



**Universitat Oberta
de Catalunya**

MyHAL App

Jaume Barnés Cerrato
Enginyeria Tècnica Informàtica de Gestió
Desenvolupament d'aplicacions mòbils

Consultors

Robert Clarisó Viladrosa
Jordi Almirall López
Marc Domingo Prieto



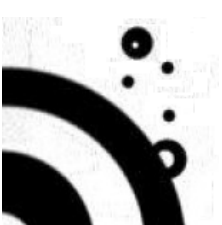


Contingut

1. Descripció del projecte.....	6
2. Introducció	7
2.1 Justificació.....	7
2.2 Objectius	7
Fase 1:	7
Fase 2:	8
2.3. Funcionalitats	8
2.4. Productes obtinguts	8
2.5. Mètode de desenvolupament	9
3 Planificació.....	10
3.1 Planificació de les tasques.....	10
3.2 Descripció de les Tasques	10
Tasca 1:	10
Tasca 2:	11
Tasca 3:	11
Tasca 4:	11
4. Anàlisi de requisit.....	12
4.1 Requisits del usuari	12
5. Disseny conceptual.....	13
5.1 Contexts d'ús.....	13
5.2 Definició dels casos d'ús.....	13
5.3 Descripció dels perfils d'usuari.....	14
5.4 Escenaris	15
5.5 Fluxos d'interacció.....	16
5.6 Diagrama de flux de l'aplicació	18
5.7 Diagrama de casos d'ús.....	19
6. Prototip.....	20
7. Test d'usabilitat.....	26
7.1 Anàlisi dels tests	32
7.2 Conclusions	32
7.3 Recomanacions d'usabilitat.....	32
8. Disseny.....	33
8.1 Arquitectura i comunicació de la App i la llar	33



8.2	Arquitectura del servei web.....	34
8.3	Arquitectura bàsica d'aplicacions Android.....	34
8.4	Arquitectura d'Arduino.....	35
8.5	Disseny de Base de Dades.....	36
9	Control de versions.....	38
10	Decisions APP.....	39
10.1	Preferències.....	40
10.2	Idioma.....	41
10.3	Menú superior.....	42
10.4	SOAP.....	42
11	Servei Web.....	43
11.1	SQLite.....	44
11.1.1	Implementació.....	44
11.2	Servei SOAP.....	45
11.3	Tasques.....	47
12	Arduino.....	48
12.1	Lectura codis IR.....	49
12.2	Implementació Servidor Arduino.....	49
12.3	Consulta de temperatura.....	51
12.4	Enviament codi Infraroig.....	52
13	Punts de millora.....	52
14	Conclusions.....	53
15	Bibliografia.....	54



1. Descripció del projecte

En l'actualitat l'ús de la tecnologia permet que les llars siguin administrades i organitzades amb facilitat des de la distància gràcies a la domòtica.

La domòtica és una tecnologia que permet interactuar amb un habitatge, a través d'un dispositiu mòbil. Els habitatges automatitzats ofereixen realitzar diferents activitats sense que el resident estigui present en la llar:

- Regular la calefacció.
- Tancament de persianes.
- Encendre i apagar llums.
- Activar i desactivar Alarmes.

Existeixen diversos canals de comunicació amb els diferents dispositius de la llar, en aquest cas utilitzarem la tecnologia d'infrarojos mitjançant un sistema amb un microprocessador anomenat Arduino.

Arduino és una plataforma de *hardware* lliure, basat en una tarjeta amb un micro controlador i un entorn de desenvolupament. Arduino ofereix la possibilitat d'afegir components i desenvolupar solucions que satisfacin qualsevol necessitat.

D'altra banda, la necessitat de comptar amb una aplicació mòbil per executar ordres via remota fa que el sistema operatiu Android sigui la solució més evident.

Android és un sistema operatiu pensat per a dispositius mòbils. Està basat en Linux, el nucli del qual de sistema és lliure, gratuït i multiplataforma. El sistema operatiu proporciona diferents eines necessàries per desenvolupar aplicacions que accedeixin a funcions del telèfon utilitzant el llenguatge JAVA.

Donada la naturalesa mòbil del propi sistema i dels dispositius en els quals s'instal·la cada vegada són més habituals les aplicacions que accedeixen a dades allotjades en el núvol. Normalment els dispositius no compten amb una gran capacitat de emmagatzematge a més que normalment l'usuari necessita accedir a l'aplicació des de diferents dispositius, per la qual cosa necessita obtenir les dades actualitzades al dispositiu actual. Aquest fet ha ocasionat que cada vegada més les aplicacions mòbils accedeixin a recursos en el núvol utilitzant serveis web (*web services*).

Un servei web no és més que un conjunt de protocols i estàndards que serveixen per intercanviar dades entre aplicacions. Les diferents aplicacions poden estar desenvolupades sota qualsevol plataforma i programades en un llenguatge diferent.

Per tant, l'objectiu d'aquest TFC serà desenvolupar una aplicació que permeti a l'usuari controlar dispositius de la llar utilitzant un dispositiu mòbil.

2. Introducció

2.1 Justificació

Sempre he cregut que en el futur totes les operacions del dia a dia que actualment fem es podan realitzar des d'un únic dispositiu. Fa relativament pocs anys el dispositiu que ara coneixem com a telèfon mòbil només servia per trucar o enviar missatges. És increïble que ara puguem consultar dades, realitzar pagaments en qualsevol supermercat o mesurar-nos en nivell de sucre en sang mitjançant un *gadget*. En definitiva el ventall de possibilitats és infinit.

El plantejament d'aquest treball final crec que ve de fa molt temps, quan era adolescent. El meu pare era electricista i com a bon electricista casa nostra estava plena d'aparells per a tot, des d'apagar o encendre llums quan estàvem fora de casa utilitzant un temporitzador (per espantar als lladres) fins a un sistema d'alarma bàsic creat per ell.

Ara de gran, amb una mica de visió de les possibilitats que ofereixen les noves eines que existeixen en l'actualitat, com per exemple les plaques d'Arduino, la meua imaginació es va disparar. Podia realitzar els invents del meu pare i arribar un pas més enllà.

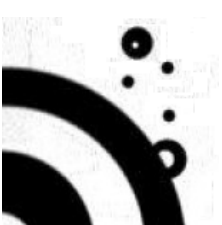
D'altra banda, com a conseqüència del camí que està prenent el món del desenvolupament sé que treballar amb Android i sistemes en el núvol és cada vegada més comú, per la qual cosa realitzar el projecte d'aquesta temàtica em permetrà afermar més el meu coneixement professional.

2.2 Objectius

L'objectiu principal d'aquest projecte és el d'obtenir coneixements detallats del sistema operatiu Android i de compondre un sistema de comunicació entre el dispositiu Android i Arduino. Per a això es realitzarà un aplicatiu Android que comunicarà amb un servei web el qual oferirà diferents opcions que comunicaran amb el dispositiu Arduino.

Fase 1:

- Estudi del sistema operatiu Android i de les característiques generals dels dispositius mòbils que ho incorporen.
- Estudi de les diferents solucions web de comunicació entre sistemes.
- Estudi del component Arduino per satisfer les necessitats del projecte.



Fase 2:

- Desenvolupament d'un aplicatiu Java para Android que consumi un servei web.
- Desenvolupament d'un web service que comuniqui els dos sistemes.
- Desenvolupament d'un aplicatiu Arduino que incorpori les funcionalitats acordades.

2.3. Funcionalitats

L'objectiu del projecte es divideix en el disseny de diverses parts i la seva interconnexió. Les diferents parts i les seves funcionalitats seran:

- **Aplicació Android:** Es desenvoluparà una App que es comunicarà amb un servei web i permetrà realitzar les següents accions:
 - Autenticació d'usuari mitjançant credencials.
 - Consulta temperatura llar.
 - Encès/Apagat aire condicionat llar.
 - Encès/Apagat llums.
 - Encès/Apagat alarma.
- **Servei Web:** Es desenvoluparà un servei web que permetrà la interconnexió entre la App i la tarjeta Arduino.
- **Tarjeta Arduino:** S'ha de desenvolupar tota la lògica per a l'obtenció de les dades dels diferents components de la tarjeta.

2.4. Productes obtinguts

Els productes obtinguts seran principalment dos, **l'aplicació** i la **documentació de la memòria**. A més d'aquests, es lliuraran altres productes secundaris. A continuació veurem una descripció de tots els productes obtinguts:

- **Aplicació:** Producte final del projecte. S'adjuntarà el l'instal·lable de l'aplicació android (apk) i un fitxer comprimit zip amb el servei web i el codi d'Arduino. A més es lliurarà el codi font del lliurament final, tant de la App com del Servei Web i Arduino. No obstant això s'adjuntarà un enllaç amb el repositori del codi font utilitzat.
- **Memòria / documentació:** Consisteix d'una banda aquest document com a memòria del projecte i d'altre les diferents PACS. També es lliurarà l'arxiu de pla de projecte i els diferents manuals tant per configurar el servei web, la App i Arduino. La distribució de la documentació serà la següent:
 - **Links:** Document amb tots els links utilitzats durant el projecte.
 - **Manuals:** Manuals d'instal·lació, configuració i ús dels diferents sistemes.
 - **PACS:** Les 4 *pacs de les quals consten el projecte.
 - **Presentació:** Presentació final del projecte en format *power *paint.

- **Pla de Projecte:** Pla de projecte
- **Videos:** Videos realitzats durant la pràctica i que acompanyarà el lliurament final.

2.5. Mètode de desenvolupament

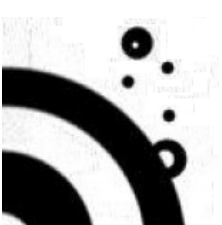
El mètode de desenvolupament utilitzat serà utilitzant el paradigma d'orientació objectes (OO) per a l'aplicació Android i el Web Service.

Per a la intercomunicació dels sistemes utilitzant un servei web em basaré en el protocol estàndard SOAP i serà desenvolupat sota tecnologia .NET sota el llenguatge C#.

Per al desenvolupament de la solució de Arduino utilitzaré el propi framework que ofereix Arduino el qual desenvolupa baix llenguatge C++.

Els objectius són:

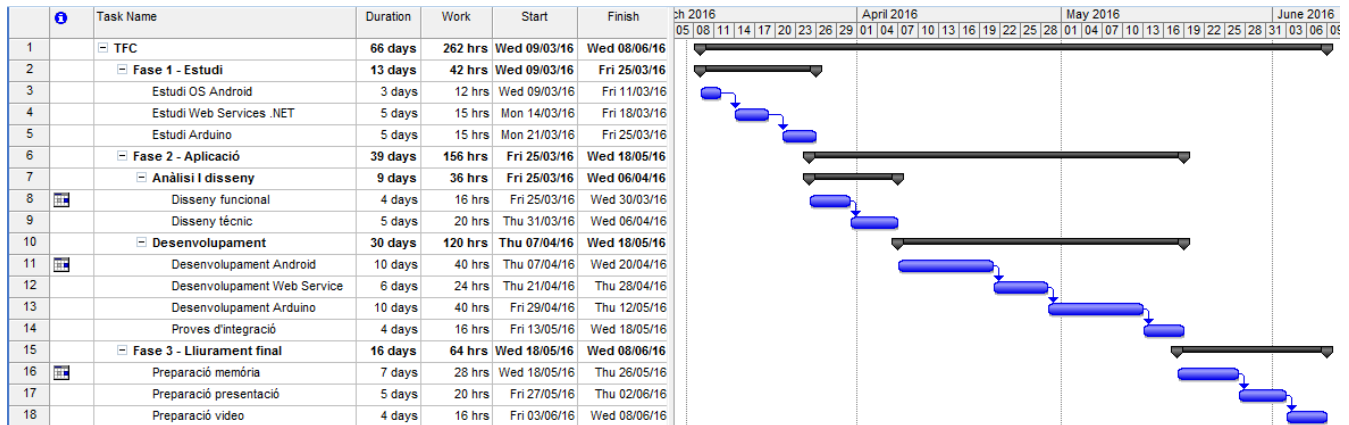
- Conèixer el funcionament del llenguatge Java para Android i les llibreries útils per connectar amb serveis web.
- Conèixer el funcionament de les plaques Arduino centrant-me en les funcionalitats requerides.
- Aprofundir en la metodologia de desenvolupament d'aplicacions Android.
- Aprofundir en la metodologia de desenvolupament de serveis web i comunicació entre sistemes.



3 Planificació

3.1 Planificació de les tasques

A continuació el diagrama de Gantt mostra una aproximació a la planificació del projecte. El calendari laboral ha sigut adaptat per a una jornada laboral de 4h de dilluns a divendres, que pertany a la mitja de hores setmanals que podré dedicar al projecte.



Com es pot apreciar en la imatge anterior, el projecte es desenvoluparà per una persona pel que no hauran tasques que es solapin en el temps ni que es duguin a terme de forma paral·lela.

El projecte es divideix en tres grans blocs que componen a gran escala el desenvolupament d'una aplicació. A continuació passaré a descriure amb més profunditat les diferents tasques a realitzar.

3.2 Descripció de les Tasques

Tasca 1:

- **Temporització:** 3 setmanes (9 de Març al 25 Març)
- **Descripció:** Recollida i classificació de tota la informació referent als diferents components en els quals es basa en TFC:
 - Temes relacionats amb Android.
 - Temes relacionats amb el desenvolupament del Servei Web.
 - Temes relacionats amb Arduino.
 - Temes relacionat amb la interconnexió dels diferents sistemes.

- **Objectius:**
 - Tenir una visió clara dels diferents sistemes i la connexió entre ells.
 - Adquirir els coneixements necessaris per al desenvolupament.

Tasca 2:

- **Temporització:** 2 setmanes (25 Març al 06 abril)
- **Descripció:** Anàlisi i disseny de l'aplicació
- **Objectius:**
 - Establir els requisits de l'aplicació.
 - Compondre el disseny final (*mockup*) de les diferents pantalles de l'aplicació.
 - Analitzar i dissenyar cadascuna de les parts del sistema, emfatitzant en una composició escalable i utilitzant patrons estàndards.
 - Prendre les decisions necessàries per al posterior desenvolupament.
 - Generar la documentació tècnica de tot el sistema.

Tasca 3:

- **Temporització:** 6 setmanes (06 Abril al 18 Maig)
- **Descripció:** Implementació
- **Objectius:**
 - Implementar el codi font de l'aplicació.
 - Generar jocs de proves necessaris per garantir el correcte funcionament.
 - Generar documentació sobre el procés de desenvolupament.

Tasca 4:

- **Temporització:** 3 setmanes (18 Maig al 08 Juny)
- **Descripció:** Lliurament final
- **Objectius:**
 - Finalització de la memòria del TFC.
 - Generar presentació de l'aplicació
 - Generar els manuals necessaris per a la correcta configuració i posterior ús de l'aplicació.
 - Crear video de presentació.



4. Anàlisi de requisit

En aquest apartat veurem tot el relacionat amb el disseny de la nostra aplicació per controlar de forma remota els diferents aparells de la nostra llar. Parlarem de l'arquitectura, del flux del producte i de les classes que ho conformessin. A més realitzarem una recerca amb tal de revelar els detalls i motivacions de les persones enfront de l'aplicació i així oferir una aplicació amb el màxim de requisits possible.

Amb motiu que l'aplicació que volem desenvolupar és molt difícil d'aconseguir, ja que els productes oposats requereixen de maquinari de suport, no m'és possible realitzar una recerca dels diferents productes del mercat i de com l'usuari es comporta amb la seva utilització. És per això que, en ser una aplicació nova he decidit començar amb unes funcionalitats bàsiques i finalment realitzar la recerca contextual amb el meu prototip. D'aquesta manera, podré veure com l'usuari es comporta amb l'aplicació que estem desenvolupant i així podré ajustar el màxim possible a les demandes dels aquests.

4.1 Requisits del usuari

És necessari que l'usuari disposi d'un dispositiu mòbil amb sistema operatiu Android. A més el mòbil, com a mínim, ha de tenir connexió a Internet, ja sigui per Wifi o 3G ja que serà a través d'Internet on l'aplicació obtindrà la informació.

Perquè el mòbil pugui connectar-se amb la placa de Arduino, l'usuari haurà de disposar d'un ordinador amb connexió a Internet.

La placa Arduino amb tots els seus components entrarà en el paquet de compra de l'aplicació.



5. Disseny conceptual

5.1 Contexts d'ús

Atès que l'aplicació està desenvolupada sota tecnologia mòbil i es vol fer una èmfasi al fet de controlar la llar des de fora de casa, aquesta aplicació no pot excloure cap lloc com a context d'ús.

Com a part de la recerca contextual es realitzen observacions de l'activitat amb dispositius mòbils de quatre subjectes. És necessari dir que l'usuari 3 pel seu desconeixement en aquest camp i al món de la informàtica en general, no ha realitzat la tasca i per això no apareix en l'observació.

5.2 Definició dels casos d'ús

Cas d'ús:	Donar-se d'alta com usuari a l'aplicació i configurar els paràmetres de connexió amb el Web Service
Identificador	CU1
Nom	Donar-se d'alta com usuari a l'aplicació i configurar els paràmetres de connexió amb el Web Service
Prioritat	Alta
Descripció	Iniciar l'aplicació i configurar els paràmetres de connexió segons les dades proporcionades per el proveïdor. Després registrar-se com usuari.
Actors	Usuari final
Precondicions	L'usuari ha de tenir una clau proporcionada pel proveïdor.
Iniciat per	Usuari final
Flux	Usuari inicia l'aplicació Accedeix a la configuració utilitzant la icona superior dreta. Usuari modifica els paràmetres de connexió. Usuari torna al menú principal Usuari accedeix a la finestra de registre Usuari introdueix les dades i Accepta Usuari torna al menú principal Usuari s'identifica
Post-Condicions	Accés al menú principal amb les diferents opcions
Notes	

Cas d'ús:	Consultar la temperatura de la llar.
Identificador	CU2
Nom	Consultar la temperatura de la llar.
Prioritat	Alta
Descripció	Un cop identificat, l'usuari ha de poder consultar la temperatura de la llar.
Actors	Usuari final

Precondicions	L'usuari ha de estar identificat a l'aplicació
Iniciat per	Usuari final
Flux	Usuari selecciona la opció de menú Temperatura Usuari consulta la temperatura de les diferents parts de la llar
Post-Condicions	Consulta de les diferents temperatures de la llar.
Notes	

Cas d'ús:	Encendre/Apagar Aire Condicionat.
Identificador	CU3
Nom	Encendre/Apagar Aire condicionat.
Prioritat	Alta
Descripció	Seleccionar un programa del Aire condicionat o desactivar-lo
Actors	Usuari final
Precondicions	Usuari està identificat
Iniciat per	Usuari final
Flux	Usuari accedeix a l'opció de menú Aire condicionat Usuari selecciona modificar Usuari selecciona un programa del Aire condicionat
Post-Condicions	Usuari consulta el programa modificat a la informació que apareix a la finestra de Aire condicionat.
Notes	

5.3 Descripció dels perfils d'usuari

Com he dit abans, aquest sistema ha de ser d'accés lliure així que davant el gran ventall de perfils que es poden seleccionar finalment he realitzat les tasques de recerca contextual i test d'usabilitat a 4 usuaris que englobin tots els perfils possibles.

Dades dels usuaris del test	
Usuari 1	
Edat	24
Gènere	Home
Estat civil	Solter
Ocupació	Estudiant
Relació amb l'alumne	Amic
Ús d'internet	Disposa de PC, Tablet i Smartphone. Familiaritzat amb aplicacions d'ofimàtica, missatgeria, xarxes socials, jocs i multimèdia. Habitualment utilitza pàgines de compra online i agències de viatges.
Usuari 2	
Edat	33

Gènere	Dona
Estat civil	Soltera
Ocupació	Treballadora
Relació amb l'alumne	Germana
Ús d'internet	Disposa de Mac, Tablet i Smartphone. Familiaritzada amb aplicacions d'ofimàtica, missatgeria i multimèdia. Utilitza una aplicació que connecta amb la alarma de casa. Només li deixa fer captures de la llar.

Usuari 3	
Edat	37
Gènere	Dona
Estat civil	Soltera amb fills
Ocupació	Treballador
Relació amb l'alumne	Amic
Ús d'internet	Disposa de Smartphone. Familiaritzada amb aplicacions de missatgeria i alguns jocs. Habitualment utilitza aplicacions de missatgeria.

Usuari 4	
Edat	62
Gènere	Home
Estat civil	Casat amb fills
Ocupació	Treballador
Relació amb l'alumne	Familiar
Ús d'internet	Disposa de PC i Smartphone. Familiaritzat amb aplicacions d'ofimàtica, missatgeria i multimèdia. Descarrega aplicacions d'edició de vídeo i música. Comprador habitual de comerç electrònic.

5.4 Escenaris

- Escenari 1 (CU1): Donar-se d'alta com usuari a l'aplicació i configurar els paràmetres de connexió amb el Web Service.



- Escenari 2 (CU2): Consultar la temperatura de la llar.
- Escenari 3 (CU3): Encendre/Apagar l'Aire condicionat.

A continuació descriuré possibles escenaris per a les funcionalitats que estem desenvolupant.

Escenari 1 (CU1): Donar-se d'alta com usuari a la aplicació i configurar els paràmetres de connexió con el Web Service.

- L'usuari X vol poder controlar la seva llar tal com fa la seva dona des del mòbil.
- L'usuari X instal·la l'aplicació i l'arrenca. Busca l'opció Registrar-se.
- L'usuari X introdueix les dades requerides i espera la confirmació de l'Administrador de la llar per E-mail.
- L'usuari rep la validació, obre l'aplicació i s'identifica.

Escenari 2 (CU2): Consultar la temperatura de la llar.

- L'usuari X ha deixat a la mascota a casa i està preocupat perquè al ser Agost i tenir 30 graus de temperatura, la seva mascota ho estigui passant malament.
- L'usuari X arrenca l'aplicació i s'identifica.
- L'usuari X cerca l'opció de consultar la temperatura de la llar, al veure que és molt alta truca a un familiar perquè atengui a la seva mascota.

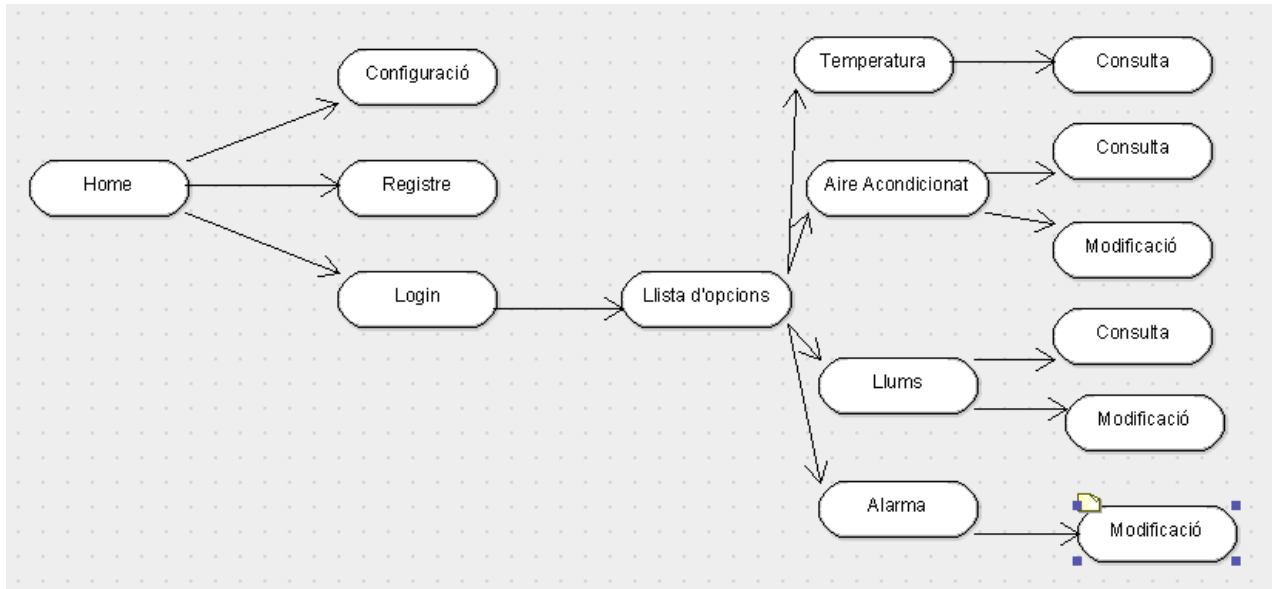
Escenari 3 (CU3): Encendre/Apagar Aire condicionat.

- L'usuari X està fora de casa i vol encendre l'aire condicionat per que quan arribi a la llar estigui a una temperatura agradable
- L'usuari X arrenca l'aplicació i s'identifica
- L'usuari X accedeix a la opció de menú Aire Condicionat i comprova que l'aire no està encès.
- L'usuari selecciona la opció Modificar
- L'usuari selecciona una de les opcions predefinides.
- Al arriba a casa la temperatura de dintre de la llar es molt agradable.

5.5 Fluxos d'interacció

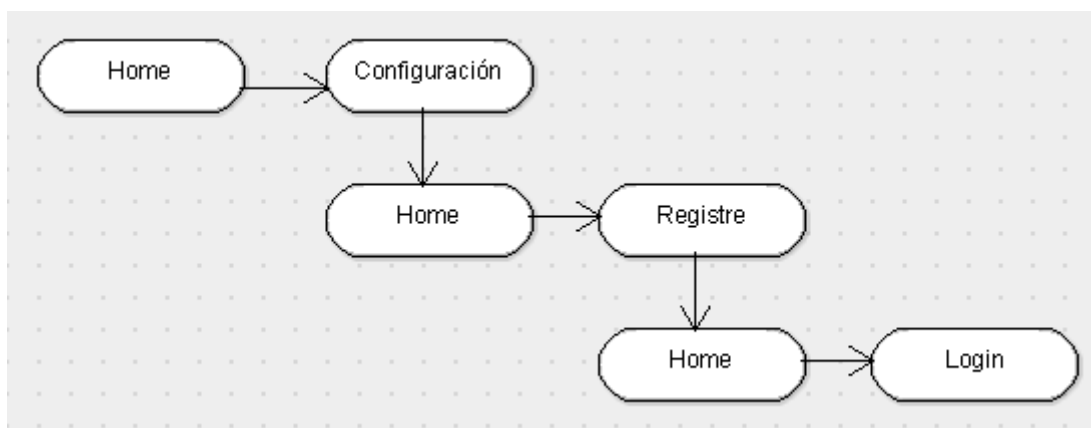


La figura 1 mostra el que seria el flux d'interacció per a qualsevol funcionalitat de l'aplicació. Totes les tasques començant des de la pantalla de Login. Un usuari no podria realitzar cap acció si no està registrat a l'aplicació.



Un cop identificat l'aplicació li permetrà accedir a un menú d'opcions en el qual l'usuari pot configurar les dades de connexió i utilitzar les diferents opcions que li permetran interaccionar amb la seva llar.

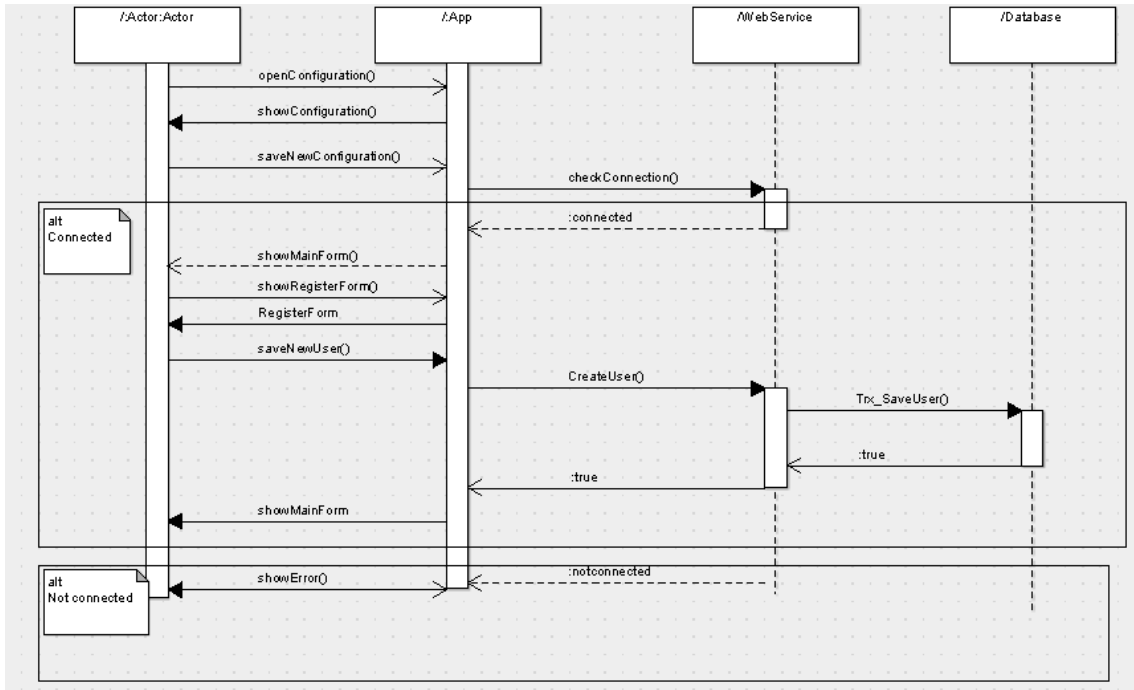
Entre les diferents opcions l'usuari podrà controlar i consultar la temperatura, controlar l'Aire condicionat, la llum de la seva llar o l'alarma. Cada opció tindrà la seva pantalla de consulta en la qual apareixeran les dades referents a la secció que estigui consultant i diverses opcions per interactuar amb la seva llar.



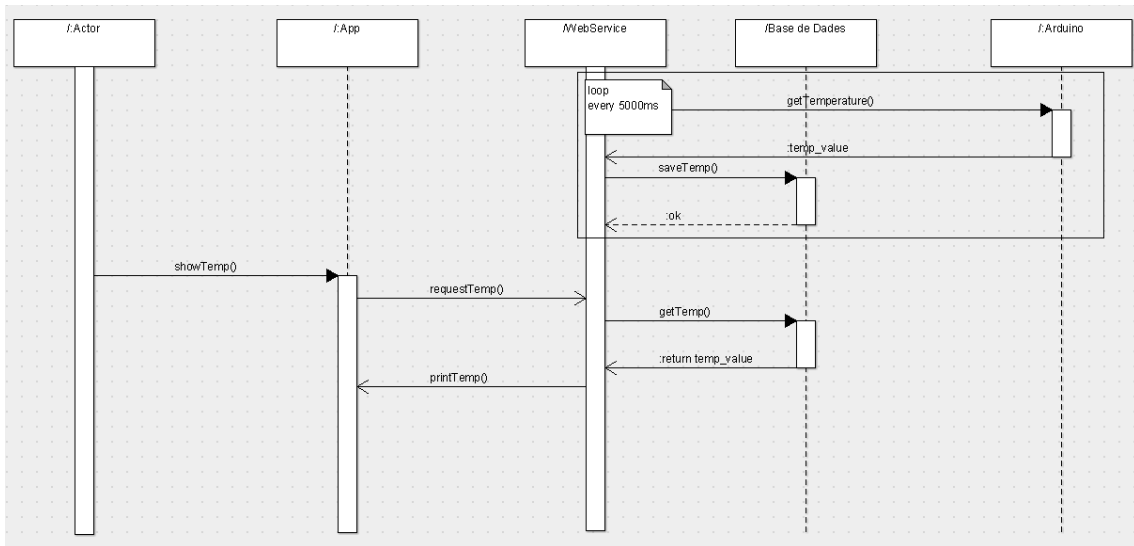
La figura 2 correspon al flux d'interacció pel registre del usuari. Des de la finestra d'inici de l'aplicació podrà accedir al formulari de Registre. Un cop registrat l'aplicació tornarà al inici i l'usuari podrà identificar-se.

5.6 Diagrama de flux de l'aplicació

A continuació s'indiquen els diferents fluxos d'interacció pels diferents escenaris comentats abans.

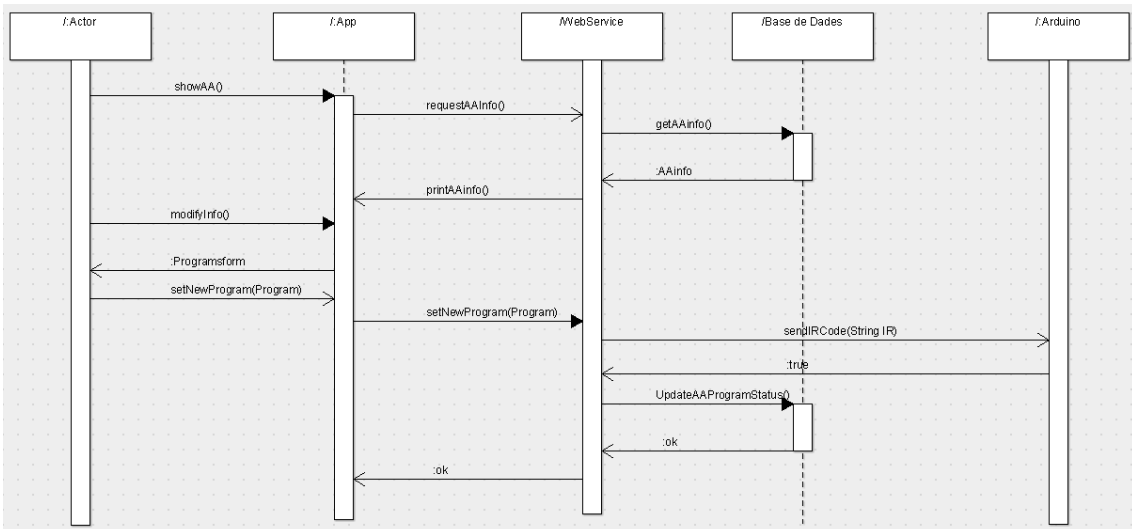


Flux d'interacció de CU1



Flux d'interacció de CU2

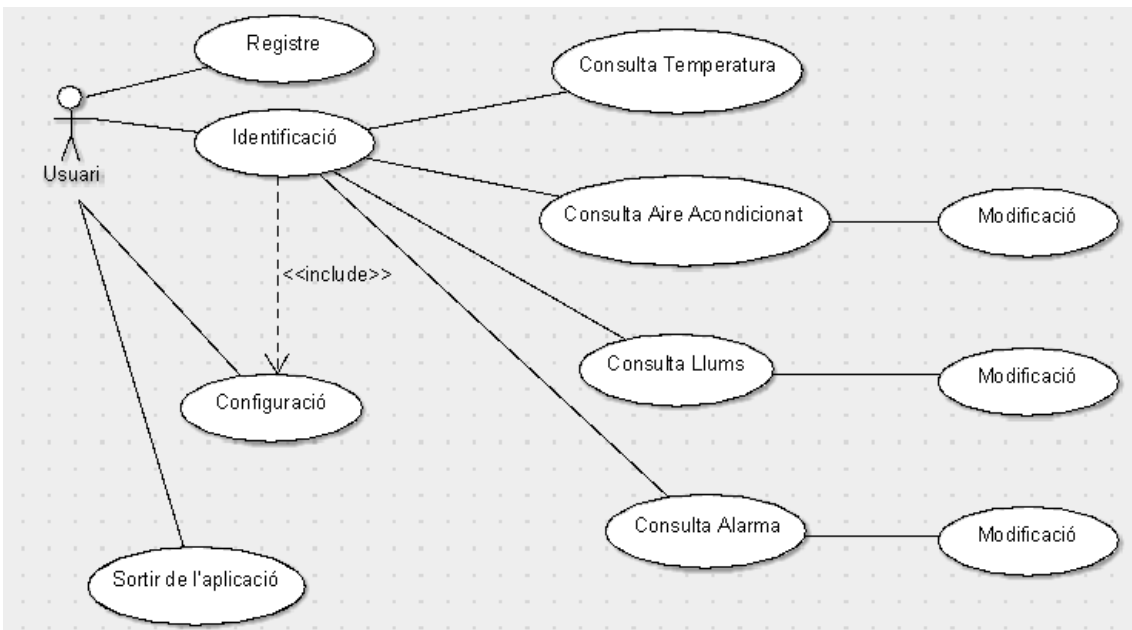




Flux d'interacció de CU3

5.7 Diagrama de casos d'ús

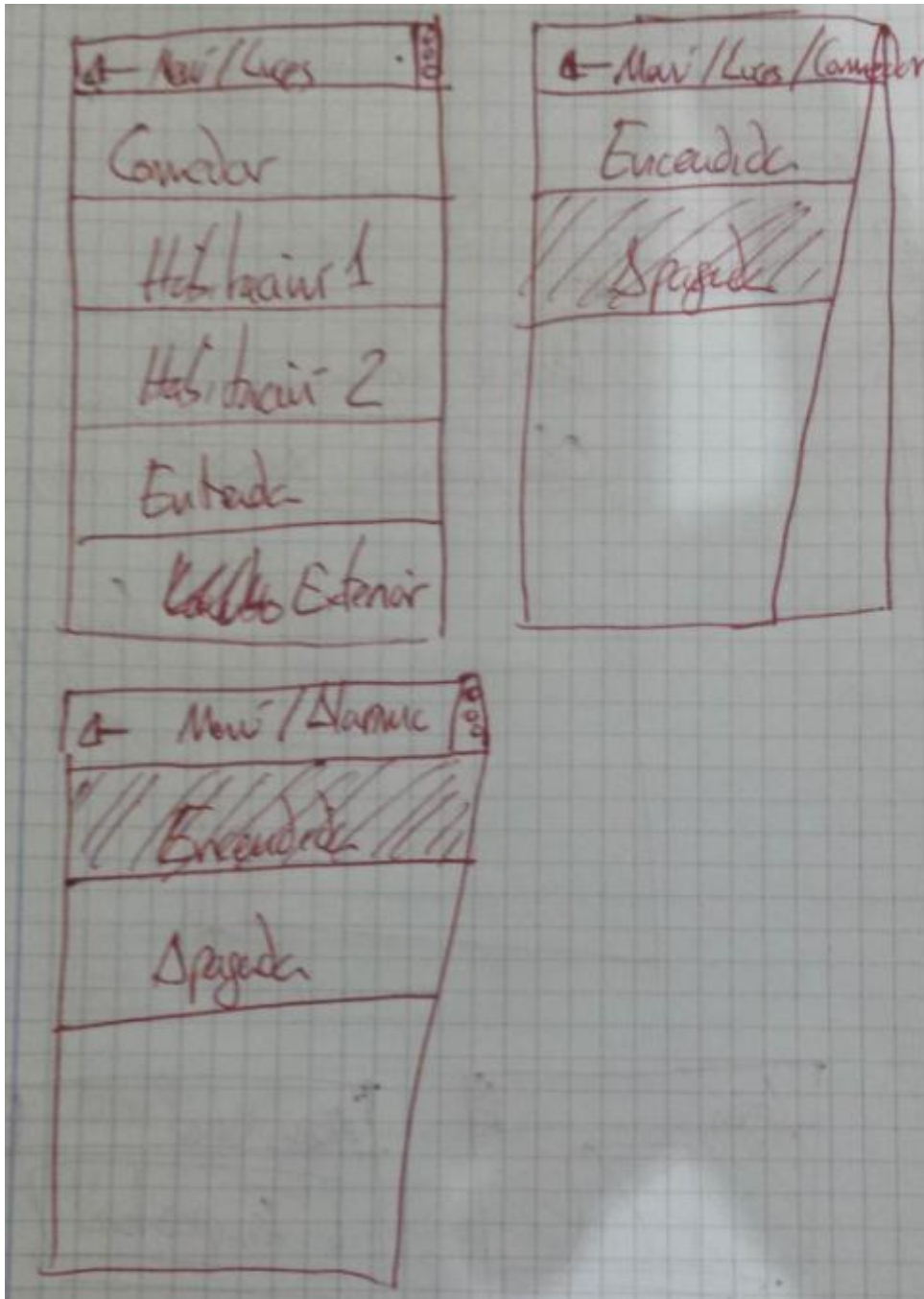
El següent diagrama recull una vista global del actor “usuari” i els casos d'ús que descriuen els requisits funcionals de l'aplicació:



6. Prototip

S'ha realitzat un esbós tenint en compte els criteris d'ús bàsics. És important destacar la barra superior amb el títol, la icona d'opcions i la fletxa per navegar a la finestra anterior.

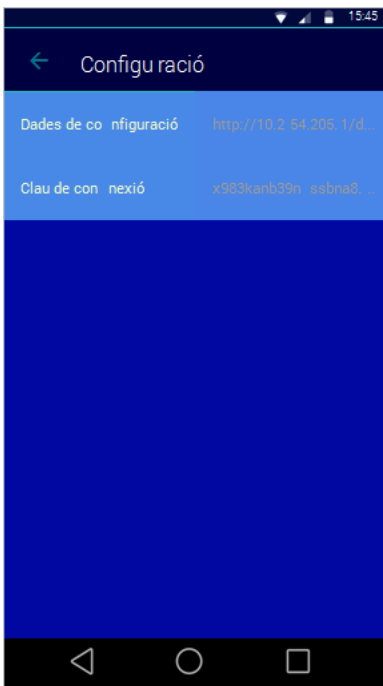




Finalment s'ha realitzat un prototip funcional real amb l'objectiu de poder testear amb els usuaris potencials les diferents pantalles de l'aplicació. El prototip s'ha realitzat amb l'eina de Justinmind, Justinmind Prototyper 7.1.0.



Pantalla principal

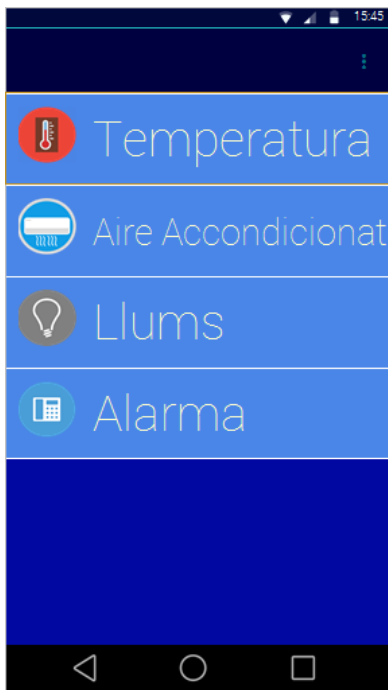


Pantalla de configuració





Pantalla de registre



Pantalla de Menú d'opcions

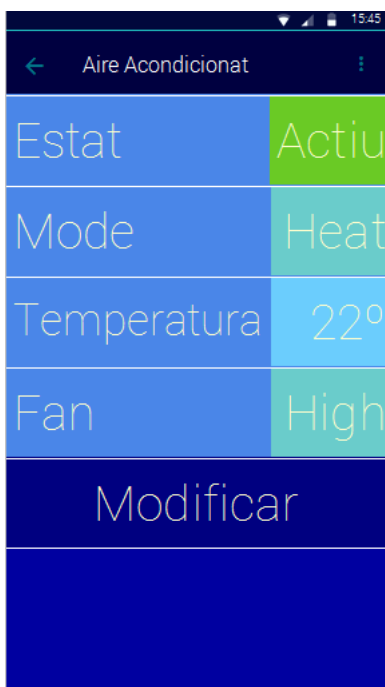




The screenshot shows a mobile application interface titled 'Temperatura'. It features a list of five items, each with a room name on the left and a temperature value on the right. The background is dark blue. The items are: Menjador (22°), Habitació 1 (22°), Habitació 2 (30°), Exterior (32°), and Lavabo (25°). The temperature values are displayed in green or red text on colored backgrounds. At the bottom, there is a dark blue bar with three white navigation icons: a back arrow, a circle, and a square.

Room	Temperature
Menjador	22°
Habitació 1	22°
Habitació 2	30°
Exterior	32°
Lavabo	25°

Pantalla de Consulta de Temperatura



The screenshot shows a mobile application interface titled 'Aire Acondicionat'. It features a list of five settings, each with a label on the left and a value on the right. The background is dark blue. The items are: Estat (Actiu), Mode (Heat), Temperatura (22°), Fan (High), and a 'Modificar' button. The values are displayed in green or light blue text on colored backgrounds. At the bottom, there is a dark blue bar with the word 'Modificar' in white text.

Setting	Value
Estat	Actiu
Mode	Heat
Temperatura	22°
Fan	High
Modificar	

Pantalla de consulta de Aire condicionat





Pantalla de Configuració d'Aire condicionat



Pantalla de Consulta d'estat de les llums





Pantalla de modificació de Llums



Pantalla de consulta/modificació d'Alarma

7. Test d'usabilitat

Gracies al prototip realitzat en l'apartat anterior podré realitzar un test d'usabilitat dels diferents Escenaris descrits a l'apartat 2.3.

Per realitzar el test d'usabilitat als quatre usuaris abans descrits he realitzat un qüestionari que

crec que recull les diferents parts importants d'usabilitat. Aquí descriu les diferents parts del qüestionari i el sistema de puntuació.

- **Localització de la tasca.**
Representa si la tasca ha estat fàcil de localitzar o l'usuari ha perdut temps buscant-la. Un valor molt alt indicarà que la tasca es de fàcil localització.
- **Autonomia per realitzar la tasca.**
Representa si l'usuari ha necessitat ajuda externa per realitzar la tasca. En cas de haver-la requerit, a quin nivell. Un valor molt baix d'autonomia representa una tasca massa complicada per l'usuari, en canvi un valor molt alt representa que qualsevol usuari pot realitzar-a sense ajuda.
- **Navegabilitat.**
Representa si l'usuari ha trobat que el flux de la tasca (finestres, icones, imatges) l'hi ha semblat agradable. Un valor alt indicarà que l'usuari s'ha trobat còmode realitzant la tasca.
- **Temps empleat.**
Representa si l'usuari ha trigat molt temps en realitzar la tasca o en canvi la ha realitzat ràpid. Un valor molt baix indicarà que la tasca es pot realitzar ràpidament.
- **Freqüència de la tasca.**
Representa si la tasca es realitza molts cops o es una tasca que es realitza només un cop.
- **Precisió en l'execució de la tasca.**
Representa si l'usuari ha realitzat els passos mínims per realitzar la tasca o en canvi, en algun moment s'ha perdut i ha hagut de tornar a l'inici i repetir algun pas. Un valor molt baix indicarà que la tasca té risc de fer que l'usuari faci o arribi a un altre tasca no desitjada, en canvi un valor molt alt indicarà que els passos estan visibles i es difícil equivocar-se.
- **Impressió general de la tasca.**
Opinió final de l'usuari sobre la totalitat de la tasca.



Per millorar la puntuació, s'ha afegit un tipus de puntuació en format text per facilitar als usuaris a l'hora de realitzar el test.

Finalment s'ha realitzar un qüestionari amb el següent format:

Localització de la tasca.

Valoració: Molt Bona/Bona/Normal/Baix/Molt Baix

Puntuació:_/10

Autonomia per realitzar la tasca.

Valoració: Molt Bona/Bona/Normal/Baix/Molt Baix

Puntuació:_/10

Navegabilitat.

Valoració: Molt Bona/Bona/Normal/Baix/Molt Baix

Puntuació:_/10

Temps empleat.

Valoració: Molt Alt/Alt/Normal/Baix/Molt Baix

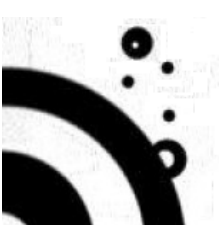
Puntuació:_/10

Freqüència de la tasca.

Valoració: Molt freqüent/Freqüent /Poc freqüent/Única

Puntuació:_/10

Precisió en l'execució de la tasca.



Valoració: Molt Bona/Bona/Normal/Baix/Molt Baix

Puntuació: _/10

Impressió general de la tasca.

Valoració: Molt Bona/Bona/Normal/Baix/Molt Baix

Puntuació: _/10

Test d'usabilitat CU1

Cas d'ús: Donar-se d'alta com usuari a l'aplicació i configurar els paràmetres de connexió amb el Web Service

Objectiu: Configurar els paràmetres de connexió i donar-se d'alta al sistema

Usuari 1

Localització de la tasca	Molt Bona	9
Autonomia per realitzar la tasca	Molt Bona	9
Navegabilitat	Molt Bona	9
Temps empleat	Baix	8
Freqüència de la tasca	Única	9
Precisió en l'execució de la tasca	Molt Bona	9
Impressió general de la tasca	Molt Bona	10

Usuari 2

Localització de la tasca	Molt Bona	9
Autonomia per realitzar la tasca	Molt Bona	10
Navegabilitat	Molt Bona	9
Temps empleat	Baix	9
Freqüència de la tasca	Única	9
Precisió en l'execució de la tasca	Molt Bona	10
Impressió general de la tasca	Molt Bona	10

Usuari 3

Localització de la tasca	Molt Bona	7
Autonomia per realitzar la tasca	Molt Bona	10
Navegabilitat	Molt Bona	10
Temps empleat	Baix	8
Freqüència de la tasca	Única	10
Precisió en l'execució de la tasca	Bona	6
Impressió general de la tasca	Molt Bona	8

Usuari 4

Localització de la tasca	Molt Bona	6
Autonomia per realitzar la tasca	Molt Bona	7
Navegabilitat	Molt Bona	10
Temps empleat	Alt	6
Freqüència de la tasca	Única	9
Precisió en l'execució de la tasca	Bona	7

Impressió general de la tasca	Bona	7
-------------------------------	------	---

Test d'usabilitat CU2

Cas d'ús: Consultar la temperatura de la llar.

Objectiu: Identificar-se a l'aplicació i poder consultar la temperatura de la llar

Usuari 1

Localització de la tasca	Molt Bona	10
Autonomia per realitzar la tasca	Molt Bona	10
Navegabilitat	Bona	8
Temps empleat	Molt Baix	10
Freqüència de la tasca	Molt freqüent	8
Precisió en l'execució de la tasca	Molt Bona	10
Impressió general de la tasca	Molt Bona	10

Usuari 2

Localització de la tasca	Molt Bona	10
Autonomia per realitzar la tasca	Molt Bona	10
Navegabilitat	Bona	9
Temps empleat	Baix	9
Freqüència de la tasca	Molt freqüent	8
Precisió en l'execució de la tasca	Molt Bona	9
Impressió general de la tasca	Molt Bona	9

Usuari 3

Localització de la tasca	Molt Bona	10
Autonomia per realitzar la tasca	Molt Bona	10
Navegabilitat	Bona	10
Temps empleat	Molt Baix	9
Freqüència de la tasca	Molt freqüent	8
Precisió en l'execució de la tasca	Bona	8
Impressió general de la tasca	Molt Bona	9

Usuari 4

Localització de la tasca	Molt Bona	9
Autonomia per realitzar la tasca	Molt Bona	9
Navegabilitat	Molt Bona	9
Temps empleat	Molt Baix	9
Freqüència de la tasca	Molt freqüent	9
Precisió en l'execució de la tasca	Molt Bona	9
Impressió general de la tasca	Molt Bona	9

Test d'usabilitat CU3

Cas d'ús: Encendre/Apagar Aire condicionat.

Objectiu: Seleccionar una opció per l'aire condicionat o desactivar-lo

Usuari 1

Localització de la tasca	Bona	10
--------------------------	------	----

Autonomia per realitzar la tasca	Molt Bona	9
Navegabilitat	Bona	10
Temps empleat	Baix	10
Freqüència de la tasca	Alta	8
Precisió en l'execució de la tasca	Molt Bona	9
Impressió general de la tasca	Bona	10
Usuari 2		
Localització de la tasca	Bona	9
Autonomia per realitzar la tasca	Bona	9
Navegabilitat	Bona	9
Temps empleat	Baix	7
Freqüència de la tasca	Alta	9
Precisió en l'execució de la tasca	Bona	9
Impressió general de la tasca	Bona	9
Usuari 3		
Localització de la tasca	Bona	8
Autonomia per realitzar la tasca	Bona	7
Navegabilitat	Bona	7
Temps empleat	Baix	8
Freqüència de la tasca	Alta	8
Precisió en l'execució de la tasca	Bona	8
Impressió general de la tasca	Bona	8
Usuari 4		
Localització de la tasca	Bona	8
Autonomia per realitzar la tasca	Bona	8
Navegabilitat	Molt Bona	9
Temps empleat	Mitjà	7
Freqüència de la tasca	Alta	9
Precisió en l'execució de la tasca	Bona	7
Impressió general de la tasca	Molt Bona	9

Total	CU1	CU2	CU3
Usuari 1	9,00	9,43	9,43
Usuari 2	9,43	9,14	8,71
Usuari 3	8,43	9,14	7,71
Usuari 4	7,43	9,00	8,14
Mitjana	8,57	9,18	8,50

D'aquest resultat podem extreure diferents anàlisis:

- La puntuació de CU2 es alta, ja que, els usuaris han trobat que la localització de la tasca es molt bona i té una gran facilitat de realitzar-la. Pot ser a causa de la claredat del menú principal.



- La puntuació de la CU1 és bona però no excel·lent. De fet, el cas del usuari 4 que té una puntuació molt baixa ens indica que per un usuari d'edat avançada es fa complicat el fet de localitzar la configuració i entendre el procés.
- La puntuació de CU3 és bona però no excel·lent. El fet que per característiques del producte ja estiguin definits els diferents programes de l'Aire condicionat ha despistat una mica als usuaris.

7.1 Anàlisi dels tests

- L'Usuari 1 troba que l'aplicació és molt directa però es queixa que el disseny és pobre.
- L'Usuari 4 agraeix que els botons i lletres siguin grans, la qual cosa facilita la seva lectura. No és sent molt còmode configurant les opcions o amb els programes preestablerts de l'Aire condicionat, ja que, opina que no són tant personalitzables com li agradaria.
- L'Usuari 2 i 3 opinen que el disseny de l'aplicació és bastant pobre, però funcionalment creu que li fa un bon servei.

7.2 Conclusions

- Tots els usuaris han tingut molta facilitat per a realitzar les funcions que els he fet realitzar. Només un usuari amb un nivell més baix que la resta en quan a treballar amb aplicacions mòbils ha tingut algun problema.
- És molt important realitzar una interfície intentant no carregar a l'usuari. Tots els usuaris han tingut facilitat per realitzar les modificacions als dispositius de la seva llar.
- La gran majoria de les queixes venen donades pel poc abast de la simulació o pels mecanismes de la mateixa.

7.3 Recomanacions d'usabilitat

- Seria convenient comptar amb un dissenyador gràfic a l'equip de desenvolupament, ja que les combinacions de color i l'estètica en general són molt importants.
- S'hauria de veure més clar que l'usuari està identificat, ja que ara no es veu en cap moment amb quin usuari estem identificats.
- Es podria millorar el procés de realitzar la configuració de l'aplicació per connectar-se amb el servei de la llar ja que per un usuari que no té coneixements informàtics és complicat d'entendre.

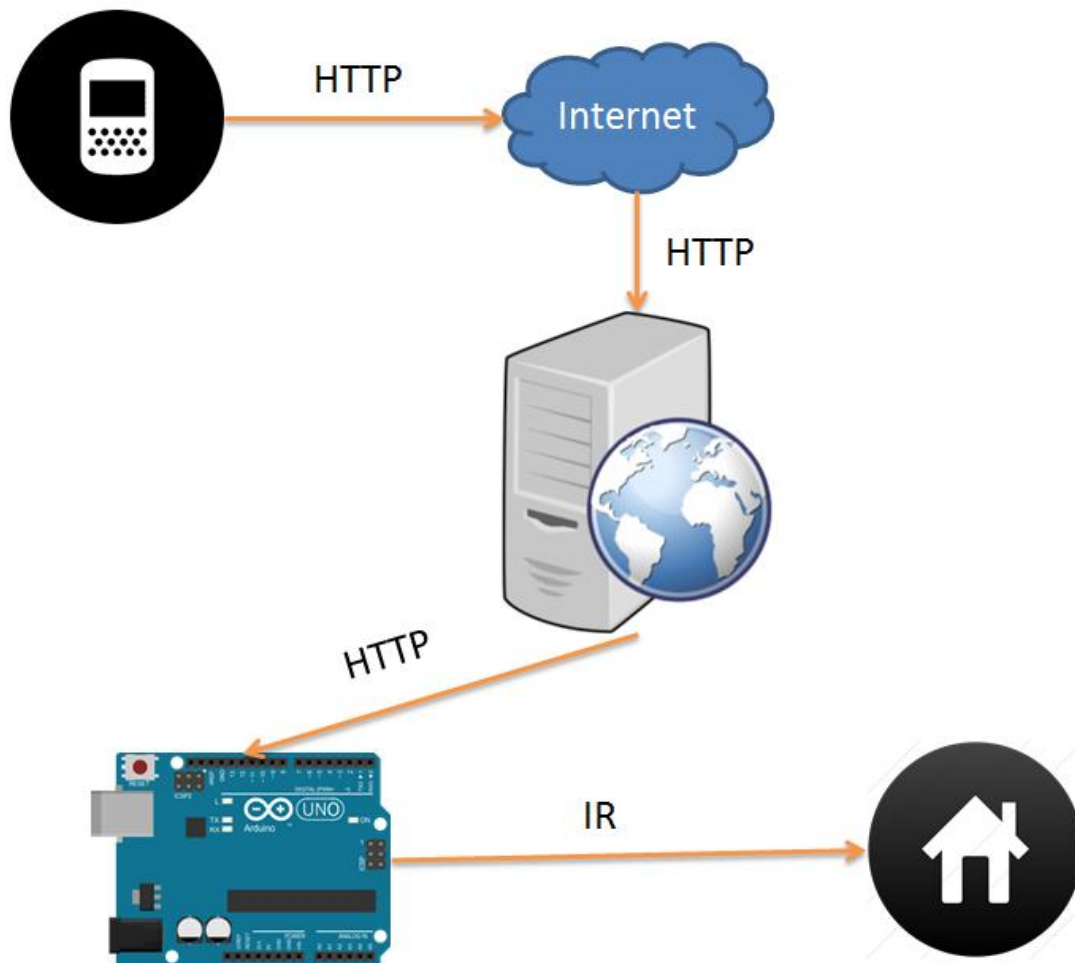


8. Disseny

8.1 Arquitectura i comunicació de la App i la llar

Els Web Services o Serveis web permeten suportar operacions i interaccions màquina a màquina a través d'una xarxa. Se'ls considera un conjunt de protocols i estàndards que serveixen per intercanviar informació de dades entre aplicacions.

L'arquitectura global de la informació funcionarà de la següent manera:



La il·lustració anterior correspon a la connexió dels diferents dispositius dintre del projecte.

Para una fase beta, el client haurà d'instal·lar i arrancar un servei en la màquina de la seva llar. Aquest servei s'encarregarà de realitzar consultes a la placa Arduino demanant informació dels diferents sensors i emmagatzemant-la.

D'altra banda, el servei també s'encarregarà de processar les peticions que es vagin realitzant des de l'aplicació Android.

A continuació entraré a detallar la implementació del servei web.

8.2 Arquitectura del servei web

Un servei web és un programa informàtic que processa una aplicació del costat del servidor, realitzant connexions bidireccionals i/o unidireccionals i síncrones o asíncrones amb el client i generant o cedint una resposta en qualsevol llenguatge. Per a la transmissió de totes aquestes dades se sol utilitzar un protocol, sent l'HTTP el més habitual.

Existeixen diverses maneres d'implementar tant el servei com la comunicació entre ells i per a això m'he basat en les dues arquitectures que més s'estan utilitzant en l'actualitat: REST i SOAP.

L'arquitectura REST correspon a exposar una API pública per manejar las operacions CRUD (create, read, update and delete) en les dades. D'aquesta manera, REST permet accedir als recursos a través d'una única interfície consistent.

L'arquitectura SOAP en canvi ofereix el seu propi protocol i que centra a exposar peces de la lògica de l'aplicació (no dades) com a servei. SOAP exposa operacions i se centra en l'accés a operacions cadascuna d'elles implementada amb una mica de lògica de negoci a través de diferents interfícies.

Donada la gran diferència entre les dues arquitectures he cregut convenient que el Web Service ofereixi SOAP perquè l'aplicació mòbil es comuniqui amb ell en canvi, donat el maquinari limitat que ens ofereix Arduino, s'haurà d'implementar una petita lògica imitant a SOAP en la qual el Web *Service enviarà peticions a la placa i aquesta respondrà en XML o JSON.

8.3 Arquitectura bàsica d'aplicacions Android

Una aplicació Android declara totes les seves activitats, punts d'entrada, interfícies de comunicació, permisos i intencions a través del AndroidManifest.xml.

Les aplicacions Android poden tenir 4 blocs principals en els quals es pot dividir:

- **Activity:** Representa una finestra amb una interfície d'usuari. És recomanable sobrecarregar els mètodes de “onCreate”, “onStop”, “onPause”, “onResume” de manera que es pugui gestionar de forma més precisa tot el flux de l'aplicació.
- **Services:** Es un procés llançat en segon pla de manera asíncrona i que no té cap interacció per part de l'usuari.
- **Content Providers:** Gestiona la informació compartida de la aplicació. Permet emmagatzemar les dades en un fitxer del sistema o en una base de dades SQLite.
- **Broadcast receivers:** És un component que es troba a l'espera de rebre Intents per a realitzar tasques específiques.

8.4 Arquitectura d'Arduino

La placa Arduino estarà connectada a la xarxa local permetent la comunicació amb el servei web. D'aquesta manera el servei web anirà demanant dades fent peticions HTTP utilitzant comunicació REST a la placa Arduino.

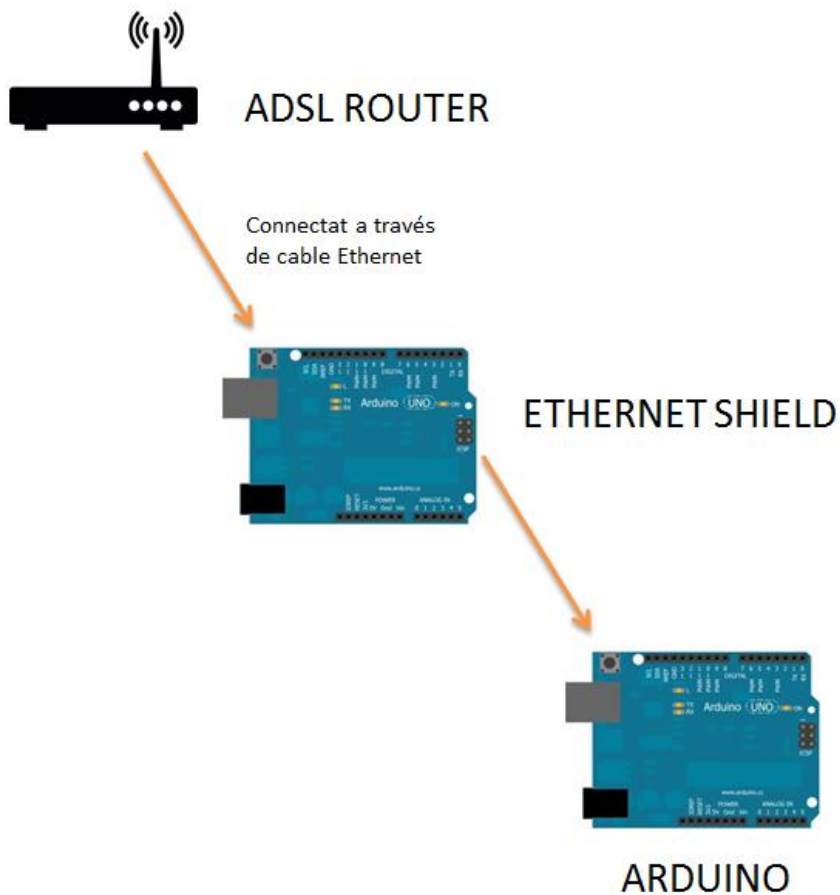
Hauré d'implementar la lògica de lectura i resposta de les peticions dins del servei llançat sobre el propi Arduino.

Per altra banda, la targeta comptarà amb diversos sensors que utilitzaran per interactuar amb els diferents dispositius. Per tant comptarà amb:

- Sensor de temperatura/humitat.
- Receptor de IR.
- IR LED

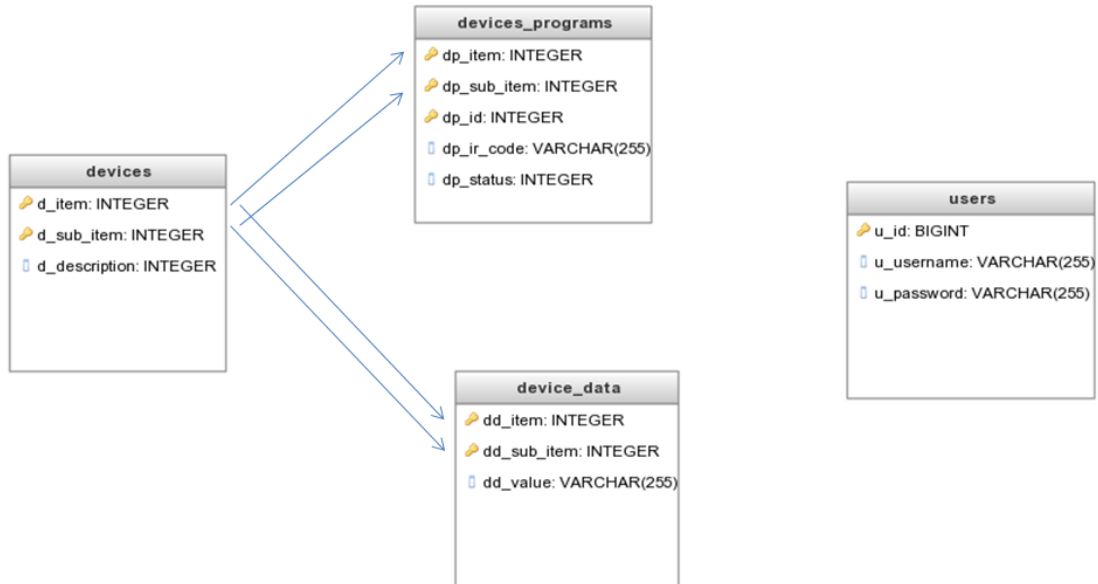
Arduino compta amb un entorn de desenvolupament propi, encara que existeixen plugins que permeten desenvolupar sota altres entorns com per exemple Visual Studio.

A continuació podem veure un breu diagrama de l'arquitectura que utilitzaré per el desenvolupament.



8.5 Disseny de Base de Dades

La base de dades estarà allotjada a la mateixa màquina on estarà funcionant el servei web.



Es crearan tres taules per emmagatzemar la informació

USERS		
camp	tipus	descripció
u_id	BIGINT	ID autonumèric del usuari
u_username	VARCHAR(255)	nom de l'usuari que utilitzarà per identificar-se
u_password	VARCHAR(255)	contrasenya del usuari que utilitzarà per identificar-se

devices		
camp	tipus	descripció
d_item	INTEGER	identificador del grup al que pertany el sensor
d_sub_item	INTEGER	sub grup del sensor
d_description	INTEGER	ID del recurs del text que apareixerà a l'aplicació

device_programs		
camp	tipus	descripció
dp_item	INTEGER	identificador del grup al que pertany el sensor
dp_sub_item	INTEGER	subgrup del sensor
dp_id	INTEGER	identificador del programa
dp_ir_code	VARCHAR(255)	codi infraroig del programa
dp_status	INTEGER	actiu/inactiu

device_data		
camp	tipus	descripció
dd_item	INTEGER	identificador del grup al que pertany el sensor
dd_sub_item	INTEGER	sub grup del sensor
dd_value	VARCHAR(255)	Valor del sensor

L'emmagatzemament de les dades ens donarà un millora en quant a la velocitat de resposta de les peticions. D'aquesta manera no haurem de demanar informació a la targeta Arduino si no que tindrem emmagatzemades les dades a la base de dades.

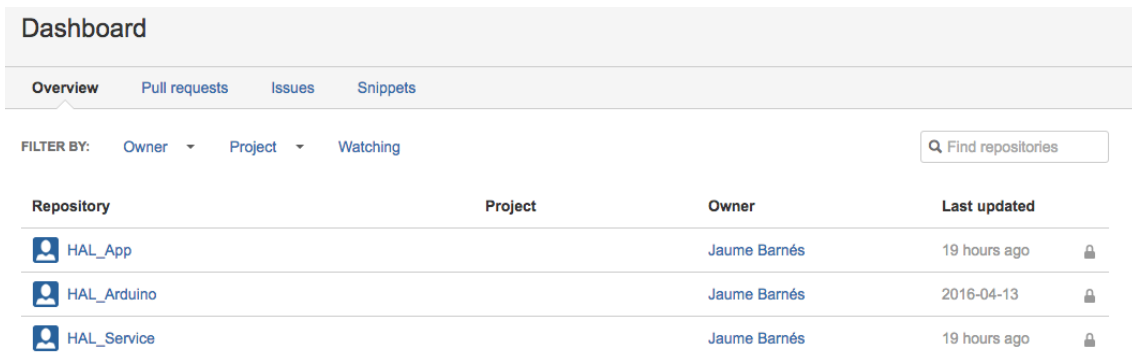


9 Control de versions







Per al control de versions s'ha utilitzat un repositori online anomenat Bitbucket. Bitbucket suporta control de versions amb Mercurial o Git, en aquest cas hem utilitzat Git.

El repositori ho he organitzat en 3 projectes:

- Projecte Arduino.
- Projecte Android.
- Projecte Web Service.

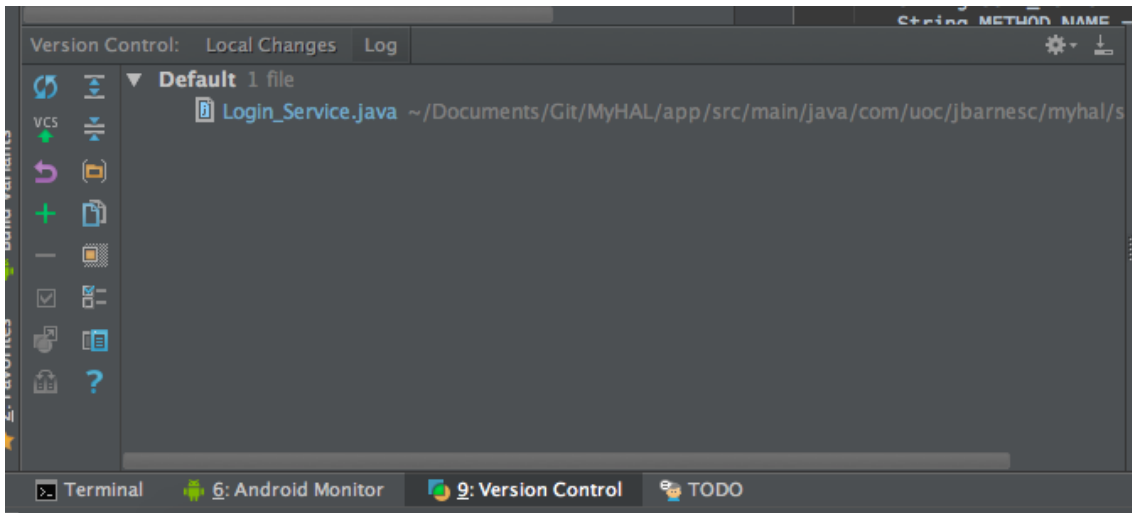


The screenshot shows the Bitbucket Dashboard interface. At the top, there's a 'Dashboard' header with navigation tabs for 'Overview', 'Pull requests', 'Issues', and 'Snippets'. Below this, there are filter options: 'FILTER BY: Owner', 'Project', and 'Watching', along with a search box labeled 'Find repositories'. The main content is a table listing three repositories.

Repository	Project	Owner	Last updated
 HAL_App		Jaume Barnés	19 hours ago 
 HAL_Arduino		Jaume Barnés	2016-04-13 
 HAL_Service		Jaume Barnés	19 hours ago 

[Imatge 1: Plana web de Bitbucket.org]

Un dels grans avantatges del repositori ha estat la perfecta integració tant amb l'entorn de desenvolupament per android “Android Studio” com per a Visual Studio 2013. De manera que des del propi entorn de desenvolupament ens permetia gestionar tots els canvis i pujar-los al repositori online.



[Imatge 2:Control de versions integrat amb Android Studio]

Per al projecte d'Arduino, degut que l'entorn de desenvolupament d'Arduino no permet la integració amb Git, s'ha decidit utilitzar les eines que ofereix Git, com Git bash, la qual permet executar sentències per pujar codi al repositori directament.

10 Decisions APP

El projecte d'Android és anomenat MyHAL App i ha estat desenvolupat sota la API mínima 23, que correspon a la versió de Android 2.2. D'aquesta manera l'aplicació amplia el mercat ja que actualment la gran majoria de dispositius Android suporten aquesta versió.

El projecte ha estat dividit en 3 parts lògiques:

- **Classes:** Corresponen als arxius que pertanyen a objectes bàsics. Si bé l'aplicació no té una gran càrrega de programació orientada a objectes ja que aquesta més enfocada a programació orientada a esdeveniments, existeixen diverses classes.
- **Serveis:** Corresponen als arxius que pertanyen a esdeveniments de comunicació amb el servei Web. Internament s'han separat els esdeveniments dels diferents dispositius de manera que quedi encapsulada cada petició en un arxiu.
- **Vistes:** Corresponen a les activitats que gestionen tots les pantalles i els seus recursos. S'ha creat una Activitat per pantalla.

La resta del projecte segueix la jerarquia bàsica d'un projecte de Android. Si bé cal destacar unes pocs directoris:

- **res/layout:** Conté tots els .xml que pertanyen a les diferents pantalles.
- **res/drawable:** Conté la llista d'icones i imatges que seran utilitzades en l'aplicació.
- **res/values:** Conté els fitxers de recursos d'idioma, en aquest cas anglès i espanyol.

10.1 Preferències

Per guardar les preferències d'un usuari existeixen diferents mètodes. Si bé un d'ells crec que exigeix massa implementació (base de dades SQLite) per al treball que ha de realitzar he optat per utilitzar l'objecte SharedPreferences.

Aquest objecte que gestiona internament Android, permet afegir elements a una col·lecció i capturar-los quan sigui necessari.

Un exemple de la implementació:

```
private void GetSettings() {
    m_prefs = this.getSharedPreferences(
        "com.uoc.jbarnesc.myhal", Context.MODE_PRIVATE);

    TextView _txt_con = ((TextView) findViewById(R.id.txt_conf_data));
    TextView _txt_key = ((TextView) findViewById(R.id.txt_conf_key));
    RadioButton _rb_en = (RadioButton) findViewById(R.id.rb_en);
    RadioButton _rb_es = (RadioButton) findViewById(R.id.rb_es);

    String _con = m_prefs.getString("com.uoc.jbarnesc.myhal.conn",
        "http://192.168.1.173:24108/HALService.asmx");
    String _key = m_prefs.getString("com.uoc.jbarnesc.myhal.key", "xbamACBDhdn12314kn");
    String _language = m_prefs.getString("com.uoc.jbarnesc.myhal.language", "en");
}
```

La gran avantatge d'aquesta classe és que permet accedir a les dades en qualsevol lloc de la nostra aplicació.



10.2 Idioma

S'ha adaptat l'aplicació perquè compleixi els estàndards de multi idioma, per la qual cosa s'han creat diferents fitxers de recursos per a cada idioma.

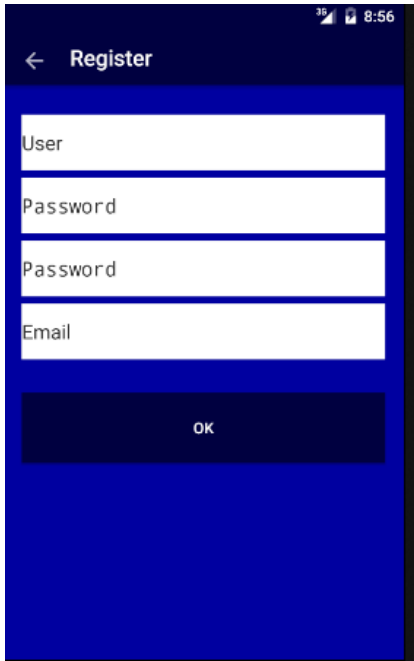
A més de la pròpia gestió que utilitza Android de l'idioma del dispositiu, s'ha implementat la funcionalitat de canviar-ho dins de la finestra de Configuració.

D'aquesta manera l'usuari pot canviar d'idioma sense canviar l'idioma del propi dispositiu.



10.3 Menú superior

Per a comoditat de l'usuari, s'ha implementat una barra d'eines que permet moure's cap a enrere de manera còmoda i intuïtiva.



10.4 SOAP

L'aplicació per dur a terme qualsevol acció necessita obtenir les dades del servei web MyHAL Service. Per a això necessita realitzar una petició al servei utilitzant el protocol de comunicacions SOAP.

La llibreria ksoap2 ens ofereix totes les eines necessàries per dur a terme la tasca, per això he decidit implementar-la.

```
public void attemptTemperature() {
    String SOAP_ACTION = "http://tempuri.org/GetDeviceData";
    String METHOD_NAME = "GetDeviceData";
    String NAMESPACE = "http://tempuri.org/";
    String URL = m_prefs.getString("com.uoc.jbarnesc.myhal.conn",
"http://192.168.1.173:24108/HALService.asmx");
    SoapPrimitive _result_string;
    String KEY = m_prefs.getString("com.uoc.jbarnesc.myhal.key", "xxxx");

    try {
```

```

SoapObject _request = new SoapObject(NAMESPACE, METHOD_NAME);
_request.addProperty("DeviceId", m_device_id);
_request.addProperty("SubitemId", m_sub_item_id);

SoapSerializationEnvelope _soap_envelope = new SoapSerializationEnvelope(SoapEnvelope.VER11);
_soap_envelope.dotNet = true;
_soap_envelope.setOutputSoapObject(_request);

HttpTransportSE _transport = new HttpTransportSE(URL);
_transport.call(SOAP_ACTION, _soap_envelope);

_result_string = (SoapPrimitive) _soap_envelope.getResponse();

m_list_temp = FormatData(_result_string.toString());

} catch (Exception _ex) {
    Log.d("LOGIN_SERVICE", _ex.getMessage());
}
}

```

En aquest exemple podem veure com coneixent els paràmetres d'un mètode de la API SOAP MyHAL Service realitza l'anomenada passant-li els paràmetres necessaris. Una vegada realitzada la petició el servei contesta i l'aplicació Android processa el resultat i ho mostra per pantalla.

Per inicialitzar la petició SOAP he necessitat l'adreça web del servei i el nom del mètode que hem d'executar.

Una anotació interessant és novament l'ús de l'objecte SharedPreferences, el qual hem utilitzat per obtenir les dades que va configurar l'usuari quan va indicar la ruta del servei Web.

11 Servei Web

El servei Web es va decidir desenvolupar sota tecnologia .Net a causa que tenia una major soltesa dins d'aquesta plataforma. El desenvolupament s'ha realitzat sota el Framework 4.5 utilitzant l'entorn de desenvolupament Visual Studio 2013.

Una de les característiques del servei web és que ha de treballar com de web API, processant les peticions que rep dels dispositius mòbils a més de treballar com a pont entre els dispositius i la placa Arduino. D'aquesta manera el servei web haurà d'enviar peticions a la placa Arduino.

La instal·lació del servei Web al client s'haurà de fer en una màquina fent ús de la eina Internet Information Services (IIS) per tal de fer el servei accessible des de la xarxa.

El projecte ha estat creat com a Servei Web d'ASP .NET, per la qual cosa ha generat l'estructura d'un servei web típic.

Posteriorment he dividit el projecte en diferents parts:

- **DAL:** Conté les classes que treballen directament amb la base de dades. Aquestes classes contenen tots els mètodes d'accés a dades.
- **Database:** Conté la classe bàsic de connexió a base de dades. Tots els accessos a base de dades utilitzen una instància d'aquesta classe.
- **Db_data:** Conté l'arxiu de base de dades SQLite.
- **Tasks:** Conté les classes que s'encarreguen d'executar-se cada cert període de temps en diferents fils de treball.

11.1 SQLite

SQLite és un sistema de base de dades relacional continguda en una petita llibreria escrita en C. Una dels grans avantatges de SQLite és la seva grandària reduïda i el seu aïllament ja que aquesta està encapsulada en un arxiu que pot ser ubicat en qualsevol lloc.

Per a major comoditat de l'usuari s'ha utilitzat SQLite per així evitar consum d'espai i recursos que tenen altres motors de bases de dades que ofereixen més característiques que el projecte necessita.

El projecte incorpora una base de dades neta, per la qual cosa l'usuari podrà començar a utilitzar-la una vegada rebí el paquet d'instal·lació.

11.1.1 Implementació

La implementació s'ha dut a terme utilitzant la llibreria externa de System.Data.SQLite. La implementació en si s'assembla a qualsevol implementació SQL (System.Data.SqlClient).



La classe DB_Trx permet establir la connexió amb la base de dades i obrir una transacció. Atès que SQLite permet l'ús de transaccions he cregut convenient fer ús d'elles de manera que ens ofereixi la possibilitat de realitzar rollback dels processos en els quals hi hagi algun error.

Per a l'execució de sentències SQL s'ha utilitzat les eines que ofereix ADO .NET.

```

Stringbuilder _str_query;
try
{
    using (DB_TRX _db = new DB_TRX())
    {
        _str_query = new StringBuilder();
        _str_query.Append("UPDATE DEVICES_PROGRAMS");
        _str_query.Append(" SET DP_STATUS = 0");
        _str_query.Append(" WHERE DP_ITEM = @pItem AND DP_SUB_ITEM = @pSubItem");

        using (SQLiteCommand _sql_cmd = new SQLiteCommand(_str_query.ToString(), _db.SqlTransaction.Connection, _db.SqlTransaction))
        {
            _sql_cmd.Parameters.Add("@pItem", System.Data.DbType.Int32).Value = DeviceId;
            _sql_cmd.Parameters.Add("@pSubItem", System.Data.DbType.Int32).Value = SubitemId;

            _sql_cmd.ExecuteNonQuery();
        }

        _str_query = new StringBuilder();
        _str_query.Append("UPDATE DEVICES_PROGRAMS");
        _str_query.Append(" SET DP_STATUS = 1");
        _str_query.Append(" WHERE DP_ITEM = @pItem AND DP_SUB_ITEM = @pSubItem AND DP_ID = @pProgramId");

        using (SQLiteCommand _sql_cmd = new SQLiteCommand(_str_query.ToString(), _db.SqlTransaction.Connection, _db.SqlTransaction))
        {
            _sql_cmd.Parameters.Add("@pItem", System.Data.DbType.Int32).Value = DeviceId;
            _sql_cmd.Parameters.Add("@pSubItem", System.Data.DbType.Int32).Value = SubitemId;
            _sql_cmd.Parameters.Add("@pProgramId", System.Data.DbType.Int32).Value = ProgramId;

            _num_rows = _sql_cmd.ExecuteNonQuery();
        }

        _db.Commit();
    }
    return (_num_rows > 0);
}

```

11.2 Servei SOAP

El servei publica els següents mètodes:

- **GetDeviceData(DeviceId,SubitemId)**
- **GetDeviceProgramEnabled(DeviceId,SubitemId)**
- **GetDevicePrograms(DeviceId,SubitemId)**
- **Login(Username,Password)**
- **Register(Username,Password,Email)**
- **SendProgram(DeviceId,SubItemid,ProgramId)**

L'arxiu HalService.asmx.cs conté els mètodes que quedaran publicats pel servei. Cada mètode es compon d'una etiqueta que ho qualifica com a mètode web el qual serà contestat en format JSON.

```
[WebMethod]
[ScriptMethod(ResponseFormat = ResponseFormat.Json)]
public string Login(String Username, String Password)
{
```

Una vegada rebuda la petició, es crearà una instància de DAL que accedirà a la base de dades i retornarà un valor que serà enviat al dispositiu.

Les diferents funcionalitats implementades han estat dividides en tres blocs:

- **DAL_Arduino:** Implementa les funcionalitats de envio de dades feia la placa Arduino.
 - **Mètode:**
 - **SendMessage:** Envia una petició REST de consulta de temperatura o enviament de codi Infraroig.

- **DAL_Devices:** Implementa les funcionalitats relacionades amb les dades dels diferents dispositius accedint a buscar-los a la base de dades.
 - **Mètodes:**
 - **GetDeviceData:** Consulta el valor de base de dades del dispositiu indicat.
 - **GetDeviceProgram:** Retorna el codi Infraroig del programa sol·licitat.
 - **GetDeviceProgramEnabled:** Retorna el codi del programa actiu.
 - **GetProgramsFromDevice:** Retorna la llista de programes del dispositiu.
 - **UpdateDeviceData:** Actualitza el valor del dispositiu.
 - **UpdateDeviceProgramStatus:** Actualitza l'estat del programa del dispositiu.

- **DAL_User:** Implementa les funcionalitats relacionades amb les dades dels usuaris.
 - **Mètodes:**
 - **Login:** Implementa el control d'accés de l'aplicació.
 - **Register:** Implementa el registre de l'usuari.

11.3 Tasques

S'ha implementat un thread que s'executarà cada 5 minuts. Aquest thread realitzarà peticions REST a la placa Arduino amb tal de capturar la temperatura en aquest moment. Una vegada capturada la temperatura serà guardada en la base de dades en el registre del dispositiu.

S'ha decidit aquesta implementació amb tal d'evitar el temps d'espera entre l'enviament i resposta entre el servei i la placa Arduino. D'aquesta manera, en tenir guardat el valor en base de dades ens permet capturar directament el valor optimitzant el procés.

Per a l'enviament REST cap a Arduino s'ha implementat un mètode d'enviament de peticions http indicant els paràmetres necessaris.

```
String _str_response;
String _address;
String _parameters;
try
{
    _xml_response = new XmlDocument();
    _client = new HttpClient();

    // Add an Accept header for JSON format.
    _client.DefaultRequestHeaders.Accept.Add(new MediaTypeWithQualityHeaderValue("application/json"));

    switch (MessageType)
    {
        case MESSAGE_TYPE.TEMPERATURE:
            _address = "http://" + WebConfigurationManager.AppSettings["arduino_ip"].ToString() + "/";
            _parameters = "?DEVICE=TEMP";
            break;

        case MESSAGE_TYPE.AC:
            _address = "http://" + WebConfigurationManager.AppSettings["arduino_ip"].ToString() + "/";
            _parameters = "?DEVICE=AC[" + IRCode + "]";
            break;

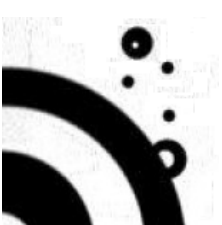
        default:
            _address = "";
            _parameters = "";
            break;
    }

    _client.BaseAddress = new Uri(_address);

    // List data response.
    _response = _client.GetAsync(_parameters).Result;

    if (_response.IsSuccessStatusCode)
    {
        // Parse the response body
        _str_response = _response.Content.ReadAsStringAsync().Result;
        Response = _str_response;
    }
}
```

En el cas de l'Aire condicionat, s'ha implementat de tal manera que en la pròpia petició anirà el codi Infraroig que posteriorment enviarà la placa Arduino.



12. Arduino

El paquet de Arduino comptarà amb les següents aplicacions:

- **Lector IR:** Aquesta aplicació permetrà capturar els codis infrarojos dels diferents dispositius.
- **Web Service:** Implementa el servei REST que processarà les peticions que arribin del servei web MyHALService.

A part de les aplicacions, el client rebrà un dispositiu físic perfectament ensamblat que internament comptarà amb els segons dispositius:

- Sensor DHT temperatura
- Arduino Mega 2560
- Ethernet Board
- IR Sensor Receiver
- IR led
- Cable USB
- Cable d'alimentació

Atès que al mercat existeixen molts dispositius amb diferents codis IR, la instal·lació del sistema l'haurà de realitzar un tècnic.

El procés serà el següent:

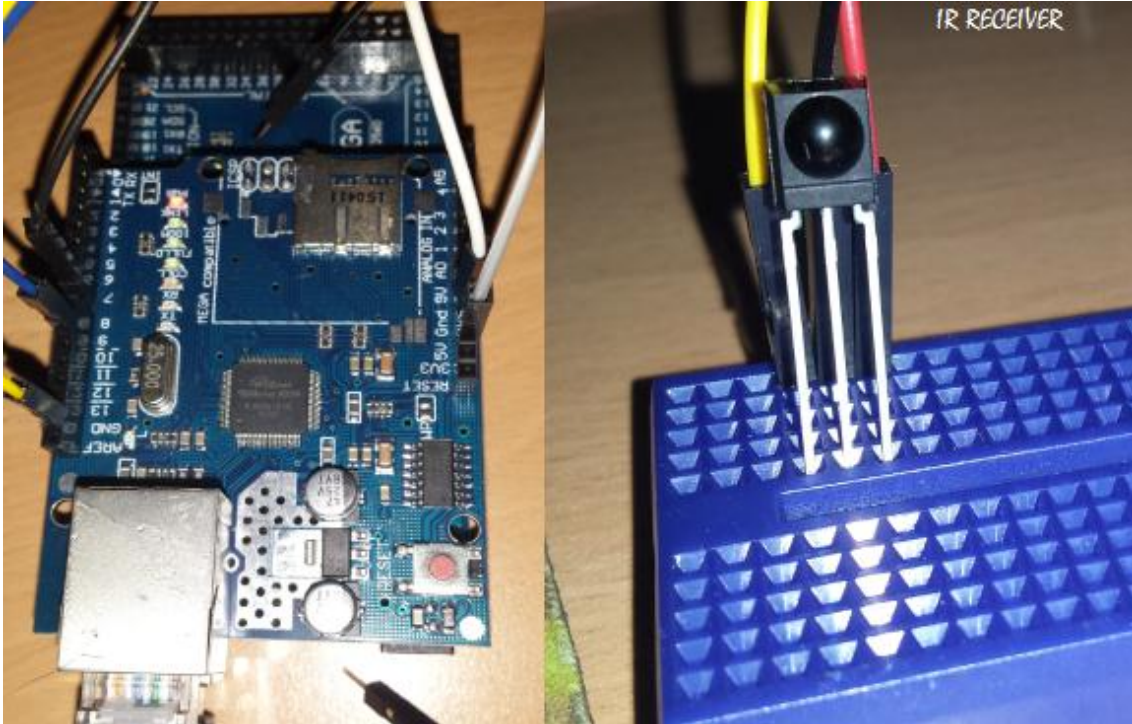
Un tècnic haurà de muntar el sistema de lectura de codi IR. Aquest sistema constarà de la placa Arduino amb l'aplicació Lector IR instal·lada i un lector de codis Infrarojos.

El tècnic haurà d'indicar el programa en el comandament del dispositiu i llançar el programa utilitzant el comandament. El sistema capturarà l'esdeveniment i mostrarà el codi IR. Aquest codi haurà de ser introduït com un nou programa dins de la base de dades.

Raw: (197) 6100, 7340, 588, 1644, 560, 1620, 560, 1620, 584, 1620, 552, 1656, 560, 1620, 552, 1656, 556, 1620, 584, 560, 556, 560, 552, 588, 560, 556, 552, 592, 552, 564, 548, 592, 552, 564, 560, 1644, 560, 1620, 552, 1680, 536, 1616, 588, 1620, 552, 1652, 560, 1624, 548, 1656, 560, 556, 556, 584, 560, 556, 556, 588, 560, 552, 560, 584, 560, 1620, 552, 1652, 552, 1620, 576, 1628, 556, 1624, 500, 1624, 556, 1652, 552, 1620, 576, 564, 560, 556, 608, 536, 556, 556, 556, 588, 560, 556, 608, 532, 560, 556, 556, 1648, 556, 560, 604, 1600, 552, 564, 612, 528, 552, 564, 616, 524, 588, 1592, 612, 528, 584, 1596, 616, 524, 560, 1644, 592, 1588, 604, 1600, 592, 1588, 616, 528, 584, 528, 616, 1592, 548, 1656, 580, 536, 556, 1648, 588, 1592, 612, 528, 584, 1596, 608, 1600, 552, 584, 592, 524, 556, 1648, 588, 528, 552, 588, 588, 1592, 612, 528, 584, 532, 604, 1600, 584, 532, 612, 1592, 588, 528, 608, 1596, 552, 588, 592, 524, 556, 1648, 588, 528, 552, 1652, 584, 532, 612, 1592, 556, 560, 608, 1596, 556, 1648, 588, 7308, 584,

12.1 Lectura codis IR

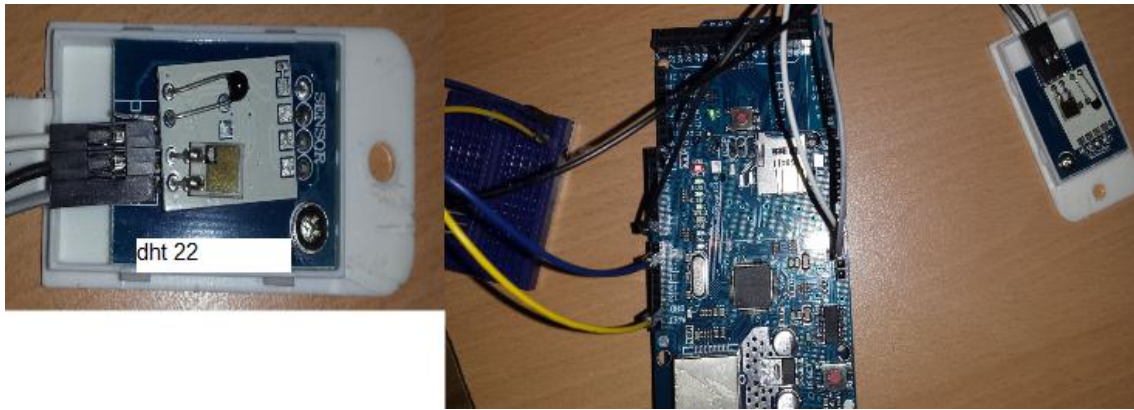
Tenint en compte els components següents:



Per a la lectura del codi IR he utilitzat la llibreria `IRremote.h` i `IRremoteInt.h`. Aquesta llibreria permet, indicant el PIN receptor obtenir el valor.

12.2 Implementació Servidor Arduino

Per la recepció d'esdeveniments s'ha implementat una solució que permet processar missatges enviats via http amb la targeta Arduino. Per a això he hagut d'afegir una targeta de LAN a la targeta Arduino.



Aquesta targeta permet oferir un punt d'entrada/sortida cap a la xarxa de la placa Arduino.

Utilitzant la llibreria Ethernet.h i indicant una MAC Address, una IP i un port s'estableixen els paràmetres necessaris per a que el dispositiu accedeixi a la xarxa arrancant el servei.

```
//MAC Address per defecte
byte m_mac[] = {
    0xDE, 0xAD, 0xBE, 0xEF, 0xFE, 0xED
};
IPAddress m_ip(192, 168, 1, 177);

// Initialize the Ethernet server library
// with the IP Address and port you want to use
// (port 80 is default for HTTP):
EthernetServer m_server(80);
```

Donat que el procés principal d'un programa Arduino es repeteix constantment, al arrancar el servei el programa quedarà a la espera de peticions entrants.

Amb una nova petició el sistema emmagatzemarà el text d'entrada fins que el missatge arribi al seu final i llavors processarà la petició.

S'han implementat dos funcionalitats:

- Consulta de temperatura.
- Enviament de codi IR

12.3 Consulta de temperatura

Utilitzant la llibreria DHT.h i indicant el tipus DHT22 (ja que, el sensor proporcionat és el DHT22) podem posteriorment consultar el valor de la temperatura.

```
if (_buffer.indexOf("GET /?DEVICE=TEMP") >= 0) {  
    //Read Temperature  
    int _h = m_dht.readHumidity();// Llegeix la humitat  
    int _t = m_dht.readTemperature(); // Llegeix la  
temperatura  
  
    Serial.print("REQUEST DEVICE=TEMP");  
    // send a Standard http response header  
    _client.println("HTTP/1.1 200 OK");  
    _client.println("Content-Type: text/html");  
    _client.println("Connection: close");  
    _client.println();  
    _client.print(_t);  
}
```



12.4 Enviament codi Infraroig

Utilitzant la llibreria de IRremote.h i IRremoteInt.h realitzarem l'enviament utilitzant la sentència

```
m_ir_send.sendRaw(_array, sizeof(_array) /
sizeof(int), m_khz);
```

Per a això, atès que l'enviament REST proporciona el codi ANAR, s'ha implementat un procés de lectura i processament del codi que posteriorment és transformat al tipus de dada que accepta el mètode sendRaw.

13 Punts de millora

- **Token de Seguretat:**

Donat que l'aplicació accedeix a una xarxa interna de la llar, s'ha de finalitzar d'implementar algun tipus de seguretat a més del Login. La seguretat pensada consisteix en generar una clau amb una clau privada interna del client. Aquesta clau estaria en possessió del usuari i per accedir un altre dispositiu seria necessari tenir aquesta clau.

- **Disseny**

Donades les infinites possibilitats de disseny, es fa necessari el treball de un dissenyador per millorar la visualització i accessibilitat de l'aplicació.

- **Mode configuració usuari:**

Actualment l'aplicació no permet que el usuari pugui afegir o esborrar dispositius de la llar que utilitzin codis IR. Per poder satisfer aquest requisit s'hauria d'implementar un mode de configuració per afegir nous programes.

El procés podria ser el següent: El usuari podria afegir un nou programa i entrar en mode lectura de codi IR. En aquest moment utilitzaria el comandament d'Infrarojos i la aplicació capturaria el codi IR.



14 Conclusions

Aquest projecte m'ha permès confeccionar des de l'inici un projecte amb les diferents fases que aquest comporta (Planificació, Anàlisi, Desenvolupament i Testing).

Crec que no vaig saber tenir present el fet que el desenvolupament requeria un procés d'aprenentatge previ de les noves tecnologies i que aquest requeria de molt temps. Aquest fet em va provocar un endarreriment important en quant els temps establerts en un inici.

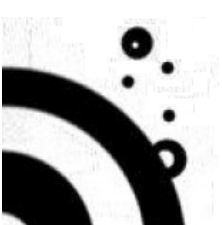
Durant el desenvolupament de la part mòbil vaig tenir problemes per implementar la llibreria KSOAP2 i es que al ser de codi obert, eren molts els llocs dels qual es podia descarregar i just el que vaig implementar donava problemes. Això em va fer perdre un temps que no tenia.

No obstant això, els resultats obtinguts penso que son satisfactoris i els coneixements que he obtingut en quan a tecnologia Android son suficients per tenir una base per continuar aprenent.

Per altra banda, també vaig descobrir un món nou amb la targeta Arduino. La targeta em permetia connectar diferents dispositius i interaccionar amb ells. Gràcies als coneixements adquirits durant l'enginyeria vaig poder desenvolupar sense gaires problemes les diferents funcionalitats. Tot i aquest avantatge, el procés d'aprenentatge de la connexió dels diferents components a més de la utilització de les diferents llibreries de Arduino (tant del propi Framework com de tercers) va ser complicat.

Tot plegat, juntament amb el servei, va fer que hagués d'aprendre a comunicar i desenvolupar sistemes totalment diferents entre ells i em va permetre investigar sobre formes de comunicació i traspàs de dades a través de la xarxa. Si bé els coneixements els vaig adquirir durant l'enginyeria, vaig necessitar ampliar-los.

Amb aquest projecte he volgut plasmar tots els coneixements adquirits durant l'enginyeria en els diferents àmbits de la programació. Des de utilització de components electrònics, la programació, les comunicacions i el desenvolupament amb tecnologia mòbil. Penso que amb tot plegat sorgeix un projecte força complet.



15 Bibliografia

Introduction to Android. API Guide[en línia]. <https://developer.android.com/guide/index.html> [data de consulta: 1 de abril de 2016]

Infrared Remote Tutorial. API Guide [en línia]. <http://www.instructables.com/id/Arduino-Infrared-Remote-tutorial/> [data de consulta: 5 de abril de 2016]

SOAP vs REST [en línia]. <http://spf13.com/post/soap-vs-rest> [data de consulta: 7 de abril de 2016]

Difference between WCF and Web Services [en línia].

<http://www.codeproject.com/Articles/139787/What-s-the-Difference-between-WCF-and-Web-Services> [data de consulta: 20 de maig de 2016]

Ethernet Shield Tutorial [en línia]. <http://www.educachip.com/arduino-ethernet-shield/> [data de consulta: 7 de maig de 2016]

SQLite Framework[en línia].

<https://system.data.sqlite.org/index.html/doc/trunk/www/index.wiki> [data de consulta: 17 de maig de 2016]

SQLite Advantages [en línia]. <https://www.sqlite.org/different.html> [data de consulta: 20 de maig de 2016]

What is SQLite? [en línia] <http://programmerguru.com/android-tutorial/what-is-sqlite/> [data de consulta: 5 de maig de 2016]

Control Air Conditioner with Arduino [en línia]. <http://www.instructables.com/id/How-to-control-the-air-conditioner-AC-at-home-with/> [data de consulta: 14 de maig de 2016]

Git Info [en línia]. <http://readwrite.com/2013/10/02/github-for-beginners-part-2/> [data de consulta: 1 de abril de 2016]

Bitbucket Info [en línia]. <https://bitbucket.org/support> [data de consulta: 1 de abril de 2016]

How to User IR Remote Arduino [en línia]. <http://www.instructables.com/id/The-Easiest-Way-to-Use-Any-IR-Remote-with-Ardiuno/> [data de consulta: 5 de maig de 2016]

Home Automation [en línia]. https://en.wikipedia.org/wiki/Home_automation [data de consulta: 7 de maig de 2016]

KSOAP2 [en línia]. <http://kobjects.org/ksoap2/index.html> [data de consulta: 20 de maig de 2016]

