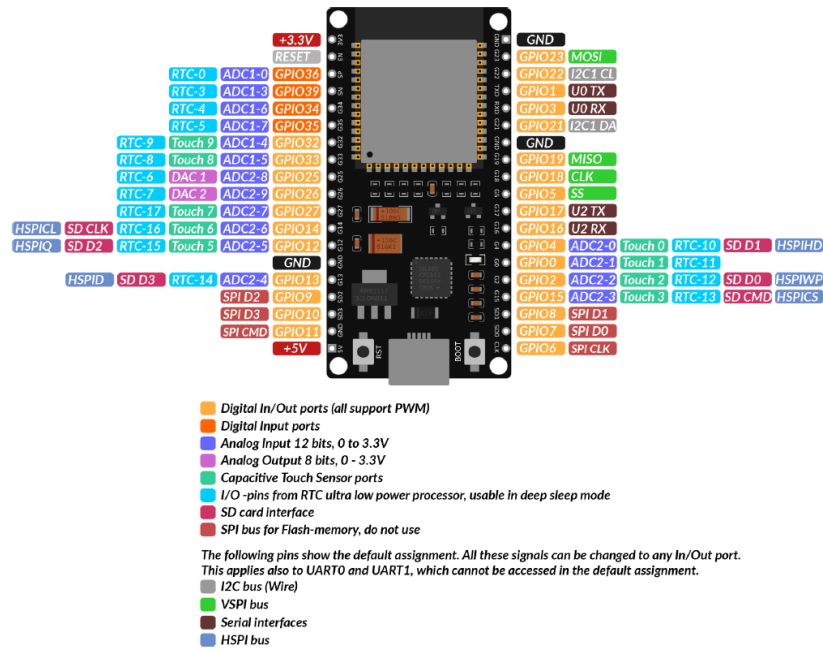


Figura 21. Distribució de pins del ESP-WROOM-32

- **Seguretat:** compta amb funcions de seguretat integrades, com per exemple xifrat AES, generació de nombres aleatoris i funcions de *hash*, que ajuda a protegir les dades i les comunicacions en aplicacions sensibles a la seguretat.
- **Baix cost econòmic:** pot ser adquirit a un preu baix, d'entre 10 i 15€, la qual cosa el fa accessible per a un rang superior de consumidors.

En resum, s'ha escollit aquest microcontrolador per realitzar el treball domòtic ja que l'ESP32 és versàtil, potent, econòmic i ofereix una combinació de característiques molt adequades per a un projecte de l'àmbit domòtic.

1.7.4. Hardware

El hardware és un component fonamental a l'hora d'implementar un sistema domòtic, ja que comprèn tots els dispositius físics encarregats d'interactuar amb l'entorn domèstic per automatitzar les tasques programades. En aquest apartat, es tracta amb detall cada un dels diferents dispositius que componen el sistema domòtic dissenyat, analitzant el seu funcionament, connexionat i propòsit.

Seguidament, cal destacar les raons per les quals s'han escollit els diferents dispositius enfront altres alternatives existents en el mercat. Per aquest motiu, a continuació es detallen els criteris en els que s'ha basat l'elecció del hardware:

- **Cost:** ha de ser el més reduït possible.
- **Accessibilitat:** ha de ser fàcilment accessible en el mercat.
- **Principi de funcionament:** ha de ser el més elemental i bàsic possible.
- **Dimensions:** han de ser les més reduïdes possible per minimitzar l'impacte visual i facilitar la instal·lació.

- **Pes:** ha de ser el més reduït possible per facilitar la instal·lació.
- **Cablejat:** ha de requerir del mínim indispensable pel correcte funcionament i per facilitar la instal·lació.
- **Eficient energèticament:** ha de tenir un consum el més reduït possible.
- **Relació rendiment-cost:** ha de ser econòmic però mantenir unes bones prestacions i garantir la fiabilitat en l'execució de la funció designada.
- **Compatibilitat amb el microcontrolador ESP32.**

Per avaluar el grau de compliment de cada un dels criteris mencionats, s'assignarà una puntuació entre 1 i 5 per cada un d'ells, essent aquestes la nota mínima i màxima respectivament. En cas que hi hagi diverses alternatives per realitzar una mateixa tasca, s'escollirà aquell dispositiu que obtingui una valoració superior en la suma aritmètica de totes les notes.

1.7.4.1. Il·luminació LED

La il·luminació del sistema domòtic dissenyat es basa en tires LED RGB WS2812B o també anomenades NeoPixel. Es tracta de LEDs que disposen de lògica integrada de tal forma que, a diferència de les tires convencionals en les que tots els píxels canvien de color a la vegada, és possible variar el color de forma individual.

Està basada en LED amb unes dimensions de 5 x 5 mm, de baix consum i que integra els tres colors RGB amb un únic encapsulat. S'anomena que aquest tipus d'il·luminació és adreçable individualment ja que dintre de l'encapsulat també afegeix un petit circuit integrat que permet accedir a cada píxel.



Figura 22. Tira LED RGB WS2812B

En aquest projecte, s'utilitza una tira de 5 metres de longitud inicials que es divideix en seccions de diferents llargades en funció de l'habitacle o zona a il·luminar i que es controla de forma independent a través de la pàgina web.

La connexió entre cada secció de LED i el microcontrolador és realitza a través de 3 pins: VCC (5 V), GND i entrada de dades. Els dos primers s'utilitzen com a alimentació i el darrer es connecta a un port GPIO per rebre el senyal de control dels píxels que componen la secció. Cada secció ha de ser connectada a diferents ports per tal de realitzar un control independent de la il·luminació.



Figura 23. Pins tira LED NeoPixel

El funcionament d'un WS2812B és el següent: cada LED disposa d'un integrat que emmagatzema 3 bytes que corresponen als 3 colors (RGB). Cada píxel pot tenir 256 nivells diferents en cada color, el que suposa un total de 16.777.216 (256^3) possibles combinacions.

Quan el microcontrolador envia el senyal de control, el primer LED rep el flux de bytes, els emmagatzema i els transmet al següent. Aquesta cadena d'informació es realitza per tota la tira fins que finalment, amb un senyal específic, cada LED mostra l'últim valor emmagatzemat.

En la composició d'aquest projecte, s'utilitza una tira amb les següents especificacions:

- Densitat de 30 LED per metre.
- Consum d'uns 60 mA (0,3 W) per LED en color blanc a màxima brillantor.
- Nivell de protecció IP30, és a dir, no és impermeable.

Els criteris d'elecció que compleix aquest tipus d'il·luminació són els següents:

Criteri d'elecció	Puntuació	Criteri d'elecció	Puntuació
Cost	4	Cablejat	4
Accessibilitat	5	Eficiència energètica	4
Principi de funcionament	3	Relació rendiment-cost	4
Dimensions	4	Compatibilitat amb ESP32	5
Pes	5	Total:	38

Taula 2. Criteris d'elecció de la il·luminació LED RGB WS2812B

En referència a possibles alternatives a aquest tipus de il·luminació, s'ha considerat l'ús de bombetes LED. Tot i que el principi de funcionament és lleugerament més simple, tenen un cost significativament superior i una instal·lació més complexa degut a les seves dimensions i pes superiors. [12]

criteri d'elecció	Puntuació	criteri d'elecció	Puntuació
Cost	5	Cablejat	4
Accessibilitat	5	Eficiència energètica	4
Principi de funcionament	4	Relació rendiment-cost	4
Dimensions	2	Compatibilitat amb ESP32	5
Pes	2	Total:	35

Taula 3. Puntuació criteris d'elecció de la il·luminació amb bombetes LED

Comparant el grau de compliment dels criteris de selecció, s'observa com la diferència entre ambdues opcions no és gaire significativa, la qual cosa significa que les dues alternatives són vàlides. El principal motiu pel qual s'ha optat per instal·lar una tira LED en lloc de bombetes és la major facilitat en la instal·lació degut a les seves dimensions i pes.

1.7.4.2. Servomotor

Un servomotor és un dispositiu electromecànic utilitzat per controlar la posició angular d'un eix amb un angle específic. A diferència que els motors convencionals, aquests són dispositius de llaç tancat, és a dir, contenen en el seu interior sensors i circuits interns per rebre senyals de control i ajustar la posició amb l'angle desitjat.



Figura 24. Servomotor

Aquest tipus de dispositius són àmpliament utilitzats en projectes de l'àmbit robòtic, impressores 3D i sistemes on es requereix d'un control de moviment precís. En el context d'aquest treball, els servomotors són els encarregats del moviment de les persianes instal·lades en el menjador, pujant-les i baixant-les a mesura que l'usuari ho indica a través de la pàgina web.

La connexió entre el servomotor i l'ESP32 es realitza per mitjà de 3 cables: V_{CC} (5 V), GND i senyal. Els dos primers s'utilitzen com a alimentació mentre que el de control és utilitzat per enviar la senyal de posicionament del motor amb una seqüència de polsos PWM.

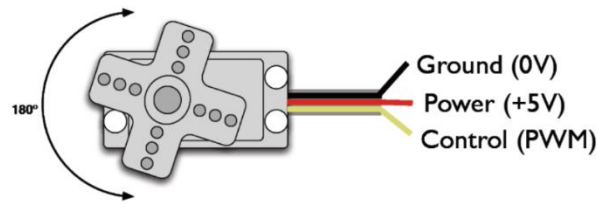


Figura 25. Pins d'un servomotor

El senyal PWM que entra al controlador del servomotor es processat i, en funció de l'amplada del pols, aquest es desplaçarà un nombre de graus determinat. Generalment, els servomotors comercials processen un pols amb un període de 20 ms per tal d'aconseguir realitzar tot el moviment de forma adequada. Més concretament, per posicionar el servomotor en 0° el pols d'entrada ha de ser de 1 ms, per posicionar-lo en 90° ha de ser de 1,5 ms i per posicionar-lo en 180° ha de tenir una durada de 2 ms, tal i com es pot observar en l'esquema següent.

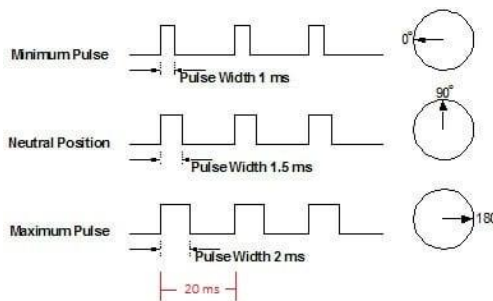


Figura 26. Polsos de control de la posició d'un servomotor

Per tal de realitzar el prototip de la casa domòtica, s'utilitza un servomotor continu o també anomenat de 360° graus. Aquests, a diferència dels anteriors que tan sols poden variar la seva posició en un angle d'entre 0° i 180° , giren de forma contínua. Per aquest motiu, tot i que el principi de funcionament és el mateix, ja no es tracta d'un control de posició en llaç tancat sinó de velocitat i en llaç obert.

En aquest cas, el senyal PWM és utilitzat per realitzar un control sobre el sentit de rotació de l'eix que pot ser a favor o en contra del sentit de les agulles del rellotge en funció del cicle de treball escollit.

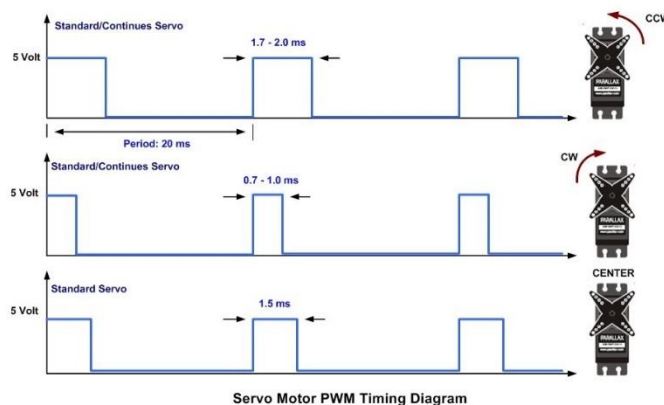


Figura 27. Polsos de control de la rotació d'un servomotor continu

Els criteris d'elecció que compleix aquest tipus de motor són els següents:

criteri d'elecció	Puntuació	Criteri d'elecció	Puntuació
Cost	4	Cablejat	4
Accessibilitat	5	Eficiència energètica	4
Principi de funcionament	2	Relació rendiment-cost	4
Dimensions	5	Compatibilitat amb ESP32	5
Pes	5	Total:	38

Taula 4. Puntuació criteris d'elecció del servomotor continu

Per realitzar aquesta tasca, s'ha considerat també l'ús de motors pas a pas. Els principals inconvenients que presenten són el seu elevat cost enfront els escollits, el major consum d'energia i les seves superiors dimensions i pes que suposen una dificultat afegida en la instal·lació del sistema. [13]

criteri d'elecció	Puntuació	Criteri d'elecció	Puntuació
Cost	2	Cablejat	4
Accessibilitat	5	Eficiència energètica	2
Principi de funcionament	4	Relació rendiment-cost	5
Dimensions	3	Compatibilitat amb ESP32	5
Pes	3	Total:	33

Taula 5. Puntuació criteris d'elecció del motor pas a pas

1.7.4.3. Sensor DHT22

El DHT22 es tracta d'un sensor digital que permet realitzar mesures de temperatura i humitat de forma simultània. Per realitzar les mesures, integra un sensor capacitiu d'humitat i un termistor per avaluar l'aire circumdant i disposa d'un processador intern que genera un senyal digital el resultat del qual s'analitza posteriorment a través del microcontrolador.



Figura 28. Sensor DHT22

Memòria

És tracta d'un sensor molt popular i utilitzat en aplicacions de control automàtic del clima ja sigui en ambients domèstics o en l'agricultura.

La connexió entre el microcontrolador escollit i el sensor es realitza a través de 3 pins tal i com s'observa en la Figura 10: V_{CC} (3,3 V), GND i senyal. Els dos primers s'utilitzen com a alimentació mentre que el restant és el que conté el senyal digital amb la mesura realitzada.

Tot i que no és tracta d'un sensor d'altra precisió, disposa d'unes característiques molt acceptables en relació al seu preu que el fan idoni per instal·lar-lo en projectes reals de monitoreig. Les característiques tècniques més rellevants del sensor DHT22 són les següents:

- Mesura de la temperatura entre -40 °C i 80 °C, amb una precisió de 0,5 °C.
- Mesura de la humitat entre 0 i 100%, amb una precisió d'entre 2 i 5%.
- Freqüència de mostreig de 2 Hz (realitza 2 mesures per segon).

Un dels principals desavantatges d'aquest tipus de sensors de temperatura i humitat és que tenen una inèrcia elevada i temps de resposta grans i per tant, tarden més en reflectir en la mesura resultant els canvis que sofreix l'entorn. Tot i això, aquesta característica no suposa cap problema per al desenvolupament del projecte ja que no és necessari un temps de resposta molt ràpid al tractar-se de la monitorització de l'ambient domèstic en què el mostreig es realitza en intervals de temps elevats.

Les puntuacions assignades per cada criteri de selecció són les mostrades en la taula a continuació:

Criteri d'elecció	Puntuació	Criteri d'elecció	Puntuació
Cost	5	Cablejat	4
Accessibilitat	5	Eficiència energètica	4
Principi de funcionament	4	Relació rendiment-cost	5
Dimensions	5	Compatibilitat amb ESP32	5
Pes	5	Total:	42

Taula 6. Puntuació criteris d'elecció del sensor de temperatura i humitat

Tal i com s'observa, es un dispositiu que compleix a la perfecció amb les necessitats del projecte, ja que es tracta d'un sensor molt econòmic i accessible, amb unes característiques físiques ideals per ser instal·lat en un sistema domòtic i amb un rendiment superior al necessari. Per aquests motius, no s'ha valorat incorporar qualsevol altre model de sensor per realitzar aquesta tasca de monitorització del clima interior.

1.7.4.4. Sensor d'infrarojos passiu

Els sensors d'infrarojos passius, o també anomenats sensors PIR, són dispositius per a la detecció de moviment basats en la mesura de la radiació infraroja. Tots els cossos, ja estiguin vius o no, emeten una determinada quantitat d'energia infraroja en funció de la seva temperatura i els dispositius PIR són capaços de captar-la i convertir-la en un senyal elèctric.



Figura 29. Sensor PIR

El principi de funcionament és el següent: cada sensor està dividit en dos camps i es disposa d'un circuit elèctric que compensa totes dues mesures. Si tots dos camps reben la mateixa quantitat d'infrarojos el senyal elèctric resultant és nul. Per contra, si els dos camps realitzen una mesura diferent, es genera un senyal elèctric. D'aquesta forma, si un objecte travessa un dels camps es genera un senyal elèctric diferencial que és captat pel sensor i emet un senyal digital.

L'altre element restant perquè tot funcioni és l'òptica del sensor. Consisteix en una cúpula de plàstic que divideix l'espai en zones i enfoca la radiació infraroja a cadascun dels camps del PIR. D'aquesta manera, el sensor capta una mitjana de la radiació infraroja de l'entorn i quan un objecte entra en rang, alguna de les zones marcades per l'òptica rebrà una quantitat diferent de radiació, que serà captada per un dels camps.

Aquests dispositius són molt utilitzats en aplicacions domòtiques d'il·luminació o en sistemes de seguretat. En el context d'aquest projecte, la seva funció és doble. Per una banda, s'encarrega de detectar la presència humana en determinades zones i habitatges del domicili per tal de controlar la il·luminació i per l'altra, s'encarrega de detectar intrusos si l'alarma es troba connectada. A més, tenen un paper fonamental en l'eficiència energètica ja que la il·luminació LED incorporada tan sols consumirà quan aquest sensor detecti una presència.

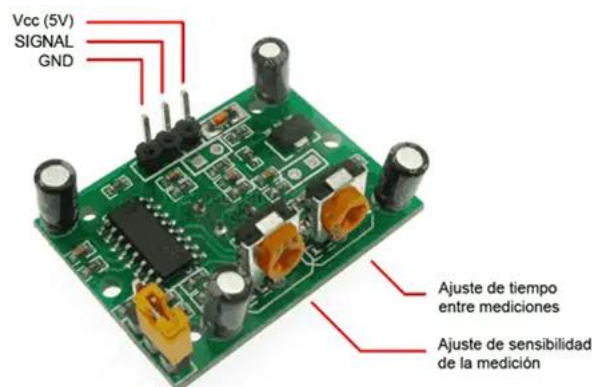


Figura 30. Pins del sensor PIR

La connexió entre el microcontrolador escollit i el sensor es realitza a través de 3 pins tal i com s'observa en la Figura 14: V_{CC} (5V), GND i senyal. Els dos primers s'utilitzen com a alimentació mentre que el restant és el que conté el senyal digital amb la mesura realitzada. També s'observa que disposa de dos potenciòmetres que l'usuari pot variar per tal d'ajustar la sensibilitat en la mesura i el temps entre mesures.

La puntuació obtinguda en referència als criteris que compleix aquest tipus de sensor utilitzat per detectar la presència humana són els mostrats en la taula següent:

Criteri d'elecció	Puntuació	Criteri d'elecció	Puntuació
Cost	4	Cablejat	4
Accessibilitat	5	Eficiència energètica	4
Principi de funcionament	3	Relació rendiment-cost	5
Dimensions	5	Compatibilitat amb ESP32	5
Pes	5	Total:	40

Taula 7. Puntuació criteris d'elecció del sensor PIR

En quant al tipus de sensor per realitzar aquesta funcionalitat, no s'ha considerat utilitzar cap altre tipus de dispositiu ja que aquest és el majoritàriament implementat per aquestes tasques. Per aquest motiu, l'elecció s'ha basat en escollir el que més s'adapta al prototip de la casa en quant a dimensions i pes.

1.7.4.5. Sensor detector de flama

El sensor detector de flama o també anomenat sensor de flama òptic és un dispositiu que permet detectar l'existència d'una combustió a partir de la llum que emet aquesta i enviar un senyal digital al microcontrolador per tal de ser processat.

La flama és un fenomen d'emissió de llum generat durant els processos de combustió els quals alliberen grans quantitats d'energia en forma de calor. Durant aquesta reacció es generen compostos intermedis que alliberen part de la seva energia mitjançant l'emissió de llum d'una determinada longitud d'ona. Específicament, en les combustions de productes que contenen carboni en presència d'oxigen, s'observen dos pics d'emissió característics en l'ultraviolat, amb longituds d'ona d'entre 185 nm i 260 nm, i en l'infraroig, amb longituds d'entre 4400 nm i 4600 nm. Finalment, per detectar aquests pics i amb ells, la presència d'una flama, els sensors s'ajusten a les longituds característiques mencionades i mitjançant la combinació de senyals ultraviolats i infrarojos es realitza la detecció.

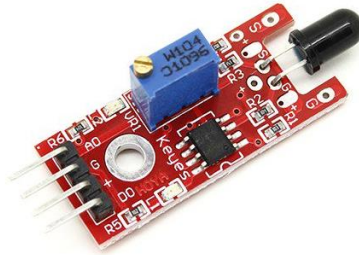


Figura 31. Sensor detector de flama

Aquest tipus de sensors són àmpliament utilitzats en el sector industrial. Degut a l'existència de maquinària que executa processos susceptibles a generar flames en entorns amb gran facilitat de combustió, s'incorporen aquests dispositius com a mesura de seguretat per tal d'aturar qualsevol procés en cas de detectar l'inici d'un incendi.

En el context d'aquest projecte, el sensor s'instal·la en l'espai més susceptible a sofrir un incendi com és la cuina. Mitjançant la monitorització de la possible presència d'un incendi, la combinació d'aquest sensor amb una alarma fan d'aquest un bon sistema de seguretat per alertar l'usuari d'un possible perill en el seu domicili.

La connexió entre el sensor i el microcontrolador escollit consisteix en connectar de 3 dels 4 pins que incorpora: VCC (5 V), GND i DO. Els dos primers s'utilitzen com a alimentació mentre que el restant és el que conté el senyal digital amb la mesura realitzada. Si l'usuari desitja processar el senyal de sortida en forma analògica, ha de substituir la connexió de sortida del pin DO (Digital Output) pel pin AO (Analog Output).

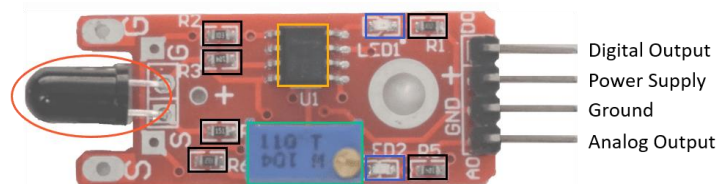


Figura 32. Pins d'un sensor de flama

Tot i que no és tracta d'un sensor d'altra precisió, disposa d'unes característiques molt acceptables en relació al seu preu que el fan idoni per instal·lar-lo en un projecte com aquest. Les característiques tècniques més rellevants són les següents:

- Disposa d'un únic sensor d'infrarojos ajustable entre 760 nm i 1100 nm.
- Angle de detecció de 60°.
- Distància de detecció entre 4 cm i 80 cm.

Aquest sensor compleix amb els criteris d'elecció següents, obtenint la puntuació mostrada en la taula següent:

Criteri d'elecció	Puntuació	Criteri d'elecció	Puntuació
Cost	5	Cablejat	4
Accessibilitat	5	Eficiència energètica	4
Principi de funcionament	4	Relació rendiment-cost	4
Dimensions	5	Compatibilitat amb ESP32	5
Pes	5	Total:	41

Taula 8. Puntuació criteris d'elecció del sensor detector de flama

Per realitzar aquesta tasca, s'ha considerat també l'ús de dispositius detectors de fum. Aquests realitzen la detecció d'un incendi mitjançant l'anàlisi de la quantitat de diòxid de carboni en l'atmosfera de l'habitatge, un principi de funcionament més complex. A més, el cost i les dimensions d'aquests tipus de dispositius són considerablement superiors, un factor amb molta importància en el cas d'un prototip com el dissenyat [14]. La puntuació obtinguda per aquest dispositiu és la mostrada a continuació:

Criteri d'elecció	Puntuació	Criteri d'elecció	Puntuació
Cost	3	Cablejat	4
Accessibilitat	5	Eficiència energètica	4
Principi de funcionament	3	Relació rendiment-cost	5
Dimensions	3	Compatibilitat amb ESP32	5
Pes	3	Total:	35

Taula 9. Puntuació criteris d'elecció del sensor detector de fum

1.7.4.6. Brunzidor actiu

Un brunzidor actiu és un dispositiu que genera un so d'una freqüència determinada i fixa quan s'alimenta a una determinada tensió. Integra en el seu interior un oscil·lador simple de tal forma que tan sols es necessari subministrar corrent al dispositiu per fer-lo sonar, la qual cosa el converteix amb un dispositiu amb un principi de funcionament bàsic.



Figura 33. Brunzidor actiu

Un dels grans avantatges d'aquest tipus de dispositiu per emetre so és que no suposa cap càrrega per al processador del microcontrolador ja que no ha de generar l'ona elèctrica que es convertirà en so sinó que el resultat a emetre és sempre el mateix.

La funcionalitat d'aquest dispositiu en el sistema domòtic dissenyat va associada al sensor detector de flama i al de presència. Quan es detecta l'aparició d'un incendi en l'habitatge o la intrusió a l'interior de l'habitatge quan l'alarma està connectada, s'alimenta el brunzidor per tal que funcioni a mode d'alarma sonora i alerti als usuaris del perill existent.

La connexió entre el microcontrolador escollit i el brunzidor es realitza a través de 3 pins: V_{CC} (5 V), GND i senyal. Els dos primers s'utilitzen com a alimentació mentre que el darrer funciona a mode de relé per activar el pas de corrent o no a l'oscil·lador.

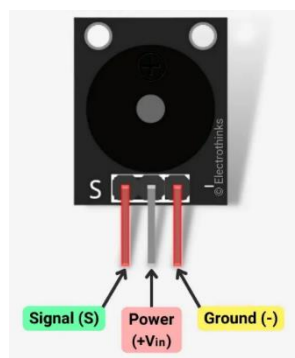


Figura 34. Pins d'un brunzidor

La puntuació obtinguda per aquest dispositiu per realitzar la tasca d'alarma sonora és la mostrada en la següent taula:

Criteri d'elecció	Puntuació	Criteri d'elecció	Puntuació
Cost	5	Cablejat	4
Accessibilitat	5	Eficiència energètica	5
Principi de funcionament	5	Relació rendiment-cost	4
Dimensions	5	Compatibilitat amb ESP32	5
Pes	5	Total:	43

Taula 10. Puntuació criteris d'elecció del brunzidor actiu

Com a possible alternativa s'ha considerat l'ús d'un brunzidor passiu, que a diferència de l'actiu, permet modificar el senyal elèctric per variar el to a emetre i aconseguir així diferents resultats. El principal desavantatge és la complexitat existent en el control i modulació del senyal, significat així una dificultat afegida en el principi de funcionament i per tant, una puntuació menor en aquest criteri. La resta de característiques són les mateixes ja que disposen de la mateixa morfologia, cablejat i cost. [15]

Criteri d'elecció	Puntuació	Criteri d'elecció	Puntuació
Cost	5	Cablejat	4
Accessibilitat	5	Eficiència energètica	5
Principi de funcionament	2	Relació rendiment-cost	4
Dimensions	5	Compatibilitat amb ESP32	5
Pes	5	Total:	40

Taula 11. Puntuació criteris d'elecció d'un brunzidor passiu

1.7.4.7. Bomba hidràulica

Una bomba hidràulica és un dispositiu que permet incrementar l'energia cinètica d'un cabal d'aigua. Es tracta de dispositius àmpliament reconeguts i implementats en la indústria degut a la seva aplicació en instal·lacions de sistemes de proveïment, climatització i depuració d'aigua.

Actualment, el mercat ofereix una gran varietat de bombes hidràuliques que abasten un ampli ventall de potències i característiques hidràuliques. Per a la implementació d'aquest projecte s'utilitza una bomba submergible amb un corrent nominal de 0,18 A i una capacitat de descàrrega d'aigua de 100 L/h.



Figura 35. Bomba hidràulica

La funcionalitat d'aquest dispositiu en el sistema domòtic dissenyat és la de simular el moviment d'aigua entre la piscina i la depuradora, de tal forma que quan aquesta s'actua agafarà aigua, la transportarà a la depuradora i un cop realitzat el procés de neteja, retornarà a la piscina.

Memòria

El principi de funcionament es basa en un motor elèctric acoblat a un element rotatiu format per aspes. Aquestes, al girar transmeten part de l'energia cinètica al fluid circulant i la seva ergonomia determina la quantitat d'energia transmesa i el grau en què es reparteix.

En el cas de les bombes submergibles, el motor elèctric es troba segellat en un encapsulat que l'aïlla del fluid a transmetre, de tal forma que tot el dispositiu pot introduir-se evitant així qualsevol curtcircuit o la problemàtica d'haver d'instal·lar una canonada d'aspiració.

La bomba no disposa d'un controlador incorporat de tal forma que l'accionament i gestió del motor elèctric es realitza mitjançant l'acció combinada entre el microcontrolador seleccionat i un relé. L'esquema de connexió és el mostrat a continuació:

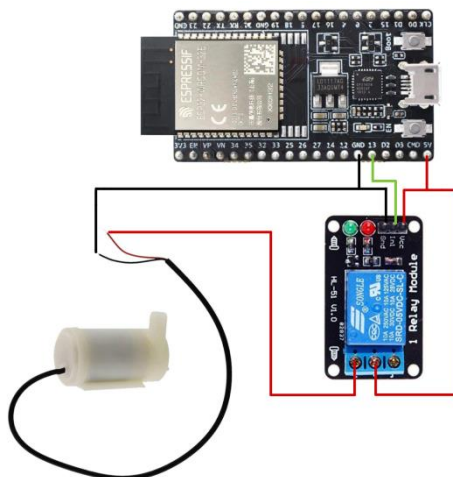


Figura 36. Esquema de connexió del conjunt bomba-relé

El grau de compliment dels diferents criteris de selecció per part de la bomba hidràulica seleccionada són els mostrats en la taula següent:

Criteri d'elecció	Puntuació	Criteri d'elecció	Puntuació
Cost	4	Cablejat	4
Accessibilitat	5	Eficiència energètica	3
Principi de funcionament	5	Relació rendiment-cost	5
Dimensions	4	Compatibilitat amb ESP32	4
Pes	4	Total:	38

Taula 12. Puntuació criteris d'elecció de la bomba hidràulica

Com a alternativa possible, tan sols s'ha considerat l'ús de diferents bombes amb el mateix principi de funcionament però prestacions superiors, ja sigui fent referència al consum o a la capacitat de descàrrega. Cal destacar però que un augment en les característiques funcionals suposa un increment en les dimensions, pes i cost. Per aquest motiu, s'ha seleccionat l'opció existent en el mercat amb una relació ideal entre rendiment i cost.

1.7.4.8. Fotoresistor

Un fotoresistor és un dispositiu la resistència del qual varia en funció de la llum rebuda. També conegut com sensor LDR (*Light-Dependent Resistor*), permet realitzar una estimació de la intensitat de la llum en un ambient determinat gràcies a aquesta variació resistiva i mitjançant una lectura analògica.

Actualment, existeixen en el mercat un gran nombre de fotoresistors diferents en funció de la seva resistència quan es troben exposats en l'obscuritat i en il·luminació directa o en funció del temps de resposta. Específicament, en aquest projecte s'implementa un fotoresistor de 10 k Ω amb un temps de resposta de 30 ms, significat així que presenta una resistència d'aquest valor en un espai amb il·luminació ambiental.

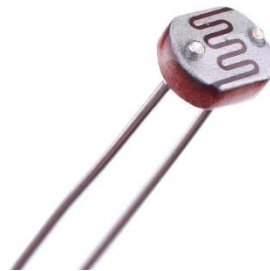


Figura 37. Fotoresistor

El principi de funcionament d'aquest dispositiu es basa en un semiconductor que incorpora, sulfur de cadmi (CdS) generalment ja que és un material sensible a totes les radiacions lluminoses de l'espectre visible. Per una banda, quan no està exposat a radiació, els electrons es troben units fermament en els àtoms que el conformen. Per contra, quan incideix radiació lumínica, una certa quantitat de fotons són absorbits, provocant així que els electrons passin a la banda de conducció disminuint la resistència del component. Resumidament, un fotoresistor disminueix la seva resistència a mesura que la intensitat lumínica augmenta sobre ell i la incrementa quan l'ambient es torna més fosc.

La funcionalitat d'aquest sensor en el projecte domòtic dissenyat és la de mesurar la intensitat de llum en determinats espais per controlar el moviment de les persianes. Per exemple, si la quantitat de llum que arriba a la casa és significativa donarà l'ordre de pujar les persianes amb l'objectiu d'estalviar energia.

La connexió entre el microcontrolador escollit i el fotoresistor es realitza a través de 3 pins: V_{CC} (5V), GND i senyal. Els dos primers s'utilitzen com a alimentació mentre que el restant és el que conté el senyal amb la mesura realitzada.

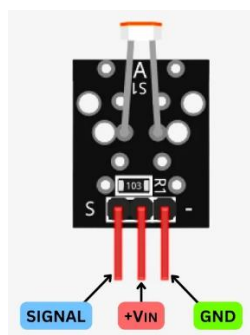


Figura 38. Esquema del fotoresistor

La puntuació obtinguda en funció dels criteris de selecció per aquest sensor d'il·luminació són els mostrats en la taula a continuació:

Criteri d'elecció	Puntuació	Criteri d'elecció	Puntuació
Cost	5	Cablejat	4
Accessibilitat	5	Eficiència energètica	4
Principi de funcionament	4	Relació rendiment-cost	4
Dimensions	5	Compatibilitat amb ESP32	5
Pes	5	Total:	41

Taula 13. Puntuació criteris d'elecció del fotoresistor

Com a possible alternativa al fotoresistor de 10 k Ω escollit, s'ha considerat l'ús d'un sensor de llum crepuscular. Es tracta d'un sensor electrònic que permet l'activació automàtica d'un dispositiu secundari quan baixa la llum natural d'un cert nivell. Tot i que el seu principi de funcionament és molt similar al d'un fotoresistor, els principals inconvenients que presenta són el seu elevat cost, dimensions i pes. [16]

Criteri d'elecció	Puntuació	Criteri d'elecció	Puntuació
Cost	3	Cablejat	4
Accessibilitat	5	Eficiència energètica	3
Principi de funcionament	3	Relació rendiment-cost	4
Dimensions	3	Compatibilitat amb ESP32	4
Pes	3	Total:	32

Taula 14. Puntuació criteris d'elecció d'un sensor de llum crepuscular

1.7.4.9. Taula resum

Un cop realitzada la definició de les característiques de tots els dispositius físics que incorpora el sistema domòtic i l'anàlisi de les diferents alternatives disponibles en cada cas, a continuació es mostra una taula a mode de resum amb els dispositius finalment seleccionats.

Dispositiu	Funcionalitat	Puntuació	Datasheet
Tira LED RGB	Il·luminació instal·lada en totes les zones l'habitatge. Molt personalitzable (color i intensitat) i poc consum.	38	[17]
Servomotor continu	Motor utilitzat per realitzar el moviment de les persianes.	38	[18]

Sensor DHT22	Sensor per monitoritzar la temperatura i humitat de l'interior de l'habitatge.	42	[19]
Sensor d'infrarojos passiu	Sensor per monitoritzar la detecció de moviment i presència humana en determinades zones de l'habitatge.	40	[20]
Sensor detector de flama	Sensor per monitoritzar i detectar la possible aparició d'un incendi en l'interior de l'habitatge.	41	[21]
Brunzidor actiu	Dispositiu d'alarma sonora per alertar a l'usuari de determinats perills (incendi, intrusió...)	43	[22]
Bomba hidràulica	Motor utilitzat per transportar l'aigua de l'interior de la piscina a la depuradora.	38	[23]
Fotoreistor	Sensor per monitoritzar la intensitat de llum solar que penetra a l'habitatge.	41	[24]

Taula 15. Taula resum dels dispositius de hardware

1.7.5. Software

El software és el cervell de qualsevol sistema domòtic ja que es l'encarregat de proporcionar la aparent intel·ligència i la capacitat de control necessària per tal d'automatitzar les tasques, gestionar els diferents dispositius i oferir a l'usuari una experiència personalitzada. En aquest apartat, s'explora amb detall tant el programari utilitzat en la programació del sistema domòtic com el codi dissenyat per realitzar totes les tasques especificades anteriorment.

1.7.5.1. Entorn de programació

A continuació, es mostren els diferents entorns de programació utilitzats per al desenvolupament del sistema domòtic, analitzant les seves característiques i avantatges i justificant la seva funcionalitat i elecció en el treball.

1.7.5.1.1. Arduino IDE

L'entorn de desenvolupament integrat (IDE) d'Arduino és una plataforma de codi obert àmpliament utilitzada que facilita l'escriptura, compilació i càrrega de codi en microcontroladors compatibles. És una eina molt popular entre els desenvolupadors de software degut a la seva simplicitat, facilitat d'ús i àmplia comunitat d'usuaris i molt utilitzat en projectes d'IoT i per tant, de domòtica.

No tan sols s'utilitza per programar microcontroladors d'Arduino sinó que amb l'ajuda de nuclis de tercers, es pot utilitzar pel desenvolupament de codi en microcontroladors d'altres proveïdors. En el cas de l'ESP32, es necessari instal·lar el suport per aquest microcontrolador abans d'iniciar el projecte. Un cop instal·lat, l'usuari pot seleccionar la placa ESP32 i configurar els paràmetres necessaris com el port de comunicació o la velocitat de transferència.

La funcionalitat de l'Arduino IDE en aquest projecte és la de carregar el codi en el microcontrolador escollit. Com es pot veure en apartats posteriors, el codi realitzat amb aquest entorn de desenvolupament té dos objectius fonamentals: controlar els dispositius en funció de les lectures dels sensors o de les comandes de l'usuari i establir la comunicació via Wi-Fi entre la pàgina web, la base de dades i el microcontrolador.



Figura 39. Icona Arduino IDE

Memòria

Les principals raons per les quals s'ha escollit aquest entorn per programar l'ESP32 són les següents:

- Arduino IDE ofereix una àmplia gamma de biblioteques i llibreries que faciliten la interacció entre els perifèrics i dispositius (sensors, actuadors, etc.) amb la placa. A més, moltes de les llibreries són pròpies d'Arduino la qual cosa proporciona una major seguretat i accés a més documentació.
- És un entorn completament gratuït, molt fàcil de descarregar i d'utilitzar.
- És totalment compatible amb els microcontroladors de la sèrie ESP32 i a més, es disposa de molta documentació en línia referent a aquest.
- Compta amb una gran i activa comunitat de desenvolupadors de software que comparteixen projectes, demostracions, manuals d'ús en línia.

Finalment, cal destacar que Arduino IDE és una opció molt popular per desenvolupar projectes domòtics ja que permet controlar fàcil i eficientment la gran varietat de sensors i actuadors incorporats en un sistema com el dissenyat.

1.7.5.1.2. Visual Studio Code

Visual Studio Code (també anomenat, VS Code) és un editor de codi font implementat per Microsoft que ofereix una àmplia gamma de característiques i extensions pel desenvolupament de software. Es tracta d'un programari lliure i multiplataforma que es troba disponible en els sistemes operatius més utilitzats com Windows, Linux i macOS.

Actualment, dintre del sector de la programació és un dels entorns de desenvolupament més utilitzats a nivell mundial gràcies a que disposa d'un suport per a la depuració de codi i una gran varietat d'extensions que permeten l'escriptura i execució de programes en gairebé qualsevol llenguatge.

La principal tasca del Visual Studio Code en aquest projecte és la de servir d'entorn de programació pels codis referents a la pàgina web que controla el sistema domòtic, és a dir, els codis HTML, CSS i JavaScript. A més, s'ha instal·lat l'extensió "Live Server", que crea un servidor de desenvolupament local amb funció de recàrrega en directe amb l'objectiu de veure com els canvis introduïts en el codi tenen efecte en la pàgina durant la programació.



Figura 40. Icona Visual Studio Code

Els principals motius pels quals s'ha escollit Visual Studio Code per realitzar gran part de la programació d'aquest projecte són els esmenats a continuació:

- VS Code proporciona un editor de text altament personalitzable que inclou funcions com ressaltat de sintaxis o autocompletat de codi que faciliten l'escriptura i edició de programes.
- Integra la funció de depuració que ajuda a detectar errors en el codi de forma automàtica, facilitant així la identificació i detecció d'errors.
- És tracta d'un software gratuït, molt fàcil de descarregar i d'utilitzar.
- Incorpora una gran varietat d'extensions que permeten a l'usuari personalitzar i ampliar les capacitats de l'editor de text d'acord amb les necessitats.
- Permet l'escriptura i depuració de codi en qualsevol llenguatge, de tal forma que la tasca de programació és s'unifica en un sol entorn.

En resum, l'entorn de desenvolupament que ofereix Visual Studio Code és molt adequat per realitzar les tasques que se li assignen en aquest treball degut a la facilitat d'ús i les funcionalitats que incorpora.

1.7.5.2. Firebase Realtime Database

Firebase Realtime Database és una base de dades al núvol en temps real que proporciona als desenvolupadors de software una forma d'emmagatzemar i sincronitzar dades entre usuaris i dispositius en temps real. Desenvolupat per Google com a part de la plataforma Firebase, Realtime Database és una solució escalable i flexible per a aplicacions que requereixen un emmagatzematge i gestió de dades en temps real i una sincronització instantània entre clients i dispositius.

La tasca que realitza en el projecte és la d'emmagatzemar les dades obtingudes a través dels sensors o bé a través de les interaccions que l'usuari realitza amb la pàgina web. Posteriorment, aquestes dades són utilitzades per realitzar una determinada funció, ja sigui encendre o apagar una llum, activar una alarma, etc.



Figura 41. Icona Firebase Realtime Database

Les principals característiques de Firebase Realtime Database i que fan d'aquesta base de dades una bona opció per tal de ser introduïda en un projecte domòtic són les següents:

- **Emmagatzematge en temps real:** Firebase Realtime Database emmagatzema dades al núvol i les sincronitza en temps real entre tots els clients connectats. D'aquesta manera qualsevol canvi realitzat a les dades es reflectirà instantàniament en tots els dispositius connectats a la base de dades.
- **Sincronització *offline*:** proporciona suport per a la sincronització sense connexió, permetent així als usuaris accedir i modificar dades fins i tot quan no tenen connexió a Internet. Els canvis realitzats fora de línia es sincronitzen automàticament amb la base de dades una vegada que es restableix la connexió.
- **Seguretat:** la base de dades ofereix opcions de seguretat mitjançant regles personalitzables que permeten als propietaris controlar els usuaris que poden accedir i modificar les dades.

Firebase Realtime Database és una opció molt implementada en projectes de l'àmbit domòtic ja que requereixen la sincronització de dades en temps real entre múltiples dispositius i usuaris. Per aquest motiu, els usuaris d'un sistema domòtic poden utilitzar la base de dades per emmagatzemar i sincronitzar informació sobre l'estat dels dispositius, esdeveniments del sistema i configuracions obtenint així una experiència domòtica fluida, sincronitzada i interactiva.

1.7.5.3. Codi per a la pàgina web

En aquest apartat, s'analitzen els codis corresponents a la pàgina web dissenyada per al control i monitorització del sistema domòtic. Aquesta té com a nom *My Home* i és una part fonamental del projecte, ja que proporciona la interfície principal a través de la qual els usuaris interactuen amb el sistema des de qualsevol dispositiu amb accés a Internet.

Per tal de dissenyar, personalitzar i fer funcional una pàgina web orientada a la domòtica s'han creat un total de 3 tipus de codi diferents en funció de la seva tasca.

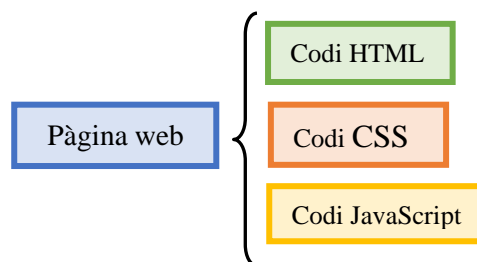


Figura 42. Diagrama dels diferents codis que componen la pàgina web

En primer lloc, s'ha dissenyat un codi amb llenguatge HTML que és l'encarregat de crear la web i la estructura que la compon. Seguidament, mitjançant un codi CSS s'ha aplicat l'estil corresponent per personalitzar-la amb unes característiques visuals agradables per a l'usuari. Finalment, amb la implementació del codi JavaScript s'ha dotat d'interactivitat i dinamisme la interfície web a través de la comunicació bidireccional amb la base de dades.

1.7.5.3.1. HTML

Per tal de tenir una visió global del programa i previ a l'explicació detallada de com s'ha dissenyat, es mostra un esquema que resumeix l'estructura definida pel document HTML:

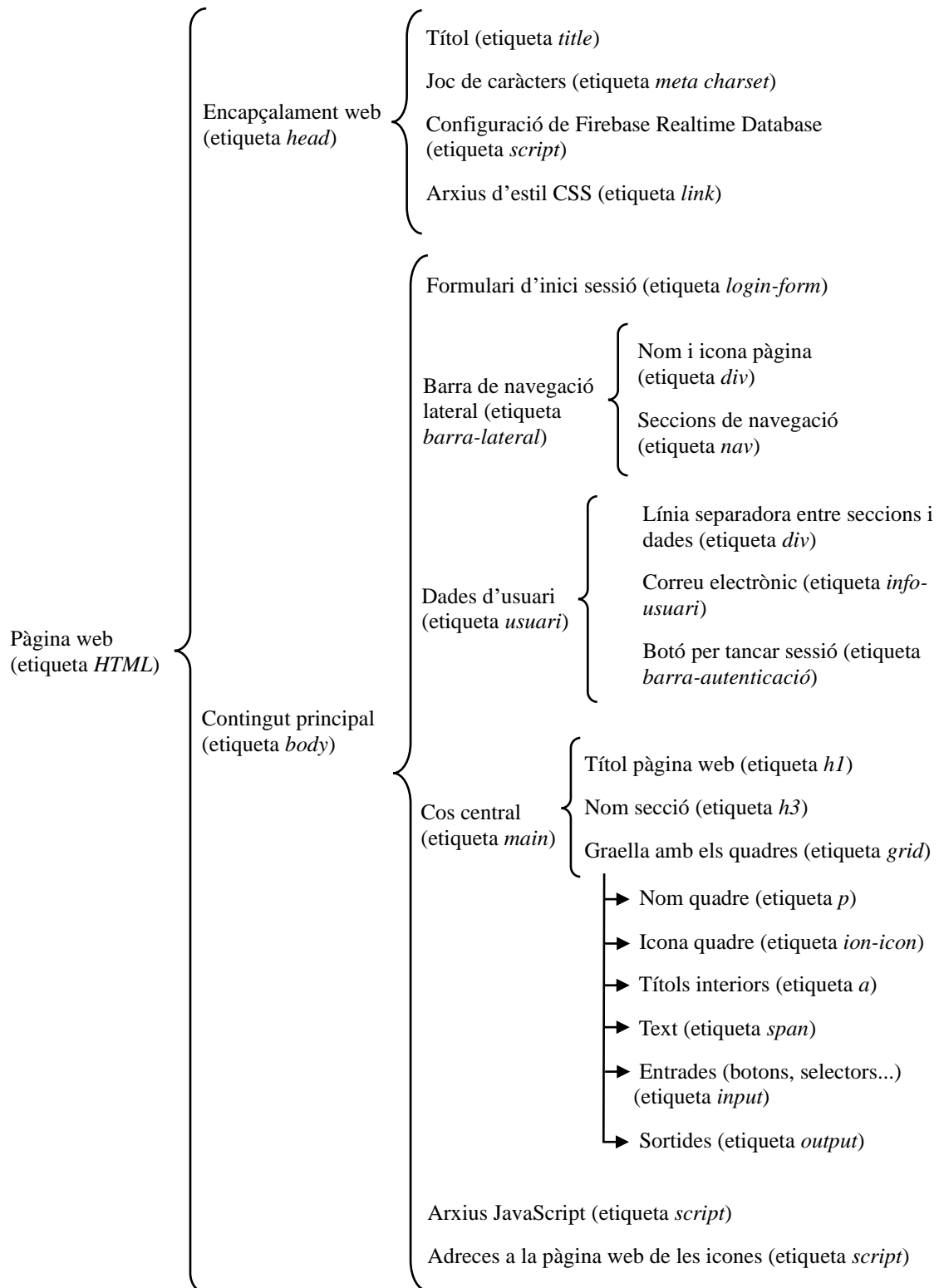


Figura 43. Diagrama estructural del codi HTML

Memòria

Per començar, es defineix i s'inicia el document HTML amb la primera línia del codi. Dintre l'estructura *head*, en primer lloc es realitza la declaració de diverses característiques de la pàgina amb les instruccions següents:

- `<meta charset="utf-8">` : especifica el conjunt de caràcters UTF-8 per tal d'admetre una àmplia gamma de caràcters.
- `<meta name="viewport" content="width=device-width, initial-scale=1">` : ajusta la visualització inicial de la pàgina per tal que s'ajusti a l'amplada del dispositiu i estableix el nivell inicial de zoom.

Seguidament es realitza la declaració de les adreces externes utilitzades com la Firebase Realtime Database i la llibreria HighCharts [25] per realitzar tot tipus de gràfics. Aquest codi s'obté directament de la pròpia pagina web de la llibreria.

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <title>My Home</title>

    <!--Llibreria Highcharts (gràfics)-->
    <script src="https://code.highcharts.com/highcharts.js"></script>
    <script src="https://cdn.rawgit.com/Mikhus/canvas-gauges/gh-
pages/download/2.1.7/all/gauge.min.js"></script>
```

En referència a la base de dades, s'inclouen els scripts de Firebase App, Auth i Database, la funcionalitat dels quals és l'autenticació d'usuaris i la interacció amb la base de dades en temps real. A continuació, s'adjunta el codi proporcionat en els ajustos de Firebase que conté l'objecte de configuració amb els identificadors i les credencials de l'aplicació. Cal destacar que aquestes són úniques i privades per cada projecte i per tal de no ser mostrades es defineixen com a "XXXXXX".

```
<!--Versió de Firebase -->
<script src="https://www.gstatic.com/firebasejs/8.8.1/firebase-
app.js"></script>
<script src="https://www.gstatic.com/firebasejs/8.8.1/firebase-
auth.js"></script>
<script src="https://www.gstatic.com/firebasejs/8.8.1/firebase-
database.js"></script>

<!-- Configuració web app Firebase -->
<script>
const firebaseConfig = {
  apiKey: "XXXXXXXXXXXXXXXXXX",
  authDomain: "XXXXXXXXXXXXXXXXXX",
  databaseURL: "XXXXXXXXXXXXXXXXXX",
  projectId: "XXXXXXXXXXXXXXXXXX",
```

Memòria

```
storageBucket: "XXXXXXXXXXXXXXXXXX",
messagingSenderId: "XXXXXXXXXXXXXXXXXX",
appId: "XXXXXXXXXXXXXXXXXX",
measurementId: "XXXXXXXXXXXXXXXXXX",
};

<!--Inicialitza Firebase-->
firebase.initializeApp(firebaseConfig);

<!-- Crea les referència per a l'uatenticació i la data base-->
const auth = firebase.auth();
const db = firebase.database();
</script>
```

Finalment, per tancar l'estructura *head* es defineixen les adreces als diferents codis CSS que donen estil a la pàgina web i que posteriorment es tractaran.

```
<!--Adreces als arxius d'estil CSS-->
<link rel="stylesheet" type="text/css" href="styles/login-form-
style.css">
<link rel="stylesheet" type="text/css" href="styles/nav-bar-
style.css">
<link rel="stylesheet" type="text/css" href="styles/main-style.css">
</head>
```

Seguint amb el codi, es defineix el cos principal que es troba dividit en dues grans seccions: el formulari d'inici de sessió i el contingut de la pàgina principal.

En referència al formulari d'inici de sessió, aquest es mostra quan es carrega la pàgina web per primer cop i la sessió no està iniciada o quan l'usuari decideix tancar-la. La seva funcionalitat és la de gestionar l'accés mitjançant la introducció per part de l'usuari d'un correu electrònic i una contrasenya, prèviament registrats i autoritzats en la base de dades.

La seva estructura, definida amb el nom *container-form*, està formada per un contenidor principal que conté en el seu interior el nom de la pàgina, el formulari amb l'espai per introduir les credencials d'accés i el botó per inicial la sessió.

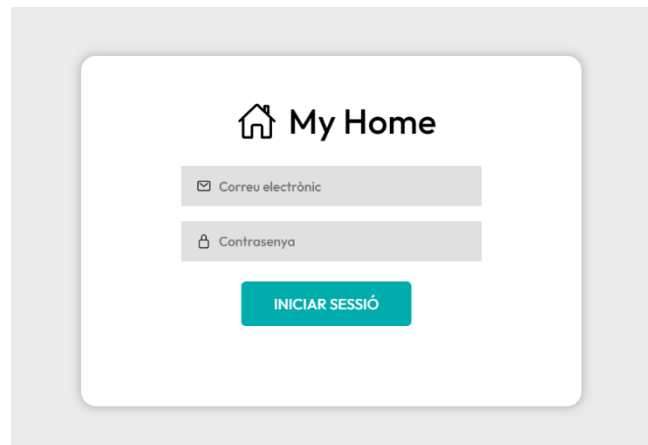


Figura 44. Formulari d'inici de sessió

Memòria

```
<body>

<!--FORMULARI D'INICI DE SESSIÓ-->
<!--Contenidor amb el formulari-->
<div class="container-form" id="container-form">
  <form id="login-form" style="display: none;">
    <!--Nom pàgina web-->
    <div class="nombre-pagina" >
      <ion-icon id="home" name="home-outline"></ion-icon>
      <span>My Home</span>
    </div>
    <!--Contenidor pel correu electrònic-->
    <div class="container-input">
      <ion-icon name="mail-outline"></ion-icon>
      <input type="text" placeholder="Correu electrònic" id="email-
entrada" required>
    </div>
    <!--Contenidor per la contrasenya-->
    <div class="container-input">
      <ion-icon name="lock-closed-outline"></ion-icon>
      <input type="contrasenya" placeholder="Contrasenya"
id="contrasenya-entrada" required>
    </div>
    <!--Botó per iniciar la sessió-->
    <button class="button" id="login-button">INICIAR SESSIÓ</button>
    <p id="missatge-error" style="color:red;"></p>
  </form>
</div>
```

Cal destacar les línies de codi següents en referència al formulari d'inici de sessió:

- `<input type="text" placeholder="Correu electrònic" id="email-entrada" required>` : s'encarrega de crear el camp d'entrada de text per tal que l'usuari introdueixi el seu correu. A més, declara l'entrada com a obligatòria de tal manera que si l'usuari no la introdueix, mostra una indicació de l'obligatorietat.
- `<input type="contrasenya" placeholder="Contrasenya" id="contrasenya-entrada" required>` : s'encarrega de crear el camp d'entrada de text per tal que l'usuari introdueixi la contrasenya. De la mateixa forma que l'anterior, és un camp obligatori.
- `<button class="button" id="login-button">INICIAR SESSIÓ</button>` : crea el botó necessari per iniciar sessió en quant les credencials d'accés s'han introduït.

Memòria

Un cop l'usuari inicia sessió satisfactòriament, es mostra el contingut principal de la pàgina. Aquesta està formada per 3 blocs clarament diferenciats: la barra de navegació lateral, la informació de l'usuari i el cos principal on es mostren les diferents tasques possibles o l'estat dels dispositius.

A continuació es mostra una imatge de la pàgina principal un cop s'ha iniciat la sessió amb la diferenciació de les tres zones mencionades anteriorment. En color verd es delimita la secció corresponent a la barra de navegació lateral, en color vermell la informació de l'usuari i en color blau es mostra la zona principal on es controlen els dispositius.



Figura 45. Seccions de la pàgina principal de la web

Seguint l'ordre del codi, primerament es defineix mitjançant l'identificador *barra-lateral* la barra de navegació que conté les diferents seccions del sistema domòtic i permet identificar i accedir al control dels dispositius de forma més ràpida i clara. En la part superior d'aquesta es mostra el nom i la icona de la pàgina web i seguidament les seccions definides en el projecte que són les següents: llums interior, llums exterior, persianes, climatització, piscina i seguretat.

Cada secció segueix el mateix patró i per aquest motiu tan sols es mostra el codi d'una d'elles.

- `` : s'atribueix el corresponent identificador per permetre la interacció posterior gràcies al codi JavaScript.
- `<ion-icon name="nom icona"></ion-icon>` : insereix una icona representativa de la secció.
- `Títol de la secció` : insereix el nom de la secció.

```
<!--CONTINGUT PÀGINA PRINCIPAL-->
<!--Únicament es mostra quan s'inicia la sessió satisfactòriament-->
<div class="contingut-sign-in" id="contingut-sign-in">
```

```
<!--BARRA NAVEGACIÓ LATERAL-->
<div class="barra-lateral">
```

Memòria

```
<!--Nom i icona de la pàgina-->
<div>
  <div class="nombre-pagina" id="link-home">
    <ion-icon id="home" name="home-outline"></ion-icon>
    <span>My Home</span>
  </div>
</div>

<!--Seccions pàgina-->
<nav class="navegacio">
  <ul>
    <!--Secció llums interiors-->
    <a href="#" id="link-llums-interior">
      <!--Nom i icona-->
      <ion-icon name="bulb-outline"></ion-icon>
      <span>Llums interior</span>
    </a>
```

A sota de les diferents seccions que formen la barra de navegació situada en el lateral, es pot observar l'estructura amb l'identificador *usuari* que conté la informació de l'usuari que ha iniciat la sessió i el botó per tancar-la. Primerament es mostra la seva imatge de perfil i seguidament el seu correu electrònic acompanyat del botó mencionat.

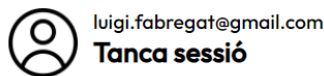


Figura 46. Informació d'usuari i tancament de sessió

```
<div>
  <!--Línia separació entre seccions i dades de l'usuari-->
  <div class="linia"></div>
  <!--SECCIÓ D'AUTENTICACIÓ (correu i tancar sessió)-->
  <div class="usuari">
    
    <div class="info-usuari">
      <!--Mostra el correu electrònic de l'usuari-->
      <!--Mostra el botó per tancar la sessió-->
      <div id="barra-autenticacio" style="display: none;">
        <span class="email" id="info-usuari">USEREMAIL</span>
        <span class="sortir" href="/" id="logout-link">Tanca
sessió</span>
      </div>
    </div>
  </div>
</div>
```

Memòria

Com s'ha pogut observar al llarg del codi, les estructures tipus *span* són les que contenen el text a mostrar. A més, en aquest cas dintre s'hi assigna un identificador ja que el text a mostrar, el correu electrònic de l'usuari, és una variable.

Finalment, el contingut del centre de la pàgina web es troba sota l'estructura *main*. Aquesta part de la web és des d'on es gestionen tots els dispositius i s'observa l'estat del sistema domòtic. Com es pot observar en la figura 43, s'ha dissenyat seguint una estructura en forma de graella de quadres, on cada un d'aquests representa una informació diferent ja sigui un dispositiu, una mesura o un estat.

En primer lloc, en la part superior es mostra de forma permanent el nom de la pàgina i la secció en què es troba l'usuari. Aquesta última, identificada com a estructura *h3* s'actualitza en funció de la navegació que realitza l'usuari per la barra lateral.

```
<!--COS PRINCIPAL DE LA PÀGINA-->
<main>
  <!--Títol de la pàgina i nom de la secció-->
  <h1>MY HOME</h1>
  <h3>Menú principal</h3>
```

A continuació, amb la creació de l'estructura identificada com a *graella*, es creen tots els quadres corresponents als dispositius, la funcionalitat dels quals s'explica a continuació. Cal destacar que tots els quadres tenen una part de codi comuna corresponent a la declaració de l'identificador, del nom d'aquest i de la icona que el representa de forma més clara mitjançant les següents línies de codi.

- `<div class="quadre" id="quadre-nom quadre">` : crea el quadre amb l'identificador assignat per tal de ser tractat en el codi JavaScript.
- `<p><ion-icon name="nom icona "></ion-icon>nom quadre</p>` : mostra una icona representativa i el títol del quadre.

El primer quadre que es mostra en la pàgina principal és el vinculat a l'alarma d'intrusos. Mitjançant l'identificador *quadre-alarma*, aquest conté l'interruptor per activar-la i desactivar-la en funció de les necessitats de l'usuari. També disposa d'un contenidor on en cas de detectar una intrusió quan l'alarma està activada, alerta a l'usuari del perill de forma visual.

En l'estructura identificada com a *candados* s'insereix un candau obert i un tancat de tal forma que entre aquests dos es defineix l'interruptor per activar i desactivar l'alarma.

- `<input type="checkbox" onchange="toggleAlarma();" id="estat-alarma">` : aquesta línia de codi és l'encarregada de crear una entrada de tipus interruptor que crida a la funció *toggleAlarma* quan es realitza un canvi en el seu estat. Aquest funció mencionada s'encarrega d'enviar l'estat de l'alarma (ON/OFF) a la base de dades.

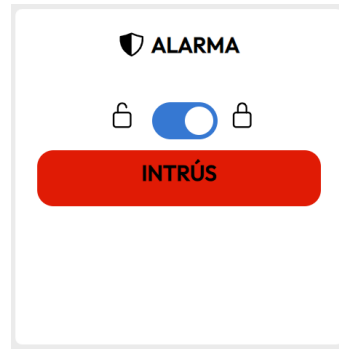


Figura 47. Quadre de control de l'alarma (connectada i informant d'intrusió)

```

<!--Graella amb tots els quadres-->
<div class="graella">
  <!--ALARMA INTRUSOS-->
  <div class="quadre" id="quadre-alarma">
    <!--Nom i icona del quadre-->
    <p><ion-icon name="shield-half-outline"></ion-icon>ALARMA</p>
    <!--Botó per activar/desactivar alarma i icones-->
    <div class="candados">
      <ion-icon name="lock-open-outline"></ion-icon>
      <label class="switch">
        <input type="checkbox" onchange="toggleAlarma();" id="estat-alarma">
        <span class="slider"></span>
      </label>
      <ion-icon name="lock-closed-outline"></ion-icon>
    </div>
    <!--Contenedor d'advertència en cas que hi hagi intrús -->
    <p><span id="pir-estat" style="display: none;"></span></p>
    <div id="pir-estat-container">
      <p id="pir-estat-out"></p>
    </div>
  </div>
</div>

```

El següent quadre és l'encarregat de mostrar la mesura de temperatura i humitat en l'ambient. S'identifica com a *quadre-climatitzacio* i la seva estructura és la següent: a sota del nom i la icona del quadre es mostra la lectura de temperatura, obtinguda a través de la base de dades, expressada en graus centígrads. Seguidament, de la mateixa forma es mostra el percentatge corresponent a la humitat.

- ` °C` : gràcies a la classe *lectura* es mostra en pantalla el valor d'una lectura associada a l'identificador definit. En aquest cas, degut a l'identificador *temp* es mostra el valor d'aquesta variable de temperatura en grau Celsius.

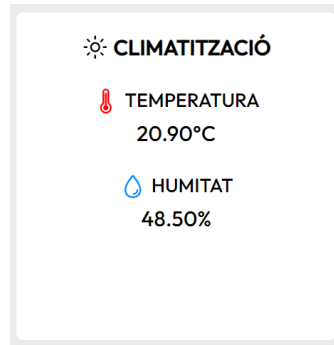


Figura 48. Quadre de visualització de la mesura del clima

```

<!--TEMPERATURA & HUMITAT-->
<div class="quadre" id="quadre-climatitzacio">
  <!--Nom i icona del quadre-->
  <p><ion-icon name="sunny-outline"></ion-icon>CLIMATITZACIÓ</p>
  <!--Lectura valor de temperatura-->
  <a><ion-icon name="thermometer-outline" style="color: red"></ion-
icon>TEMPERATURA</a>
  <p><span class="lectura"><span id="temp"></span> &deg;C</span></p>
  <!--Lectura valor d'humitat-->
  <a><ion-icon name="water-outline" style="color: rgb(0, 140,
255)"></ion-icon>HUMITAT</a>
  <p><span class="lectura"><span id="hum"></span> &percent;</span></p>
</div>

```

A continuació, es mostra el codi referent a l'alarma d'incendis. El quadre per realitzar aquesta funcionalitat es troba dividit en dos ja que la detecció es monitoritza en dues zones principals de l'habitatge: el menjador i la cuina. Per aquest motiu, en el quadre es mostra l'estat dels sensors en funció de la seva mesura.

- <p></p>

<div id="estat-foc-container-menj"> : crea un contenidor amb l'identificador definit per emmagatzemar l'estat del sensor de flama situat en el menjador.
- <p id="estat-foc-out-menj"></p> : mostra el text de sortida, és a dir, si es detecta foc o no.



Figura 49. Quadre de detecció d'incendis

Memòria

```
<!--ALARMA INCENDIS-->
<div class="quadre" id="quadre-incendis">
  <!--Nom i icona del quadre-->
  <p><ion-icon name="flame-outline"></ion-icon>DETECCIÓ D'INCENDIS</p>
  <!--Lectura zona del menjador-->
  <a>ZONA MENJADOR</a>
  <!--Contenedor amb el resultat (canvia en funció de la mesura)-->
  <p><span id="estat-foc-menj" style="display: none;"></span></p>
  <div id="estat-foc-container-menj">
    <p id="estat-foc-out-menj"></p>
  </div>
</div>
```

Destacar que per la zona de la cuina es tracta del mateix codi però intercanviant els identificadors actuals pels definits per monitoritzar la zona.

La següent estructura definida és la que permet realitzar un control sobre els accessos, és a dir, l'encarregada de bloquejar o desbloquejar els panys electrònics situats en les principals entrades de la casa. El codi segueix el mateix patró per ambdues portes tot i que amb els identificadors corresponents per cada una d'elles.

Primerament, es mostra el text que identifica la porta i de la mateixa forma que amb l'alarma, es crea un interruptor per cada accés acompanyat de dos candaus per facilitar la identificació de quan estan obertes o tancades.

- `<ion-icon name="lock-open-outline"></ion-icon>` : conté la icona d'un candau, en aquest cas obert.
- `<input type="checkbox" onchange="togglePorta('porta-exterior');" id="estat-porta-ext">` : crea l'entrada de tipus interruptor i crida la funció *togglePorta* quan hi ha un canvi en el seu estat. Es passa com a paràmetre l'identificador de la porta per poder enviar l'estat a la *database* correctament.



Figura 50. Quadre de control dels accessos a l'habitatge

```
<!--CONTROL DE PORTES-->
<div class="quadre" id="quadre-portes">
  <!--Nom i icona del quadre-->
  <p><ion-icon name="key-outline"></ion-icon>CONTROL D'ACCÉS</p>
```

Memòria

```
<!--Botó per controlar l'estat de la porta de l'exterior-->
<a>PORTA EXTERIOR</a>
<div class="candados">
  <ion-icon name="lock-open-outline"></ion-icon>
  <label class="switch">
    <input type="checkbox" onchange="togglePorta('porta-exterior');"
id="estat-porta-ext">
    <span class="slider"></span>
  </label>
  <ion-icon name="lock-closed-outline"></ion-icon>
</div>
```

A continuació es mostra el quadre referent al control de temperatura de l'interior de l'habitatge. Aquest està format per dos parts: l'ajust de la temperatura desitjada i l'estat de l'aire condicionat.

- `<input type="number" id="select-temp" onchange="actualitzaTemp(this.value)" min="16" max="30" step="0.5" value="20">` : crea una entrada de tipus selector numèric amb un valor mínim de 16, un màxim de 30, un interval de pas de 0,5 i es configura a 20 a l'iniciar la pàgina web. A més, crida a la funció *actualitzaTemp* per tal d'actualitzar el valor en la base de dades en quan s'hi realitza un canvi.
- `<output for="select-temp" id="temp-desitjada" style="display:none;"></output>` : indica que el valor de sortida corresponent a la temperatura desitjada no es mostra en pantalla ja que aquest ja es mostra en l'interior del selector.

En referència a l'estat de l'aire condicionat, es crea un contenidor amb l'estat d'aquest (ON/FF) i per tant, s'implementa el mateix codi que el vist anteriorment però amb els identificadors definits per aquesta funció.

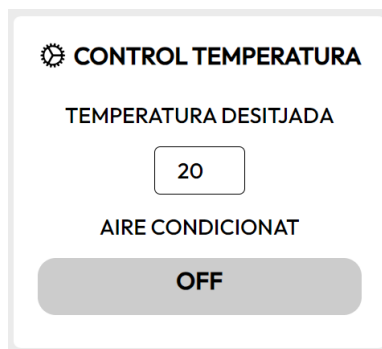


Figura 51. Quadre de control de la temperatura

```
<!--AIRE CONDICONAT-->
<div class="quadre" id="quadre-control-temp">
  <!--Nom i icona del quadre-->
  <p><ion-icon name="cog-outline"></ion-icon>CONTROL TEMPERATURA</p>
  <!--Selector de temperatura desitjada-->
  <a>TEMPERATURA DESITJADA</a>
  <div class="selector-temp">
```

Memòria

```
<input type="number" id="select-temp" onchange="actualitzaTemp(this.value)" min="16" max="30" step="0.5" value="20">

<output for="select-temp" id="temp-desitjada" style="display:none;"></output>
</div>
<!--Contenedor amb l'estat de l'aire condicionat-->
<!--Canvia en funció de la temperatura seleccionada i la mesurada-->
<a>AIRE CONDICIONAT</a>
<p><span id="ac-estat"></span></p>
<div id="ac-estat-container">
  <p id="ac-estat-out"></p>
</div>
</div>
```

En quant als quadres de control de la il·luminació, aquests es troben diferenciats en dos tipus: la manual i l'automàtica. La diferència entre aquests resideix en la forma en què s'engega o s'apaga ja que si es tracta d'il·luminació manual és l'usuari qui controla l'estat d'aquesta però si es tracta d'automàtica dependrà de si els sensors de presència detecten o no un humà.

Tal i com s'especifica en el disseny del sistema domòtic, cal recordar que la zona del menjador, del jardí i de la piscina disposen d'il·luminació manual, mentre que la cuina, el rebedor i el bany es realitza mitjançant el control automàtic.

Tot i aquestes diferències, ambdós tipus comparteixen la forma en què es pot ajustar la intensitat lumínica i el color del LED i per tant, el codi es tracta del mateix però amb els identificadors corresponents a cada zona.

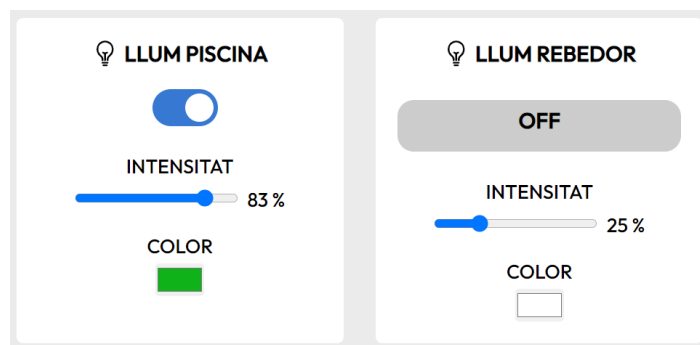


Figura 52. Quadres de control de la il·luminació manual (esquerra) i automàtica (dreta)

El control de l'estat de la il·luminació manual es realitza seguint les mateixes instruccions que les vistes per activar l'alarma, controlar els accessos, etc. Es defineix una entrada de tipus interruptor i es crida la funció *toggleLed* quan hi ha un canvi d'estat en aquest. La funció mencionada passa com a paràmetre l'identificador de la llum en qüestió per tal d'enviar el valor a la base de dades.

Memòria

Per controlar la intensitat lumínica, es defineixen les següents estructures:

- `<input type="range" min="0" max="100" value="25" id="intensitat" onchange="actualitzaLedIntensitat(this.value, 'menjador')">` : es crea una entrada de tipus rang, és a dir, una barra que permet lliscament lateral per ajustar al valor desitjat. S'ha definit aquesta entrada amb un rang de valors possibles d'entre 0 i 100 amb un valor inicial de 25. A més, quan hi ha un canvi en l'estat de l'entrada es crida la funció *actualitzaLedIntensitat* per escriure el valor en la base de dades.
- `<output for="intensitat" id="intensitat-menj">25 %</output>` : mostra el valor de sortida del selector de rang, és a dir, el valor de la intensitat. Està configurat per tal que el valor inicial sigui 25 %.

Finalment, per controlar el color de la tira LED es realitza un procés molt similar però en aquest cas l'entrada és de tipus color.

- `<input type="color" id="selector-color" onchange="actualitzaColor(this.value, 'menjador')">`: es crea una entrada de tipus color en què es pot escollir el color desitjat o bé ajustant els valors RGB per separat de forma manual o bé movent el cursor fins obtenir el color desitjat. Quan es realitza un canvi en el color, es crida la funció *actualitzaColor* per enviar el valor escollit a l'adreça corresponent en la base de dades. Finalment, s'ha declarat el color blanc com a valor predeterminat.
- `<output for="selector-color" id="color-menj" style="display:none;"></output>` : s'oculta el valor de les components RGB de sortida del selector.

```
<!--IL·LUMINACIÓ-->
<!--Llum menjador (manual)-->
<div class="quadre" id="quadre-llum-menjador">
  <!--Nom i icona del quadre-->
  <p><ion-icon name="bulb-outline"></ion-icon>LLUM MENJADOR</p>
  <!--Selector d'estat-->
  <label class="switch">
    <input type="checkbox" onchange="toggleLed('menjador');" id="estat-menj">
    <span class="slider"></span>
  </label>
  <!--Selector d'intensitat-->
  <div class="led-intensitat">
    <a>INTENSITAT</a>
    <input type="range" min="0" max="100" value="25" id="intensitat"
      onchange="actualitzaLedIntensitat(this.value, 'menjador')">
    <output for="intensitat" id="intensitat-menj">25 %</output>
  </div>
```

Memòria

```
<!--Selector de color-->

<div class="selector-color">
  <a>COLOR</a>
  <input type="color" id="selector-color"
onchange="actualitzaColor(this.value, 'menjador')" >
  <output for="selector-color" id="color-menj" style="display:
none;"></output>
</div>
</div>
```

En referència a les zones en què la il·luminació és automàtica l'única part diferent del codi és aquella on es mostra si està funcionant o no. Més concretament, es realitza de la mateixa forma en què es mostra l'estat de l'aire condicionat o del detector d'incendis: mitjançant un contenidor que canvia en funció de la mesura.

```
<!--Llum rebedor (automàtica)-->
<div class="quadre" id="quadre-llum-rebedor">
  <!--Nom i icona del quadre-->
  <p><ion-icon name="bulb-outline"></ion-icon>LLUM REBEDOR</p>
  <!--Estat de la llum en funció de la lectura del sensor PIR-->
  <p><span id="pir-estat-rebedor" style="display: none;"></span></p>
  <div id="pir-estat-container-reb">
    <p id="pir-estat-out-reb"></p>
  </div>
```

Per tal de gestionar el motor de la depuradora, l'usuari el pot posar en funcionament i parar-lo de forma manual mitjançant el quadre que hi té assignat. Aquest és molt senzill ja que tan sols disposa d'un interruptor que permet donar sortida les necessitats de l'usuari.

El codi implementat per crear aquest interruptor és el mateix que el vist anteriorment per diferents quadres però en aquest cas crida la funció *toggleDepuradora* quan hi ha un canvi d'estat en el botó.

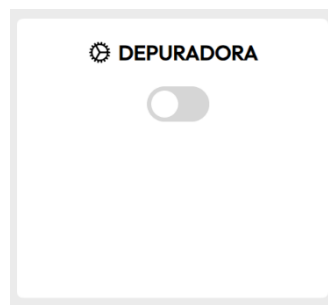


Figura 53. Quadre de control de la depuradora

```
<!--DEPURADORA-->
<div class="quadre" id="quadre-depuradora">
  <!--Nom i icona del quadre-->
  <p><ion-icon name="cog-outline"></ion-icon>DEPURADORA</p>
```

Memòria

```
<!--Botó per controlar l'estat de la depuradora-->
<label class="switch">
  <input type="checkbox" onchange="toggleDepuradora();" id="estat-
depuradora">
  <span class="slider"></span>
</label>
</div>
```

L'últim dels quadres associats als dispositius domòtics és el de control de les persianes. Tal i com es menciona en les especificacions de disseny, aquestes poden controlar-se de forma automàtica en funció de la intensitat lumínica solar o de forma manual en funció de les necessitats de l'usuari.

En referència al quadre dissenyat, en primer lloc s'hi troba un selector per tal que l'usuari pugui activar o desactivar el mode automàtic. La programació d'aquest segueix el mateix patró que en els interruptors anteriors i en aquest cas crida la funció *toggleModePersiana* en quan hi ha un canvi d'estat.

Per controlar les persianes de forma manual, s'han creat un total de 3 botons corresponents a les accions que es poden realitzar: pujar, parar o baixar la persiana. El codi implementat per crear els botons és el mostrat a continuació.

- `<button onclick="manualPersiana('acció', 'nom persiana')"><ion-icon name="chevron-up-circle-outline"></ion-icon>Acció</button>` : crea el botó amb el nom de l'acció i una icona representava en el seu interior i crida la funció *manualPersiana* quan l'usuari el selecciona. Aquesta funció té com a paràmetre l'acció a realitzar i l'identificador de la persiana.



Figura 54. Quadre de control de la persiana

```
<!--PERSIANES-->
<!--Persiana menjador-->
<div class="quadre" id="quadre-persiana-menjador">
  <!--Nom i icona del quadre-->
  <p><ion-icon name="reorder-four-outline"></ion-icon>PERSIANA MENJADOR</p>
  <!--Botó per escollir el mode de funcionament-->
  <div class="mode-func">
    <a>Mode automàtic</a>
```

Memòria

```
<label class="switch">
  <input type="checkbox" onchange="toggleModePersiana('-menjador');"
id="mode-persiana-menj">
  <span class="slider"></span>
</label>
</div>
<!--Botons amb les diferents accions-->
<div class="persiana-botons">
  <button onclick="manualPersiana('pujar', '-menjador')"><ion-icon
name="chevron-up-circle-outline"></ion-icon>Pujar</button>
  <button onclick="manualPersiana('parar', '-menjador')"><ion-icon
name="close-circle-outline"></ion-icon>Parar</button>
  <button onclick="manualPersiana('baixar', '-menjador')"><ion-icon
name="chevron-down-circle-outline"></ion-icon>Baixar</button>
</div>
</div>
```

Per acabar els elements situats en el cos principal de la pàgina web, es mostra el codi dissenyat per mostrar els gràfics de temperatura i d'humitat. De la mateixa forma que els diferents dispositius, els gràfics es troben a l'interior d'una estructura definida com quadre, en què a l'interior es mostra el nom del gràfic, el nom dels eixos amb les unitats que amb què es realitza i el propi gràfic.

- `<div id="grafic-temp" class="grafic-container"></div>` : crea un gràfic per mitjà de l'identificador definit, el funcionament del qual es s'explica en el codi JavaScript.

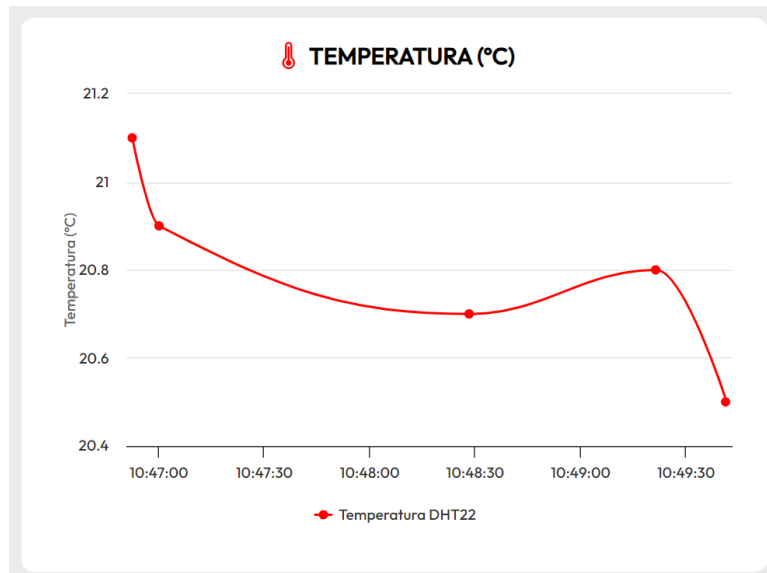


Figura 55. Quadre amb el gràfic de temperatura

Memòria

```
<!--GRÀFICS CLIMA-->
<!--Temperatura-->
<div class="graella">
  <div class="quadre" id="grafic-temperatura">
    <!--Nom i icona del gràfic-->
    <p><ion-icon name="thermometer-outline" style="color: red"></ion-
icon>TEMPERATURA (°C)</p>
    <div id="grafic-temp" class="grafic-container"></div>
  </div>
</div>
```

Amb aquest codi referent als gràfics, el document HTML tanca la part associada a l'estructura main del programa. Finalment, per tancar el codi, s'insereixen les adreces a la pàgina web d'on s'extreuen les icones [26] i als diferents arxius JavaScript, que són els encarregats d'escriure i rebre les mesures i els diferents estats de la base de dades en temps real.

```
<!--Adreces a la pàgina web de les icones-->
<script type="module" src="https://unpkg.com/ionicons@7.1.0/dist/
ionicons/ionicons.esm.js"></script>
<script nomodule src="https://unpkg.com/ionicons@7.1.0/dist/ionicons/
ionicons.js"></script>
<!--Adreces als arxius JavaScript-->
<script src="scripts/index.js"></script>
<script src="scripts/auth.js"></script>
<script src="scripts/charts-definitions.js"></script>
</body>
</html>
```

1.7.5.3.2. CSS

En aquesta secció es tracta el codi CSS encarregat de donar estil i personalització a tots els elements que componen la pàgina web. Pel desenvolupament d'aquest projecte, s'han definit 3 arxius d'estil diferents en funció de la zona de la qual s'encarreguen: estil del formulari d'inici de sessió, estil de la barra de navegació lateral i estil del cos principal de la web.

Tal i com s'ha definit en el codi HTML, l'estructura de la pàgina està dividida per un seguit de contenidors on es mostra una determinada informació. Per aquest motiu, els estils s'encarreguen de posicionar i donar la forma desitjada a aquests contenidors i de gestionar com se situen els elements interiors (mesura, alineació, marges...).

Cal destacar que la font utilitzada en tot el text que forma la pàgina s'ha obtingut a través de Google Fonts i es pot identificar amb el títol "Outfit" [27] .

El funcionament del codi CSS segueix una forma jeràrquica, de tal forma que primer es dona estil als elements de nivell superior i seguidament s'accedeix als seus elements interiors per personalitzar-los. Per accedir als diferents contenidors d'informació s'utilitza l'identificador definit en el codi HTML a l'hora de crear les estructures tipus <div> amb un punt al davant.

Per exemple, per accedir a l'estil del contenidor d'inici de sessió i posteriorment al contenidor amb el nom de la pàgina, el codi és el següent:

```
/* Estil del contenidor del formulari d'inici de sessió */
.container-form{
    width: 500px;           /* Alçada */
    height: 350px;         /* Amplada */
    position: fixed;       /* Posicionament en la pàgina */
    margin-top: 140px;     /* Marges superior i esquerre */
    margin-left: 400px;
    display: flex;         /* Posicionament dels elements interiors */
    justify-content: center; /* Justificat al centre */
    overflow: hidden;     /* Oculta tot allò que sobresurti */
}
/* Estil del nom de la pàgina situat a l'interior del formulari */
.container-form .nom-pagina{
    height: 50px;          /* Alçada */
    color: var(--color-text); /* Color del text */
    display: flex;        /* Posicionament elements interiors */
    align-items: center;  /* Alineat al centre */
    justify-content: center; /* Justificat al centre */
}
```

D'aquesta forma, s'accedeix als diferents elements que formen part d'un conjunt de forma ordenada i de major a menor nivell.

Per consultar el codi complet referent als estils CSS, visita [Codi d'estil CSS del formulari d'inici de sessió](#), [Codi d'estil CSS de la barra lateral de navegació](#) i [Codi d'estil CSS del cos principal de la pàgina](#).

1.7.5.3.3. JavaScript

Tal i com es menciona repetides vegades durant el desenvolupament del projecte, el codi JavaScript és l'encarregat de donar vida i dinamisme als diferents elements que formen la pàgina web. A més, la incorporació d'aquest codi en el projecte és fonamental ja que s'encarrega de rebre les dades de les diferents adreces de la data base i d'escriure en aquestes qualsevol canvi realitzat per part de l'usuari.

El projecte disposa de tres codis JavaScript: el codi d'autenticació, el codi que defineix i crea els gràfics i el codi que conté les funcions que interactuen amb la base de dades.

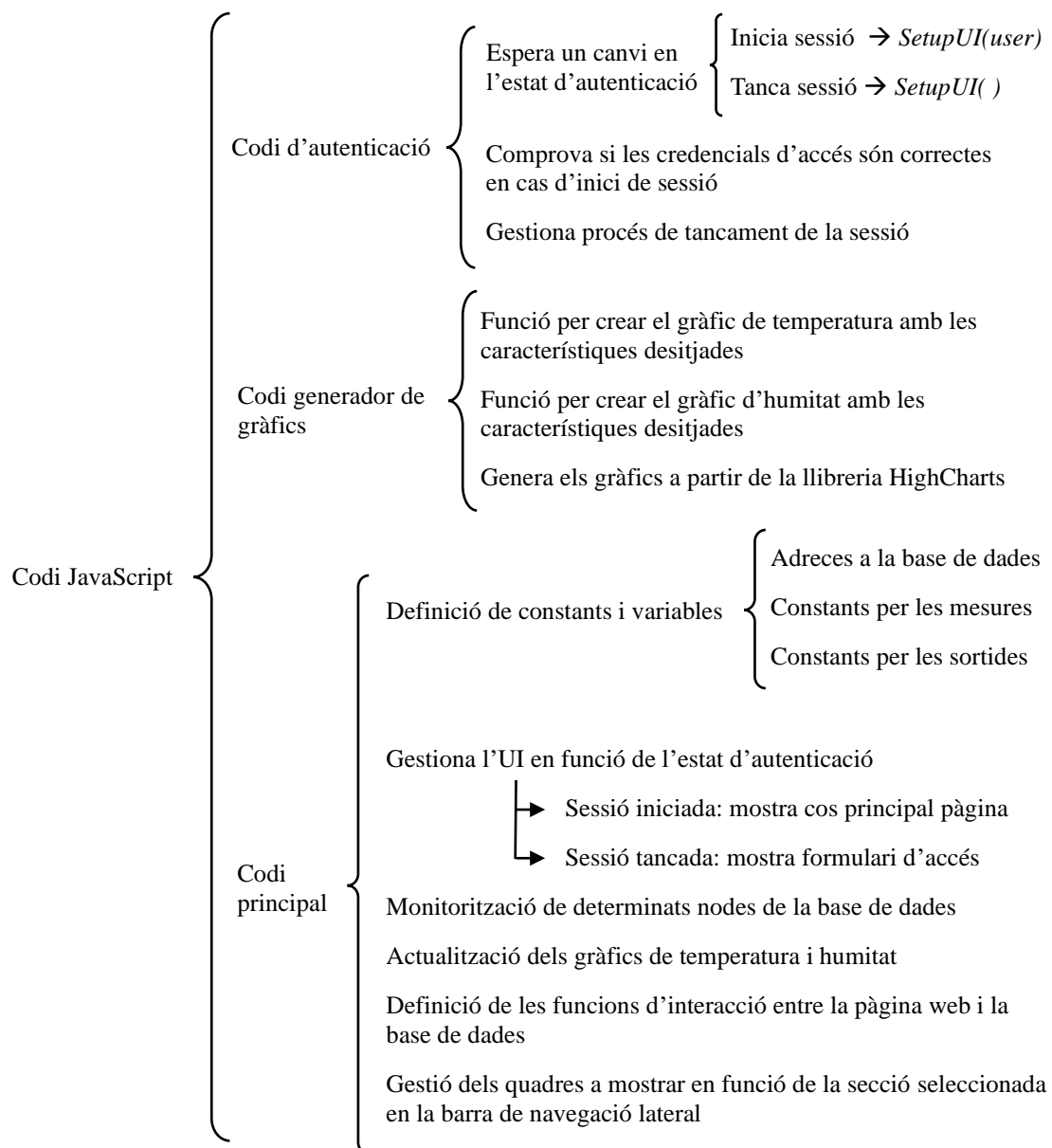


Figura 56. Diagrama estructural del codi JavaScript

- **Codi d'autenticació:** s'encarrega de gestionar la pàgina quan hi ha un canvi en l'estat de l'autenticació, és a dir, quan un usuari inicia o tanca sessió.

El codi està format per 3 parts principals: la primera tan sols s'executa quan hi ha un canvi en l'estat i en ella es configura la interfície d'usuari mitjançant la funció *setupUI*. Aquesta es troba declarada en el darrer arxiu JavaScript i la seva funció és la de gestionar quines parts de la pàgina web es mostren en funció de si la sessió està iniciada o no.

```
// Espera un canvi en l'estat de l'autenticació
auth.onAuthStateChanged(user => {
  // Si inicia sessió
  if (user) {
    // Mostra a la consola que s'ha iniciat sessió i l'usuari
    console.log("user logged in");
    console.log(user);
    // Configura la interfície d'usuari segons l'usuari que ha iniciat
    setupUI(user);
    // Obté l'ID de l'usuari i el mostra en consola
    var uid = user.uid;
    console.log(uid);
  } else {
    // Si tanca sessió, mostra a la consola que l'usuari ha tancat sessió
    console.log("user logged out");
    // Configura la interfície d'usuari per a l'estat de tancament
    setupUI();
  }
});
```

La següent part del codi s'executa quan s'envia el formulari d'inici de sessió, és a dir, quan l'usuari ha escrit les credencials d'accés i prem el botó per accedir a la web. D'aquesta forma, gestiona si les credencials inserides són correctes o no i en cas que no ho siguin, mostra un missatge en pantalla.

```
// Inici de sessió
// Selecciona el formulari d'inici de sessió amb l'id #login-form
const loginForm = document.querySelector('#login-form');
// Escolta l'esdeveniment d'enviament del formulari
loginForm.addEventListener('submit', (e) => {
  // Evita que la pàgina s'actualitzi
  e.preventDefault();
  // Obté la informació de l'usuari (correu i contrasenya) del formulari
  const email = loginForm['email-entrada'].value;
  const contrasenya = loginForm['contrasenya-entrada'].value;
```

Memòria

```
// Inicia sessió amb el correu i la contrasenya proporcionats
auth.signInWithEmailAndPassword(email, contrasenya).then((cred) => {
  // Si les credencials són correctes, restableix el formulari
  loginForm.reset();
  // Mostra el correu en consola
  console.log(email);
})
// Detecta errors d'inici de sessió i mostra el missatge
.catch((error) =>{
  const errorMessage = error.message;
  console.log(errorMessage);
  document.getElementById("missatge-error").innerHTML = "Credencials
d'accés incorrectes";
});
});
```

La darrera part d'aquest primer arxiu JavaScript s'encarrega de gestionar el procés de tancament de sessió quan l'usuari prem el botó destinat a aquesta funció.

```
// Tanca la sessió
// Selecciona l'enllaç de sortida a través de l'id
const logout = document.querySelector('#logout-link');
// Escolta l'esdeveniment de tancament de sessió
logout.addEventListener('click', (e) => {
  // Evita que la pàgina s'actualitzi
  e.preventDefault();
  // Tanca la sessió de Firebase
  auth.signOut();
});
```

- **Codi generador de gràfics:** el codi mostrat a continuació s'encarrega de crear els gràfics de temperatura i humitat mitjançant la llibreria HighCharts.

```
// S'afegeix un esdeveniment de càrrega a la finestra
// que crida a la funció onload quan la pàgina es carrega completament
window.addEventListener('load', onload);

// Funció onload on es criden les funcions per crear els gràfics
function onload(event) {
  // Crea el gràfic de temperatura
  graficT = creaGraficTemp();
  // Crea el gràfic d'humitat
  graficH = creaGraficHum();
}
```

Memòria

```
// Funció per crear el gràfic de temperatura
function creaGraficTemp() {
  // Es crea el gràfic utilitzant la biblioteca HighCharts
  var grafic = new Highcharts.Chart({
    chart: {
      // S'especifica l'element HTML on es carrega el gràfic
      renderTo: 'grafic-temp',
      // S'especifica el tipus de gràfic
      type: 'spline'
    },
    // Configura la sèrie de dades de la temperatura
    series: [{
      // Nom de la sèrie de dades
      name: 'Temperatura DHT22'
    }],
    // No s'especifica el nom del gràfic
    title: {
      text: undefined
    },
    plotOptions: {
      line: {
        // Es deshabilita l'animació de la línia
        animation: false,
        // S'habiliten les etiquetes de les dades
        dataLabels: {
          enabled: true
        }
      },
    },
    series: {
      // Color de la línia
      color: '#ff0000'
    }
  },
  // Valors de l'eix X (data i hora)
  xAxis: {
    type: 'datetime',
    // Defineix el format de l'hora en HH:MM:SS
    dateTimeLabelFormats: { second: '%H:%M:%S' }
  },
  // Defineix el títol de l'eix Y
  yAxis: {
    title: {
      text: 'Temperatura (°C)'
    }
  },
}
```

Memòria

```
// Deshabilita els crèdits de la llibreria HighCharts
credits: {
  enabled: false
}
});
// La funció retorna el gràfic creat
return grafic;
}
```

Com es pot observar a partir del codi, la llibreria HighCharts ofereix una gran possibilitat de personalització de les gràfiques que es creen a través de la modificació d'un gran nombre de paràmetres diferents.

- **Codi principal:** aquest és l'encarregat de gestionar la interacció entre les mesures de la base de dades i les peticions que realitza l'usuari a través dels elements de la pàgina web com els botons, interruptors, selectors d'intensitat, etc.

En aquest arxiu es defineixen totes les constants i variables per emmagatzemar els diferents estats i valors dels dispositius i elements de la pàgina web, totes les adreces d'accés a la base de dades i totes les funcions que llegeixen o escriuen valors a Firebase i que s'han vist anteriorment en el codi HTML.

Amb l'objectiu de tenir una visió global d'aquest codi i tal i com es mostra en el diagrama de la Figura 56, es presenta l'estructura que segueix:

1. Declaració de constants i variables associades a les mesures i estats dels dispositius.
2. Gestiona la interfície d'usuari en funció de l'estat d'autenticació de la sessió.
 - 2.1. Si la sessió està iniciada: mostra elements de la pàgina principal, declara les adreces a la base de dades i monitoritza dades dels sensors.
 - 2.2. Si la sessió està tancada: amaga elements de la pàgina principal i mostra el formulari.
3. Declaració de les funcions d'interacció amb la base de dades.
4. Gestiona els elements que es mostren a la pàgina principal en funció de la secció seleccionada en la barra de navegació lateral.

A continuació, es mostra i detalla aquest darrer codi JavaScript.

La primera tasca que es realitza és la de definir les diferents variables i constants en funció dels identificadors especificats en el codi HTML. Primerament, es declaren els elements que es mostren en funció de l'estat de l'autenticació, és a dir, en funció si la sessió es troba iniciada o no.

Per realitzar aquesta tasca, s'utilitza el següent mètode:

- `const nom_constant = document.querySelector ('#identificador');` es tracta d'un mètode en JavaScript utilitzat per seleccionar i retornar el primer element del document HTML que coincideixi amb l'identificador seleccionat, que ha de coincidir amb l'especificat anteriorment.

Memòria

Com es pot observar en les línies de codi següents, es declaren mitjançant el mètode `querySelector` elements com el formulari d'inici de sessió, la informació de l'usuari, o els continguts mostrats amb la sessió iniciada.

```
// Elements per a la configuració d'inici/tancament de sessió
const loginElement = document.querySelector('#login-form');
const containerElement = document.querySelector('#container-form');
const contingutElement = document.querySelector("#contingut-sign-in");
const infoUserElement = document.querySelector('#info-usuari');
const authBarElement = document.querySelector("#barra-autenticacio");
```

Seguidament, es declaren les constants associades als dispositius, sensors i elements de la pàgina web (botons, selectors...) i el nom de les variables que contenen les adreces a la data base, l'assignació de les quals es realitza posteriorment en el codi. Aquesta tasca s'ha realitzat agrupada per blocs en funció del dispositiu al que pertanyen i mitjançant el mètode següent:

- `const nom_element = document.getElementById ('#identificador')`: es tracta d'un mètode en JavaScript utilitzat per seleccionar l'element del document HTML que coincideixi amb l'identificador especificat.

A l'hora de realitzar la declaració, es tracta en tot moment d'assignar un nom representatiu de l'element que representa per tal d'identificar posteriorment amb major facilitat a l'hora de crear les funcions.

A continuació es mostra un exemple del codi que declara els elements per obtenir els valors de temperatura i humitat. Per veure la resta de codi que realitza aquesta tasca consultar [Codi JavaScript principal](#).

```
// Elements per a les lectures del sensor DHT22
var dbPathClima
const tempElement = document.getElementById("temp");
const humElement = document.getElementById("hum");
```

Seguidament de la declaració de totes les constants i variables, es troba el codi encarregat de gestionar la interfície d'usuari en funció de l'estat d'autenticació. Com és evident, no es mostren els mateixos elements en funció de si la sessió ha estat iniciada o no i és en aquesta part on es gestiona.

- `nomElement.style.display = 'none'` : no mostra en pantalla el contingut associat a l'element especificat.
- `nomElement.style.display = 'block'` : fixa en la pantalla el contingut associat a l'element especificat.

Mitjançant aquests dos comandaments es gestiona el contingut que es mostra en pantalla. Més concretament, si la sessió s'ha iniciat satisfactòriament s'amaga el contingut relacionat amb el formulari d'inici i es mostra la pàgina principal.

Memòria

```
// Gestiona l'UI (Interfície d'usuari) en l'inici/tancament de sessió
const setupUI = (user) => {

  // L'usuari inicia sessió
  if (user) {
    // Intercanvia elements UI (amaga formulari, mostravpàgina principal)
    loginElement.style.display = 'none';
    containerElement.style.display = 'none';
    contingutElement.style.display = 'block';
    authBarElement.style.display = 'block';
    infoUserElement.style.display = 'block';
    infoUserElement.innerHTML = user.email;
  }
}
```

Seguidament, es declaren totes les adreces als diferents nodes de la base de dades. Cal destacar que aquests són els nodes general i que contenen en el seu interior nodes més específics ja que per exemple, dintre de la climatització es troba el node amb el valor de la temperatura, la humitat, la temperatura desitjada, etc.

```
// Adreces a la database
var dbPathFire = 'UsersData/' + uid.toString() + '/alarma-incendis';
var dbPathClima = 'UsersData/' + uid.toString() + '/climatitzacio';
var dbPathLed = 'UsersData/' + uid.toString() + '/llums';
var dbPathPersiana = 'UserData/' + uid.toString() + '/persianes/persiana';
var dbPathAlarma = 'UsersData/' + uid.toString() + '/alarma';
var dbPathPortes = 'UsersData/' + uid.toString() + '/control-acces/porta';
var dbPathDepuradora = 'UsersData/' + uid.toString() + '/depuradora';
```

A continuació es defineixen les referències als diferents nodes específics de la *database*. Aquestes són necessàries per apuntar a ubicacions concretes i detectar canvis en temps real. Per aquest motiu, tan sols es creen en aquells nodes que contenen mesures a monitoritzar i detectar els seus canvis com per exemple els valors de sortida del sensor de flama o de presència.

```
// Referències a la database (necessari per dades que s'han de llegir)
var dbRefTemp = firebase.database().ref(dbPathClima + '/temperatura');
var dbRefHum = firebase.database().ref(dbPathClima + '/humitat');
var dbRefFoc_Menj = firebase.database().ref(dbPathFire + '/menjador');
var dbRefFoc_Cuina = firebase.database().ref(dbPathFire + '/cuina');
var dbRefLed_Rebedor = firebase.database().ref(dbPathLed + '/llum-
rebedor/estat');
var dbRefLed_Cuina = firebase.database().ref(dbPathLed + '/llum-
cuina/estat');
var dbRefLed_WC = firebase.database().ref(dbPathLed + '/llum-
wc/estat');
var dbRefTemp_Desitj = firebase.database().ref(dbPathClima + '/temp-
desitjada');
var dbRefSensorLlum_Menj = firebase.database().ref(dbPathPersiana + '-
menj/intensitat-llum');
```

Memòria

```
dbRefSensorLlum_Hab = firebase.database().ref(dbPathPersiana + '-hab/intensitat-llum');
dbRefPIR_Rebedor = firebase.database().ref(dbPathAlarma + '/pir-rebedor');
dbRefPIR_Cuina = firebase.database().ref(dbPathAlarma + '/pir-cuina');
```

Un cop definides totes les adreces i referències als diferents nodes de la base de dades, es realitza la monitorització dels dispositius que així ho requereixen. La seqüència de codi mostrada a continuació té com a objectiu escoltar qualsevol canvi sofert en la referència especificada i actuar envers el valor que s'obté.

- `NomReferència.on ('value', snap => { })` : estableix un oient d'esdeveniments en la referència especificada. L'esdeveniment 'value' es dispara cada cop que la dada de la ubicació canvia i s'executa la funció de devolució de trucada. La variable `snap` conté un objecte que representa les dades de la ubicació seleccionada.
- `NomElement.innerText = snap.val()` : s'actualitza el contingut de l'element HTML especificat amb el valor obtingut de la base de dades.

Aquesta tasca es realitza de control d'esdeveniments es realitza per les funcions mostrades a continuació.

- **Monitorització del clima:** controla els canvis en el valor de la temperatura i la humitat i actualitza el gràfic amb el valor més recent

```
// Actualitza la pàgina amb la lectura de temperatura més recent
// Actualitza el gràfic de la temperatura
dbRefTemp.on('value', snap => {
  // Obté la mesura de la database amb dos decimals
  tempElement.innerText = snap.val().toFixed(2);
  // Obté la variable temporal del gràfic
  x = (new Date()).getTime(),
  // Converteix temperatura a tipus float
  y = parseFloat(snap.val().toFixed(2));
  // Configura els valors en el gràfic
  if(graficT.series[0].data.length > 40) {
    graficT.series[0].addPoint([x, y], true, true, true);
  } else {
    graficT.series[0].addPoint([x, y], true, false, true);
  }
});
```

- **Monitorització de l'aparició d'incendis:** controla qualsevol esdeveniment de canvi en la lectura del sensor de flama. Si el valor obtingut a través de la base de dades és igual a 1, significat així que el sensor detecta foc, es canvia l'estat del contenidor amb el resultat per alertar a l'usuari que es detecta un incendi.

Per canviar el contenidor, s'elimina l'estil assignat a la no detecció de foc (contenidor de color verd) i es mostra l'estil definit per la detecció (contenidor de color vermell). Aquesta acció es realitza per mitjà del comandament `.classList.add` si es vol mostrar o `.classList.remove` si es vol amagar un determinat estil prèviament definit en el codi CSS.

Destacar que cada sensor de flama utilitzat el mateix codi però amb els identificadors corresponents definits anteriorment per cada zona.

```
// Actualitza la pàgina segons lectura del sensor d'incendis situat
al menjador
dbRefFoc_Menj.on('value', snap => {
  // Obté la mesura de la database
  focEstat_Menj.innerText = snap.val();
  // Converteix la mesura en un valor de tipus int
  focEstat = parseInt(focEstat_Menj.textContent);

  if (focEstat === 1) {
    // Si es detecta foc intercanvia l'estat i mostra el resultat
    focEstatContainer_Menj.classList.remove('no-foc');
    focEstatContainer_Menj.classList.add('foc-detectat');
    focEstatOut_Menj.textContent = "FOC DETECTAT!";
  } else {
    // Si NO es detecta, intercanvia l'estat i escriu el resultat
    focEstatContainer_Menj.classList.remove('foc-detectat');
    focEstatContainer_Menj.classList.add('no-foc');
    focEstatOut_Menj.textContent = "NO HI HA FOC";
  }
});
```

- **Monitorització de detecció de presència:** pel cas en què es monitoritza el sensor de presència, es realitza el mateix procediment per la detecció d'un esdeveniment en la referència especificada. En cas que el sensor PIR detecti una presència i per tant s'hagi engegat la llum de la respectiva zona, s'indica en la pàgina web l'estat de la il·luminació mitjançant l'intercanvi d'estats del corresponent contenidor.

```
// Actualitza la pàgina amb la lectura del sensor de presència situat
al rebedor
dbRefLed_Rebedor.on('value', snap => {
  // Obté la mesura de la database
  pirEstat_Rebedor.innerText = snap.val();
  // Converteix la mesura en un valor de tipus int
  pirEstat = parseInt(pirEstat_Rebedor.textContent);
```

Memòria

```
if (pirEstat === 1) {
  // Si es detecta intercanvia l'estat i escriu el resultat
  pirEstatContainer_Rebedor.classList.remove('off');
  pirEstatContainer_Rebedor.classList.add('on');
  pirEstatOut_Rebedor.textContent = "ON";
} else {
  // Si NO es detecta, intercanvia l'estat i escriu el resultat
  pirEstatContainer_Rebedor.classList.remove('on');
  pirEstatContainer_Rebedor.classList.add('off');
  pirEstatOut_Rebedor.textContent = "OFF";
}
});
```

Finalment, per acabar la gestió de la interfície d'usuari es tracta el cas en què la sessió està tancada. En aquesta situació es deixa de mostrar el contingut principal i es fixa en la pantalla el formulari.

```
else {
  // L'usuari tanca la sessió
  // Amaga elements pàgina principal i mostra el formulari
  loginElement.style.display = 'block';
  containerElement.style.display = 'block';
  authBarElement.style.display = 'none';
  infoUserElement.style.display = 'none';
  contingutElement.style.display = 'none';
}
}
```

Un cop vista aquesta segona part del codi, la següent a detallar es tracta de la que conté totes les funcions definides i utilitzades en el codi HTML per comunicar els elements de la pàgina web amb Firebase.

Les funcions més utilitzades són les següents:

- `console.log (Text desitjat)`: funció utilitzada per escriure i mostrar en la consola el text desitjat. Generalment, els missatges mostrats contenen canvis d'estat dels diferents elements com botons, interruptors...
- `firebase.database().ref(AdreçaEspecífica).set(Dada)`: funció utilitzada per establir un valor específic en una ubicació determinada de la base de dades en temps real. Com a paràmetres es passa en primer lloc l'adreça específica al node on es vol escriure i seguidament la dada, valor o cadena de caràcters desitjada.
- `NomElement.checked`: funció que retorna el valor *true* si l'element que es comprova es troba seleccionat. S'utilitza per identificar l'estat dels diferents botons i interruptors de la pàgina web.

Memòria

A continuació, es mostra cada una d'aquestes funcions dissenyades explicant la seva tasca dintre del projecte i destacant els elements més significatius pel seu correcte desenvolupament.

- Funció que escriu a la *database* l'acció seleccionada per l'usuari per a una determinada persiana quan està en mode manual. Es passa com a paràmetre el nom de l'acció i l'última part de l'adreça de la persiana desitjada a mode d'identificador.

Exemple: manualPersiana ('pujar', '-menjador')

```
function manualPersiana(accio, persiana_id) {
  // Mostra en la consola el canvi realitzat
  console.log("Toggle Persiana " + persiana_id + " - Accio: " + accio);
  // Escriu en l'adreça corresponent l'acció seleccionada
  firebase.database().ref(dbPathPersiana + persiana_id + '/accio-
manual').set(accio);
}
```

- Funció que escriu a la *database* el mode de funcionament de la persiana (manual o automàtic) en funció de si l'interruptor de la pagina web està seleccionat o no. Es passa com a paràmetre l'última part de l'adreça de la persiana desitjada a mode d'identificador.

Exemple: toggleModePersiana ('-hab')

```
function toggleModePersiana(persiana_id){
  // Persiana menjador
  if (persiana_id === '-menjador'){
    // Interruptor està seleccionat
    if (mode_Menj.checked)
    {
      // Mostra en la consola el mode de funcionament
      console.log("Mode de funcionament persiana menjador: AUTO");
      // Escriu en l'adreça corresponent el mode de funcionament AUTO
      firebase.database().ref(dbPathPersiana + persiana_id + '/mode-
func').set("AUTO");
    }
    // Si no està seleccionat
    else{
      // Mostra en la consola el mode de funcionament
      console.log("Mode de funcionament persiana menjador: MANUAL");
      // Escriu en l'adreça corresponent el mode de funcionament MANUAL
      firebase.database().ref(dbPathPersiana + persiana_id + '/mode-
func').set("MANUAL");
    }
  }
  // Compara el paràmetre de la funció amb la resta persianes
```

Es repeteix el codi per la persiana de l'habitació amb les adreces actualitzades i utilitzant l'identificador "-hab".

Memòria

- Funció que escriu a la base de dades l'estat (ON/OFF) d'una determinada tira LED de control manual en funció de l'estat de l'interruptor corresponent. Es passa com a paràmetre l'identificador assignat a la llum desitjada.

Exemple: toggleLed ('menjador')

```
function toggleLed(led_id) {
  if (led_id === 'menjador'){
    // Si el botó està seleccionat
    if (estat_Menj.checked)
    {
      // Mostra en la consola l'estat ON del LED
      console.log("Toggle LED menjador -> LED ON");
      // Escriu en l'adreça corresponent l'estat ON del LED
      firebase.database().ref(dbPathLed + '/llum-
menjador/estat').set("ON");
    }
    // Si no està seleccionat
    else{
      // Mostra en la consola l'estat OFF del LED
      console.log("Toggle LED menjador -> LED OFF");
      // Escriu en l'adreça corresponent l'estat OFF del LED
      firebase.database().ref(dbPathLed + '/llum-
menjador/estat').set("OFF");
    }
  }
  // Compara el paràmetre de la funció amb la resta de llums possibles
```

Dintre la funció es realitza el mateix procediment per comprovar l'estat de la resta de llums manuals com són la del jardí i la de la piscina en funció de l'identificador passat com a paràmetre.

- Funció que escriu el la intensitat seleccionada d'un determinat LED. Es passa com a paràmetre el valor de la intensitat obtinguda a partir de l'entrada de tipus selector i l'identificador de la llum corresponent.

Exemple: actualitzaLedIntensitat(this.value, 'menjador')

```
function actualitzaLedIntensitat(valor, led_id) {
  if (led_id === 'menjador'){
    // Actualitza el valor en la pàgina web
    intensitat_Menj.textContent = valor + ' %';
    // Mostra en consola l'intensitat seleccionada
    console.log("Set LED menjador intensitat: " + valor);
    // Escriu en l'adreça corresponen el valor (tipus int) seleccionat
    firebase.database().ref(dbPathLed + '/llum-menjador/intensitat')
.set(parseInt(valor));
  }
  // Compara el 2n paràmetre de la funció amb la resta de llums possibles
```

Memòria

Cal recordar que el control de la intensitat lumínica s’aplica a totes les zones amb il·luminació per igual, i per tant, dintre la funció es compara l’identificador d’entrada amb totes les llums possibles ja siguin manuals o automàtiques.

- Funció que escriu a la base de dades les components RGB del color seleccionat per un determinat LED. Es passa com a paràmetre el valor RGB obtingut a partir de l’entrada de tipus selector de color i l’identificador de la llum corresponent.

Previ a l’enviament de les dades, es realitza un tractament del valor de sortida del selector per convertir-lo del format hexadecimal a un format RGB amb les components separades en variables independents.

Exemple: `actualitzaColor (this.value, ‘menjador’)`

```
function actualitzaColor(color, led_id) {
  // Converteix el valor del color d'hexadecimal a RGB
  // Emmagatzema com una string amb els elements separats per una coma
  var rgb = hexToRgb(color).split(", ");
  // Obté les components RGB per separat i les converteix a tipus int
  var r = parseInt(rgb[0]);
  var g = parseInt(rgb[1]);
  var b = parseInt(rgb[2]);

  if (led_id === 'menjador'){
    // Envia a l'adreça corresponent els valors RGB
    firebase.database().ref(dbPathLed + '/llum-menjador/color').set({
      r: r,
      g: g,
      b: b
    });
  }
  // Compara el 2n paràmetre de la funció amb la resta de llums possibles
```

De la mateixa forma que amb la intensitat, el control del color s’aplica a totes les zones amb il·luminació per igual, i per tant, dintre la funció es compara l’identificador d’entrada amb totes les llums possibles ja siguin manuals o automàtiques.

- Funció per convertir el valor hexadecimal de sortida del selector de color a una cadena de caràcters amb el valor de les components RGB separats per una coma. Es passa com a paràmetre el valor hexadecimal resultant de la selecció per part de l’usuari.

`bigint = parseInt (NomCadena.substring (1), 16)` : converteix la cadena hexadecimal especificada com a primer paràmetre en un nombre enter utilitzant la funció `parseInt ()`. El segon paràmetre especifica que la cadena s’interpreta com un nombre hexadecimal. Finalment, la funció `substring(1)` s’utilitza per eliminar el primer caràcter de la cadena que correspon al prefix “0x” utilitzat per indicar que la cadena està en base hexadecimal.

Memòria

`(bigint >> n) & 255` : funció utilitzada per separar les components RGB del color. Primerament es desplacen els bits cap a la dreta un nombre de posicions especificat i a continuació es realitza una operació AND amb el nombre 255. Aquest nombre en binari correspon a “11111111”, de tal forma que tan sols es preserven els 8 bits menys significatius del resultat.

Exemple: `hexToRgb (color);`

```
function hexToRgb(hex) {
  // Converteix el valor hexadecimal a tipus int.
  // Omet el primer caràcter per ignorar el símbol '0x'
  var bigint = parseInt(hex.substring(1), 16);
  // Separa les components RGB
  // Utilitza operacions de desplaçament de bits i màscares
  var r = (bigint >> 16) & 255;
  var g = (bigint >> 8) & 255;
  var b = bigint & 255;
  // Retorna una string amb els valors RGB separats per una coma
  return r + ", " + g + ", " + b;
}
```

- Funció que escriu a la base de dades l'estat (activat o desactivat) de l'alarma d'intrusos en funció de l'interruptor de la pàgina web destinat a aquesta funció.

Exemple: `toggleAlarma ()`

```
function toggleAlarma(){
  // Si el botó està seleccionat
  if (estat_Alarma.checked)
  {
    // Mostra en la consola l'estat de l'alarma
    console.log("Toggle alarma -> alarma ACTIVADA");
    // Escriu en l'adreça corresponent l'estat de l'alarma
    firebase.database().ref(dbPathAlarma + '/estat').set("ACTIVADA");
  }
  // Si no està seleccionat
  else{
    // Mostra en la consola l'estat de l'alarma
    console.log("Toggle alarma -> alarma DESACTIVADA");
    // Escriu en l'adreça corresponent l'estat de l'alarma
    firebase.database().ref(dbPathAlarma + '/estat').set("DESACTIVADA");
  }
}
```

- Funció que, quan l'alarma està activada, monitoritza els nodes de la *database* que contenen el valor de sortida dels sensors de presència situats al rebedor i la cuina. Si algun d'ambdós sensors detecta una presència, s'indica en la pàgina web que hi ha un intrús.

Memòria

Destacar que el mostreig dels valors es realitza seguint les mateixes instruccions que s'han vist pel sensor de flama a l'inici del codi.

Exemple: checkAlarma ()

```
function checkAlarma() {
  // Quan l'alarma està activada, mostreja el valor dels sensors PIR
  if (estat_Alarma.checked) {
    // Obté el valor del sensor del rebedor i el converteix a tipus int
    dbRefPIR_Rebedor.once('value', snap => {
      pirAlarmaState.innerText = snap.val()
      const pirRebedor_State = parseInt(pirAlarmaState.textContent);
      // Obté el valor del sensor de la cuina i el converteix a tipus int
      dbRefPIR_Cuina.once('value', snap => {
        pirAlarmaState.innerText = snap.val()
        const pirCuina_State = parseInt(pirAlarmaState.textContent);
        // Si es detecta presència intercanvia l'estat
        // Mostra el resultat en la web
        if ((pirRebedor_State === 1) || (pirCuina_State === 1)) {

          pirAlarmaStateContainer.classList.add('detect');
          pirAlarmaStateOut.textContent = "INTRÚS";
        } else {
          // Si NO es detecta, intercanvia l'estat
          // Mostra el resultat en la web
          pirAlarmaStateContainer.classList.remove('detect');
          pirAlarmaStateOut.textContent = "";
        }
      })
    })
  }
  // No mostra el resultat si l'alarma no està activada
  else {
    pirAlarmaStateContainer.classList.remove('detect');
    pirAlarmaStateOut.textContent = "";
  }
}
```

- Funció que escriu a la base de dades l'estat (obert o tancat) d'una determinada porta en funció de l'interruptor corresponent en la pàgina web. Es passa com a paràmetre l'identificador assignat a la porta determinada.

Exemple: togglePorta ('porta-exterior')

```
function togglePorta(porta_id) {
  // Porta exterior d'accés a la parcel·la
  if (porta_id === 'porta-exterior') {
    // Si el botó està seleccionat
    if (estat_PortaExt.checked)
    {
```

Memòria

```
// Mostra en la consola l'estat de la porta
console.log("Toggle porta exterior -> porta exterior OBERTA");
// Escriu en l'adreça corresponent l'estat de la porta
firebase.database().ref(dbPathPortes + '-exterior').set("OBERTA");
}
// Si no està seleccionat
else{
  // Mostra en la consola l'estat de la porta
  console.log("Toggle porta exterior -> porta exterior TANCADA");
  // Escriu en l'adreça corresponent l'estat de la porta
  firebase.database().ref(dbPathPortes + '-exterior').set("TANCADA");
}
}
// Porta d'accés a la casa
else if (porta_id === 'porta-casa') {
  if (estat_PortaCasa.checked)
  {
    console.log("Toggle porta casa -> porta casa OBERTA");
    firebase.database().ref(dbPathPortes + '-casa').set("OBERTA");
  }
  else{
    console.log("Toggle porta casa -> porta casa TANCADA");
    firebase.database().ref(dbPathPortes + '-casa').set("TANCADA");
  }
}
}
```

- Funció que escriu a la base de dades l'estat (ON o OFF) de la depuradora en funció que l'interruptor destinat a aquesta tasca.

Exemple: toggleDepuradora ()

```
function toggleDepuradora(){
  // Si el botó està seleccionat
  if (estat_depuradora.checked)
  {
    // Mostra en la consola l'estat de la depuradora
    console.log("Toggle depuradora -> depuradora ON");
    // Escriu en l'adreça corresponent l'estat de la depuradora
    firebase.database().ref(dbPathDepuradora).set("ON");
  }
  // Si no està seleccionat
  else{
    // Mostra en la consola l'estat de la depuradora
    console.log("Toggle depuradora -> alarma OFF");
    // Escriu en l'adreça corresponent l'estat de la depuradora
    firebase.database().ref(dbPathDepuradora).set("OFF");
  }
}
```

Memòria

- Funció que escriu a la base de dades el valor de la temperatura seleccionada per l'usuari en la pàgina web. Es passa com a paràmetre el valor resultant de l'entrada on l'usuari pot establir la temperatura desitjada a l'interior de l'habitatge.

Exemple: `actualitzaTemp(this.value)`

```
function actualitzaTemp(valor) {
  // Mostra en consola la temperatura seleccionada
  console.log("Set temperatura desitjada: " + valor);
  // Escriu en l'adreça corresponent el valor de tipus float seleccionat
  firebase.database().ref(dbPathClima + '/temp-desitjada')
  .set(parseFloat(valor));
}
```

- Funció que monitoritza el valor de la temperatura ambient a través del seu node específic en la base de dades i que mostra en la pàgina web l'estat de l'aire condicionat en funció d'aquest valor obtingut i del valor desitjat per l'usuari.

Més concretament, si la temperatura ambient és superior a la desitjada, s'intercanvia el resultat mostrat en el contenidor i s'indica que l'aire condicionat està funcionant per adaptar l'ambient a la desitjada. Si és dona la situació contrària, es mostra en el contenidor que l'aire està apagat.

Exemple: `setAc()`

```
function setAC(){
  dbRefTemp_Desitj.once('value', snap => {
    // Obté el valor de la temperatura desitjada i el converteix a float
    tempDesitjada.innerText = snap.val();
    const tempDesitj = parseFloat(tempDesitjada.textContent);
    // Obté el valor de la mesura de temperatura realitzada pel DHT22
    const tempMedida = parseFloat(tempElement.innerText);

    // Si la temperatura ambiental es major a la desitjada, engega AC
    if(tempDesitj < tempMedida){
      acEstatContainer.classList.remove('off');
      acEstatContainer.classList.add('on');
      acEstatOut.textContent = "ON";
      // Mostra en consola que l'AC passa a ON
      console.log("Set AC: ON");
      // Escriu en l'adreça corresponen l'estat de l'AC
      firebase.database().ref(dbPathClima + '/estat-AC').set("ON");
    }
    // Si la temperatura ambiental es menor a la desitjada, apaga AC
    else {
      acEstatContainer.classList.remove('on');
      acEstatContainer.classList.add('off');
      acEstatOut.textContent = "OFF";
    }
  }
}
```

Memòria

```
// Mostra en consola que l'AC passa a OFF
console.log("Set AC: OFF");
// Escriu en l'adreça corresponen l'estat de l'AC
firebase.database().ref(dbPathClima + '/estat-AC').set("OFF");
}
});
}
```

Aquestes són totes les funcions dissenyades per interactuar entre els elements de la pàgina web i els diferents nodes de la base de dades en temps real. Per finalitzar el codi JavaScript, es realitza la gestió dels elements a mostrar en el cos principal de la pàgina en funció de la secció escollida per l'usuari en la barra de navegació.

Aquesta tasca permet accedir als diferents dispositius de forma més ràpida i clara ja que tan sols es mostren els dispositius associats a la secció especificada per l'usuari. A més, també es mostra en la part superior el títol corresponent. Per exemple, si es selecciona en la barra de navegació la secció de seguretat, es mostra en la part superior el títol i tan sols es mostren en pantalla aquells quadres que hi tenen relació com el de l'alarma, el control d'accessos, l'alarma d'incendis, etc.

Per tal de realitzar aquesta tasca principalment s'utilitzen tres instruccions:

- `document.querySelector('#NomId').addEventListener('click', function() : aquest codi assigna un esdeveniment de clic a l'element amb l'identificador definit. Quan es realitza clic en aquest, s'executa la funció de retornament de trucada proporcionada que en aquest cas, s'encarrega de canviar el títol en la part superior i de crida la funció que gestiona els elements a mostrar.`
- `document.querySelectorAll('.quadre').forEach(quadre => {
quadre.style.display = 'none'; }) : es realitza una iteració sobre tots els elements definits com a quadre i s'estableix el seu estil de visualització en "none", significat així que es deixen de mostrar tots els elements.`
- `document.querySelector('#NomQuadre').style.display = 'block' : selecciona el quadre identificat amb el nom definit i el mostra en pantalla.`

El codi que realitza aquesta tasca i utilitza les instruccions mencionades és el següent:

```
// Gestiona els elements que es mostren en la secció llums interiors
// Assigna un event de click a l'element amb id 'link-llums-interior'
document.querySelector('#link-llums-interior').addEventListener('click',
function() {
// Mostra el corresponent títol en la part superior
document.querySelector('main h3').innerText = "Llums interiors";
mostraLlumInterior();
});
```

Memòria

```
// Funció que mostra en pantalla els quadres especificats
function mostraLlumInterior() {
  // Amaga tots els quadres
  document.querySelectorAll('.quadre').forEach(quadre => {
    quadre.style.display = 'none';
  });
  // Mostra els especificats per la secció
  document.querySelector('#quadre-llum-menjador').style.display = 'block';
  document.querySelector('#quadre-llum-rebedor').style.display = 'block';
  document.querySelector('#quadre-llum-cuina').style.display = 'block';
  document.querySelector('#quadre-llum-wc').style.display = 'block';
}
```

Aquest codi s'implementa per totes les seccions especificades en la barra de navegació lateral, actualitzant els identificadors definits per aquestes. Es pot consultar la resta de codi accedint a [Codi JavaScript principal](#).

1.7.5.4. Codi per al microcontrolador ESP32

Els darrers arxius de codi dissenyats per implementar sistema domòtic són els encarregats que comunicar els dispositius físics de hardware amb la base de dades en temps real per mitjà del microcontrolador que governa la domotització. Gràcies a la comunicació bidireccional entre la *database* i el ESP32, és a dir, d'escriptura i de lectura, els sensor poden realitzar les mesures per tal que quedin establides en els nodes específics o els diferents dispositius poden actuar en funció de les peticions de l'usuari obtingudes de la base de dades.

El sistema domòtic d'aquest projecte està dissenyat per controlar els dispositius amb un total de 3 microcontroladors i, com s'ha mencionat anteriorment, aquests es programen en l'entorn de desenvolupament Arduino IDE. Els dispositius que gestiona cada una de les plaques ESP32 es divideixen en funció de la seva tasca en l'habitatge de manera que queden repartits de la següent forma:

- **Microcontrolador de sensors:** s'encarrega d'escriure a la base de dades les mesures dels sensors incorporats (temperatura, humitat, sensor de flama i sensor de presència), realitza el control de l'aire condicionat en funció de la temperatura ambient i la desitjada i de gestiona l'activació de l'alarma sonora en els casos necessaris.
- **Microcontrolador de motors:** s'encarrega de gestionar el moviment automàtic dels servomotors en funció de la intensitat lumínica mesurada pels fotoresistors o manual d'acord amb les peticions de l'usuari i de controlar l'activació i aturada del motor de la depuradora.
- **Microcontrolador d'il·luminació:** s'encarrega de gestionar l'estat, la intensitat i el color de totes les tires LED implementades en l'habitatge ja siguin de funcionament manual o automàtic.

Amb l'objectiu de tenir una visió més global dels codis introduïts en els microcontroladors que es mostren a continuació es defineix l'estructura global que segueixen:

1. Declaració de llibreries, definició de constants, variables globals i objectes.
2. Definició de funcions.
3. Inicialització de dispositius i de Firebase Realtime Database i configuració de la connexió Wi-Fi i aquesta.
4. Bucle principal del programa on es realitzen les mesures dels sensors i s'executen les funcions definides.

En el desenvolupament de projectes electrònics i de sistemes embeguts, les llibreries prenen un paper fonamental ja que s'encarreguen de proporcionar funcionalitats predefinides i abstraccions que simplifiquen la programació dels sensors i dispositius. Seguidament es mostren les llibreries utilitzades en el disseny del codi Arduino:

- **WiFi:** llibreria utilitzada en dispositius ESP32 per connectar-se i configurar xarxes de Wi-Fi. Permet que el microcontrolador es connecti a la xarxa especificada per tal de realitzar la comunicació amb la base de dades. [28]
- **WiFiMulti:** extensió de la llibreria anterior que facilita la connexió del microcontrolador en diverses xarxes Wi-Fi prèviament definides. Aquesta llibreria simplifica l'ús de múltiples punts de connexió, resultant molt útil per dispositius que necessiten canviar entre diferents xarxes segons la disponibilitat. [29]
- **Firestore_ESP_Client:** es tracta d'una llibreria específica pel microcontrolador ESP32 que permet interactuar amb la Firebase Realtime Database. S'encarrega de facilitar la connexió i comunicació amb la base de dades mitjançant un seguit d'instruccions específiques. [30]
- **Wire:** llibreria pròpia d'Arduino que proporciona una interfície per a la comunicació I2C, un protocol tipus sèrie de dos fils que permet comunicar diferents perifèrics de forma eficient. [31]
- **ESP32Servo:** es tracta d'una llibreria que s'encarrega de proporcionar una interfície per controlar servomotors utilitzant l'ESP32. Aquesta facilita la connexió i l'ús d'aquests dispositius permetent un control precís de la posició. [32]
- **Adafruit_Sensor:** llibreria de suport desenvolupada per Adafruit que proporciona una interfície comuna per llegir dades d'una gran varietat de sensors, entre ells el DHT22. Realitza una abstracció de les diferències entre els diversos tipus de sensors, permetent així que les biblioteques específiques de cada un d'aquests s'integren fàcilment amb les aplicacions a dissenyar. [33]
- **Adafruit_NeoPixel:** llibreria desenvolupada per Adafruit que proporciona funcions per controlar matrius de tires LED WS2812B. Aquestes matrius consisteixen en una sèrie de LED RGB que poden ser controlats de forma individual. La llibreria s'encarrega de facilitar la configuració dels LED mitjançant el control de la coloració, intensitat i efectes d'animació. [34]

Memòria

A continuació, es defineixen el conjunt d'instruccions més utilitzades a l'hora de desenvolupar els diferents codis Arduino. Aquestes instruccions són pròpies de la llibreria de Firebase Realtime Database i són les encarregades de comunicar el microcontrolador amb la base de dades mitjançant la escriptura i la lectura de valors en els diferents nodes.

- `Firebase.RTDB.getTipusDada(&fbdo, Adreça.c_str())` : instrucció utilitzada per iniciar el procés de lectura des de l'adreça especificada d'un valor el tipus del qual és el definit. Els diferents tipus de dada amb què es treballa són la cadena de caràcters (*string*), el valor enter (*int*) o el valor amb decimals (*float*).
- `NomVariable = fbdo.TipusDadaData`: instrucció utilitzada per emmagatzemar en la variable especificada el valor resultant de la lectura. Cal definir el tipus de dada que s'emmagatzema i cal que sigui concordant amb el declarat en la instrucció anterior.
- `Firebase.RTDB.setTipusDada(&fbdo, Adreça.c_str(), valor)` : instrucció utilitzada per escriure en l'adreça especificada el valor passat com a paràmetre. De la mateixa forma que anteriorment, cal definir el tipus de valor que es vol escriure en la base de dades.

Finalment, cal destacar que tots tres codis disposen d'una part comuna on és realitzada la connexió a les diferents xarxes Wi-Fi disponibles i la configuració a la Firebase Realtime Database. El codi corresponent a aquesta tasca és el següent:

```
void initWiFi() {
  // Xarxes Wi-Fi definides
  wifiMulti.addAP(ssid_1, contrasenya_1);
  wifiMulti.addAP(ssid_2, contrasenya_2);
  wifiMulti.addAP(ssid_3, contrasenya_3);
  wifiMulti.addAP(ssid_4, contrasenya_4);
  // Estableix el mode de funcionament com a estació (STA)
  WiFi.mode(WIFI_STA);
  Serial.print("Connectant a Wi-Fi ..");
  while (wifiMulti.run(TempsEsperaWifi) != WL_CONNECTED) {
    Serial.print(".");
  }
  Serial.println(WiFi.localIP());
  Serial.println();
}
```

La funció mostrada s'encarrega d'inicialitzar la connexió Wi-Fi del dispositiu. En primer lloc, es defineixen totes les xarxes a les quals el microcontrolador s'hi pot connectar i posteriorment s'estableix el mode de funcionament del dispositiu com a estació (STA), és a dir, actuarà com un client que es connecta a un punt d'accés existent. Seguidament, s'executa un bucle mentre el dispositiu s'intenta connectar a alguna de les xarxes definides mitjançant la instrucció `wifiMulti.run`, que retorna l'estat de la connexió. Quan ja s'ha realitzat la connexió satisfactòriament, es mostra en el port sèrie la direcció IP assignada al microcontrolador.

Memòria

Un cop definida la funció, cal situar-se en la part del codi on és realitzen les configuracions inicials, és a dir, on es crida la funció mostrada i es configura la base de dades.

```
void setup() {
  Serial.begin(115200)
  // Inicialitza dispositius i sensors
  // Inicialitza connexió Wi-Fi
  initWiFi();
  // Assigna la clau de l'API del projecte Firebase
  config.api_key = API_KEY;
  // Assigna les credencials d'inici de sessió de l'usuari
  auth.user.email = USER_EMAIL;
  auth.user.password = USER_PASSWORD;
  // Assigna la URL de la database
  config.database_url = DATABASE_URL;
  // Configura la reconexió automàtica del Wi-Fi en cas de desconexió
  Firebase.reconnectWiFi(true);
  fbdo.setResponseSize(4096);
  // Assigna la funció de callback per a la tasca de generació de token
  // de llarga durada
  config.token_status_callback = tokenStatusCallback;
  // Assigna el màxim nombre d'intents en la generació del token
  config.max_token_generation_retry = 5;
  // Inicialitza la llibreria amb l'autenticació i la configuració de
  // Firebase
  Firebase.begin(&config, &auth);

  // Espera mentre obté l'UID de l'usuari
  Serial.println("Obtenint UID usuari");
  while ((auth.token.uid) == "") {
    Serial.print('.');
    delay(1000);
  }
  // Imprimeix user UID
  uid = auth.token.uid.c_str();
  Serial.print("UID usuari: ");
  Serial.println(uid);

  // Adreça a la database
  databasePath = "/UsersData/" + uid;

  // Adreces als diferents elements de la database
}
```

Finalment, cal destacar que la definició de les xarxes Wi-Fi a les quals es pot connectar el microcontrolador i de les claus d'accés a la base de dades es realitza en un arxiu diferent, el qual es crida a l'inici del codi, ja que es tracta d'informació única i intransferible.

Memòria

○ Codi microcontrolador de sensors:

Com es pot observar al llarg del codi, aquest segueix l'estructura global mencionada anteriorment, començant per la definició de les llibreries de connexió tant a les xarxes Wi-Fi com a la base de dades. A més, inclou la llibreria de suport d'Adafruit utilitzada per sensors i l'arxiu específic pel sensor DHT utilitzat.

```
/****** LLIBRERIES & FITXERS *****/  
// Llibreries de connexió al Wi-Fi i a la Firebase Database  
#include <WiFi.h>  
#include <WiFiMulti.h>  
#include <Firebase_ESP_Client.h>  
#include <Wire.h>  
#include "data.h"  
// Llibreries del sensor de temperatura i humitat DHT22  
#include <Adafruit_Sensor.h>  
#include "DHT.h"
```

El següent bloc del codi està constituït per la definició de totes les constants, variables globals i objectes necessaris pel correcte desenvolupament de les funcions posteriors. Entre les declaracions es poden trobar els objectes equivalents a la base de dades Firebase amb la corresponent autenticació i configuració o el creat per inicialitzar la connexió Wi-Fi, les variables de tipus cadena de caràcters que contenen les adreces als diferents nodes, el temps de mostreig del programa, les constants on es declaren els pins d'entrada/sortida del microcontrolador on es connecta cada dispositiu o les diferents variables on s'emmagatzemen les lectures i estats.

```
/****** VARIABLES & OBJECTES *****/  
// Objectes Firebase  
FirebaseData fbdo;      // Database  
FirebaseAuth auth;     // Autenticació  
FirebaseConfig config; // Configuració  
  
// Variable per guardar USER UID  
String uid;  
  
// Variables per guardar les adreces a la database  
String databasePath;  
String climatitzacioPath;  
String focPath;  
String alarmaIntrusosPath;  
String PIR_LedPath;  
  
// Objecte Wi-Fi  
WiFiMulti wifiMulti;  
const uint32_t TempsEsperaWifi = 5000;
```

Memòria

```
// Variables temporals (temps de mostreig)
unsigned long millisCicleAnterior = 0;
unsigned long tempsMostreig = 2000;

/**** Sensor DHT22 ****/
#define pinDHT 4
DHT sensorTH (pinDHT, DHT22);
float temperatura;
float humitat;

/**** Control de temperatura ****/
#define pinAC 14
float temp_desitjada;

/**** Sensor de flama ****/
// Sensor menjador
#define pinFoc_Menj 18
int foc_estat_menj;
// Sensor cuina
#define pinFoc_Cuina 22
int foc_estat_cuina;
// Buzzer
#define pinBuzzer 19

/**** Sensors PIR ****/
#define pinSensor_Rebedor 32
int pir_estat_rebedor;
#define pinSensor_Cuina 33
int pir_estat_cuina;
#define pinSensor_WC 34
int pir_estat_WC;
```

Un cop s'ha realitzat la definició de tots els paràmetres necessaris, es realitza el desenvolupament de les funcions dissenyades per aquest microcontrolador. Cal destacar que la primera funció present en el codi es tracta de la identificada com *initWifi* (), la qual s'ha tractat anteriorment degut a la seva presència en tots els arxius Arduino.

- Funció que envia a la base de dades els valors resultants de la mesura del sensor DHT22. Es passa com a paràmetre l'adreça al node específic i el valor tipus *float* corresponent a la temperatura o a la humitat.

Per escriure el valor en la base de dades, utilitza la instrucció vista anteriorment.

Memòria

```
void enviaTH(String path, float valor){
  // Configura el valor
  if (Firebase.RTDB.setFloat(&fbdo, path.c_str(), valor)){
    Serial.print("Escribint valor: ");
    Serial.print (valor);
    Serial.print(" a la següent adreça: ");
    Serial.println(path);
  }
  else {
    // Indica el motiu de l'error en la escriptura a la database
    Serial.println("ERROR");
    Serial.println("CAUSA: " + fbdo.errorReason());
  }
}
```

- Funció que realitza la lectura de la base de dades per obtenir del valor de temperatura desitjada per l'usuari. També s'encarrega de comparar aquesta temperatura obtinguda amb la mesurada a través del sensor per tal de gestionar l'estat de l'aire condicionat en funció d'aquesta comparació.

Per realitzar la lectura de la base de dades, s'utilitzen les instruccions vistes.

Si la temperatura mesurada és superior a la desitjada, es posa en funcionament l'aire condicionat per tal d'adaptar l'ambient al desitjat per l'usuari. Quan la temperatura s'adapta a l'establerta, es canvia l'estat de l'aire i s'atura.

S'utilitza la instrucció *digitalWrite* (*pinAC*, *HIGH*) per establir el port de sortida associat al motor de l'aire condicionat a nivell alt, és a dir, per posar-lo en funcionament. Si es desitja aturar al motor, s'utilitza la mateixa instrucció però substituint el segon paràmetre per l'estat *LOW*.

```
void controlTemp(String path) {
  String nextEstat = "";
  // Obté el valor de la temperatura desitjada
  if (Firebase.RTDB.getFloat(&fbdo, (path + "/temp-desitjada").c_str())){
    temp_desitjada = fbdo.floatData();
    Serial.print("Llegint valor: ");
    Serial.print(temp_desitjada);
    Serial.print(" de la següent adreça: ");
    Serial.println(path + "/temp-desitjada");

    // Compara la temperatura mesurada i desitjada per controlar l'AC
    if (temp_desitjada < temperatura){
      // Activa AC per baixar la temperatura a la desitjada
      digitalWrite(pinAC, HIGH);
      // Envia a la database l'estat ON de l'AC
      nextEstat = "ON";
      Firebase.RTDB.setString(&fbdo, (path + "/estat-AC").c_str(),
nextEstat);
      Serial.println("AC estat set to ON");
    }
  }
```

Memòria

```
else {
    // Apaga AC per mantenir la temperatura desitjada
    digitalWrite(pinAC, LOW);
    // Envia a la database l'estat OFF de l'AC
    nextEstat = "OFF";
    Firebase.RTDB.setString(&fbdo, (path + "/estat-AC").c_str(),
nextEstat);
    Serial.println("AC estat set to OFF");
}
}
else {
    // Indica el motiu de l'error en la lectura de la database
    Serial.println("ERROR");
    Serial.println("CAUSA: " + fbdo.errorReason());
}
}
```

- o Funció que envia a la base de dades el valor resultant de la mesura del sensor de flama. Es passa com a paràmetre l'adreça al node específic i el valor tipus enter corresponent.

```
void enviaFoc (String path, int valor){
    // Configura el valor
    if (Firebase.RTDB.setInt(&fbdo, path.c_str(), valor)){
        Serial.print("Escribint valor: ");
        Serial.print (valor);
        Serial.print(" a la següent adreça: ");
        Serial.println(path);
    }
    else {
        // Indica el motiu de l'error en la escriptura a la database
        Serial.println("ERROR");
        Serial.println("CAUSA: " + fbdo.errorReason());
    }
}
```

- o Funció que envia a la base de dades el valor resultant de la mesura del sensor PIR. Es passa com a paràmetre l'adreça al node específic i el valor tipus enter corresponent a la detecció o no de presència.

```
void sendPresence (String path, int valor){
    // Configura el valor
    if (Firebase.RTDB.setInt(&fbdo, path.c_str(), valor)){
        Serial.print("Escribint valor: ");
        Serial.print (valor);
        Serial.print(" a la següent adreça: ");
        Serial.println(path);
    }
}
```

Memòria

```
    else {  
      // Indica el motiu de l'error en la escriptura a la database  
      Serial.println("ERROR");  
      Serial.println("CAUSA: " + fbdo.errorReason());  
    }  
  }  
}
```

- Funció que gestiona l'estat de l'alarma sonora en funció de les lectures obtingudes a través dels sensors de flama i de presència. La comprovació de l'aparició d'un incendi es realitza en tot moment, mentre que la dels sensors PIR tan sols si l'alarma d'intrusions està activa. Per aquest motiu, es realitza la lectura de l'estat prèviament a la monitorització dels sensors de presència.

En el moment en què la mesura dels sensors de flama indiquen la detecció de foc o bé es detecta una intrusió mitjançant els PIR, s'activa l'alarma sonora mitjançant la instrucció *analogWrite* (*pinBuzzer, 100*). Si es desitja desactivar, es substitueix el segon paràmetre pel valor 0.

```
void activarBuzzer(String path){  
  // Obté l'estat de l'alarma (ACTIVADA/DESACTIVADA)  
  Firebase.RTDB.getString(&fbdo, (path + "/estat").c_str());  
  String estat = fbdo.stringData();  
  Serial.print("Llegint valor: " + estat);  
  Serial.println(" de la següent adreça: " + path);  
  // Activa alarma sonora si qualsevol sensor detecta presència i  
  // l'alarma està activa  
  if((estat == "ACTIVADA") && ((pir_estat_rebedor == 1) ||  
  (pir_estat_cuina == 1))){  
    analogWrite(pinBuzzer, 100);  
  }  
  //Activa l'alarma si algun sensor detecta foc  
  else if ((foc_estat_menj == 1) || (foc_estat_cuina == 1)){  
    analogWrite(pinBuzzer, 100);  
  }  
  // Apaga l'alarma si cap sensor detecta foc ni presència  
  else {  
    analogWrite(pinBuzzer, 0);  
  }  
}
```

Un cop analitzades el total de les funcions dissenyades, el codi entra en el següent gran bloc, el de la configuració inicial. En aquest, es realitza la inicialització dels diferents ports d'entrada i sortida, la configuració vista de la connexió Wi-Fi i de la Firebase Realtime Database i la definició de les adreces als nodes.

Memòria

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

  // Inicialitza sensor DHT22
  sensorTH.begin();

  // Inicialitza sensor de flama
  pinMode(pinFoc_Menj, INPUT);
  pinMode(pinFoc_Cuina, INPUT);

  // Inicialitza motor AC
  pinMode(pinAC, OUTPUT);

  // Inicialitza sensors PIR
  pinMode(pinSensor_Rebedor, INPUT);
  pinMode(pinSensor_Cuina, INPUT);
  pinMode(pinSensor_WC, INPUT);

  // Configuració del Wi-Fi i de la base de dades vista anteriorment

  // Adreces als diferents elements de la database
  climatitzacioPath = databasePath + "/climatitzacio";
  focPath = databasePath + "/alarma-incendis";
  alarmaIntrusosPath = databasePath + "/alarma";
  PIR_LedPath = databasePath + "/llums";
}
```

Finalment, el codi executa el bucle principal propi del sistema *embedded*. Aquest és l'encarregat de realitzar les mesures dels sensors, cridar les funcions anteriors per enviar les dades als nodes corresponents i gestionar els dispositius en funció de les mesures.

Cal destacar que totes aquestes accions es realitzen en intervals de temps iguals al temps de mostreig definit al principi del codi. En el cas d'aquest projecte, el temps de mostreig es tracta de 2 segons degut a què es tracta d'un prototip, mentre que en cas d'aplicar el sistema domòtic en un cas real, aquest temps es podria incrementar fins a valors d'un minut, suposant així un menor consum d'energia per part dels sensors.

A través de la instrucció *Firestore.ready()* es comprova si la base de dades està llesta per realitzar operacions i si ho està, es comprova que hagi transcorregut el temps seleccionat mitjançant la comparació entre el temps de mostreig i el valor de la funció *millis()*, que retorna el nombre de mil·lisegons transcorreguts des de la seva crida.

Un cop el valor d'aquesta comparació es favorable, es realitzen totes les operacions necessàries començant per la lectura dels sensors, continuant per l'enviament dels valors a la base de dades i finalitzant per la gestió dels resultats.

Memòria

```
void loop(){
  // Espera el temps de mostreig seleccionat
  if (Firebase.ready() && (millis() - millisCicleAnterior > tempsMostreig
|| millisCicleAnterior == 0)){
    millisCicleAnterior = millis();

    // Realitza les mesures dels sensors
    temperatura = sensorTH.readTemperature();
    humitat = sensorTH.readHumidity();
    foc_estat_menj = int(digitalRead(pinFoc_Menj));
    foc_estat_cuina = int(digitalRead(pinFoc_Cuina));
    pir_estat_rebedor = int(digitalRead(pinSensor_Rebedor));
    pir_estat_cuina = int(digitalRead(pinSensor_Cuina));
    pir_estat_WC = int(digitalRead(pinSensor_WC));

    // Envia les mesures a la database
    enviaTH(climatitzacioPath + "/temperatura", temperatura);
    enviaTH(climatitzacioPath + "/humitat", humitat);
    enviaFoc(focPath + "/menjador", foc_estat_menj);
    enviaFoc(focPath + "/cuina", foc_estat_cuina);
    enviaPIR(PIR_LedPath + "/llum-rebedor/estat", pir_estat_rebedor);
    enviaPIR(PIR_LedPath + "/llum-cuina/estat", pir_estat_cuina);
    enviaPIR(PIR_LedPath + "/llum-wc/estat", pir_estat_WC);
    enviaPIR(alarmaIntrusosPath + "/pir-rebedor", pir_estat_rebedor);
    enviaPIR(alarmaIntrusosPath + "/pir-cuina", pir_estat_cuina);

    // Llegeix dades de la database i gestiona el dispositiu corresponent
    controlTemp(climatitzacioPath);
    activarBuzzer(alarmaIntrusosPath);
  }
}
```

o Codi microcontrolador de motors:

Seguint amb l'estructura definida inicialment, en primer lloc es mostren les llibreries utilitzades. En aquest cas en particular, s'utilitzen les vistes anteriorment i s'afegeix la llibreria ESP32Servo per controlar els servomotors que actuen en el moviment de les persianes.

```
/****** LLIBRERIES & FITXERS *****/
// Llibreries de connexió al Wi-Fi i a la Firebase Database
#include <WiFi.h>
#include <WiFiMulti.h>
#include <Firebase_ESP_Client.h>
#include <Wire.h>
#include "data.h"
// Llibreria servomotors
#include <ESP32Servo.h>
```

Memòria

El següent bloc que forma el codi és la definició de les constants, variables i objectes que s'utilitzen posteriorment en les funcions dissenyades. Entre totes aquestes declaracions es troben les corresponents a la base de dades (autenticació, adreces...), a la connexió Wi-Fi i als intervals temporals tal i com es realitza en el codi anterior. Les pròpies d'aquest arxiu són totes aquelles associades als servomotors, motor de la depuradora i als fotoresistors com per exemple, els pins de connexió o les variables per emmagatzemar els estats.

```
/****** VARIABLES & OBJECTES *****/
// Objectes Firebase
FirebaseData fbdo;      // Database
FirebaseAuth auth;     // Autenticació
FirebaseConfig config; // Configuració

// Variable per guardar USER UID
String uid;

// Variable per guardar les adreces a la database
String databasePath;
String persianaPath;
String depuradoraPath;

// Objecte Wi-Fi
WiFiMulti wifiMulti;
const uint32_t TempsEsperaWifi = 5000;

// Variables temporals (temps de mostreig)
unsigned long millisCicleAnterior = 0;
unsigned long tempsMostreig = 2000;

/***** Sensors intensitat llum *****/
#define pinSensorLlum_Menj 34
#define pinSensorLlum_Hab 35
int intensitatLlum_Menj;
int intensitatLlum_Hab;

/***** Mode de funcionament persianes *****/
String modeFunc_Menj;
String modeFunc_Hab;

/***** Servomotors persianes *****/
#define pinPersiana1 14
#define pinPersiana2 12
Servo persiana1;
Servo persiana2;

/**** Motor depuradora ****/
#define pinDepuradora 13
int estat_depuradora;
```

Memòria

Les funcions dissenyades per gestionar els diferents motors són les mostrades a continuació.

- Funció que envia a la base de dades el valor de la intensitat lumínica resultant de la mesura del fotoresistor. Es passa com a paràmetre l'adreça al node específic i el valor tipus enter corresponent.

Per realitzar l'escriptura en la base de dades, s'utilitzen les funcions pròpies de Firebase vistes a l'inici de l'explicació.

```
void enviaLlum(String path, int intensitat){
  // Configura el valor
  if (Firebase.RTDB.setInt(&fbdo, path.c_str(), intensitat)){
    Serial.print("Escribint valor: ");
    Serial.print(intensitat);
    Serial.print(" a la següent adreça: " + path);
    Serial.println();
  }
  else {
    // Indica el motiu de l'error en la escriptura a la database
    Serial.println("ERROR");
    Serial.println("CAUSA: " + fbdo.errorReason());
  }
}
```

- Funció que realitza la lectura de la base de dades per obtenir el mode de funcionament de les persianes desitjat per l'usuari. Cal recordar que es pot realitzar un control del moviment manual o un automàtic en funció de la llum.

Es passa com a paràmetre l'adreça que accedeix al node corresponent i l'identificador de la persiana a la qual es dirigeix la lectura.

```
void modeFuncPersiana(String path, String id){
  // Obté el mode de funcionament
  if (Firebase.RTDB.getString(&fbdo, (path).c_str())){
    String modeFunc = fbdo.stringData();
    Serial.println();
    Serial.print("Llegint valor: " + modeFunc);
    Serial.println(" de la següent adreça: " + path);
    // Assigna el mode de funcionament a la persiana especificada
    if (id == "menj"){ modeFunc_Menj = modeFunc;
    }
    else if (id == "hab"){ modeFunc_Hab = modeFunc;
    }
  }
  else {
    // Indica el motiu de l'error en la lectura de la database
    Serial.println("ERROR");
    Serial.println("CAUSA: " + fbdo.errorReason());
  }
}
```

- Funció que gestiona el moviment de la persiana especificada en funció del seu mode de funcionament. Si aquesta es troba amb un control automàtic, s'utilitza el valor de la intensitat lumínica per situar-la en un punt concret. Si aquesta es troba en mode manual, es realitza una lectura de la base de dades per obtenir la cadena de caràcters corresponent al moviment desitjat per part de l'usuari.

Es passa com a paràmetre l'adreça i l'objecte de tipus servo corresponent a la persiana que es vol controlar, el mode de funcionament obtingut en la funció anterior i el valor de la intensitat lumínica.

Abans de gestionar el moviment en funció del mode especificat per l'usuari, es realitza una lectura de la base de dades de l'estat de la persiana. En funció de la seva posició, s'han definit 3 estats: pujada, a la meitat o baixada.

Per gestionar el sentit de gir dels servomotors, s'utilitzen les instruccions següents:

- `Persiana.write(180)` : puja la persiana girant l'eix en sentit antihorari.
- `Persiana.write(90)` : atura el servomotor.
- `Persiana.write(0)` : baixa la persiana girant l'eix en sentit horari.

En referència al mode automàtic, es realitza un moviment de pujada o de baixada durant un cert temps en funció de la intensitat lumínica i de la posició anterior de la persiana. Més concretament es poden donar els casos següents:

- **Intensitat de llum alta:** correspon a un valor en la mesura del fotoresistor d'entre 0 i 1500. Si la persiana es troba totalment baixada, realitza un moviment ascendent durant 4 segons. Si aquesta es troba a la meitat, realitza un moviment ascendent durant 2 segons.
- **Intensitat de llum mitja:** correspon a un valor en la mesura del fotoresistor d'entre 1500 i 2500. Si la persiana es troba totalment baixada, realitza un moviment ascendent durant 2 segons. Si aquesta es troba totalment pujada, realitza un moviment descendent durant 2 segons.
- **Intensitat de llum baixa:** correspon a un valor en la mesura del fotoresistor d'entre 2500 i 4095. Si la persiana es troba totalment pujada, realitza un moviment descendent durant 4 segons. Si aquesta es troba a la meitat, realitza un moviment descendent durant 2 segons.

En tots els casos, un cop finalitzat el moviment es realitza una escriptura en la base de dades del nou estat posicional de la persiana.

Finalment, en el cas que l'usuari desitgi realitzar un control manual de la posició, es realitza una lectura del node corresponent on s'indica el moviment a realitzar. Els tres possibles moviments són: pujar, aturar o baixar.

Memòria

```
void setPersiana(String path, Servo &persiana, String modeFunc, int
intLlum ){
    String accio = "";
    String estat = "";
    // Variables per gestionar el temps d'engegada
    unsigned long tempsInici1 = millis();
    bool fiTemps1 = true;
    unsigned long tempsInici2 = millis();
    bool fiTemps2 = true;

    // Verifica l'estat de la persiana (pujada, mitant, baixada)
    Firebase.RTDB.getString(&fbdo, (path + "/estat").c_str());
    estat = fbdo.stringData();
    Serial.print("Estat persiana: " + estat);
    Serial.println(" de la següent adreça: " + path + " /estat");
    // Control en mode automàtic
    if (modeFunc == "AUTO"){
        // Intensitat llum alta (0-1500) puja persiana al 100%
        if (intLlum <= 1500){
            if (estat == "mitant"){
                // Motor gira en sentit antihorari (puja) durant 2 segons
                persiana.write(180);
                while (millis() - tempsInici1 < 2000) { }
                // Atura el motor i escriu a la database l'estat
                persiana.write(90);
                Firebase.RTDB.setString(&fbdo, (path + "/estat").c_str(),
"pujada");
            }
            else if (estat == "baixada"){
                // Motor gira en sentit antihorari (puja) durant 4 segons
                persiana.write(180);
                while (millis() - tempsInici1 < 4000) { }
                // Atura el motor i escriu a la database l'estat
                persiana.write(90);
                Firebase.RTDB.setString(&fbdo, (path + "/estat").c_str(),
"pujada");
            }
        }
    }
}
```

Per la resta de casuístiques possibles en funció de la llum i de l'estat anterior, s'utilitza el mateix codi però intercanviant els intervals que pot prendre la intensitat, el moviment ascendent o descendent i els temps durant el qual el realitza.

Memòria

```
else if (modeFunc == "MANUAL"){
  // Obté la dada amb l'acció a realitzar
  Firebase.RTDB.getString(&fbdo, (path + "/accio-manual").c_str());
  accio = fbdo.stringData();

  // Si l'acció és baixar, el motor gira en sentit horari
  if(accio == "baixar"){
    // Configura motor per tal que giri en sentit horari
    persiana.write(0);
    // Espera fins que es compleixin els 4 segons o es rebí un senyal
    per parar/pujar
    while (millis() - tempsInicil < 4000) {
      // Verifica si s'ha rebut un senyal des de la base de dades per
      parar/pujar
      if (Firebase.RTDB.getString(&fbdo, (path + "/accio-
manual").c_str())) {
        accio = fbdo.stringData();
        // Si s'ha rebut un senyal per parar...
        if (accio == "parar") {
          // ...atura el motor i el rellotge
          persiana.write(90);
          fiTemps1 = false;
          break;
        }
        // Si s'ha rebut un senyal per pujar...
        else if (accio == "pujar") {
          // ...canvia el sentit de gir i atura el rellotge
          persiana.write(180);
          fiTemps1 = false;
          break;
        }
      }
    }
    // Si es compleix el temps, atura el motor i escriu "parar"
    if (fiTemps1 == true) {
      Firebase.RTDB.setString(&fbdo, (path + "/accio-manual").c_str(),
"parar");
      persiana.write(90);
    }
  }
}
```

Com es pot observar, el codi contempla que l'usuari realitzi canvis d'estat mentre s'està realitzant, en aquest cas, la baixa de la persiana inicialment especificada per ell mateix. Si durant els 4 segons que realitza el moviment no hi ha cap petició nova, un cop passat el temps s'atura i escriu l'estat en la base de dades. Si es llegeix un canvi en el moviment desitjat s'executa aquest canvi sobre el servomotor.

Per les peticions de baixada i aturada es realitza el mateix procés però actualitzant les instruccions del servomotor.

Memòria

- Funció que realitza la lectura de la base de dades per obtenir l'estat (ON/OFF) de la depuradora desitjat per l'usuari. En cas que hi hagi una petició d'engegada, activa el motor i en cas contrari, l'atura.

Es passa com a paràmetre l'adreça corresponent al node on s'emmagatzema l'estat d'aquest dispositiu.

```
void activarDepuradora (String path) {
  String nextEstat = "";
  // Obté valor de la base de dades (ON/OFF)
  if (Firebase.RTDB.getString(&fbdo, (path).c_str())) {
    nextEstat = fbdo.stringData();
    Serial.println();
    Serial.print("Llegint valor: " + nextEstat);
    Serial.println(" de la següent adreça: " + path);
    Serial.println();

    if(nextEstat=="ON"){
      // Engega el motor de la depuradora
      digitalWrite(pinDepuradora, HIGH);
    }
    else{
      // Apaga el motor de la depuradora
      digitalWrite(pinDepuradora, LOW);
    }
  }
  else {
    // Indica el motiu de l'error en la lectura de la database
    Serial.println("ERROR");
    Serial.println("CAUSA: " + fbdo.errorReason());
  }
}
```

Amb tot el conjunt de funcions definides, el següent bloc del codi és la configuració inicial. A part del codi per configurar la connexió amb la xarxa Wi-Fi i amb la base de dades, es realitzen les següents inicialitzacions.

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

  // Inicialitza sensor intensitat llum
  pinMode(pinSensorLlum_Menj, INPUT);
  pinMode(pinSensorLlum_Hab, INPUT);

  // Inicialitza servomotors persianes
  persiana1.attach(pinPersiana1);
  persiana1.write(90);
  persiana2.attach(pinPersiana2);
  persiana2.write(90);
}
```

Memòria

```
// Inicialitza motor depuradora
pinMode(pinDepuradora, OUTPUT);

// Configuració del Wi-Fi i de la base de dades vista anteriorment

// Adreces als diferents elements de la database
persianaPath = databasePath + "/persianes";
depuradoraPath = databasePath + "/depuradora";
}
```

Per finalitzar el codi, el bucle principal del programa executa les funcions vistes seguint els intervals de temps definits pel temps de mostreig.

```
void loop() {
  // Espera el temps de mostreig seleccionat
  if (Firebase.ready() && (millis() - millisCicleAnterior > tempsMostreig
  || millisCicleAnterior == 0)){
    millisCicleAnterior = millis();

    // Realitza les mesures dels sensors de llum
    // Entrades analògiques del ESP32 tenen 12 bits de resolució
    // Per tant, es poden obtenir valors entre 0 i 4095
    intensitatLlum_Menj = analogRead(pinSensorLlum_Menj);
    intensitatLlum_Hab = analogRead(pinSensorLlum_Hab);

    // Envia el resultat de la mesura a la base de dades
    enviaLlum(persianaPath + "/persiana-menjador/intensitat-llum",
    intensitatLlum_Menj);
    enviaLlum(persianaPath + "/persiana-hab/intensitat-llum",
    intensitatLlum_Hab);

    // Llegeix dades de la database i gestiona el motor corresponent
    modeFuncPersiana(persianaPath + "/persiana-menjador/mode-func",
    "menj");
    setPersiana(persianaPath + "/persiana-menjador", persiana1,
    modeFunc_Menj, intensitatLlum_Menj);

    modeFuncPersiana(persianaPath + "/persiana-hab/mode-func", "hab");
    setPersiana(persianaPath + "/persiana-hab", persiana2, modeFunc_Hab,
    intensitatLlum_Hab);

    activarDepuradora(depuradoraPath);
  }
}
```

Memòria

○ Codi microcontrolador d'il·luminació:

De la mateixa forma que amb els codis Arduino anteriors, inicialment es realitza la declaració de les llibreries. En aquest cas, s'inclou addicionalment la llibreria per controlar les tires d'il·luminació LED WS2812B o també anomenades NeoPixel.

```
/****** LLIBRERIES & FITXERS *****/  
// Llibreries de connexió al Wi-Fi i a la Firebase Database  
#include <WiFi.h>  
#include <WiFiMulti.h>  
#include <Firebase_ESP_Client.h>  
#include <Wire.h>  
#include "data.h"  
  
// Llibreria tira LED NeoPixel  
#include <Adafruit_NeoPixel.h>
```

En quant a les variables i objectes a definir, s'afegeixen a les vistes anteriorment les corresponents a les diferents tires LED a instal·lar. Aquestes inclouen el pin de connexió, la quantitat de píxels que incorpora cada una, l'adreça al node corresponent de la *database* i la creació de l'objecte de tipus NeoPixel on s'especifiquen les dades de la tira (nombre píxels, pin i freqüència de funcionament).

```
/****** VARIABLES & OBJECTES *****/  
// Objectes Firebase  
FirebaseData fbdo; // Database  
FirebaseAuth auth; // Autenticació  
FirebaseConfig config; // Configuració  
  
// Variable per guardar USER UID  
String uid;  
  
// Variable per guardar les adreces a la database  
String databasePath;  
  
// Objecte Wi-Fi  
WiFiMulti wifiMulti;  
const uint32_t TempsEsperaWifi = 5000;  
  
// Variables temporals (temps de mostreig)  
unsigned long millisCicleAnterior = 0;  
unsigned long tempsMostreig = 2000;  
  
/**** Llums ****/  
//---> Llum menjador  
#define pin_Menj 13  
#define Quantitat_Menj 49  
Adafruit_NeoPixel llum_Menj = Adafruit_NeoPixel(Quantitat_Menj, pin_Menj,  
NEO_GRB + NEO_KHZ800);  
String pathLed_Menj;
```

Memòria

```
//---> Llum jardí
#define pin_Jardi 14
#define Quantitat_Jardi 15
Adafruit_NeoPixel llum_Jardi = Adafruit_NeoPixel(Quantitat_Jardi,
pin_Jardi, NEO_GRB + NEO_KHZ800);
String pathLed_Jardi;

//---> Llum piscina
#define pin_Pisc 12
#define Quantitat_Pisc 7
Adafruit_NeoPixel llum_Pisc = Adafruit_NeoPixel(Quantitat_Pisc, pin_Pisc,
NEO_GRB + NEO_KHZ800);
String pathLed_Pisc;

//---> Llum rebedor (automàtica)
#define pin_Rebedor 5
#define Quantitat_Rebedor 9
Adafruit_NeoPixel llum_Rebedor = Adafruit_NeoPixel(Quantitat_Rebedor,
pin_Rebedor, NEO_GRB + NEO_KHZ800);
String pathLed_Rebedor;

//---> Llum cuina (automàtica)
#define pin_Cuina 4
#define Quantitat_Cuina 15
Adafruit_NeoPixel llum_Cuina = Adafruit_NeoPixel(Quantitat_Cuina,
pin_Cuina, NEO_GRB + NEO_KHZ800);
String pathLed_Cuina;

//---> Llum WC (automàtica)
#define pin_WC 15
#define Quantitat_WC 10
Adafruit_NeoPixel llum_WC = Adafruit_NeoPixel(Quantitat_WC, pin_WC,
NEO_GRB + NEO_KHZ800);
String pathLed_WC;
```

Les funcions que s'encarreguen de gestionar l'estat, intensitat i color de les diferents tires LED són les següents.

- Funció que realitza una lectura de l'estat d'una determinada tira LED i la gestiona en funció d'aquest. Si l'usuari realitza una petició d'engegada a través de la pàgina web, mitjançant la lectura del node corresponent s'obté l'estat desitjat, s'ajusta la intensitat i el color i s'encén la tira LED amb la configuració especificada.

Es passa com a paràmetre l'adreça al node corresponent i l'objecte al qual es vol realitzar aquesta gestió.

Memòria

```
void setLed(String path, Adafruit_NeoPixel &llum){
  String nextEstat ="";
  // Obté el valor de l'estat de la tira LED (ON/OFF)
  if (Firebase.RTDB.getString(&fbdo, (path + "/estat").c_str())){
    nextEstat = fbdo.stringData();
    Serial.print("Llegint valor: " + nextEstat);
    Serial.print(" de la següent adreça: ");
    Serial.println(path + "/estat");

    if(nextEstat=="ON"){
      // Llegeix i configura intensitat de la tira LED
      setLedIntensitat(path + "/intensitat", llum);
      // Llegeix i configura color de la tira LED
      setLedColor(path + "/color", llum);
    }
    else{
      // Apaga tira LED
      llum.clear();
      llum.show();
    }
  }
  else {
    // Indica el motiu de l'error en la lectura de la database
    Serial.println("ERROR");
    Serial.println("CAUSA: " + fbdo.errorReason());
  }
}
```

- o Funció que gestiona l'estat de les llums amb activació automàtica en funció de la lectura del sensor de presència. En el cas que el PIR assignat detecti una presència, s'engega el LED d'aquella zona amb la intensitat i color especificats per l'usuari.

```
void setAutoLed(String path, Adafruit_NeoPixel &llum, int estat){
  // Si detecta presència, encén tira LED
  if(estat == 1){
    //Llegeix i configura intensitat de la tira LED
    setLedIntensitat(path + "/intensitat", llum);
    // Llegeix i configura color de la tira LED
    setLedColor(path + "/color", llum);
  }
  else{
    // Apaga tira LED
    llum.clear();
    llum.show();
  }
}
```

Memòria

- Funció que realitza una lectura en la base de dades del valor de la intensitat desitjada per l'usuari per una determinada llum. Es passa com a paràmetre l'adreça corresponent a la tira LED que es desitja ajustar i l'objecte associat.

Un cop realitzada la lectura, s'aplica un factor de conversió al valor per passar-lo escala de 0 a 100 a una escala de 0 a 255. Donat que el valor de la intensitat està representat per 8 bits, aquest rang de 0 fins 255 és el nombre total de combinacions possibles ($2^8 = 256$), sent el valor 255 la màxima intensitat que pot tenir la tira LED. Finalment, mitjançant la instrucció pròpia de la llibreria NeoPixel s'ajusta la intensitat de la tira amb *setBrightness (valor)*.

```
void setLedIntensitat(String path, Adafruit_NeoPixel &llum) {
  // Obté valor de la base de dades
  if (Firebase.RTDB.getInt(&fbdo, path.c_str())) {
    Serial.println("LED Intensitat: " + String(fbdo.intData()));

    // Factor de conversió de 0-100 a 0-255
    int intensitat = fbdo.intData() * 2.55;
    // Ajusta la intensitat del LED
    llum.setBrightness(intensitat);
  }
  else {
    // Indica el motiu de l'error en la lectura de la database
    Serial.println("ERROR");
    Serial.println("CAUSA: " + fbdo.errorReason());
  }
}
```

- Funció que realitza una lectura en la base de dades dels valors RGB indicadors del color desitjat per l'usuari per una determinada llum. Es passa com a paràmetre l'adreça corresponent a la tira LED que es desitja ajustar i l'objecte associat.

Les diferents components del color s'obtenen per separat i s'emmagatzemen en les variables de tipus enter corresponents. Posteriorment, es realitza un escombrat per cada píxel que forma la tira aplicant aquest color RGB a cada un d'ells i, finalment, s'encén el LED.

```
void setLedColor(String path, Adafruit_NeoPixel &llum) {
  // Adreça per a la lectura de cada valor del color (RGB)
  String path_r = path + "/r";
  String path_g = path + "/g";
  String path_b = path + "/b";
  // Llegeix el valor R del color i l'emmagatzema com a int
  if (Firebase.RTDB.getInt(&fbdo, path_r.c_str())) {
    Serial.println("LED color R: " + String(fbdo.intData()));
    int r = fbdo.intData();
    // Llegeix el valor G del color i l'emmagatzema com a int
    if (Firebase.RTDB.getInt(&fbdo, path_g.c_str())) {
      Serial.println("LED color G: " + String(fbdo.intData()));
      int g = fbdo.intData();
    }
  }
}
```

Memòria

```
// Llegeix el valor B del color i l'emmagatzema com a int
if (Firebase.RTDB.getInt(&fbdo, path_b.c_str())) {
    Serial.println("LED color B: " + String(fbdo.intData()));
    int b = fbdo.intData();

    // Configura cada píxel de la tira LED amb el color seleccionat
    uint32_t color = llum.Color(r, g, b);
    for (int i = 0; i < llum.numPixels(); i++) {
        llum.setPixelColor(i, color);
    }
    // Encén la tira LED
    llum.show();
}
}
else {
    // Indica el motiu de l'error en la lectura de la database
    Serial.println("ERROR");
    Serial.println("CAUSA: " + fbdo.errorReason());
}
}
```

- Funció que realitza la lectura de la base de dades per obtenir els valors resultants de la mesura dels sensors de presència. Es passa com a paràmetre l'adreça d'accés al node del sensor corresponent i retorna el valor de la lectura.

```
int llegirPresencia (String path){
    int valor = 0;
    // Obté el valor de la base de dades i el guarda com a int
    if (Firebase.RTDB.getInt(&fbdo, path.c_str())){
        valor = fbdo.intData();
        Serial.print("Llegint valor: ");
        Serial.print (valor);
        Serial.print(" de la següent adreça: ");
        Serial.println(path);
    }
    else {
        // Indica el motiu de l'error en la escriptura a la database
        Serial.println("ERROR");
        Serial.println("CAUSA: " + fbdo.errorReason());
    }
    return valor;
}
```

Amb tot el conjunt de funcions definides, el següent bloc del codi és la configuració inicial. A part del codi per configurar la connexió amb la xarxa Wi-Fi i amb la base de dades, es realitzen les següents inicialitzacions.

Memòria

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

  // Configuració del Wi-Fi i de la base de dades vista anteriorment

  // Adreces als diferents elements de la database
  pathLed_Menj = databasePath + "/llums" + "/llum-menjador";
  pathLed_Jardi = databasePath + "/llums" + "/llum-jardi";
  pathLed_Pisc = databasePath + "/llums" + "/llum-piscina";
  pathLed_Rebedor = databasePath + "/llums" + "/llum-rebedor";
  pathLed_Cuina = databasePath + "/llums" + "/llum-cuina";
  pathLed_WC = databasePath + "/llums" + "/llum-wc";
}
```

Per acabar, en cada iteració del bucle principal que s'executa en intervals iguals al temps de mostreig, s'obtenen les dades dels sensors PIR i es gestionen les diferents tires LED utilitzant les funcions vistes depenent si tenen un control manual o automàtic.

```
void loop() {
  // Espera el temps de mostreig seleccionat
  if (Firebase.ready() && (millis() - millisCicleAnterior > tempsMostreig
  || millisCicleAnterior == 0)){
    millisCicleAnterior = millis();

    // Obté les dades de presència de la database
    int pir_estat_rebedor = llegirPresencia(pathLed_Rebedor + "/estat");
    int pir_estat_cuina = llegirPresencia(pathLed_Cuina + "/estat");
    int pir_estat_wc = llegirPresencia(pathLed_WC + "/estat");

    // Gestiona la il·luminació LED automàtica
    setAutoLed(pathLed_Rebedor, llum_Rebedor, pir_estat_rebedor);
    setAutoLed(pathLed_Cuina, llum_Cuina, pir_estat_cuina);
    setAutoLed(pathLed_WC, llum_WC, pir_estat_wc);

    // Llegeix dades de la database i gestiona el LED corresponent
    setLed(pathLed_Menj, llum_Menj);
    setLed(pathLed_Jardi, llum_Jardi);
    setLed(pathLed_Pisc, llum_Pisc);
  }
}
```

1.7.6. Cost de construcció del prototip

En aquest apartat s'analitzen els costos associats a la implementació del sistema domòtic dissenyat. Aquesta anàlisi té com a objectiu proporcionar una visió clara i comprensiva dels costos associats en la fase inicial de prototipat.

En primer lloc, es detalla el cost dels components i materials utilitzats per construir la maqueta funcional del sistema domòtic. Aquesta serveix com a versió simplificada i reduïda del sistema final, dissenyada per demostrar la viabilitat tècnica i funcional del projecte.

Dispositiu	Unitats	Preu unitari	Preu total	Enllaç de compra
Microcontrolador ESP-WROOM-32	3	8,33 €	24,99 €	[35]
Servomotor continu	4	7,49 €	29,98 €	[36]
Tira LED RGB WS2812B IP30, 5m i 30 píxels/m	1	31,99 €	31,99 €	[37]
Mòdul sensor DHT22	1	9,49 €	9,49 €	[38]
Mòdul sensor detector de flama	3	1,99 €	5,99 €	[39]
Mòdul brunzidor actiu	3	1,33 €	3,99 €	[40]
Sensor d'infrarojos passiu (detector de presència)	5	2,79 €	13,99 €	[41]
Mòdul fotoresistor 10 k Ω	3	1,83 €	5,49 €	[42]
Bomba hidràulica 5 V, 1 m canonada PVC i mòdul relé 5 V d'1 canal	1	9,99 €	9,99 €	[43]
Cost total:			135,90 €	

Taula 16. Costos associats als components i dispositius utilitzats pel prototip

Material	Unitats	Preu unitari	Preu total	Enllaç de compra
Pins Dupont mascle 2,54 mm de pas	1	8,99 €	8,99 €	[44]
Pins Dupont femella 2,54 mm de pas	1	8,99 €	8,99 €	[45]
Connector LED de 3 pins i 10 mm, 5 m de cable d'extensió tira LED	1	9,99 €	9,99 €	[46]
6 bobines de 10 m de cable sòlid, 0,13 mm ² de secció	2	18,99 €	37,98 €	[47]

Làmina cartró-ploma 100x70 cm, gruix 5 mm	2	12,30 €	24,60 €	[48]
Taulell fusta MDF 61x122x0,3 cm	2	6,29 €	12,58 €	[49]
Llistó de fusta 13x63 mm x 2,4 m	3	5,29 €	15,87 €	[50]
Pintura blanc os mate	2	6,59 €	13,18 €	[51]
Pintura gris mate	2	6,59 €	13,18 €	[52]
Cost total:			145,36 €	

Taula 17. Costos associats als materials utilitzats pel prototip

En total, el cost associat a la construcció del prototip funcional del sistema domòtic dissenyat amb l'objectiu de demostrar la viabilitat del projecte és de 281,26 €. Aquest valor és el resultat de la suma entre els costos associats a dispositius i sensors i els costos associats a la compra de material.

1.7.7. Aplicació en un entorn real

El sistema domòtic dissenyat en el desenvolupament d'aquest projecte correspon a la implementació de forma experimental en un habitatge amb les característiques vistes. Per aquest motiu, en cas que es vulgui implementar en un entorn real, cal considerar diversos factors i realitzar ajustaments per assegurar que el sistema domòtic funcioni de manera òptima en la vida quotidiana dels usuaris.

Els canvis i ajustaments necessaris són els següents:

1. Escalabilitat del sistema:

- Hardware i software: cal assegurar que tant la part física com el codi siguin escalables per gestionar un major nombre de dispositius i sensors. Per tant, cal considerar l'ús d'un major nombre de microcontroladors ESP32 agrupant en cada un d'ells la gestió d'un determinat tipus de dispositius i una arquitectura de programari modular que permeti l'addició fàcil de noves funcions.
- Capacitat de xarxa: cal implementar una xarxa Wi-Fi domèstica robusta i específica pel sistema domòtic. Aquesta ha de ser robusta, amb una suficient amplada de banda i cobertura per suportar tots els dispositius connectats. Aquesta implementació pot incloure l'ús de repetidors o sistemes de malla per assegurar una bona connexió en tota la superfície.

2. Robustesa i fiabilitat:

- Components de qualitat: cal utilitzar components de qualitat industrial certificada per assegurar la durabilitat i fiabilitat del sistema a llarg termini, tot i que suposi un augment en el cost de la instal·lació.
- Tolerància a fallades: cal implementar un mecanisme de redundància i tolerància a fallades com per exemple la introducció de fonts d'alimentació ininterrompuda per assegurar que el sistema funcioni en cas de talls en el subministrament elèctric.

3. Seguretat:

- Ciberseguretat: cal incorporar mesures de ciberseguretat robustes per protegir el sistema de comunicació via xarxa contra accessos no autoritzats. Aquesta mesura pot incloure l'ús de protocols de xifratge, autenticació de doble factor i actualitzacions de seguretat periòdiques.
- Protecció física: cal assegurar que els components del sistema estiguin adequadament protegits contra danys i manipulacions no autoritzades, per exemple mitjançant la instal·lació de carcasses protectores i l'ús de fixacions de seguretat per als dispositius situats a l'exterior.

4. Facilitat d'ús i accessibilitat:

- Assistència i suport: cal proporcionar manuals de funcionament detallats i un servei de suport tècnic per ajudar als usuaris a resoldre qualsevol problema que puguin trobar.

5. Manteniment i actualitzacions:

- Manteniment regular: cal establir un pla de manteniment regular per assegurar que tots els dispositius del sistema funcionin de forma correcta. Aquesta pot incloure la neteja de sensors, revisió de connexions, etc.
- Actualitzacions de software: cal proporcionar actualitzacions de programari periòdiques per millorar les prestacions, corregir possibles errors o assegurar la compatibilitat amb noves tecnologies.

6. Compliment normatiu:

- Regulacions i normatives: cal assegurar que el sistema compleixi totes les regulacions i normatives locals en matèria de seguretat, emissions i compatibilitat electromagnètica.
- Certificacions: cal obtenir les certificacions necessàries per garantir que el sistema és segur i fiable per al seu ús en un entorn residencial.

Memòria

Finalment, per tal de tenir una visió més clara del que suposaria econòmicament per a l'usuari implementar en un habitatge real un sistema domòtic com el dissenyat, es mostra el pressupost bàsic de la domotització de nivell 1 que considera tots els aspectes de la instal·lació. Cal destacar que aquests costos són aproximats i poden sofrir variacions en funció de la qualitat dels components, ubicació geogràfica, etc.

Aquest pressupost s'ha realitzat sobre un habitatge que segueix les característiques vistes al llarg del projecte. La superfície interior d'aquest es troba dividida en un total de 7 estances: rebedor, menjador, passadís, cuina, dues habitacions i un bany.

Per tal de generar el pressupost s'ha utilitzat la *Base de datos de construcción* de l'Institut Valencià de l'Edificació [53]. Aquest *software*, actualitzat el juliol de l'any 2023, permet seleccionar la comunitat autònoma on s'hi vol pressupostar i el tipus d'instal·lació, de tal forma que s'ha realitzat un pressupost per una instal·lació domòtica a Catalunya.

Consultar les seccions d'AMIDAMENTS i de PRESSUPOST situades al final del document per veure amb detall tots els costos associats a la instal·lació d'aquest sistema domòtic de nivell 1 en una casa com la tractada al llarg del projecte.

La realització d'aquest pressupost garanteix una visió clara i precisa dels recursos necessaris per dur a terme el projecte en un entorn real. La inversió total reflecteix el compromís per assegurar una instal·lació robusta, fiable i segura, capaç de proporcionar els beneficis esperats en termes de comoditat, eficiència energètica i seguretat. D'aquesta forma, es confirma la viabilitat del sistema domòtic desenvolupat, establint una base sòlida per a futures implementacions i expansions del projecte en entorns residencials.

1.7.8. Extensions i millores del projecte

Aquest apartat proposa una sèrie de millores i ampliacions que es poden incorporar tant a la part física del sistema domòtic com a la interfície web amb l'objectiu d'augmentar les prestacions del disseny i la seguretat dels usuaris que es beneficien dels seus avantatges.

Cal destacar que aquestes propostes de millora es troben fora de l'abast d'aquest projecte i que no s'ha considerat la seva implementació en el mateix. Tan sols es basen en algunes de les necessitats recollides durant la fase de desenvolupament del prototip inicial, així com en les tendències actuals del mercat domòtic.

Les extensions suggerides són les següents:

- **Permisos d'usuari:** implementar un sistema de permisos que defineixi què pot fer cada usuari en funció del rang que hi té assignat. La introducció d'aquesta gestió permetria una major flexibilitat i seguretat ja que diferents usuaris podrien tenir accés a diferents funcionalitats del sistema. Per exemple, aplicar limitacions d'accés al control de determinats dispositius als usuaris associats a nens.
- **Centralització de dispositius de seguretat:** agrupar en un únic microcontrolador tots els dispositius la funcionalitat dels quals es troba en l'àmbit de la seguretat domèstica. Mitjançant la centralització mencionada i l'aplicació de mesures més restrictives en l'accés a la comunicació d'aquests dispositius, es reforçaria la seguretat de l'habitatge en protegir-lo de possibles intrusions a informació sensible.
- **Integració del sistema amb assistents de veu:** connectar el sistema domòtic amb els assistents de veu més coneguts del mercat com Amazon Alexa, Google Assistant o Apple Siri. Aquesta millora permetria als usuaris controlar el sistema domòtic mitjançant el comandament dels dispositius a través de la veu, augmentant així significativament la comoditat i l'accessibilitat del sistema.
- **Addició o eliminació de dispositius:** desenvolupar en la pàgina web una funcionalitat que permeti afegir o eliminar dispositius de la pàgina web. Per una banda, aquesta inclouria la configuració automàtica dels nous dispositius i la seva integració al sistema existent sense necessitat de programació manual per part de l'usuari. D'altra banda, inclouria també l'eliminació de la interfície web d'aquells que ja no interessa controlar o ja no formen part del sistema.
- **Monitorització energètica:** implementar un sistema de monitorització del consum energètic de la llar, permetent així als usuaris veure en temps real la quantitat d'energia consumida pels diferents dispositius.
- **Escenaris personalitzats:** crear la possibilitat de programar escenaris personalitzats, on diverses accions es poden activar simultàniament amb un sol comandament. Per exemple, un escenari de "Bona nit" en el qual es tanquen totes les llums, es baixen totes les persianes i s'activa l'alarma.

1.8. Conclusions

Aquest projecte de disseny i implementació d'una casa domòtica controlada mitjançant una pàgina web ha abordat diverses fases, des del desenvolupament de la idea inicial fins a la creació d'un prototip funcional. L'objectiu principal ha estat demostrar la viabilitat d'integrar sistemes domòtics en entorns residencials utilitzant tecnologies modernes com l'Internet de les Coses i bases de dades en temps real. A continuació, es detallen els objectius globals definits a l'inici del projecte i com s'ha complert al llarg del desenvolupament d'aquest.

1. Integració del microcontrolador ESP32:

S'ha utilitzat aquest microcontrolador com a eina principal per a la gestió eficient dels dispositius domòtics incorporats en l'habitatge dissenyat, assegurant en tot moment la compatibilitat i la connectivitat necessàries pel correcte funcionament del sistema.

2. Programació en l'entorn Arduino IDE:

S'ha dissenyat el codi necessari per als microcontroladors utilitzant llenguatge C a través de l'entorn de desenvolupament Arduino, garantint d'aquesta forma un accés fiable a les diferents llibreries existents i una programació segura per als diferents dispositius incorporats.

3. Creació d'una interfície web interactiva:

S'ha desenvolupat una pàgina web combinant la programació a través dels llenguatges HTML, CSS i JavaScript, accessible des de qualsevol dispositiu amb connexió a Internet i que permet als usuaris registrats controlar de manera intuïtiva els diferents dispositius domòtics incorporats i l'estat del seu habitatge.

4. Disseny i personalització de la interfície web:

A través de la investigació del gran ventall de possibilitats que ofereix el llenguatge CSS, s'ha creat una pàgina web atractiva i molt fàcil d'utilitzar, proporcionant així una experiència a l'usuari satisfactòria.

5. Utilització de Firebase Realtime Database:

S'ha implementat l'eina de Google per crear una base de dades en temps real capaç de gestionar les dades provinents dels diferents dispositius i de la pàgina web de manera eficient i ordenada, garantint en tot moment una sincronització ràpida entre els dispositius física i la interfície web.

6. Incorporació de varietat en els dispositius:

S'han utilitzat molts i diversos dispositius físics per tal de poder abordar totes les àrees possibles en les quals la domòtica hi té cabuda en un entorn domèstic, permetent així una gestió total de l'habitatge i un aprenentatge més ampli sobre els principis de funcionament dels diferents elements.

7. Construcció d'un prototip funcional de la casa:

S'ha implementat un petit prototip de la casa domòtica per tal de demostrar la viabilitat i el correcte funcionament del sistema dissenyat i de totes les possibles interaccions que l'usuari pot realitzar a través de la pàgina web.

El grau de compliment de tots els objectius marcats a l'inici del projecte ha estat altament satisfactori. En les següents conclusions, es busca oferir una visió global i crítica dels resultats obtinguts i del potencial impacte de la domòtica en la vida quotidiana.

La domòtica ofereix una sèrie de beneficis significatius orientats principalment a millorar la qualitat de vida dels usuaris. En primer lloc, proporciona un alt nivell de confort mitjançant l'automatització de tasques quotidianes com el control de la il·luminació, la climatització o dels electrodomèstics. Aquesta automatització facilita la vida quotidiana, reduint la necessitat d'intervenció manual constant i la preocupació generada per la realització de diverses tasques. Addicionalment, la domòtica ajuda a millorar la seguretat de la llar mitjançant sistemes de vigilància intel·ligents connectats a la xarxa que es poden monitoritzar i controlar remotament i de forma centralitzada. Un altre benefici destacable i que dia a dia pren més importància, és l'eficiència energètica, ja que els sistemes domòtics permeten una gestió òptima del consum energètic, contribuint així a la reducció del consum de les fonts d'energia, de la facturació associada a aquestes i de l'impacte ambiental.

En l'actualitat, la domòtica és una tendència creixent a nivell global. Amb l'avenç de l'Internet de les Coses, els dispositius connectats són cada vegada més comuns a les llars, oferint noves possibilitats d'interacció i control. La integració de la intel·ligència artificial en aquests sistemes permet una adaptació personalitzada a les necessitats dels usuaris, aprenent dels seus hàbits i preferències per oferir una experiència més eficient i còmoda. La demanda de llars intel·ligents està augmentant, impulsada per la recerca de solucions que combinin comoditat, seguretat i sostenibilitat.

El sistema domòtic dissenyat en aquest projecte es basa en la utilització del microcontrolador ESP32 com a peça central per a la gestió i control dels dispositius. Aquest es caracteritza per la seva robustesa i per la incorporació de tecnologies de connectivitat Wi-Fi i Bluetooth, permetent una comunicació eficient i fiable. La interfície web, desenvolupada amb HTML, CSS i JavaScript, permet als usuaris controlar i monitoritzar de forma interactiva gran diversitat de dispositius domèstics. El sistema permet una comunicació fiable i en temps real entre els dispositius i la pàgina web, gràcies a la implementació de Firebase Realtime Database, una base de dades en el núvol que facilita la sincronització instantània de les dades.

La viabilitat del sistema ha estat demostrada a través de la construcció d'un prototip funcional, que integra amb èxit els diferents components de maquinari i programari. Aquesta configuració assegura una experiència d'usuari intuïtiva i satisfactòria, i és fàcilment escalable per a la seva aplicació en entorns reals més complexos.

L'aplicació d'aquest sistema domòtic en un entorn real ofereix múltiples avantatges pràctics. La capacitat de controlar i monitoritzar la llar des de qualsevol lloc amb accés a internet proporciona als usuaris una gran flexibilitat i tranquil·litat. A més, l'eficiència energètica i la seguretat millorada fan que aquest sistema sigui una opció atractiva per a molts propietaris. La personalització de la interfície web permet adaptar el sistema a les necessitats específiques de cada usuari, garantint una integració harmoniosa amb l'estil de vida de cada família. Així, aquest projecte no només demostra la viabilitat tècnica de la domòtica, sinó també la seva aplicabilitat pràctica i els beneficis reals que pot oferir.

Tal i com s'ha vist, la implementació d'un sistema domòtic comporta una inversió inicial significativa amb un temps d'amortització fixat a llarg termini. A través de l'augment de l'eficiència energètica, el sistema domòtic contribueix a la reducció substancial de la facturació elèctrica i per tant, a la recuperació dels costos inicials fent que la inversió sigui rendible a llarg termini. A més, cal tenir en consideració els beneficis indirectes que suposa la implementació de domòtica com per exemple l'augment del valor de la propietat o la millora en la qualitat de vida dels usuaris, justificant així aquesta despesa inicial.

És important destacar que, tot i els nombrosos beneficis, la implementació de sistemes domòtics també comporta certs reptes. La seguretat de les dades i la privacitat dels usuaris són aspectes crítics que cal abordar amb solucions robustes de ciberseguretat. A més, l'accessibilitat econòmica dels dispositius domòtics pot ser una barrera per a la seva adopció massiva. Tanmateix, amb l'evolució tecnològica i la disminució dels costos, s'espera que aquestes barreres es redueixin progressivament. En conclusió, la domòtica representa una innovació prometedora que està destinada a transformar les nostres llars i la nostra manera de viure, fent-les més intel·ligents, eficients i segures.

2. ANNEXOS

2.1. Taula de càlcul del nivell domòtic

Aplicación domòtica	Dispositivos	Columna de referencia		Usuario
		N.º dispositivos o condición a cumplir	Puntuación	Puntuación
Alarma de intrusión	Detectores de presencia	2	1	
		1 cada 20 m ²	2	
		1 por estancia	3	
	Teclado codificado, llave electrónica o equivalente	1	1	
	Sirena interior	No	0	
		Si	1	
	Contactos de ventanas y/o impactos	En puntos de fácil acceso	1	
		En todas las ventanas	2	
	Sistemas de mantenimiento de alimentación en caso de fallo de suministro eléctrico	No	0	
		Si	2	
	Módulo de habla/escucha, destinado a la escucha en caso de alarma. También se admite cualquier tipo de control que permita conocer si realmente existe un intruso (cámaras web...)	No	0	
		Si	3	
	Sistema conectable con central de alarmas	No	0	
Si		3		
Suma parcial Alarma intrusión				
Alarmas técnicas	Detectores de inundación necesarios en zonas húmedas (baños, cocina, lavadero, garaje)	Los necesarios ¹⁾	1	
	Electroválvula de corte de agua con instalación para "bypass" manual.	Las necesarias ¹⁾	1	
	Detectores de concentraciones de gas butano y/o natural en zonas donde se prevea que habrá elementos que funcionen con gas.	Los necesarios ¹⁾	1	
	Electroválvula de corte de gas con instalación para "bypass" manual.	Las necesarias ¹⁾	1	
	Detector de incendios	1 en cocina	1	
		1 cada 30 m ²	2	
En todas las estancias		3		
Suma parcial Alarmas técnicas				

Annexos

Simulación de presencia		No	0	
		Relacionada con las persianas motorizadas o los puntos de luz	2	
		Relacionada con las persianas motorizadas y los puntos de luz	3	
Suma parcial Simulación de presencia				
Videoportero		No	0	
		Si	1	
Suma parcial Videoportero				
Control de persianas	Motorización y control de persianas	Todas las de superficie superior a 2 m ²	1	
		Todas	2	
Suma parcial Control de persianas				
Control de iluminación	Regulación lumínica con control de escenas	No	0	
		En dependencias dedicadas al ocio	2	
		En salón y dormitorios	3	
	En jardín o grandes terrazas mediante interruptor crepuscular o interruptor horario astronómico	No	0	
		Si	2	
	Conexión/desconexión general de iluminación	Un acceso	1	
		Todos los accesos	2	
	Control de puntos de luz y tomas de corriente más significativas	No	0	
		50% puntos de luz	2	
80% puntos de luz + 20% tomas de corriente		3		
Suma parcial Control de iluminación				
Control de clima	Cronotermostato	1 en salón	1	
		Zonificando la vivienda en un mínimo de dos zonas	2	
		Zonificando la vivienda por estancias	3	
Suma parcial Control de clima				
Programaciones	Posibilidad de realizar programaciones horarias sobre los equipos controlados	No	0	
		Si	2	
	Gestor energético	No	0	
		Si	2	
Suma parcial Programaciones				
Interfaz de usuario	Consola o equivalente	No	0	
		Si	2	
	Control telefónico bidireccional	Si	1	
		Interacción mediante SMS	2	
	Equipo para control a través de Internet, Wap o equivalente	No	0	
Si		3		
Suma parcial Interfaz de usuario				

Annexos

Dispositivos conectables a empresas suministradoras a través de redes de comunicación		1	1	
		2	2	
		3 o más	3	
Suma parcial Dispositivos conectables a empresas suministradoras				
Red Multimedia	Tomas satélites y tomas multimedia	No	0	
		3 SAT + 3 multimedia	2	
		3 SAT + 1 multimedia en todas las estancias, incluido terraza	3	
	Punto de acceso inalámbrico	No	0	
Wi-Fi		1		
Suma parcial Red Multimedia				
SUMA TOTAL				
Número de aplicaciones domóticas cubiertas				

2.2. Codi HTML

```

<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <title>My Home</title>

    <!--Llibreria Highcharts (gràfics)-->
    <script src="https://code.highcharts.com/highcharts.js"></script>
    <script src="https://cdn.rawgit.com/Mikhus/canvas-gauges/gh-
pages/download/2.1.7/all/gauge.min.js"></script>

    <!--Versió de Firebase -->
    <script src="https://www.gstatic.com/firebasejs/8.8.1/firebase-
app.js"></script>
    <script src="https://www.gstatic.com/firebasejs/8.8.1/firebase-
auth.js"></script>
    <script src="https://www.gstatic.com/firebasejs/8.8.1/firebase-
database.js"></script>
    <!-- Configuració web app Firebase -->
    <script>
      const firebaseConfig = {
        apiKey: "XXXXXXXXXXXXXXXXXX",
        authDomain: "XXXXXXXXXXXXXXXXXX",
        databaseURL: "XXXXXXXXXXXXXXXXXX",
        projectId: "XXXXXXXXXXXXXXXXXX",
        storageBucket: "XXXXXXXXXXXXXXXXXX",
        messagingSenderId: "XXXXXXXXXXXXXXXXXX",
        appId: "XXXXXXXXXXXXXXXXXX",
        measurementId: "XXXXXXXXXXXXXXXXXX"
      };
      <!-- Inicialitza Firebase -->
      firebase.initializeApp(firebaseConfig);
      <!-- Crea les referència per a l'autenticació i la database -->
      const auth = firebase.auth();
      const db = firebase.database();
    </script>

    <!--Adreces als arxius d'estil CSS-->
    <link rel="stylesheet" type="text/css" href="styles/login-form-
style.css">
    <link rel="stylesheet" type="text/css" href="styles/nav-bar-
style.css">
    <link rel="stylesheet" type="text/css" href="styles/main-style.css">
  </head>

```

Annexos

```
<body>
  <!--FORMULARI D'INICI DE SESSIÓ-->
  <!--Contenidor amb el formulari-->
  <div class="container-form" id="container-form">
    <form id="login-form" style="display: none;">
      <!--Nom pàgina web-->
      <div class="nom-pagina" >
        <ion-icon id="home" name="home-outline"></ion-icon>
        <span>My Home</span>
      </div>
      <!--Contenidor pel correu electrònic-->
      <div class="container-input">
        <ion-icon name="mail-outline"></ion-icon>
        <input type="text" placeholder="Correu electrònic" id="email-
          entrada" required>
      </div>
      <!--Contenidor per la contrasenya-->
      <div class="container-input">
        <ion-icon name="lock-closed-outline"></ion-icon>
        <input type="contrasenya" placeholder="Contrasenya"
          id="contrasenya-entrada" required>
      </div>
      <!--Botó per iniciar la sessió-->
      <button class="button" id="login-button">INICIAR SESSIÓ</button>
      <p id="missatge-error" style="color:red;"></p>
    </form>
  </div>

  <!--CONTINGUT PÀGINA PRINCIPAL-->
  <!--Únicament es mostra quan s'inicia la sessió satisfactòriament-->
  <div class="contingut-sign-in" id="contingut-sign-in">

  <!--BARRA NAVEGACIÓ LATERAL-->
  <div class="barra-lateral">
    <!--Nom i icona de la pàgina-->
    <div>
      <div class="nom-pagina" id="link-home">
        <ion-icon id="home" name="home-outline"></ion-icon>
        <span>My Home</span>
      </div>
    </div>
  </div>

  <!-- Seccions pàgina -->
  <nav class="navegacio">
    <ul>
      <!--Secció llums interiors-->
      <a href="#" id="link-llums-interior">
```

Annexos

```
<!--Nom i icona-->
<ion-icon name="bulb-outline"></ion-icon>
<span>Llums interior</span>
</a>
<!--Secció llums exteriors-->
<a href="#" id="link-llums-exterior">
  <!--Nom i icona-->
  <ion-icon name="bonfire-outline"></ion-icon>
  <span>Llums exterior</span>
</a>
<!--Secció persianes-->
<a href="#" id="link-persianes">
  <!--Nom i icona-->
  <ion-icon name="reorder-four-outline"></ion-icon>
  <span>Persianes</span>
</a>
<!--Secció climatització-->
<a href="#" id="link-climatitzacio">
  <!--Nom i icona-->
  <ion-icon name="thermometer-outline"></ion-icon>
  <span>Climatització</span>
</a>
<!--Secció piscina-->
<a href="#" id="link-piscina">
  <!--Nom i icona-->
  <ion-icon name="water-outline"></ion-icon>
  <span>Piscina</span>
</a>
<!--Secció seguretat-->
<a href="#" id="link-seguretat">
  <!--Nom i icona-->
  <ion-icon name="lock-closed-outline"></ion-icon>
  <span>Seguretat</span>
</a>
</ul>
</nav>

<div>
  <!--Línia separació entre seccions i dades de l'usuari-->
  <div class="linia"></div>
  <!--SECCIÓ D'AUTENTICACIÓ (correu i tancar sessió)-->
  <div class="usuari">
    
    <div class="info-usuari">
```

Annexos

```
<!--Mostra el correu electrònic de l'usuari-->
<!--Mostra el botó per tancar la sessió-->
<div id="barra-autenticacio" style="display: none;">
  <span class="email" id="info-usuari">USEREMAIL</span>
  <span class="sortir" href="/" id="logout-link">Tanca
    sessió</span>
</div>
</div>
</div>
</div>
</div>

<!--COS PRINCIPAL DE LA PÀGINA-->
<main>
  <!--Títol de la pàgina i nom de la secció-->
  <h1>MY HOME</h1>
  <h3>Menú principal</h3>
  <!--Graella amb tots els quadres-->
  <div class="graella">

    <!--ALARMA INTRUSOS-->
    <div class="quadre" id="quadre-alarma">
      <!--Nom i icona del quadre-->
      <p><ion-icon name="shield-half-outline"></ion-icon>ALARMA</p>
      <a></a>
      <!--Botó per activar/desactivar alarma i icones-->
      <div class="candados">
        <ion-icon name="lock-open-outline"></ion-icon>
        <label class="switch">
          <input type="checkbox" onchange="toggleAlarma();" id="estat-
            alarma">
          <span class="slider"></span>
        </label>
        <ion-icon name="lock-closed-outline"></ion-icon>
      </div>
      <!--Contenidor d'advertència en cas que hi hagi intrús -->
      <p><span id="pir-estat" style="display: none;"></span></p>
      <div id="pir-estat-container">
        <p id="pir-estat-out"></p>
      </div>
    </div>

    <!--TEMPERATURA & HUMITAT-->
    <div class="quadre" id="quadre-climatitzacio">
      <!--Nom i icona del quadre-->
      <p><ion-icon name="sunny-outline"></ion-icon>CLIMATITZACIÓ</p>
```

Annexos

```
<!--Lectura valor de temperatura-->
<a><ion-icon name="thermometer-outline" style="color:
red"></ion-icon>TEMPERATURA</a>
<p><span class="lectura"><span id="temp"></span>
&deg;C</span></p>

<!--Lectura valor d'humitat-->
<a><ion-icon name="water-outline" style="color: rgb(0, 140,
255)"></ion-icon>HUMITAT</a>
<p><span class="lectura"><span id="hum"></span>
&percent;</span></p>
</div>

<!--ALARMA INCENDIS-->
<div class="quadre" id="quadre-incendis">
  <!--Nom i icona del quadre-->
  <p><ion-icon name="flame-outline"></ion-icon>DETECCIÓ
D'INCENDIS</p>
  <!--Lectura zona del menjador-->
  <a>ZONA MENJADOR</a>
  <!--Contenidor amb el resultat (canvia en funció de la mesura)-->
  <p><span id="estat-foc-menj" style="display: none;"></span></p>
  <div id="estat-foc-container-menj">
    <p id="estat-foc-out-menj"></p>
  </div>
  <!--Lectura zona de la cuina-->
  <a>ZONA CUINA</a>
  <!--Contenidor amb el resultat (canvia en funció de la mesura)-->
  <p><span id="estat-foc-cuina" style="display: none;"></span></p>
  <div id="estat-foc-container-cuina">
    <p id="estat-foc-out-cuina"></p>
  </div>
</div>

<!--CONTROL DE PORTES-->
<div class="quadre" id="quadre-portes">
  <!--Nom i icona del quadre-->
  <p><ion-icon name="key-outline"></ion-icon>CONTROL D'ACCÉS</p>
  <!--Botó per controlar l'estat de la porta de l'exterior-->
  <a>PORTA EXTERIOR</a>
  <div class="candados">
    <ion-icon name="lock-open-outline"></ion-icon>
    <label class="switch">
      <input type="checkbox" onchange="togglePorta('porta-
exterior');" id="estat-porta-ext">
      <span class="slider"></span>
    </label>
    <ion-icon name="lock-closed-outline"></ion-icon>
  </div>
</div>
```

```

<!--Botó per controlar l'estat de la porta de la casa-->
<a>PORTA CASA</a>
<div class="candados">
  <ion-icon name="lock-open-outline"></ion-icon>
  <label class="switch">
    <input type="checkbox" onchange="togglePorta('porta-casa');"
      id="estat-porta-casa">
    <span class="slider"></span>
  </label>
  <ion-icon name="lock-closed-outline"></ion-icon>
</div>
</div>

<!--AIRE CONDICIONAT-->
<div class="quadre" id="quadre-control-temp">
  <!--Nom i icona del quadre-->
  <p><ion-icon name="cog-outline"></ion-icon>CONTROL
    TEMPERATURA</p>
  <!--Selector de temperatura desitjada-->
  <a>TEMPERATURA DESITJADA</a>
  <div class="selector-temp">
    <input type="number" id="select-temp"
      onchange="actualitzaTemp(this.value)" min="16" max="30"
      step="0.5" value="20">
    <output for="select-temp" id="temp-desitjada" style="display:
      none;"></output>
  </div>
  <!--Contenidor amb l'estat de l'aire condicionat-->
  <!--Canvia en funció de la temperatura seleccionada i la
    mesurada-->
  <a>AIRE CONDICIONAT</a>
  <p><span id="ac-estat"></span></p>
  <div id="ac-estat-container">
    <p id="ac-estat-out"></p>
  </div>
</div>

<!--IL·LUMINACIÓ-->
<!--Llum menjador (manual)-->
<div class="quadre" id="quadre-llum-menjador">
  <!--Nom i icona del quadre-->
  <p><ion-icon name="bulb-outline"></ion-icon>LLUM MENJADOR</p>
  <!--Selector d'estat-->
  <label class="switch">
    <input type="checkbox" onchange="toggleLed('menjador');"
      id="estat-menj">
    <span class="slider"></span>
  </label>
  <!--Selector d'intensitat-->

```

Annexos

```
<div class="led-intensitat">
  <a>INTENSITAT</a>
  <input type="range" min="0" max="100" value="25"
    id="intensitat" onchange="actualitzaLedIntensitat
      (this.value, 'menjador')">
  <output for="intensitat" id="intensitat-menj">25 %</output>
</div>
<!--Selector de color-->
<div class="selector-color">
  <a>COLOR</a>
  <input type="color" id="selector-color"
    onchange="actualitzaColor(this.value, 'menjador')" >
  <output for="selector-color" id="color-menj" style="display:
    none;"></output>
</div>
</div>
<!--Llum jardí (manual)-->
<div class="quadre" id="quadre-llum-jardi">
  <!--Nom i icona del quadre-->
  <p><ion-icon name="bulb-outline"></ion-icon>LLUM JARDÍ</p>
  <!--Selector d'estat-->
  <label class="switch">
    <input type="checkbox" onchange="toggleLed('jardi');"
      id="estat-jardi">
    <span class="slider"></span>
  </label>
  <!--Selector d'intensitat-->
  <div class="led-intensitat">
    <a>INTENSITAT</a>
    <input type="range" min="0" max="100" value="25"
      id="intensitat" onchange="actualitzaLedIntensitat
        (this.value, 'jardi')">
    <output for="intensitat" id="intensitat-jardi">25 %</output>
  </div>
  <!--Selector de color-->
  <div class="selector-color">
    <a>COLOR</a>
    <input type="color" id="selector-color"
      onchange="actualitzaColor(this.value, 'jardi')" >
    <output for="selector-color" id="color-jardi" style="display:
      none;"></output>
  </div>
</div>
<!--Llum piscina (manual)-->
<div class="quadre" id="quadre-llum-piscina">
  <!--Nom i icona del quadre-->
  <p><ion-icon name="bulb-outline"></ion-icon>LLUM PISCINA</p>
  <!--Selector d'estat-->
  <label class="switch">
```

Annexos

```
<input type="checkbox" onchange="toggleLed('piscina');"
  id="estat-pisc">
<span class="slider"></span>
</label>
<!--Selector d'intensitat-->
<div class="led-intensitat">
  <a>INTENSITAT</a>
  <input type="range" min="0" max="100" value="25"
    id="intensitat" onchange="actualitzaLedIntensitat
      (this.value, 'piscina')">
  <output for="intensitat" id="intensitat-pisc">25 %</output>
</div>
<!--Selector de color-->
<div class="selector-color">
  <a>COLOR</a>
  <input type="color" id="selector-color"
    onchange="actualitzaColor(this.value, 'piscina') " >
  <output for="selector-color" id="color-pisc" style="display:
    none;"></output>
</div>
</div>
<!--Llum rebedor (automàtica)-->
<div class="quadre" id="quadre-llum-rebedor">
  <!--Nom i icona del quadre-->
  <p><ion-icon name="bulb-outline"></ion-icon>LLUM REBEDOR</p>
  <!--Estat de la llum en funció de la lectura del sensor PIR-->
  <p><span id="pir-estat-rebedor" style="display:
    none;"></span></p>
  <div id="pir-estat-container-reb">
    <p id="pir-estat-out-reb"></p>
  </div>
  <!--Selector d'intensitat-->
  <div class="led-intensitat">
    <a>INTENSITAT</a>
    <input type="range" min="0" max="100" value="25"
      id="intensitat" onchange="actualitzaLedIntensitat
        (this.value, 'rebedor')">
    <output for="intensitat" id="intensitat-rebedor">25 %</output>
  </div>
  <!--Selector de color-->
  <div class="selector-color">
    <a>COLOR</a>
    <input type="color" id="selector-color"
      onchange="actualitzaColor(this.value, 'rebedor') " >
    <output for="selector-color" id="color-rebedor"
      style="display: none;"></output>
  </div>
</div>
```

Annexos

```
<!--Llum cuina (automàtica)-->
<div class="quadre" id="quadre-llum-cuina">
  <!--Nom i icona del quadre-->
  <p><ion-icon name="bulb-outline"></ion-icon>LLUM CUINA</p>
  <!--Estat de la llum en funció de la lectura del sensor PIR-->
  <p><span id="pir-estat-cuina" style="display: none"></span></p>
  <div id="pir-estat-container-cuina">
    <p id="pir-estat-out-cuina"></p>
  </div>
  <!--Selector d'intensitat-->
  <div class="led-intensitat">
    <a>INTENSITAT</a>
    <input type="range" min="0" max="100" value="25"
      id="intensitat" onchange="actualitzaLedIntensitat
        (this.value, 'cuina')">
    <output for="intensitat" id="intensitat-cuina">25 %</output>
  </div>
  <!--Selector de color-->
  <div class="selector-color">
    <a>COLOR</a>
    <input type="color" id="selector-color"
      onchange="actualitzaColor(this.value, 'cuina') " >
    <output for="selector-color" id="color-cuina" style="display:
      none;"></output>
  </div>
</div>
<!--Llum bany (automàtica)-->
<div class="quadre" id="quadre-llum-wc">
  <!--Nom i icona del quadre-->
  <p><ion-icon name="bulb-outline"></ion-icon>LLUM BANY</p>
  <!--Estat de la llum en funció de la lectura del sensor PIR-->
  <p><span id="pir-estat-wc" style="display: none"></span></p>
  <div id="pir-estat-container-wc">
    <p id="pir-estat-out-wc"></p>
  </div>
  <!--Selector d'intensitat-->
  <div class="led-intensitat">
    <a>INTENSITAT</a>
    <input type="range" min="0" max="100" value="25"
      id="intensitat" onchange="actualitzaLedIntensitat
        (this.value, 'wc')">
    <output for="intensitat" id="intensitat-wc">25 %</output>
  </div>
  <!--Selector de color-->
  <div class="selector-color">
    <a>COLOR</a>
    <input type="color" id="selector-color"
      onchange="actualitzaColor(this.value, 'wc') " >
```

Annexos

```
<output for="selector-color" id="color-wc" style="display:
  none;"></output>
</div>
</div>

<!--DEPURADORA-->
<div class="quadre" id="quadre-depuradora">
  <!--Nom i icona del quadre-->
  <p><ion-icon name="cog-outline"></ion-icon>DEPURADORA</p>
  <!--Botó per controlar l'estat de la depuradora-->
  <label class="switch">
    <input type="checkbox" onchange="toggleDepuradora();"
      id="estat-depuradora">
    <span class="slider"></span>
  </label>
</div>

<!--PERSIANES-->
<!--Persiana menjador-->
<div class="quadre" id="quadre-persiana-menjador">
  <!--Nom i icona del quadre-->
  <p><ion-icon name="reorder-four-outline"></ion-icon>PERSIANA
  MENJADOR</p>
  <!--Botó per escollir el mode de funcionament-->
  <div class="mode-func">
    <a>Mode automàtic</a>
    <label class="switch">
      <input type="checkbox" onchange="toggleModePersiana('-
        menjador');" id="mode-persiana-menj">
      <span class="slider"></span>
    </label>
  </div>
  <!--Botons amb les diferents accions-->
  <div class="persiana-botons">
    <button onclick="manualPersiana('pujar', '-menjador')"><ion-
      icon name="chevron-up-circle-outline"></ion-
      icon>Pujar</button>
    <button onclick="manualPersiana('parar', '-menjador')"><ion-
      icon name="close-circle-outline"></ion-icon>Parar</button>
    <button onclick="manualPersiana('baixar', '-menjador')"><ion-
      icon name="chevron-down-circle-outline"></ion-
      icon>Baixar</button>
  </div>
</div>

<!--Persiana habitació-->
<div class="quadre" id="quadre-persiana-hab">
  <!--Nom i icona del quadre-->
  <p><ion-icon name="reorder-four-outline"></ion-icon>PERSIANA
  HABITACÓ</p>
```

Annexos

```
<!--Botó per escollir el mode de funcionament-->
<div class="mode-func">
  <a>Mode automàtic</a>
  <label class="switch">
    <input type="checkbox" onchange="toggleModePersiana('-
      hab');" id="mode-persiana-hab">
    <span class="slider"></span>
  </label>
</div>
<!--Botons amb les diferents accions-->
<div class="persiana-botons">
  <button onclick="manualPersiana('pujar', '-hab')"><ion-icon
    name="chevron-up-circle-outline"></ion-icon>Pujar</button>
  <button onclick="manualPersiana('parar', '-hab')"><ion-icon
    name="close-circle-outline"></ion-icon>Parar</button>
  <button onclick="manualPersiana('baixar', '-hab')"><ion-icon
    name="chevron-down-circle-outline"></ion-icon>Baixar</button>
</div>
</div>
</div>

<!--GRÀFICS CLIMA-->
<!--Temperatura-->
<div class="graella">
  <div class="quadre" id="grafic-temperatura">
    <!--Nom i icona del gràfic-->
    <p><ion-icon name="thermometer-outline" style="color:
      red"></ion-icon>TEMPERATURA (°C)</p>
    <div id="grafic-temp" class="grafic-container"></div>
  </div>
</div>
<!--Humitat-->
<div class="graella">
  <div class="quadre" id="grafic-humitat">
    <!--Nom i icona del gràfic-->
    <p><ion-icon name="water-outline" style="color: #008cff"></ion-
      icon>HUMITAT (%)</p>
    <div id="grafic-hum" class="grafic-container"></div>
  </div>
</div>
</main>
</div>
<!--Adreces als arxius JavaScript i a la pàgina web de les icones-->
<script type="module"
  src="https://unpkg.com/ionicons@7.1.0/dist/ionicons/ionicons.esm.js"></script>
<script nomodule
  src="https://unpkg.com/ionicons@7.1.0/dist/ionicons/ionicons.js"></script>
<script src="scripts/index.js"></script>
<script src="scripts/auth.js"></script>
```

Annexos

```
<script src="scripts/charts-definitions.js"></script>  
</body>  
</html>
```

2.3. Codi d'estil CSS del formulari d'inici de sessió

```
/* Adreça per importar la tipologia de la font */
@import
url('https://fonts.googleapis.com/css2?family=Outfit:wght@500&display=swap');

/* Variables globals (colors) */
:root{
  --color-formulari-login:rgb(255,255,255);
  --color-formulari-cajetin:rgb(224, 224, 224);
  --color-boto:rgb(0, 172, 172);
}

/***** Estils pel formulari d'inici de sessió *****/
/* Estil del contenidor del formulari */
.container-form{
  width: 500px;          /* Mesures */
  height: 350px;
  position: fixed;      /* Posicionament en la pàgina */
  margin-top: 140px;    /* Marges superior i esquerre */
  margin-left: 400px;
  display: flex;        /* Posicionament dels elements interiors */
  justify-content: center;
  background-color: var(--color-formulari-login); /* Color de fons */
  box-shadow: 0 0 10px rgb(0, 0,0,0.3); /* Ombra del quadre */
  border-radius: 15px;  /* Radi de la vora */
  overflow: hidden;     /* Oculta tot allò que sobresurti */
}

/* Estil del nom de la pàgina */
.container-form .nom-pagina{
  height: 50px;         /* Mesura */
  color: var(--color-text);
  margin-top: 40px;     /* Marges superior i inferior */
  margin-bottom: 20px;
  display: flex;        /* Posicionament elements interiors */
  align-items: center;  /* Alineat al centre */
  justify-content: center;
}

/* Estil per als elements de tipus span (títols o encapçalament) */
.container-form .nom-pagina span{
  margin-top: 15px;     /* Marges superior i inferior */
  margin-bottom: 15px;
  font-size: 35px;     /* Mida de la font */
}
```

Annexos

```
/* Estil per a les icones */
.container-form .nom-pagina ion-icon {
    padding: 6px;          /* Espai interior */
    min-width: 50px;      /* Amplada mínima */
    font-size: 40px;      /* Mida de la font */
}

/* Estil per als quadres d'entrada de credencials */
.container-input{
    width: 300px;          /* Mesures */
    height: 40px;
    margin-top: 15px;      /* Marges */
    margin-bottom: 10px;
    margin-left: 100px;
    display: flex;         /* Posicionament elements interior */
    justify-content: center; /* Alineat al centre */
    align-items: center;
    gap: 8px;              /* Espai entre elements */
    padding: 0 15px;       /* Espai interior */
    background-color: var(--color-formulari-cajetin);
}

/* Estil per a l'interior dels quadres d'entrada de credencials*/
.container-input input{
    border: none;          /* Sense vora */
    outline: none;         /* Sense contorn */
    width: 100%;           /* Mesures */
    height: 100%;
    background-color: inherit;
}

/* Estil per al botó d'inici de sessió */
.button{
    width: 170px;          /* Mesures */
    height: 45px;
    margin-left: 160px;    /* Marge esquerre i superior */
    margin-top: 10px;
    font-size: 15px;       /* Mida de la font */
    border: none;
    border-radius: 5px;
    cursor: pointer;       /* Cursor */
    background-color: var(--color-boto);
    color: white;          /* Color del text */
}
```

2.4. Codi d'estil CSS de la barra lateral de navegació

```
/* Adreça per importar la tipologia de la font */
@import
url('https://fonts.googleapis.com/css2?family=Outfit:wght@500&display=swap');

/* Variables per als colors */
:root{
  --color-barra-lateral:rgb(255,255,255);
  --color-text:rgb(0, 0, 0);
  --color-text-menu:rgb(134,136,144);
  --color-menu-hover:rgb(238,238,238);
  --color-menu-hover-text:rgb(0,0,0);
  --color-linia:rgb(180,180,180);
  --color-scroll:rgb(192,192,192);
  --color-scroll-hover:rgb(134,134,134);
}

/***** Estils per a la barra lateral *****/
.barra-lateral{
  position: fixed;
  display: flex;                               /* Disposició flexbox */
  flex-direction: column;                     /* Direcció en columna */
  justify-content: space-between;
  width: 250px;                               /* Mesures */
  height: 100%;
  padding: 20px 15px;                         /* Espai interior */
  background-color: var(--color-barra-lateral);
}

/* Estils per a les etiquetes de la barra lateral */
.barra-lateral span{
  width: 100px;                               /* Amplada */
  white-space: nowrap;                        /* No afegir línia nova si no cap */
  font-size: 18px;                           /* Mida de la font */
  text-align: left;                           /* Text alineat a l'esquerra */
  opacity: 1;
}

/* Estil per a l'espai on està el nom de la pàgina */
.barra-lateral .nom-pagina{
  width: 100%;                               /* Mesures */
  height: 45px;
  color: var(--color-text);
  margin-bottom: 70px;                       /* Marge inferior */
  display: flex;                              /* Disposició flexbox */
  align-items: center;                        /* Alineació al centre */
  cursor: pointer;
}
```

Annexos

```
/* Estil per a la icona de la pàgina */
.barra-lateral .nom-pagina ion-icon{
    min-width: 50px;          /* Amplada mínima */
    font-size: 40px;         /* Mida */
    cursor: pointer;
}
/* Estil per al text del nom de la pàgina*/
.barra-lateral .nom-pagina span{
    margin-left: 5px;        /* Marge esquerre */
    font-size: 25px;        /* Mida de la font */
}

/***** Estils per al menú de navegació (seccions) *****/
.barra-lateral .navegacio{
    height: 100%;           /* Alçada */
    overflow-y: auto;       /* Barra de desplaçament vertical si hi
                             ha desbordament */
    overflow-x: hidden;     /* Sense barra de desplaçament
                             horitzontal */
}
/* Estils per a la barra de desplaçament vertical */
.barra-lateral .navegacio::-webkit-scrollbar{
    width: 5px;            /* Amplada */
}
/* Estils per a l'element mòbil de la barra de desplaçament vertical */
.barra-lateral .navegacio::-webkit-scrollbar-thumb{
    background-color: var(--color-scroll);
    border-radius: 5px;
}
/* Estil per quan el cursor està al damunt */
.barra-lateral .navegacio::-webkit-scrollbar-thumb:hover{
    background-color: var(--color-scroll-hover);
}
/* Estil per als títols de cada secció */
.barra-lateral .navegacio a{
    width: 100%;           /* Mesures */
    height: 45px;
    margin-bottom: 5px;    /* Marge inferior */
    display: flex;         /* Disposició flexbox */
    align-items: center;   /* Alineació al centre */
    text-decoration: none; /* Text sense subratllar */
    border-radius: 10px;
    color: var(--color-text-menu);
}
```

Annexos

```
/* Estil per als títols de cada secció quan el cursor està al damunt */
.barra-lateral .navegacio a:hover{
    background-color: var(--color-menu-hover);
    color: var(--color-menu-hover-text);
}
/* Estil per a les icones de cada secció */
.barra-lateral .navegacio ion-icon{
    min-width: 50px;          /* Amplada mínima */
    font-size: 20px;         /* Mida */
}

/***** Estil per a la línia separadora *****/
.barra-lateral .linia{
    width: 100%;              /* Amplada */
    height: 2px;              /* Altura = gruix */
    background-color: var(--color-linia);
}

/***** Estil per a les dades d'usuari *****/
.barra-lateral .usuari{
    width: 100%;              /* Amplada */
    margin-top: 30px;         /* Marge superior */
    display: flex;           /* Disposició flexbox */
}
/* Estil per a la foto de perfil */
.barra-lateral .usuari img{
    width: 45px;              /* Amplada */
    min-width: 45px;
    border-radius: 10px;
}
/* Estil per a l'espai amb correu electrònic */
.barra-lateral .usuari .info-usuari{
    width: 100%;              /* Amplada */
    margin-left: 10px;        /* Marge esquerra */
}
/* Estil per al text del correu electrònic */
.barra-lateral .usuari .email{
    font-weight: 500;         /* Gruix de la font */
    font-size: 13px;         /* Mida de la font */
}
/* Estil per al botó per tancar la sessió */
.barra-lateral .usuari .sortir{
    font-weight: 700;         /* Gruix de la font */
    font-size: 18px;         /* Mida de la font */
    cursor: pointer;
}
}
```

2.5. Codi d'estil CSS del cos principal de la pàgina

```
/* Adreça per importar la tipologia de la font */
@import
url('https://fonts.googleapis.com/css2?family=Outfit:wght@500&display=swap');

/* Variables per als colors */
:root{
  --color-fondo-body:rgb(235, 235, 235);
  --color-quadre:rgb(255,255,255);
  --color-fondo-switch: rgb(4, 86, 199);
  --color-text:rgb(0,0,0);
}
/* Estils globals */
*{
  margin: 0;          /* Marge exterior */
  padding: 0;        /* Espai interior */
  box-sizing: border-box; /* Model de caixa */
  font-family: 'Outfit', sans-serif; /* Font de la lletra */
}
/* Estil del cos de la pàgina */
body{
  height: 100vh;      /* Mesures */
  width: 100%;
  background-color: var(--color-fondo-body);
  display: flex;      /* Disposició flexbox */
  flex-wrap: wrap;    /* Els elements es disposen en múltiples */
                      /* línies en cas que no hi hagi prou espai */
  justify-content: space-between;
}
/* Estil per al contingut principal */
main{
  margin-left: 250px; /* Marge esquerre */
  padding: 20px;      /* Espai interior */
  text-align: center; /* Centrat horitzontalment */
}
/* Estil per al títol de la pàgina web situat a la part superior */
h1{
  font-size: 25px;    /* Mida de la font */
}
/* Estil per als títol de secció situat a la part superior */
h3{
  margin-top: 15px;   /* Marge superior */
  font-size: 30px;    /* Mida de la font */
}
```

Annexos

```
/***** Estils per a la graella *****/
.graella{
  width: 100%;           /* Mesures */
  height: auto;
  margin-top: 25px;     /* Marge superior */
  display: grid;        /* Disposició en forma de graella */
  grid-gap: 1rem;       /* Espai entre elements de la graella */
  grid-template-columns: 1fr 1fr 1fr; /* Graella de 3 columnes */
}
/* Estil per als quadres */
.quadre{
  width: fit-content;   /* L'amplada s'ajusta al contingut */
  min-width: 300px;
  padding: 20px;        /* Espai interior */
  background-color: var(--color-quadre);
  border-radius: 2%;
  display: flex;         /* Disposició flexbox */
  flex-direction: column; /* Elements disposats en columna */
  align-items: center;   /* Alineació al centre */
  text-align: center;
}
/* Estil per als títols dels quadres */
.graella .quadre p{
  font-size: 20px;      /* Mida de la font */
  font-weight: 600;
  margin-bottom: 10px;  /* Marge inferior */
  display: flex;         /* Disposició flexbox */
  align-content: center; /* Alineació al centre */
  justify-content: center;
}
/* Estil per al text interior dels quadres */
.graella .quadre a{
  font-size: 18px;      /* Mida de la font */
  margin-top: 15px;     /* Marge superior i inferior */
  margin-bottom: 5px;
  display: flex;         /* Disposició flexbox */
  align-content: center; /* Alineació al centre */
  justify-content: center;
}
/* Estil per a les icones dels quadres */
.graella .quadre ion-icon{
  margin-right: 5px;    /* Marge dret */
  min-width: 20px;     /* Mesures */
  font-size: 25px;
}
```

Annexos

```
/* Estil per a les mesures mostrades en els quadres */
.lectura{
    display: flex;                /* Disposició flexbox */
    align-content: center;       /* Alineació al centre */
    font-weight: 100;
}

/***** Estils per als interruptors (toggle switch) *****/
.switch {
    margin-top: 10px;            /* Marge superior i esquerra */
    margin-left: 10px;
    position: relative;         /* Posició relativa */
    display: inline-block;
    width: 60px;                /* Mesures */
    height: 34px;
}
.switch input {
    display: none;              /* No mostrar l'entrada del switch */
}
/* Estil per al lliscador del botó */
.slider {
    position: absolute;         /* Posició absoluta */
    top: 0;                     /* Posicionament */
    left: 0;
    right: 0;
    bottom: 0;
    background-color: #ccc;
    border-radius: 34px;
    opacity: 0.8;
    cursor: pointer;
}
/* Estil per al lliscador quan està en OFF */
.slider:before {
    position: absolute;         /* Posició absoluta */
    content: "";
    height: 26px;              /* Mesures */
    width: 26px;
    left: 4px;                 /* Posicionament */
    bottom: 4px;
    background-color: #fff;
    -webkit-transition: .4s;
    transition: .4s;
    border-radius: 34px
}
/* Estil per al botó quan està en ON */
input:checked+.slider {
    /* Canvia el color de fons */
    background-color: var(--color-fondo-switch);
}
```

Annexos

```
/* Estil del lliscador quan està en ON */
input:checked+.slider:before {
    /* Realitza una translació horitzontal de 26px */
    -webkit-transform: translateX(26px);
    -ms-transform: translateX(26px);
    transform: translateX(26px)
}
/* Estil del botó per configurar el mode de control de les persianes */
.mode-func{
    display: flex;                /* Disposició flexbox */
    align-items: center;         /* Alineació al centre */
    justify-content: center;
}
/* Estil per a les icones de candaus */
.candados{
    display: flex;                /* Disposició flexbox */
    align-items: center;         /* Alineació al centre */
    justify-content: center;
}
.candados ion-icon{
    margin-left: 10px;           /* Marge esquerre */
}

/***** Estils para el selector d'intensitat *****/
.led-intensitat {
    margin-top: 10px;
}
/* Estil per al lliscador de la intensitat */
.led-intensitat input[type="range"] {
    margin-top: 5px;
    width: 150px;                /* Amplada */
}
/* Estil per al valor de la intensitat */
.led-intensitat output {
    font-size: 18px;             /* Mida de font */
    margin-left: 5px;
}

/***** Estils para al selector de colors *****/
.selector-color {
    display: flex;                /* Disposició flexbox */
    flex-direction: column;      /* Direcció en columna */
    align-items: center;         /* Alineació al centre */
    margin-top: 5px;
}
}
```

Annexos

```
/* Estils per a la caixa de selecció de color */
.selector-color input[type="color"] {
    width: 45px;           /* Mesures */
    height: 30px;
    border: none;         /* Sense vora */
    border-radius: 10%;
    cursor: pointer;
}
/***** Estils per als botons de las persianes *****/
.persiana-botons {
    width: 100%;          /* Amplada */
    display: flex;        /* Disposició flexbox */
    flex-direction: column; /* Direcció en columna */
    gap: 1rem;           /* Espai entre botons */
    margin-top: 20px;     /* Marge superior i inferior */
    margin-bottom: 20px;
}
/* Estil per a l'interior del botó */
.persiana-botons button {
    background-color: var(--color-fondo-switch);
    font-size: 15px;     /* Mida de font */
    color: white;
    display: flex;        /* Disposició flexbox */
    align-items: center;  /* Alineació al centre */
    justify-content: center;
    padding: 10px;       /* Espai interior */
    border: none;
    border-radius: 4px;
    cursor: pointer;
}
/* Estil del botó quan el cursor està al damunt */
.persiana-botons button:hover {
    opacity: 0.8;
}
/***** Estils per a l'estat de les llums automàtiques *****/
#pir-estat-container-reb, #pir-estat-container-cuina, #pir-estat-
container-wc{
    width: 100%;          /* Amplada */
    margin-top: 10px;     /* Marge superior */
    padding: 6px;         /* Espai interior */
    border-radius: 15px;
}
/* Estil quan està en ON */
#pir-estat-container-reb.on, #pir-estat-container-cuina.on, #pir-estat-
container-wc.on {
    background-color: var(--color-fondo-switch);
    color: rgb(255, 255, 255);
}
```

Annexos

```
/* Estil quan està en OFF */
#pir-estat-container-reb.off, #pir-estat-container-cuina.off, #pir-estat-
container-wc.off {
    background-color: #ccc;
}
/***** Estils per a l'alerta de foc *****/
#estat-foc-container-menj, #estat-foc-container-cuina {
    width: 100%;                /* Amplada */
    margin-bottom: 12px;        /* Marge inferior */
    padding: 8px;               /* Espai interior */
}
/* Estil per quan NO es detecta foc */
#estat-foc-container-menj.no-foc, #estat-foc-container-cuina.no-foc {
    background-color: #4baf75; /* Color de fons verd */
}
/* Estil per quan es detecta foc */
#estat-foc-container-menj.foc-detectat, #estat-foc-container-cuina.foc-
detectat {
    background-color: #e01b05; /* Color de fons roig */
}
/***** Estils per al selector de temperatura *****/
.selector-temp {
    margin-top: 10px;           /* Marge superior */
}
.selector-temp input[type="number"] {
    width: 75px;                /* Mesures */
    height: 40px;
    text-align: center;         /* Alineació al centre */
    font-size: 18px;           /* Mida de la font */
    border: 1px solid black;    /* Color i arrodoniment de la vora */
}
/***** Estils per a l'estat de l'AC *****/
#ac-estat-container {
    width: 100%;                /* Amplada */
    padding: 6px;               /* Espai interior */
}
/* Estil quan està en ON */
#ac-estat-container.on {
    background-color: var(--color-fondo-switch);
    color: rgb(255, 255, 255);
}
/* Estil quan està en OFF */
#ac-estat-container.off {
    background-color: #ccc;
    color: rgb(0, 0, 0);
}
```

2.6. Codi JavaScript d'autenticació

```

// Espera un canvi en l'estat de l'autenticació
auth.onAuthStateChanged(user => {
  // Si inicia sessió
  if (user) {
    // Mostra a la consola que s'ha iniciat sessió i l'usuari
    console.log("S'ha iniciat la sessió");
    console.log(user);
    // Configura la UI segons l'usuari que ha iniciat sessió
    setupUI(user);
    // Obté l'ID de l'usuari i el mostra en consola
    var uid = user.uid;
    console.log(uid);
  } else {
    // Si tanca sessió, mostra a la consola que l'usuari ha tancat sessió
    console.log("S'ha tancat la sessió");
    // Configura la UI per a l'estat de tancament de sessió
    setupUI();
  }
});

// Inici de sessió
const loginForm = document.querySelector('#login-form');
loginForm.addEventListener('submit', (e) => {
  // Evita que la pàgina s'actualitzi
  e.preventDefault();
  // Obté la informació de l'usuari (correu i contrasenya) del formulari
  const email = loginForm['email-entrada'].value;
  const contrasenya = loginForm['contrasenya-entrada'].value;
  // Inicia sessió amb el correu i la contrasenya proporcionats
  auth.signInWithEmailAndPassword(email, contrasenya).then((cred) => {
    // Si les credencials són correctes, restableix el formulari
    loginForm.reset();
  })
  // Detecta errors d'inici de sessió i mostra el missatge
  .catch((error) =>{
    const errorMessage = error.message;
    console.log(errorMessage);
    document.getElementById("missatge-error").innerHTML = "Credencials
    d'accés incorrectes";
  });
});

// Tanca la sessió
const logout = document.querySelector('#logout-link');
logout.addEventListener('click', (e) => {
  e.preventDefault();
  auth.signOut();
});

```

2.7. Codi JavaScript generador de gràfics

```
// S'afegeix un esdeveniment de càrrega a la finestra
// que crida a la funció onload quan la pàgina es carrega completament
window.addEventListener('load', onload);
// Funció onload on es criden les funcions per crear els gràfics
function onload(event) {
  // Es criden les funcions per crear els gràfics
  graficT = creaGraficTemp();
  graficH = creaGraficHum();
}

// Funció per crear el gràfic de temperatura
function creaGraficTemp() {
  // Es crea el gràfic utilitzant la biblioteca Highcharts
  var grafic = new Highcharts.Chart({
    chart: {
      // S'especifica l'element HTML on es carrega el gràfic
      renderTo: 'grafic-temp',
      // S'especifica el tipus de gràfic
      type: 'spline'
    },
    series: [{
      // Nom de la sèrie de dades
      name: 'Temperatura DHT22'
    }],
    title: {
      text: undefined
    },
    plotOptions: {
      line: {
        // Es deshabilita l'animació de la línia
        animation: false,
        dataLabels: {
          // S'habiliten les etiquetes de dades
          enabled: true
        }
      },
    },
    series: {
      // Color de la línia
      color: '#ff0000'
    }
  },
  xAxis: {
    // Valors de l'eix X (data i hora)
    type: 'datetime',
    dateTimeLabelFormats: { second: '%H:%M:%S' }
  },
},
```

Annexos

```
yAxis: {
  title: {
    // Títol de l'eix Y
    text: 'Temperatura (°C)'
  }
},
credits: {
  // Desactiva els crèdits de Highcharts
  enabled: false
}
});
// Es retorna el gràfic creat
return grafic;
}

// Funció per crear el gràfic de humitat
function creaGraficHum(){
  // Es crea el gràfic utilitzant la biblioteca Highcharts
  var grafic = new Highcharts.Chart({
    chart:{
      // S'especifica l'element HTML on es carrega el gràfic
      renderTo:'grafic-hum',
      // S'especifica el tipus de gràfic
      type: 'spline'
    },
    series: [{
      // Nom de la sèrie de dades
      name: 'Humitat DHT22'
    }],
    title: {
      text: undefined
    },
    plotOptions: {
      line: {
        // Es deshabilita l'animació de la línia
        animation: false,
        dataLabels: {
          // S'habiliten les etiquetes de dades
          enabled: true
        }
      }
    },
    series: {
      // Color de la línia
      color: '#008cff'
    }
  },
```

Annexos

```
xAxis: {
  // Valors de l'eix X (data i hora)
  type: 'datetime',
  dateTimeLabelFormats: { second: '%H:%M:%S' }
},
yAxis: {
  title: {
    // Títol de l'eix Y
    text: 'Humitat (%)'
  }
},
credits: {
  // Desactiva els crèdits de Highcharts
  enabled: false
}
});
// Es retorna el gràfic creat
return grafic;
}
```

2.8. Codi JavaScript principal

```
// Elements per a la configuració d'inici/tancament de sessió
const loginElement = document.querySelector('#login-form');
const containerElement = document.querySelector('#container-form');
const contingutElement = document.querySelector("#contingut-sign-in");
const infoUserElement = document.querySelector('#info-usuari');
const authBarElement = document.querySelector("#barra-autenticacio");

// Elements per a les lectures del sensor DHT22
var dbPathClima
const tempElement = document.getElementById("temp");
const humElement = document.getElementById("hum");

// Elements per al control de temperatura
//---> Temperatura desitjada
var dbRefTemp_Desitj;
const tempDesitjada = document.getElementById("temp-desitjada");
//---> Estat de l'aire condicionat
const acEstat = document.getElementById("ac-estat");
const acEstatOut = document.getElementById("ac-estat-out");
const acEstatContainer = document.getElementById('ac-estat-container');

// Elements per a les lectures del sensor de flama
//---> Sensor zona menjador
const focEstat_Menj = document.getElementById("estat-foc-menj");
const focEstatOut_Menj = document.getElementById('estat-foc-out-menj');
const focEstatContainer_Menj = document.getElementById('estat-foc-
container-menj');
//---> Sensor zona cuina
const focEstat_Cuina = document.getElementById("estat-foc-cuina");
const focEstatOut_Cuina = document.getElementById('estat-foc-out-cuina')
const focEstatContainer_Cuina = document.getElementById('estat-foc-
container-cuina');

// Element per a les lectures del sensor de presència
//---> Sensor zona rebedor
const pirEstat_Rebedor = document.getElementById("pir-estat-rebedor");
const pirEstatOut_Rebedor = document.getElementById("pir-estat-out-reb");
const pirEstatContainer_Rebedor = document.getElementById('pir-estat-
container-reb');
//---> Sensor zona cuina
const pirEstat_Cuina = document.getElementById("pir-estat-cuina");
const pirEstatOut_Cuina = document.getElementById("pir-estat-out-cuina");
const pirEstatContainer_Cuina = document.getElementById('pir-estat-
container-cuina');
```

Annexos

```
//---> Sensor zona WC
const pirEstat_WC = document.getElementById("pir-estat-wc");
const pirEstatOut_WC = document.getElementById("pir-estat-out-wc");
const pirEstatContainer_WC = document.getElementById('pir-estat-
container-wc');

// Elements per a les llums
var dbPathLed;
//---> Llum rebedor
const intensitat_Rebedor = document.getElementById("intensitat-rebedor");
const color_Rebedor = document.getElementById("color-rebedor");

//---> Llum cuina
const intensitat_Cuina = document.getElementById("intensitat-cuina");
const color_Cuina = document.getElementById("color-cuina");

//---> Llum WC
const intensitat_WC = document.getElementById("intensitat-wc");
const color_WC = document.getElementById("color-wc");

//---> Llum menjador
const estat_Menj = document.getElementById("estat-menj");
const intensitat_Menj = document.getElementById("intensitat-menj");
const color_Menj = document.getElementById("color-menj");

//---> Llum jardi
const estat_Jardi = document.getElementById("estat-jardi");
const intensitat_Jardi = document.getElementById("intensitat-jardi");
const color_Jardi = document.getElementById("color-jardi");

//---> Llum piscina
const estat_Pisc = document.getElementById("estat-pisc");
const intensitat_Pisc = document.getElementById("intensitat-pisc");
const color_Pisc = document.getElementById("color-pisc");

// Elements per a les persianes
var dbPathPersiana;
//---> Persiana menjador
const mode_Menj = document.getElementById("mode-persiana-menj");
var dbRefSensorLlum_Menj;
//---> Persiana habitació
const mode_Hab = document.getElementById("mode-persiana-hab");
var dbRefSensorLlum_Hab;

// Elements per a l'alarma
var dbPathAlarma;
const estat_Alarma = document.getElementById("estat-alarma");
const pirAlarmaEstat = document.getElementById("pir-estat");
const pirAlarmaEstatOut = document.getElementById("pir-estat-out");
```

Annexos

```
const pirAlarmaEstatContainer = document.getElementById('pir-estat-
container');
//---> Sensor PIR rebedor
var dbRefPIR_Rebedor;
//---> Sensor PIR cuina
var dbRefPIR_Cuina;

// Elements per al control d'accés (portes)
var dbPathPortes;
// ---> Porta exterior
var estat_PortaExt = document.getElementById("estat-porta-ext");
// ---> Porta casa
var estat_PortaCasa = document.getElementById("estat-porta-casa");

// Elements per a la piscina
var dbPathDepuradora;
const estat_depuradora = document.getElementById("estat-depuradora");

// Gestiona l'UI (Interfície d'usuari) en l'inici/tancament de sessió
const setupUI = (user) => {
  // L'usuari inicia sessió
  if (user) {
    // Intercanvia elements UI (amaga el formulari i mostra la pàgina principal)
    loginElement.style.display = 'none';
    containerElement.style.display = 'none';
    contingutElement.style.display = 'block';
    authBarElement.style.display = 'block';
    infoUserElement.style.display = 'block';
    infoUserElement.innerHTML = user.email;

    // Obté l'UID d'usuari per obtenir dades de la database
    var uid = user.uid;
    console.log(uid);

    // Adreces a la database
    var dbPathFire = 'UsersData/' + uid.toString() + '/alarma-incendis';
    dbPathClima = 'UsersData/' + uid.toString() + '/climatitzacio';
    dbPathLed = 'UsersData/' + uid.toString() + '/llums';
    dbPathPersiana = 'UsersData/' + uid.toString() + '/persianes/persiana';
    dbPathAlarma = 'UsersData/' + uid.toString() + '/alarma';
    dbPathPortes = 'UsersData/' + uid.toString() + '/control-acces/porta';
    dbPathDepuradora = 'UsersData/' + uid.toString() + '/depuradora';

    // Referències a la database (necessari realitzar monitoritzacions)
    var dbRefTemp = firebase.database().ref(dbPathClima + '/temperatura');
    var dbRefHum = firebase.database().ref(dbPathClima + '/humitat');
    var dbRefFoc_Menj = firebase.database().ref(dbPathFire + '/menjador');
    var dbRefFoc_Cuina = firebase.database().ref(dbPathFire + '/cuina');
```

Annexos

```
var dbRefLed_Rebedor = firebase.database().ref(dbPathLed + '/llum-
rebedor/estat');
var dbRefLed_Cuina = firebase.database().ref(dbPathLed + '/llum-
cuina/estat');
var dbRefLed_WC = firebase.database().ref(dbPathLed + '/llum-
wc/estat');
dbRefTemp_Desitj = firebase.database().ref(dbPathClima + '/temp-
desitjada');
dbRefSensorLlum_Menj = firebase.database().ref(dbPathPersiana + '-
menj/intensitat-llum');
dbRefSensorLlum_Hab = firebase.database().ref(dbPathPersiana + '-
hab/intensitat-llum');
dbRefPIR_Rebedor = firebase.database().ref(dbPathAlarma + '/pir-
rebedor');
dbRefPIR_Cuina = firebase.database().ref(dbPathAlarma + '/pir-
cuina');

// Actualitza la pàgina amb la lectura de temperatura més recent
// Actualitza el gràfic de la temperatura
dbRefTemp.on('value', snap => {
  // Obté la mesura de la database
  tempElement.innerText = snap.val().toFixed(2);
  // Obté la variable temporal del gràfic
  x = (new Date()).getTime(),
  // Converteix temperatura a tipus float
  y = parseFloat(snap.val().toFixed(2));
  // Configura els valors en el gràfic
  if(graficT.series[0].data.length > 40) {
    graficT.series[0].addPoint([x, y], true, true, true);
  } else {
    graficT.series[0].addPoint([x, y], true, false, true);
  }
});
// Actualitza la pàgina amb la lectura d'humitat més recent
// Actualitza el gràfic d'humitat
dbRefHum.on('value', snap => {
  // Obté la mesura de la database
  humElement.innerText = snap.val().toFixed(2);
  // Obté la variable temporal del gràfic
  var x = (new Date()).getTime(),
  // Converteix temperatura a tipus float
  y = parseFloat(snap.val().toFixed(2));
  // Configura els valors en el gràfic
  if(graficH.series[0].data.length > 40) {
    graficH.series[0].addPoint([x, y], true, true, true);
  } else {
    graficH.series[0].addPoint([x, y], true, false, true);
  }
});
```

Annexos

```
// Actualitza la pàgina amb la lectura del sensor d'incendis situat
al menjador
dbRefFoc_Menj.on('value', snap => {
  // Obté la mesura de la database
  focEstat_Menj.innerHTML = snap.val();
  // Converteix la mesura en un valor de tipus int
  focEstat = parseInt(focEstat_Menj.textContent);

  if (focEstat === 1) {
    // Si es detecta foc intercanvia l'estat i mostra el resultat
    focEstatContainer_Menj.classList.remove('no-foc');
    focEstatContainer_Menj.classList.add('foc-detectat');
    focEstatOut_Menj.textContent = "FOC DETECTAT!";
  } else {
    // Si NO es detecta, intercanvia l'estat i mostra el resultat
    focEstatContainer_Menj.classList.remove('foc-detectat');
    focEstatContainer_Menj.classList.add('no-foc');
    focEstatOut_Menj.textContent = "NO HI HA FOC";
  }
});

// Actualitza la pàgina amb la lectura del sensor d'incendis situat a
la cuina
dbRefFoc_Cuina.on('value', snap => {
  // Obté la mesura de la database
  focEstat_Cuina.innerHTML = snap.val();
  // Converteix la mesura en un valor de tipus int
  focEstat = parseInt(focEstat_Cuina.textContent);

  if (focEstat === 1) {
    // Si es detecta foc intercanvia l'estat i escriu el resultat
    focEstatContainer_Cuina.classList.remove('no-foc');
    focEstatContainer_Cuina.classList.add('foc-detectat');
    focEstatOut_Cuina.textContent = "FOC DETECTAT!";
  } else {
    // Si NO es detecta, intercanvia l'estat i escriu el resultat
    focEstatContainer_Cuina.classList.remove('foc-detectat');
    focEstatContainer_Cuina.classList.add('no-foc');
    focEstatOut_Cuina.textContent = "NO HI HA FOC";
  }
});

// Actualitza la pàgina amb la lectura del sensor de presència situat
al rebedor
dbRefLed_Rebedor.on('value', snap => {
  // Obté la mesura de la database
  pirEstat_Rebedor.innerHTML = snap.val();
  // Converteix la mesura en un valor de tipus int
  pirEstat = parseInt(pirEstat_Rebedor.textContent);
```

Annexos

```
if (pirEstat === 1) {
  // Si es detecta presència intercanvia l'estat i escriu el
  // resultat en la web
  pirEstatContainer_Rebedor.classList.remove('off');
  pirEstatContainer_Rebedor.classList.add('on');
  pirEstatOut_Rebedor.textContent = "ON";
} else {
  // Si NO es detecta, intercanvia l'estat i escriu el resultat en
  // la web
  pirEstatContainer_Rebedor.classList.remove('on');
  pirEstatContainer_Rebedor.classList.add('off');
  pirEstatOut_Rebedor.textContent = "OFF";
}
});

// Actualitza la pàgina amb la lectura del sensor de presència situat
// a la cuina
dbRefLed_Cuina.on('value', snap => {
  // Obté la mesura de la database
  pirEstat_Cuina.innerText = snap.val();
  // Converteix la mesura en un valor de tipus int
  pirEstat = parseInt(pirEstat_Cuina.textContent);

  if (pirEstat === 1) {
    // Si es detecta presència intercanvia l'estat i escriu el
    // resultat en la web
    pirEstatContainer_Cuina.classList.remove('off');
    pirEstatContainer_Cuina.classList.add('on');
    pirEstatOut_Cuina.textContent = "ON";
  } else {
    // Si NO es detecta, intercanvia l'estat i escriu el resultat en
    // la web
    pirEstatContainer_Cuina.classList.remove('on');
    pirEstatContainer_Cuina.classList.add('off');
    pirEstatOut_Cuina.textContent = "OFF";
  }
});

// Actualitza la pàgina amb la lectura del sensor de presència situat
// al bany
dbRefLed_WC.on('value', snap => {
  // Obté la mesura de la database
  pirEstat_WC.innerText = snap.val();
  // Converteix la mesura en un valor de tipus int
  pirEstat = parseInt(pirEstat_WC.textContent);
```

Annexos

```
    if (pirEstat === 1) {
      // Si es detecta presència intercanvia l'estat i escriu el
      // resultat en la web
      pirEstatContainer_WC.classList.remove('off');
      pirEstatContainer_WC.classList.add('on');
      pirEstatOut_WC.textContent = "ON";
    } else {
      // Si NO es detecta, intercanvia l'estat i escriu el resultat en
      // la web
      pirEstatContainer_WC.classList.remove('on');
      pirEstatContainer_WC.classList.add('off');
      pirEstatOut_WC.textContent = "OFF";
    }
  });

  // L'usuari tanca la sessió
} else{
  // Amaga elements de la pàgina principal i ensenya el formulari
  // d'inici de sessió
  loginElement.style.display = 'block';
  containerElement.style.display = 'block';
  authBarElement.style.display = 'none';
  infoUserElement.style.display = 'none';
  contingutElement.style.display = 'none';
}
}

// Funció que escriu a la database l'acció seleccionada per l'usuari per
// a una determinada persiana quan està en mode manual
function manualPersiana(accio, persiana_id) {
  // Mostra en la consola el canvi realitzat
  console.log("Toggle Persiana " + persiana_id + " - Accio: " + accio);
  // Escriu en l'adreça corresponent l'acció seleccionada
  firebase.database().ref(dbPathPersiana + persiana_id + '/accio-
  manual').set(accio);
}

// Funció que escriu a la database el mode de funcionament de la persiana
function toggleModePersiana(persiana_id){
  // Persiana menjador
  if (persiana_id === '-menjador'){
    // Interruptor està seleccionat
    if (mode_Menj.checked)
    {
      // Mostra en la consola el mode de funcionament
      console.log("Mode de funcionament persiana menjador: AUTO");
      // Escriu en l'adreça corresponent el mode de funcionament AUTO
      firebase.database().ref(dbPathPersiana + persiana_id + '/mode-
      func').set("AUTO");
    }
  }
}
```

Annexos

```
}
// Si no està seleccionat
else{
    // Mostra en la consola el mode de funcionament
    console.log("Mode de funcionament persiana menjador: MANUAL");
    // Escriu en l'adreça corresponent el mode de funcionament MANUAL
    firebase.database().ref(dbPathPersiana + persiana_id + '/mode-
func').set("MANUAL");
}
}
// Persiana habitació
if (persiana_id === '-hab'){
    // Interruptor està seleccionat
    if (mode_Hab.checked)
    {
        // Mostra en la consola el mode de funcionament
        console.log("Mode de funcionament persiana hab: AUTO");
        // Escriu en l'adreça corresponent el mode de funcionament AUTO
        firebase.database().ref(dbPathPersiana + persiana_id + '/mode-
func').set("AUTO");
    }
    // Si no està seleccionat
    else{
        // Mostra en la consola el mode de funcionament
        console.log("Mode de funcionament persiana hab: MANUAL");
        // Escriu en l'adreça corresponent el mode de funcionament MANUAL
        firebase.database().ref(dbPathPersiana + persiana_id + '/mode-
func').set("MANUAL");
    }
}
}

// Funció que escriu a la database l'estat d'una determinada tira LED
function toggleLed(led_id) {
    if (led_id === 'menjador'){
        // Si el botó està seleccionat
        if (estat_Menj.checked)
        {
            // Mostra en la consola l'estat ON del LED
            console.log("Toggle LED menjador -> LED ON");
            // Escriu en l'adreça corresponent l'estat ON del LED
            firebase.database().ref(dbPathLed + '/llum-menjador/estat')
            .set("ON");
        }
        // Si no està seleccionat
        else{
            // Mostra en la consola l'estat OFF del LED
            console.log("Toggle LED menjador -> LED OFF");
            // Escriu en l'adreça corresponent l'estat OFF del LED
```

Annexos

```
        firebase.database().ref(dbPathLed + '/llum-menjador/estat')
        .set("OFF");
    }
}
// Llum jardí
else if (led_id === 'jardi') {
    if (estat_Jardi.checked)
    {
        console.log("Toggle LED jardi -> LED ON");
        firebase.database().ref(dbPathLed + '/llum-jardi/estat').set("ON");
    }
    else{
        console.log("Toggle LED jardi -> LED OFF");
        firebase.database().ref(dbPathLed + '/llum-jardi/estat').set("OFF");
    }
}
// Llum piscina
else if (led_id === 'piscina') {
    if (estat_Pisc.checked)
    {
        console.log("Toggle LED piscina -> LED ON");
        firebase.database().ref(dbPathLed + '/llum-piscina/estat').set("ON");
    }
    else{
        console.log("Toggle LED piscina -> LED OFF");
        firebase.database().ref(dbPathLed + '/llum-piscina/estat').set("OFF");
    }
}
}

// Funció que escriu a la database la intensitat seleccionada per
l'usuari d'un determinat LED
function actualitzaLedIntensitat(valor, led_id) {
    // Llum menjador
    if (led_id === 'menjador'){
        // Actualitza el valor en la pàgina web
        intensitat_Menj.textContent = valor + ' %';
        // Mostra en consola la intensitat seleccionada
        console.log("Set LED menjador intensitat: " + valor);
        // Escriu en l'adreça corresponen el valor (tipus int) seleccionat
        firebase.database().ref(dbPathLed + '/llum-menjador/intensitat')
        .set(parseInt(valor));
    }
    // Llum rebedor
    else if (led_id === 'rebedor'){
        intensitat_Rebedor.textContent = valor + ' %';
        console.log("Set LED rebedor intensitat: " + valor);
        firebase.database().ref(dbPathLed + '/llum-rebedor/intensitat')
        .set(parseInt(valor));
    }
}
```

Annexos

```
}
// Llum cuina
else if (led_id === 'cuina'){
    intensitat_Cuina.textContent = valor + ' %';
    console.log("Set LED cuina intensitat: " + valor);
    firebase.database().ref(dbPathLed + '/llum-cuina/intensitat')
        .set(parseInt(valor));
}
// Llum bany
else if (led_id === 'wc'){
    intensitat_WC.textContent = valor + ' %';
    console.log("Set LED wc intensitat: " + valor);
    firebase.database().ref(dbPathLed + '/llum-wc/intensitat')
        .set(parseInt(valor));
}
// Llum jardí
else if (led_id === 'jardi'){
    intensitat_Jardi.textContent = valor + ' %';
    console.log("Set LED jardí intensitat: " + valor);
    firebase.database().ref(dbPathLed + '/llum-jardi/intensitat')
        .set(parseInt(valor));
}
// Llum piscina
else if (led_id === 'piscina'){
    intensitat_Pisc.textContent = valor + ' %';
    console.log("Set LED piscina intensitat: " + valor);
    firebase.database().ref(dbPathLed + '/llum-piscina/intensitat')
        .set(parseInt(valor));
}
}

// Funció que escriu a la database el color seleccionat per l'usuari d'un
determinat LED
function actualitzaColor(color, led_id) {
    // Converteix el valor del color d'hexadecimal a RGB
    // Emmagatzema com una string amb els elements separats per una coma
    var rgb = hexToRgb(color).split(", ");
    // Obté les components RGB per separat i les converteix a tipus int
    var r = parseInt(rgb[0]);
    var g = parseInt(rgb[1]);
    var b = parseInt(rgb[2]);
    // Llum menjador
    if (led_id === 'menjador'){
        // Envia a l'adreça corresponent els valors RGB
        firebase.database().ref(dbPathLed + '/llum-menjador/color').set({
            r: r,
            g: g,
            b: b
        });
    }
}
```

Annexos

```
}
// Llum rebedor
else if (led_id === 'rebedor'){
  firebase.database().ref(dbPathLed + '/llum-rebedor/color').set({
    r: r,
    g: g,
    b: b
  });
}
// Llum cuina
else if (led_id === 'cuina'){
  firebase.database().ref(dbPathLed + '/llum-cuina/color').set({
    r: r,
    g: g,
    b: b
  });
}
// Llum bany
else if (led_id === 'wc'){
  firebase.database().ref(dbPathLed + '/llum-wc/color').set({
    r: r,
    g: g,
    b: b
  });
}
// Llum jardí
else if (led_id === 'jardi'){
  firebase.database().ref(dbPathLed + '/llum-jardi/color').set({
    r: r,
    g: g,
    b: b
  });
}
// Llum piscina
else if (led_id === 'piscina'){
  firebase.database().ref(dbPathLed + '/llum-piscina/color').set({
    r: r,
    g: g,
    b: b
  });
}
}

// Funció per convertir el valor hexadecimal de color a RGB
function hexToRgb(hex) {
  // Converteix el valor hexadecimal a tipus int.
  // Omet el primer caràcter per ignorar el símbol '0x'
  var bigint = parseInt(hex.substring(1), 16);
  // Separa les components RGB
```

Annexos

```
// Utilitza operacions de desplaçament de bits i màscares
var r = (bigint >> 16) & 255;
var g = (bigint >> 8) & 255;
var b = bigint & 255;
// Retorna una string amb els valors RGB separats per una coma
return r + ", " + g + ", " + b;
}

// Funció que escriu a la database l'estat de l'alarma
function toggleAlarma(){
  // Si el botó està seleccionat
  if (estat_Alarma.checked)
  {
    // Mostra en la consola l'estat de l'alarma
    console.log("Toggle alarma -> alarma ACTIVADA");
    // Escriu en l'adreça corresponent l'estat de l'alarma
    firebase.database().ref(dbPathAlarma + '/estat').set("ACTIVADA");
  }
  // Si no està seleccionat
  else{
    // Mostra en la consola l'estat de l'alarma
    console.log("Toggle alarma -> alarma DESACTIVADA");
    // Escriu en l'adreça corresponent l'estat de l'alarma
    firebase.database().ref(dbPathAlarma + '/estat').set("DESACTIVADA");
  }
}

// Funció que monitoritza els sensors PIR quan l'alarma està activada
function checkAlarma(){
  // Quan l'alarma està activada, mostreja el valor dels sensors PIR
  if (estat_Alarma.checked){
    // Obté el valor del sensor del rebedor i el converteix a tipus int
    dbRefPIR_Rebedor.once('value', snap => {
      pirAlarmaEstat.innerText = snap.val()
      const pirRebedor_Estat = parseInt(pirAlarmaEstat.textContent);
      // Obté el valor del sensor de la cuina i el converteix a tipus int
      dbRefPIR_Cuina.once('value', snap => {
        pirAlarmaEstat.innerText = snap.val()
        const pirCuina_Estat = parseInt(pirAlarmaEstat.textContent);

        // Si es detecta presència intercanvia l'estat
        // Mostra el resultat en la web
        if ((pirRebedor_Estat === 1) || (pirCuina_Estat === 1)) {
          pirAlarmaEstatContainer.classList.add('detect');
          pirAlarmaEstatOut.textContent = "INTRÚS";
        } else {
          // Si NO es detecta, intercanvia l'estat
          // Mostra el resultat en la web
          pirAlarmaEstatContainer.classList.remove('detect');
        }
      }
    }
  }
}
```

Annexos

```
        pirAlarmaEstatOut.textContent = "";
    }
    })
  })
}
// No mostra el resultat si l'alarma no està activada
else {
  pirAlarmaEstatContainer.classList.remove('detect');
  pirAlarmaEstatOut.textContent = "";
}
}
// Executa la funció checkAlarma() cada 3 segons
setInterval(checkAlarma, 3000);

// Funció que escriu a la database l'estat d'una determinada porta
function togglePorta(porta_id){
  // Porta exterior d'accés a la parcel·la
  if (porta_id === 'porta-exterior'){
    // Si el botó està seleccionat
    if (estat_PortaExt.checked)
    {
      // Mostra en la consola l'estat de la porta
      console.log("Toggle porta exterior -> porta exterior OBERTA");
      // Escriu en l'adreça corresponent l'estat de la porta
      firebase.database().ref(dbPathPortes + '-exterior').set("OBERTA");
    }
    // Si no està seleccionat
    else{
      // Mostra en la consola l'estat de la porta
      console.log("Toggle porta exterior -> porta exterior TANCADA");
      // Escriu en l'adreça corresponent l'estat de la porta
      firebase.database().ref(dbPathPortes + '-exterior').set("TANCADA");
    }
  }
  // Porta d'accés a la casa
  else if (porta_id === 'porta-casa') {
    if (estat_PortaCasa.checked)
    {
      console.log("Toggle porta casa -> porta casa OBERTA");
      firebase.database().ref(dbPathPortes + '-casa').set("OBERTA");
    }
    else{
      console.log("Toggle porta casa -> porta casa TANCADA");
      firebase.database().ref(dbPathPortes + '-casa').set("TANCADA");
    }
  }
}
}
```

Annexos

```
// Funció que escriu a la database l'estat de la depuradora
function toggleDepuradora(){
  // Si el botó està seleccionat
  if (estat_depuradora.checked)
  {
    // Mostra en la consola l'estat de la depuradora
    console.log("Toggle depuradora -> depuradora ON");
    // Escriu en l'adreça corresponent l'estat de la depuradora
    firebase.database().ref(dbPathDepuradora).set("ON");
  }
  // Si no està seleccionat
  else{
    // Mostra en la consola l'estat de la depuradora
    console.log("Toggle depuradora -> alarma OFF");
    // Escriu en l'adreça corresponent l'estat de la depuradora
    firebase.database().ref(dbPathDepuradora).set("OFF");
  }
}

// Funció que escriu a la database la temperatura seleccionada per
l'usuari
function actualitzaTemp(valor) {
  // Mostra en consola la temperatura seleccionada
  console.log("Set temperatura desitjada: " + valor);
  // Escriu en l'adreça corresponent el valor seleccionat
  firebase.database().ref(dbPathClima + '/temp-desitjada').set
  (parseFloat(valor));
}

// Funció que canvia l'estat del AC en funció de la temperatura
seleccionada i mesurada
function setAC(){
  dbRefTemp_Desitj.once('value', snap => {
    // Obté el valor de la temperatura desitjada i el converteix a float
    tempDesitjada.innerText = snap.val();
    const tempDesitj = parseFloat(tempDesitjada.textContent);
    // Obté el valor de la mesura de temperatura realitzada pel DHT22
    const tempMedida = parseFloat(tempElement.innerText);

    // Si la temperatura ambiental es major a la desitjada, engega AC
    if(tempDesitj < tempMedida){
      acEstatContainer.classList.remove('off');
      acEstatContainer.classList.add('on');
      acEstatOut.textContent = "ON";
      // Mostra en consola que l'AC passa a ON
      console.log("Set AC: ON");
      // Escriu en l'adreça corresponen l'estat de l'AC
      firebase.database().ref(dbPathClima + '/estat-AC').set("ON");
    }
  }
}
```

Annexos

```
// Si la temperatura ambiental es menor a la desitjada, apaga AC
else {
  acEstatContainer.classList.remove('on');
  acEstatContainer.classList.add('off');
  acEstatOut.textContent = "OFF";
  // Mostra en consola que l'AC passa a OFF
  console.log("Set AC: OFF");
  // Escriu en l'adreça corresponen l'estat de l'AC
  firebase.database().ref(dbPathClima + '/estat-AC').set("OFF");
}
});
}
// Executa la funció setAC() cada 3 segons
setInterval(setAC, 3000);

// Gestiona els elements que es mostren en la pàgina principal
// Assigna un esdeveniment de click a l'element amb id 'link-home'
document.querySelector('#link-home').addEventListener('click', function()
{
  // Mostra el corresponent títol en la part superior
  document.querySelector('main h3').innerText = "Menú principal";
  mostraHome();
});
// Funció que mostra en pantalla els quadres especificats
function mostraHome() {
  // Mostra tots els quadres
  document.querySelectorAll('.quadre').forEach(quadre => {
    quadre.style.display = 'block';
  });
}

// Gestiona els elements que es mostren en la secció llums interiors
// Assigna un esdeveniment de click a l'element amb id 'link-llums-
interior'
document.querySelector('#link-llums-interior').addEventListener('click',
function() {
  // Mostra el corresponent títol en la part superior
  document.querySelector('main h3').innerText = "Llums interiors";
  mostraLlumInterior();
});
// Funció que mostra en pantalla els quadres especificats
function mostraLlumInterior() {
  // Amaga tots els quadres
  document.querySelectorAll('.quadre').forEach(quadre => {
    quadre.style.display = 'none';
  });
  // Mostra els especificats per la secció
  document.querySelector('#quadre-llum-menjador').style.display = 'block';
  document.querySelector('#quadre-llum-rebedor').style.display = 'block';
}
```

Annexos

```
document.querySelector('#quadre-llum-cuina').style.display = 'block';
document.querySelector('#quadre-llum-wc').style.display = 'block';
}

// Gestiona els elements que es mostren en la secció llums exteriors
// Assigna un esdeveniment de click a l'element amb id 'link-llums
exterior'
document.querySelector('#link-llums-exterior').addEventListener('click',
function() {
    // Mostra el corresponent títol en la part superior
    document.querySelector('main h3').innerText = "Llums exteriors";
    mostraLlumExterior();
});
// Funció que mostra en pantalla els quadres especificats
function mostraLlumExterior() {
    // Amaga tots els quadres
    document.querySelectorAll('.quadre').forEach(quadre => {
        quadre.style.display = 'none';
    });
    // Mostra els especificats per la secció
    document.querySelector('#quadre-llum-jardi').style.display = 'block';
    document.querySelector('#quadre-llum-piscina').style.display = 'block';
}

// Gestiona els elements que es mostren en la secció persianes
// Assigna un esdeveniment de click a l'element amb id 'link-persianes'
document.querySelector('#link-persianes').addEventListener('click',
function() {
    // Mostra el corresponent títol en la part superior
    document.querySelector('main h3').innerText = "Persianes";
    mostraPersianes();
});
// Funció que mostra en pantalla els quadres especificats
function mostraPersianes() {
    // Amaga tots els quadres
    document.querySelectorAll('.quadre').forEach(quadre => {
        quadre.style.display = 'none';
    });
    // Mostra els especificats per la secció
    document.querySelector('#quadre-persiana-menjador').style.display =
'block';
    document.querySelector('#quadre-persiana-hab').style.display = 'block';
}

// Gestiona els elements que es mostren en la secció climatització
// Assigna un esdeveniment de click a l'element amb id 'link-
climatitzacio'
document.querySelector('#link-climatitzacio').addEventListener('click',
function() {
```

Annexos

```
// Mostra el corresponent títol en la part superior
document.querySelector('main h3').innerText = "Climatització";
mostraClima();
});
// Funció que mostra en pantalla els quadres especificats
function mostraClima() {
  // Amaga tots els quadres
  document.querySelectorAll('.quadre').forEach(quadre => {
    quadre.style.display = 'none';
  });
  // Mostra els especificats per la secció
  document.querySelector('#quadre-climatitzacio').style.display = 'block';
  document.querySelector('#quadre-control-temp').style.display = 'block';
  document.querySelector('#grafic-temperatura').style.display = 'block';
  document.querySelector('#grafic-humitat').style.display = 'block';
}

// Gestiona els elements que es mostren en la secció piscina
// Assigna un esdeveniment de click a l'element amb id 'link-piscina'
document.querySelector('#link-piscina').addEventListener('click',
function() {
  // Mostra el corresponent títol en la part superior
  document.querySelector('main h3').innerText = "Piscina";
  mostraPiscina();
});
// Funció que mostra en pantalla els quadres especificats
function mostraPiscina() {
  // Amaga tots els quadres
  document.querySelectorAll('.quadre').forEach(quadre => {
    quadre.style.display = 'none';
  });
  // Mostra els especificats per la secció
  document.querySelector('#quadre-llum-piscina').style.display = 'block';
  document.querySelector('#quadre-depuradora').style.display = 'block';
}

// Gestiona els elements que es mostren en la secció seguretat
// Assigna un esdeveniment de click a l'element amb id 'link-seguretat'
document.querySelector('#link-seguretat').addEventListener('click',
function() {
  // Mostra el corresponent títol en la part superior
  document.querySelector('main h3').innerText = "Seguretat";
  mostraSeguretat();
});
// Funció que mostra en pantalla els quadres especificats
function mostraSeguretat() {
  // Amaga tots els quadres
  document.querySelectorAll('.quadre').forEach(quadre => {
    quadre.style.display = 'none';
```

Annexos

```
});  
// Mostra els especificats per la secció  
document.querySelector('#quadre-incendis').style.display = 'block';  
document.querySelector('#quadre-alarma').style.display = 'block';  
document.querySelector('#quadre-portes').style.display = 'block';  
}
```

2.9. Codi Arduino per al microcontrolador de sensors

```
/****** LLIBRERIES & FITXERS *****/
// Llibreries de connexió al Wi-Fi i a la Firebase Database
#include <WiFi.h>
#include <WiFiMulti.h>
#include <Firebase_ESP_Client.h>
#include <Wire.h>
#include "data.h"
// Llibreries del sensor de temperatura i humitat DHT22
#include <Adafruit_Sensor.h>
#include "DHT.h"

/****** VARIABLES & OBJECTES *****/
// Objectes Firebase
FirebaseData fbdo; // Database
FirebaseAuth auth; // Autenticació
FirebaseConfig config; // Configuració

// Variable per guardar USER UID
String uid;

// Variables per guardar les adreces a la database
String databasePath;
String climatitzacioPath;
String focPath;
String alarmaIntrusosPath;
String PIR_LedPath;

// Objecte Wi-Fi
WiFiMulti wifiMulti;
const uint32_t TempsEsperaWifi = 5000;

// Variables temporals (temps de mostreig)
unsigned long millisCicleAnterior = 0;
unsigned long tempsMostreig = 2000;

/***** Sensor DHT22 ****/
#define pinDHT 4
DHT sensorTH (pinDHT, DHT22);
float temperatura;
float humitat;

/***** Control de temperatura ****/
#define pinAC 14
float temp_desitjada;
```

Annexos

```
/** Sensor de flama */
// Sensor menjador
#define pinFoc_Menj 18
int foc_estat_menj;
// Sensor cuina
#define pinFoc_Cuina 22
int foc_estat_cuina;
// Buzzer
#define pinBuzzer 19

/** Sensors PIR */
#define pinSensor_Rebedor 32
int pir_estat_rebedor;
#define pinSensor_Cuina 33
int pir_estat_cuina;
#define pinSensor_WC 34
int pir_estat_WC;

/** FUNCIONS */
// Inicialitza Wi-Fi
void initWiFi() {
    // Xarxes Wi-Fi definides
    wifiMulti.addAP(ssid_1, contrasenya_1);
    wifiMulti.addAP(ssid_2, contrasenya_2);
    wifiMulti.addAP(ssid_3, contrasenya_3);
    wifiMulti.addAP(ssid_4, contrasenya_4);

    WiFi.mode(WIFI_STA);
    Serial.print("Connectant a Wi-Fi ..");
    while (wifiMulti.run(TempsEsperaWifi) != WL_CONNECTED) {
        Serial.print(".");
    }
    Serial.println(WiFi.localIP());
    Serial.println();
}

// Envia el valor de la lectura del sensor DHT22 (temperatura i humitat)
void enviaTH(String path, float valor){
    // Configura el valor
    if (Firebase.RTDB.setFloat(&fbdo, path.c_str(), valor)){
        Serial.print("Escribint valor: ");
        Serial.print(valor);
        Serial.print(" a la següent adreça: ");
        Serial.println(path);
    }
    else {
        // Indica el motiu de l'error en la escriptura a la database
        Serial.println("ERROR");
    }
}
```

Annexos

```
        Serial.println("CAUSA: " + fbdo.errorReason());
    }
}

// Llegeix de la database el valor de temperatura desitjada per l'usuari
void controlTemp(String path) {
    String nextEstat = "";
    // Obté el valor de la temperatura desitjada
    if (Firebase.RTDB.getFloat(&fbdo, (path + "/temp-desitjada").c_str())){
        temp_desitjada = fbdo.floatData();
        Serial.print("Llegint valor: ");
        Serial.print(temp_desitjada);
        Serial.print(" de la següent adreça: ");
        Serial.println(path + "/temp-desitjada");

        // Compara la temperatura mesurada i desitjada per controlar l'AC
        if (temp_desitjada < temperatura){
            // Activa AC per baixar la temperatura a la desitjada
            digitalWrite(pinAC, HIGH);
            // Envia a la database l'estat ON de l'AC
            nextEstat = "ON";
            Firebase.RTDB.setString(&fbdo, (path + "/estat-AC").c_str(),
                nextEstat);
            Serial.println("AC estat set to ON");
        }
        else {
            // Apaga AC per mantenir la temperatura desitjada
            digitalWrite(pinAC, LOW);
            // Envia a la database l'estat OFF de l'AC
            nextEstat = "OFF";
            Firebase.RTDB.setString(&fbdo, (path + "/estat-AC").c_str(),
                nextEstat);
            Serial.println("AC estat set to OFF");
        }
    }
    else {
        // Indica el motiu de l'error en la lectura de la database
        Serial.println("ERROR");
        Serial.println("CAUSA: " + fbdo.errorReason());
    }
}

// Envia el valor de la lectura del sensor de flama
void enviaFoc (String path, int valor){
    // Configura el valor
    if (Firebase.RTDB.setInt(&fbdo, path.c_str(), valor)){
        Serial.print("Escribint valor: ");
        Serial.print (valor);
        Serial.print(" a la següent adreça: ");
    }
}
```

Annexos

```
        Serial.println(path);
    }
    else {
        // Indica el motiu de l'error en la escriptura a la database
        Serial.println("ERROR");
        Serial.println("CAUSA: " + fbdo.errorReason());
    }
}

// Envia el valor de la lectura del sensor PIR (presència)
void enviaPIR (String path, int valor){
    // Configura el valor
    if (Firebase.RTDB.setInt(&fbdo, path.c_str(), valor)){
        Serial.print("Escribint valor: ");
        Serial.print (valor);
        Serial.print(" a la següent adreça: ");
        Serial.println(path);
    }
    else {
        // Indica el motiu de l'error en la escriptura a la database
        Serial.println("ERROR");
        Serial.println("CAUSA: " + fbdo.errorReason());
    }
}

// Gestiona l'alarma sonora en funció de la lectura del sensor de flama i
de presència
void activarBuzzer(String path){
    // Obté l'estat de l'alarma (ACTIVADA/DESACTIVADA)
    Firebase.RTDB.getString(&fbdo, (path + "/estat").c_str());
    String estat = fbdo.stringData();
    Serial.print("Llegint valor: " + estat);
    Serial.println(" de la següent adreça: " + path);
    // Activa alarma sonora si qualsevol sensor detecta presència i
l'alarma està activa
    if((estat == "ACTIVADA") && ((pir_estat_rebedor == 1) ||
(pir_estat_cuina == 1))){
        analogWrite(pinBuzzer, 100);
    }
    //Activa l'alarma si algun sensor detecta foc
    else if ((foc_estat_menj == 1) || (foc_estat_cuina == 1)){
        analogWrite(pinBuzzer, 100);
    }
    // Apaga l'alarma si cap sensor detecta foc ni presència
    else {
        analogWrite(pinBuzzer, 0);
    }
}
}
```

Annexos

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

  // Inicialitza sensor DHT22
  sensorTH.begin();

  // Inicialitza sensor de flama
  pinMode(pinFoc_Menj, INPUT);
  pinMode(pinFoc_Cuina, INPUT);

  // Inicialitza motor AC
  pinMode(pinAC, OUTPUT);

  // Inicialitza sensors PIR
  pinMode(pinSensor_Rebedor, INPUT);
  pinMode(pinSensor_Cuina, INPUT);
  pinMode(pinSensor_WC, INPUT);

  // Inicialitza connexió Wi-Fi
  initWiFi();
  // Assigna la clau de l'API
  config.api_key = API_KEY;
  // Assigna les credencials d'inici de sessió de l'usuari
  auth.user.email = USER_EMAIL;
  auth.user.password = USER_PASSWORD;
  // Assigna la URL de la database
  config.database_url = DATABASE_URL;
  // Configura la reconexió automàtica del Wi-Fi en cas de desconexió
  Firebase.reconnectWiFi(true);
  fbdo.setResponseSize(4096);
  // Assigna la funció de callback per a la tasca de generació de token
  // de llarga durada
  config.token_status_callback = tokenStatusCallback;
  // Assigna el màxim nombre d'intents en la generació del token
  config.max_token_generation_retry = 5;
  // Inicialitza la llibreria amb l'autenticació i la configuració de
  // Firebase
  Firebase.begin(&config, &auth);
  // Espera mentre obté user UID
  Serial.println("Obtenint UID usuari");
  while ((auth.token.uid) == "") {
    Serial.print('.');
    delay(1000);
  }
  // Imprimeix user UID
  uid = auth.token.uid.c_str();
  Serial.print("UID usuari: ");
  Serial.println(uid);
}
```

Annexos

```
// Adreça a la database
databasePath = "/UsersData/" + uid;

// Adreces als diferents elements de la database
climatitzacioPath = databasePath + "/climatitzacio";
focPath = databasePath + "/alarma-incendis";
alarmaIntrusosPath = databasePath + "/alarma";
PIR_LedPath = databasePath + "/llums";
}

void loop(){
  // Espera el temps de mostreig seleccionat
  if (Firebase.ready() && (millis() - millisCicleAnterior > tempsMostreig
  || millisCicleAnterior == 0)){
    millisCicleAnterior = millis();

    // Realitza les mesures dels sensors
    temperatura = sensorTH.readTemperature();
    humitat = sensorTH.readHumidity();
    foc_estat_menj = int(digitalRead(pinFoc_Menj));
    foc_estat_cuina = int(digitalRead(pinFoc_Cuina));
    pir_estat_rebedor = int(digitalRead(pinSensor_Rebedor));
    pir_estat_cuina = int(digitalRead(pinSensor_Cuina));
    pir_estat_WC = int(digitalRead(pinSensor_WC));

    // Envia les mesures a la database
    enviaTH(climatitzacioPath + "/temperatura", temperatura);
    enviaTH(climatitzacioPath + "/humitat", humitat);
    enviaFoc(focPath + "/menjador", foc_estat_menj);
    enviaFoc(focPath + "/cuina", foc_estat_cuina);
    enviaPIR(PIR_LedPath + "/llum-rebedor/estat", pir_estat_rebedor);
    enviaPIR(PIR_LedPath + "/llum-cuina/estat", pir_estat_cuina);
    enviaPIR(PIR_LedPath + "/llum-wc/estat", pir_estat_WC);
    enviaPIR(alarmaIntrusosPath + "/pir-rebedor", pir_estat_rebedor);
    enviaPIR(alarmaIntrusosPath + "/pir-cuina", pir_estat_cuina);

    // Llegeix dades de la database i gestiona el dispositiu corresponent
    controlTemp(climatitzacioPath);
    activarBuzzer(alarmaIntrusosPath);
  }
}
```

2.10. Codi Arduino per al microcontrolador de motors

```
/****** LLIBRERIES & FITXERS *****/
// Llibreries de connexió al Wi-Fi i a la Firebase Database
#include <Arduino.h>
#include <WiFi.h>
#include <WiFiMulti.h>
#include <Firebase_ESP_Client.h>
#include <Wire.h>
#include "data.h"
// Llibreria servomotors
#include <ESP32Servo.h>

/****** VARIABLES & OBJECETS *****/
// Objectes Firebase
FirebaseData fbdo; // Database
FirebaseAuth auth; // Autenticació
FirebaseConfig config; // Configuració

// Variable per guardar USER UID
String uid;

// Variable per guardar les adreces a la database
String databasePath;
String persianaPath;
String depuradoraPath;

// Objecte Wi-Fi
WiFiMulti wifiMulti;
const uint32_t TempsEsperaWifi = 5000;

// Variables temporals (temps de mostreig)
unsigned long millisCicleAnterior = 0;
unsigned long tempsMostreig = 2000;

/***** Sensors intensitat llum ****/
#define pinSensorLlum_Menj 34
#define pinSensorLlum_Hab 35
int intensitatLlum_Menj;
int intensitatLlum_Hab;

/***** Mode de funcionament persianes ****/
String modeFunc_Menj;
String modeFunc_Hab;
```

Annexos

```

/**** Servomotors persianes ****/
#define pinPersiana1 14
#define pinPersiana2 12
Servo persiana1;
Servo persiana2;

/** Motor depuradora **/
#define pinDepuradora 13
int estat_depuradora;

/***** FUNCIONS *****/
// Inicialitza Wi-Fi
void initWiFi() {
    // Xarxes Wi-Fi definides
    wifiMulti.addAP(ssid_1, contrasenya_1);
    wifiMulti.addAP(ssid_2, contrasenya_2);
    wifiMulti.addAP(ssid_3, contrasenya_3);
    wifiMulti.addAP(ssid_4, contrasenya_4);

    WiFi.mode(WIFI_STA);
    Serial.print("Connectant a Wi-Fi ..");
    while (wifiMulti.run(TempsEsperaWifi) != WL_CONNECTED) {
        Serial.print(".");
    }
    Serial.println(WiFi.localIP());
    Serial.println();
}

// Escriu en la database la mesura de la intensitat de la llum
void enviaLlum(String path, int intensitat){
    // Configura el valor
    if (Firebase.RTDB.setInt(&fbdo, path.c_str(), intensitat)){
        Serial.print("Escribint valor: ");
        Serial.print(intensitat);
        Serial.print(" a la següent adreça: " + path);
        Serial.println();
    }
    else {
        // Indica el motiu de l'error en la escriptura a la database
        Serial.println("ERROR");
        Serial.println("CAUSA: " + fbdo.errorReason());
    }
}

// Llegeix de la database el mode de funcionament seleccionat per les
persianes
void modeFuncPersiana(String path, String id){
    // Obté el mode de funcionament
    if (Firebase.RTDB.getString(&fbdo, (path).c_str())){

```

Annexos

```
String modeFunc = fbdo.stringData();
Serial.println();
Serial.print("Llegint valor: " + modeFunc);
Serial.println(" de la següent adreça: " + path);
// Assigna el mode de funcionament a la persiana especificada
if (id == "menj"){
    modeFunc_Menj = modeFunc;
}
else if (id == "hab"){
    modeFunc_Hab = modeFunc;
}
}
else {
    // Indica el motiu de l'error en la lectura de la database
    Serial.println("ERROR");
    Serial.println("CAUSA: " + fbdo.errorReason());
}
}

// Llegeix les dades de la database i gestiona el moviment dels
servomotors
void setPersiana(String path, Servo &persiana, String modeFunc, int
intLlum ){
    String accio = "";
    String estat = "";
    // Variables per gestionar el temps d'engegada
    unsigned long tempsInici1 = millis();
    bool fiTemps1 = true;
    unsigned long tempsInici2 = millis();
    bool fiTemps2 = true;
    // Verifica l'estat de la persiana (pujada, mitant, baixada)
    Firebase.RTDB.getString(&fbdo, (path + "/estat").c_str());
    estat = fbdo.stringData();
    Serial.print("Estat persiana: " + estat);
    Serial.println(" de la següent adreça: " + path + " /estat");

    // Control en mode automàtic
    if (modeFunc == "AUTO"){
        // Intensitat llum alta (0-1500) puja persiana al 100%
        if (intLlum <= 1500){
            if (estat == "mitant"){
                // Motor gira en sentit antihorari (puja) durant 2 segons
                persiana.write(180);
                while (millis() - tempsInici1 < 2000) { }
                // Atura el motor i escriu a la database l'estat
                persiana.write(90);
                Firebase.RTDB.setString(&fbdo, (path + "/estat").c_str(),
                "pujada");
            }
        }
    }
}
```

Annexos

```
else if (estat == "baixada"){
    // Motor gira en sentit antihorari (puja) durant 4 segons
    persiana.write(180);
    while (millis() - tempsInicil < 4000) { }
    // Atura el motor i escriu a la database l'estat
    persiana.write(90);
    Firebase.RTDB.setString(&fbdo, (path + "/estat").c_str(),
        "pujada");
    }
}
// Intensitat llum mitja (1500-2500) puja persiana al 50%
else if ((intLlum >= 1500) && (intLlum < 2500)) {
    if (estat == "pujada"){
        // Motor gira en sentit horari (baixa) durant 2 segons
        persiana.write(0);
        while (millis() - tempsInicil < 2000) { }
        // Atura el motor i escriu a la database l'estat
        persiana.write(90);
        Firebase.RTDB.setString(&fbdo, (path + "/estat").c_str(),
            "mitant");
        }
    }
else if (estat == "baixada"){
    // Motor gira en sentit antihorari (puja) durant 2 segons
    persiana.write(180);
    while (millis() - tempsInicil < 2000) { }
    // Atura el motor i escriu a la database l'estat
    persiana.write(90);
    Firebase.RTDB.setString(&fbdo, (path + "/estat").c_str(),
        "mitant");
    }
}
// Intensitat llum baixa (2500-4095) baixa persiana al 100%
else if (intLlum >= 2500) {
    if (estat == "pujada"){
        // Motor gira en sentit horari (baixa) durant 4 segons
        persiana.write(0);
        while (millis() - tempsInicil < 4000) { }
        // Atura el motor i escriu a la database l'estat
        persiana.write(90);
        Firebase.RTDB.setString(&fbdo, (path + "/estat").c_str(),
            "baixada");
        }
    }
else if (estat == "mitant"){
    // Motor gira en sentit horari (baixa) durant 2 segons
    persiana.write(0);
    while (millis() - tempsInicil < 2000) { }
```

Annexos

```
// Atura el motor i escriu a la database l'estat
    persiana.write(90);
    Firebase.RTDB.setString(&fbdo, (path + "/estat").c_str(),
        "baixada");
    }
}
}
else if (modeFunc == "MANUAL"){
    // Obté la dada amb l'acció a realitzar
    Firebase.RTDB.getString(&fbdo, (path + "/accio-manual").c_str());
    accio = fbdo.stringData();
    Serial.print("Llegint valor: " + accio);
    Serial.println(" de la següent adreça: " + path + "/accio-manual");

    // Si l'acció és baixar, el motor gira en sentit horari
    if(accio == "baixar"){
        // Configura motor per tal que giri en sentit horari
        persiana.write(0);
        // Espera fins que es compleixin els 4 segons o es rebí un senyal
        per parar/pujar
        while (millis() - tempsInici1 < 4000) {
            // Verifica si s'ha rebut un senyal des de la base de dades per
            parar/pujar
            if (Firebase.RTDB.getString(&fbdo, (path + "/accio-manual").c_str())) {
                accio = fbdo.stringData();
                // Si s'ha rebut un senyal per parar...
                if (accio == "parar") {
                    // ...atura el motor i el rellotge
                    persiana.write(90);
                    fiTemps1 = false;
                    break;
                }
                // Si s'ha rebut un senyal per pujar...
            } else if (accio == "pujar") {
                // ...canvia el sentit de gir i atura el rellotge
                persiana.write(180);
                fiTemps1 = false;
                break;
            }
        }
    }
}
// Si es compleix el temps, atura el motor i escriu en la database
"parar"
if (fiTemps1 == true) {
    Firebase.RTDB.setString(&fbdo, (path + "/accio-manual").c_str(),
        "parar");
    persiana.write(90);
}
}
```

Annexos

```
// Si l'acció és pujar, el motor gira en sentit antihorari
if(accio == "pujar"){
    // Configura motor per tal que giri en sentit antihorari
    persiana.write(180);
    // Espera fins que es compleixin els 5 segons o es rebi un senyal
    per parar/baixar
    while (millis() - tempsInici2 < 4000) {
        // Verifica si s'ha rebut un senyal des de la base de dades per
        parar/baixar
        if (Firebase.RTDB.getString(&fbdo, (path + "/accio-manual").c_str())) {
            accio = fbdo.stringData();
            // Si s'ha rebut un senyal per parar...
            if (accio == "parar") {
                // ...atura el motor i el rellotge
                persiana.write(90);
                fiTemps2 = false;
                break;
            }
            // Si s'ha rebut un senyal per baixar...
            else if (accio == "baixar") {
                // ...canvia el sentit de gir i atura el rellotge
                persiana.write(0);
                fiTemps2 = false;
                break;
            }
        }
    }
}
// Si es compleix el temps, atura el motor i escriu en la database
"parar"
if (fiTemps2 == true) {
    Firebase.RTDB.setString(&fbdo, (path + "/accio-manual").c_str(),
    "parar");
    persiana.write(90);
}
}
// Si l'acció és parar, atura el motor
else{
    persiana.write(90);
}
}
}

// Llegeix la petició d'engegada/parada de la depuradora i actua
void activarDepuradora(String path){
    String nextEstat = "";
    // Obté valor de la base de dades (ON/OFF)
    if (Firebase.RTDB.getString(&fbdo, (path).c_str())){
        nextEstat = fbdo.stringData();
    }
}
```

Annexos

```
Serial.println();
Serial.print("Llegint valor: " + nextEstat);
Serial.println(" de la següent adreça: " + path);
Serial.println();

if(nextEstat=="ON"){
    // Engega el motor de la depuradora
    digitalWrite(pinDepuradora, HIGH);
}
else{
    // Apaga el motor de la depuradora
    digitalWrite(pinDepuradora, LOW);
}
}
else {
    // Indica el motiu de l'error en la lectura de la database
    Serial.println("ERROR");
    Serial.println("CAUSA: " + fbdo.errorReason());
}
}

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

    // Inicialitza sensor intensitat llum
    pinMode(pinSensorLlum_Menj, INPUT);
    pinMode(pinSensorLlum_Hab, INPUT);

    // Inicialitza servomotors persianes
    persiana1.attach(pinPersiana1);
    persiana1.write(90);
    persiana2.attach(pinPersiana2);
    persiana2.write(90);

    // Inicialitza motor depuradora
    pinMode(pinDepuradora, OUTPUT);

    // Inicialitza connexió Wi-Fi
    initWiFi();
    // Assigna la clau de l'API
    config.api_key = API_KEY;
    // Assigna les credencials d'inici de sessió de l'usuari
    auth.user.email = USER_EMAIL;
    auth.user.password = USER_PASSWORD;
    // Assigna la URL de la database
    config.database_url = DATABASE_URL;
    // Configura la reconexió automàtica del Wi-Fi en cas de desconnexió
    Firebase.reconnectWiFi(true);
    fbdo.setResponseSize(4096);
}
```

Annexos

```
// Assigna la funció de callback per a la tasca de generació de token
de llarga durada
config.token_status_callback = tokenStatusCallback;
// Assigna el màxim nombre d'intents en la generació del token
config.max_token_generation_retry = 5;
// Inicialitza la llibreria amb l'autenticació i la configuració de
Firebase
Firebase.begin(&config, &auth);
// Espera mentre obté user UID
Serial.println("Obtenint UID usuari");
while ((auth.token.uid) == "") {
  Serial.print('.');
  delay(1000);
}
// Imprimeix user UID
uid = auth.token.uid.c_str();
Serial.print("UID usuari: ");
Serial.println(uid);

// Adreça a la database
databasePath = "/UsersData/" + uid;

// Adreces als diferents elements de la database
persianaPath = databasePath + "/persianes";
depuradoraPath = databasePath + "/depuradora";
}

void loop(){
  // Espera el temps de mostreig seleccionat
  if (Firebase.ready() && (millis() - millisCicleAnterior > tempsMostreig
|| millisCicleAnterior == 0)){
    millisCicleAnterior = millis();

    // Realitza les mesures dels sensors de llum
    // Entrades analògiques del ESP32 tenen 12 bits de resolució
    // Per tant, es poden obtenir valors entre 0 i 4095
    intensitatLlum_Menj = analogRead(pinSensorLlum_Menj);
    intensitatLlum_Hab = analogRead(pinSensorLlum_Hab);

    // Envia el resultat de la mesura a la base de dades
    enviaLlum(persianaPath + "/persiana-menjador/intensitat-llum",
    intensitatLlum_Menj);
    enviaLlum(persianaPath + "/persiana-hab/intensitat-llum",
    intensitatLlum_Hab);
  }
}
```

Annexos

```
// Llegeix dades de la database i gestiona el motor corresponent
modeFuncPersiana(persianaPath + "/persiana-menjador/mode-func",
  "menj");
setPersiana(persianaPath + "/persiana-menjador", persiana1,
  modeFunc_Menj, intensitatLlum_Menj);

modeFuncPersiana(persianaPath + "/persiana-hab/mode-func", "hab");
setPersiana(persianaPath + "/persiana-hab", persiana2, modeFunc_Hab,
  intensitatLlum_Hab);

activarDepuradora(depuradoraPath);
}
}
```

2.11. Codi Arduino per al microcontrolador d'il·luminació

```
/****** LLIBRERIES & FITXERS *****/
// Llibreries de connexió al Wi-Fi i a la Firebase Database
#include <Arduino.h>
#include <WiFi.h>
#include <WiFiMulti.h>
#include <Firebase_ESP_Client.h>
#include <Wire.h>
#include "data.h"
// Llibreria tira LED NeoPixel
#include <Adafruit_NeoPixel.h>

/****** VARIABLES & OBJECTES *****/
// Objectes Firebase
FirebaseData fbdo;      // Database
FirebaseAuth auth;     // Autenticació
FirebaseConfig config; // Configuració

// Variable per guardar USER UID
String uid;

// Variable per guardar les adreces a la database
String databasePath;

// Objecte Wi-Fi
WiFiMulti wifiMulti;
const uint32_t TempsEsperaWifi = 5000;

// Variables temporals (temps de mostreig)
unsigned long millisCicleAnterior = 0;
unsigned long tempsMostreig = 2000;

/***** Llums ****/
//---> Llum menjador
#define pin_Menj 13
#define Quantitat_Menj 49
Adafruit_NeoPixel llum_Menj = Adafruit_NeoPixel(Quantitat_Menj, pin_Menj,
NEO_GRB + NEO_KHZ800);
String pathLed_Menj;

//---> Llum jardí
#define pin_Jardi 14
#define Quantitat_Jardi 15
Adafruit_NeoPixel llum_Jardi = Adafruit_NeoPixel(Quantitat_Jardi,
pin_Jardi, NEO_GRB + NEO_KHZ800);
String pathLed_Jardi;
```

Annexos

```
//---> Llum piscina
#define pin_Pisc 12
#define Quantitat_Pisc 7
Adafruit_NeoPixel llum_Pisc = Adafruit_NeoPixel(Quantitat_Pisc, pin_Pisc,
NEO_GRB + NEO_KHZ800);
String pathLed_Pisc;

//---> Llum rebedor (automàtica)
#define pin_Rebedor 5
#define Quantitat_Rebedor 9
Adafruit_NeoPixel llum_Rebedor = Adafruit_NeoPixel(Quantitat_Rebedor,
pin_Rebedor, NEO_GRB + NEO_KHZ800);
String pathLed_Rebedor;

//---> Llum cuina (automàtica)
#define pin_Cuina 4
#define Quantitat_Cuina 15
Adafruit_NeoPixel llum_Cuina = Adafruit_NeoPixel(Quantitat_Cuina,
pin_Cuina, NEO_GRB + NEO_KHZ800);
String pathLed_Cuina;

//---> Llum WC (automàtica)
#define pin_WC 15
#define Quantitat_WC 10
Adafruit_NeoPixel llum_WC = Adafruit_NeoPixel(Quantitat_WC, pin_WC,
NEO_GRB + NEO_KHZ800);
String pathLed_WC;

/***** FUNCIONS *****/
// Inicialitza Wi-Fi
void initWiFi() {
    // Xarxes Wi-Fi definides
    wifiMulti.addAP(ssid_1, contrasenya_1);
    wifiMulti.addAP(ssid_2, contrasenya_2);
    wifiMulti.addAP(ssid_3, contrasenya_3);
    wifiMulti.addAP(ssid_4, contrasenya_4);

    WiFi.mode(WIFI_STA);
    Serial.print("Connectant a Wi-Fi ..");
    while (wifiMulti.run(TempsEsperaWifi) != WL_CONNECTED) {
        Serial.print(".");
    }
    Serial.println(WiFi.localIP());
    Serial.println();
}
```

Annexos

```
// Llegeix les dades de la database i gestiona la tira LED corresponent
void setLed(String path, Adafruit_NeoPixel &llum){
  String nextEstat ="";
  // Obté el valor de l'estat de la tira LED (ON/OFF)
  if (Firebase.RTDB.getString(&fbdo, (path + "/estat").c_str())){
    nextEstat = fbdo.stringData();
    Serial.print("Llegint valor: " + nextEstat);
    Serial.print(" de la següent adreça: ");
    Serial.println(path + "/estat");

    if(nextEstat=="ON"){
      // Llegeix i configura intensitat de la tira LED
      setLedIntensitat(path + "/intensitat", llum);
      // Llegeix i configura color de la tira LED
      setLedColor(path + "/color", llum);
    }
    else{
      // Apaga tira LED
      llum.clear();
      llum.show();
    }
    delay(10);
  }
  else {
    // Indica el motiu de l'error en la lectura de la database
    Serial.println("ERROR");
    Serial.println("CAUSA: " + fbdo.errorReason());
  }
}

// Gestiona les llums automàtiques (rebedor i cuina) en funció de l'estat
del sensor de presència
void setAutoLed(String path, Adafruit_NeoPixel &llum, int estat){
  // Si detecta presència, encén tira LED
  if(estat == 1){
    //Llegeix i configura intensitat de la tira LED
    setLedIntensitat(path + "/intensitat", llum);
    // Llegeix i configura color de la tira LED
    setLedColor(path + "/color", llum);
  }
  else{
    // Apaga tira LED
    llum.clear();
    llum.show();
  }
}
}
```

Annexos

```
// Llegeix valor intensitat des de la database i el configura
void setLedIntensitat(String path, Adafruit_NeoPixel &llum) {
  // Obté valor de la base de dades
  if (Firebase.RTDB.getInt(&fbdo, path.c_str())) {
    Serial.println("LED Intensitat: " + String(fbdo.intData()));

    // Factor de conversió de 0-100 a 0-255
    int intensitat = fbdo.intData() * 2.55;
    // Ajusta la intensitat del LED
    llum.setBrightness(intensitat);
  }
  else {
    // Indica el motiu de l'error en la lectura de la database
    Serial.println("ERROR");
    Serial.println("CAUSA: " + fbdo.errorReason());
  }
}

// Llegeix valor del color des de la database i el configura
void setLedColor(String path, Adafruit_NeoPixel &llum) {
  // Adreça per a la lectura de cada valor del color (RGB)
  String path_r = path + "/r";
  String path_g = path + "/g";
  String path_b = path + "/b";
  // Llegeix el valor R del color i l'emmagatzema com a int
  if (Firebase.RTDB.getInt(&fbdo, path_r.c_str())) {
    Serial.println("LED color R: " + String(fbdo.intData()));
    int r = fbdo.intData();
    // Llegeix el valor G del color i l'emmagatzema com a int
    if (Firebase.RTDB.getInt(&fbdo, path_g.c_str())) {
      Serial.println("LED color G: " + String(fbdo.intData()));
      int g = fbdo.intData();
      // Llegeix el valor B del color i l'emmagatzema com a int
      if (Firebase.RTDB.getInt(&fbdo, path_b.c_str())) {
        Serial.println("LED color B: " + String(fbdo.intData()));
        int b = fbdo.intData();

        // Configura cada pixel de la tira LED amb el color seleccionat
        uint32_t color = llum.Color(r, g, b);
        for (int i = 0; i < llum.numPixels(); i++) {
          llum.setPixelColor(i, color);
        }
        // Encén la tira LED
        llum.show();
      }
    }
  }
}
```

Annexos

```
else {
    // Indica el motiu de l'error en la lectura de la database
    Serial.println("ERROR");
    Serial.println("CAUSA: " + fbdo.errorReason());
}
}

// Obté el valor de la lectura del sensor PIR (presència)
int llegirPresencia (String path){
    int valor = 0;
    if (Firebase.RTDB.getInt(&fbdo, path.c_str())){
        valor = fbdo.intData();
        Serial.print("Llegint valor: ");
        Serial.print (valor);
        Serial.print(" de la següent adreça: ");
        Serial.println(path);
    }
    else {
        // Indica el motiu de l'error en la escriptura a la database
        Serial.println("ERROR");
        Serial.println("CAUSA: " + fbdo.errorReason());
    }
    return valor;
}

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

    // Inicialitza connexió Wi-Fi
    initWiFi();
    // Assigna la clau de l'API
    config.api_key = API_KEY;
    // Assigna les credencials d'inici de sessió de l'usuari
    auth.user.email = USER_EMAIL;
    auth.user.password = USER_PASSWORD;
    // Assigna la URL de la database
    config.database_url = DATABASE_URL;
    // Configura la reconexió automàtica del Wi-Fi en cas de desconnexió
    Firebase.reconnectWiFi(true);
    fbdo.setResponseSize(4096);
    // Assigna la funció de callback per a la tasca de generació de token
    // de llarga durada
    config.token_status_callback = tokenStatusCallback;
    // Assigna el màxim nombre d'intents en la generació del token
    config.max_token_generation_retry = 5;
    // Inicialitza la llibreria amb l'autenticació i la configuració de
    // Firebase
    Firebase.begin(&config, &auth);
```

Annexos

```
// Espera mentre obté user UID
Serial.println("Obtenint UID usuari");
while ((auth.token.uid) == "") {
  Serial.print('.');
  delay(1000);
}
// Imprimeix user UID
uid = auth.token.uid.c_str();
Serial.print("UID usuari: ");
Serial.println(uid);

// Adreça a la database
databasePath = "/UsersData/" + uid;

// Adreces als diferents elements de la database
pathLed_Menj = databasePath + "/llums" + "/llum-menjador";
pathLed_Jardi = databasePath + "/llums" + "/llum-jardi";
pathLed_Pisc = databasePath + "/llums" + "/llum-piscina";
pathLed_Rebedor = databasePath + "/llums" + "/llum-rebedor";
pathLed_Cuina = databasePath + "/llums" + "/llum-cuina";
pathLed_WC = databasePath + "/llums" + "/llum-wc";
}

void loop(){
  // Espera el temps de mostreig seleccionat
  if (Firebase.ready() && (millis() - millisCicleAnterior > tempsMostreig
  || millisCicleAnterior == 0)){
    millisCicleAnterior = millis();

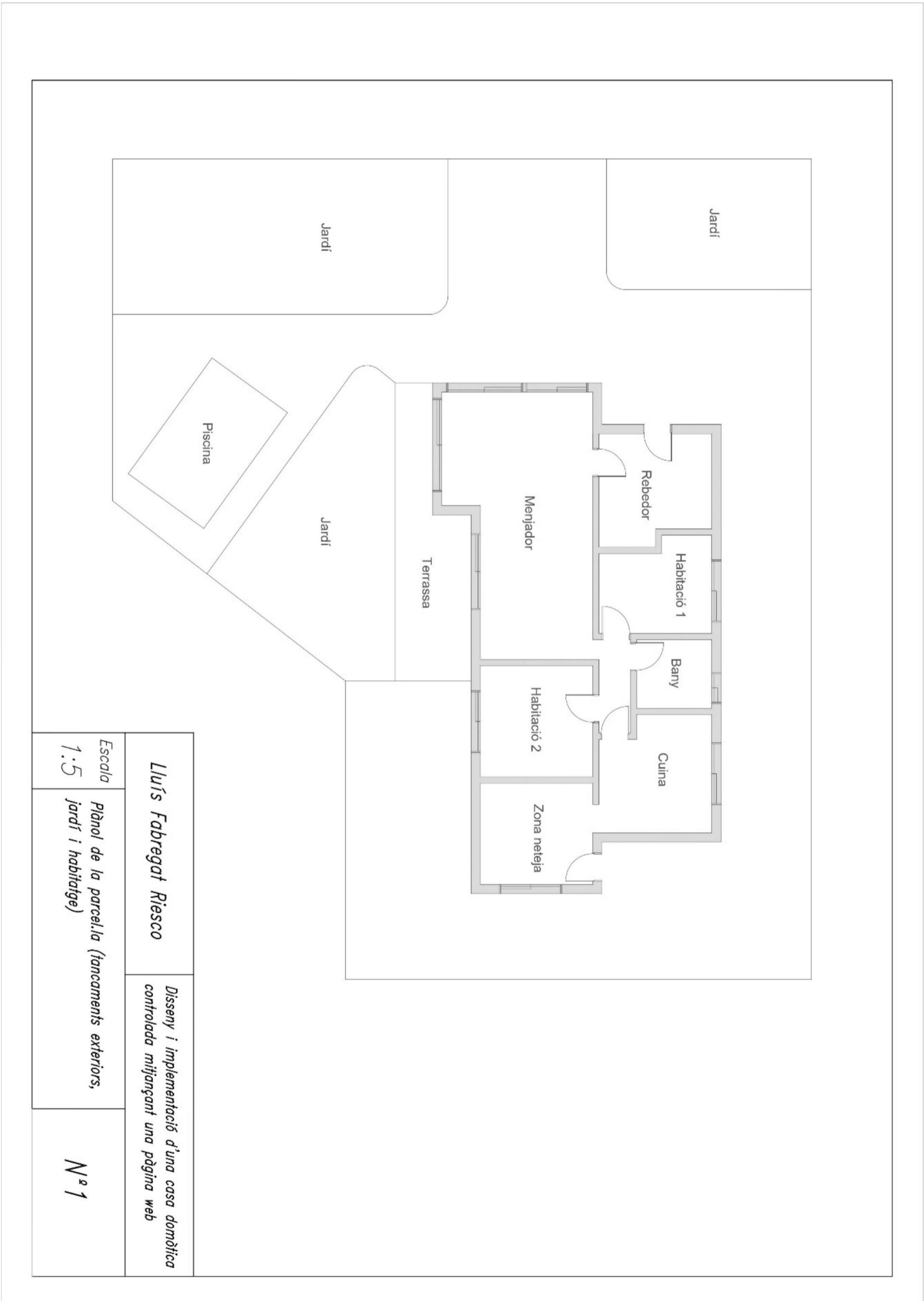
    // Obté les dades de presència de la database
    int pir_estat_rebedor = llegirPresencia(pathLed_Rebedor + "/estat");
    int pir_estat_cuina = llegirPresencia(pathLed_Cuina + "/estat");
    int pir_estat_wc = llegirPresencia(pathLed_WC + "/estat");

    // Gestiona la il·luminació LED automàtica
    setAutoLed(pathLed_Rebedor, llum_Rebedor, pir_estat_rebedor);
    setAutoLed(pathLed_Cuina, llum_Cuina, pir_estat_cuina);
    setAutoLed(pathLed_WC, llum_WC, pir_estat_wc);

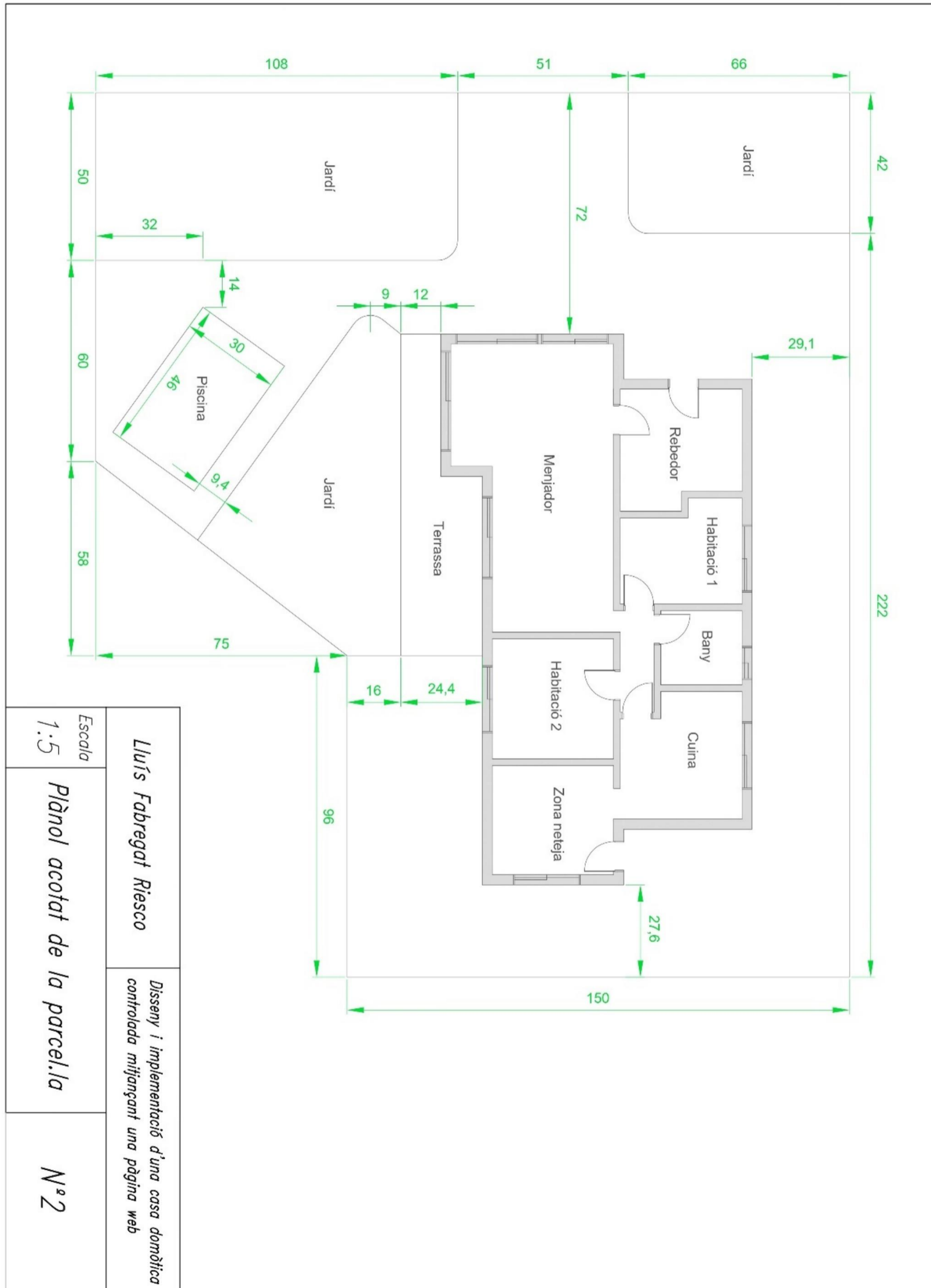
    // Llegeix dades de la database i gestiona el LED corresponent
    setLed(pathLed_Menj, llum_Menj);
    setLed(pathLed_Jardi, llum_Jardi);
    setLed(pathLed_Pisc, llum_Pisc);
  }
}
```

3. PLÀNOLS

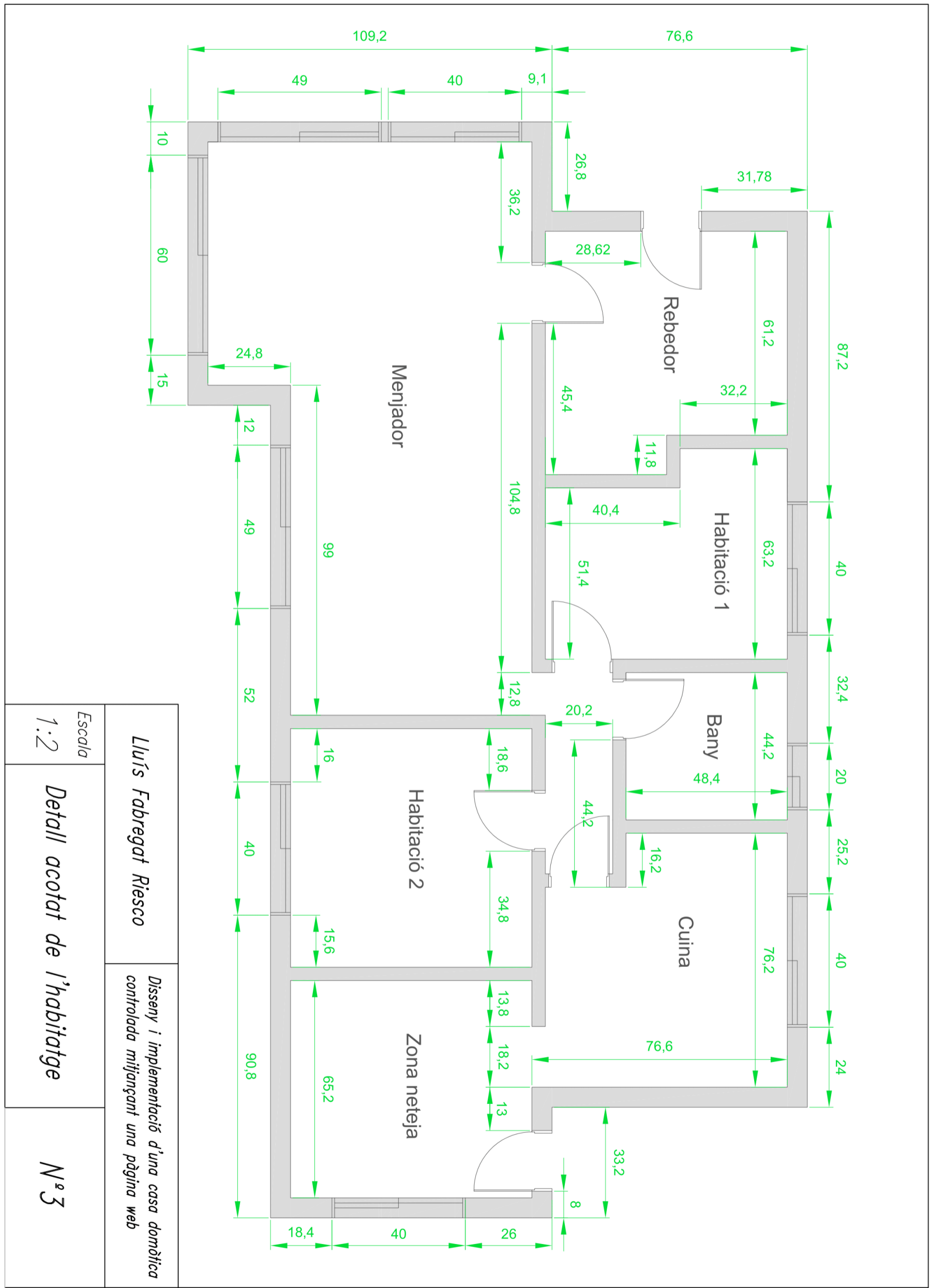
3.1. Plànol de la parcel·la (tancaments exteriors, jardí i habitatge)



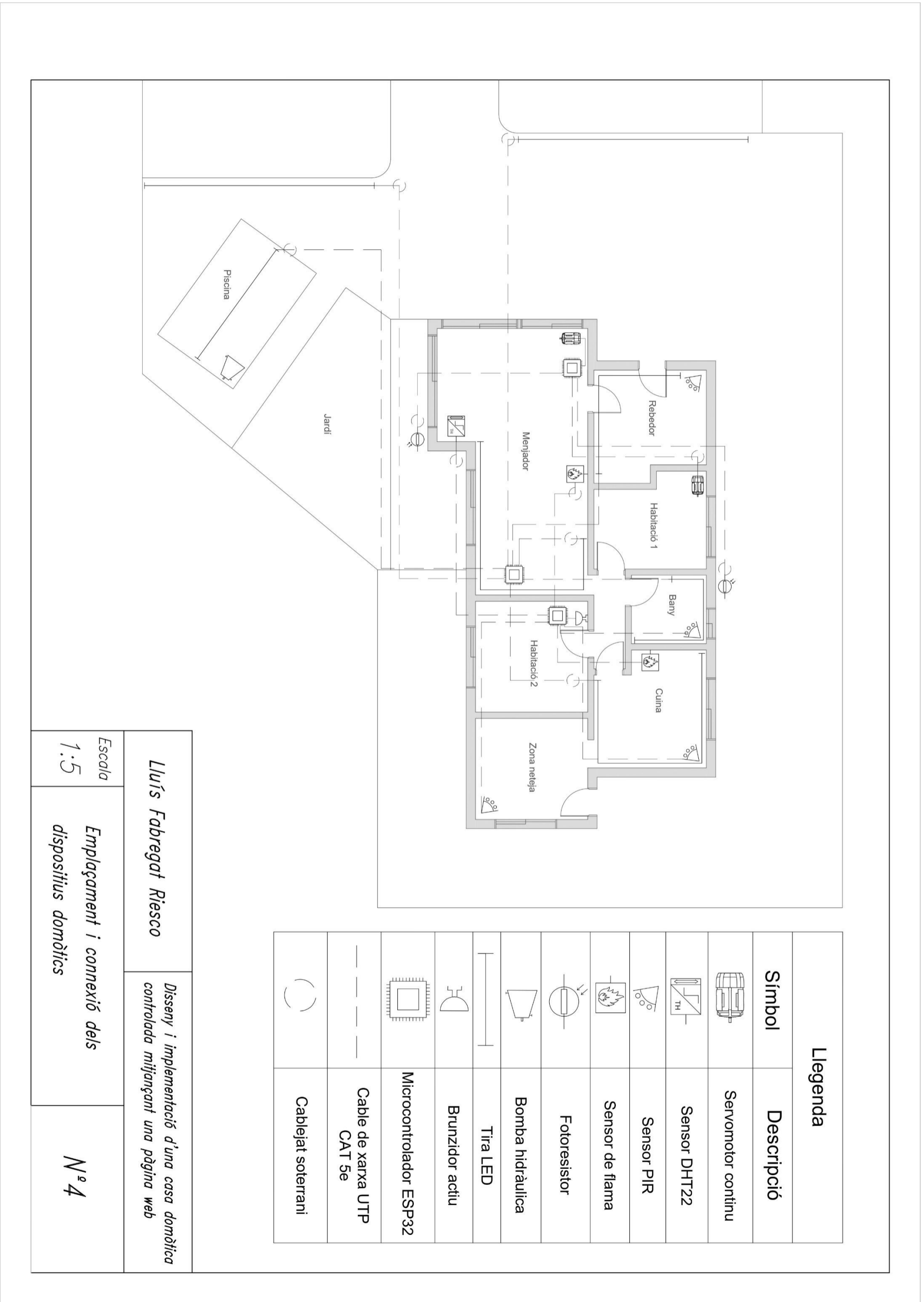
3.2. Plànol acotat de la parcel·la



3.3. Detall acotat de l'habitatge



3.4. Emplaçament i connexió dels dispositius domòtics



Lluís Fabregat Riesco *Disseny i implementació d'una casa domòtica controlada mitjançant una pàgina web*

Escola 1.5 **Emplaçament i connexió dels dispositius domòtics**

Nº 4

4. AMIDAMENTS

Amidaments

CODI	DESCRIPCIÓ	UNITATS	LONGITUD	ALTURA	PARCIAL	TOTAL
CAPÍTOL 1. INSTAL·LACIÓ ELÈCTRICA						
01.01	Cable de xarxa CAT 5e UTP Cable de parells UTP per a xarxa de dades de categoria 6 i coberta normal, les característiques dels cables estan recollides a categoria 6 i l'especificació de classe E de la norma ISO/IEC 11801 i de categoria 6 de la norma EIA/TIA 568B.					
		m	200		200	200

5. PRESSUPOST

5.1. Pressupost instal·lació elèctrica

CAPÍTOL 1. INSTAL·LACIÓ ELÈCTRICA					
Codi	Unitat	Descripció	Quantitat	Preu unitari	Import
EIDI.3ba	ud	Instal·lació domotització habitatge nivell 1			
<p>Instal·lació completa de sistema domòtic descentralitzat amb transmissió de senyal per cable específic (cable de parell trenat) per a l'automatització d'un habitatge en bloc compost de rebedor, saló-menjador, distribuïdor, cuina, 2 dormitoris i 1 bany, amb un nivell de domotització 1 (instal·lacions amb un nivell mínim de dispositius i/o aplicacions domòtiques) segons la taula desenvolupada a partir de la proposta espanyola confeccionada pel CTN202/SC205 "Sistemes electrònics per a habitatges i edificis".</p> <p>La instal·lació inclou sistemes integrats de seguretat: sistema d'alarmes tècniques compost de 2 detectors d'inundació connectats a sondes d'aigua instal·lats a cuina i bany i electrovàlvula de tall d'aigua que serà activada en cas d'inundació, 1 detector de fuga de gasos tòxics a cuina i electrovàlvula de tall de gas i 1 detector iònic de fums instal·lat a cuina per a detecció d'incendis. Inclou sistema de confort: sistema d'accionament de la il·luminació compost d'encesos no regulats a totes les estances, encès i apagat automàtic de cadascuna de les estades mitjançant sensors de presència per detectar moviments i mesurar la intensitat de la llum natural, sistema d'accionament i regulació de totes les persianes de l'habitatge i el tendal del balcó controlador per mitjà de 2 polsadors, una per a cada sentit del motor que envien les ordres als actuadors de les persianes, sistema de control del clima compost per controlador d'estada amb teclat multifunció, termòstat continu per a fred i calor i display LCD i accionament electromotoritzat per a control continu de la temperatura.</p> <p>Preinstal·lació de sistema domòtic compost de: sistema d'alimentació ininterrompuda per evitar que el sistema es quedi sense subministrament elèctric amb una font d'alimentació per generar la tensió de bus i dos acumuladors en paral·lel per a la línia de bus, interfície de comunicació per programar, parametritzar, adreçar o diagnosticar qualsevol dispositiu de bus, cablejat de la instal·lació amb cable de bus de parell trenat i borns de connexió a bus, tots els aparells han de connectar-se al cable de parell trenat i únicament es connecten a la xarxa de 220 V els actuadors, tot això totalment instal·lat, comprovat i en funcionament correcte segons la normativa EA 0026:2006 i la ITC-BT-51 de REBT del 2002.</p>					
1		Materials			
PIDJ.1aaba	u	Polsador d'1 element, amb 2 tecles de comandament, indicador de funcionament i d'estat i camp per retolar, funcions de connexió, commutació, regulació, persiana, activació d'escenes i bloqueig que s'activen amb pulsacions de les tecles provocant l'enviament de telegrams al bus, amb acoblador de bus integrat que li permet connectar-se directament al sistema domòtic per cable específic, fabricat en termoplàstic mat amb acabat en color blanc.	24,000	139,98	3359,52

Pressupost

PIDA.2ab	u	Actuador binari amb marcatge CE per connectar 2 consums independentment per mitjà de contactes normalment oberts, sortides accionables manualment amb interruptors manuals, funcionament com a contacte normalment tancat o normalment obert, temporitzacions, funció d'il·luminació d'escala i preavís de desconexió d'aquesta funció, bloqueig i enllaç adicional o conducció forçada, escenes, resposta per canal davant de qualsevol eventualitat i comportament parametitzable davant d'una fallada en l'alimentació; 16 A de corrent nominal muntatge en carril DIN.	2,000	296,64	593,28
PIDP.3cb	u	Actuador amb marcatge CE per controlar 8 accionaments de persianes independentment, sortides accionables manualment, control de temps de funcionament, pas i pausa, funció de bloqueig de la persiana en un punt determinat, alarmes meteorològiques, control de la posició de les lames i de la altura de la persiana, escenes i funcions d'estat i resposta diferenciades; 6 A de corrent nominal, muntatge en carril DIN.	1,000	671,49	671,49
PIDJ.5cc	u	Polsador multifunció de 4 elements amb display i termòstat, 8 tecles de comandament, indicador de funcionament, 8 indicadors d'estat controlables i camp per retolar, funcions de control de temperatura de calefacció i refrigeració a través del termòstat i de connexió, commutació, regulació, persiana, activació d'escenes, bloqueig i alarma a través del polsador multifunció, necessita mòdul encastable amb acoblador de bus per connectar-se al sistema domòtic per cable específic, fabricat en metall amb acabat en color acer.	1,000	305,16	305,16
PIDJ.7a	u	Mòdul encastable amb marcatge CE per connectar els polsadors multifunció amb termòstat al sistema domòtic per cable específic.	1,000	109,36	109,36

PIDT.6a	u	Accionament electromotoritzat per a vàlvules d'accionament proporcional, amb acoblador de bus integrat i control per microprocessador amb detecció automàtica de carrera de vàlvula per a control continu de la temperatura, funciona amb un termòstat continu del qual rep valors d'1 byte que tradueix en 256 posicions possibles de obertura de vàlvula, incorpora dues entrades binàries per portar l'èmbol a una posició forçada i generar telegrams d'accionament, regulació... que seran enviats al bus. No requereix font d'alimentació, es connecta directament al sistema domòtic per cable específic i és alimentat per la tensió d'aquest, 24 V DC, no disposa de cap botó de programació, per la qual cosa la direcció física se li assigna apropant al cos de l'aparell un imant de programació.	1,000	219,82	219,82
PIDT.7a	u	Imant per a programació sense contacte de la direcció física de l'accionament electromotoritzat per a vàlvules d'accionament proporcional	1,000	7,50	7,50
PIDD.3a	u	Detector de presència en interiors per detectar moviments en una habitació i enviar telegrams de dades a través del sistema domòtic per cable específic, permet activar fins a 3 funcions simultàniament en cas de detecció de moviment, 360 ° d'angle de cobertura, 2,5 m d'alçada òptima de muntatge i radi d'ascens de 7 m si l'alçada de muntatge és de 2,5 m, sensor de lluminositat continu des de 10 fins a 1000 lux per controlar la lluminositat a l'habitació, muntatge al sostre amb caixa d'encastar i acoblador de bus integrat.	8,000	232,02	1856,16
PIDD.6a	u	Caixa de superfície per muntar detectors de presència	8,000	13,15	105,20
PIDS.8a	u	Detector d'inundació per connectar a sonda d'aigua, en detectar aigua acciona el relé inversor que donarà senyal a una entrada binària emetent un senyal acústic i lluminós	2,000	68,85	137,70

Pressupost

PIDS.9a	u	Sonda d'aigua	2,000	10,83	21,66
PIDS11b	u	Electrovàlvula oberta de rearmament automàtic per al tall en el subministrament d'aigua	1,000	138,40	138,40
PIDS.3a	u	Detector de fum òptic a piles per detectar incendis sense flama o amb emissió de fum, en iniciar-se la combustió el detector capta les partícules de fum, acciona el relé inversor i aquest dóna un senyal a una entrada binària alhora que emet senyals acústics i lluminoses	1,000	60,18	60,18
PIDS.4a	u	Junta per situar entre el sostre i el detector de fum	1,000	8,29	8,29
PIDS.5a	u	Mòdul de relé per establir comunicació entre detector de fum i entrada binària del sistema domòtic per cable específic.	1,000	28,70	28,70
PIDS.6a	u	Detector de gas per detectar gasos tòxics i explosius, en detectar aquests gasos acciona el seu relé inversor lliure de potencial i dona un senyal a una entrada binària, disposa d'indicadors acústics i lluminosos.	1,000	86,05	86,05
PIDS11a	u	Electrovàlvula oberta de rearmament automàtic per al tall en el subministrament de gas.	1,000	107,11	107,11
PIDB.3b	u	Entrada binària compacta encastable de 4 canals per enviar telegrams d'accionament, regulació i control de persianes, valors de lluminositat, escenes, regulació o temperatura, proporcionar paràmetres per esdevenir dos comptadors d'accionament o d'impuls o accionar LEDs.	1,000	115,78	115,78
			Subtotal materials		7931,36
2	Mà d'obra				
MOOE.8a	h	Oficial 1a electricista	11,400	25,19	287,17
MOOE.12a	h	Ajudant electricista	11,400	20,22	230,51
			Subtotal mà d'obra		517,68
3	Costos directes complementaris				
	%		2,000	8449,04	168,98

EIDI.1a	ud	Preinstal·lació d'un sistema domòtic descentralitzat de 64 elements			
<p>Preinstal·lació d'un sistema domòtic descentralitzat per cable de bus de dos conductors, per a la gestió i el control de l'energia en instal·lacions en un habitatge o edifici, composta de: 1 font d'alimentació per generar la tensió de bus per a 1 línia (64 elements) i 2 acumuladors en paral·lel per assegurar el funcionament correcte de tots els dispositius de seguretat de la llar en garantir el subministrament elèctric en cas de fallada de tensió de xarxa; interfície de comunicació per programar, parametritzar, direccions o diagnosticar qualsevol dispositiu del bus; fins i tot cablejat, tot això totalment instal·lat, comprovat i en correcte estat de funcionament segons la normativa EA 0026:2006 i la ITC-BT-51 del REBT del 2002.</p>					
<p>1 Materials</p>					
PIDC11bb	u	<p>Font d'alimentació amb marcatge CE, per generar la tensió de bus per a una línia amb un màxim de 64 components de bus, amb filtre integrat per desacoblar la font d'alimentació, borns a cargol endollables connectables a la xarxa, muntatge en carril DIN, tensió d'alimentació 230 V AC, tensió de sortida de 29 V DC i intensitat de sortida màxima de 640 mA, conforme a les especificacions disposades a la norma UNE-EN 61204.</p>	1,000	259,81	259,81
PIDC13a	u	<p>Acumulador de gel plom per connectar-lo a l'entrada de corrent d'emergència de fonts d'alimentació amb connexió d'acumulador, tensió nominal de 12 V DC i 7,2 Ah de capacitat.</p>	2,000	65,83	131,66
PIDF.2b	u	<p>Interfície USB per connectar un dispositiu de programació o de diagnòstic USB al sistema domòtic per cable específic a través d'un connector USB per programar, parametritzar, adreçar o diagnosticar qualsevol dispositiu de bus, a més de controlar el sistema mitjançant el programa de visualització, inclou acoblador de bus, muntatge en carril DIN.</p>	1,000	248,86	248,86
PIAD.4aba	m	<p>Cable de parells UTP per a xarxa de dades de categoria 6 i coberta normal, les característiques dels cables estan recollides a categoria 6 i l'especificació de classe E de la norma ISO/IEC 11801 i de categoria 6 de la norma EIA/TIA 568B.</p>	200,000	0,89	178
Subtotal materials					818,33

Pressupost

2		Mà d'obra			
MOOE.8a	h	Oficial 1a electricista	0,550	25,19	13,85
MOOE.12a	h	Ajudant electricista	0,550	20,22	11,12
				Subtotal mà d'obra	24,97
3		Costos directes complementaris			
	%		2,000	843,30	16,87

5.2. Resum del pressupost

Capítol	Resum	Import	%
1	Instal·lació elèctrica	9.478,19 €	100%
TOTAL EXECUCIÓ MATERIAL		9478,19 €	

13% despeses generals	1.232,16 €
6% benefici industrial	568,69 €
Suma despeses generals + benefici industrial	1.800,86 €
PRESSUPOST D'EXECUCIÓ PER CONTRACTA	11.279,05 €
IVA 21%	2.368,60 €
TOTAL PRESSUPOST	13.647,65 €

Referències

- [1] Institut d'Estudis Catalans, «Diccionari Essencial de la Llengua Catalana.» [En línia]. Available: https://deiec.iec.cat/#c_dom%C3%B2tica_0_54934.
- [2] E-ficiencia, «Domòtica. ¿Qué es y cómo funciona?» [En línia]. Available: <https://e-ficiencia.com/domotica-que-es-y-como-funciona/>.
- [3] Red Eléctrica Española, «Informe del Sistema Eléctrico Español 2020.» 2021.
- [4] BBVA, «¿Cómo la domótica puede ayudar a rebajar la factura de la luz?» 23 Novembre 2023. [En línia]. Available: <https://www.bbva.com/es/salud-financiera/como-la-domotica-puede-ayudar-a-rebajar-la-factura-de-la-luz/>.
- [5] El Periódico, «El 26% de la població mundial del 2050 tindrà més de 60 anys.» 2007.
- [6] AENOR, «AENOR EA0026: "Instal·lacions de sistemes domòtics en habitatges. Prescripcions generals d'instal·lació i avaluació",» 2006.
- [7] AENOR, «UNE-EN 50491: "Requisits generals per a Sistemes Electrònics d'Habitatges i Edificis (HBES) i Sistemes d'Automatització i Control d'Edificis (BACS)",» 2014.
- [8] AENOR, «EN 50090: "Home and Building Electronic Systems",» 2017.
- [9] AENOR, «EN/ISO 16484: "Building Automations and Control Systems",» 2017.
- [10] Dessault Systemes SE, «HomeByMe.» 2023. [En línia]. Available: <https://home.by.me/es/project/15508096>.
- [11] Espressif Systems, Datasheet ESP-WROOM-32, versió 3.4., 2023.
- [12] Philips Lighting, «Philips Bombilla LED.» [En línia]. Available: https://www.lighting.philips.es/prof/tubos-y-lamparas-led/bombillas-led/corepro-ledestandar/LP_CF_LEDLAMPS_EU/family.
- [13] Transmotec, «Motor paso a paso 1,8°.» [En línia]. Available: https://www.transmotec.es/product/shw3920-18-040/?vat=true&gad_source=1&gclid=CjwKCAjwrcKxBhBMEiwAIVF8rNs61ngQ3RB422W_7vNosSn_RSO1ZgscbIXSUC2Hlr06tNR8Y5FahBoCSuoQAvD_BwE.
- [14] Garza, «Detector de humo Standart.» [En línia]. Available: <https://garza.es/products/detector-de-humo-im128a>.
- [15] AZ-Delivery, «Módulo de alarma buzzer pasivo.» [En línia]. Available: <https://www.az-delivery.de/es/products/buzzer-modul-passiv>.
- [16] EfectoLED, «Sensor de luz crepuscular.» [En línia]. Available: <https://www.efectoled.com/es/comprar-detectores-presencia-luminicos/1092-sensor-de-luz-fotoelectronico-crepuscular.html>.
- [17] Adafruit, «Datasheet LED WS2812B.» [En línia]. Available: <https://cdn-shop.adafruit.com/datasheets/WS2812B.pdf>.
- [18] Az-Delivery, «Datasheet servomotor continuo.» [En línia]. Available: https://cdn.shopify.com/s/files/1/1509/1638/files/Micro_Servo_SG90_Datenblatt.pdf?17879403308522525661.
- [19] Az-Delivery, «Datasheet DHT22.» [En línia]. Available: https://cdn.shopify.com/s/files/1/1509/1638/files/DHT_22_-_AM2302_Temperatur_und_Luftfeuchtigkeitssensor_Datenblatt.pdf?11983326290748777409.
- [20] Az-Delivery, «Datasheet sensor PIR.» [En línia]. Available: https://cdn.shopify.com/s/files/1/1509/1638/files/Bewegungsmelder_Modul_Datenblatt.pdf?352607869924912987.
- [21] Az-Delivery, «Datasheet sensor de flama.» [En línia]. Available: https://cdn.shopify.com/s/files/1/1509/1638/files/Flammensensor_Modul_Datenblatt.pdf?5577882845362823060.
- [22] Az-Delivery, «Datasheet buzzer actiu.» [En línia]. Available: https://cdn.shopify.com/s/files/1/1509/1638/files/KY-012_Buzzer_Modul_Aktiv_Datenblatt_AZ-Delivery_Vertriebs_GmbH.pdf?v=1605115550.
- [23] Unit Electronics, «Datasheet bomba de agua sumergible.» [En línia]. Available: <https://uelectronics.com/wp-content/uploads/2018/02/Hoja-de-datos-Bomba-de-Agua-sumergible-Arduino-1.pdf>.

Referències

- [24] Az-Delivery, «Datasheet mòdul fotoresistor,» [En línia]. Available: https://cdn.shopify.com/s/files/1/1509/1638/files/Licht_Sensor_Modul_Datenblatt.pdf?17254477765271537692.
- [25] Highcharts, «Descargar JavaScript Highcharts,» [En línia]. Available: <https://www.highcharts.com/download/>.
- [26] Ionic.io, «Icons,» [En línia]. Available: <https://ionic.io/icons>.
- [27] Google, «Google Fonts,» [En línia]. Available: <https://fonts.google.com/selection?hl=es-419&query=Outfit>.
- [28] «Llibreria WiFi,» [En línia]. Available: <https://github.com/espressif/arduino-esp32/blob/master/libraries/WiFi/src/WiFi.h>.
- [29] «Llibreria WiFiMulti,» [En línia]. Available: <https://github.com/espressif/arduino-esp32/blob/master/libraries/WiFi/src/WiFiMulti.h>.
- [30] «Llibreria Firebase_ESP_Client,» [En línia]. Available: <https://github.com/mobizt/Firebase-ESP-Client>.
- [31] «Llibreria Wire,» [En línia]. Available: <https://www.arduino.cc/reference/en/language/functions/communication/wire/>.
- [32] «Llibreria ESP32Servo,» [En línia]. Available: <https://madhephaestus.github.io/ESP32Servo/annotated.html>.
- [33] «Llibreria Adafruit_Sensor,» [En línia]. Available: https://github.com/adafruit/Adafruit_Sensor.
- [34] «Llibreria Adafruit_NeoPixel,» [En línia]. Available: https://github.com/adafruit/Adafruit_NeoPixel.
- [35] Az-Delivery, «Microcontrolador ESP-WROOM-32,» [En línia]. Available: <https://www.az-delivery.de/es/products/esp32-developmentboard>.
- [36] Amazon, «Servomotor continu,» [En línia]. Available: <https://www.amazon.es/DIYmalls-Rotaci%C3%B3n-continua-Anal%C3%B3gico-Microbit/dp/B09K74X3MZ>.
- [37] Amazon, «Tira LED RGB WS2812B,» [En línia]. Available: <https://www.amazon.es/gp/product/B01CDTE9AW>.
- [38] Az-Delivery, «Mòdul sensor DHT22,» [En línia]. Available: <https://www.az-delivery.de/es/products/dht22-temperatursensor-modul>.
- [39] Az-Delivery, «Mòdul sensor detector de flama,» [En línia]. Available: <https://www.az-delivery.de/es/products/flammensensor-modul>.
- [40] Az-Delivery, «Mòdul bronzidor actiu,» [En línia]. Available: <https://www.az-delivery.de/es/products/buzzer-modul-aktiv>.
- [41] Az-Delivery, «Sensor d'infrarojos passiu,» [En línia]. Available: <https://www.az-delivery.de/es/products/bewegungsmelde-modul>.
- [42] Az-Delivery, «Mòdul fotoresistor 10 kOhms,» [En línia]. Available: <https://www.az-delivery.de/es/products/licht-sensor-modul>.
- [43] Amazon, «Bomba hidràulica 5 V DC,» [En línia]. Available: <https://www.amazon.es/dp/B0814HXWVV>.
- [44] Amazon, «Pins Dupont mascle,» [En línia]. Available: <https://www.amazon.es/gp/product/B096DNYH52>.
- [45] Amazon, «Pins Dupont femella,» [En línia]. Available: <https://www.amazon.es/gp/product/B096DC1J3X>.
- [46] Amazon, «Connector LED de 3 pins + Cable extensió 5m,» [En línia]. Available: <https://www.amazon.es/gp/product/BOC2Z35N96>.
- [47] Amazon, «Bobines 10m cable sòlid,» [En línia]. Available: <https://www.amazon.es/gp/product/B0CB3D24VB>.
- [48] Amazon, «Làmina cartró-ploma 100x70cm,» [En línia]. Available: <https://www.amazon.es/gp/product/B08V99FTKJ>.
- [49] Leroy Merlin, «Taulell fusta MDF 61x122x0,3cm,» [En línia]. Available: <https://www.leroymerlin.es/productos/madera/tablas-y-tableros/tableros-mdf/tablero-de-mdf-crudo-61x122x0-3-cm-anchoxaltogrosor-13677755.html?src=clk>.

Referències

- [50] Leroy Merlin, «Llistó de fusta 13x63 mm x 2,4 m,» [En línia]. Available: <https://www.leroymerlin.es/productos/madera/molduras-listones-y-rodapiés/listones/liston-de-abeto-sin-cepillar-13x63mm-x-2-4m-ancho-x-espesor-x-largo-82200689.html>.
- [51] Bauhaus, «Pintura blanca mate,» [En línia]. Available: <https://www.bauhaus.es/aerosoles-de-colores/mtn-spray-94/p/22886365>.
- [52] Bauhaus, «Pintura gris mate,» [En línia]. Available: <https://www.bauhaus.es/aerosoles-de-colores/mtn-spray-94/p/24549275>.
- [53] Instituto Valenciano de la Edificación, «Base de datos de construcción,» Juliol 2023. [En línia]. Available: <https://bdc.f-ive.es/BDC23/12/EI>.