

TFG Arduino

GRAU MULTIMÈDIA

Projecte

MyDay

PLATAFORMA TELEASSISTÈNCIA PREVENTIVA

Desembre de 2019

Alumne

Pedro Puertas Estivill

Professors

Antoni Morell

Pere Tusset Peiró

Agraïments

*A la meva dona **Susana** i els meus fills **David** i **Alex**, per creure en mi i recolzar-me durant aquests cinc anys, i en general durant tots els anys que portem junts. Han estat anys intensos en els quals semestre a semestre he assolit petits reptes.*

*A **Nacho** i **Gema**, per el seu suport incondicional i desitjar-me sort a cada prova de síntesi.*

*Al meu germà **Àngel**, per compartir la passió per la tecnologia i engrescar-me en el projecte *Seniordomo*, que ens permetrà ajudar a moltes persones que volen envellir a casa.*

I en general a totes les persones que m'estimen i es preocupen per mi.



Pedro Puertas Estivil

Reconeixement - NoComercial – SenseObraDerivada

No es permet un ús comercial de l'obra original ni la generació d'obres derivades

FITXA DEL TREBALL FINAL

Títol del treball:	<i>PLATAFORMA TELEASSISTÈNCIA PREVENTIVA</i>
Nom de l'autor:	<i>Pedro Puertas Estivill</i>
Nom del consultor/a:	<i>Antoni Morell Pérez</i>
Nom del PRA:	<i>Pere Tuset Peiró</i>
Data de lliurament (mm/aaaa):	<i>12/2019</i>
Titulació o programa:	<i>Grau Multimèdia</i>
Àrea del Treball Final:	<i>Arduino</i>
Idioma del treball:	<i>Català</i>
Paraules clau	<i>Arduino, teleassistència , IOT</i>

Resum del Treball (màxim 250 paraules): *Amb la finalitat, context d'aplicació, metodologia, resultats i conclusions del treball*

Els grans avenços a la medicina ha permès reduir notòriament les taxes de mortalitat, el que està produint un grup important de població que requereix un grau important de dependència, que per altra banda vol continuar vivint a casa i no traslladar-se a una residència.

La teleassistència tradicional aporta solucions quan existeix un problema amb la persona (caiguda o botó de SOS), però no permet prevenir patologies o accidents amb antelació. La teleassistència preventiva permet estudiar el comportament de les persones per definir un patró de comportament que serveix de guia per detectar alteracions en la conducta del sènior.

Per assolir l'objectiu, el projecte desenvolupa un sistema hardware a través de la plataforma Arduino i un conjunt de sensors de presència i moviment que envien a un servidor central les dades per què avalú el comportament comparant les dades amb el patró prèviament calculat.

Les dades es poden visualitzar a través d'una aplicació Android o una aplicació web que permet a més realitzar la gestió del sistema, podent gestionar el conjunt de sensors i instal·lacions de la plataforma.

El sistema serveix com a prova de concepte per la plataforma Seniorsdomo, després de les proves finals i el resultat d'aquestes, s'ha corroborat que el projecte és viable tant econòmicament com tècnicament, per la qual cosa, el projecte s'incorporarà a la primavera de 2020.

Abstract (in English, 250 words or less):

The great advances in medicine have allowed to significantly reduce mortality rates, which is producing an important population group that requires a significant degree of dependence that, on the other hand, wants to continue living at home and not moving to a residence.

Traditional tele assistance provides solutions when there is a problem with the person (fall or SOS button), but it does not allow to prevent pathologies or accidents in advance. Preventive tele assistance allows to study the behavior of people to define a pattern of behavior that serves as a guide to detect alterations in the conduct of the elderly.

To achieve the goal, the project develops a hardware system through the Arduino platform and a set of presence and movement sensors that send to a central server all recollected data, which will evaluate the behavior comparing the data with the previous calculated pattern.

The data will be displayed by an Android application and a web application that also allows you to manage the system, configure the set of sensors and installations of the platform.

The system serves as a proof of concept for the Seniordomo platform, after the conclusive tests; it has been corroborated that the project is viable both economically and technically, so the project will be incorporated into the spring of 2020.

Índex

1	Introducció	1
1.1	Context.....	1
1.2	Necessitat.....	2
1.3	Objectiu.....	3
1.4	Descripció de la solució.....	4
1.4.1	Sistema de sensors.....	4
1.4.2	Hub	4
1.4.3	Servidor	5
1.4.4	Aplicació Android	5
1.5	Planificació	6
2	Viabilitat.....	7
2.1	Estudi de la competència.....	7
2.2	Diagrama DAFO.....	8
2.3	Anàlisi de riscos.....	8
2.3.1	R1. Problemes a la detecció d'un dispositiu RF.....	8
2.3.2	R2. Avaria del microcontrolador	8
2.3.3	R3. Programa massa gran per la memòria del microcontrolador	9
2.3.4	R4. Subestimar el temps de desenvolupament	9
2.4	Viabilitat tècnica	9
2.5	Estudi econòmic.....	10
3	Arquitectura	12
3.1	Arduino	12
3.2	ESP32.....	12
3.2.1	Justificació de l'elecció	13
3.3	RXB6.....	13
3.3.1	Justificació de l'elecció	13
4	Anàlisi funcional	14
4.1	Funcionament general	14
4.2	Sensors.....	14
4.2.1	Sensor presència PIR	14
4.2.2	Sensor presència permanent Bluetooth	15

4.2.3	Sensor de moviment Bluetooth	15
4.3	Hub.....	15
4.3.1	Funcionament del hub	15
4.4	Aplicació web	16
4.4.1	Accés a l'aplicació.....	16
4.4.2	Pantalla principal.....	17
4.4.3	Monitoratge	17
4.4.4	Manteniments.....	18
4.5	Aplicació Android	20
4.5.1	Accés a l'aplicació.....	20
4.5.2	Pantalla principal.....	21
4.5.3	Alertes	22
4.5.4	Sensors	23
4.5.5	Perfil d'usuari	24
5	Prototip HUB	25
5.1	Esquema.....	25
5.2	Programa Arduino.....	26
5.3	Tecnologia BLE	28
5.3.1	Advertising payload.....	28
5.3.2	Advertising interval	29
5.4	Detecció de dispositius Bluetooth	30
5.5	Tecnologia RF.....	32
5.6	Detecció de dispositius RF	33
5.7	Relotge en temps real RTC.....	36
5.8	Comunicacions.....	37
5.8.1	get_config.....	38
5.8.2	send_event.....	40
6	BackEnd.....	41
6.1	Protocol Oauth2.....	41
6.1.1	Rols del protocol.....	41
6.1.2	Flux de comunicació	41
6.2	Serveis REST	42
6.3	Accés al model de dades. Objectes DAO	43
6.4	Estructura del backoffice	45

6.5	Processos en background	46
6.5.1	Càlcul de patrons de comportament	46
6.5.2	Càlcul esdeveniment d'excés de temps	49
7	Entorn real i proves	50
7.1	Instal·lació dels sensors	50
7.1.1	Configurant els sensors	50
7.1.2	Configurar els dispositius en el sistema	51
7.2	Supervisió del funcionament	53
7.2.1	Dispositiu presència permanent	54
7.2.2	Dispositiu presència	57
7.2.3	Prova ús d'un objecte.....	57
8	Conclusions i línies de futur	59
8.1	Assoliment d'objectius.....	59
8.2	Possibles millores.....	60
8.2.1	Millores hardware	60
8.2.2	Millores programari.....	60
9	Bibliografia	61
Annexos	65	
A.	Frontend.....	65
A.1	Llibreria oauth2.js.....	65
A.2	Pantalla principal	66
A.3	Manteniments	67
B.	Esquema E-R	68
B.1	Entitats	69
C.	Endpoints	73
D.	Testeig càlcul de patrons	76
D.1	Horaris fixes.....	77

Taules

Taula 1 - Diagrama DAFO	8
Taula 2 - Detall cost fabricació kit teleassistència.....	10
Taula 3 - Detall del càlcul del retorn de la inversió	11
Taula 4 - Característiques ESP32	12
Taula 5 - Configuració sensors Bluetooth	51
Taula 6 - Llista de dispositius proves.....	54
Taula 7 - Proves beacon presència permanent.....	54
Taula 8 - Resultat proves distància.....	56
Taula 9 - Entitat user_type	69
Taula 10 - Entitat user	69
Taula 11 - Entitat token.....	69
Taula 12 - Entitat installation	70
Taula 13 - Entitat device.....	70
Taula 14 - Entitat device_type.....	70
Taula 15 - Entitat event_type	71
Taula 16 - Entitat device_event.....	71
Taula 17 - Entitat event	71
Taula 18 - Entitat profile.....	72

Il·lustracions

Il·lustració 1. Gràfica evolució de la població.....	1
Il·lustració 2. Plataforma teleassistència Seniordomo	2
Il·lustració 3. Esquema de la plataforma MyDay	4
Il·lustració 4 - Diagrama de GANTT del projecte.....	6
Il·lustració 5- Gràfic retorn de la inversió.....	11
Il·lustració 6 - Microcontrolador ESP32.....	12
Il·lustració 7 - Component RXB6	13
Il·lustració 8 - Sensors de presència i moviment.....	14
Il·lustració 9 - Exemple paquet esdeveniment.....	16
Il·lustració 10 - Pantalla entrada a l'aplicació.....	16
Il·lustració 11 - Pantalla principal	17
Il·lustració 12 - Pantalla de Monitoratge.....	17

Il·lustració 13 - Manteniment.....	18
Il·lustració 14 - Crear / Modificar un registre.....	18
Il·lustració 15 - Pantalla d'entrada a l'aplicació	20
Il·lustració 16 - Pantalla principal App.....	21
Il·lustració 17 - Pantalla alertes App	22
Il·lustració 18 - Pantalla sens.....	23
Il·lustració 19 - Pantalla perfil App	24
Il·lustració 20 - Esquema placa prototip HUB	25
Il·lustració 21 - Digrama general programa Arduino.....	26
Il·lustració 22 - Representació de transmissió BLE.....	29
Il·lustració 23 - Diagrama de flux detecció dispositius BT.....	30
Il·lustració 24 - Protocol Manchester (font Viquipèdia).....	33
Il·lustració 25 - Diagrama de flux detecció RF	35
Il·lustració 26 - Esquema flux comunicació HUB - Backoffice	37
Il·lustració 27 - Estructura resposta configuració	38
Il·lustració 28 - Estructura paquet esdeveniment.....	40
Il·lustració 29 - Digrama flux comunicació protocol Oauth2.....	41
Il·lustració 30 - Resultat d'una crida al servei REST.....	42
Il·lustració 31 - Esquema general Backoffice.....	45
Il·lustració 32 - Campana de Gauss	46
Il·lustració 33 - Exemple distribució normal 1.....	47
Il·lustració 34 - Exemple distribució normal 2.....	48
Il·lustració 35 - Diagrama de flux procés de càlcul de patrons	49
Il·lustració 36 – Pantalles aplicació BeaconSet+	50
Il·lustració 37 - BeaconSet+.....	51
Il·lustració 38 - Dispositius donats d'alta dintre del sistema.....	52
Il·lustració 39 - Instal·lació dels sensors a cada electrodomèstic	52
Il·lustració 40 - Estat inicial del dispositiu HUB	53
Il·lustració 41 - Consola del backoffice.....	53
Il·lustració 42 - Detall log HUB proves detecció beacon presència permanent.....	55
Il·lustració 43 - Detecció d'entrades i sortides a l'app	56
Il·lustració 44 - Prova ús d'un objecte	57
Il·lustració 45 - Visualització alerta a l'aplicació web	58

Il·lustració 46 - Acabat final del producte	58
Il·lustració 47 - Art final pantalla monitoratge	66
Il·lustració 48 - Detall gestió alerta	66
Il·lustració 49 - Art final manteniment	67
Il·lustració 50 - Art final manteniment editar registre	67
Il·lustració 51 - Registres generats aleatòriament	78
Il·lustració 52 - Esdeveniments generats aleatoriament	79

Codi font

Codi font 1 - Inicialització escàner Bluetooth	31
Codi font 2 - Funció Callback per gestionar un dispositiu detectat	31
Codi font 3 - Funció control dispositius Bluetooth.....	32
Codi font 4 - Llibreria control dispositius RF	33
Codi font 5 - Inicialització llibreria RCSwitch.....	33
Codi font 6 - Funció rf_handler	34
Codi font 7 - Funcions setTime i getTime.....	36
Codi font 8 - Llibreries accés a internet	37
Codi font 9 - Funció getConfig.....	39
Codi font 10 - Funció send_event	40
Codi font 11 - Exemple d'un objecte DAO	43
Codi font 12 - Càlcul interval patró de comportament	48
Codi font 13 - Exemple crida javascript oauth2.request.....	65
Codi font 14 - Rutina per generar esdeveniment fixes de tot l'any	76
Codi font 15 - Rutina per generar esdeveniments fixes d'un dia	77
Codi font 16 - Exemple control del patró.....	79

1 Introducció

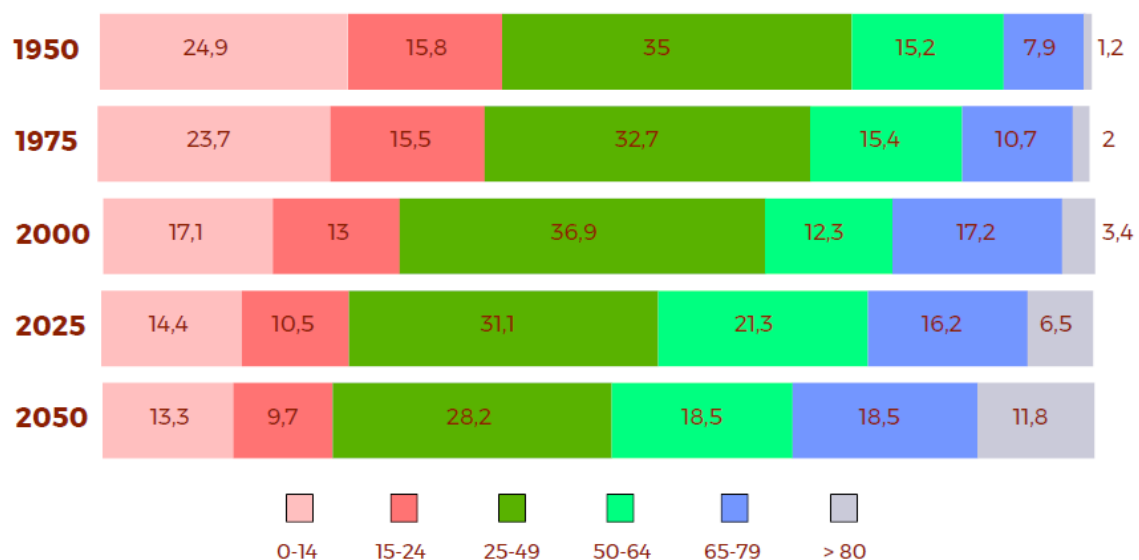
Aquest capítol descriu el context del projecte, la problemàtica que es vol resoldre, com es resoldrà i la metodologia i planificació per a la seva consecució.

1.1 Context

En els últims 20 anys la medicina ha aconseguit allargar la vida pràcticament en 12 o 14 anys amb una qualitat de vida bastant acceptable. Però aquest canvi de paradigma provoca un gran repte que la societat haurà de resoldre en els pròxims anys.

L'envelliment de la població és un gran problema pels governs i la societat en general, que veuen com la piràmide de població s'està invertint, la qual cosa provoca que en els pròxims anys no hi hagi suficient mà d'obra per mantenir l'establishment actual.

Evolució estimada de la població a la Unió europea entre 1950 i 2050



Il·lustració 1. Gràfica evolució de la població

La medicina certament ha aconseguit allargar la vida de la societat, i encara que amb un nivell de vida força acceptable, no està mancada de diversos problemes com el creixement del nombre de persones més grans de 65 anys que viuen soles i que requereixen un cert control diari, o l'aparició de malalties degeneratives com l'Alzheimer, l'esclerosi lateral amiotròfia (ELA)[45] o el càncer. Aquest creixement implica que no hi haurà suficients places en centres residencials, per altra banda, hi ha diversos estudis [1] que demostren que la gran majoria de persones volen envellir a casa.

Hi ha principalment tres causes per les quals la gent gran té necessitats de recolzar-se en el seu entorn [2]:

- Persones que viuen soles en el seu domicili habitual
- Persones amb malalties geriàtriques i/o geronto-psiquiàtriques
- Les que pateixen malalties cròniques degeneratives amb risc a perdre la seva autonomia

Totes elles tenen en comú moltes de les següents casuístiques:

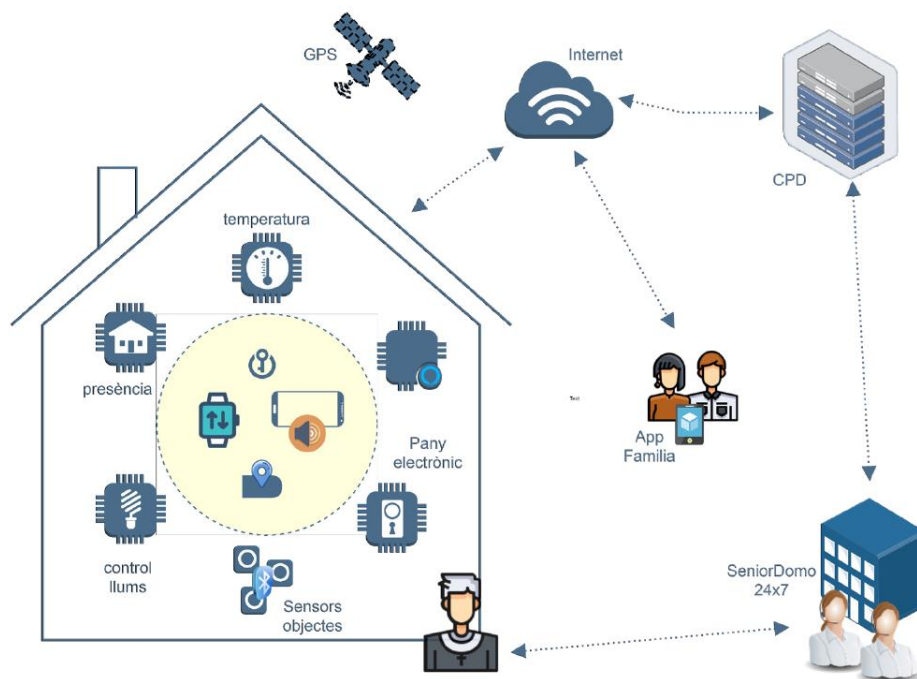
- Soledat
- Control de constants
- Estimulació cognitiva
- Ajuda en les tasques de la llar
- Ajudes a la mobilitat
- Desorientació
- Rehabilitació

1.2 Necessitat

Seniordomo neix arran d'una experiència personal amb el meu pare que va patir ELA [45]. Durant la malaltia vàrem evidenciar les dificultats que tenen les persones grans per envellir a casa quan perden una part de l'autonomia. La primera solució és sempre una residència, però el meu pare, com la majoria de persones grans volia envellir a casa seva, on se sentia còmode i tenia tots els records. Amb aquesta premissa es va iniciar una ronda de xerrades amb diferents famílies que tenien un familiar d'edat avançada i que sense malalties de cura intensiva volien envellir a casa. El resultat d'aquest treball de camp ens va aportar diverses conclusions:

- La gran preocupació són les caigudes.
- La soledat és un factor molt preocupant.
- La necessitat de saber que la persona gran fa vida normal.

Després de mesos de treball i cerca de dispositius es va dissenyar una solució que abasteix principalment els dos primers punts, ja que disposem d'un rellotge amb prou autonomia i un bon algorisme de detecció de caigudes. Per altra banda hem posat en marxa un sistema de suport social que acompanya a les famílies en el seu dia a dia.



Il·lustració 2. Plataforma teleassistència Seniordomo

1.3 Objectiu

L'objectiu d'aquest projecte és el de donar solució al tercer punt, és a dir, la necessitat d'un sistema que analitzi el comportament de les persones grans i detecti comportaments anormals per tal de fer una teleassistència preventiva que complementi l'existent.

Els paràmetres que es volen controlar són:

- Entrades i sortides de l'habitatge
- Moviment dintre de l'habitatge
- Control d'objectes quotidians (nevera, pastiller, armaris, llit)
- Creació d'un patró de comportament a partir de les dades emmagatzemades (temps i nombre de cops que s'ha fet servir)
- Detecció d'activitat fora del patró calculat:
 - Més o menys temps d'estança en una habitació.
 - Més o menys temps fora / dintre de casa.
 - Sobre ús d'objectes.
 - Activitat estant fora de la llar.

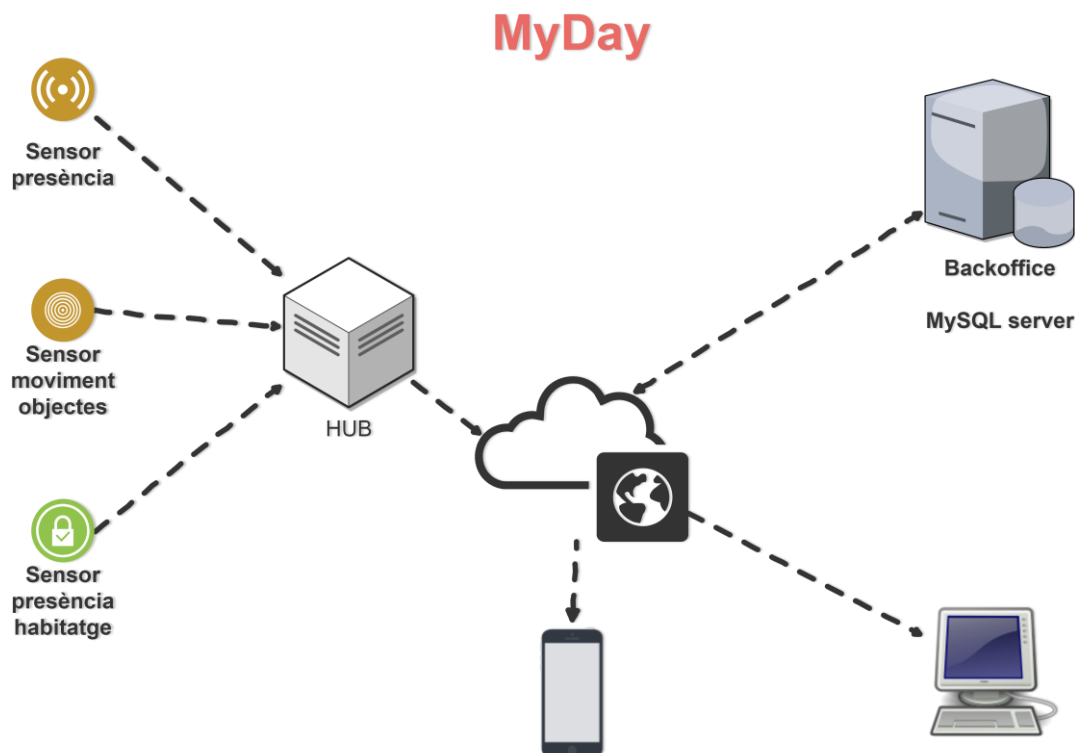
Per analitzar el comportament de les persones, serà necessari incorporar sensors entorn de l'usuari. Es vol que el sistema sigui el menys intrusiu possible, per la qual cosa es descarten elements com càmeres o elements que hagi de dur l'usuari.

Com a requisits addicionals:

- Sistema sense fils.
- Sensors estàndards.
- Capacitat per controlar fins a 10 sensors diferents, per exemple:
 - 4 sensors de presència habitacions.
 - 5 sensors d'objectes.
 - 1 sensor de presència habitatge.
- Cobertura d'habitatges d'un mínim de 100m².
- Aplicació Android multiusuari.
- Comunicacions segures (encriptació de credencials, autenticació).
- Sistema de gestió de la plataforma:
 - Manteniment d'usuaris (altes / baixes / modificacions).
 - Manteniment de dispositius (altes / baixes / modificacions).
 - Monitoratge d'alertes.

1.4 Descripció de la solució

Es planteja una solució basada en Arduino[4] que permeti interaccionar amb sensors de radiofreqüència i/o Bluetooth. Aquests sensors es comuniquen amb un hub que és el que interacciona amb un servidor, que serà l'encarregat d'emmagatzemar les dades, calcular els patrons i proveir la informació a les aplicacions web i Android.



Il·lustració 3. Esquema de la plataforma MyDay

1.4.1 Sistema de sensors

Es preveuen 3 tipus de sensors:

- Presència permanent, beacon[5] Bluetooth
- Entrada a una habitació, sensor infrarojos per radiofreqüència (433 MHz)[7]
- Control d'activitat a través de beacons (Bluetooth) amb acceleròmetre.

Tots els sensors utilitzen tecnologia sense fils i estan alimentats per bateries.

1.4.2 Hub

El sistema Hub serà l'encarregat de comunicar les dades captades dels sensors al servidor, que les emmagatzemarà a la base de dades, gestionarà els patrons de comportament i la detecció d'alertes.

Aquest dispositiu estarà basat en tecnologia Arduino, el model triat ha de tenir prou capacitat per gestionar les comunicacions Bluetooth i radiofreqüència en temps real.

1.4.3 Servidor

1.4.3.1 Backend

- Habilita un servei REST[21][22] per l'intercanvi de dades amb sensors i les aplicacions web i Android.
- Analitza les dades rebudes per generar un patró de comportament.
- Executa un motor d'anàlisi de patrons per la detecció de canvis de comportament.
- Emmagatzema les dades a una base de dades relacional tipus MySQL [6].
- Gestiona l'accés a les dades amb protecció de credencials SHA256 i protocol OAuth2[13][15].

1.4.3.2 Frontend (aplicació web)

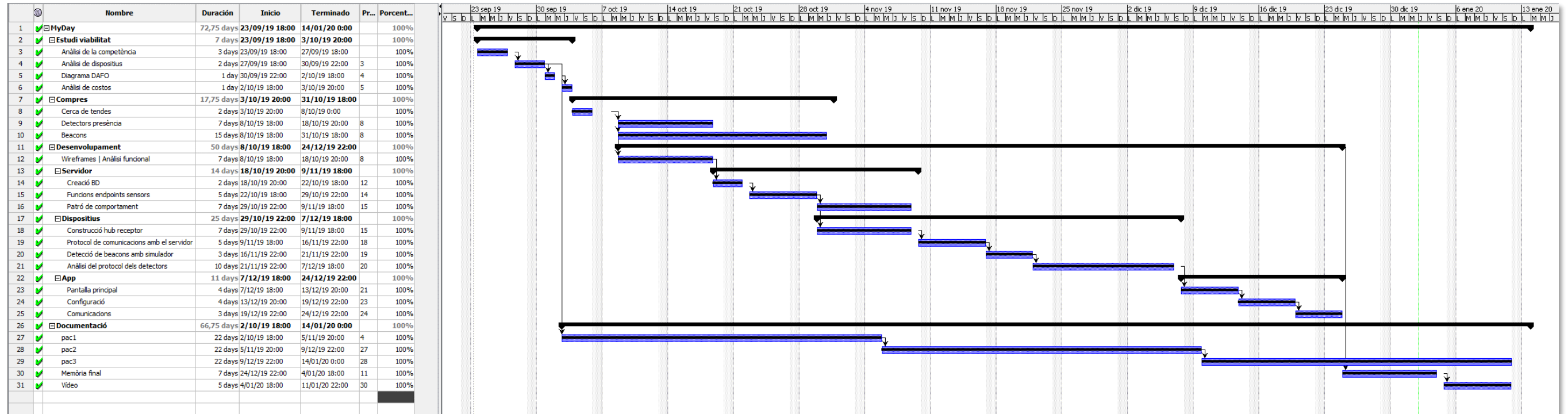
- Habilita una pàgina per accedir a l'aplicació (usuari i paraula de pas).
- Habilita un manteniment de dispositius (alta / baixa / modificació).
- Habilita manteniment per associar dispositius a una instal·lació (alta / baixa / modificació).
- Monitoratge d'alertes
 - Temps inusual a una estança.
 - Utilització inusual d'un objecte.
 - Presència estant fora de casa.

1.4.4 Aplicació Android

L'aplicació Android permet la gestió dels esdeveniments del sènior. Habilita les següents funcionalitats:

- Accés per credencials.
- Històric d'esdeveniments de les últimes 24h.
- Estat actual del sènior (habitació on es troba, temps, etc.).
- Estat dels dispositius:
 - Temps total d'ús.
 - Nombre d'entrades a una habitació.
 - Detall amb data i hora dels esdeveniments detectats.
- Monitoratge d'alertes
 - Temps inusual a una estança
 - Utilització inusual d'un objecte
 - Presència estant fora de casa

1.5 Planificació



II-lustració 4 - Diagrama de GANTT del projecte

2 Viabilitat

2.1 Estudi de la competència

La viabilitat del projecte requereix un estudi previ dels productes de teleassistència existents al mercat per tal d'establir la diferenciació del producte envers la resta. És important destacar que el projecte és un complement a la tecnologia Seniordomo i per tant es realitzarà una anàlisi dels productes que ofereixen un sistema similar de control dels sèniors que puguin ser competència amb el producte MyDay.



El sistema Beprevent es basa en unes etiquetes que s'enganxen a objectes d'ús quotidià. Fan servir tecnologia Bluetooth per la comunicació. Inclou 5 etiquetes i un dispositiu central.

És el competidor més directe en tecnologia, igual que MyDay fa servir beacons per detectar l'ús d'objectes.

No inclou però altres tipus de dispositius com detecció de presència, detecció de fums, gas, etc.

El preu de venda és de 299€ més una quota de 5,99€ mensuals.

<https://beprevent.es/comprar>



El sistema Sensovida és un conjunt de 4 dispositius endollables que s'ubiquen a cada habitació i que permeten detectar la presència d'una persona a través d'una polsera.

Mesuren per tant el comportament de la persona a través del moviment a través de les habitacions. La polsera també té un sensor que permet a l'usuari activar una alarma.

No incorpora detecció d'objectes ni dispositius de detecció de fum, gas, etc.

Té un preu de 39,90€ mensuals.

https://www.idia.es/wp-content/uploads/2016/11/Dossier_Sensovida_2016_Buyer.pdf



Gigaset presenta un sistema de sensors que possibilita un servei de teleassistència.

Inclou tota mena de sensors de detecció de presència, fum, etc. Inclou també sensors adhesius per adherir-los a qualsevol objecte. A la pàgina oficial no anuncien el servei de teleassistència que en canvi es pot trobar per Amazon.

Té un preu de 399€ del Kit més una quota mensual de 14,99€

<https://www.amazon.es/Gigaset-notificaci%C3%B3n-Inteligente-Seguridad-suscripci%C3%B3n/dp/B07LC5T1ZX>

Si analitzem els diferents productes, probablement el més complet és el de Gigaset que inclou tota la varietat de sensors, però en canvi no sembla del tot enfocat a donar un servei complet de teleassistència. En dos d'ells el preu inicial és important, només Sensovida ofereix una quota mensual sense un preu inicial, però el seu cost mensual és excessiu.

2.2 Diagrama DAFO

Debilitats	Amenaces
Desconeixements dels protocols de radiofreqüència. Capacitat de fabricació Manteniment dels dispositius	Inestabilitat econòmica Evolució sistemes actuals de teleassistència Captació de capital per la producció en cadena
Fortaleses	Oportunitats
Cartera de clients Experiència en desenvolupament de sistemes embeguts Agilitat en desenvolupament de producte	Primer producte multisensorial del mercat (presència, objectes, etc.) Millora del producte Seniorsdomo Incorporar tecnologia IoT i IA

Taula 1 - Diagrama DAFO

El diagrama DAFO [3] és una forma de reflexionar sobre la viabilitat d'un projecte, perquè obliga a pensar quins factors poden afectar el nostre projecte, quins són els punts forts del projecte així com les debilitats de forma que es poden pensar eines per reduir les debilitats i aprofitar les fortaleses.

2.3 Anàlisi de riscos

Aquest apart descriu la forma d'actuar sobre els possibles riscos que es puguin produir en el desenvolupament del projecte, poden ser riscos de tipus tècnic o de personal. Realitzar una anàlisi d'aquest tipus permet tenir estudiat com es pot actuar en el cas que es produeixi el risc en qüestió. També pot servir com a eina per actuar en casos de riscos similars no contemplats.

2.3.1 R1. Problemes a la detecció d'un dispositiu RF

És possible que els sensors de radiofreqüència adquirits funcionin amb un estàndard diferent i que el receptor triat no sigui capaç de llegir.

2.3.1.1 Possibles solucions

- Comprar sensors alternatius d'altre marca
- Canviar el receptor

2.3.2 R2. Avaria del microcontrolador

Existeix la possibilitat que en el moment del muntatge o durant les proves, el dispositiu deixi de funcionar, per la qual cosa es podria veure afectada la planificació del projecte.

2.3.2.1 Possibles solucions

Duplicar l'estoc de components de forma que es pugui substituir el component espatllat

Revisar els terminis d'entrega del producte per estimar el temps màxim de restitució del component.

2.3.3 R3. Programa massa gran per la memòria del microcontrolador

Existeix la possibilitat que totes les funcionalitats que es volen implementar en el sistema, requereixin molta memòria i no hi hagi espai disponible a la memòria flash del microcontrolador. Això afectaria greument la viabilitat del projecte.

2.3.3.1 Possibles solucions

Utilitzar variables el més petites possibles

Evitar l'ús de Serial per fer el seguiment del programa, fer-ho servir exclusivament en el procés de proves.

Configurar les diferents opcions de compilació d'Arduino perquè compacti al màxim el codi.

2.3.4 R4. Subestimar el temps de desenvolupament

Existeix la possibilitat que la planificació no sigui real, per la qual cosa perilla l'entrega final del projecte, ja que si no es pot desenvolupar completament el projecte, seria inviable realitzar el conjunt de proves i documentació final.

2.3.4.1 Possibles solucions

Verificar la viabilitat de totes les parts i posteriorment desenvolupar completament el programari, és a dir, a la fase inicial comprovar que realment és possible llegir els dispositius, que es poden establir les comunicacions, etc. Posteriorment es desenvoluparà el programa més en detall per donar-li forma.

2.4 Viabilitat tècnica

Per avaluar la viabilitat tècnica del sistema, primer s'ha de fet una cerca dels diferents elements i la compatibilitat entre ells. Gràcies a la plataforma Arduino, s'han creat un ventall important de microcontroladors compatibles amb capacitats molt variables. Es va trobar el microcontrolador ESP32[12] que permet connexions WIFI i Bluetooth a un preu molt raonable i compatibilitat amb l'entorn Arduino. Una altra característica molt interessant d'aquest microcontrolador és que té un doble nucli, per la qual cosa permet executar dues tasques simultàniament.

Entre el ventall de dispositius per realitzar comunicacions amb freqüències de 433 MHz[31], el dispositiu RXB6 és compatible amb el voltatge de 3,3v que treballa l'ESP32, a més, es va trobar algun article[30] sobre un correcte funcionament en la lectura de dispositius RF.

A partir d'aquest component es podrà generar un prototip per posteriorment crear un circuit imprès (PCB) per abaratir els costos.

Tota la part de servidor s'ha previst desenvolupar-la en Java, per facilitar la integració futura amb el sistema Seniorsdomo, però en tractar-se d'un projecte acadèmic s'ha desenvolupat el projecte des de zero, però amb la intenció d'integrar gran part del codi de la part servidora de gestió dels patrons, etc.

2.5 Estudi econòmic

L'estudi econòmic ha de permetre determinar la viabilitat del projecte, i diversos factors que determinaran el preu final del producte, i per tant la viabilitat d'accés al mercat. Alguns d'aquests factors són:

Cost de fabricació. Indica el cost per fabricar el producte industrialment, sobre aquest preu s'ha d'aplicar un benefici que marcarà el preu final. Com és un producte que ofereix un servei, el benefici es pot afegir en les quotes mensuals i/o en el preu del Kit inicial.

Component	Unitats	Preu U.	Preu
Hub			
ESP32	1	6€	6€
RXB6 433MHz	1	3€	3€
PCB (100 pcs 2€ u)	1	2€	2€
Muntatge (soldadura, manipulació, etc.)	1	2€	2€
Capsa plàstic	1	2€	2€
Alimentador AC/DC	1	2€	2€
Detector presència (Sonoff PIR2)	3	10€	30€
Adhesiu Beacon (Minew E8) [42]	3	4€	12€
		Total	59€

Taula 2 - Detall cost fabricació kit teleassistència

Sobre el preu de cost, s'haurà d'afegir un percentatge per recuperar la inversió del desenvolupament, també s'ha d'afegir un marge de costos estructurals, etc. Per la qual cosa s'afegirà un 50% al preu de venda final.

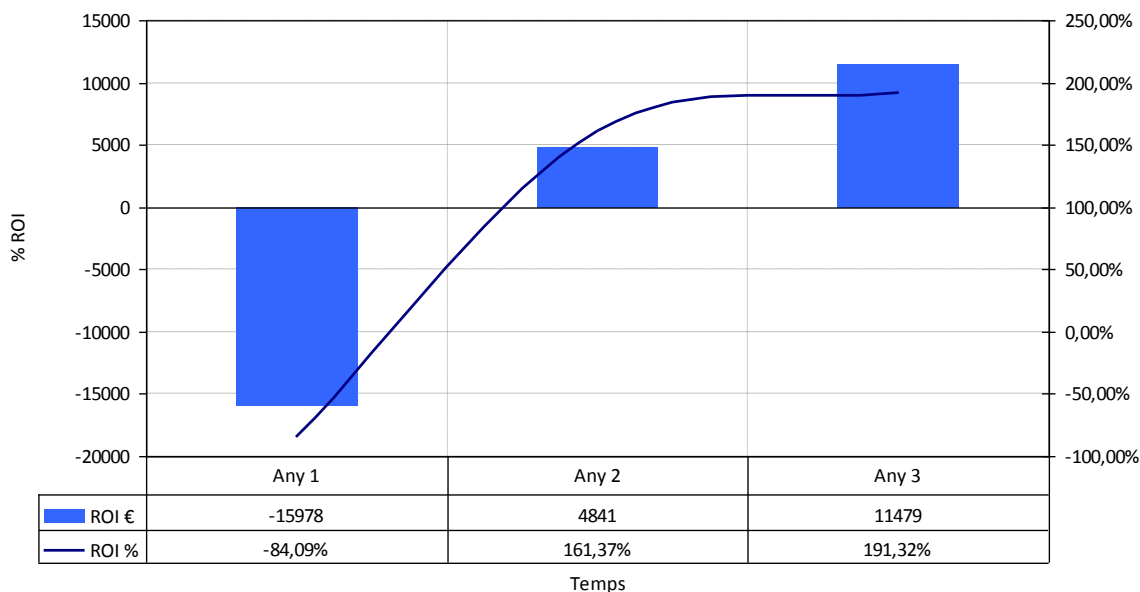
El preu del Kit serà de 119€, per fer-ho més atractiu es vendrà per 49€ el Kit i una quota mensual de 5,99€

Cost de desenvolupament. Indica el cost per crear els prototips i programari, ha de tenir en compte el temps de desenvolupament, materials, etc.

Per obtenir aquests costos s'ha de realitzar una planificació del desenvolupament que inclogui el personal y les hores dedicades. En aquest cas s'han estimat unes 350h que a un preu de 50€ l'hora serien 17.500€

Per recuperar per tant la inversió, s'hauria de dividir els 17.500€ entre el marge de preu (~60€) que hem afegit al producte, la qual cosa indica que s'han de vendre prop de 75 equips per amortitzar la inversió. No s'han tingut en compte costos de promoció de producte, etc. que incrementarien el preu si per exemple es modifica la pàgina web per afegir el nou producte, campanyes a Google ADS, etc.

Càlcul ROI



Il·lustració 5- Gràfic retorn de la inversió

La conclusió és que **el producte és viable i es podria recuperar la inversió entre el segon i tercer any** estimant unes vendes de 25 unitats el primer any, 50 segon any i 100 el tercer any. A un preu raonable i inferior al de la competència.

		Any 1	Any 2	Any 3	
Cost inicial	17.500 €	Cost	19.000 €	3.000 €	6.000 €
Cost fab kit	60 €	Vendes	3.022 €	7.841 €	17.479 €
Preu kit	49 €	Unitats venudes	25	50	100
Quota	5.99 €	ROI €	-15.978 €	4.841	11.479
		ROI %	-84,09%	161,37%	191,32%

Taula 3 - Detall del càlcul del retorn de la inversió

$$\text{Càlcul vendes} = (<Preu kit> * \text{Clients nous}) + (<quota> * \text{clients})$$

3 Arquitectura

En aquest capítol es fa una breu descripció dels components utilitzats per desenvolupar el HUB, concretament el microcontrolador ESP32[11][12] i el receptor de freqüències 433Mhz [7][30]

3.1 Arduino

Arduino és una plataforma per la creació de prototips de codi obert, es compon d'una placa amb un microcontrolador ATMEGA que proveeix ports d'entrada i sortida tant analògics com digitals. Les plaques de desenvolupament es connecten a través del port USB i són compatibles amb els sistemes operatius Linux, Windows i OSx. La plataforma també proveeix un entorn de desenvolupament gratuït que facilita les tasques de desenvolupament i bolcat del codi en el microcontrolador.

Altres fabricants de dispositius, han adoptat la plataforma i faciliten la integració dels seus microcontroladors en el mateix entorn. Aquest és el cas de l'empresa xinesa Espressif, que disposa de diversos models compatibles que ofereixen característiques integrades al mateix processador, com ara connexió wifi o Bluetooth. L'empresa va destacar principalment amb l'aparició del dispositiu ESP8266 [10], un petit mòdul que inicialment permetia a les plaques Arduino tenir capacitat de connexió a xarxes wifi, però al fer servir un processador Atmel, l'empresa el va acabar integrant dintre de la plataforma l'octubre de 2014.

3.2 ESP32

El dispositiu ESP32 és un dispositiu que evoluciona el microcontrolador ESP8266, amb importants millores de capacitat de memòria, processament i gestió de ports.



Il·lustració 6 -
Microcontrolador

Les principals característiques són:

Processador	Xtensa® Dual-Core 32-bit LX6 600 DMIPS
802.11 b/g/n Wi-Fi	HT40 (banda 40Mhz)
Bluetooth	4.2
Freqüència típica	160 / 240 Mhz
SRAM	512 Kbytes
Flaix	SPI, fins a 16 Mbytes
Ports d'entrada / sortida	36
SPI / I2C / I2S / UART	4/2/2/2
RTC	Sí
Temperatura de treball	-40°C a 125°C
Alimentació	3.3v
Preu	

Taula 4 - Característiques ESP32

3.2.1 Justificació de l'elecció

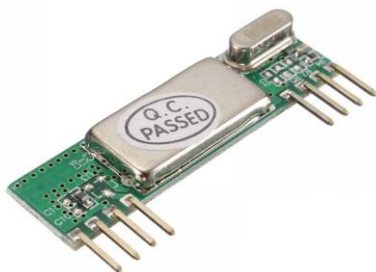
S'ha escollit aquest dispositiu principalment per la seva relació qualitat preu i la integració de pràcticament totes les tecnologies sense fils requerides en el projecte.

Les característiques més rellevants avaluades en la decisió són:

- Bluetooth integrat, possibilita l'accés als dispositius sense necessitat d'afegir cap mòdul extra, la qual cosa estalvia costos de desenvolupament del hardware.
- Mida de la memòria FLASH, aquesta memòria és on s'emmagatzema el programari desenvolupat, el fet de tenir una memòria gran permet tenir un ample marge per afegir futures millores en el programari.
- Relotge en temps real RTC [8], en aquest cas és necessari un rellotge amb prou exactitud per poder mantenir l'hora dels dos sistemes sincronitzats.
- Doble nucli, el fet de disposar de dos nuclis permet executar diverses tasques simultàniament, la qual cosa permetrà dedicar un nucli a la tasca de control de sensors Bluetooth i l'altre als sensors RF.

3.3 RXB6

El dispositiu RXB6 és un dispositiu transceptor que permet la transmissió i recepció de dades a una freqüència de 433Mhz.



Il·lustració 7 - Component RXB6

Les característiques més rellevants són:

- Voltatge de treball : 3.3 a 5v
- Consum 6mA
- Sensibilitat -110 dBm
- Preu aproximat 2€

3.3.1 Justificació de l'elecció

S'ha triat aquest dispositiu perquè permet una perfecta integració amb el microcontrolador ESP32, el fet de poder treballar amb voltatges de 3.3v i la possibilitat de connectar una antena exterior juntament amb una bona relació qualitat preu.

4 Anàlisi funcional

Aquest apartat descriu el funcionament dels apartats més importants que conformen el sistema de gestió de sensors. I serveix com a guia pel disseny dels diferents algorismes en el programari de cadascuna de les parts involucrades.

Per facilitar l'enteniment de la plataforma, s'explicarà un funcionament general i posteriorment el funcionament exacte que tindran els apartats més importants.

4.1 Funcionament general

Diversos sensors (presència per infrarojos, presència Bluetooth, moviment Bluetooth) són gestionats per un hub (Arduino) que recull les dades i les envia a un servidor central.

Cada dada rebuda és emmagatzemada a la base de dades i comparada amb un patró de comportament, si la diferència entre les dades processades i les dades del patró sobrepasa el lílindar del mínim o el màxim, es crea un esdeveniment d'alerta de què hi ha algun tipus de problema amb la instal·lació.

Hi ha dos tipus d'aplicacions que gestionen els esdeveniments d'alerta, per una banda l'aplicació web i per l'altra l'app Android. Totes dues accedeixen a la base de dades per cercar esdeveniments pendents a les últimes 24h.

Cada sensor utilitza una o més tecnologies per executar la seva funcionalitat, per realitzar les comunicacions hi ha sensors Bluetooth i sensors de radiofreqüència, que detecten moviment a través d'un acceleròmetre o a través d'infrarojos.

4.2 Sensors

4.2.1 Sensor presència PIR

El sensor PIR detecta la presència d'un cos de més de 25Kg que desprengui calor a una distància màxima de 10m. Cada cop que detecta un diferencial de calor sobre l'àrea de control (120º) s'emet una trama de 5 paquets indicant que s'ha detectat un cos. Per realitzar les comunicacions, aquests sensors fan servir tecnologia de radiofreqüència a 433 MHz.



Il·lustració 8 - Sensors de presència i moviment

4.2.2 Sensor presència permanent Bluetooth

El sensor de presència permanent emet un senyal cada segon que és captat pel hub, la no presència d'aquest senyal després de 30 segons determina que el sensor està fora de l'abast (fora de casa). Si de nou es detecta el senyal, es determina que el sensor ha tornat a casa.

El sensor emet la seva presència a través de la tecnologia Bluetooth i concretament amb les especificacions beacon[23][25], que permeten rebre algunes dades del dispositiu sense necessitat de realitzar una connexió.

4.2.3 Sensor de moviment Bluetooth

El sensor de moviment emet un senyal cada cop que detecta moviment a través d'un acceleròmetre de tres eixos. Igual que el sensor permanent, es fa servir la tecnologia Bluetooth per comunicar amb el hub que el sensor està actiu. Per reduir el nombre de moviments reals, es consolidaran les dades, per la qual cosa els moviments han de produir-se amb una diferència de 60 segons per ser comptabilitzats.

4.3 Hub

El hub és el dispositiu que comunica amb els sensors i transmet les dades al servidor central, necessita tenir capacitats per llegir informació a través de la tecnologia Bluetooth i per radiofreqüència 433Mhz. S'ha triat el dispositiu ESP32, un dispositiu SOC (System on chip) de la marca Espressif que integra en un únic dispositiu un processador de doble nucli de 32bits, connectivitat Wifi i Bluetooth a més de 32 ports GPIO dels quals 11 són del tipus DAC. Incorpora també un rellotge en temps real RTC i 3 ports UART (sèrie).

Per poder llegir els paquets de radiofreqüència 433Mhz, s'ha seleccionat el transceptor RXB6. El principal avantatge d'aquest dispositiu és que admet una tensió de treball de 3.3v, per la qual cosa és ideal per treballar amb l'ESP32.

Aprofitant que el microcontrolador és de doble nucli, es distribuirà la càrrega de treball entre els dos nuclis, configurant la lectura de dispositius Bluetooth en el nucli 1, i la lectura de dispositius de radiofreqüència en el nucli 2. Les comunicacions wifi amb els servidors també s'implementarà en aquest segon nucli.

4.3.1 Funcionament del hub

Quan el dispositiu s'inicia, connectarà al servidor per rebre la configuració i l'hora actual. La configuració subministra el nombre de dispositius, el tipus i la seva adreça única que els identifica. Un cop rebudes les dades, s'iniciaran les dues tasques de lectura de dispositius.

Cada tipus de dispositiu té un tractament diferent, per la qual cosa quan es detecta el dispositiu, es mira el tipus i s'adequa el tractament en cada cas.

4.3.1.1 Sensor de presència PIR

Quan es detecta un paquet d'una freqüència 433Mhz, es compara l'adreça detectada amb la llista de dispositius a controlar, s'ha de tenir en compte que poden haver-hi altres sensors a les llars adjacents o inclòs d'una alarma instal·lada a la mateixa llar. Si es detecta que hi és a la llista, es compara amb si és el primer cop que es detecta, en cas afirmatiu s'envia un paquet d'esdeveniment del tipus **IN_ROOM** en el qual s'indica l'adreça del dispositiu i l'hora de detecció. Posteriorment s'emmagatzema el nombre de dispositiu que s'ha detectat i no es tornarà a enviar un paquet fins que canviï el dispositiu o es detecti que l'usuari no hi és a casa.

4.3.1.2 Sensor de presència permanent

Quan es detecta un paquet Bluetooth, es compara l'adreça detectada amb la llista de dispositius, si coincideix, es revisa si és del tipus moviment o presència. En aquest segon cas, el primer cop s'envia un paquet d'esdeveniment **AT_HOME** en el qual s'indica l'adreça del dispositiu i l'hora de detecció. A partir d'aquest moment, cada cop que es detecta el dispositiu s'emmagatzema l'hora de la detecció. Aquesta hora es compara permanentment amb l'hora actual, si la diferència entre l'últim cop que s'ha detectat i el moment actual és més gran de 30 segons s'enviarà un esdeveniment del tipus **LEAVE_HOME** indicant l'adreça del dispositiu i l'hora de detecció. A més es reiniciarà la variable de detecció per deixar el sistema preparat per tornar a detectar el dispositiu.

4.3.1.3 Sensor de moviment

Si el dispositiu detectat és del tipus moviment, es compara l'hora de l'últim paquet amb l'hora actual, si és menor de seixanta segons, el paquet s'ignora, en cas contrari s'enviarà un paquet d'esdeveniment **USED_OBJECT** en el qual s'indica l'adreça del dispositiu i l'hora actual.

Cada hora, el hub connectarà amb el servidor per rebre canvis en la configuració i sincronitzar l'hora.

Un exemple d'un paquet d'esdeveniment seria:

```
{
  "type","AT_HOME",
  "address" : "AA:AA:AA:AA",
  "time":"2019-10-02T12:10:00Z"
}
```

Il·lustració 9 - Exemple paquet esdeveniment

4.4 Aplicació web

Aquest apart descriu l'aspecte visual (wireframes) de les aplicacions d'escriptori i el seu funcionament previst.

4.4.1 Accés a l'aplicació

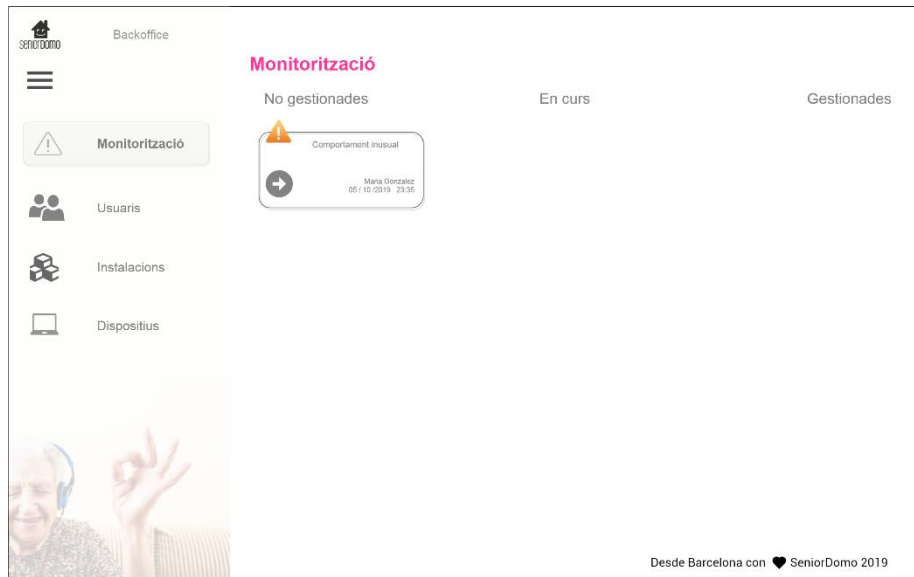
Dóna accés a l'aplicació, inclou un camp per indicar l'usuari i la paraula de pas, els camps s'identifiquen amb text i una icona que representa el significat de cada camp.

El wireframe mostra una interfície d'usuari per a l'entrada a l'aplicació. A la part superior hi ha un botó de 'Login' de color rosa. A sota hi ha dos camps de text: 'Username' amb una icona de persona i 'Paraula de pas' amb una icona de clau. A la part inferior dreta hi ha un botó de 'Aceptar' de color rosa.

Il·lustració 10 - Pantalla entrada a l'aplicació

4.4.2 Pantalla principal

La pantalla principal s'organitza en dos apartats.



II·lustració 11 - Pantalla principal

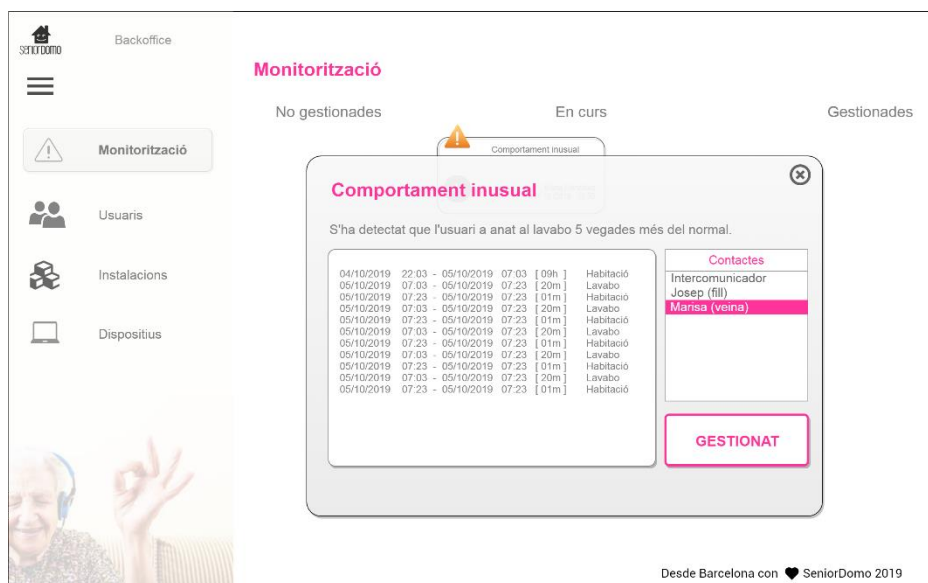
Menú. Dóna accés a les diferents opcions que es poden gestionar en el programa. Cadascuna de les opcions s'identifica per un text i una icona representativa.

L'opció activa queda emmarcada per un rectangle semitransparent.

Àrea de treball. En aquesta àrea cada opció representa la informació depenent del tipus de pantalla. Si és un manteniment, permetrà gestionar altes, modificacions i esborrat de les dades. Si és la pantalla de gestió d'alertes, permetrà la gestió de cadascuna de les alertes.

4.4.3 Monitoratge

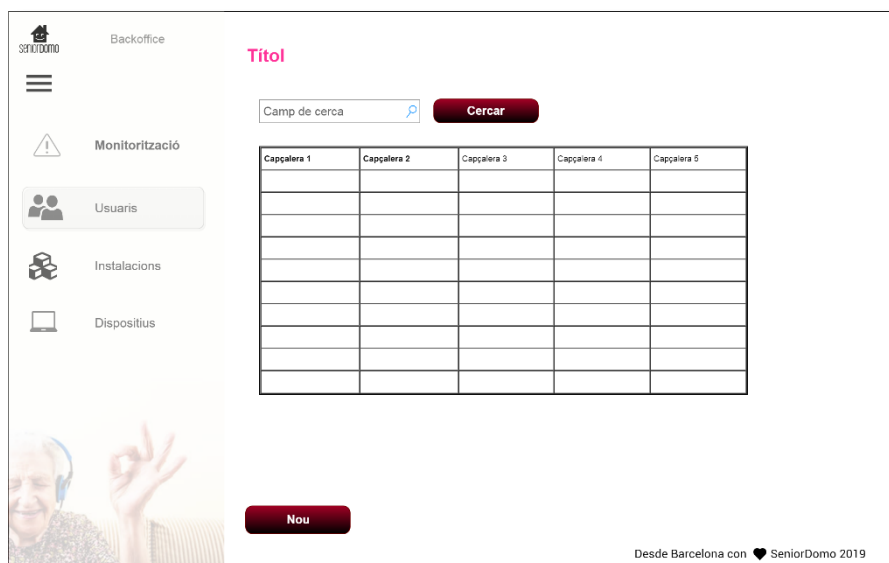
La pantalla permet controlar i gestionar les alertes detectades i els diferents esdeveniments enviats pels sensors.



II·lustració 12 - Pantalla de Monitoratge

4.4.4 Manteniments

Hi ha diverses pantalles que són manteniments, per la qual cosa totes elles tindran una estructura similar que permetrà a l'usuari familiaritzar-se amb l'ús d'aquestes.



II-lustració 13 - Manteniment

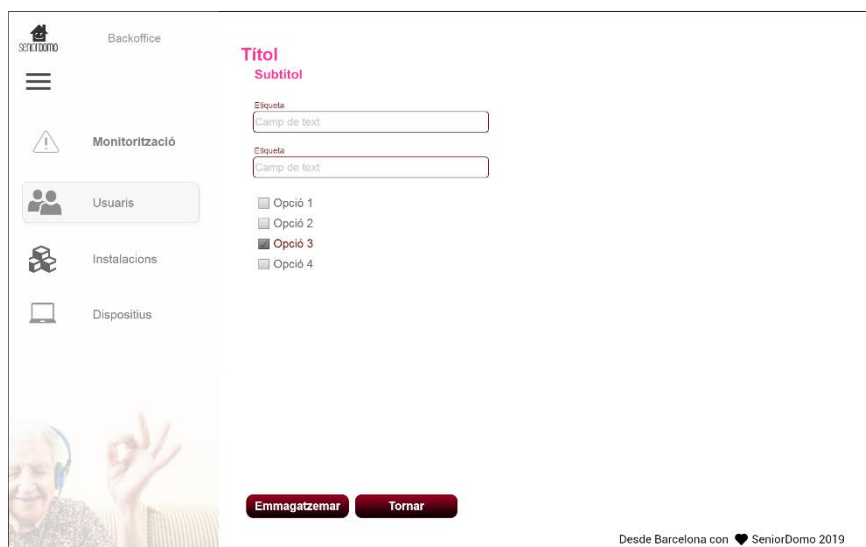
Cercar amb informació de les seves dades. Per exemple nom, mac-address, etc.

Detall de dades. Es representen les dades més importants de cada entitat. Cada fila té l'opció d'editar o donar de baixa el registre.

Al peu de la pàgina es visualitzen diferents opcions com crear un nou element, i/o altres opcions com imprimir un llistat, exportar el llistat a excel, etc.

4.4.4.1 Crear / Modificar un registre

Les pantalles d'edició i creació d'un registre són idèntiques, i permeten accedir a cadascun dels camps, depenent del tipus de camp es representarà per una entrada de text, un selector de llista o de botons.



II-lustració 14 - Crear / Modificar un registre

Al peu de pàgina, es mostren les opcions d'emmagatzemar el registre o tornar sense emmagatzemar els canvis.

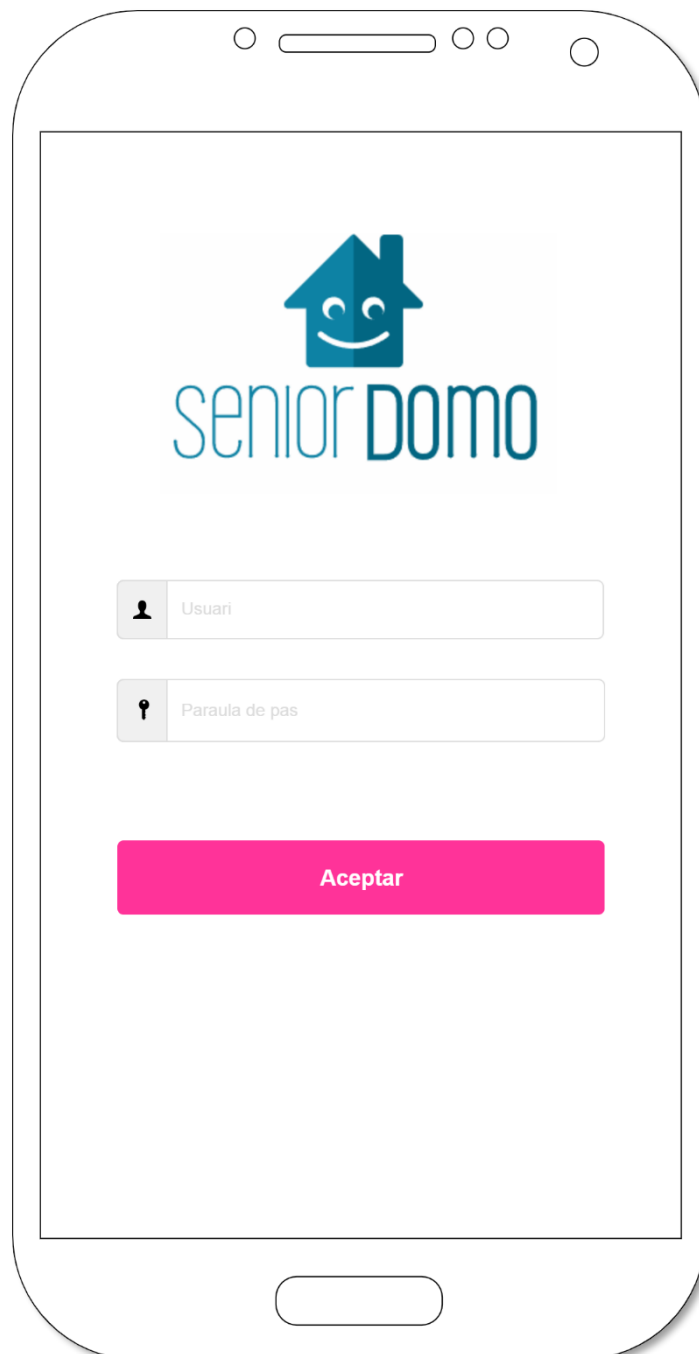
Emmagatzemar

Tornar

4.5 Aplicació Android

4.5.1 Accés a l'aplicació

Dóna accés a l'aplicació, inclou un camp per indicar l'usuari i la paraula de pas, els camps s'identifiquen amb text i una icona que representa el significat de cada camp.

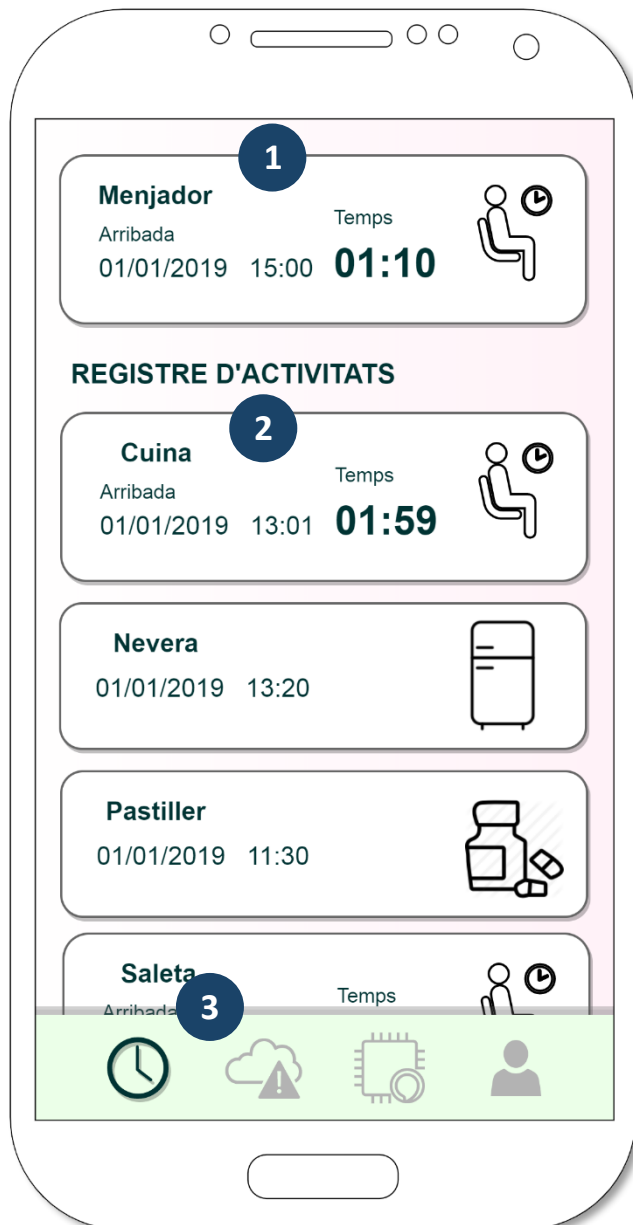


Il·lustració 15 - Pantalla d'entrada a l'aplicació

4.5.2 Pantalla principal

La pantalla principal mostra les activitats realitzades pel sènior durant les últimes 48h. Cada activitat queda representada per un requadre amb la informació i una icona representativa.

Hi ha 3 zones diferenciades a la pantalla:

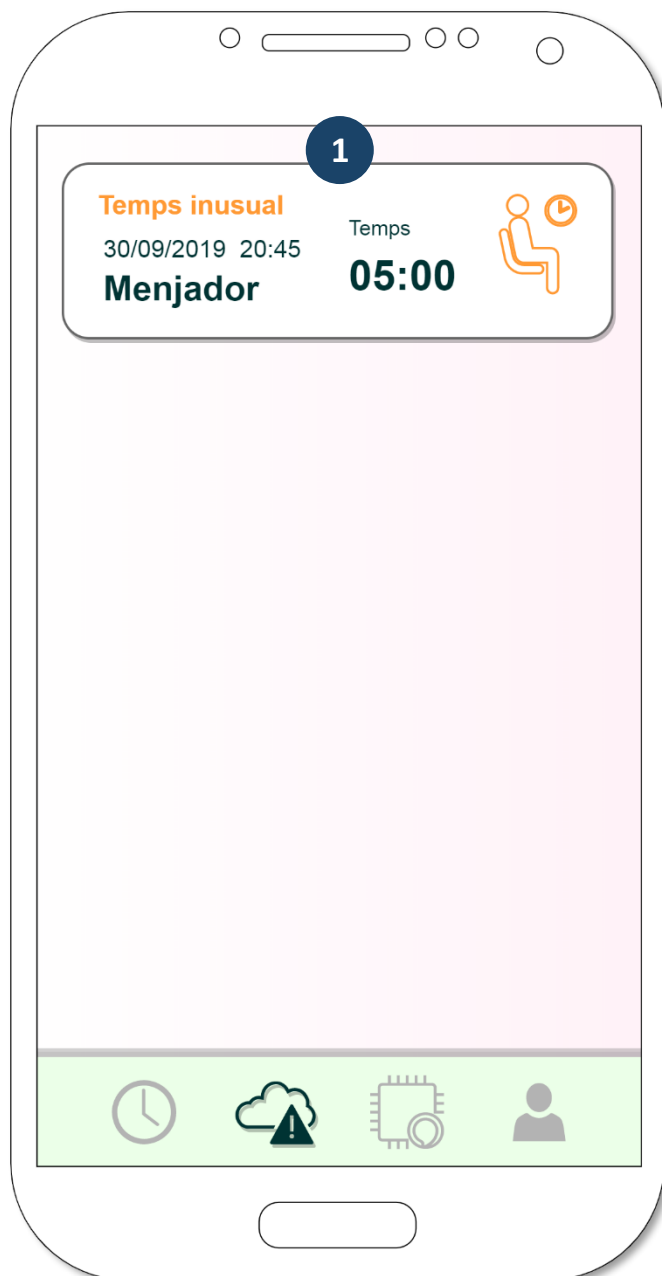


- 1 Última ubicació o última alerta.
- 2 Registre d'activitats. La llista de tots els moviments detectats pels sensors de moviment i presència.
- 3 Accés a altres opcions. Permet accedir a la resta d'opcions que dona accés a les alertes, estat dels sensors i perfil d'usuari.

Il·lustració 16 - Pantalla principal App

4.5.3 Alertes

La pantalla mostra les alertes generades en detectar diferències entre el patró del sènior i les activitats actuals.

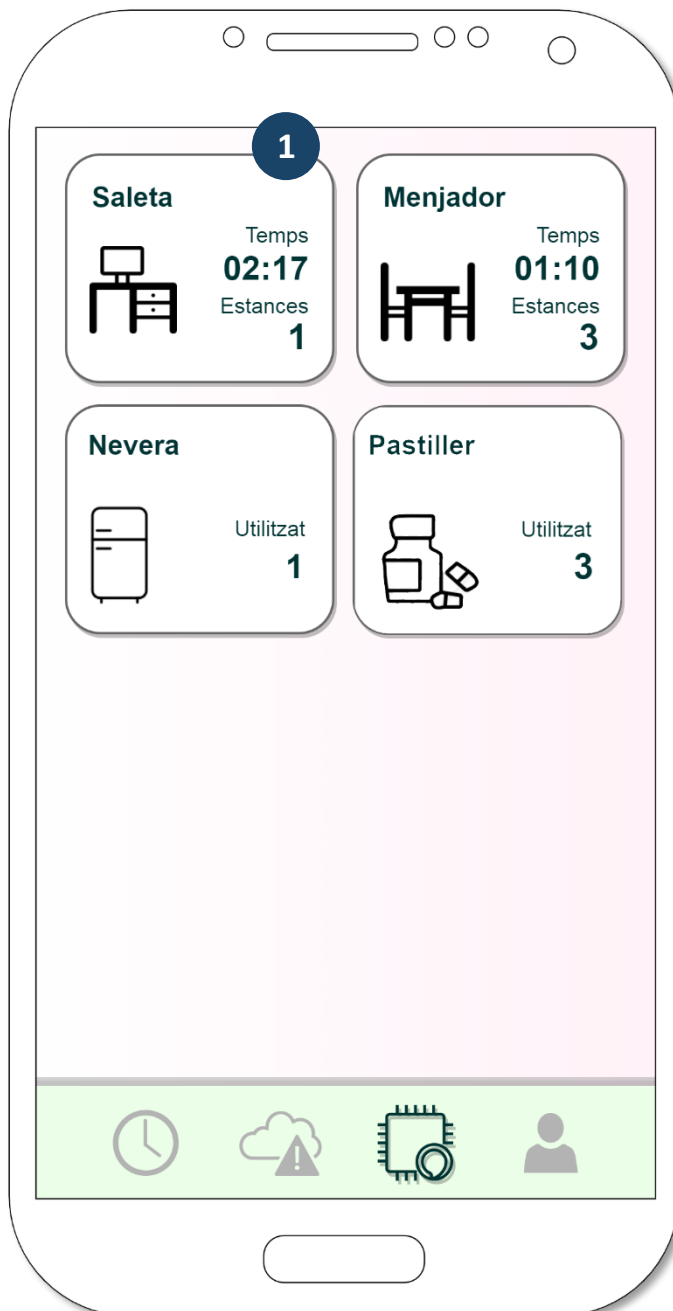
**1**

Cada alerta queda representada per un rectangle amb la informació més rellevant. La icona o el color indicant el tipus d'alerta i la rellevància d'aquesta.

Il·lustració 17 - Pantalla alertes App

4.5.4 Sensors

Aquest apartat permet veure l'estat de cada sensor, indicant el temps acumulat en els sensors de presència, i el nombre de cops que s'ha utilitzat en el dia.



Il·lustració 18 - Pantalla sensors App

1

Cada sensor queda representat per un requadre en el qual es mostra el temps acumulat que ha estat a l'habitació i el nombre de vegades.

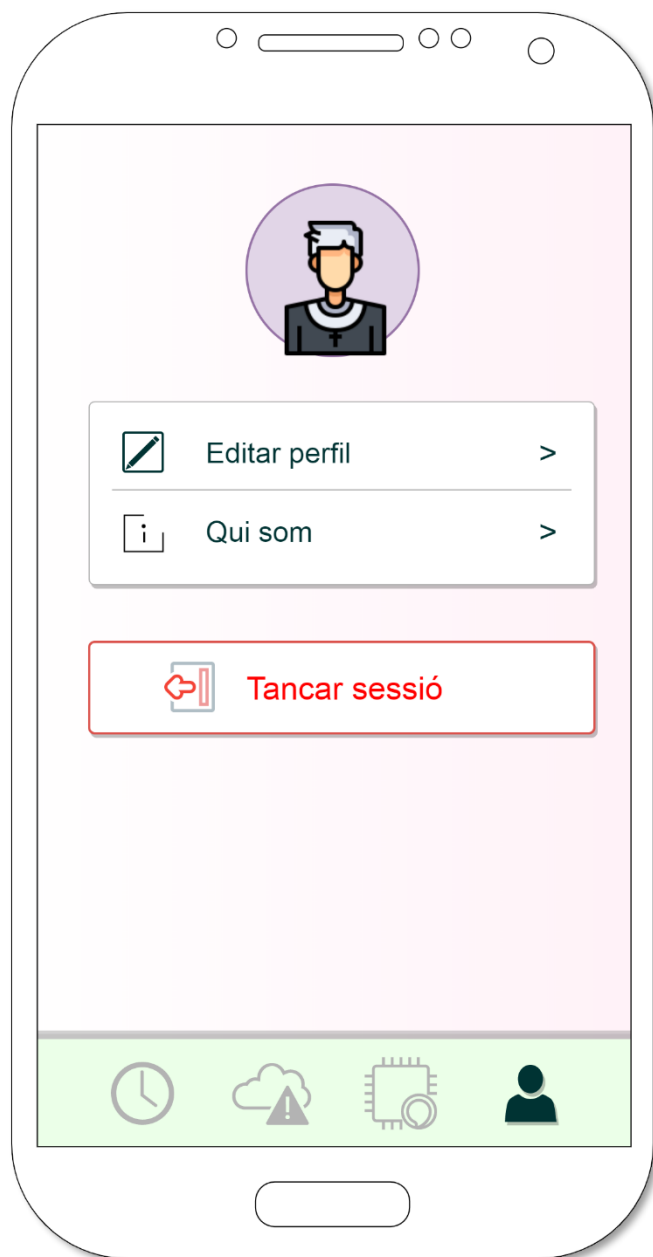
En el cas d'un objecte, es mostra el nombre de cops que ha estat utilitzat.

Una icona identifica visualment l'objecte o l'estança relacionada.

Quan es pitja sobre el requadre, s'obre una pantalla que mostra el detall de les entrades o sortides amb els horaris.

4.5.5 Perfil d'usuari

Aquest apartat permet a l'usuari editar el seu perfil per canviar el nom o la paraula de pas. També permet obtenir informació de l'empresa i tancar la sessió.



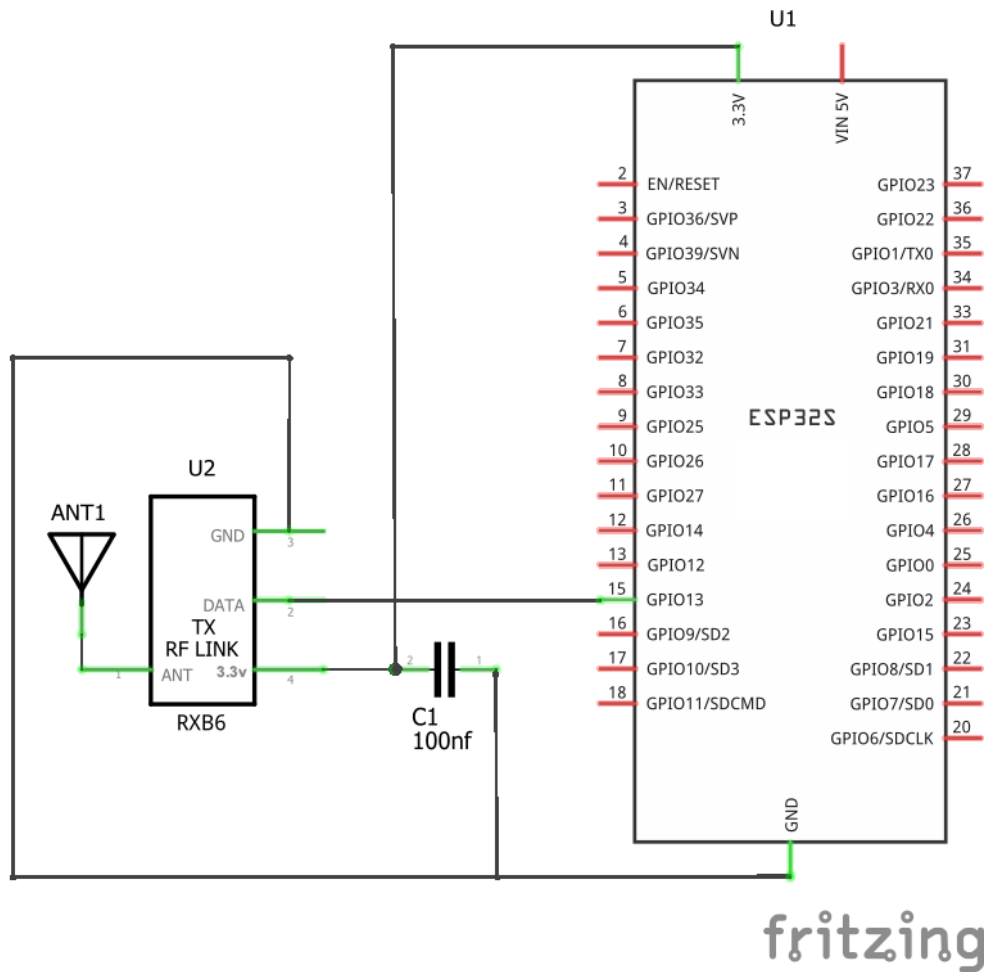
Il·lustració 19 - Pantalla perfil App

5 Prototip HUB

Aquest apartat descriu quines són les connexions entre els diferents elements que configuren el HUB. I el funcionament de les tecnologies Bluetooth i RF així com la seva implementació.

5.1 Esquema

L'esquema s'ha desenvolupat amb l'eina **Fritzing** [14] que permet d'una forma molt visual realitzar les connexions entre els diferents components.

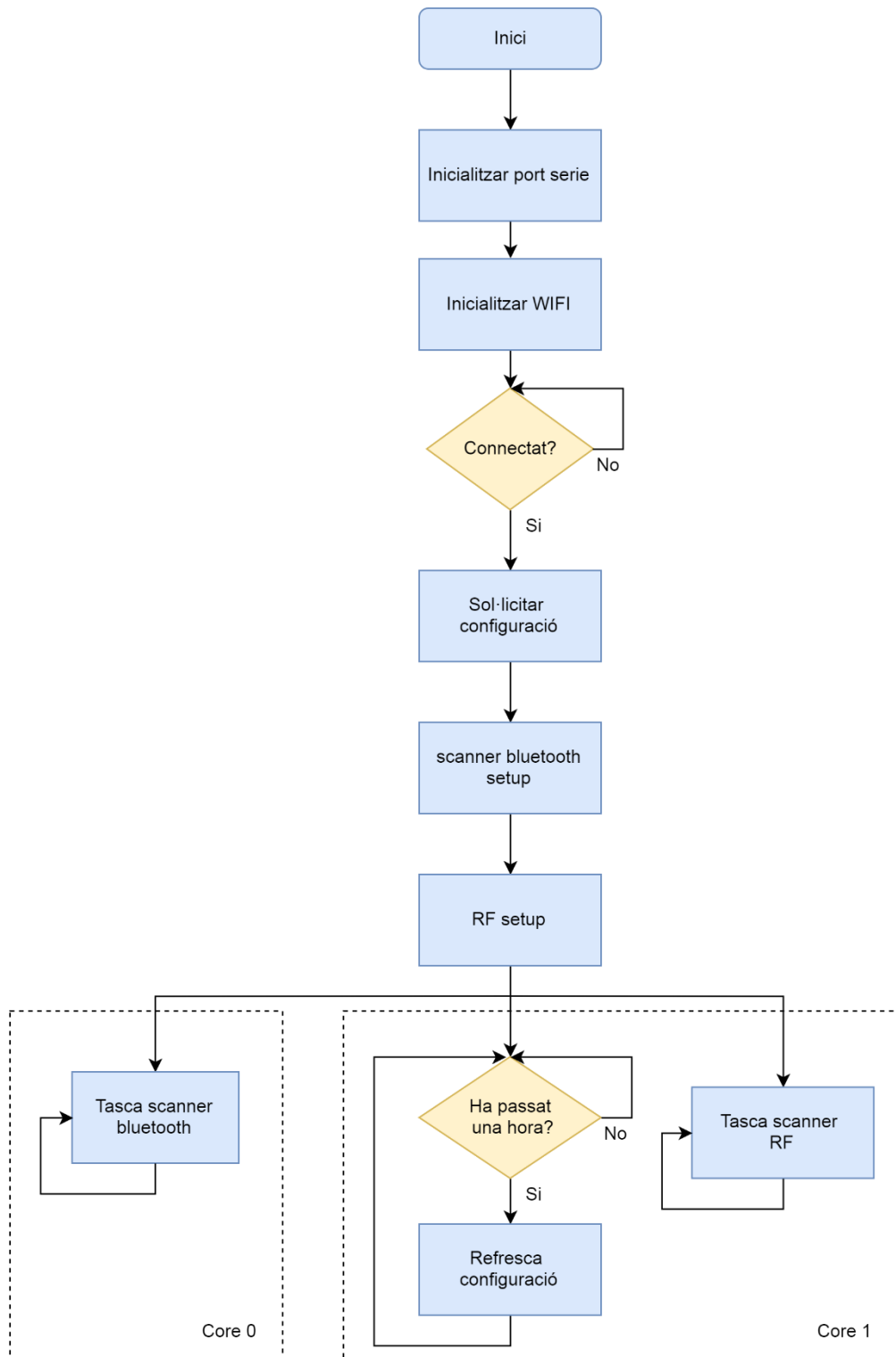


Il·lustració 20 - Esquema placa prototip HUB

El circuit és bastant senzill, ja que tots dos components ja incorporen la majoria de components discrets, per la qual cosa només cal interconnectar els pins. S'ha afegit un condensador de 100nf que millora la resposta del component RXB6.

5.2 Programa Arduino

El diagrama de flux del programa és el següent:



Il·lustració 21 - Digrama general programa Arduino

Una de les funcionalitats més interessants del microcontrolador ESP32 és que disposa de dos nuclis XTensa de 32bits, la qual cosa permet que simultàniament es poden executar dues tasques diferents, és a dir es poden iniciar dues funcions “loop” una a cada nucli.

Per associar una funció a un nucli, es fa servir la funció

```
xTaskCreatePinnedToCore(handler, name, stack_size, parameters, priority, createdTask, core)
```

Paràmetre	Descripció
handler	Punter a la funció que executa el nucli, la qual ha de ser un bucle infinit
Name	Nom de la tasca
Stack_size	Mida de la pila, indica el nombre de variables que pot gestionar
Parameters	Punter que serà passat a la tasca
Priority	Prioritat de la tasca dintre del nucli
createdTask	Punter per retornar una referència a la tasca creada
nucli	Nucli en el qual es vol associar la tasca

En el projecte es crearà una tasca per la detecció de dispositius Bluetooth i un altre per les RF, les instruccions per inicialitzar cada tasca són:

```
xTaskCreatePinnedToCore(beacon_handler, "BeaconHandler", 10000, NULL, 1, NULL, 0);
```

```
xTaskCreatePinnedToCore(rf_handler, "RfHandler", 10000, NULL, 1, NULL, 1);
```

Per tant s'executaran simultàniament les funcions *beacon_handler* i *rf_handler*, juntament amb la funció *loop* principal, que s'executa sempre al nucli 1. Cadascuna de les tasques són pràcticament independents, però comparteixen els recursos de comunicacions, ja que la comunicació wifi és comuna. Per evitar problemes de concurrència en aquest punt, es controla si ja s'està enviant informació a través de la variable *http_busy*, si és el cas, es retorna false indicant que està ocupat, per la qual cosa a cada funció que necessita enviar dades pel client http, ha d'implementar un sistema de control d'aquest valor retornat. En el nostre cas s'implementa un bucle fins que la funció retorni true.

```
while (!send_event(event)) delay(10);
```

5.3 Tecnologia BLE

La tecnologia BLE (Bluetooth low energy) està basada a l'estàndard Bluetooth 4.0 [23], funciona en el rang de freqüències de 2.4GHz i va ser creada principalment com a mitja de màrqueting permet realitzar anuncis a distàncies de fins a 100m.

Com a sistema de màrqueting, un dispositiu de baix cost *BEACON*[5], emet un missatge amb informació de l'establiment (ofertes, pàgina web, etc.). Aquesta informació es rebuda pels dispositius mòbil visualitzant la informació.

La tecnologia ràpidament va derivar en altres aplicacions com el posicionament en interiors o la detecció de presència. La incorporació de sensors dintre dels *beacons* permet per exemple tenir estacions que indiquen la temperatura ambient. Amb un acceleròmetre poden saber si el *beacon* està en moviment i emetre un senyal d'alerta.

5.3.1 Advertising payload

Qualsevol dispositiu Bluetooth, envia un paquet de 31 bytes per identificar-se amb la resta de dispositius[25][27], dintre d'aquest paquet s'identifica un número únic i els serveis que ofereix el dispositiu.

L'estructura d'aquests paquets es la següent:

- Byte 1, longitud del paquet.
- Byte 2, tipus AD, indica el tipus de dada que es transmet.
- Byte 3..N dades pròpies de cada tipus de paquet.

Alguns tipus AD són

- 0x01 – Flags.
- 0x03 – Servei UUID (16 bit service).
- 0x09 – Nom complet.
- 0x08 – Nom abreuiat.

Per exemple un beacon de termòmetre emetria un paquet amb aquest format:

```
020106030309181409546865726d6f6d65746572204578616d706c65
```

Aquesta trama es divideix en 3 paquets:

020106 - La primera trama d'una longitud 2 indica un paquet del tipus Flags

03030918 – Aquest paquet indica el tipus servei termòmetre 0x1809 (Health Thermometer)

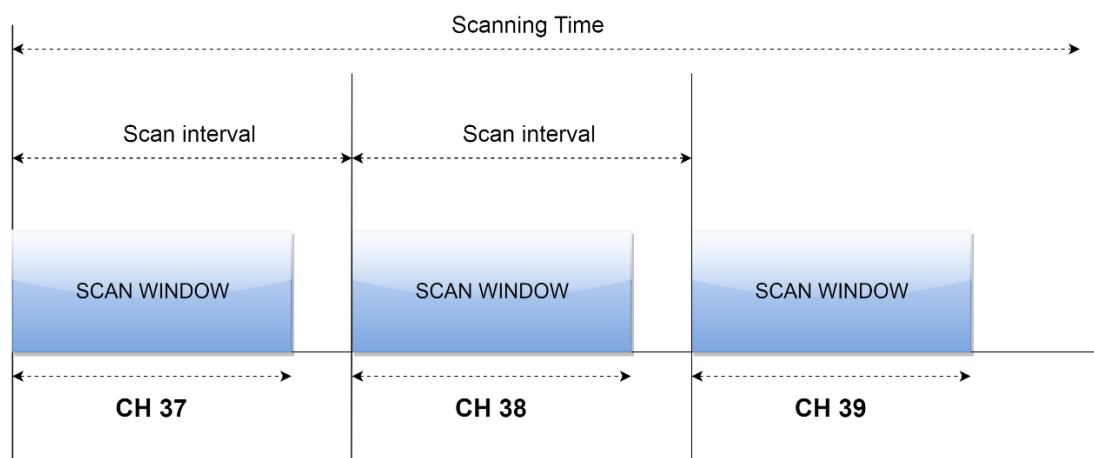
1409546865726d6f6d65746572204578616d706c65 – Aquest paquet indicaria el nom complet del dispositiu

La capacitat dels dispositius BLE per emetre un senyal continu únic o intermitent quan es produeix un moviment, permet adaptar aquesta tecnologia a la detecció de presència, per la qual cosa si una persona sempre porta a sobre un beacon, el sistema és capaç de detectar la seva presència.

Altra aplicació és detectar quan es mou un objecte, en aquest cas els beacons amb acceleròmetre permeten rebre durant uns segons un paquet indicant que l'objecte està en moviment.

5.3.2 Advertising interval

Per assegurar una millor recepció dels paquets els dispositius transmeten les dades a 3 canals diferents 37, 38 i 39 que corresponen a les freqüències 2402 MHz, 2426 MHz i 2480 MHz respectivament. A més a més, per estalviar energia, la duració de la transmissió i la freqüència varia. A més freqüència més consum d'energia i per tant menys duració de la bateria, per la qual cosa és necessari ajustar adequadament aquests temporitzadors per tal de tenir un equilibri correcte per cada aplicació. Per exemple, si l'aplicació és posicionar a través de la triangulació del senyal, és important que es configuri amb una gran potència i una alta freqüència de forma que es perdin el mínim de paquets possibles per part dels receptors.



Il·lustració 22 - Representació de transmissió BLE

En el nostre cas només es vol detectar presència, per la qual cosa es pot reduir la freqüència per estalviar bateria. La potència en la qual es transmet. Una part important a configurar és l'scan interval. Ha de tenir una duració suficient per captar els paquets però no massa llarga per evitar perdre paquets. Després de llegir diversos articles [28][29], en els quals s'indica que la finestra ha de ser menor que l'interval d'escaneig i a més no es recomanen llargs períodes d'escaneig, s'han determinat aquests valors:

Temps d'escaneig : 2s Scan interval : 100ms Scan window: 99ms

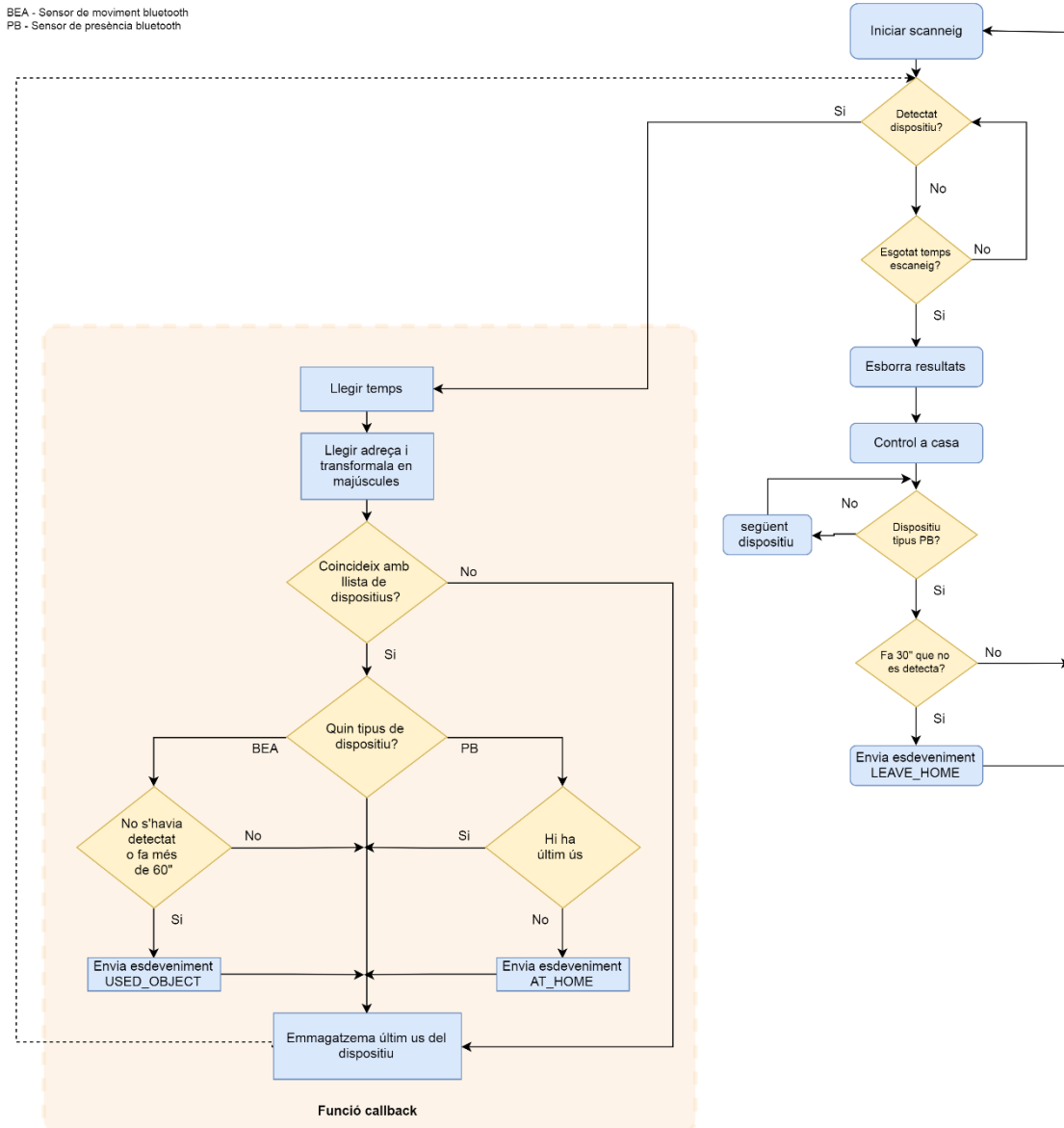
Durant 2 segons el sistema restarà a l'espera de detectar paquets, en els quals cada 100ms hi haurà un canvi de canal. Significa doncs que es detectaran durant 2 segons tots els paquets enviats pels *beacons* i un cop rebuts es processaran per tornar a quedar a l'escolta. Si el temps d'escaneig es massa curt i hi ha molts dispositius Bluetooth que transmeten dades, es pot donar el cas de rebre només els que tenen una potència i freqüència superior, per la qual cosa s'ha estimat que ampliant aquest temps hi ha menys possibilitat de perdre paquets.

5.4 Detecció de dispositius Bluetooth

Com ja s'ha esmentat anteriorment, el microcontrolador ESP32 incorpora internament un dispositiu Bluetooth compatible amb les tecnologies BLE4.2.

El fabricant subministra una llibreria que facilita la tasca de configurar el controlador Bluetooth, per poder accedir a les seves funcions s'han d'incloure les següents llibreries i configurar-les.

El diagrama de flux de la tasca de detecció de dispositius Bluetooth és el següent:



II-lustració 23 - Diagrama de flux detecció dispositius BT

Per iniciar el component Bluetooth, s'ha de fer una crida a la funció init. Un cop iniciat, es crea un objecte escàner (BLEScan) que serà l'encarregat de cercar els sensors beacons. Com es pot apreciar es configura l'interval i la finestra. Un cop configurat, s'ha de fer una crida a la funció start indicant quant de temps volem escanejar. Aquesta crida es realitza a la funció loop.


```
#include <BLEDevice.h>
#include <BLEUtils.h>
#include <BLEScan.h>

void beacon_setup() {
  BLEDevice::init("");
  pBLEScan = BLEDevice::getScan(); //create new scan
  // Informs the callback object
  pBLEScan->setAdvertisedDeviceCallbacks(new MyAdvertisedDeviceCallbacks());
  pBLEScan->setActiveScan(true); //active scan uses more power, but get results faster
  pBLEScan->setInterval(100); // Scanning interval
  pBLEScan->setWindow(99); // less or equal setInterval value
  // Create the task in core 0
  xTaskCreatePinnedToCore(beacon_handler, "BeaconHandler", 10000, NULL, 1, NULL, 0);
}
```

Codi font 1 - Inicialització escàner Bluetooth

Cada cop que es detecta un paquet Bluetooth, es farà una crida a la funció callback

```
class MyAdvertisedDeviceCallbacks: public BLEAdvertisedDeviceCallbacks {
  void onResult(BLEAdvertisedDevice advertisedDevice) {
    // Get actual time in seconds
    long time = millis() / 1000;
    // Get device address detected
    string address = advertisedDevice.getAddress().toString();
    // Uppercase address
    transform(address.begin(), address.end(), address.begin(), ::toupper);
    // Looking for device in the list
    for (int i = 0; i < num_devices; i++) {
      // The item in list is the same?
      if (device[i].address.compare(address) == 0) {
        Event event;
        event.device_id = device[i].id;
        event.date = getTime();
        // Checks the type of Bluetooth device
        switch (device[i].type) {
          case BEA: if (device[i].last_use == -1 || time - device[i].last_use > 60) {
            // If it's a movement device checks if there isn't an event in the last 60s
            // If there isn't a recent event sends to server the event
            event.type = USED_OBJECT;
            while (!send_event(event)) delay(10);
            Serial.printf("Detected movement beacon: %s \n", address.c_str());
          } break;
          case PB: if (device[i].last_use == -1) {
            // If it's a presence device checks if the same device was detected
            // If not sends the event to the server
            event.type = AT_HOME;
            while (!send_event(event)) delay(10);
            Serial.printf("AT HOME: %s \n", address.c_str());
          } break;
        }
        // Refresh last use on device
        device[i].last_use = time;
      }
    }
  }
};
```

Codi font 2 - Funció Callback per gestionar un dispositiu detectat

A cada crida de la funció es compara l'adreça detectada amb la llista de sensors que ha de controlar el HUB, es realitza un bucle que recorre tota la llista i amb la funció compare mira de quin dispositiu es tracta.

Un cop detectat, analitza el tipus de dispositiu de què es tracta, si és un beacon de moviment, analitza si hi ha hagut un esdeveniment en els últims 60 segons, si és el cas, l'esdeveniment es descarta. La funció del beacon de moviment és detectar l'ús d'un objecte i per tant només interessa detectar el primer cop que es fa servir i no cada cop que està en moviment per tant al

primer moviment es detecta i no tornarà a ser detectat fins que el sensor torni a estar en repòs durant 60 segons.

En el cas d'un beacon de presència busca si no s'havia detectat prèviament la seva presència, si és el cas, enviarà un esdeveniment AT_HOME, en cas contrari no farà res fins que deixi de detectar-ne.

Finalment en tots els casos, s'actualitza l'hora de l'última detecció.

Al bucle principal, es reactiva l'escàner i es fa un control del dispositiu de presència. Si fa més de 30 segons que no s'ha detectat, s'envia un esdeveniment LEAVE_HOME i s'inicialitza la variable de temps per indicar que no hi és present.

```
void beacon_handler( void * pvParameters ) {
  Event event;
  for(;;) {
    // Start new scanning
    BLEScanResults foundDevices = pBLEScan->start(scanTime, false);
    // Delete results fromBLEScan buffer to release memory
    pBLEScan->clearResults();
    // Get actual time
    long time = millis() / 1000;
    // Check if the presence beacon is still presence
    for (int i = 0; i < num_devices; i++) {
      // If the device is PresenceBeacon and was detected and the last time was older than
      // 30 s means the beacon is not presence
      if (device[i].type == PB && device[i].last_use > -1 && time - device[i].last_use > 30){
        // Set in device is not detected
        device[i].last_use = -1;
        last_presence_device = -1;
        // Sends the event to the server
        Serial.printf("LEAVE HOME: %s \n", device[i].address.c_str());
        event.device_id = device[i].id;
        event.date = getTime();
        event.type = LEAVE_HOME;
        while (!send_event(event)) delay(10);
      }
    }
  }
}
```

Codi font 3 - Funció control dispositius Bluetooth

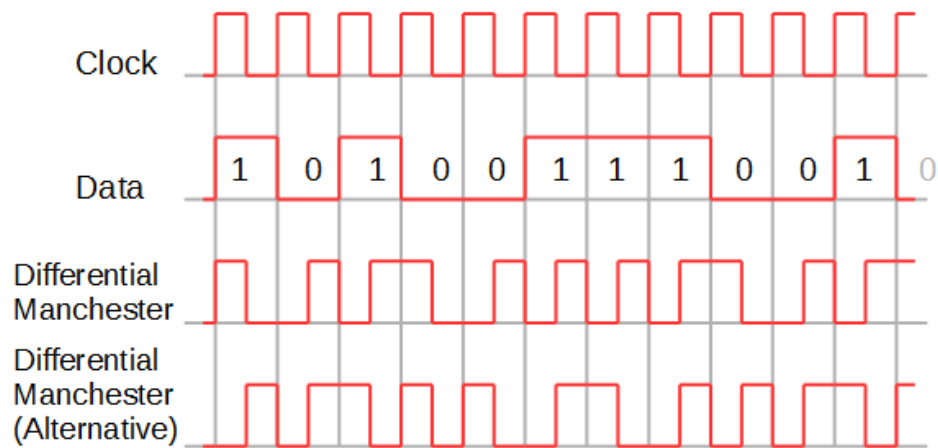
La funció beacon_setup serà cridada dintre de la funció setup d'Arduino. Normalment a Arduino, la funció principal loop és l'encarregada de gestionar repetidament tots els processos, però en el nostre cas es fa servir la capacitat del microcontrolador ESP32 de tenir dues tasques simultànies, una a cada nucli.

5.5 Tecnologia RF

Els dispositius estàndard que treballen a freqüències de 433Mhz [7], utilitzen normalment el protocol Manchester [32], aquest tipus de protocol és de polarització neutral en el qual cada bit està compost sempre per un senyal alt i baix de la mateixa longitud.

Les propietats més rellevants són:

- El senyal no té component contínua.
- El senyal porta implícitament un rellotge de sincronisme.
- El senyal ocupa el doble d'amplada de banda.



Il·lustració 24 - Protocol Manchester (font Viquipèdia)

La transmissió és de tipus sèrie, per la qual cosa s'han de llegir tots els bits i agrupar-los en grups de 8 bits per formar un byte. Per llegir els diferents bits, és necessari configurar un rellotge a una freqüència suficient per captar cada canvi d'estat. El més adient és associar una interrupció del microcontrolador al pin on connectarem el receptor, de forma que per cada canvi d'estat el microcontrolador rebrà una interrupció.

Els sensors de presència envien per norma paquets de 24bits cada cop que detecten presència, l'adreça del sensor normalment és de 20 bits deixant 4 bits addicionals per indicar diferents dades pròpies del sensor. Per exemple hi ha models que indiquen l'estat de la bateria, etc.

5.6 Detecció de dispositius RF

El programador Suat Özgür (sui77) va crear la llibreria RCSwitch[30] [31] per gestionar la recepció i transmissió en dispositius RF, la llibreria gestiona la lectura de les dades en format sèrie. Cada cop que hi ha una dada, es produeix una interrupció que s'agrupa formant paquets i es tracten segons diversos protocols existents en els emissors RF. Per incloure la llibreria en el projecte s'ha d'afegir a la capçalera i crear un objecte RCSwitch.

```
#include <RCSwitch.h>
// Create RC object to control RF devices
RCSwitch mySwitch = RCSwitch();
```

Codi font 4 - Llibreria control dispositius RF

Després de crear l'objecte, cal configurar-lo per indicar a quin pin del microcontrolador està connectat el receptor RF.

```
void rf_setup() {
  // Reads from RXB6 in pin 13
  mySwitch.enableReceive(13);
  // Create handler task
  xTaskCreatePinnedToCore(rf_handler, "RfHandler", 10000, NULL, 1, NULL, 1);
}
```

Codi font 5 - Inicialització llibreria RCSwitch

El receptor RXB6 està connectat al pin 13 del microcontrolador ESP32, per la qual cosa, indiquem a la llibreria amb l'ordre **enableReceive** quin és el pin associat. És important que el pin tingui capacitat per ser associat a una interrupció perquè depenent del model d'Arduino no tots els pins d'entrada ho permeten.

```
void rf_handler( void * pvParameters ) {
    Event event;
    for (;;) {
        // Is device detector available
        if (mySwitch.available()) {
            // Check if a device was detected
            Unsigned long value = mySwitch.getReceivedValue();
            // Gets actual time
            long time = millis() / 1000;
            // If value <> 0 a device was detected
            if (value != 0) {
                // Get device address
                std::stringstream ss; ss << value;
                string address = ss.str();
                if (num_devices == 0) Serial.printf("Device: %s \n", address.c_str());
                for (int i = 0; i < num_devices; i++) {
                    if (device[i].type == PIR && device[i].address.compare(address) == 0 &&
                        last_presence_device != i) {
                        // If the item in list is a PIR and has the same address
                        // and wasn't already detected
                        last_presence_device = i; // Save the device
                        // Sends the event to the server
                        event.device_id = device[i].id;
                        event.date = getTime();
                        event.type = AT_ROOM;
                        while (!send_event(event)) delay(10);
                        Serial.printf("AT ROOM: %s \n", address.c_str());
                    }
                }
            }
            // Cleans result to restart the detection
            mySwitch.resetAvailable();
        }
    }
}
```

Codi font 6 - Funció rf_handler

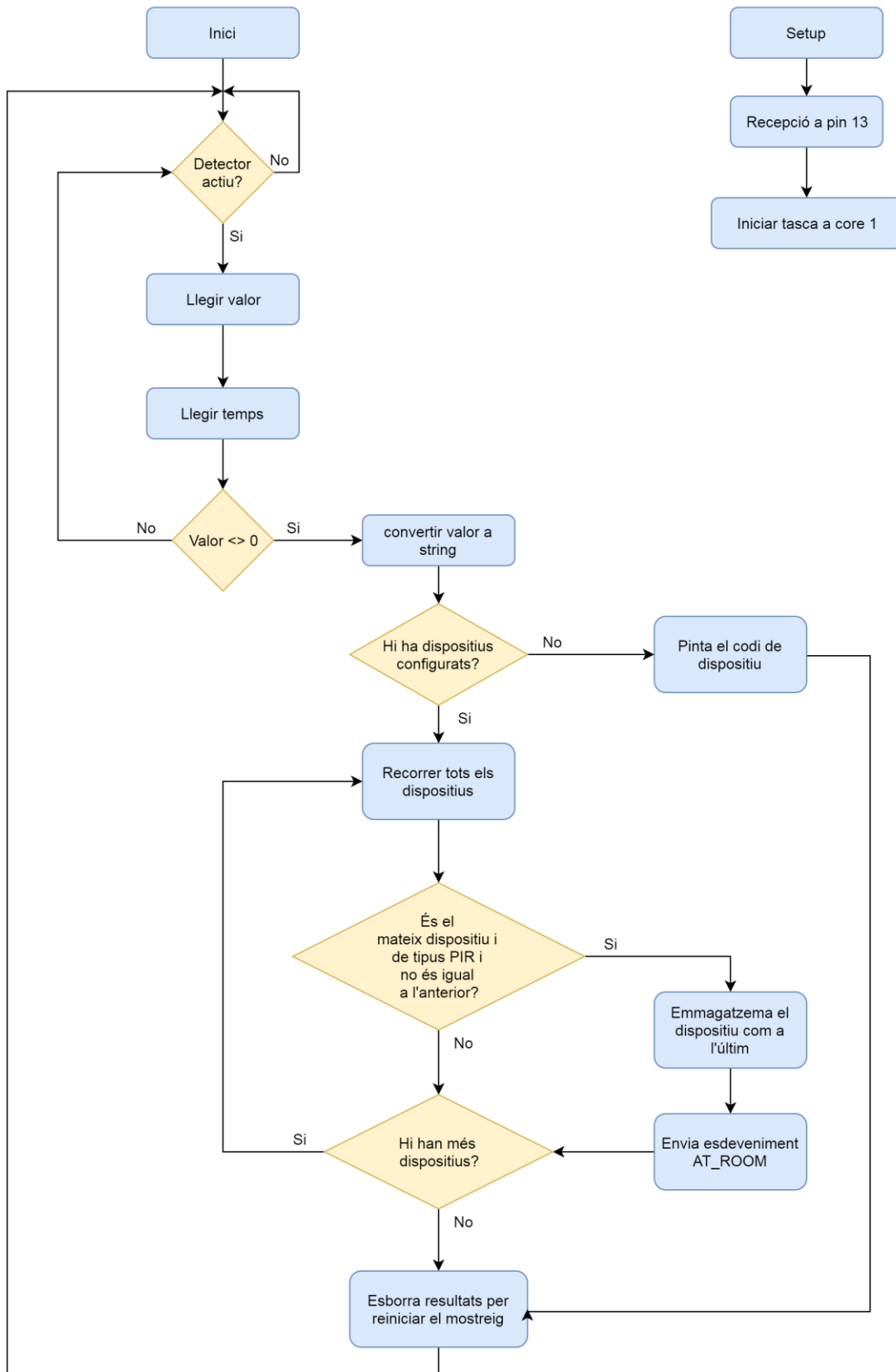
Quan la llibreria detecta un paquet emès per un dispositiu RF compatible, ho indicarà amb la funció *available*, que retornarà un valor *true*. A través de la funció *getReceivedValue*, rebrem un codi de 32 bits, aquest codi varia entre cada fabricant.

Un cop llegida l'adreça del sensor, es converteix de numèric a string per poder comparar-ho amb les adreces dels dispositius que controla el hub.

Si es detecta un sensor PIR de la mateixa adreça i l'últim detectat no és el mateix sensor, es crea un esdeveniment per informar del canvi d'habitació. A partir d'aquest moment encara que el sensor continuï detectant la presència, no es tornarà a enviar l'esdeveniment perquè no es detecta un canvi de sensor. Aquest control es fa amb la variable *last_presence_device* que emmagatzema l'últim sensor detectat.

Un cop analitzats tots els sensors, es neteja el buffer per tornar a començar.

El diagrama de flux de la detecció de sensors RF és:



II-lustració 25 - Diagrama de flux detecció RF

5.7 Rellotge en temps real RTC

Un rellotge en temps real RTC [8] permet al sistema Arduino tenir control sobre el temps en el qual succeeixen els esdeveniments. Per defecte les plaques Arduino no inclouen un component que permeti tenir un control del temps i és necessari un component extern com el DS1307 per disposar d'aquesta funcionalitat, però també hi ha implementacions per programari. En el cas del microcontrolador ESP32, si incorpora internament un component RTC que permetria mantenir l'hora si el dispositiu no té corrent, però requereix una bateria externa per mantenir-la.

En el nostre cas, s'ha decidit que l'hora la subministri directament el servidor dintre del paquet de configuració envers un servidor NTP. El motiu principal és per una banda l'estalvi de codi, és més òptim llegir l'hora en el paquet de configuració que afegir la llibreria de temps. Per altra banda, el dispositiu HUB necessita refrescar la configuració periòdicament, per la qual cosa sincronitzar els rellotges en el mateix moment que es rep la configuració, permet que els dos sistemes estiguin perfectament sincronitzats. Per evitar problemes amb el control de les zones horàries el HUB treballa en format UTC, serà el servidor l'encarregat d'ajustar l'hora a la zona horària adequada.

```
// Set time from string time to RTC
void setTime(string time) {
    int year = atoi(time.substr(0,4).c_str());
    int month = atoi(time.substr(5,2).c_str());
    int day = atoi(time.substr(8,2).c_str());
    int hour = atoi(time.substr(11,2).c_str());
    int min = atoi(time.substr(14,2).c_str());
    int sec = atoi(time.substr(17,2).c_str());

    Serial.printf("%d %d %d - %d %d %d \n", year, month, day, hour, min, sec);
    struct tm localTime = {sec,min,hour,day,month-1,year-1900,0,0,0};
    int32_t unixTime = mktime( &localTime );
    timeval epoch = {unixTime, 0};
    settimeofday((const timeval*)&epoch, 0);
}

// Get time from RTC
string getTime() {
    char time[32];
    struct tm now;
    getLocalTime(&now,0);
    strftime(time, 21, "%Y-%m-%dT%H:%M:%S", &now);
    sprintf(time, "%s.000Z", time);
    return time;
}
```

Codi font 7 - Funcions setTime i getTime

Per gestionar el rellotge, es va servir la llibreria <time.h> que permet configurar el rellotge intern i s'han programat dues funcions que permet escriure i llegir el rellotge en format UTC.

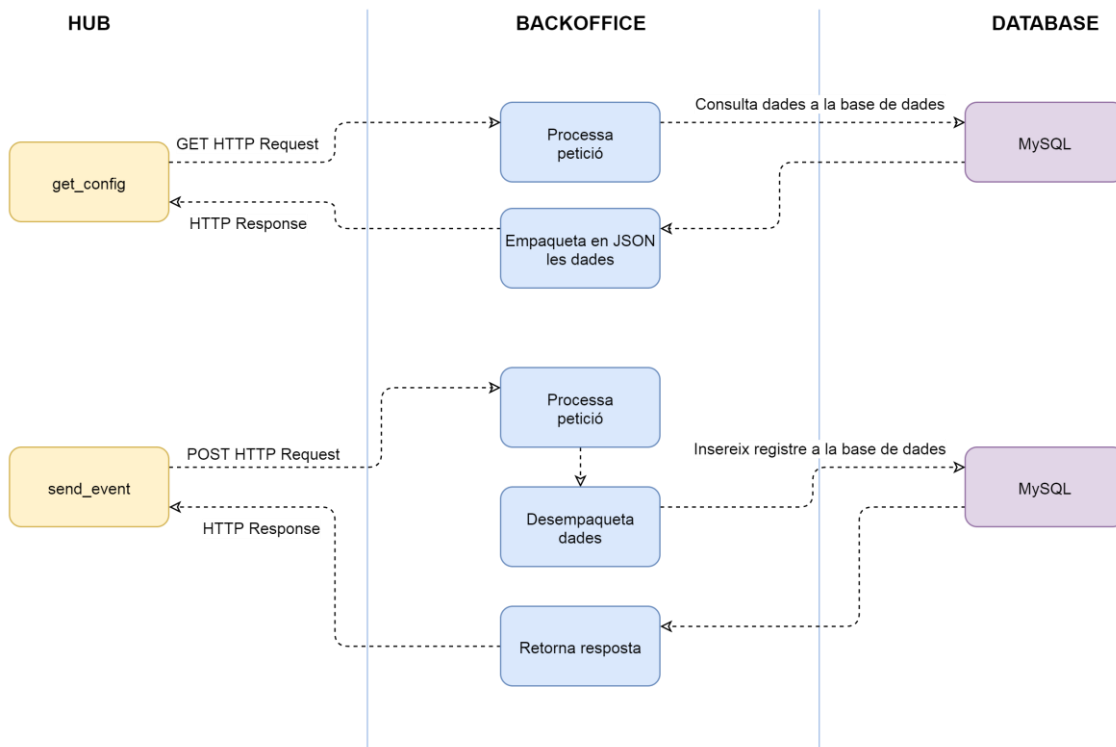
La funció setTime configura el rellotge a partir d'una cadena de text amb el següent format d'hora yyyy-MM-ddThh:mm:ss.000Z

El primer que fa la funció és trossejar la cadena de text per obtenir l'any, mes, dia, etc. i l'emmagatzema en diferents variables. A partir d'aquestes variables crea l'estructura tm necessària per obtenir l'hora en format Unix i executar la funció settimeofday que és l'encarregada de configurar el rellotge intern.

La funció getTime fa l'operació contrària, a partir de l'estructura tm creada a través de la crida a la funció getLocalTime, es converteix a una cadena de text en el format yyyy-MM-ddThh:mm:ss.000Z

5.8 Comunicacions

El HUB requereix connexió a Internet per una banda per rebre la configuració dels dispositius que ha de controlar i per altra banda per comunicar els diferents esdeveniments que es detectin. Aquest intercanvi d'informació es realitza amb el backend a través d'un servei REST [21][22], el següent esquema mostra els fluxos entre el servidor i el hub.



Il·lustració 26 - Esquema flux comunicació HUB - Backoffice

En aquest cas la connexió a Internet s'implementa a través de wifi. Es requereixen dues llibreries, l'accés al mòdul wifi i el client http.

```
#include <WiFi.h>
#include <HTTPClient.h>
```

Codi font 8 - Llibreries accés a internet

El HUB té dues funcions per comunicar amb el backoffice, per una banda la funció *get_config* que li permet llegir la configuració dels dispositius que ha de controlar i per altre banda la funció *send_event* que es fa servir cada cop que s'ha detectat un esdeveniment i es vol informar al backoffice del mateix. Les dades que s'intercanvien s'empaqueten en format JSON [17] que és el mateix format en el qual s'estructura un objecte javascript i que permet serialitzar estructures de dades entre sistemes.

5.8.1 get_config

La funció `get_config` és l'encarregada de demanar la configuració al servidor, realitza una crida a l'endpoint `/ws/devices/config/{address}` i rebrà la configuració en format JSON[17] amb la següent estructura.

```
{
  "result": "OK",
  "message": "Successful",
  "object": {
    "beaconScanningTime": 2000,
    "dateTime": "yyyy-MM-ddTth:mm:ss.000Z",
    "devices": [
      { "id": 1,
        "type": 1,
        "address": "XX:XX:XX:XX:XX:XX"
      },
      ...
      { "id": N,
        "type": X,
        "address": "XXXXXXXXXX"
      }
    ]
  }
}
```

Il·lustració 27 - Estructura resposta configuració

L'estructura de les respostes del servidor és sempre la mateixa, retorna un objecte amb 3 camps: `result`, `message` i `object`.

result. Cadena de text que indica el resultat de l'operació, si tot ha funcionat correctament retorna OK i en cas d'error ERROR.

message. Cadena de text indicant amb un text explicatiu el resultat de l'operació, en el cas d'un error es detalla exactament el tipus d'error.

object. Objecte que varia depenent de la funció REST executada, en el cas de la configuració es rep un objecte amb els següents camps:

- `beaconScanningTime`. Indica el temps que estarà actiu l'escàner Bluetooth.
- `dateTime`. Indica el temps en format UTC per sincronitzar el rellotge.
- `devices`. Array dels sensors que ha de controlar el HUB, cada sensor queda configurat amb les següents propietats:
 - `id`. Identificador únic en el sistema.
 - `type`. Tipus de sensor (0 - BEA, 1 - PB, 2 - PIR)
 - `address`. Adreça que identifica el sensor, en el cas de sensors Bluetooth té un format tipus mac-address `XX:XX:XX:XX:XX:XX` i en el cas de sensors RF es normalment un valor numèric.

L'objecte es rep com una cadena de text i és necessari transformar-lo a una estructura de dades, per aquesta tasca es fa servir la llibreria `<Arduino_JSON.h>` [20] que permet la manipulació d'objectes en format JSON però molt optimitzada en la seva mida.


```

// URL endpoint
String server = "http://192.168.1.4:8080/MyDay/ws/devices";

boolean get_config(bool reset) {
  if (http_busy) return false;
  http_busy = true;
  // Endpoint to config server
  String url = server + "/config/" + WiFi.macAddress();
  // Connect to server
  http.begin(url);
  http.setTimeout(30000);
  // Do a GET request
  int httpCode = http.GET();
  if (httpCode > 0) {
    // Get the request response data
    String json = http.getString();
    //Serial.println(json);
    // Parse JSON object returned
    JSONVar obj = JSON.parse(json.c_str());
    scanTime = (((int)obj["object"]["beaconScanningTime"])/1000);
    // Get dateTime from server and synchronize internal RTC
    string dateTime = (const char*) obj["object"]["dateTime"];
    setTime(dateTime);
    num_devices = 0;
    int num = (int) obj["object"]["devices"].length();
    Serial.printf("Num devices %d \n",num);
    if (num > 10) num = 10; // Protect array overflow
    for (int i = 0; i < num; i++) {
      device[i].address.assign((const char*)obj["object"]["devices"][i]["address"]);
      if (reset) device[i].last_use = -1;
      device[i].id = (int)obj["object"]["devices"][i]["id"];
      device[i].type = (int)obj["object"]["devices"][i]["type"];
      Serial.printf("Device %s type %d\n",device[i].address.c_str(), device[i].type);
    }
    num_devices = num;
  }
  http.end(); //Close connection
  http_busy = false;
  return true;
}

```

Codi font 9 - Funció getConfig

La funció fa una crida de tipus GET (`http.GET()`) a l'endpoint passant com a {id} la mac-address del microcontrolador ESP32, aquesta adreça és única per cada dispositiu, per la qual cosa permet identificar unívocament el dispositiu en el sistema. La crida al servidor cercarà a la base de dades tots els sensors que tenen en comú la mateixa instal·lació que el HUB.

La conversió de text a un objecte es realitza amb la crida a la funció `JSON.parse` la qual crearà una estructura d'arrays i subarrays de l'objecte. Per exemple per accedir al camp result s'accedirà amb `obj["result"]` per accedir a la data i hora `obj["object"]["dateTime"]`

Un cop rebuda la configuració, es configuren els sensors rebuts emmagatzemant l'adreça, id i el tipus de sensor.

Finalment es tanca la connexió HTTP amb la instrucció `http.end()`

5.8.2 send_event

La funció `send_event` envia un esdeveniment al servidor. Fa una crida POST en la qual el cos de la petició és un objecte JSON amb la següent estructura:

```
{
  "device_id": N,
  "type": T,
  "date": "yyyy-MM-ddThh:mm:ss.000Z"
}
```

Il·lustració 28 - Estructura paquet esdeveniment

La funció rep com a paràmetre un objecte `Event` que el transforma a una cadena de text en format JSON.

```
boolean send_event(struct Event e) {
  if (http_busy) return false;
  http_busy = true;
  char json[100];
  sprintf(json, "{\"device_id\": \"%d\", \"type\": %d, \"date\": \"%s\"}", e.device_id, e.type,
    e.date.c_str());
  Serial.println(json);
  // Endpoint to config server
  String url = server + "/event";
  // Connect to server
  http.begin(url);
  http.addHeader("Content-Type", "application/json");

  int httpCode = http.POST(json);
  if (httpCode == HTTP_CODE_OK) {
    String result = http.getString();
    Serial.println(result);
  } else {
    Serial.printf("[HTTP] POST... failed, error: %s\n", http.errorToString(httpCode).c_str());
  }
  http.end(); //Close connection
  http_busy = false;
  return true;
}
```

Codi font 10 - Funció `send_event`

Per crear la cadena JSON, es fa servir la funció `sprintf`, en aquest cas no es fa servir la llibreria `Arduino_JSON` perquè l'estructura és molt simple, per la qual cosa és més òptim crear la cadena de text amb aquest mètode.

La funció **`sprintf`** [34] permet donar format a diferents tipus de variables que s'inclouran en el text amb un caràcter `%` i el tipus de paràmetre. Per exemple per incloure en el text una variable [string], es farà servir `%s` en el cas d'una variable [int] es farà servir `%d`. Com que els noms de les variables i els valors de cadena a JSON s'indiquen amb cometes, és necessari indicar-ho dintre de la cadena de text. Per poder incloure dintre d'una cadena de text caràcters especials, s'ha d'incloure un caràcter d'escapament que indica al compilador que el proper caràcter és especial. En el cas de les dobles cometes, s'indica amb `\"`

Com el contingut de la crida POST és un objecte JSON, s'ha d'indicar a la capçalera HTTP. Per fer-ho, s'afegeix a la capçalera un paràmetre **`Content-Type`** amb valor **`application/json`**.

6 BackEnd

6.1 Protocol Oauth2

El protocol Oauth2 [13][15], és un protocol d'autenticació que permet a les aplicacions WEB i aplicacions mòbils obtenir accés limitat a recursos HTTP.

6.1.1 Rols del protocol

El sistema defineix quatre rols diferents:

Resource owner (usuari). És l'usuari que vol accedir a recursos del sistema

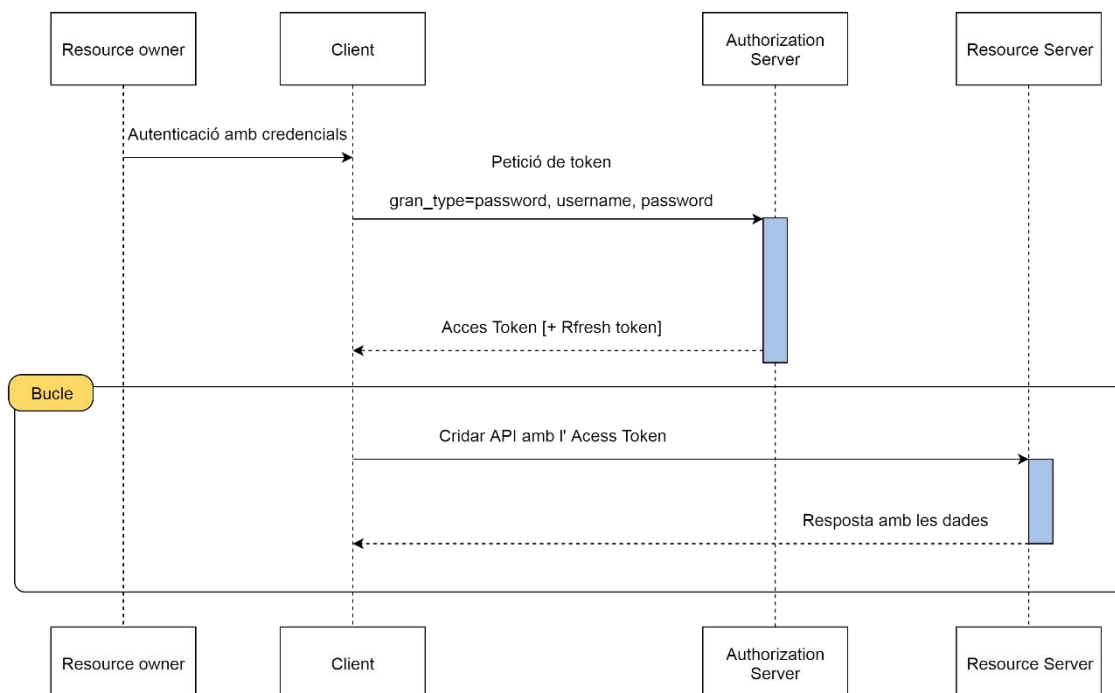
Resource server (servei). És el servidor que subministra els recursos de l'aplicació.

Client (aplicació). Aplicació que sol·licita les dades per mostrar-les a l'usuari.

Authorization server. És un servidor que autentica l'usuari i emet un "token" d'accés temporal als recursos. És habitual que el resource server i l'authorization server estiguin junt.

6.1.2 Flux de comunicació

El flux d'una comunicació basada en Oauth2 seria:



Il·lustració 29 - Dígrama flux comunicació protocol Oauth2

El client accedeix a l'aplicació que sol·licita una autorització, en aquest cas usuari i paraula de pas.

Aquestes dades són enviades al servidor d'autorització que les verifica i emet un *token* d'accés que és enviat a l'aplicació i que haurà de fer servir en totes les comunicacions amb el servidor de recursos.

El *token* té una duració limitada a una hora, que pot ser renovada a través del *refresh_token* que té una duració d'un dia. Quan es detecta que ha caducat el token, es fa una nova petició de

token però enviant com a credencial el *refresh_token*. Amb aquest sistema no és necessari que l'aplicació de client emmagatzemi les credencials de l'usuari.

Altra dels avantatges és que el sistema de *token* no depèn de l'accés a un únic servidor, per la qual cosa si el sistema està configurat amb múltiples instàncies, el mateix *token* permet accedir a qualsevol d'elles.

6.2 Serveis REST

Un servei REST és un sistema d'intercanvi d'informació creat per Roy Fielding que utilitza el protocol HTTP[18] amb la definició RFC 2616 com a base per realitzar l'intercanvi de les dades. Les dades es poden encapsular en format JSON o XML.

Les característiques principals d'un servei REST són:

- Protocol client/servidor sense estat
- Operacions de les dades relacionades amb l'especificació HTTP, POST crear registres, PUT modificar registres, GET llegir registres i DELETE eliminar registres.
- L'identificador de la funcionalitat és la URI (direcció única).

Un exemple seria `/ws/user/123` si l'operació indicada es GET, indica que es volen obtenir les dades de l'usuari 123, en canvi si l'operació es PUT indica que es volen modificar les dades de l'usuari 123.

- Separació entre client i servidor, es poden desenvolupar les parts per separat la qual cosa augmenta l'escalabilitat i l'ús de components diferents.
- Independent de plataformes, permet adaptar-se a qualsevol llenguatge de programació o plataforma, per la qual cosa és ideal per projectes d'IOT.

El sistema habilita dos serveis REST que donen accés a les dades o permeten manipular-les. La comunicació amb el HUB es realitza a través del servei `/devices` i la comunicació amb l'aplicació web i l'aplicació Android amb el servei `/backoffice`. El servei `/devices` no requereix del sistema d'autorització, ja que principalment serveix per recollir dades dels sensors i les adreces del mateixos permeten tenir un control de les comunicacions.

En canvi tots els recursos necessaris en la gestió de les aplicacions, requereixen l'autorització del sistema, per la qual cosa en totes dues aplicacions és necessari subministrar les credencials per ser identificat en el sistema.

En cada operació, el sistema retornarà un objecte JSON amb la següent estructura:

```
{
  "result": "OK",
  "message": "<missatge sobre l'operació>",
  "object": <objecte si és necessari a la resposta>
}
```

Il·lustració 30 - Resultat d'una crida al servei REST

A l'[annexa C](#) es detallen totes les funcions que subministra el servidor.

6.3 Accés al model de dades. Objectes DAO

El backoffice s'estructura segons el model MVC en el qual el model de dades està diferenciat dels controladors i de la part de visualització de les dades. La capa del model, queda implementada a través d'objectes DAO[9] que s'implementen en una classe Java que s'encarrega de realitzar les consultes a la base de dades i encapsular els resultats en objectes JAVA.

Depenent de les dades que es vulguin consultar o manipular, són necessàries unes taules de la base de dades diferents, per la qual cosa per cada taula o funcionalitat, es crea un objecte DAO que dóna accés a aquestes. Per exemple si es vol accedir a les dades d'un usuari, és necessari utilitzar l'objecte DAO implementat a la classe UserDAO.

```

@SuppressWarnings("unchecked")
public class UserDAO implements IResultDAO {
    private static final int USER = 1;
    private static final String CREATE_USER = "insert into user(email, name, password,
        installation_id, user_type_id) values (?,?,,?,?)";
    private static final String MODIFY_USER = "update user set email=?, name=?,
        password=?, installation_id = ?, user_type_id = ? where id = ?";
    private static final String MODIFY_USER_WITHOUT_PASSWORD = "update user set email=?,
        name=?, installation_id = ?, user_type_id = ? where id = ?";
    private static final String DELETE_USER = "delete from user where id = ? ";
    private static final String QUERY_USERS = "select id, email, name, '' as password,
        installation_id, user_type_id from user ";

    // Acceso a la base de datos
    private static BBDD bbdd = App.getBD();

    private static UserDAO _instancia = new UserDAO();

    public static boolean validateUser(String user, String pass) {
        try {
            StringBuilder sql = new StringBuilder();
            sql.append(QUERY_USERS);
            sql.append("where email = ? and password = ?");
            User u = (User) bbdd.doSelect(sql.toString(), USER, _instancia, user, pass);
            return (u != null);
        } catch (Exception ex) {
            return false;
        }
    }

    @Override
    public Object dataToBean(ResultSet res, int tipo) throws SQLException {
        switch (tipo) {
            case USER:
                User us = new User();
                us.setId(res.getLong("id"));
                us.setEmail(res.getString("email"));
                us.setName(res.getString("name"));
                us.setPassword(res.getString("password"));
                if (res.getObject("installation_id") != null)
                    us.setInstallation_id(res.getLong("installation_id"));
                us.setType((res.getObject("user_type_id") != null) ?
                    res.getInt("user_type_id") : 2);
                return us;
            }
        }
        return null;
    }
}

```

Codi font 11 - Exemple d'un objecte DAO

Totes les classes DAO utilitzen la classe BBDD i implementen la interfície IResult. La classe subministra diversos mètodes d'accés a les dades a través de consultes SQL.

Al principi de la classe, es creen les diferents consultes que gestiona l'objecte DAO, també es crea accés a la classe BBDD que dona accés a la connexió amb MySQL [6].

A l'exemple, es fa una consulta a la base de dades per comprovar que l'usuari té accés a l'aplicació. Aquesta consulta es realitza amb la funció doSelect que requereix els següents paràmetres:

sql.toString. Consulta SQL amb les taules i camps a consultar

USER. És una constant que indica a l'empaquetador de les dades IResult el tipus de dades que ha de transformar, és una constant numèrica que es declara a la capçalera de cada objecte DAO

_instancia. Indica quina classe s'encarrega d'empaquetar les dades rebudes des de la base de dades en un objecte Java

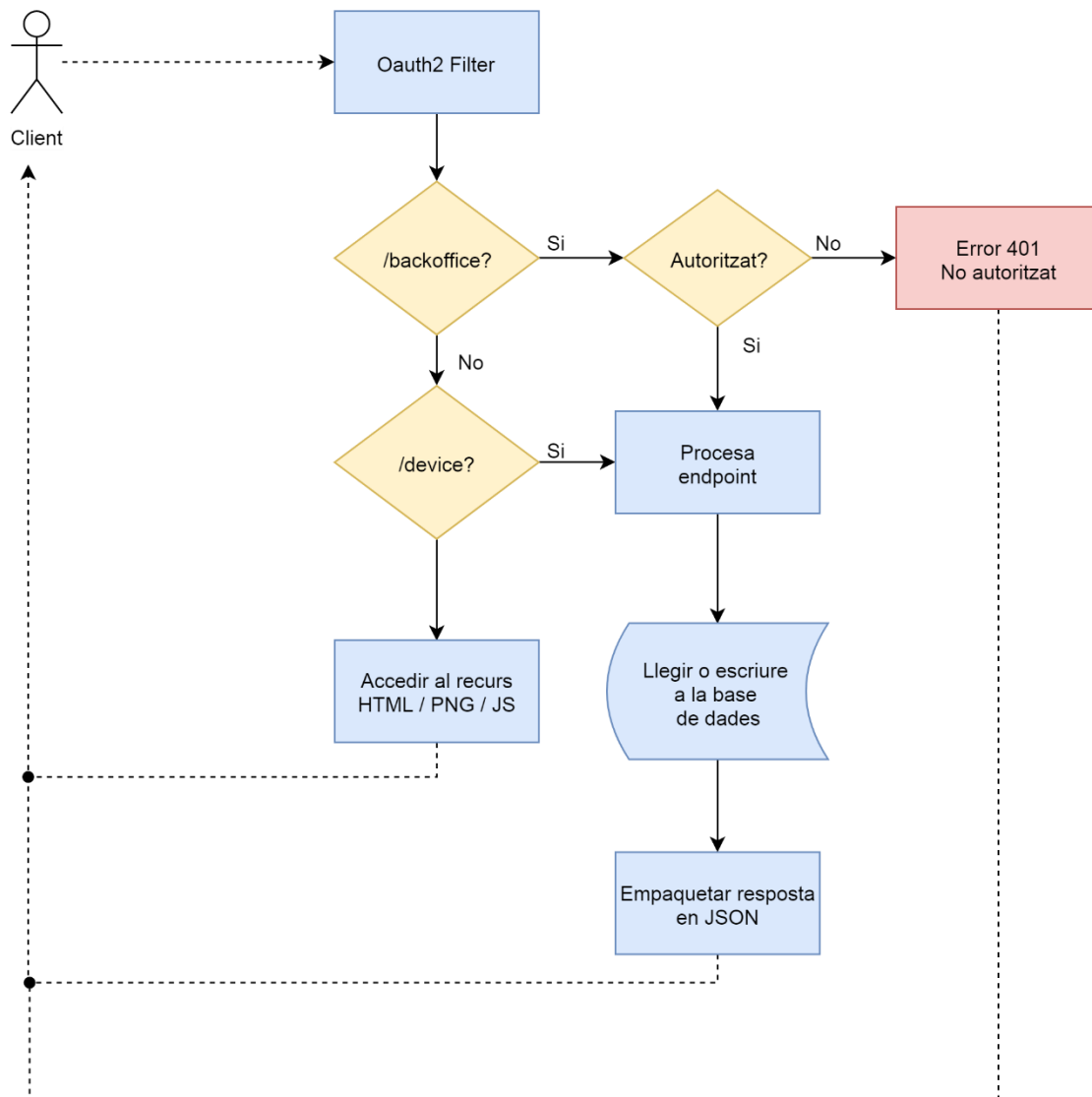
user, pass. Paràmetres que requereix la consulta SQL

En Java, les consultes SQL es realitzen a través del driver JDBC [36] particular de cada base de dades, les dades retornades són accessibles a través de l'objecte ResultSet. La interfície IResult creada el que fa és permetre a la classe BBDD convertir les dades des de l'objecte ResultSet a un objecte de l'aplicació. Aquesta operació de transformació es realitza en el mètode dataToBean que rep un objecte ResultSet i el tipus d'objecte que es vol obtenir i retorna un objecte.

En el cas de l'exemple com es pot apreciar, es crea un nou usuari `User us = new User()` i posteriorment s'assignen els valors a les seves propietats.

6.4 Estructura del backoffice

El backoffice té la funció de gestionar les peticions dels clients (usuaris apps i HUBs), la gran part d'aquesta gestió es realitza a través del servlet que gestiona les peticions en el port 8080 i que permet accedir a recursos del sistema com pàgines HTML, imatges o endpoints. Depenent del recurs al qual es vol accedir, el sistema controla si la petició està autoritzada a través del protocol OAuth2.



Il·lustració 31 - Esquema general Backoffice

Tal com es veu a la il·lustració, totes les peticions s'encaminen cap al filtre Oauth2, si la petició sol·licita accedir a un endpoint /backoffice, revisarà que estigui autoritzat, és a dir si la petició té un token vàlid. En cas afirmatiu, realitzarà les operacions necessàries per a cada endpoint, que normalment a partir de les dades de la petició fan una consulta o modificació a la base de dades. El resultat d'aquestes operacions s'empaqueta en un objecte JSON que serà retornat al client.

Si la petició s'encamina a un endpoint /devices, directament es gestiona, ja que no es requereix autorització. El mateix passa amb altres tipus de recursos com pàgines html, javascript o imatges.

6.5 Processos en background

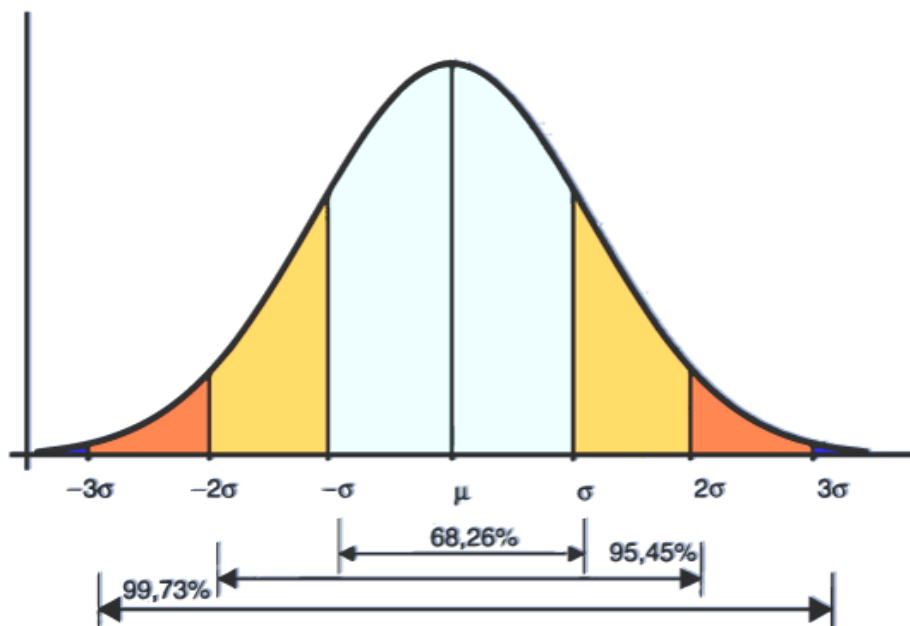
El backoffice a l'iniciar-se l'aplicació executa 2 processos per realitzar càlculs sobre les dades recollides pels sensors:

6.5.1 Càlcul de patrons de comportament

El procés de càlcul de patrons s'executa diàriament per calcular a partir de les dades recollides dels sensors un patró de comportament per cada dia de la setmana. En general les persones grans habitualment realitzen les mateixes activitats i horaris durant el dia. Per norma s'aixequen a la mateixa hora, esmorzen, fan alguna activitat, etc.

L'algorisme es basa en aquesta suposició per emmagatzemar el nombre de vegades que es realitza una activitat (detectada pel sensor) i/o el temps que es fa l'activitat. Per fer-ho es fa servir un càlcul estadístic que a partir de l'històric de valors determina els valors mínim i màxim. A la natura hi ha una tendència a què les dades compleixin un comportament gaussià [37][40]. Aquesta teoria va ser enunciada per **Carl Friedrich Gauss** [39] i indica que els valors més comuns s'agrupen cap al centre, mentre que els mes rars se situen en els extrems.

Es necessari calcular la mitja dels valors i la **desviació típica o estandar** [35] de la mostra. La desviació típica és un valor numèric que representa el terme mitjà de diferència entre les dades i la mitja. A partir d'aquestes dades es pot saber que un percentatge de les dades tindran com a valor entre el mínim i el màxim.



Il·lustració 32 - Campana de Gauss

El càlcul de l'interval bé donat per la fórmula:

$$\text{Interval} = VC * \frac{\text{DES}V}{VC} \quad \frac{\text{DES}V}{1.5} \quad \text{Desviació estàndard}$$

El valor crític per un percentatge en concret es pot obtenir a través de les taules de distribució normal, en el nostre cas considerem que tot el que estigui fins a 1.5 cops la desviació estandar es considerarà en el límit permès, a més el límit inferior no serà mai inferior al valor més petit de la sèrie i el valor superior més gran que el valor màxim de la sèrie.

Suposem un exemple en el qual es recull els cops que es fa servir un objecte en el temps

[3,3,3,3,3,3,3,3,3,3,4,4,4,4,4,4,4,4,4,4,4,4,5,5,5,5,5,5,5,5,5,5,5,5,5,5,5,5,6,6,6,6,6,6,6,6]

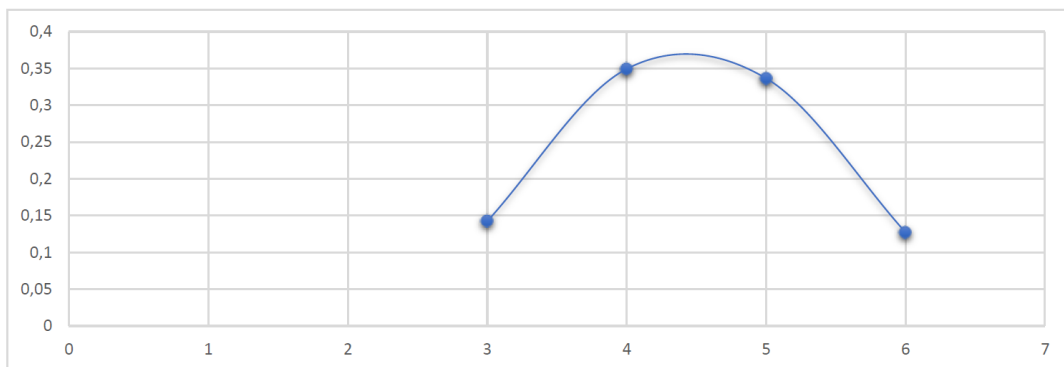
Es calcula la mitja (4.46) i la desviació (1.034), a partir d'aquestes dades es calcula l'interval de variació respecte a la mitja en el nostre cas 1.5 cops la desviació.

Per saber el mínim i el màxim, es resta o suma l'interval a la mitja, en aquest cas és:

$$(4.46 - 1.55, 4.46 + 1.55) \rightarrow (2.91, 6.01) \rightarrow (\text{inferior} : 3, \text{superior} : 6)$$

Es pot apreciar a la representació gràfica com dintre de la corba, els valor més habituals estan entre 4 i 5 però el rang contempla valors d'entre 3 i 6

Mitja: 4.459999999999999
 Desviació: 1.034309391994169
 Interval: 1.5514640879912536
 Limit inferior: 3.0
 Limit superior: 6.0



Il·lustració 33 - Exemple distribució normal 1

El sistema però repeteix el càlcul diàriament per adaptar el patró al comportament de la persona, per la qual cosa si el canvi és progressiu, el sistema no detectarà el canvi com una alerta, però si el canvi és molt sobtat, es detectarà una alerta fins al moment en el qual el sistema aprengui.

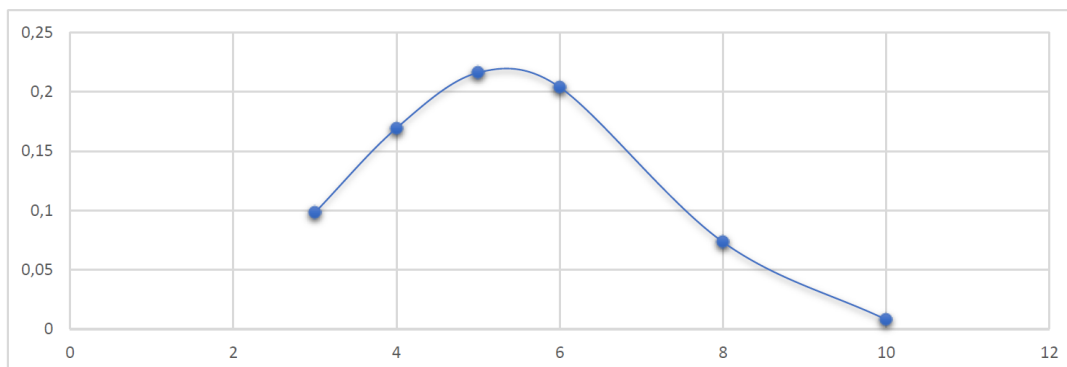
Suposem que el mateix individu incrementa l'ús de l'objecte a 6 o 8 vegades, el sistema recalcula el patró i determina que a partir del moment del càlcul l'objecte es farà servir entre 3 i 8 vegades, és a dir, ha detectat una tendència a l'alça envers els últims dies.

[3,3,3,3,3,3,3,3,3,3,4,4,4,4,4,4,4,4,4,4,4,4,5,5,5,5,5,5,5,5,5,5,5,5,5,5,5,5,6,6,6,6,6,6,6,6,8,8,8,8,8,8,8,8,8,8,8,8,10]

$$(5.30 - 2.73, 5.30 + 2.73) \rightarrow (2.57, 8.03) \rightarrow (\text{inferior} : 3, \text{superior} : 8)$$

L'avantatge és que aquest canvi de patró es progressiu, per la qual cosa, els primers cops que hi ha valors de 8, saltarà la alerta perquè estan fora del patró, però al cap de 10 o 12 vegades el sistema ja ho deixa de considerar una anomalia, per la qual cosa ja no saltarà l'alerta.

Mitja: 5.307692307692308
 Desviació: 1.8193669651409392
 Interval: 2.729050447711409
 Limit inferior: 3.0
 Limit superior: 8.0



Il·lustració 34 - Exemple distribució normal 2

Per realitzar el càlcul es fa servir la llibreria **Apache Commons Math** [38], que permet realitzar diversos càlculs matemàtics i estadístics.

```
public static Profile calcUseProfile(Profile pr) {
    DescriptiveStatistics stats = new DescriptiveStatistics();
    List<Long> data = null;

    data = pr.getCount_samples();
    // Add values to DescriptiveStats object
    for (int i = 0; i < data.size(); i++) {
        stats.addValue(data.get(i));
    }
    // Vars to calculate mean
    double mean, li, ls, desv, interval;
    desv = stats.getStandardDeviation();
    mean = stats.getMean();

    interval = (1.5 * desv) ;
    li = Math.round(mean-interval);
    ls = Math.round(mean+interval);

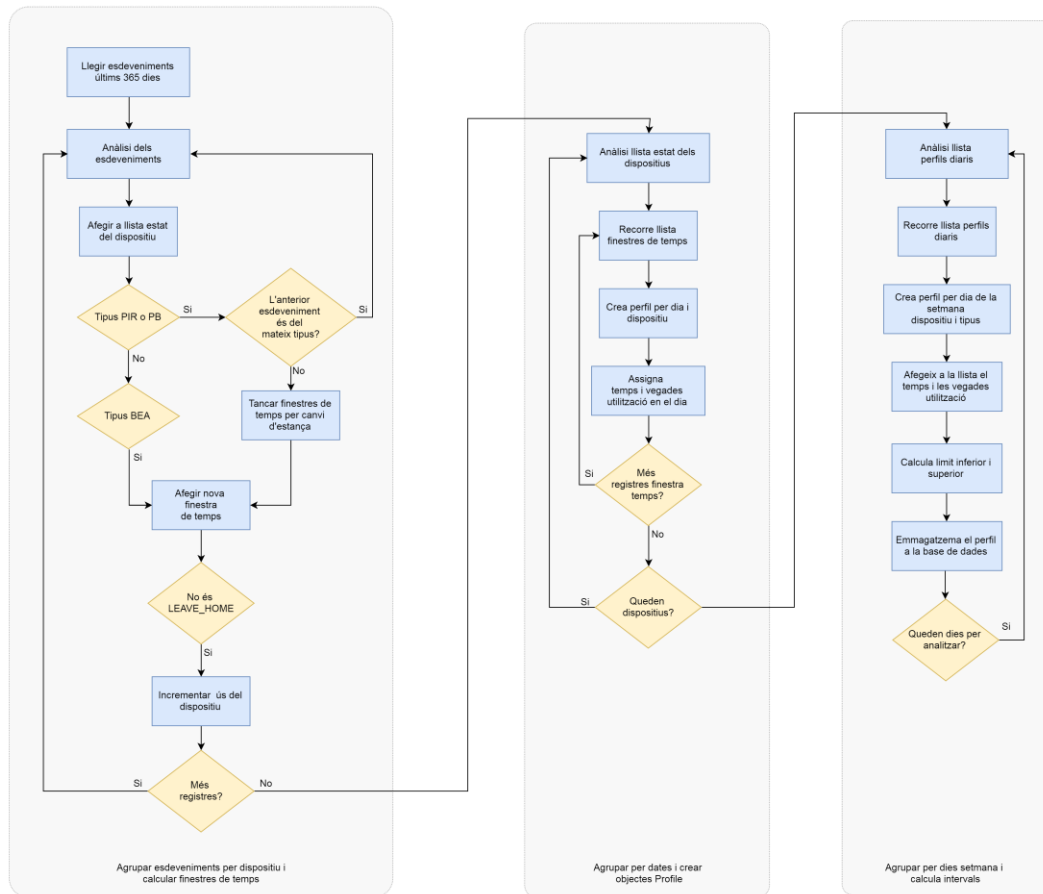
    if (li < stats.getMin()) li = stats.getMin();
    if (ls > stats.getMax()) ls = stats.getMax();

    // Print results to screen:
    System.out.print(" Mitja: " + mean);
    System.out.print(" Desviació: " + desv);
    System.out.print(" Interval: " + interval);
    System.out.print(" Limit inferior: " + li);
    System.out.println(" Limit superior: " + ls);

    pr.setCount((long)mean);
    pr.setMin_count((long)li);
    pr.setMax_count((long)ls);
    pr.setNum_samples(stats.getN());
    return pr;
}
```

Codi font 12 - Càlcul interval patró de comportament

El diagrama de flux del procés de càlcul és el següent:



II-lustració 35 - Diagrama de flux procés de càlcul de patrons

El càlcul dels patrons recull totes les dades i les organitza fins a tenir les dades d'ús per cada dia de la setmana, tipus d'esdeveniment i dispositiu. Per fer aquesta organització, primer s'agrupen tots els esdeveniments per dispositiu i tipus d'esdeveniment. A cada grup es crea una llista de finestres de temps i ús del dispositiu. Amb les dades agrupades es classifiquen per dies, obtenint una llista d'ús per cada dia dels 365 dies analitzats. Finalment es classifica i agrupa la informació per dia de la setmana, obtenint una llista per cada dia de la setmana amb la llista de duracions i utilitzacions del dispositiu i el tipus d'esdeveniment associat.

Amb aquestes dades es calcula d'interval segons una distribució normal, la qual cosa permet saber el mínim i màxim de cops que es farà servir un sensor segons l'ús habitual del dispositiu en l'últim any. Finalment s'emmagatzema cada perfil a la base de dades que serà consultat cada cop que es produeixi un esdeveniment.

A l'[annex D](#) es descriuen com s'ha provat l'algorisme de creació de patrons.

6.5.2 Càlcul esdeveniment d'excés de temps

El segon procés actiu en segon pla en el sistema, és un procés que revisa si l'usuari fa molt de temps que està en una habitació. S'ha de fer en segon pla, perquè si no es produeix cap altre esdeveniment, el sistema no detectaria un comportament erroni, per la qual cosa cada hora es revisa el temps que un usuari està en una habitació i el compara amb el patró de comportament emmagatzemat a la base de dades. En el cas de detectar una discrepància amb el patró, crearà una alerta UNUSUAL_TIME.

7 Entorn real i proves

Aquest apartat descriu el procés de configuració dels sensors en una instal·lació real, i les proves posteriors realitzades per comprovar el correcte funcionament de la solució.

7.1 Instal·lació dels sensors

La posada en marxa dels sensors, implica la configuració dels paràmetres dels beacons, la incorporació de les seves dades a la plataforma i la ubicació dintre del domicili.

El kit d'instal·lació estarà format per:

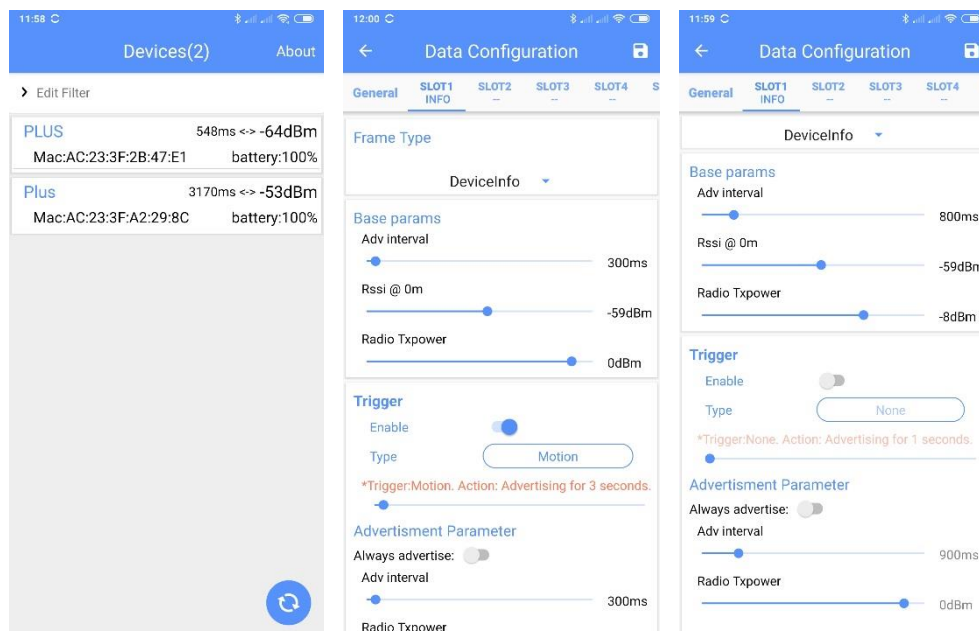
- 3 sensors de moviment.
- 3 sensors de presència per infrarojos.
- 1 sensor de presència permanent.
- 1 HUB

A més per visualitzar els diferents esdeveniments s'utilitzarà l'aplicació mòbil i l'aplicació web.

7.1.1 Configurant els sensors

Per configurar els sensors de moviment, es requereix l'app del fabricant Minew[42]. Amb aquesta aplicació es pot accedir a les propietats de cada beacon i configurar la freqüència i la forma en la qual s'envien els paquets quan es detecta moviment. Amb aquesta mateixa aplicació es pot configurar el sensor de presència permanent.

En obrir l'aplicació mostra la llista de dispositius detectats, pitjant sobre cada dispositiu, es poden configurar les dades després d'haver introduït la paraula de pas, que per defecte és minew123.



Il·lustració 36 – Pantalles aplicació BeaconSet+

Per assegurar una bona cobertura i duració de la bateria, els sensors es configuren de forma diferent depenent de la seva utilitat, en el cas del de moviment, com no transmet constantment,

es pot ajustar la potència a pràcticament el màxim (0db), ja que només transmetrà durant 5s, aquest últim temps és una quantitat arbitrària que s'ha considerat suficient perquè el HUB detecti el senyal, podria ser un temps menor, però s'ha preferit donar un marge més ampli per assegurar la recepció dels paquets. En el cas del sensor de presència, la potència és més continguda -8db perquè és un dispositiu que transmet constantment, per la qual cosa s'ha preferit no donar la màxima potència per estalviar bateria. En aquest cas l'interval també és més conservador, ja que emetrà el senyal cada 800ms. És possible que aquests valors hagin de ser modificats en instal·lacions reals depenent de l'estructura de la casa, tipus de parets, etc. però són un punt de partida com a configuració de fàbrica.

Sensors de moviment Bluetooth		Sensor de presència Bluetooth	
Interval	300ms	Interval	800ms
Time	5s	Time	
TxPower	0db	TxPower	-8db

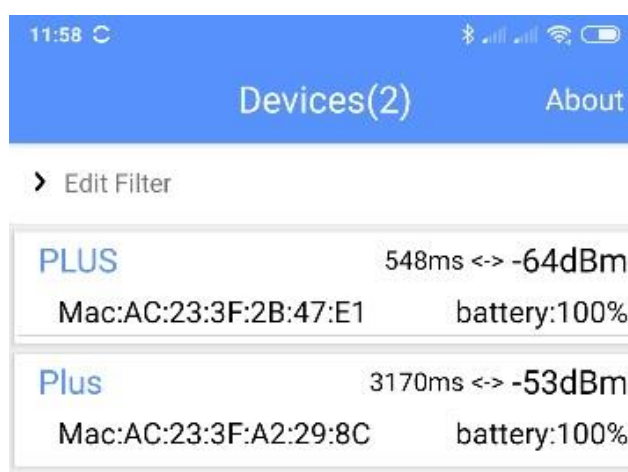
Taula 5 - Configuració sensors Bluetooth

Els sensors de radiofreqüència no tenen cap configuració.

7.1.2 Configurar els dispositius en el sistema

Es donaran d'alta els diferents sensors i el HUB en el sistema a través de l'opció Dispositius de l'aplicació web.

Les adreces dels diferents sensors s'han obtingut amb l'aplicació BeaconSet+ [41], les adreces dels sensors de radiofreqüència es poden visualitzar directament amb el HUB, quan no té cap dispositiu connectat permet visualitzar les dades de qualsevol dispositiu detectat.



Il·lustració 37 - BeaconSet+

Backoffice

Monitorització

usuaris

Instal·lacions

Dispositius

Dispositius

Adreça

Id	Tipus	Adreça	alias	n/s	Utilis
1	PB	AC:23:3F:2B:47:E1	CASA		
2	BEA	AC:23:3F:A2:29:8C	PASTILLER	bbbb	
15	PIR	15430494	MENJADOR	11111123sss	
41	HUB	30:AE:A4:C0:88:E8	HUB		
42	PIR	15308718	CUINA	1234	

Desde Barcelona amb SeniorDomo 2019

Il·lustració 38 - Dispositius donats d'alta dintre del sistema

Cadascun dels sensors s'instal·la en un electrodomèstic (cafetera, nevera, caps de pastilles, etc.).



Il·lustració 39 - Instal·lació dels sensors a cada electrodomèstic

Acció	Dispositiu	Esdeveniment esperat
Entrada a casa	AC:23:3F:2B:47:E1	AT_HOME
Entrada a cuina	3596031003	AT_ROOM
Fer un cafè	AC:23:3F:A2:29:8C	USED_OBJECT
Pastiller	AC:23:3F:A2:29:7E	USED_OBJECT
Entrada a menjador	15430494	AT_ROOM
Entrada a cuina	15308718	AT_ROOM

Taula 6 - Llista de dispositius proves

Per poder documentar les proves, s'activarà un gravador de pantalla al mòbil, de forma que posteriorment es crearà un vídeo amb les imatges sincronitzades dels logs, la pantalla del mòbil i la imatge real.

El següent conjunt de proves, serà generar alertes en el sistema, com ja s'ha generat un patró de referència a partir d'esdeveniments preestablerts, si utilitzem els objectes més del normal, el sistema alertarà del canvi detectat. En aquest cas es faran múltiples moviments del pastiller.

Una altra alerta que es provocarà, és la d'activitat estant fora de casa, en aquest cas s'apagarà el dispositiu beacon, per la qual cosa el sistema al no detectar-lo en 30s crearà un esdeveniment LEAVE_HOME, i a partir d'aquest moment, qualsevol altre esdeveniment de presència a una habitació o ús d'un objecte provocarà l'alerta.

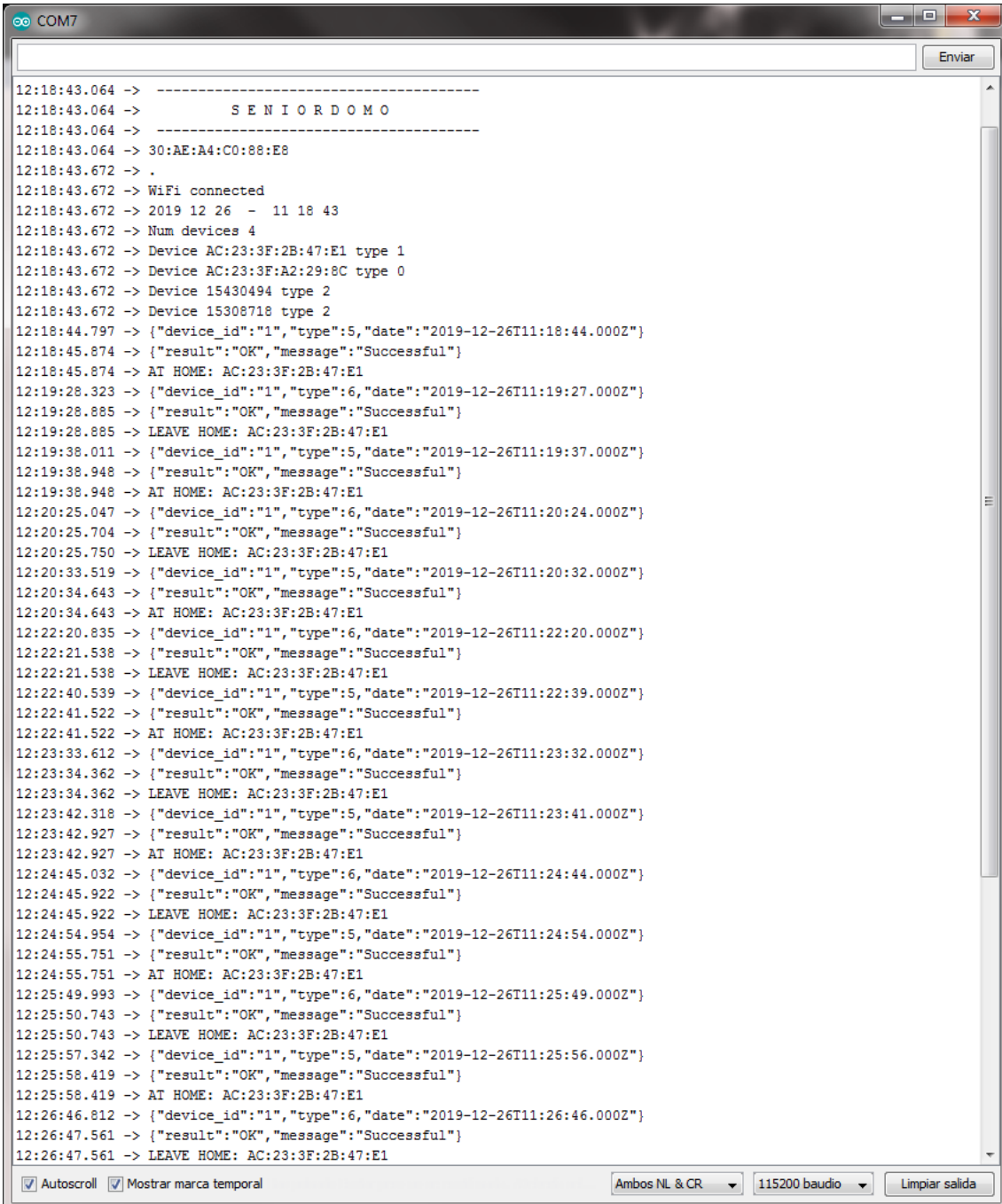
Abans de realitzar la prova de simulació en vídeo, es realitzaran proves per cada tipus de sensor. Les proves es realitzaran 10 vegades per corroborar que el funcionament és estable, el resultat de les proves s'expressarà amb un percentatge que hauria de ser del 100% pel correcte funcionament de cada sensor. Per altra banda es realitzaran també proves de distància per comprovar la cobertura del sistema. Per cada prova s'indicarà la distància i la potència de l'emissor en el cas dels sensors en què es pot configurar la potència.

7.2.1 Dispositiu presència permanent

Hora	Esdeveniment	Resultat	Hora	Esdeveniment	Resultat
12:18:44	Entrar a casa	Detectat	12:24:54	Entrar a casa	Detectat
12:19:27	Sortir de casa	Detectat	12:25:49	Sortir de casa	Detectat
12:19:37	Entrar a casa	Detectat	12:25:56	Entrar a casa	Detectat
12:20:24	Sortir de casa	Detectat	12:26:46	Sortir de casa	Detectat
12:20:32	Entrar a casa	Detectat	12:26:58	Entrar a casa	Detectat
12:22:20	Sortir de casa	Detectat	12:27:39	Sortir de casa	Detectat
12:22:39	Entrar a casa	Detectat	12:27:57	Entrar a casa	Detectat
12:23:32	Sortir de casa	Detectat	12:28:45	Sortir de casa	Detectat
12:23:41	Entrar a casa	Detectat	12:29:00	Entrar a casa	Detectat
12:24:44	Sortir de casa	Detectat	12:29:46	Sortir de casa	Detectat

Taula 7 - Proves beacon presència permanent

El resultat de les proves és d'un 100% de detecció



```

COM7
12:18:43.064 -> -----
12:18:43.064 ->          S E N I O R D O M O
12:18:43.064 -> -----
12:18:43.064 -> 30:AE:A4:C0:88:E8
12:18:43.672 -> .
12:18:43.672 -> WiFi connected
12:18:43.672 -> 2019 12 26 - 11 18 43
12:18:43.672 -> Num devices 4
12:18:43.672 -> Device AC:23:3F:2B:47:E1 type 1
12:18:43.672 -> Device AC:23:3F:A2:29:8C type 0
12:18:43.672 -> Device 15430494 type 2
12:18:43.672 -> Device 15308718 type 2
12:18:44.797 -> {"device_id":"1","type":5,"date":"2019-12-26T11:18:44.000Z"}
12:18:45.874 -> {"result":"OK","message":"Successful"}
12:18:45.874 -> AT HOME: AC:23:3F:2B:47:E1
12:19:28.323 -> {"device_id":"1","type":6,"date":"2019-12-26T11:19:27.000Z"}
12:19:28.885 -> {"result":"OK","message":"Successful"}
12:19:28.885 -> LEAVE HOME: AC:23:3F:2B:47:E1
12:19:38.011 -> {"device_id":"1","type":5,"date":"2019-12-26T11:19:37.000Z"}
12:19:38.948 -> {"result":"OK","message":"Successful"}
12:19:38.948 -> AT HOME: AC:23:3F:2B:47:E1
12:20:25.047 -> {"device_id":"1","type":6,"date":"2019-12-26T11:20:24.000Z"}
12:20:25.704 -> {"result":"OK","message":"Successful"}
12:20:25.750 -> LEAVE HOME: AC:23:3F:2B:47:E1
12:20:33.519 -> {"device_id":"1","type":5,"date":"2019-12-26T11:20:32.000Z"}
12:20:34.643 -> {"result":"OK","message":"Successful"}
12:20:34.643 -> AT HOME: AC:23:3F:2B:47:E1
12:22:20.835 -> {"device_id":"1","type":6,"date":"2019-12-26T11:22:20.000Z"}
12:22:21.538 -> {"result":"OK","message":"Successful"}
12:22:21.538 -> LEAVE HOME: AC:23:3F:2B:47:E1
12:22:40.539 -> {"device_id":"1","type":5,"date":"2019-12-26T11:22:39.000Z"}
12:22:41.522 -> {"result":"OK","message":"Successful"}
12:22:41.522 -> AT HOME: AC:23:3F:2B:47:E1
12:23:33.612 -> {"device_id":"1","type":6,"date":"2019-12-26T11:23:32.000Z"}
12:23:34.362 -> {"result":"OK","message":"Successful"}
12:23:34.362 -> LEAVE HOME: AC:23:3F:2B:47:E1
12:23:42.318 -> {"device_id":"1","type":5,"date":"2019-12-26T11:23:41.000Z"}
12:23:42.927 -> {"result":"OK","message":"Successful"}
12:23:42.927 -> AT HOME: AC:23:3F:2B:47:E1
12:24:45.032 -> {"device_id":"1","type":6,"date":"2019-12-26T11:24:44.000Z"}
12:24:45.922 -> {"result":"OK","message":"Successful"}
12:24:45.922 -> LEAVE HOME: AC:23:3F:2B:47:E1
12:24:54.954 -> {"device_id":"1","type":5,"date":"2019-12-26T11:24:54.000Z"}
12:24:55.751 -> {"result":"OK","message":"Successful"}
12:24:55.751 -> AT HOME: AC:23:3F:2B:47:E1
12:25:49.993 -> {"device_id":"1","type":6,"date":"2019-12-26T11:25:49.000Z"}
12:25:50.743 -> {"result":"OK","message":"Successful"}
12:25:50.743 -> LEAVE HOME: AC:23:3F:2B:47:E1
12:25:57.342 -> {"device_id":"1","type":5,"date":"2019-12-26T11:25:56.000Z"}
12:25:58.419 -> {"result":"OK","message":"Successful"}
12:25:58.419 -> AT HOME: AC:23:3F:2B:47:E1
12:26:46.812 -> {"device_id":"1","type":6,"date":"2019-12-26T11:26:46.000Z"}
12:26:47.561 -> {"result":"OK","message":"Successful"}
12:26:47.561 -> LEAVE HOME: AC:23:3F:2B:47:E1

 Autoscroll  Mostrar marca temporal
Ambos NL & CR 115200 baudio Limpiar salida

```

II-lustració 42 - Detall log HUB proves detecció beacon presència permanent



Il·lustració 43 - Detecció d'entrades i sortides a l'app

El següent pas a provar, és la cobertura del sistema, per la qual cosa, es mesurarà en diferents distàncies fins a trobar un punt on ja no es detecti el sensor. Es provarà amb les distàncies de 1, 5, 10, 15, 20 m. A cada punt, es farà una espera de 30s per veure si el sistema encara detecta el sensor.

Distància	Potència	Interval	Resultat
1 m	-8db	800ms	Detectat
5 m	-8db	800ms	Detectat
10 m	-8db	800ms	Detectat
15 m	-8db	800ms	No detectat
20 m	-8db	800ms	No detectat
12 m	4db	300ms	Detectat
15 m	4db	300ms	Detectat
20 m	4db	300ms	No detectat

Taula 8 - Resultat proves distància

El resultat de les proves és que el sistema és capaç de detectar el sensor a una distància màxima de 10m amb una configuració de -8db podent allargar la distància a 15m canviant la configuració del sensor a la màxima potència. Amb una distància de 10m es cobreix una llar de 100m² amb unes dimensions de per exemple 15 x 6,5 i situant el sensor a una distància intermèdia de 7,5m.

S'arriba a la conclusió que en llars més grans o amb una estructura de parets més complicada s'haurà d'incloure un segon hub que faci de repetidor i que permeti assegurar la completa cobertura de la llar.

7.2.2 Dispositiu presència

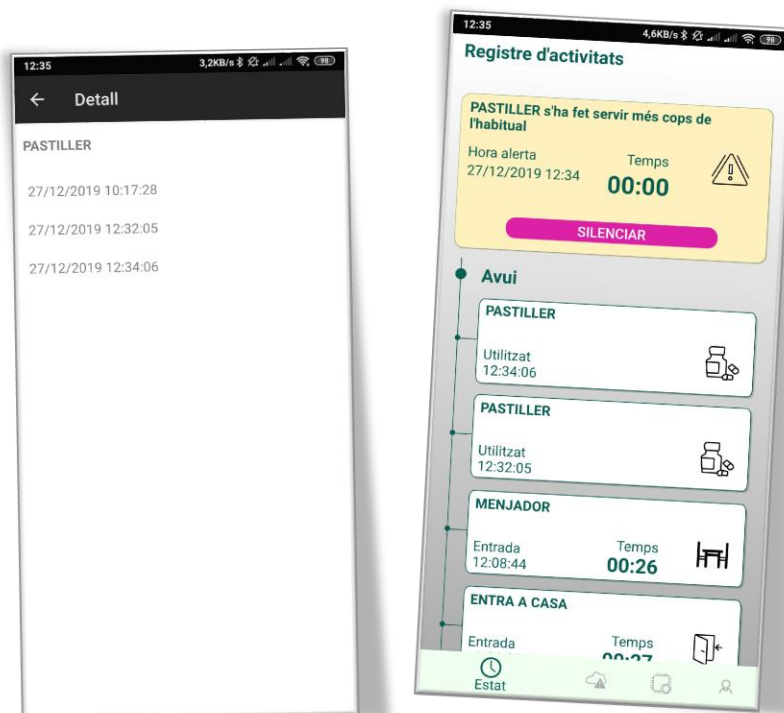
En el cas dels dispositius RF, no es pot configurar la potència per la qual cosa la cobertura ve marcada per la qualitat de l'antena de l'emissor i el receptor. Es provarà el sensor a les mateixes distàncies d' 1, 5, 10, 15 i 20 m.

Distància	Resultat
1 m	Detectat
5 m	Detectat
10 m	Detectat
15 m	Detectat
20 m	Detectat

S'ha pogut comprovar que la transmissió RF permet cobrir distàncies més grans que els dispositius Bluetooth, i s'han cobert distàncies de fins 20m sense cap problema a la recepció. La qual cosa permet assegurar que el sistema permet cobrir habitatges de 100 m² o més.

7.2.3 Prova ús d'un objecte

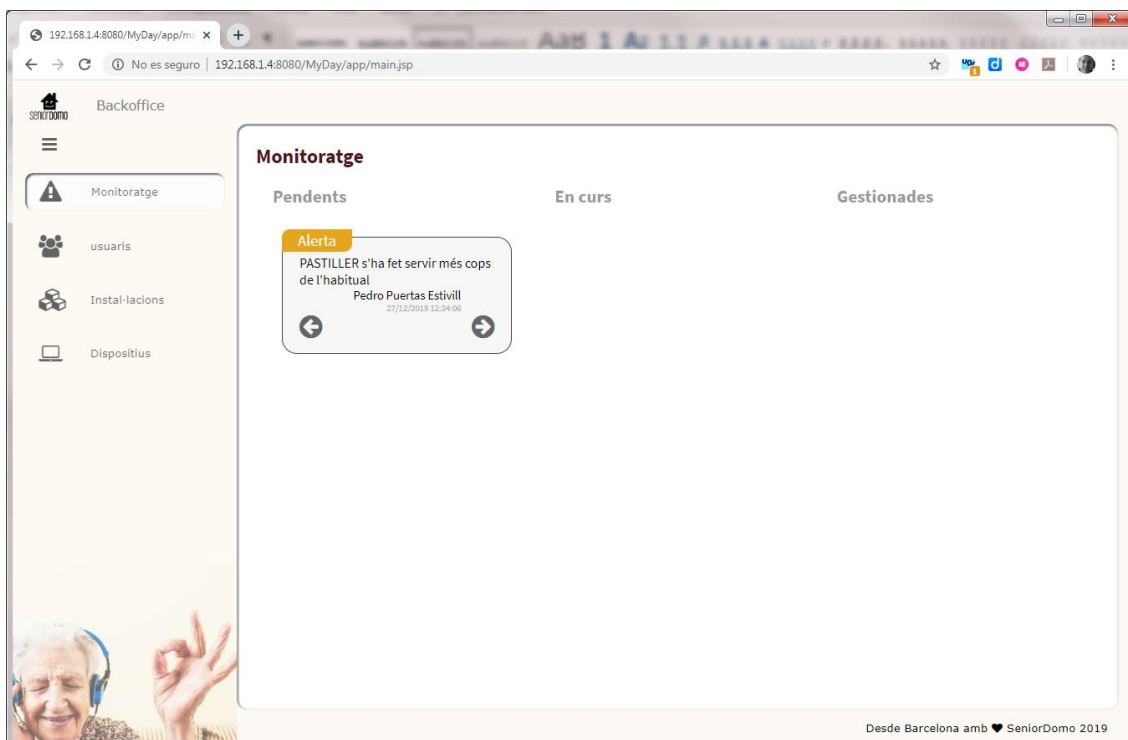
En aquesta prova es comprova que la comparació del patró calculat és capaç de detectar un ús anormal d'un objecte. En aquest cas, el pastiller té el patró calculat en un ús habitual de 2 cops al dia. Si s'utilitza el pastiller un 3r cop, hauria de generar-se una alerta d'ús no habitual.



Il·lustració 44 - Prova ús d'un objecte

S'ha fet ús del PASTILLER a les 10:17, a les 12:32 i al fer-lo servir a les 12:34, el sistema ha creat una alerta d'ús no habitual.

Per tant es pot confirmar que el sistema es capaç de detectar per cada esdeveniment si hi ha variacions amb el patró i alertar als usuaris de la discrepància.



Il·lustració 45 - Visualització alerta a l'aplicació web

Es comprova que a més l'alerta també apareix en el panell del monitor de l'aplicació web.



Il·lustració 46 - Acabat final del producte

8 Conclusions i línies de futur

El projecte m'ha permès implementar tots els coneixements assolits durant el grau multimèdia, totes les competències relatives a la gestió de projectes m'han permès planificar adequadament tots els temps i fases del mateix. Els coneixements adquirits a les assignatures de matemàtiques m'han ajudat a desenvolupar el algorisme pel càlcul de patrons. Les competències adquirides en programació web han estat implementades en el desenvolupament del backoffice i l'aplicació web. Per altre banda, el desenvolupament en C sobre la plataforma Arduino m'ha ajudat a ampliar coneixements sobre la plataforma i consolidar els coneixements adquirits. El projecte m'ha permès treballar amb llenguatges i tecnologies diverses com Java, javascript, C, html, CSS i Android. Sense oblidar que el disseny ha format part del desenvolupament de la plataforma en la qual s'han dissenyat una aplicació web i una app visualment atractives, per la qual cosa crec que el grau proporciona un alt grau de preparació per l'assoliment de projectes que es demanden actualment.

8.1 Assoliment d'objectius

El desenvolupament del projecte i les posteriors proves finals, han confirmat l'assoliment dels objectius principals.

Per una banda s'ha construït un dispositiu HUB que permet el control de diferents dispositius Bluetooth i dispositius comercials de radiofreqüència. Encara que en el projecte només s'han fet servir dispositius de presència per infrarojos, dispositius com detectors de fum, CO2, etc. utilitzen la mateixa tecnologia i podran ser interpretats per el dispositiu HUB.

Per altre banda s'ha pogut desenvolupar un algorisme de detecció de patrons que mitjançant mètodes estadístics és capaç de detectar intervals d'utilització de cada sensor i dia de la setmana, que serveixen com a referència per detectar si hi ha un ús anòmal del mateix.

La implementació del backoffice permet una gestió de tota la plataforma, permetent realitzar el manteniment de tots els elements implicats. A més permet la gestió de les possibles alertes que es produeixin en el sistema.

L'aplicació Android dona eines als familiars per controlar el dia a dia del seu familiar, permetent supervisar la seva rutina diària, així com ser alertats en cas de que el sistema detecti comportament inusuals.

L'èxit de les proves confirma la inclusió de la tecnologia dintre de la plataforma **Seniordomo**, que ho incorporarà durant el primer trimestre de l'any.

8.2 Possibles millores

Hi ha però diversos punts de millora dintre del projecte per la seva implementació comercial que es desenvoluparan en els pròxims mesos.

8.2.1 Millores hardware

- Inclusió d'un mòdul SIM800L [43] per realitzar les comunicacions amb el servidor directament sense dependre d'un enrutador wifi.
- Desenvolupament del PCB per poder fer fabricació en sèrie.
- Inclusió de sensors de fum, foc i CO2.
- Detectar patrons de son a través d'un sensor de moviment instal·lat al llit.

8.2.2 Millores programari

- Implementació d'un sistema OTA [44] que permeti realitzar actualitzacions del programari remotament.
- Possibilitat d'incloure diversos HUBs en una mateixa llar per millorar la cobertura.
- Afegir paràmetres associats a cada sensor per ajustar millor el seu comportament, per exemple el temps per decidir si un dispositiu de moviment s'ha deixat de fer servir.
- Versió IOS de l'aplicació.
- Mostrar en format de taula els horaris realitzats per el sènior de forma que es pugui entendre mol fàcilment la seva rutina.

9 Bibliografia

- [1] **OCU** (2017). Encuesta de OCU: el 82% de los mayores españoles quiere envejecer en su propia casa [Article en línia]
<<https://www.ocu.org/organizacion/prensa/notas-de-prensa/2017/mayores-vivienda-07122017>>
[Data consulta : 04/12/2019]
- [2] **Leichsenring and Billings** (2005). Integrating health and social care services for older persons - Publications - European Centre for Social Welfare Policy and Research
[Article en línia] <<https://www.euro.centre.org/publications/detail/323>>
[Data consulta : 04/12/2019]
- [3] **Ipyme** (2019). Herramienta DAFO [Article en línia]
<<https://dafo.ipyme.org/Home>>
[Data consulta : 04/12/2019]
- [4] **Arduino** (2019). What is Arduino? [Article en línia]
<<https://www.arduino.cc/en/Guide/Introduction>>
[Data consulta : 04/12/2019]
- [5] **Col·laboradors Viquipèdia** (2019). Bluetooth low energy beacon [Article en línia]
<https://en.wikipedia.org/wiki/Bluetooth_low_energy_beacon>
[Data consulta : 04/12/2019]
- [6] **Col·laboradors Viquipèdia** (2019). MySQL [Article en línia]
<<https://ca.wikipedia.org/wiki/MySQL>>
[Data consulta : 04/12/2019]
- [7] **Rivera, Jorge** (2017). Los 433 Mhz y el software libre [Article en línia]
<<https://empresas.blogthinkbig.com/los-433-mhz-y-el-software-libre-parte-1/>>
[Data consulta : 04/12/2019]
- [8] **Col·laboradors Viquipèdia** (2019). Reloj en tiempo real [Article en línia]
<https://es.wikipedia.org/wiki/Reloj_en_tiempo_real>
[Data consulta : 04/12/2019]
- [9] **Col·laboradors Viquipèdia** (2019). Data access object [Article en línia]
<https://en.wikipedia.org/wiki/Data_access_object>
[Data consulta : 04/12/2019]
- [10] **Col·laboradors Viquipèdia** (2019). NodeMCU - Wikipedia [Article en línia]
<<https://en.wikipedia.org/wiki/NodeMCU>>
[Data consulta : 21/11/2019]
- [11] **Espressif** (2019). ESP32 Overview | Espressif Systems [Article en línia]
<<https://www.espressif.com/en/products/hardware/esp32/overview>>
[Data consulta : 21/11/2019]
- [12] **Espressif** (2019) esp32_technical_reference_manual_en.pdf [Article en línia]
<https://www.espressif.com/sites/default/files/documentation/esp32_technical_reference_manual_en.pdf>
[Data consulta : 21/11/2019]
- [13] **Parecki, Aaron** (2019). OAuth 2.0 — OAuth [Article en línia]
<<https://oauth.net/2/>>
[Data consulta : 25/11/2019]

- [14] **Friends of Fritzing** (2019). Fritzing [Article en línia]
<<http://fritzing.org/home/>>
[Data consulta : 25/11/2019]
- [15] **IONOS** (2019). OAuth | Todas las claves del protocolo y de su versión OAuth2 - IONOS [Article en línia]
<<https://www.ionos.es/digitalguide/servidores/seguridad/oauth-y-su-version-oauth2/>>
[Data consulta : 25/11/2019]
- [16] **Reinke, Johann** (2016). Understanding OAuth2 « BubbleCode by Johann Reinke [Article en línia]
<<http://www.bubblecode.net/en/2016/01/22/understanding-oauth2/>>
[Data consulta : 25/11/2019]
- [17] **Col·laboradors Viquipèdia** (2019). JSON - Viquipèdia, l'enciclopèdia lliure [Article en línia]
<<https://ca.wikipedia.org/wiki/JSON>>
[Data consulta : 25/11/2019]
- [18] **W3** (1999). Hypertext transfer protocol [Article en línia]
<<https://www.w3.org/Protocols/rfc2616/rfc2616.html>>
[Data consulta : 25/11/2019]
- [19] **Tutorials point** (2019). C library function - sprintf() [Manual en línia]
<https://www.tutorialspoint.com/c_standard_library/c_function_sprintf.htm>
[Data consulta : 25/11/2019]
- [20] **Arduino** (2019). JSON Library for Arduino [Programari en línia]
<https://github.com/arduino-libraries/Arduino_JSON>
[Data consulta : 25/11/2019]
- [21] **BBVAOPEN4U** (2016). API REST: qué es y cuáles son sus ventajas en el desarrollo de proyectos [Article en línia]
<<https://bbvaopen4u.com/es/actualidad/api-rest-que-es-y-cuales-son-sus-ventajas-en-el-desarrollo-de-proyectos>>
[Data consulta : 25/11/2019]
- [22] **Rodriguez, Alex** (2015). Servicios Web de RESTful: Los aspectos básicos [Article en línia]
<<https://www.ibm.com/developerworks/ssa/library/ws-restful/index.html>>
[Data consulta : 25/11/2019]
- [23] **Col·laboradors Viquipèdia** (2019). Bluetooth Low Energy - Wikipedia [Article en línia]
<http://en.wikipedia.org/wiki/Bluetooth_Low_Energy>
[Data consulta : 25/11/2019]
- [24] **Bluetooth SIG** (2019). Service Discovery | Bluetooth Technology Website [Article en línia]
<<https://www.Bluetooth.com/specifications/assigned-numbers/service-discovery/>>
[Data consulta : 25/11/2019]
- [25] **TI.COM** (2019). Bluetooth Low Energy Scanning and Advertising [Article en línia]
<http://dev.ti.com/tirex/content/simplelink_academy_cc2640r2sdk_1_12_01_16/modules/ble_scan_adv_basic/ble_scan_adv_basic.html>
[Data consulta : 25/11/2019]
- [26] **Bluetooth SIG** (2019). GATT Services | Bluetooth Technology Website [Article en línia]
<<https://www.Bluetooth.com/specifications/gatt/services/>>
[Data consulta : 25/11/2019]
- [27] **JaakkoV** (2017). KBA_BT_0201: Bluetooth advertising data basics [Article en línia]
<https://www.silabs.com/community/wireless/Bluetooth/knowledge-base.entry.html/2017/02/10/Bluetooth_advertisin-hGsf>
[Data consulta : 25/11/2019]

- [28] **Texas Instruments** (2019). Bluetooth Low Energy Scanning and Advertising [Article en línia]
<http://dev.ti.com/tirex/content/simplelink_academy_cc2640r2sdk_1_12_01_16/modules/ble_scan_adv_basic/ble_scan_adv_basic.html>
[Data consulta : 25/11/2019]
- [29] **Nordic semiconductors** (2015). How can I decide scan window? [Forum]
<<https://devzone.nordicsemi.com/f/nordic-q-a/11377/how-can-i-decide-scan-window-interval-for-continuous-scanning>>
[Data consulta : 26/11/2019]
- [30] **Li-Wen Yip** (2013). Decoding and sending 433MHz RF codes with Arduino and rc-switch [Article en línia]
<<https://www.liwen.id.au/arduino-rf-codes/>>
[Data consulta : 26/11/2019]
- [31] **Fetzer, Christian** (2014). Reverse engineering 433 MHz sensors · Christian Fetzer [Article en línia]
<<https://fetzerch.github.io/2014/11/15/reveng433/>>
[Data consulta : 26/11/2019]
- [32] **Col·laboradors Viquipèdia** (2019). Codificació Manchester - Viquipèdia, l'enciclopèdia lliure [Article en línia]
<https://ca.wikipedia.org/wiki/Codificaci%C3%B3_Manchester>
[Data consulta : 26/11/2019]
- [33] **Pérez, Xose** (2013). Decoding 433MHz RF data from wireless switches. The data - Tinkerman [Article en línia]
<<http://tinkerman.eldiariblaui.net/decoding-433mhz-rf-data-from-wireless-switches-the-data/>>
[Data consulta : 26/11/2019]
- [34] **Tutorials point** (2019). C library function - sprintf() - Tutorialspoint [Article en línia]
<http://www.tutorialspoint.com/c_standard_library/c_function_sprintf.htm>
[Data consulta : 27/11/2019]
- [35] **Col·laboradors Monografias.com** (2019). Desviación estandar - Monografias.com [Article en línia]
<<http://www.monografias.com/trabajos89/desviacion-estandar/desviacion-estandar.shtml>>
[Data consulta : 29/11/2019]
- [36] **Tyson, Matthew** (2019). What is JDBC? Introduction to Java Database Connectivity | JavaWorld [Article en línia]
<<https://www.javaworld.com/article/3388036/what-is-jdbc-introduction-to-java-database-connectivity.html>>
[Data consulta : 29/11/2019]
- [37] **Cañas, Juan Jesús; Galo, José R.** (2019). Distribución normal Bachillerato [Article en línia]
<https://proyectodescartes.org/iCartesiLibri/materiales_didacticos/EstadisticaProbabilidadInferencia/DistribucionNormal/2LaDistribucionNormal.html>
[Data consulta : 02/12/2019]
- [38] **Apache foundation** (2016). Math – Commons Math: The Apache Commons Mathematics Library [Article en línia]
<<http://commons.apache.org/proper/commons-math/>>
[Data consulta : 03/12/2019]

- [39] **Col·laboradors Viquipèdia** (2019). Carl Friedrich Gauß - Viquipèdia, l'enciclopèdia lliure [Article en línia]
<https://ca.wikipedia.org/wiki/Carl_Friedrich_Gau%C3%9F>
[Data consulta : 03/12/2019]
- [40] **Gámez, José Carlos** (2012). La herramienta que todos quieren: La Campana de Gauss - Matemáticas Digitales [Article en línia]
<<http://www.matematicasdigitales.com/la-herramienta-que-todos-quieren-la-campana-de-gauss/>>
[Data consulta : 03/12/2019]
- [41] **Google PlayStore** (2019). Beacon+ [Programari en línia]
<<https://play.google.com/store/apps/details?id=com.minew.beaconplus&hl=es>>
[Data consulta : 03/12/2019]
- [42] **Minew** (2019). Ibeacon / Eddystone products [Producte en línia]
<<https://en.minewtech.com/ibeacon-eddystone.html>>
[Data consulta : 03/12/2019]
- [43] **Lastminuteengineers** (2019). Send Receive SMS & Call with SIM800L GSM Module & Arduino [Article en línia]
<<https://lastminuteengineers.com/sim800l-gsm-module-arduino-tutorial/>>
[Data consulta : 03/12/2019]
- [44] **Holmes, Wayne** (2013). Over The Air Programming (OTAP) [Article en línia]
<https://cwh050.blogspot.com/2013/03/otap-part-1_8.html>
[Data consulta : 03/12/2019]
- [45] **Cuidate+** (2019). ELA (esclerosis lateral amiotròfica) [Article en línia]
<<https://cuidateplus.marca.com/enfermedades/neurologicas/ela.html>>
[Data consulta : 03/12/2019]

Annexos

A. Frontend

A.1 Llibreria oauth2.js

Aquest apartat mostra el resultat final de l'aplicació WEB i alguns aspectes de la programació efectuada.

Com s'ha explicat en capítols anteriors, l'accés al sistema es controla a través del protocol OAuth2, en el cas de l'aplicació web, s'ha desenvolupat la llibreria oauth2.js que realitza totes les peticions al backend en mode asíncrona.

Les dades del token i refresh_token s'emmagatzemen a la base de dades del navegador a través de l'objecte localStorage, es pot accedir a la base de dades interna. Amb la crida setItem es poden afegir dades a la mateixa, i amb la crida getItem es recuperen.

```
    this.oauth_data = JSON.parse(localStorage.getItem('oauth_data'));
    localStorage.setItem('oauth_data', JSON.stringify(_this.oauth_data));
```

La llibreria oauth2.js subministra les següents funcions:

login(url, user, pass, callback)

Funció que sol·licita permís per accedir en el sistema, retorna un token per poder realitzar les comunicacions. La funció es asíncrona, per la qual cosa quan el servidor respon es fa una crida a la funció callback.

request(obj)

Funció que realitza les peticions al backend, automàticament subministra el token o si es necessari el refresca. Per la qual cosa l'aplicació fa servir la funció com si fos una crida normal Ajax.

```
oauth2.request({
  url : "/MyDay/ws/backoffice/device_types",
  type: "GET",
  success : function(data) {
    //alert(data.message);
    $.each(data, function(i, obj) {
      $("#select[name='type']").append($('
```

Codi font 13 - Exemple crida javascript oauth2.request

A l'exemple, es passa un objecte amb diferents paràmetres per fer la crida Ajax.

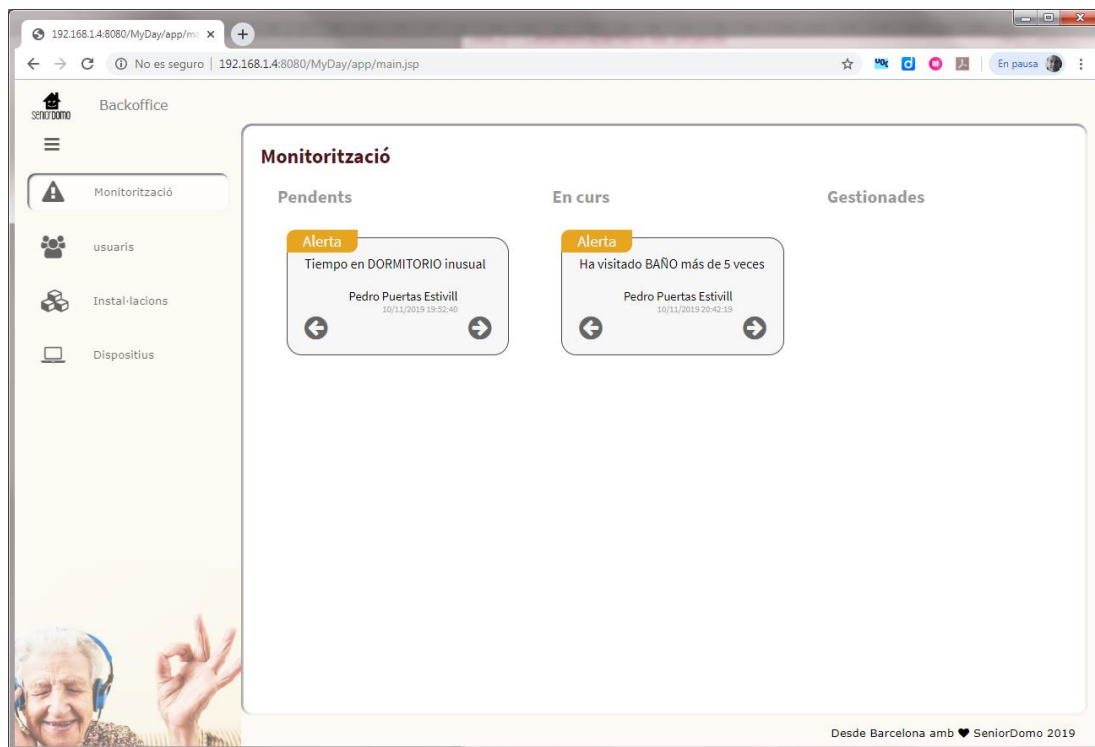
url. Indica la direcció del servidor, en aquest cas és l'endpoint per recuperar els tipus de dispositius.

type. Indica el tipus de petició, GET, POST, etc.

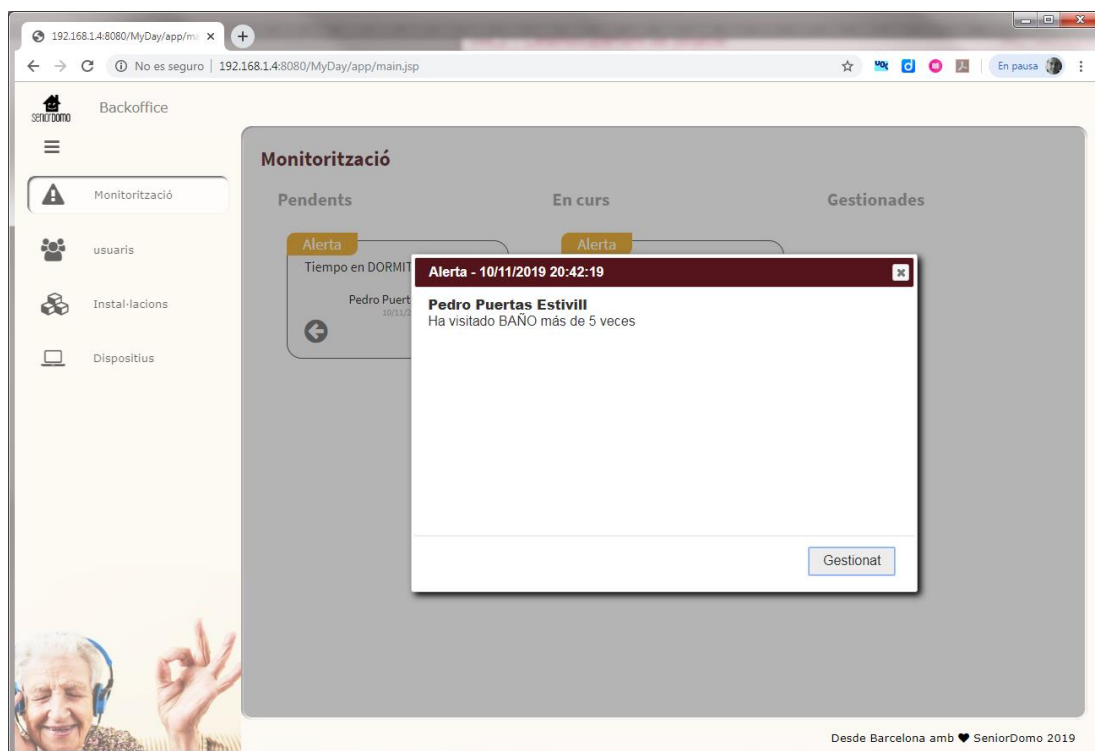
success. Funció que es crida quan el resultat de la sol·licitud no té cap error, en aquest cas amb el resultat s'omple un combobox amb les dades retornades.

A.2 Pantalla principal

El resultat final de la pantalla permet realitzar la gestió de les diferents alertes que es produeixen en el sistema.



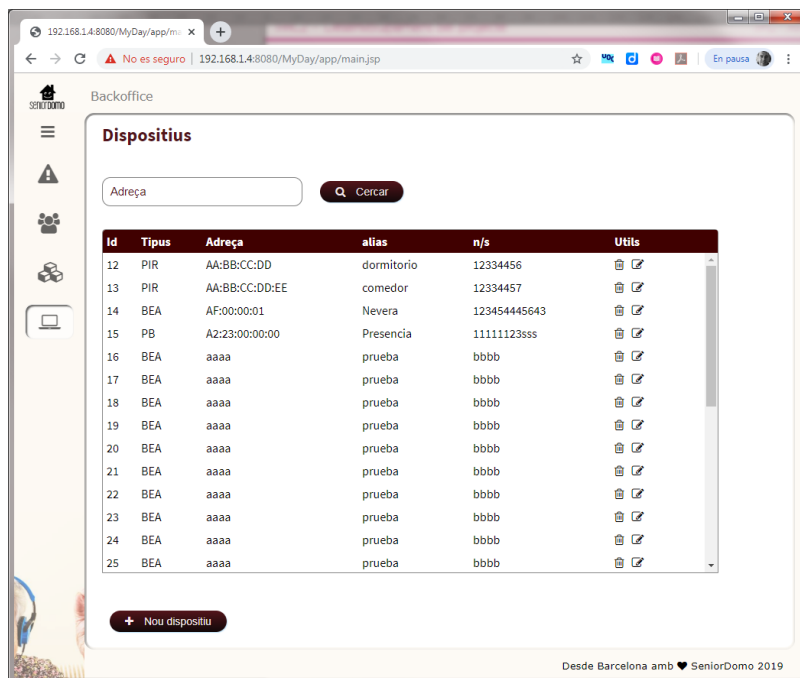
Il·lustració 47 - Art final pantalla monitoratge



Il·lustració 48 - Detall gestió alerta

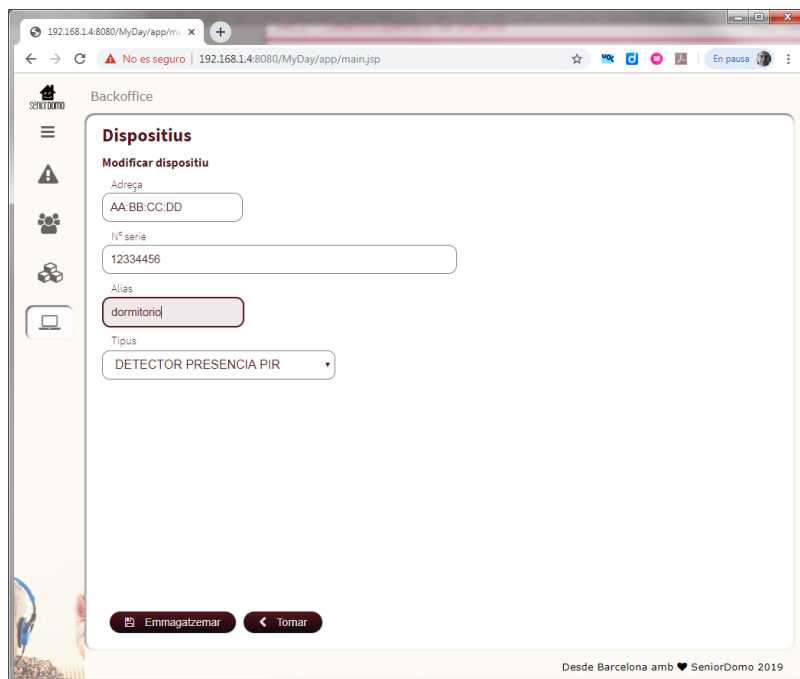
A.3 Manteniments

Un exemple d'un manteniment seria el de dispositius, en el qual es pot apreciar la capçalera amb el filtre, l'àrea de detall, que proporciona scroll per veure totes les dades, i el peu de pàgina que habilita diferents botons amb les diverses accions que es poden fer.



Il·lustració 49 - Art final manteniment

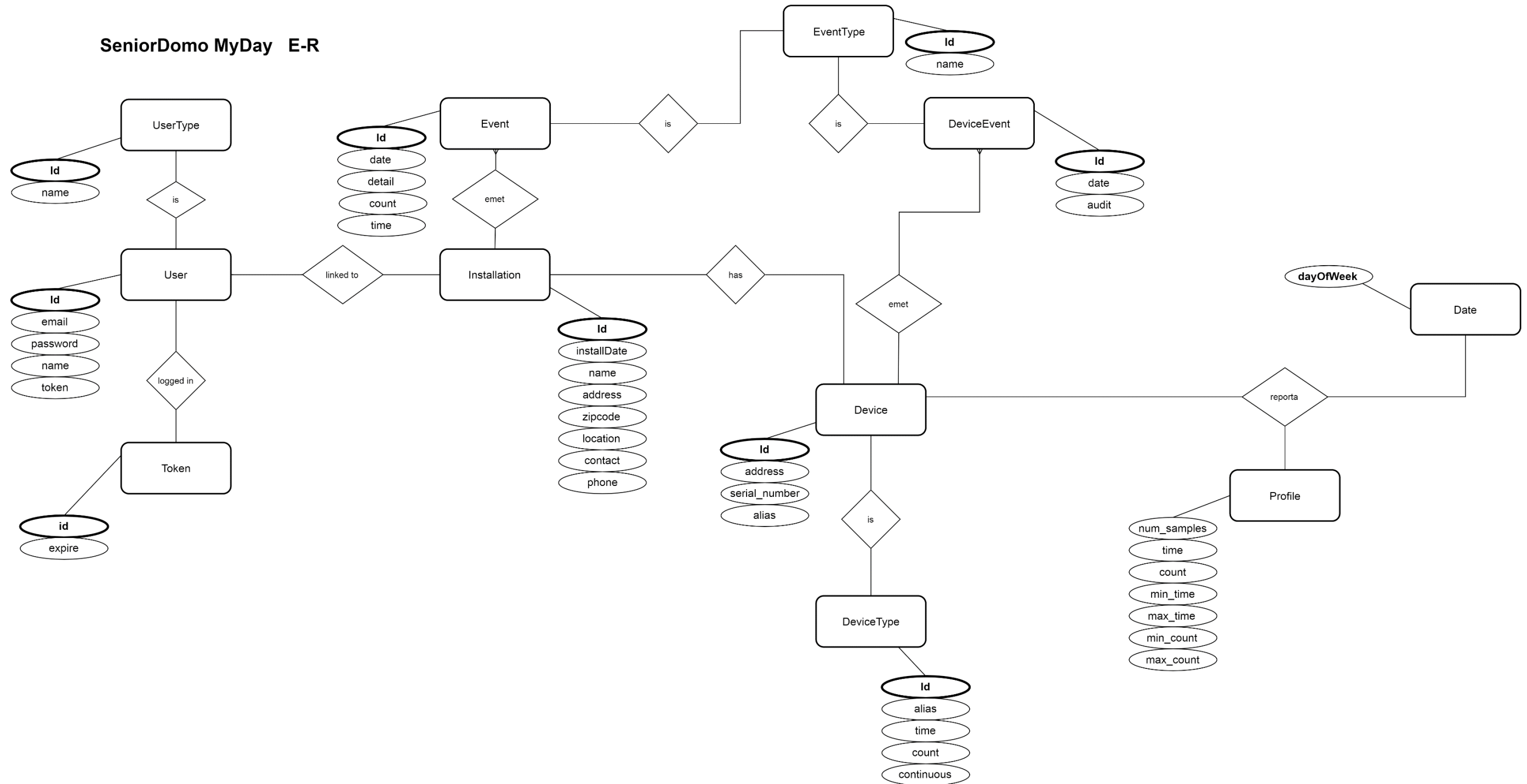
Si es pitja a l'opció d'editar el registre, es mostren els diferents camps depenent del manteniment per poder editar les dades.



Il·lustració 50 - Art final manteniment editar registre

B. Esquema E-R

SeniorDomo MyDay E-R



B.1 Entitats

Aquest apartat descriu cadascuna de les entitats creades en el sistema i les seves relacions.

user_type			
Entitat que representa el tipus d'usuari en el sistema, pot haver usuaris administradors o usuaris i permetrà controlar l'accés a diferents parts del programa.			
Camp	Tipus	Descripció	Extra
id	int(11) *	Identificador únic de tipus d'usuari	Autonumèric
name	varchar(50)	Nom descriptiu del tipus	
Índexs			
Nom	Camps	Tipus	
PRIMARY	id	PrimaryKey	

Taula 9 - Entitat user_type

user			
Entitat que representa un usuari en el sistema			
Camp	Tipus	Descripció	Extra
id	int(11) *	Identificador únic d'usuari	Autonumèric
email	varchar (255)	Correu electrònic de l'usuari	
name	varchar (255)	Nom de l'usuari	
password	varchar (255)	Paraula de pas encriptada de l'usuari	
user_type_id	int(11)	Tipus d'usuari	
installation_id	int (11)	Instal·lació associada a l'usuari	
Índexs			
Nom	Camps	Tipus	
PRIMARY	id	PrimaryKey	
IX_EMAIL	email	Unique	
IX_NAME	name	Non-unique	
FK_USER_INST_idx	installation_id	ForeignKey	
FK_USER_TYPE_idx	user_type_id	ForeignKey	

Taula 10 - Entitat user

token			
Entitat que representa un usuari en el sistema			
Camp	Tipus	Descripció	Extra
id	varchar (255) *	Identificador únic	
user_id	int (11)	Identificador de l'usuari	
expires	datetime	Data i hora en que expira el token	
Índexs			
Nom	Camps	Tipus	
PRIMARY	id	PrimaryKey	
FK_USER_TOKEN_idx	user_id	ForeignKey	

Taula 11 - Entitat token

installation			
Entitat que representa una instal·lació			
Camp	Tipus	Descripció	Extra
id	int (11) *	Identificador únic d'usuari	Autonumèric
install_date	date	Data de l'instal·lació	
name	varchar (255)	Nom associat a l'instal·lació	
address	varchar (255)	Adreça on està instal·lat el sistema	
zipcode	varchar(15)	Codi postal on està instal·lat el sistema	
location	varchar(60)	Població on està instal·lat el sistema	
contact	varchar(120)	Persona de contacte	
phone	varchar(15)	Telèfon de contacte	
Índexs			
Nom	Camps	Tipus	
PRIMARY	id	PrimaryKey	
IX_NAME	name	NonUnique	

Taula 12 - Entitat installation

device			
Entitat que representa un dispositiu (sensor)			
Camp	Tipus	Descripció	Extra
id	int (11) *	Identificador únic de dispositiu	Autonumèric
address	varchar(45)	Adreça única del dispositiu	
serial_number	varchar (45)	Numero de sèrie	
alias	varchar (45)	Sobrenom del dispositiu	
icon	int(11)	Icona que el representa	
device_type_id	varchar(20)	Tipus de dispositiu	
installation_id	int(11)	Instal·lació associada	
Índexs			
Nom	Camps	Tipus	
PRIMARY	id	PrimaryKey	
IX_ADDRESS	address	NonUnique	
FK_DEVICE_TYPE_idx	device_type_id	ForeignKey	
FK_INSTALLATION_idx	installation_id	ForeignKey	

Taula 13 - Entitat device

device_type			
Entitat que representa un tipus de dispositiu (sensor)			
Camp	Tipus	Descripció	Extra
id	varchar(20)	Identificador únic de tipus	Autonumèric
name	varchar(45)	Nom del tipus de dispositiu	
Índexs			
Nom	Camps	Tipus	
PRIMARY	id	PrimaryKey	

Taula 14 - Entitat device_type

event_type			
Entitat que representa un tipus d'esdeveniment			
Camp	Tipus	Descripció	Extra
id	int(11)	Identificador únic de tipus d'esdeveniment	Autonumèric
name	varchar(45)	Nom del tipus d'esdeveniment	
Índexs			
Nom	Camps		Tipus
PRIMARY	id		PrimaryKey

Taula 15 - Entitat event_type

device_event			
Entitat que representa un esdeveniment a un dispositiu (sensor)			
Camp	Tipus	Descripció	Extra
id	int (11) *	Identificador únic d'esdeveniment	Autonumèric
date	datetime	Data i hora de l'esdeveniment	
device_id	int(11)	Dispositiu on s'ha produït	
event_type_id	int(11)	Tipus d'esdeveniment	
audit	tinyint(4)	En observació	
Índexs			
Nom	Camps		Tipus
PRIMARY	id		PrimaryKey
IX_DE_DATE	date		NonUnique
FK_DEEV_TYPE_idx	event_type_id		ForeignKey
FK_DEEV_DEVICE_idx	device_id		ForeignKey

Taula 16 - Entitat device_event

event			
Entitat que representa un esdeveniment a una instal·lació			
Camp	Tipus	Descripció	Extra
id	int (11) *	Identificador únic d'esdeveniment	Autonumèric
installation_id	int(11)	Instal·lació on s'ha produït	
event_type_id	int(11)	Tipus d'esdeveniment	
detail	varchar(512)	Descripció de l'esdeveniment	
status	varchar(10)	Estat de la seva gestió	
date	datetime	Data i hora de l'esdeveniment	
count	int(11)	Num. Vegades inusual	
time	int(11)	Temps inusual	
Índexs			
Nom	Camps		Tipus
PRIMARY	id		PrimaryKey
IX_DATE	date		NonUnique
FK_EVENT_TYPE_idx	event_type_id		ForeignKey

Taula 17 - Entitat event

profile			
Entitat que representa els patrons de comportament associats a cada dispositiu			
Camp	Tipus	Descripció	Extra
device_id	int (11) *	Identificador únic d'esdeveniment	Autonumèric
day_of_week	varchar(3)*	Dia de la setmana en format 3 dig	
type	int(11)*	Tipus (0 - USE 1-IN 2-OUT)	
num_samples	int(11)	Nombre de mostrat per fer el càlcul	
time	int(11)	Temps mig	
count	int(11)	Mitja de cops que s'utilitza el dispositiu	
min_time	int(11)	Temps mínim d'ús	
max_time	int(11)	Temps màxim d'ús	
min_count	int(11)	Mínim utilitzacions	
max_count	int(11)	Màxim utilitzacions	
índexs			
Nom	Camps		Tipus
PRIMARY	device_id, day_of_week		PrimaryKey

Taula 18 - Entitat profile

C. Endpoints

Les principals funcionalitats subministrades pel servei REST són:

/ws/devices/config/{id}

Retorna la configuració per una instal·lació. El HUB subministra la mac-address com a identificador únic en el sistema. És una sol·licitud del tipus GET

/ws/devices/event

Aquesta funció es crida cada cop que un sensor ha detectat un canvi. La informació s'empaqueta en format JSON amb la següent estructura:

```
{
  "id": "<codi del sensor>",
  "date": "dd-MM-yyyyThh:mm:ss.000Z",
  "type": N
}
```

id. Indica l'adreça única del sensor.

date. Indica la data i hora exacte del moment en el qual s'ha produït l'esdeveniment. Es transmet en format UTC

type. Indica el tipus d'esdeveniment:

1. USED_OBJECT | 2. AT_ROOM | 5. AT_HOME | 6. LEAVE_HOME

El tipus 1 l'emeten els dispositius de detecció de moviment, els de tipus 2, són emesos pels dispositius de detecció de presència per infrarojos. I els tipus 5 i 6 són emesos per detector de presència permanent.

/ws/backoffice/users

/ws/backoffice/users/{id}

Retorna la informació de tots els usuaris o de l'usuari identificar per id. És una sol·licitud de tipus GET.

/ws/backoffice/create_user

Crea un nou usuari en el sistema. És una sol·licitud de tipus POST en la qual es cos de la petició conté una estructura JSON amb les dades de l'usuari.

```
{
  "name": "<Nom de l'usuari>",
  "email": "<Correu electrònic de l'usuari>",
  "password": "<Paraula de pas de l'usuari>",
  "user_type_id": "<Tipus d'usuari>",
  "installation_id": <instal·lació associada>
}
```

/ws/backoffice/modify_user

Modifica un usuari existent, és una sol·licitud de tipus PUT, l'estructura es la mateixa que al crear l'usuari, però s'afegeix el camp id amb el codi únic d'usuari subministrat pel sistema.

/ws/backoffice/delete_user/{id}

Dona de baixa l'usuari identificat per {id} en el sistema, és una sol·licitud de tipus DELETE

/ws/backoffice/installations**/ws/backoffice/installations/{id}**

Retorna una llista d'instal·lacions o una en concret identificada per el identificador {id}. És una sol·licitud de tipus GET.

/ws/backoffice/create_installation

Crea una nova instal·lació en el sistema. És una sol·licitud de tipus POST en la qual el cos de la petició conté una estructura JSON amb les dades de la instal·lació.

```
{
  "name": "<nom del senior>",
  "address": "<adreça on està instal·lat>",
  "zipcode": "<codi postal on està instal·lat>",
  "location": "<població on està instal·lat>",
  "contact": "<persona de contacte >",
  "phone": "<telèfon de contacte >"
}
```

/ws/backoffice/modify_installation

Modifica una instal·lació existent, és una sol·licitud de tipus PUT, l'estructura es la mateixa que al crear la instal·lació, però s'afegeix el camp id amb el codi únic d'instal·lació subministrat pel sistema.

/ws/backoffice/delete_installation/{id}

Dona de baixa la instal·lació identificada per {id} en el sistema, és una sol·licitud de tipus DELETE.

/ws/backoffice/devices**/ws/backoffice/devices/{id}**

Retorna una llista de dispositius o un en concret identificat per el identificador {id}. És una sol·licitud de tipus GET.

/ws/backoffice/unused_devices/{id}

Retorna una llista de dispositius o un en concret identificat per el identificador {id} que no estan associats a cap instal·lació. És una sol·licitud de tipus GET.

/ws/backoffice/device_types/

Retorna la llista de tipus de dispositius. És una sol·licitud de tipus GET.

/ws/backoffice/create_device

Crea un nou dispositiu en el sistema. És una sol·licitud de tipus POST en la qual el cos de la petició conté una estructura JSON amb les dades del dispositiu.

```
{
  "alias": "<sobrenom del dispositiu>",
  "serial_number": "<numero de sèrie>",
  "address": "<adreça que l'identifica>",
  "type": "<tipus de dispositiu>",
  "icon": "<icona que l'identifica>",
  "installationId": "<id de la instal·lació associada>"
}
```

/ws/backoffice/modify_device

Modifica un dispositiu existent, és una sol·licitud de tipus PUT, l'estructura es la mateixa que al crear el dispositiu, però s'afegeix el camp id amb el codi únic de dispositiu subministrat pel sistema.

/ws/backoffice/delete_device/{id}

Dona de baixa el dispositiu identificat per {id} en el sistema, és una sol·licitud de tipus DELETE.

/ws/backoffice/events

Retorna una llista d'esdeveniments de les últimes 24h. És una sol·licitud de tipus GET.

/ws/backoffice/updateStatusEvent/{id}/{status}

Actualitza un esdeveniment identificat per {id} amb l'estat {status}. Els valor status pot ser UNREAD, MANAGING o MANAGED. La funció és utilitzada per l'aplicació Web i per l'aplicació Android per canviar l'estat de les alertes.

/ws/backoffice/installationStatus/{id}

Retorna la llista de dispositius i els esdeveniments associats a cadascun, es utilitzada per l'aplicació Android per visualitzar l'estat de l'instal·lació, permet veure cada esdeveniment, temps a una estança, etc.

/ws/backoffice/installationTimeLine/{id}

Retorna la llista d'esdeveniments d'una instal·lació, es utilitzada per l'aplicació Android per visualitzar el històric d'esdeveniments que s'han produït en les últimes hores.

D. Testeig càlcul de patrons

Per comprovar el correcte funcionament de l'algorisme del càlcul de patrons, s'han desenvolupat dues rutines que generen comportaments controlats de forma que es puguin contrastar els valors calculats amb el comportament real.

```
int[] year_days = {0,31,28,31,30,31,30,31,31,30,31,30,31};
try {
    for (int m = 1; m <= 11; m++) {
        for (int i = 1; i <= year_days[m]; i++) {
            DeviceEventDAO.generateFixEvents(date_formatter.parse(
                String.format("%02d%02d", i, m)+"2019"));
        }
    }
    for (int i = 1; i <= 8; i++) {
        DeviceEventDAO.generateFixEvents(date_formatter.parse(
            String.format("%02d", i)+"122019"));
    }
} catch (Exception e) {
    e.printStackTrace();
}
```

Codi font 14 - Rutina per generar esdeveniment fixes de tot l'any

```
public static void generateFixEvents(Date date) {
    try {
        Device casa = DeviceDAO.getDevice(1);
        Device menjador = DeviceDAO.getDevice(15);
        Device cuina = DeviceDAO.getDevice(42);
        Device pastiller = DeviceDAO.getDevice(2);

        DeviceEvent ev = new DeviceEvent();

        SimpleDateFormat df = new SimpleDateFormat("dd/MM/yyyy ");
        SimpleDateFormat hf = new SimpleDateFormat("dd/MM/yyyy HH:mm");
        String dd = df.format(date);

        Date d = hf.parse(dd + "09:00");
        ev.setDate(d);
        ev.setAudit(false);
        ev.setDevice_id(cuina.getId());
        ev.setType(Event.AT_ROOM);
        saveEvent(ev);

        d = hf.parse(dd + "09:10");
        ev.setDate(d);
        ev.setDevice_id(pastiller.getId());
        ev.setType(Event.USED_OBJECT);
        saveEvent(ev);

        d = hf.parse(dd + "10:00");
        ev.setDate(d);
        ev.setDevice_id(menjador.getId());
        ev.setType(Event.AT_ROOM);
        saveEvent(ev);

        d = hf.parse(dd + "12:00");
        ev.setDate(d);
        ev.setDevice_id(casa.getId());
        ev.setType(Event.LEAVE_HOME);
        saveEvent(ev);
    }
}
```

```

d = hf.parse(dd + "14:00");
ev.setDate(d);
ev.setDevice_id(casa.getId());
ev.setType(Event.AT_HOME);
saveEvent(ev);

d = hf.parse(dd + "14:10");
ev.setDate(d);
ev.setDevice_id(cuina.getId());
ev.setType(Event.AT_ROOM);
saveEvent(ev);

d = hf.parse(dd + "14:20");
ev.setDate(d);
ev.setDevice_id(pastiller.getId());
ev.setType(Event.USED_OBJECT);
saveEvent(ev);

d = hf.parse(dd + "15:30");
ev.setDate(d);
ev.setDevice_id(menjador.getId());
ev.setType(Event.AT_ROOM);
saveEvent(ev);

d = hf.parse(dd + "18:00");
ev.setDate(d);
ev.setDevice_id(cuina.getId());
ev.setType(Event.AT_ROOM);
saveEvent(ev);

d = hf.parse(dd + "20:00");
ev.setDate(d);
ev.setDevice_id(menjador.getId());
ev.setType(Event.AT_ROOM);
saveEvent(ev);

} catch (Exception e) {
    e.printStackTrace();
}
}

```

Codi font 15 - Rutina per generar esdeveniments fixes d'un dia

D.1 Horaris fixes

Es genera una funció que crea per cada dia el següents horaris:

Cuina	09:00	Cuina	14:06	Cuina	1h+1h 24'+2h = 4h 24'
Pastiller	09:10	Pastiller	14:16	Fora casa	2h
Menjador	10:00	Menjador	15:30	Menjador	2h + 2h 30' + 13h
Surt de casa	12:00	Cuina	18:00		
Entra a casa	14:00	Menjador	20:00		

Si analitzem el patró calculat, podem veure que s'han creat 35 registres que corresponen a 7 dies de la setmana per 4 dispositius, però el sensor de presència crea un registre dintre de casa i un altre fora de casa, per tant serien 7 dies setmana x 5 = 35.

Si analitzem el dispositiu 1 (detector beacon presència) i tipus 6 (fora de casa) podem veure que té un valor 7200 segons, que corresponen a 2h.

També podem apreciar que el dispositiu 2 (pastiller) es fa servir 2 cops i que tant la cuina com el menjador s'utilitzen en 3 ocasions.

```
mysql> select * from profile order by device_id;
```

device_id	day_of_week	type	num_samples	time	count	min_time	max_time	min_count	max_count
1	dom	5	48	79200	1	79200	79200	1	1
1	dom	6	49	7200	1	7200	7200	1	1
1	jue	5	49	79200	1	79200	79200	1	1
1	jue	6	49	7200	1	7200	7200	1	1
1	lun	5	48	79200	1	79200	79200	1	1
1	lun	6	48	7200	1	7200	7200	1	1
1	mar	5	49	79200	1	79200	79200	1	1
1	mar	6	49	7200	1	7200	7200	1	1
1	mié	5	49	79200	1	79200	79200	1	1
1	mié	6	49	7200	1	7200	7200	1	1
1	sáb	5	49	79200	1	79200	79200	1	1
1	sáb	6	49	7200	1	7200	7200	1	1
1	vie	5	49	79200	1	79200	79200	1	1
1	vie	6	49	7200	1	7200	7200	1	1
2	dom	1	49	0	2	0	0	2	2
2	jue	1	49	0	2	0	0	2	2
2	lun	1	48	0	2	0	0	2	2
2	mar	1	49	0	2	0	0	2	2
2	mié	1	49	0	2	0	0	2	2
2	sáb	1	49	0	2	0	0	2	2
2	vie	1	49	0	2	0	0	2	2
15	dom	2	49	62044	3	59590	63000	3	3
15	jue	2	49	63000	3	63000	63000	3	3
15	lun	2	48	63000	3	63000	63000	3	3
15	mar	2	49	63000	3	63000	63000	3	3
15	mié	2	49	63000	3	63000	63000	3	3
15	sáb	2	49	63000	3	63000	63000	3	3
15	vie	2	49	63000	3	63000	63000	3	3
42	dom	2	49	15600	3	15600	15600	3	3
42	jue	2	49	15600	3	15600	15600	3	3
42	lun	2	48	15600	3	15600	15600	3	3
42	mar	2	49	15600	3	15600	15600	3	3
42	mié	2	49	15600	3	15600	15600	3	3
42	sáb	2	49	15600	3	15600	15600	3	3
42	vie	2	49	15600	3	15600	15600	3	3

35 rows in set (0.00 sec)

El dispositiu 15 que pertany al menjador té un temps de 63000 que correspon a 17h 30', l'excepció el diumenge be motivada per que es generen esdeveniments fins el diumenge 8/12 a las 20h, per la qual cosa es redueix el temps. El càlcul de la duració del temps de permanència en una estança, no es talla a les 00:00 si no que es manté fins el següent esdeveniment de presència. En aquest cas és l'estança al menjador des de les 20:00 fins a les 09:00 hores del dia següent que entra a la cuina.

Un cop s'ha comprovat que el funcionament de l'algorisme és correcte, s'esborren els esdeveniments fixos, i es generen esdeveniments amb petites variacions aleatòries que generen un patró molt més real. Ja que encara que les persones acostumen a fer uns horaris bastant fixos, sempre hi ha variacions d'uns pocs minuts.

```
mysql> select * from device_event limit 34;
```

id	date	device_id	event_type_id	audit
39157	2019-01-01 09:00:00	42	2	1
39158	2019-01-01 09:10:00	2	1	0
39159	2019-01-01 09:10:10	15	2	1
39160	2019-01-01 12:02:29	1	6	1
39161	2019-01-01 14:06:33	1	5	1
39162	2019-01-01 14:16:33	42	2	1
39163	2019-01-01 14:26:33	2	1	0
39164	2019-01-01 15:31:17	15	2	1
39165	2019-01-01 18:01:02	42	2	1
39166	2019-01-01 20:06:12	15	2	1
39167	2019-01-02 00:00:00	42	2	1
39168	2019-01-02 09:10:00	2	1	0
39169	2019-01-02 09:19:44	15	2	1
39170	2019-01-02 12:00:20	1	6	1
39171	2019-01-02 14:03:22	1	5	1
39172	2019-01-02 14:13:22	42	2	1
39173	2019-01-02 14:23:22	2	1	0
39174	2019-01-02 15:36:59	15	2	1
39175	2019-01-02 18:06:45	42	2	1
39176	2019-01-02 20:04:06	15	2	1
39177	2019-01-03 09:00:00	42	2	1
39178	2019-01-03 09:10:00	2	1	0
39179	2019-01-03 09:12:16	15	2	1
39180	2019-01-03 12:03:41	1	6	1
39181	2019-01-03 14:00:04	1	5	1
39182	2019-01-03 14:10:04	42	2	1
39183	2019-01-03 14:20:04	2	1	0
39184	2019-01-03 15:34:22	15	2	1
39185	2019-01-03 18:03:57	42	2	1
39186	2019-01-03 20:00:56	15	2	1
39187	2019-01-04 09:00:00	42	2	1
39188	2019-01-04 09:10:00	2	1	0
39189	2019-01-04 09:16:29	15	2	1
39190	2019-01-04 12:06:13	1	6	1

34 rows in set (0.00 sec)

Il·lustració 51 - Registres generats aleatòriament

Es pot apreciar a la imatge, que es generen esdeveniment amb petites variacions, per exemple l'entrada a casa és a las 14:06 o les 14:03, etc.

Un cop calculats els patrons, veiem que ara el mínim i el màxim no tenen el mateix valor, la qual cosa indica que el temps ja no és fixe.

```
mysql> select * from profile order by device_id;
```

device_id	day_of_week	type	num_samples	time	count	min_time	max_time	min_count	max_count
1	dom	5	48	79185	1	79111	79259	1	1
1	dom	6	49	7176	1	7092	7261	1	1
1	jue	5	49	79219	1	79115	79325	1	1
1	jue	6	49	7181	1	7077	7285	1	1
1	lun	5	48	79230	1	79139	79321	1	1
1	lun	6	48	7209	1	7132	7288	1	1
1	mar	5	49	79229	1	79133	79325	1	1
1	mar	6	49	7164	1	7086	7243	1	1
1	mié	5	49	79263	1	79173	79355	1	1
1	mié	6	49	7123	1	7025	7222	1	1
1	sáb	5	49	79206	1	79113	79300	1	1
1	sáb	6	49	7255	1	7157	7354	1	1
1	vie	5	49	79181	1	79107	79257	1	1
1	vie	6	49	7176	1	7090	7263	1	1
2	dom	1	49	0	2	0	0	2	2
2	jue	1	49	0	2	0	0	2	2
2	lun	1	48	0	2	0	0	2	2
2	mar	1	49	0	2	0	0	2	2
2	mié	1	49	0	2	0	0	2	2
2	sáb	1	49	0	2	0	0	2	2
2	vie	1	49	0	2	0	0	2	2
15	dom	2	49	64748	2	62331	66654	2	2
15	jue	2	49	65668	3	65529	65808	3	3
15	lun	2	48	65640	3	65515	65766	3	3
15	mar	2	49	65699	3	65578	65820	3	3
15	mié	2	49	65664	3	65553	65776	3	3
15	sáb	2	49	65645	3	65502	65789	3	3
15	vie	2	49	65855	3	65707	66004	3	3
42	dom	2	49	12923	3	12771	13076	3	3
42	jue	2	49	12950	3	12818	13083	3	3
42	lun	2	48	12949	3	12826	13073	3	3
42	mar	2	49	12936	3	12817	13056	3	3
42	mié	2	49	13011	3	12903	13120	3	3
42	sáb	2	49	12998	3	12753	13045	3	3
42	vie	2	49	12768	3	12616	12921	3	3

35 rows in set (0.00 sec)

II-lustració 52 - Esdeveniments generats aleatoriament

Per exemple el primer registre indica que els diumenges, hi es a casa (tipus 5) una mitja de 79185 segons (21:59:45) i pot variar entre 21:58:31 i 22:00:59. Si es produeixen variacions més importants, el marge del mínim i màxim serà més ampli.

La taula profile permet fer una cerca immediata del patró per cada esdeveniment, només cal fer una cerca per device_id, day_of_week i tipus, i retornarà el mínim i màxim de cops que es fa servir l'estança o l'objecte, i el temps mínim i màxim.

Un exemple de control del patró el podem trobar a la classe OvertimeController.java, en la qual es controla si el sènior es troba massa temps a una estança, o no ha sortit de casa o està massa temps fora de casa.

```
device = DeviceDAO.getDevice(s.getDevice_id());
profile = ProfileDAO.getProfile(s.getDevice_id(), new Date(), s.getEvent_type());
// Compare with the pattern
if (profile != null && !profile.isTimeOK(s.getDuration())) {
    // Create an event to indicate to family and operators
    Event event = new Event();
    event.setInstallation_id(device.getInstallationId());
    event.setDate(new Date());
    event.setDetail("Està més temps a casa de l'habitual");
    event.setType(Event.UNUSUAL_TIME);
    event.setIcon(Event.ICON_UNUSUAL_TIME_IN_HOME);
    event.setTime(s.getDuration());
    EventDAO.saveEvent(event);
    // mark event as managed
    // Close the event, this method must be called inside the method which creates the event
    DeviceEventDAO.closeAuditedDevice(null, s.getId());
}
```

Codi font 16 - Exemple control del patró

Com es pot veure en el codi, es consulta el patró, i es compara amb la funció isTimeOK el temps del patró amb el que porta el sènior, si és superior la funció retorna true i s'emmagatzemarà un esdeveniment per alertar als familiars.