

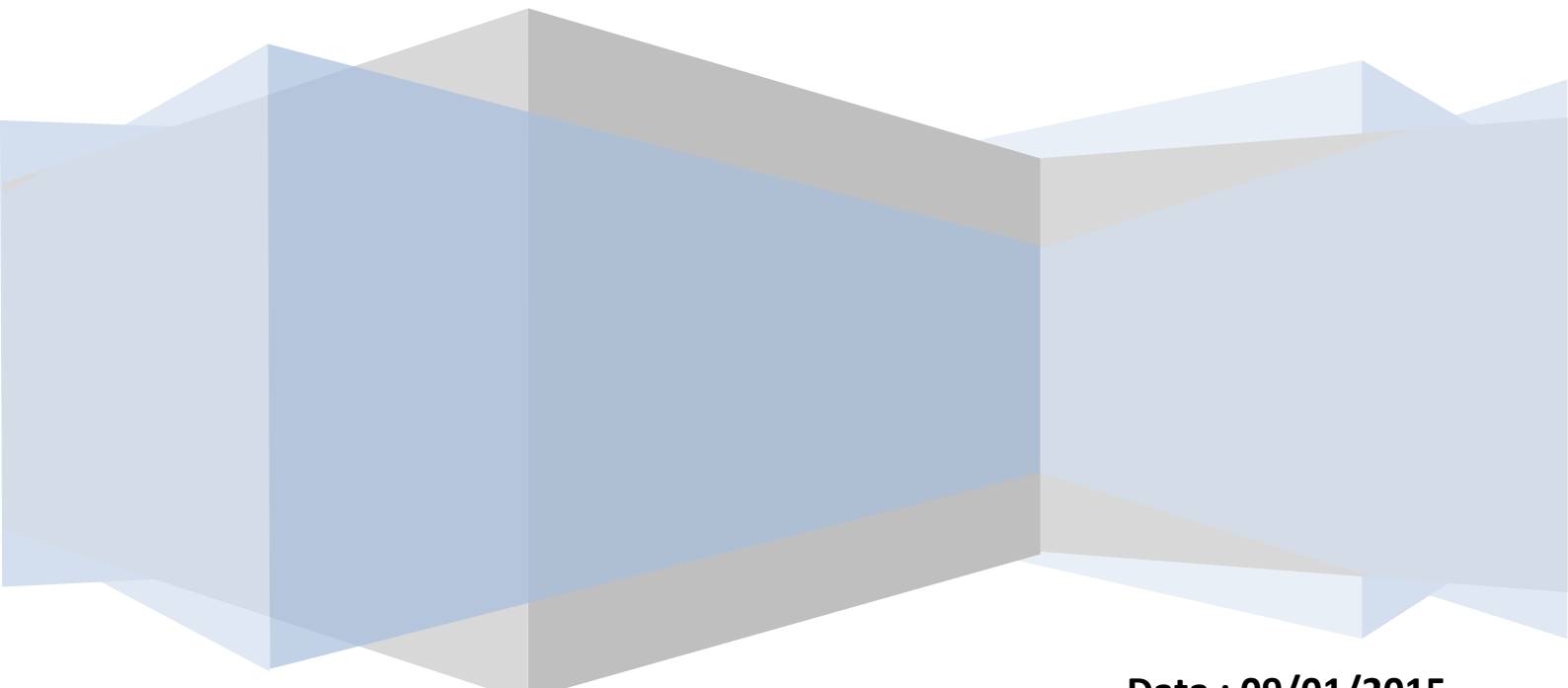


Arduino Tricorder

Enginyeria Informàtica

Autor : Sergi Sánchez Morato

Consultor : Jordi Ceballos Villach



Data : 09/01/2015

Per l'Anna i la Raquel,
pel temps que els hi pres, i que sense dubte els tornaré.

Contingut

1.	Introducció	6
1.1	Justificació del projecte.....	6
2.	Objectius	7
3.	Funcionalitats principals.....	8
4.	Recursos i infraestructura	8
4.1	Recursos de hardware.....	9
4.2	Recursos software.....	11
5.	Tecnologies a aplicar	12
5.1	Tecnologia Arduino	12
5.1.1	Models amb comunicació integrada	13
5.1.2	Shields de comunicació	15
5.1.3	Altres models.....	17
5.2	Tecnologies mòbils.....	19
6.	Riscos del projecte	19
7.	Calendari del projecte	21
7.1	Lliuraments establerts.....	21
7.2	Detall de l'entrega de la PAC1.....	22
7.3	Detall de l'entrega de la PAC2.....	23
7.4	Detall de l'entrega de la PAC3.....	25
7.5	Detall de l'entrega del Lliurament Final	27
8.	Anàlisi funcional	28
8.1	Requeriments funcionals	28
8.2	Requeriments no funcionals	28
8.2.1	Interfície	28
8.3	Funcionalitats del sistema.....	29
8.3.1	Plataforma mòbil.....	29
8.3.2	Plataforma Arduino	29
8.4	Usuaris del sistema.....	30
8.5	Casos d'ús.....	30
8.5.1	Descripció cas d'ús [C01] Crear projecte.....	30
8.5.2	Descripció cas d'ús [C02] Crear interfície tàtil.....	31
8.5.3	Descripció cas d'ús [C03] Crear interfície comandes veu	31

8.5.4	Descripció cas d'ús [C04] Crear interfície síntesi de veu.....	32
8.5.5	Descripció cas d'ús [C05] Crear interfície enregistrament de dades	32
8.5.6	Descripció cas d'ús [C06] Canviar valor sortida.....	33
8.5.7	Descripció cas d'ús [C07] Rebre dades entrada	33
9.	Disseny tècnic.....	34
9.1	Arquitectura Arduino (part servidora)	35
9.2	Arquitectura proves (part servidora)	38
9.3	Arquitectura aplicació mòbil (part client)	40
9.3.1	Control sortida digital.....	40
9.3.2	Control sortida analògica	40
9.3.3	Supervisió entrada digital.....	41
9.3.4	Supervisió entrada analògica	41
9.4	Arquitectura lògica	41
9.4.1	Arquitectura lògica de l'aplicació mòbil	41
9.4.2	Arquitectura Android	42
9.4.3	Arquitectura lògica de la plataforma Arduino	43
9.4.4	Arquitectura Arduino	45
9.4.5	Arquitectura lògica d'intercanvi d'informació entre plataformes	47
9.5	Arquitectura de base de dades	48
9.5.1	Model relacional de la base de dades	48
9.5.2	Entitat Projectes	50
9.5.3	Entitat Tàctil	50
9.5.4	Entitat Comandes_Veu.....	51
9.5.5	Entitat Sintesi_veu.....	51
9.5.6	Entitat Aquisicio_dades.....	52
9.6	Diagrama de classes	52
9.6.1	Capa de presentació.....	53
9.6.2	Capa de dades	57
9.6.3	Capa de comunicacions.....	58
9.6.4	Miscel·lània.....	59
10.	Prototip	60
	Pantalla de projectes.....	61
	Pantalla dades projecte.....	61
	Pantalla interfície tàctil	62

Pantalla alta/modificació control mode tàtil.....	62
Pantalla edició/eliminació control o comanda	63
Pantalla disseny interfície comandes de veu	63
Pantalla accions associades a comandes de veu.....	64
Pantalla disseny interfície monitorització amb síntesi de veu	64
Pantalla de selecció dels valors a monitoritzar	65
Pantalla disseny interfície enregistrament de dades	66
Pantalla de selecció dels valors a enregistrar	67
11. Implementació	67
11.1 Implementació de l'aplicació mòbil Android	67
11.2 Implementació dels sketch Arduino.....	76
11.2.1 Consideracions prèvies.....	76
12. Funcionament de l'aplicació.....	81
12.1 Creació i llistat de projectes	81
12.2 Modalitat tàtil.....	82
12.3 Modalitat comandes de veu.....	83
12.4 Modalitat síntesi de veu.....	84
12.5 Modalitat adquisició de dades	84
12.6 Llistat per editar els controls d'un projecte	85
13. Conclusions	86
13.1 Assoliment d'objectius	86
13.2 Desviacions del projecte original	86
13.3 Valoració personal.....	87
13.4 Futures millores.....	87
14. Fonts d'informació	89
14.1 Fonts sobre Android.....	89
14.2 Fonts sobre Arduino.....	90

1. Introducció

Arduino és una plataforma de hardware i codi lliure, formada per un microcontrolador, un entorn de programació i abundant documentació. Va néixer al 2005 a la Universitat de Ivrea, Itàlia. El microcontrolador Arduino ha aconseguit una gran popularitat gràcies a la seva versatilitat, facilitat de programació i baix cost. Existeixen a l'actualitat una dotzena de models diferents, amb diverses capacitats, mides i preus, que oscil·len entre els 10 i 200 euros.

La plataforma Arduino gaudeix d'un gran nombre d'usuaris arreu del món. Un nombre que va creixent any rere any, ja que ha esdevingut una eina d'iniciació a l'electrònica i a la programació degut a la seva senzillesa. Deixant enrere els temps en que aquest tipus de dispositius només es podien programar amb llenguatges tant poc amigables com l'ensamblador i similars.

Un fet determinant per a aquesta gran acollida, és que en el poc temps que fa que existeix han anat sorgint nous models, cada cop més potents, i sobretot, incorporant més capacitats de comunicació. Així empreses com Google, Intel i Telefònica han col·laborat amb els creadors d'Arduino per crear noves plaques. I això sembla només el principi, com indica el recent Intel Edison (processador de doble nucli, 1 Gb Ram, Bluetooth, Wifi) de la mida d'una targeta SD, i de només 40 euros.

És per això que paraules com "Makers", "DIY" (Do it yourself), "IoT" (Internet of things), "Wearables", són cada cop més freqüents, indicant que no només és una moda d'uns quants entusiastes de la tecnologia, sinó que cada cop veiem més dispositius "intel·ligents" i connectats. La majoria d'aquests nous dispositius es connecten - com no pot ser d'una altra manera en l'era de la mobilitat - amb el mòbil. I és que la tendència és a fer del mòbil el dispositiu que faci de nexa entre les persones i el món real.

1.1 Justificació del projecte

En aquest context, moltes persones amb interès per la tecnologia, amb formació o sense, fan servir el microcontrolador Arduino per experimentar i prototipar els seus innovadors projectes mecatrònics. Però, per a gran part d'ells la integració amb mòbil és ara mateix una barrera. La programació amb Arduino és molt senzilla (d'aquí part del seu gran èxit) però els llenguatges de programació d'alt nivell per programació de dispositius mòbils, com Java, intimida a molts usuaris d'Arduino.

Actualment al Google Play Store trobem molt poques apps que ajudin als usuaris a interactuar amb la plataforma Arduino, i que realitzin tasques de supervisió i control de projectes fent servir aquesta plataforma. Excepte una única excepció, les existents són poc configurables, tenen interfícies poc amigables i presenten moltes mancances. Es per això que es pretén facilitar als usuaris una app per al sistema operatiu Android que permeti dissenyar interfícies de comunicació bidireccional mitjançant comunicació wifi amb la plataforma Arduino de forma ràpida, fàcil i altament configurable.

A més, es vol fer un pas important en quan al tipus d'interfície, aprofitant les capacitats que ofereixen actualment els dispositius mòbils, implementar aquesta comunicació de forma verbal, amb l'enviament de comandes per veu, i la sintetització per veu de la informació obtinguda de sensòrica connectada a l'Arduino.

2. Objectius

L'objectiu principal d'aquest projecte és fer una aportació al moviment de la cultura "Maker" del que tant he après i que fa una gran tasca per impulsar que el coneixent tecnològic arribi a tothom.

En concret es pretén facilitar al usuaris de la plataforma Arduino la possibilitat de supervisar i controlar els seus projectes des d'un dispositiu Android (mòbil, tablet, android tv, etc), sense requerir de coneixements de programació en l'esmentada plataforma.

De forma senzilla i molt intuïtiva es podran controlar i/o supervisar les entrades i sortides tant digitals com analògiques amb l'ajuda d'un assistent.

A la documentació oficial d'Arduino i en el mateix entorn de programació per programar-lo existeixen gran quantitat d'exemples de com rebre paràmetres per HTTP, però en canvi és escassa pel que fa a fer accessible informació en format JSON des dels models que incorporen servidor web. Per tant, es pretén també facilitar una llibreria i alguns exemples de programes d'Arduino que recullin dades dels sensors i les facin accessibles via HTTP en format JSON.

Personalment tenia interès en desenvolupar una aplicació nativa en Android després de treballar amb plataformes més senzilles com Phonegap, i que aquesta arribés al Google Play Store per tal rebre el feedback dels usuaris. A més, voldria publicitar l'aplicació en els nombrosos fòrums i blogs sobre Arduino, per tal d'arribar a un gran nombre d'usuaris potencials.

De fet en aquest projecte s'uneixen el meu interès per la programació i l'electrònica, i és per aquesta raó que l'emprenc amb entusiasme.

Seguint el lema del moviment "Maker", he decidit ser creador i no només usuari.

3. Funcionalitats principals

A continuació descriuré les principals funcionalitats de l'aplicació mòbil :

- Crear projectes : permet donar d'alta un projecte indicant el nom , la adreça IP de l'Arduino i tipus d'interfície.
- Dissenyar interfície tàctil : permet crear una interfície per comunicar-se amb l'Arduino afegint diferents controls segons si es vol supervisar (entrades digitals i analògiques) o controlar (sortides digitals i analògiques) .
- Dissenyar interfície de comandes verbals : permet crear una interfície per enviar ordres a l'Arduino mitjançant comandes verbals.
- Dissenyar interfície síntesi de veu : permet crear una interfície que reproduïx mitjançant síntesi de veu la informació capturada de l'Arduino.
- Dissenyar interfície d'adquisició de dades : permet crear una interfície que enregistra a la memòria del dispositiu mòbil la informació capturada de l'Arduino.

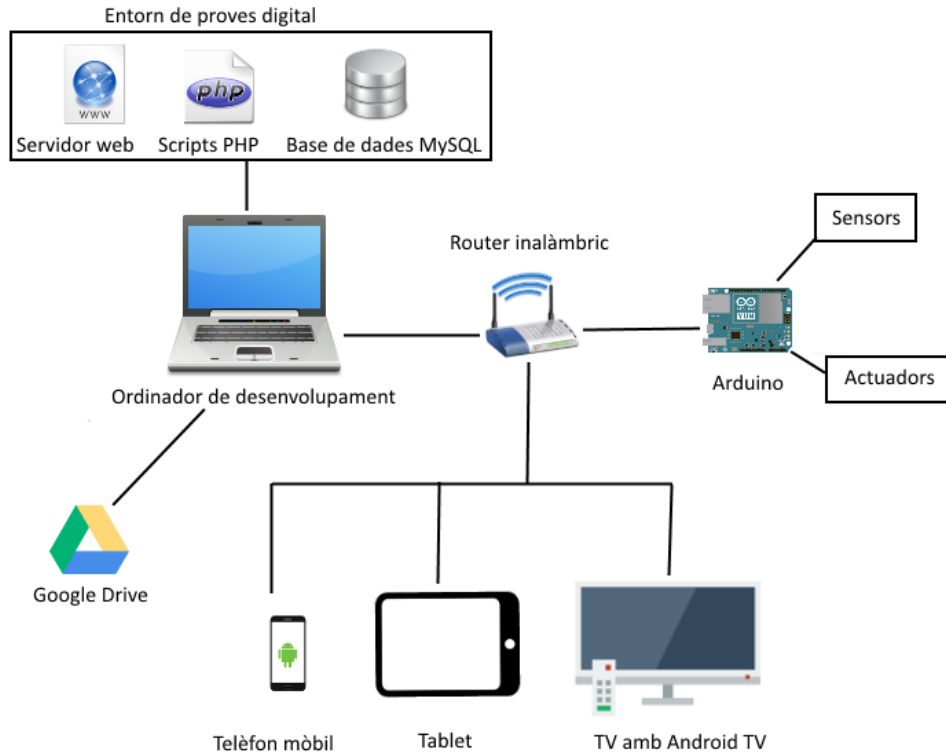
4. Recursos i infraestructura

Pel desenvolupament de l'aplicació mòbil fem servir un ordinador amb l'entorn de desenvolupament Eclipse amb l'Android SDK. En concret s'ha instal·lat l'Eclipse ADT Bundle, que incorpora directament el Android SDK dins del l'Eclipse.

En el mateix ordinador s'utilitza l'entorn de programació d'Arduino per programar i descarregar els sketches que corren sobre l'Arduino. Com en el cas anterior, es tracta d'entorns de programació gratuïts.

Per verificar que el desenvolupament es realitza correctament durant la fase d'implementació, s'ha muntat un entorn de proves digital - sense els elements físics que intervenen en el projecte (plaques Arduino, sensors i actuadors) – per tal d'agilitzar les proves.

Per comprovar el funcionament real, s'utilitzen d'una banda diversos dispositius amb sistema operatiu Android (telèfon, tablet i Android TV) i dos models d'Arduino que disposen de comunicació Ethernet i Wifi.



4.1 Recursos de hardware

La següent taula mostra els elements de maquinari que han intervingut en el desenvolupament del projecte :

Component	Característiques tècniques	Funció
Ordinador de desenvolupament	CPU : Intel Core i3 2.27 Ghz RAM : 4Gb Sistema operatiu : Windows 7 64 bits Professional	- Desenvolupament de l'aplicació Android i els scripts Arduino - Consulta i elaboració de documentació - Creació i edició d'imatges
Telèfon mòbil	Nexus 4 CPU : QuadCore a 1,5 Ghz RAM : 2Gb Sistema operatiu: Android 4.4 (Kitkat) Pantalla : 4,7" (1280 x 768)	- Comprovació del funcionament real de l'aplicació. Supervisió, control i adquisició de dades de l'Arduino.

Arduino Tricorder

	Connexió 3G i Wifi	
Tablet	<p>BQ Edison 2</p> <p>CPU : Dual Core a 1,6 Ghz</p> <p>RAM : 1Gb</p> <p>Sistema operatiu : Android 4.1 (Jelly Bean)</p> <p>Pantalla : 10.1 " (1280x800)</p> <p>Connexió Wifi</p>	- Comprovació del funcionament real de l'aplicació. Supervisió, control i adquisició de dades de l'Arduino.
Android TV	<p>Rikomagic Mk802S III</p> <p>CPU : Dual Core a 1.6 Ghz</p> <p>RAM : 1Gb</p> <p>Sisteme operatiu : Android 4.2 (Jelly Bean)</p> <p>Connexió Wifi</p>	- Comprovació del funcionament real de l'aplicació. Supervisió, control i adquisició de dades de l'Arduino.
Microcontrolador Arduino	Arduino Yún	- Lectura sensors i control actuadors
Microcontrolador Arduino	Arduino Ethernet	- Lectura sensors i control actuadors
Sensors	<p>Temperatura</p> <p>Moviment</p> <p>Ultrasons</p> <p>Fototransistor</p>	- Capturar informació de l'entorn
Actuadors	<p>Relé</p> <p>Led RGB</p>	- Actuar sobre l'entorn

4.2 Recursos software

La següent taula mostra el programari que ha intervingut en el desenvolupament del projecte :

Entorn de programació	Funció
Eclipse ADT Bundle	Entorn de desenvolupament per a l'aplicació mòbil (JAVA) https://developer.android.com/sdk/installing/index.html?pkg=adt
Arduino IDE 1.5.7	Entorn de desenvolupament per als scripts d'Arduino http://arduino.cc/en/Main/Software
GanttProject	http://www.ganttproject.biz/
Adobe Reader X	Programari per a la visualització de documentació en format PDF http://get.adobe.com/es/reader
Magic Draw UML	Programari per al disseny de diagrames UML http://www.nomagic.com/
Microsoft Office 2007	Programari per a l'elaboració de la documentació http://www.microsoft.com/es-es/download/office.aspx
Paint.net	Programari per a l'edició d'imatges http://www.getpaint.net/
Google Drive	Programari per a l'emmagatzemament segur de les dades https://www.google.com/intl/es_es/drive/
Apache Web Server (*)	Programari per a simular el servidor web de l'Arduino
MySQL i PhpMyAdmin (*)	Programari per enregistrar les peticions a l'Arduino simulades en l'entorn de proves
PHP (*)	Intèrpret per executar els scripts per enregistrar i mostrar la informació simulada en l'entorn de proves

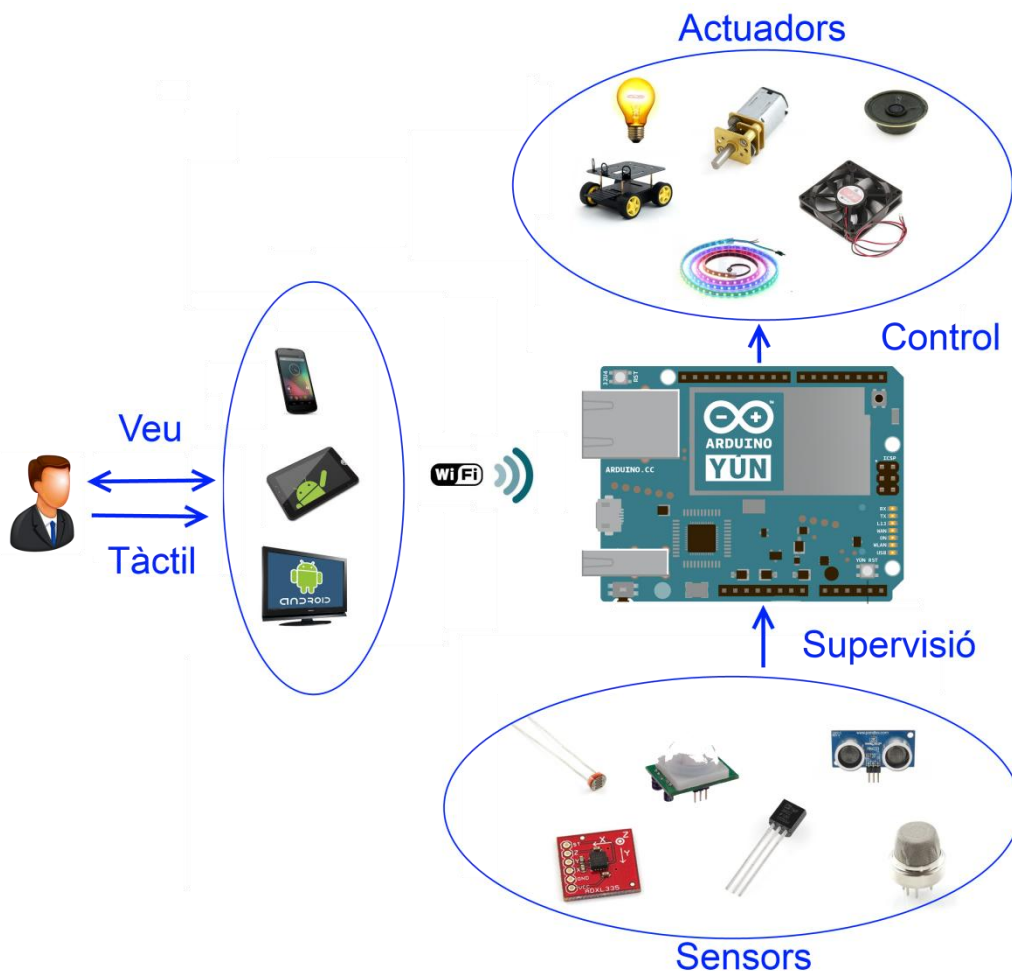
(*) Programari utilitzat dins del paquet XAMPP 1.8.3

(<https://www.apachefriends.org/es/index.html>)

5. Tecnologies a aplicar

El projecte està dividit en una part digital, formada per l'aplicació mòbil per a dispositius mòbils Android, i una part física, la formada per la placa microcontroladora Arduino, i els sensors i actuadors que hi pot tenir connectats.

Veiem amb més detall cadascuna d'aquestes parts :



5.1 Tecnologia Arduino

Per tal de fer possible la comunicació entre l'aplicació mòbil i la placa Arduino, aquesta ha de comptar amb capacitat de comunicació IP. A continuació farem una comparativa amb els diferents models disponibles actualment. Com veure existeixen plaques Arduino que disposen de comunicació Ethernet i/o Wifi. Cal distingir entre les plaques que disposen de les capacitats de comunicació integrades, i les que necessiten d'un *Shield* per incorporar-les.

5.1.1 Models amb comunicació integrada

Les plaques que disposen de comunicació integrada són les més cares, però en molts casos resulten més econòmiques que comprar una placa bàsica i posteriorment afegir-hi el *shield* de comunicació.

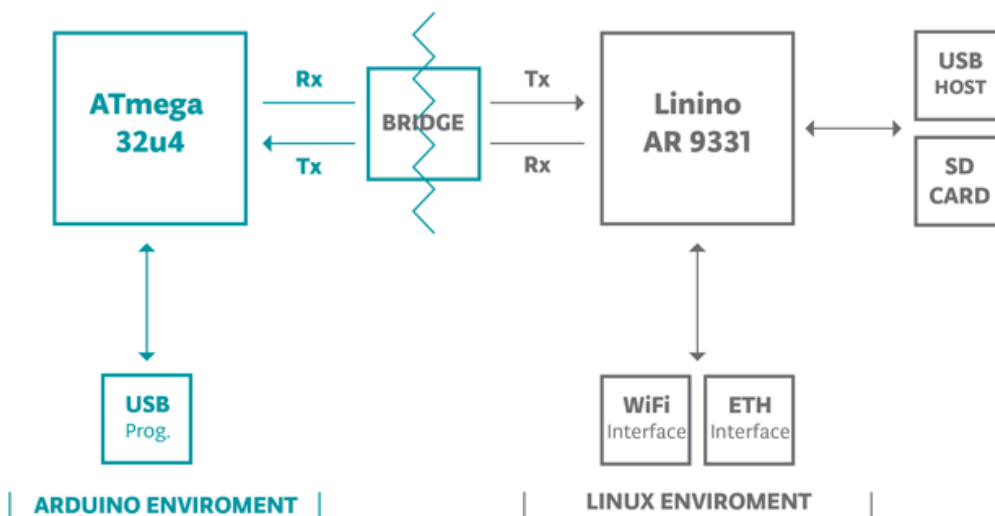
5.1.1.1 Arduino Yún

L'Arduino Yún és a l'actualitat la placa Arduino més versàtil. És la primera que incorpora 2 processadors, el ATmega32u4 i el Atheros AR9331. El procesador Atheros suporta una distribució Linux basada en OpenWrt. Aquesta placa integra connectivitat Ethernet i Wifi, un port USB i un slot per a una targeta microSD.



L'Arduino Yún disposa de 20 pins d'entrada/sortida digital (7 d'ells es poden fer servir com a sortides PWM i 12 com a entrades analògiques).

El Yún es diferencia d'altres plaques Arduino per la seva capacitat per comunicar-se amb la distribució Linux que porta integrada, convertint-lo en un ordinador amb la facilitat d'ús d'Arduino.



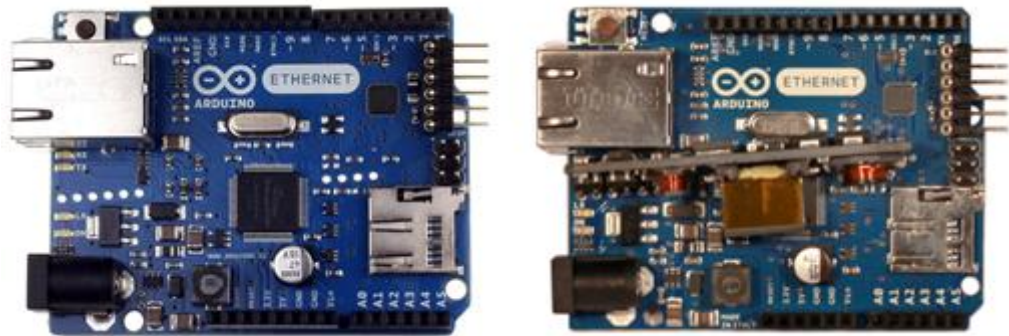
La llibreria Bridge permet la comunicació entre ambdós processadors, dotant als sketches Arduino de la possibilitat d'executar Shell Scripts, realitzar comunicacions Wifi i Ethernet i rebre informació del processador AR9331.

Destaca per la seva capacitat per d'actuar com a client i/o servidor web.

5.1.1.2 Arduino Ethernet

L'Arduino Ethernet va ser la primera placa Arduino en integrar comunicació Ethernet i està basada en el processador ATmega328.

Té 14 entrades/sortides digitals, 6 pins analògics. Els pins 10, 11, 12 i 13 estan reservats per a la comunicació amb el mòdul Ethernet. Això redueix el nombre de pins digitals disponibles a 9, 4 d'ells amb capacitat de sortida PWM.



L'Arduino Ethernet incorpora un lector de targetes SD. Requereix d'un adaptador USB Sèrie de 6 pins per a la descàrrega de sketches.

Incorpora les funcionalitats de client i servidor web.

5.1.1.3 Comparativa Arduino Yún i Arduino Ethernet

	Arduino Yún	Arduino Ethernet
Microcontrolador	ATmega32u4 i Atheros AR9331	ATmega328
Voltatge de funcionament	5v	5V
Voltatge d'entrada	5v	7-12V
Entrades/Sortides digitals	20	14

Sortides PWM	7	4
Entrades/Sortides analògiques	12	6
SRAM	2.5 KB	2 KB
Memòria Flash	32kb i 16 MB	32 KB
Velocitat rellotge	16 MHz i 400MHz	16 MHz
Ethernet	IEEE 802.3 10/100Mbit/s	IEEE 802.3 10/100Mbit/s
Wifi	IEEE 802.11b/g/n	
RAM	64 MB DDR2	
Mides i pes	73 x 53 mm i 32 g	68.6 x 53.3 mm i 28 g
Suport PoE (Power over Ethernet)	Sí	Sí

5.1.2 Shields de comunicació

La plataforma hardware Arduino és fàcilment escalable gràcies als anomenats *Shields*. Aquests elements són plaques amb una funcionalitat concreta (comunicació, emmagatzemament, pantalla, GPS, etc), que es col·loquen sobre els pins d'una placa Arduino per dotar-lo de noves capacitats.

Tanmateix, no es poden utilitzar amb els models més petits, com l'Arduino Mini, l'Arduino LilyPad i d'altres, que no segueixen la configuració de pins típica.

5.1.2.1 Ethernet Shield

L'Arduino Ethernet Shield permet dotar de comunicació Ethernet a plaques Arduino que no porten integrada de sèrie aquest tipus de comunicació.

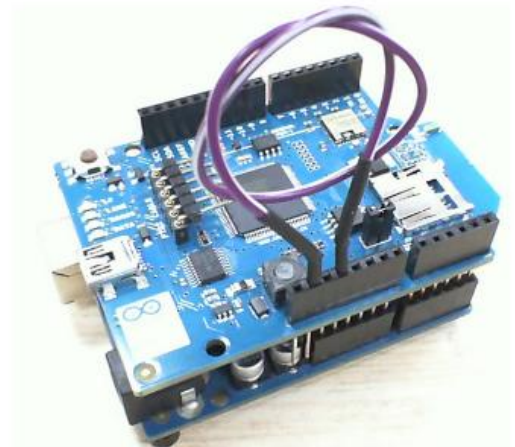
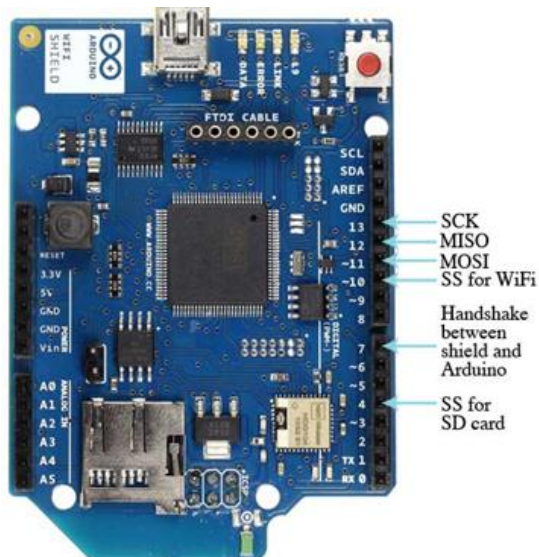


El xip Ethernet que utilitza és el Wiznet W5100 i treballa a velocitats de 10/100 Mbps. Incorpora un slot per a targetes microSD on desar documents i servir-los a través de la xarxa.

Per comunicar-se amb el mòdul Ethernet i SD, fa servir els pins 10, 11, 12 i 13, en el cas de l'Arduino UNO i Leonardo, i el 50,51 per a l'Arduino Mega.

5.1.2.2 Wifi Shield

L'Arduino Wifi Shield permet donar de connexió Wifi a les plaques Arduino que no disposen d'aquest tipus de comunicació.



Aquest mòdul és capaç de connectar-se a xarxes inalàmbriques obertes i amb seguretat WEP i WPA2 Personal i sota protocol 802.11 b/g. La xarxa a la que es connecti, ha de fer visible el seu SSID.

Incorpora un slot per a targetes microSD on desar documents i servir-los a través de la xarxa.

5.1.2.3 Gsm Shield

L'Arduino GSM Shield permet connectar una placa Arduino a internet fent servir la xarxa GPRS. És necessari l'ús d'una targeta SIM d'un operador de telefonia amb cobertura GPRS.



Com els models anteriors, també incorpora les funcionalitats tant de client com de servidor web. En aquest cas, el mòdul permet enviar i rebre SMS i fins i tots realitzar i rebre trucades telefòniques.

5.1.3 Altres models

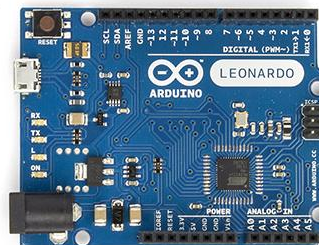
- Arduino UNO

Primera placa Arduino. És la més utilitzada. Té 14 entrades/sortides digitals (6 d'elles com a sortides PWM) i 6 entrades/sortides analògiques.



- Arduino Leonardo

Evolució de l'Arduino UNO. Té 20 entrades/ sortides digitals (7 d'elles com a sortides PWM) i 12 entrades/sortides analògiques. Incorpora un port micro USB per programar i alimentar la placa.



- Arduino Mini

Placa de reduïdes dimensions(48 x 18 mm) i pes (13 g). Té 20 entrades/sortides digitals (7 d'elles com a sortides PWM) i 12 entrades/sortides analògiques. Incorpora un port micro USB per programar i alimentar la placa.



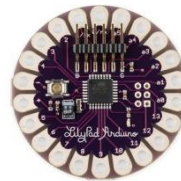
- Arduino Mega

Placa Arduino que integra major nombre de senyals. Té 54 entrades/sortides digitals (15 d'elles com a sortides PWM) i 16 entrades/sortides analògiques.



- Arduino LilyPad

Placa preparada per a projectes de tipus wereables i e-textiles i que permet cosir-se a la roba. Té 14 entrades/ sortides digitals (6 d'elles com a sortides PWM) i 6 entrades/sortides analògiques.



- Arduino Robot

Primer robot Arduino. Està format per 2 plaques amb un processador a cada placa. També integra 2 motors, 2 rodes i una pantalla. Té 4 entrades/sortides digitals (1 d'elles com a sortida PWM) i 4 entrades/sortides analògiques.



- I d'altres :
 - Arduino Fio
 - Arduino Esplora
 - Arduino Due
 - Arduino Mini
 - Intel Galileo (placa de 32 bits creada per Intel i certificada per Arduino)

5.2 Tecnologies mòbils

Pel que fa a la tecnologia mòbil, aquesta es basa en una aplicació desenvolupada en JAVA i amb el suport del SDK d'Android. Aquest últim, permet accedir a la memòria del dispositiu per enregistrar les dades capturades de l'Arduino, als serveis de reconeixement i síntesi de veu i a les comunicacions inalàmbriques. Mitjançant aquestes comunicacions inalàmbriques, es realitza la interacció amb el servidor web de l'Arduino, per tal d'enviar-li ordres i rebre'n informació.

6. Riscos del projecte

Cal considerar els riscos que existeixen a l'inici del projecte i els que poden sorgir durant la seva elaboració, per tal de prendre les mesures preventives i correctives pertinents.

Risc	Descripció	Probabilitat d'aparició	Impacte	Accions mitigadores
Pèrdua d'informació	La pèrdua d'informació a conseqüència d'un problema de hardware o un esborrament involuntari.	Mitjà	Crític	Desenvolupament sobre plataforma Google Drive. Còpia de seguretat en dispositiu extern.
Avaria ordinador	Impediment per desenvolupar el projecte degut a una avaria de l'ordinador utilitzat.	Mitjà	Alt	Disposar de equip de recanvi.
Càrrega laboral	Augment inesperat de la càrrega laboral que redueixi el temps de	Mitjà	Alt	Compensar-ho en un altre moment (si no es produeix a les etapes finals)

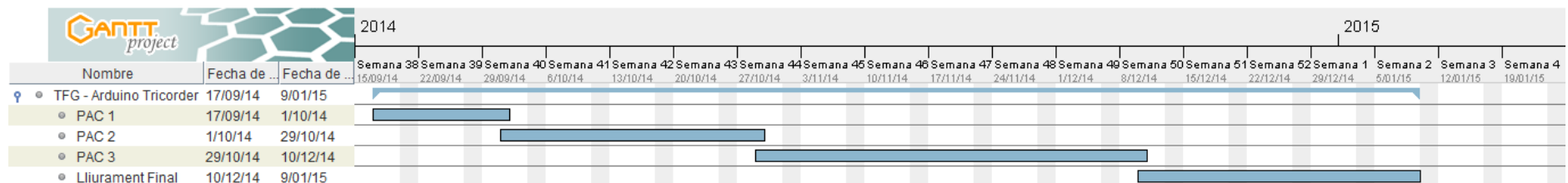
	dedicació al projecte.			
Manca de coneixements en desenvolupament Android	L'experiència en aquest tipus de desenvolupament es limitada i per tant pot provocar que no el pugui completar amb èxit.	Mitjà	Alt	La documentació existent és gran. Es poden seguir tutorials per adquirir els coneixements necessaris.
Planificació incorrecta o imprevist	Error en la planificació poden impedir la realització del projecte a temps.	Mitjà	Alt	Seguir estrictament el calendari previst. Avançar feina quan sigui possible.
Recursos insuficients	Manca de disponibilitat del hardware i software necessari per desenvolupar el projecte.	Baixa	Mig	Adquisició del material necessari.
Malaltia del desenvolupador	Període de malaltia que pugui alterar la planificació impeding la realització del projecte a temps.	Baixa	Mig	Avançar feina quan sigui possible.
Conciliació de la vida familiar	La prioritat que suposa la família, a més amb una ne-na petita, pot alterar la planificació impeding la realització del projecte a temps.	Baixa	Mig	Avançar feina quan sigui possible. Ampli suport familiar per compensar el temps dedicat al projecte.

7. Calendari del projecte

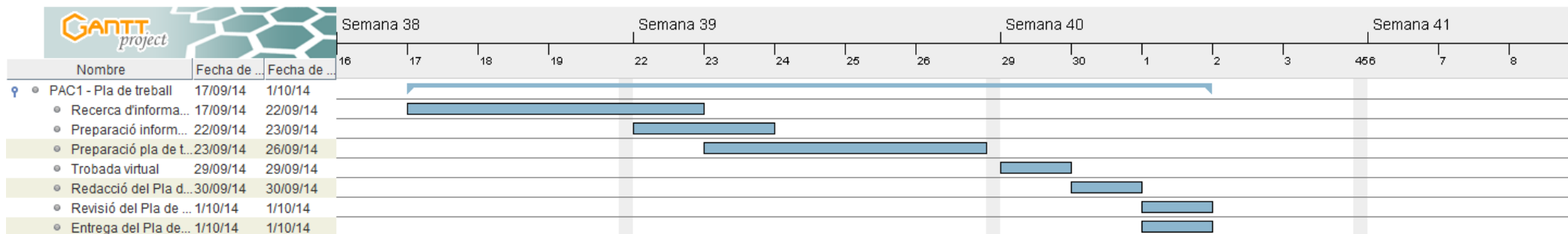
El projecte està format per 4 fases o PACS que segueixen l'estructura marcada pel cicle de vida clàssic de desenvolupament del software.

7.1 Lliuraments establerts

Data de lliurament	Fita avaluació continuada	Lliurable
01/10/2014	PAC 1	Pla de treball
29/10/2014	PAC 2	Anàlisi funcional, disseny tècnic i prototip
10/12/2014	PAC 3	Implementació
09/01/2015	Lliurament final	Memòria i vídeo-demo amb la presentació del projecte

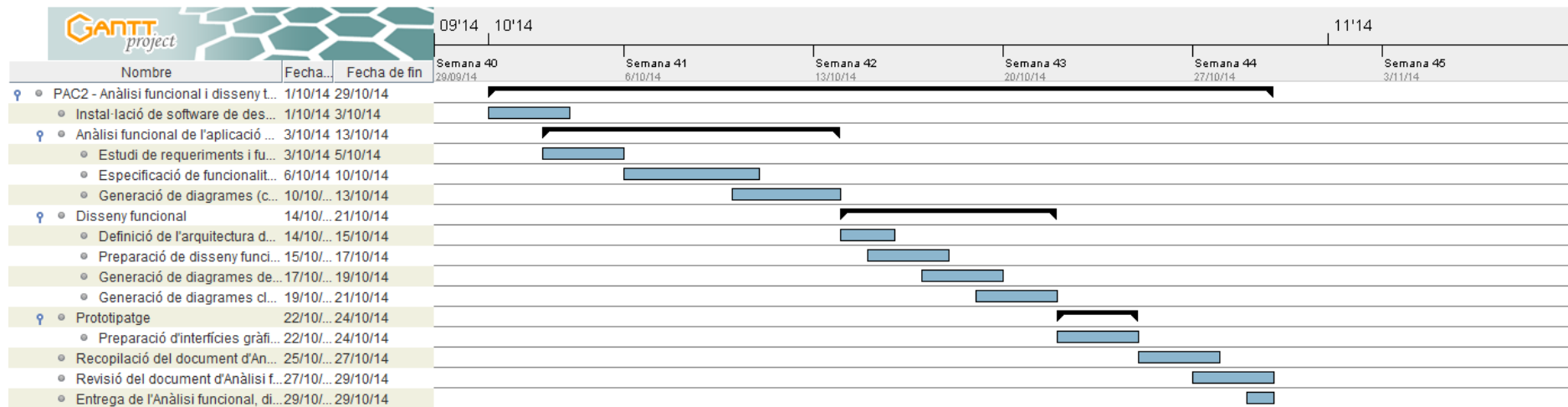


7.2 Detall de l'entrega de la PAC1



Nom de la tasca	Descripció de la tasca
Recerca d'informació	Recerca d'informació sobre el TFG, viabilitat, recursos, etc
Preparació informe previ	Redacció proposta de TFG per al consultor
Preparació Pla de Treball	Redacció preliminar de la justificació i objectius del projecte
Trobada virtual	Conversa a través de Google Hangouts amb el consultor i companys d'altres projectes
Redacció del Pla de Treball	Elaboració del Pla de Treball seguint les indicacions del consultor i l'exemple proporcionat
Revisió del Pla de Treball	Revisió del Pla de Treball abans d'entregar al consultor
Entrega del Pla de Treball	Entrega del Pla de Treball al consultor

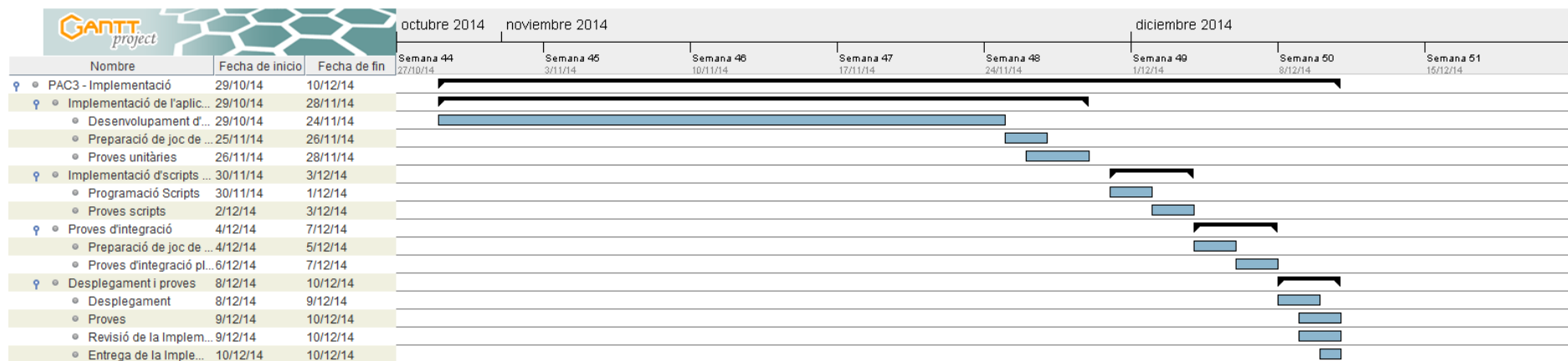
7.3 Detall de l'entrega de la PAC2



Nom de la tasca	Descripció de la tasca
Instal·lació de software de desenvolupament	Instal·lació del Eclipse ADT Bundle i Arduino IDE
Anàlisi funcional de l'aplicació mòbil	Tasques relacionades amb l'anàlisi funcional de l'aplicació mòbil
Estudi de requeriments i funcionalitats	Anàlisi exhaustiu de les funcionalitats que ha de proporcionar l'aplicació mòbil
Especificació de funcionalitats	Redacció de les funcionalitats i requeriments obtinguts a la fase d'estudi
Generació de diagrames (casos d'ús, etc.)	Generació dels diagrames que representen els casos d'ús de l'aplicació mòbil

<i>Disseny funcional</i>	<i>Tasques relacionades amb la preparació del disseny funcional</i>
Definició de l'arquitectura del sistema	Definició de l'arquitectura del sistema de comunicació bidireccional entre dispositius mòbils i l'Arduino
Preparació de disseny funcional	Definició de les classes de l'aplicació mòbil. Descripció dels scripts d'exemple d'Arduino que es faran servir.
Generació de diagrames de base de dades	Diagrama de la base de dades de l'aplicació mòbil
Generació de diagrames classes	Generació del diagrames de classes amb l'eina Magic Draw
<i>Prototipatge</i>	<i>Tasques relacionades amb el prototipatge de l'aplicació mòbil</i>
Preparació d'interfícies gràfiques	Disseny d'interfícies gràfiques de l'aplicació mòbil
Recopilació del document d'Anàlisi funcional, disseny tècnic i prototip	Elaboració del document que inclou la informació obtinguda en les fases prèvies
Revisió del document d'Anàlisi funcional, disseny tècnic i prototip	Revisió del document a entregar al consultor
Entrega de l'Anàlisi funcional, disseny tècnic i prototip	Entrega al consultor del document que inclou l'anàlisi funcional, disseny tècnic i prototip

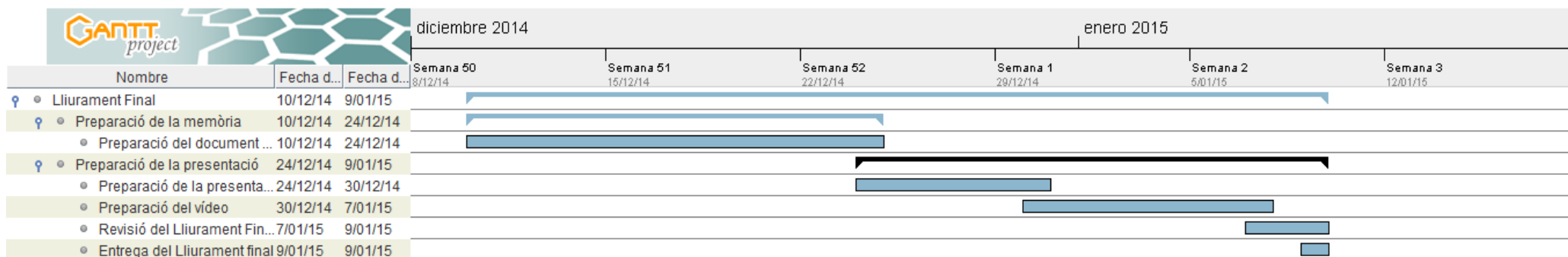
7.4 Detall de l'entrega de la PAC3



Nom de la tasca	Descripció de la tasca
Implementació de l'aplicació mòbil	Tasques relacionades amb la programació de l'aplicació mòbil
Desenvolupament d'aplicació Android	Implementació de l'aplicació mòbil
Preparació de joc de proves	Especificació del joc de proves necessari per avaluar el correcte funcionament de l'aplicació
Proves unitàries	Execució de les proves definides
Implementació d'scripts Arduino	Programació d'una llibreria i un script d'exemple per a la publicació de dades en format JSON al webser de l'Arduino

Proves scripts	Comprovar funcionament scripts
Proves d'integració	Tasques relacionades amb les proves d'integració de tots els components del projecte
Preparació de joc de proves d'integració	Especificació del joc de proves necessari per avaluar la correcta interacció entre les plataformes
Proves d'integració plataforma Arduino / plataforma mòbil	Execució de les proves definides
Desplegament i proves	Tasques relacionades amb el desplegament del projecte i les proves finals
Desplegament	Execució del desplegament i posada en producció del projecte
Proves	Execució de les proves finals
Revisió de la Implementació	Recopilació i revisió de la documentació recollida durant la implementació
Entrega de la Implementació	Lliurament del producte

7.5 Detall de l'entrega del Lliurament Final



Nom de la tasca	Descripció de la tasca
Preparació de la memòria	Tasques relacionades amb la preparació de la memòria
Preparació del document formal	Recopilació i revisió de la documentació recollida durant tot el projecte
Preparació de la presentació	Tasques relacionades amb la preparació de la presentació
Preparació de la presentació Powerpoint	Preparació de la presentació virtual del projecte en format Powerpoint
Preparació del vídeo	Preparació de la presentació en format vídeo
Revisió del Lliurament Final	Revisió de la documentació final del projecte
Entrega del Lliurament final	Lliurament de la documentació final del projecte

8. Anàlisi funcional

A continuació descriu el funcionament general de l'aplicació a través dels requeriments funcionals i no funcionals, juntament amb els casos d'ús.

8.1 Requeriments funcionals

Com s'ha comentat anteriorment, aquesta aplicació pretén facilitar la comunicació bidireccional entre dispositius Android i el microcontrolador Arduino a través del protocol IP, ja sigui via Ethernet o Wifi.

Per tal de realitzar aquesta comunicació l'aplicació permet dissenyar interfícies de diferents tipus :

- Interfície tàctil : l'usuari dissenya una interfície afegint diferents controls per canviar els valors de les sortides i per visualitzar els valors de les entrades de l'Arduino.
- Comandes de veu : l'usuari associa el canvi de valors de determinades sortides al reconeixent d'una determinada paraula o paraules.
- Síntesi de veu : l'usuari indica quines entrades de l'Arduino vol monitoritzar i l'interval de temps en que es fan les lectures. L'aplicació reproduïx amb síntesi de veu els valors obtinguts.
- Enregistrament de dades : l'usuari indica quines entrades de l'Arduino vol monitoritzar i l'interval de temps en que es fan les lectures. L'aplicació enregistra les dades en un fitxer CSV a la memòria del telèfon.

8.2 Requeriments no funcionals

En aquest apartat comentarem alguns elements que si bé no afecten directament al funcionament de l'aplicació, cal tenir-los en compte :

8.2.1 Interfície

Degut a que una part important de l'aplicació es basa en la generació d'interfícies dinàmiques, s'ha optat per fer ús principalment de l'estructura LinearLayout. Aquesta simplifica la tasca d'afegir controls en temps d'execució, i a més, s'adapta millor a tot tipus de dispositius (mòbils, tablets, android TV).

Tot i l'existència de 4 modalitats de projectes diferents, s'ha seguit una línia comuna per a tot ells. Les interfícies tenen un disseny net i clar, amb icones auto-explicatives.

8.3 Funcionalitats del sistema

Un cop descrites les funcionalitats principals, anem a descriure amb més detall quines tasques requereixen cadascuna d'elles.

8.3.1 Plataforma mòbil

L'aplicació per a dispositius mòbils Android permet realitzar les següents accions :

- Crear projecte : aquesta funció permet crear una interfície per comunicar-se amb una placa Arduino. Es pot triar entre 4 tipus de projectes, en funció de si es vol supervisar, controlar o adquirir dades, i si es vol que la interfície sigui tàctil o verbal.
- Controlar sortides digitals i analògiques de l'Arduino : aquesta funció permet activar o desactivar sortides digitals i canviar els valors de les sortides analògiques d'una placa Arduino. Si es tria una interfície tàctil, les ordres es donen mitjançant els controls de la pantalla. Si es tria la interfície verbal, les ordres es donen verbalment.
- Supervisar entrades digitals i analògiques de l'Arduino: aquesta funció permet visualitzar a la pantalla del dispositiu mòbil l'estat de les entrades de l'Arduino. Si es tria la interfície verbal, els valors són reproduïts per síntesi de veu.
- Enregistrar dades : aquesta funció permet enregistrar dades dels sensors de l'Arduino a la memòria del dispositiu en format CSV.
- Modificar i eliminar projectes : aquesta funció permet modificar les dades d'un projecte o bé eliminar-lo.
- Modificar i eliminar controls : aquesta funció permet modificar les dades dels elements que formen part de les interfícies creades o bé eliminar-los.

8.3.2 Plataforma Arduino

El microcontrolador Arduino és el dispositiu que empren per supervisar i controlar l'entorn a través dels sensors i actuadors que hi connectem. La connexió amb els dispositius Android és possible per als models d'Arduino que incorporen capacitats de comunicació ethernet o wifi. Aquests disposen d'un servidor web que permet rebre peticions via protocol HTTP.

Per tant, el servidor web de l'Arduino realitza les següents tasques :

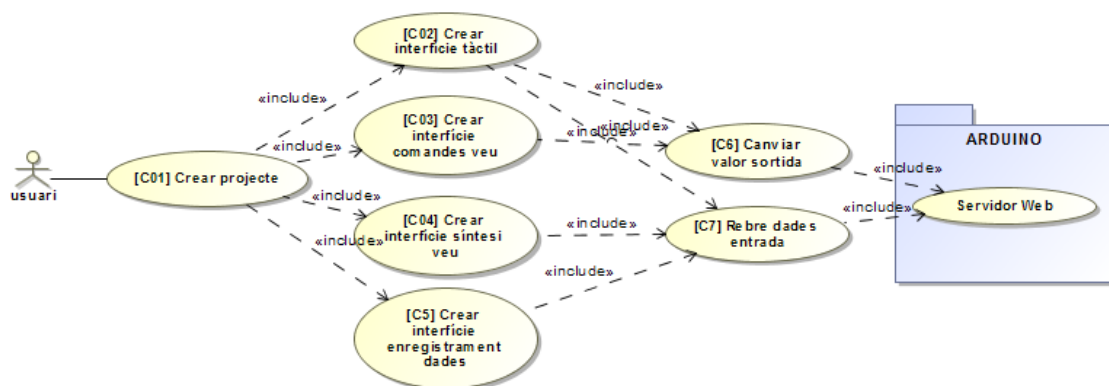
- Rebre peticions : aquesta funció permet a l'Arduino executar accions en funció de la petició rebuda.
- Publicar informació: aquesta funció fa que l'Arduino publiqui la informació dels sensors en format JSON, per a que sigui accessible per l'aplicació mòbil.

8.4 Usuaris del sistema

Cal destacar que aquesta aplicació està enfocada a treballar amb un dispositiu extern com és l'Arduino. La raó de ser de l'aplicació és fer possible aquesta interacció amb elements físics (sensors i actuadors) que estan connectats a l'Arduino. En aquest cas, només existeix un únic usuari del sistema.

8.5 Casos d'ús

A continuació es descriuran els casos d'ús de forma individualitzada, per a conèixer les seves implicacions i funcionament.



8.5.1 Descripció cas d'ús [C01] Crear projecte

Identificador	C01
Nom	Crear projecte
Descripció	L'usuari vol crear un projecte nou
Actors	Usuari
Pre-condició	Cap
Post-condició	<ul style="list-style-type: none"> - L'usuari crea un projecte. - L'usuari cancel·la la creació de la interfície.

Flux principal	<ol style="list-style-type: none"> 1) L'usuari prem la icona per crear un nou projecte 2) El sistema mostra una pantalla emergent per a introduir les dades del projecte 3) L'usuari emplena el formulari i demana al sistema que guardi les dades 4) El sistema verifica que les dades són vàlides 5) Si les dades són vàlides les guarda a la base de dades i mostra la pantalla amb el llistat de projecte
Flux alternatiu	<ul style="list-style-type: none"> - L'usuari cancel·la l'operació i no guarda el projecte 5b) Si les dades no són vàlides el sistema mostra un missatge d'error i torna al pas 2

8.5.2 Descripció cas d'ús [C02] Crear interfície tàtil

Identificador	C02
Nom	Crear interfície tàtil
Descripció	L'usuari vol crear una interfície d'interacció tàtil amb l'Arduino
Actors	Usuari
Pre-condició	L'usuari ha creat prèviament un projecte
Post-condició	<ul style="list-style-type: none"> - L'usuari crea el disseny d'una interfície amb controls tàctils - L'usuari cancel·la la creació de la interfície tàtil
Flux principal	<ol style="list-style-type: none"> 1) L'usuari prem la icona per afegir controls a la interfície 2) El sistema mostra una pantalla amb un formulari per seleccionar el tipus de control a afegir 3) L'usuari emplena el formulari i demana al sistema que guardi les dades 4) El sistema verifica que les dades són vàlides 5) Si les dades són vàlides les guarda a la base de dades i mostra la pantalla amb el disseny actual de la interfície
Flux alternatiu	<ul style="list-style-type: none"> - L'usuari cancel·la l'operació i no guarda cap dada 5b) Si les dades no són vàlides el sistema mostra un missatge d'error i torna al pas 2

8.5.3 Descripció cas d'ús [C03] Crear interfície comandes veu

Identificador	C03
Nom	Crear interfície comandes veu
Descripció	L'usuari vol crear una interfície d'interacció verbal amb l'Arduino
Actors	Usuari
Pre-condició	L'usuari ha creat prèviament un projecte
Post-condició	<ul style="list-style-type: none"> - L'usuari crea el disseny d'una interfície amb controls verbals - L'usuari cancel·la la creació de la interfície.
Flux principal	<ol style="list-style-type: none"> 1) L'usuari prem la icona per afegir controls a la interfície 2) El sistema mostra una pantalla amb un formulari per escollir quines accions s'han de

	<p>realitzar quan es reconeguïn certes comandes verbals</p> <p>3) L'usuari emplena el formulari i demana al sistema que guardi les dades</p> <p>4) El sistema verifica que les dades són vàlides</p> <p>5) Si les dades són vàlides les guarda a la base de dades i mostra la pantalla amb el disseny actual de la interfície</p>
Flux alternatiu	<p>- L'usuari cancel·la l'operació i no guarda cap dada</p> <p>5b) Si les dades no són vàlides el sistema mostra un missatge d'error i torna al pas 2</p>

8.5.4 Descripció cas d'ús [C04] Crear interfície síntesi de veu

Identificador	C04
Nom	Crear interfície síntesi de veu
Descripció	L'usuari vol crear una interfície per rebre informació de l'Arduino i reproduir-la per veu
Actors	Usuari
Pre-condició	L'usuari ha creat prèviament un projecte
Post-condició	<ul style="list-style-type: none"> - L'usuari crea el disseny d'una interfície que reproduceix la informació dels sensors de l'Arduino - L'usuari cancel·la la creació de la interfície.
Flux principal	<p>1) L'usuari prem la icona per afegir controls a la interfície</p> <p>2) El sistema mostra una pantalla amb un formulari per escollir quines entrades vol monitoritzar, l'idioma de reproducció i l'interval de lectura.</p> <p>3) L'usuari emplena el formulari i demana al sistema que guardi les dades</p> <p>4) El sistema verifica que les dades són vàlides</p> <p>5) Si les dades són vàlides les guarda a la base de dades i mostra la pantalla amb el disseny actual de la interfície</p>
Flux alternatiu	<p>- L'usuari cancel·la l'operació i no guarda cap dada</p> <p>5b) Si les dades no són vàlides el sistema mostra un missatge d'error i torna al pas 2</p>

8.5.5 Descripció cas d'ús [C05] Crear interfície enregistrament de dades

Identificador	C05
Nom	Crear interfície enregistrament de dades
Descripció	L'usuari vol crear una interfície per enregistrar informació dels sensors de l'Arduino
Actors	Usuari
Pre-condició	L'usuari ha creat prèviament un projecte
Post-condició	<ul style="list-style-type: none"> - L'usuari crea el disseny d'una interfície que enregistra la informació dels sensors

	de l'Arduino
Flux principal	<ul style="list-style-type: none"> - L'usuari cancel·la la creació de la interfície. <ol style="list-style-type: none"> 1) L'usuari prem la icona per configurar la interfície d'adquisició de dades. 2) El sistema mostra una pantalla amb un formulari per escollir quines entrades vol enregistrar i l'interval de lectura. 3) L'usuari emplena el formulari i demana al sistema que guardi les dades 4) El sistema verifica que les dades són vàlides 5) Si les dades són vàlides les guarda a la base de dades i mostra la pantalla amb el disseny actual de la interfície
Flux alternatiu	<ul style="list-style-type: none"> - L'usuari cancel·la l'operació i no guarda cap dada 5b) Si les dades no són vàlides el sistema mostra un missatge d'error i torna al pas 2

8.5.6 Descripció cas d'ús [C06] Canviar valor sortida

Identificador	C06
Nom	Canviar valor sortida
Descripció	L'usuari vol controlar un actuador connectat a l'Arduino
Actors	Usuari
Pre-condició	L'usuari ha creat prèviament una interfície d'interacció amb l'Arduino
Post-condició	<ul style="list-style-type: none"> - L'usuari canvia el valor d'una sortida digital o analògica de l'Arduino.
Flux principal	<ol style="list-style-type: none"> 1) L'usuari obre un projecte de tipus tàctil o de tipus de comandes de veu 2) El sistema mostra la interfície del projecte seleccionat 3) L'usuari fa ús d'un control tàctil o d'una comanda de veu per modificar una sortida 4) El sistema reconeix l'acció sol·licitada i realitza la petició a l'Arduino
Flux alternatiu	<ul style="list-style-type: none"> - En cas de no existir comunicació wifi disponible entre el dispositiu mòbil i l'Arduino, la petició no es realitza 4b) L'usuari dóna una comanda verbal no reconeguda i no realitza la petició

8.5.7 Descripció cas d'ús [C07] Rebre dades entrada

Identificador	C07
Nom	Rebre dades entrada
Descripció	L'usuari vol rebre informació dels sensors de l'Arduino
Actors	Usuari
Pre-condició	L'usuari ha creat prèviament una interfície per rebre dades de l'Arduino

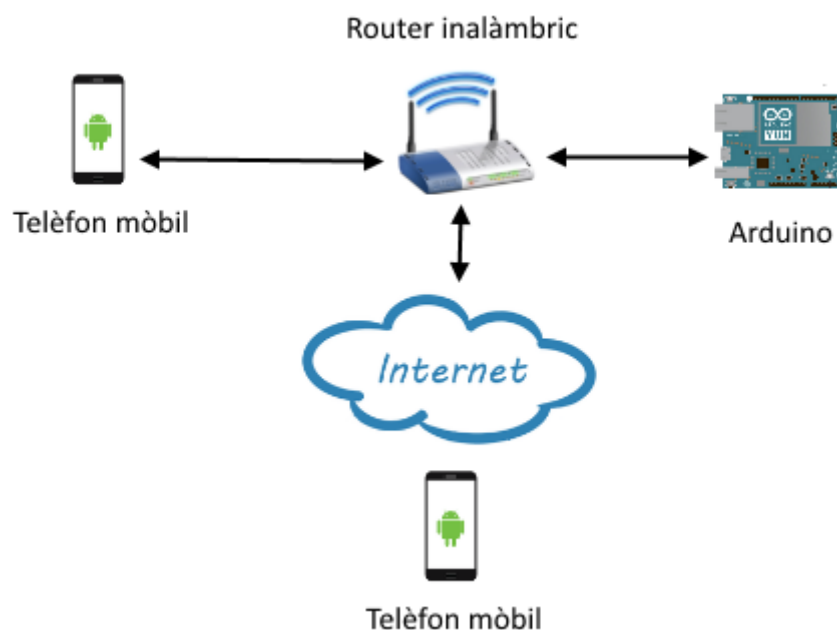
Post-condició	- L'usuari rep informació de l'estat d'una entrada digital o analògica de l'Arduino
Flux principal	1) L'usuari obre un projecte de tipus tàctil o de tipus síntesi de veu o enregistrament de dades. 2) El sistema mostra la interfície del projecte seleccionat 3) El sistema mostra en pantalla la informació de l'estat d'una entrada de l'Arduino. E n funció del projecte obert, la reproduïx verbal o l'enregistra a la memòria del dispositiu mòbil
Flux alternatiu	- En cas de no existir comunicació wifi disponible entre el dispositiu mòbil i l'Arduino, no es rep cap dada.

9. Disseny tècnic

L'arquitectura del projecte segueix clarament el model Client-Servidor. El disseny realitzat no contempla el cas en que un client pugui interactuar alhora amb més d'un servidor/Arduino des d'una sola interfície. Cada interfície que crea l'usuari està vinculada a la IP d'una sola placa Arduino.

Com a clients tenim els dispositius Android que fan ús de l'aplicació desenvolupada per enviar ordres al servidor web de la placa Arduino per a que aquest actuï sobre les sortides. I simultàniament o no, poden rebre dades que el mateix servidor web publica en format JSON a partir de les dades recollides pels sensors connectats a l'Arduino.

Els dispositius mòbils Android han d'estar a la mateixa xarxa que la placa Arduino, o bé, cal que es realitzi la configuració corresponent per a poder ésser accedit de d'una altra xarxa.



9.1 Arquitectura Arduino (part servidora)

En aquest projecte una placa Arduino dotada de capacitats de comunicació IP, ja sigui integrada o mitjançant un *shield* Ethernet o Wifi, és el dispositiu programable encarregat d'actuar com a servidor.

Per una banda, és capaç de rebre peticions dels clients mòbils i transformar-les en accions sobre el món real actuant sobre les sortides digitals i analògiques per modificar l'estat dels actuadors que té connectats. Els paràmetres rebuts pel servidor web que incorpora l'Arduino són analitzats per tal de determinar les accions corresponents sobre els pins de l'Arduino.

A la vegada, l'Arduino captura informació sobre l'entorn físic gràcies a les lectures dels sensors que té connectats, i la fa accessible aquesta informació convertint-la en format JSON i fent-la pública des del servidor web integrat.

Aquestes són les tasques bàsiques que ha de realitzar la placa Arduino per tal d'ajustar-se a les necessitats del projecte. Però, a banda d'aquestes, l'Arduino pot realitzar altres tasques autònomes. Per exemple, a la vegada que dona servei a l'aplicació Android tal i com hem explicat, podria activar i desactivar sortides en funció dels valors de les entrades (encendre i apagar llums automàticament en funció de la lluminositat, aturar un motor en funció de la distància a un obstacle, etc).

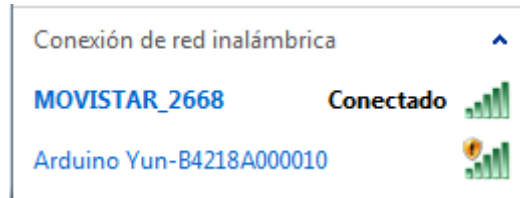
Així doncs, la placa Arduino té un rol passiu pel que fa al funcionament mínim requerit pel projecte, però que és perfectament compatible amb un rol més actiu. Tanmateix, caldrà tenir en compte possibles contradiccions. Per exemple, si intentem encendre una llum activant una sortida digital per tal d'activar un relé des de l'aplicació, però el programa carregat a l'Arduino té condicionat l'estat d'aquesta sortida al nivell de llum ambient, l'ordre serà immediatament neutralitzada.

Com a servidor, és indispensable que disposi d'una IP estàtica. En el cas de l'Arduino Ethernet i els shields Ethernet i Wifi, utilitzen unes llibreries que permeten especificar l'adreça IP de l'Arduino en el mateix sketch que es descarrega a la placa. En canvi, a l'Arduino Yún això no és possible. En aquest cas, i gràcies al petit sistema operatiu Linux que incorpora, es poden configurar les interfícies de xarxa de la mateixa manera que es fa en qualsevol distribució Linux. Una altra manera, és indicar a la configuració del router que assigni una IP estàtica a l'adreça MAC de la interfície Ethernet o Wifi de l'Arduino Yún.

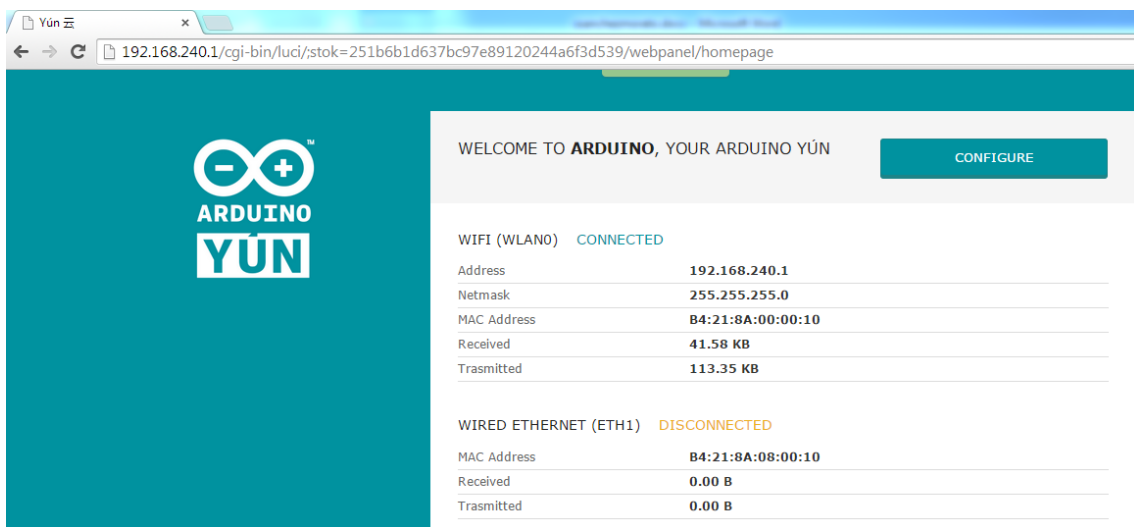
Per connectar la placa Arduino a una xarxa Ethernet tant sols hem de connectar un cable Ethernet entre la placa i el router o switch de la xarxa a la que el volem connectar, i especificar una IP dins del rang de la xarxa.

Per a la connexió Wifi, en el cas dels shields Wifi, les dades de connexió s'indiquen al sketch que es descarrega a la placa Arduino. En el cas de l'Arduino Yún, cal accedir via web a un apartat de configuració. Per accedir a aquesta configuració, en primer lloc cal connectar

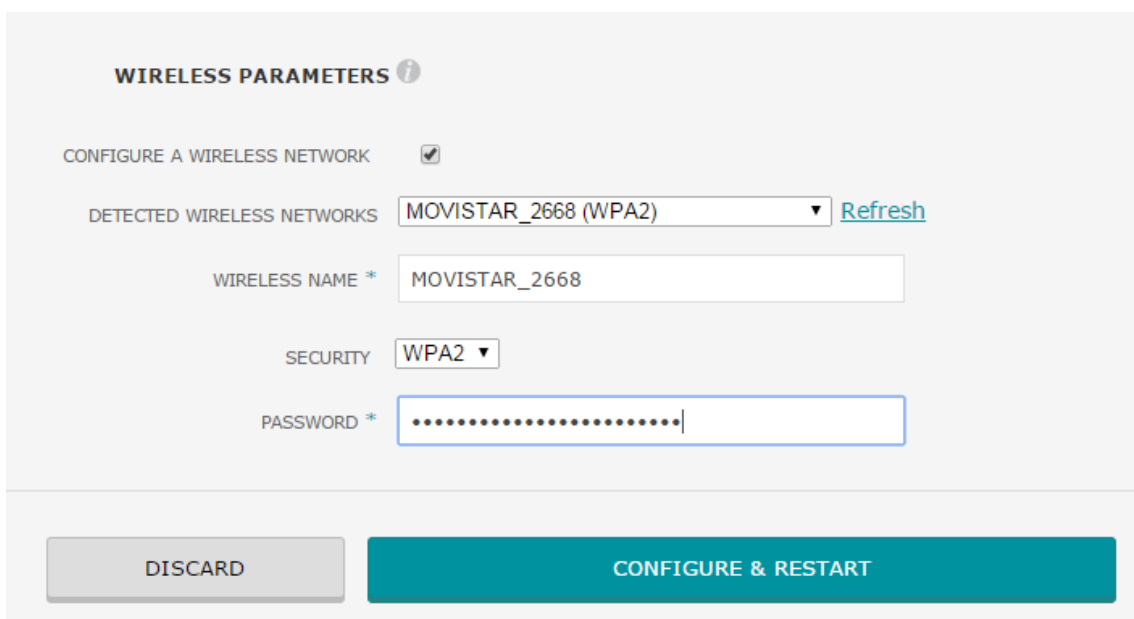
l'Arduino Yún a la xarxa amb un cable Ethernet, o bé, cercar-lo via Wifi, ja que aquest apareix com a un punt d'accés Wifi quan no troba cap xarxa a la que connectar-se.



Un cop localitzat, ens connectem a l'Arduino i posem la IP **192.168.240.1** al navegador per accedir a la configuració.

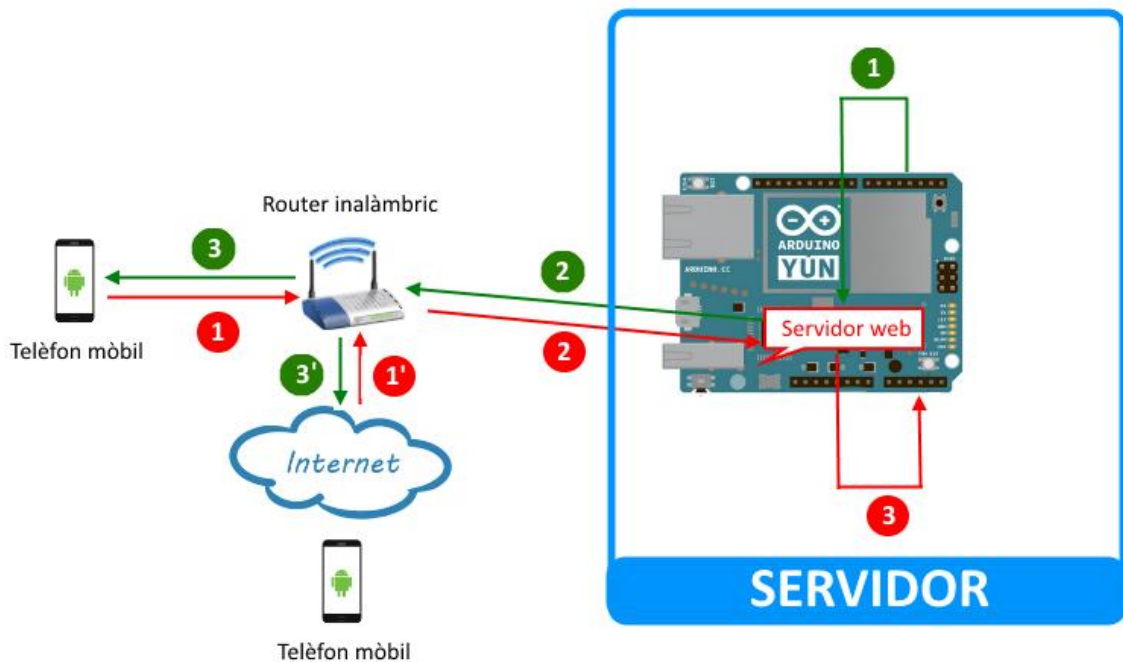


A continuació, accedim a l'apartat de configuració on indiquem el SSID i les credencials de la xarxa a la que s'ha de connectar l'Arduino.



Finalment, reiniciem l'Arduino per a que es connecti a la xarxa inalàmbrica especificada.

A banda de les connexions Ethernet i/o Wifi, els clients poden interactuar amb el servidor Arduino des de fora de la xarxa local a la que està connectat si aquest està dins d'una xarxa accés a internet. Per això, cal accedir a la configuració del router al que està connectat l'Arduino i redirigir un port cap a la IP d'aquest. A més, sinó es disposa d'una IP pública estàtica, caldrà emprar algun tipus de servei (NO-IP, etc) que permeti accedir a la xarxa objectiu a partir d'un identificador fix.



- **Control**

- **1** : El client mòbil realitza una petició al servidor web de l'Arduino indicant la IP i els paràmetres corresponents. Client i Servidor es troben a la mateixa xarxa local. La comunicació es pot realitzar mitjançant xarxa Ethernet o Wifi.
- **1'** : El client mòbil realitza una petició al servidor web de l'Arduino indicant la IP, un port i els paràmetres corresponents. Client i Servidor es troben en xarxes diferents. La comunicació és realitzada mitjançant xarxa 3G o 4G.
- **2** : El servidor web de la placa Arduino analitza els paràmetres rebuts.
- **3** : El programa carregat a la placa Arduino modifica l'estat de les sortides digitals o analògiques corresponent a la petició.

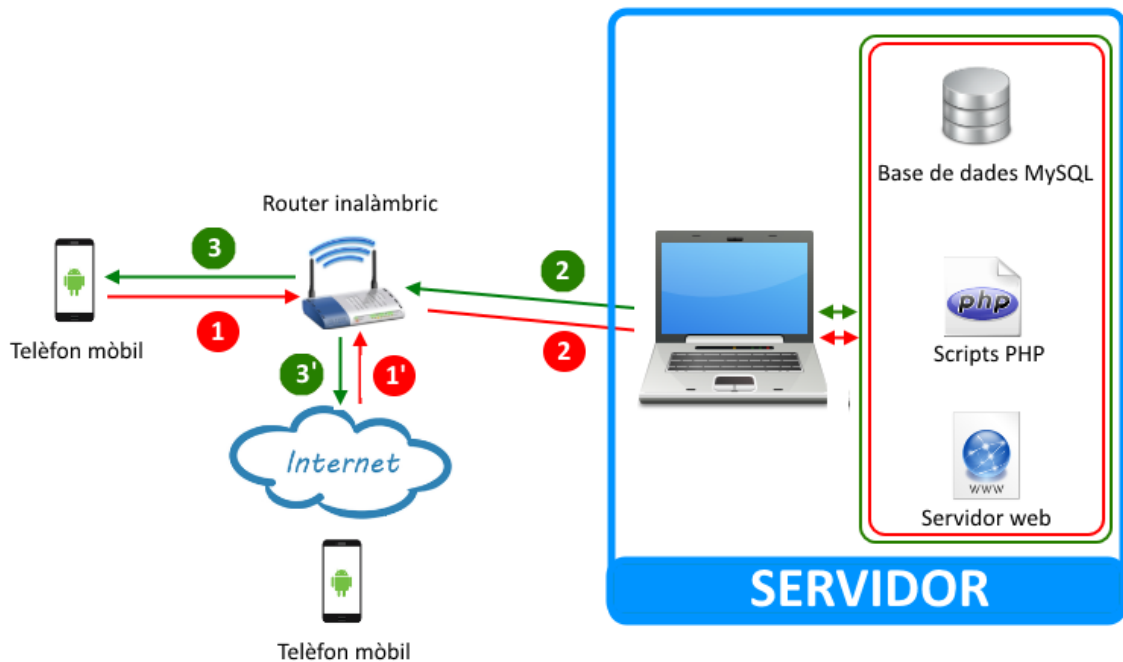
- **Supervisió i adquisició de dades**

- **1** : El servidor realitza lectures periòdiques de les entrades digitals i analògiques.

- **2** : El programa carregat a la placa Arduino converteix la informació obtinguda en format JSON i la publica al servidor web.
- **3** : El client es troba a la mateixa xarxa local i rep la informació sol·licitada. Tracta la informació en format JSON i la mostra de forma visual o verbal, o bé l'emmagatzema a la memòria del dispositiu.
- **3'** : El mateix que en punt 3, però des de fora de la xarxa local emprant connexió 3G o 4G.

9.2 Arquitectura proves (part servidora)

Per a fer més ràpida la tasca de provar l'aplicació, sobretot en la fase d'implementació, s'ha creat un entorn de proves digital. Aquest entorn pretén simular el servidor web de l'Arduino.



Per tal d'evitar haver de muntar múltiples circuits per validar la interacció entre l'aplicació i l'Arduino, es fa servir el servidor web Apache i el gestor de bases de dades MySQL que incorpora el paquet XAMPP.

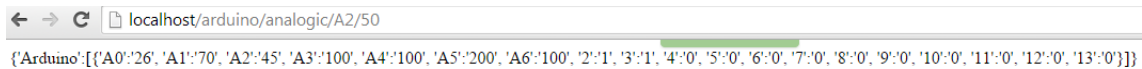
Així per simular l'enviament de peticions a l'Arduino emprant crides REST, s'ha creat un petit script PHP que fa insercions en una taula dels paràmetres de la crida. D'aquesta manera podem validar que les peticions es realitzen quan l'usuari prem els botons de la interfície o bé utilitza comandes verbals. Ha calgut modificar el fitxer http.conf de l'Apache i afegir un fitxer .htaccess en el directori on hi ha l'script PHP per fer la correspondència entre els valors del format REST i els paràmetres GET per tal de fer les insercions.

Regla ubicada en el fitxer .htacces per extreure els paràmetres de la crida REST de format (http://IP_ARDUINO/arduino/ [digital/analog]/[pin]/[valor]) :

RewriteRule ([^/]+)/([^/]+)/([^/]+) index.php?p1=\$1&p2=\$2&p3=\$3 [L]

El mateix script PHP retorna una cadena amb valors simulats per a les entrades de l'Arduino en format JSON. D'aquesta manera l'aplicació mòbil els pot recollir i mostrar en les diferents interfícies o bé reproduir aquests valors mitjançant una veu sintetitzada. També les consultes realitzades s'enregistren a la base de dades. En aquest cas els valors guardats són tots 0.

Exemple de les crides que realitza l'aplicació i els valors que recull en format JSON.



```
localhost/arduino/analogic/A2/50
{"Arduino":[{"A0":26, "A1":70, "A2":45, "A3":100, "A4":100, "A5":200, "A6":100, "2":1, "3":1, "4":0, "5":0, "6":0, "7":0, "8":0, "9":0, "10":0, "11":0, "12":0, "13":0}]}
```

Taula on es veuen les peticions per activar sortides i de recollida de dades de les entrades.

Mostrando filas 0 - 23 (total de 24, La

```
SELECT * FROM `peticions`
```

Número de filas: 25 ▼

+ Opciones

data	pin	valor
2014-12-09 22:13:10	13	1
2014-12-09 22:13:18	13	1
2014-12-09 22:13:23	13	0
2014-12-09 22:13:31	A3	0
2014-12-09 22:13:38	A3	10
2014-12-09 22:13:42	A3	30
2014-12-09 22:15:50	A2	50
2014-12-09 22:17:22	0	0
2014-12-09 22:17:22	0	0
2014-12-09 22:17:22	0	0
2014-12-09 22:17:22	0	0
2014-12-09 22:17:23	0	0

Per tal d'habilitar la comunicació entre el dispositiu mòbil i el servidor web Apache que emula el servidor web de l'Arduino, cal desactivar el tallafocs de Windows.

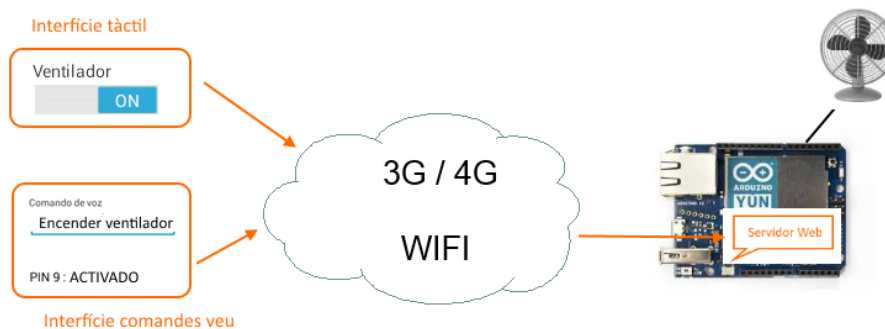
9.3 Arquitectura aplicació mòbil (part client)

La part mòbil esta formada per l'aplicació Android que permet als usuaris interactuar amb plaques Arduino dotades de capacitats de comunicació IP per tal de supervisar, controlar i adquirir dades de l'entorn on es troben.

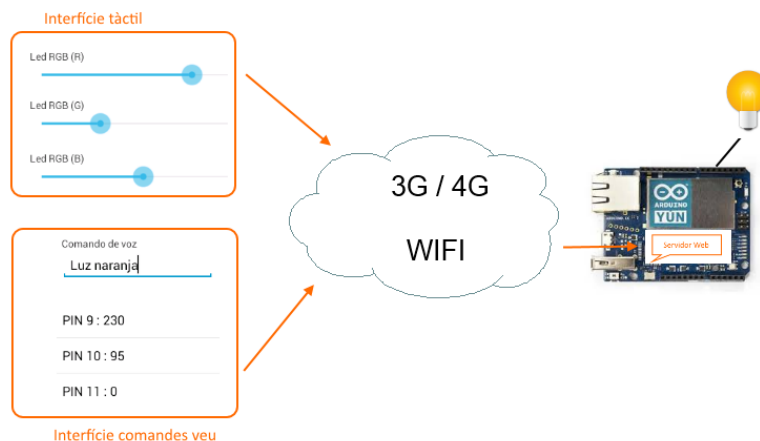
Per tant, els clients tenen un rol actiu. Aquests fan ús de les interfícies tàctils i verbals que poden crear a mida, per tal de controlar i monitoritzar remotament l'entorn físic on està ubicat la placa Arduino.

Diversos clients poden realitzar peticions al servidor de forma simultània. Així doncs, es pot donar el cas que des d'un mòbil es sol·liciti l'activació d'una sortida digital per encendre un motor, alhora que des d'un tablet s'està visualitzant els valors de temperatura i humitat d'un sensor connectat a un entrada digital de la mateixa placa Arduino.

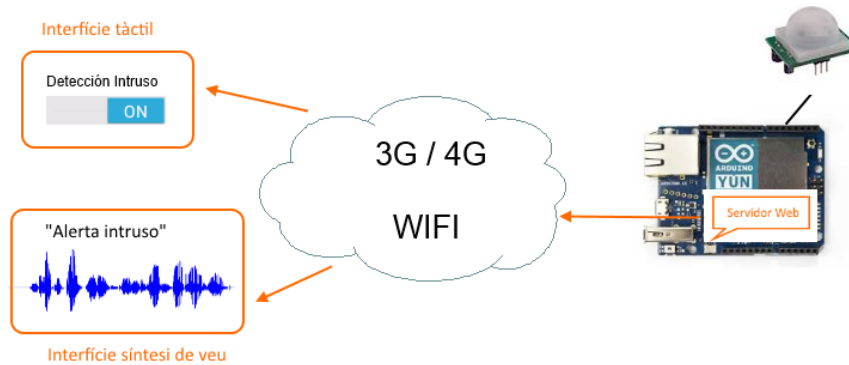
9.3.1 Control sortida digital



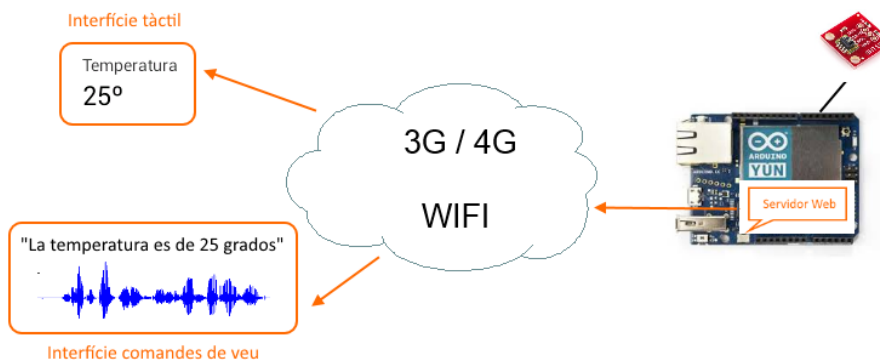
9.3.2 Control sortida analògica



9.3.3 Supervisió entrada digital



9.3.4 Supervisió entrada analògica



9.4 Arquitectura lògica

9.4.1 Arquitectura lògica de l'aplicació mòbil

Les característiques de l'aplicació permeten la divisió en 3 capes. D'una banda una capa de presentació que implementa la interfície gràfica, tant estàtica com dinàmica, aquesta última creant controls en temps d'execució. Una altra capa de gestió de dades, enfocada a la gestió de la base de dades SQLite creada al dispositiu. Finalment una tercera capa encarregada de les comunicacions amb el servidor web de l'Arduino.

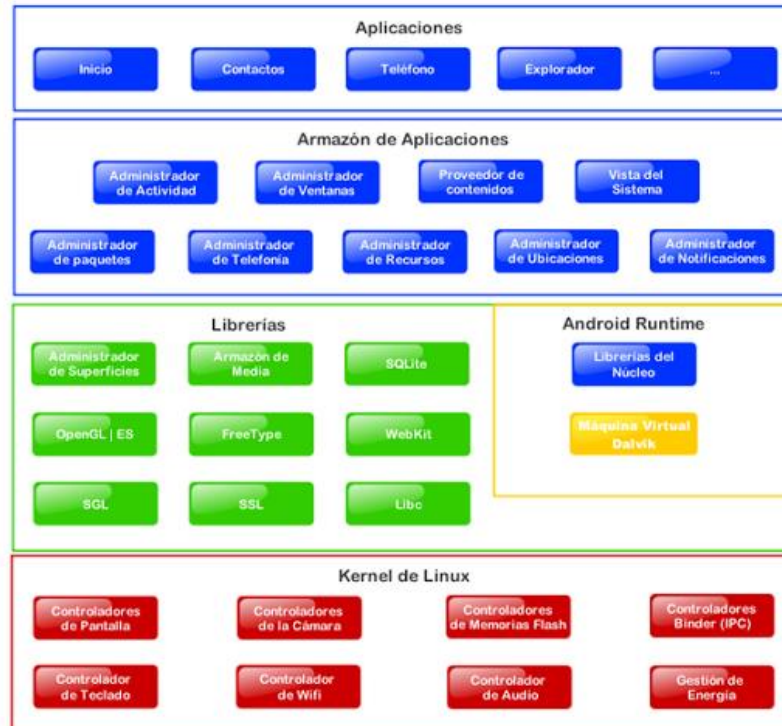
- Capa de presentació: Aquesta capa està formada per les Activitats que implementen les interfícies gràfiques de l'aplicació. L'estructura bàsica d'aquestes interfícies està composta per un LinearLayout i una ActionBar.
- Capa d'accés a les dades : Aquesta capa és l'encarregada de gestionar les operacions amb la base de dades. Permet emmagatzemar els dissenys de les interfícies creades a mida pels usuaris, i permet recuperar-los posteriorment per tal de crear interfícies dinàmiques.
- Capa de comunicacions : És la capa on es realitza la comunicació bidireccional amb l'Arduino. Per una banda es realitzen les peticions generades des de les diferents interfícies. Per l'altra es realitza el procés d'obtenció d'informació publicada en el servidor web de l'Arduino.

9.4.2 Arquitectura Android

Android és un sistema operatiu basat en el kernel de Linux dissenyat principalment per a dispositius mòbils com telèfons intel·ligents i tablets, tot i que també es troba present en altres dispositius com wereables, TV i d'altres. Inicialment va ser desenvolupat per Android Inc., però al 2005 va ser comprada per Google.

Android va ser presentat al 2007 juntament amb el Open Handset Alliance : un consorci d'empreses de maquinari, programari i telecomunicacions per avançar en el desenvolupament d'estàndards oberts per als dispositius mòbils.

A continuació podem veure una visió global per capes de l'arquitectura Android. Cadascuna d'aquestes capes utilitza serveis oferts per les capes superiors, tal i com mostra la figura següent :



- Aplicacions :
 - Aquest nivell conté totes les aplicacions instal·lades, tant les que venen pre-instal·lades, com les que instal·la posterior l'usuari. Totes aquestes aplicacions utilitzen els serveis, les API i les llibreries de nivells anteriors.
- Framework d'aplicacions :
 - Representa fonamentalment el conjunt d'eines de desenvolupament de qualsevol aplicació. Tota aplicació que es desenvolupa per a Android utilitza el mateix conjunt de API i el mateix framework.
- Llibreries :
 - Aquesta capa correspon a les llibreries utilitzades per Android. Aquestes han estat escrites en C/C++ i proporcionen a Android la major part de les seves capacitats més característiques. Juntament amb el nucli basat en Linux, aquestes llibreries constitueixen el cor d'Android.
- Runtime Android :
 - Al mateix nivell que les llibreries Android es situa l'entorn d'execució. Aquest el forment les "Core Libraries", que són llibreries amb multitud de classes Java i la màquina virtual Dalvik.
- Nucli Linux :
 - Android utilitza el nucli de Linux 2.6 com una capa d'abstracció per al hardware disponible en els dispositius mòbils. Aquesta capa conté els drivers necessaris per a que qualsevol component hardware pugui ser utilitzat mitjançant les comandes corresponents.

9.4.3 Arquitectura lògica de la plataforma Arduino

Tot i que aquest projecte es focalitza en l'aplicació mòbil, és necessari implementar algunes funcions per a fer operativa la interacció entre l'aplicació i les plaques Arduino amb comunicació IP.

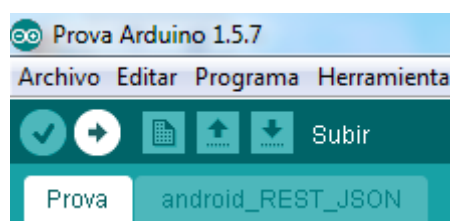
Per al cas de l'Arduino Yún, l'adaptació és mínima, ja que al ser el model més avançat actualment, està preparat per al processament de crides REST i l'execució de les corresponents accions sobre el hardware de la placa gràcies a la llibreria Bridge. Tanmateix, els sketches oficials no inclouen la possibilitat de publicar en el servidor web l'estat de les entrades digitals i analògiques en format JSON. És per aquesta raó, que s'ha hagut d'implementar aquesta funcionalitat.

Per a plaques anteriors, com l'Arduino Ethernet i els shields Ethernet i Wifi, s'ha hagut de desenvolupar a més a més una funció per al processament de crides REST equivalent a l'emprat per l'Arduino Yún.

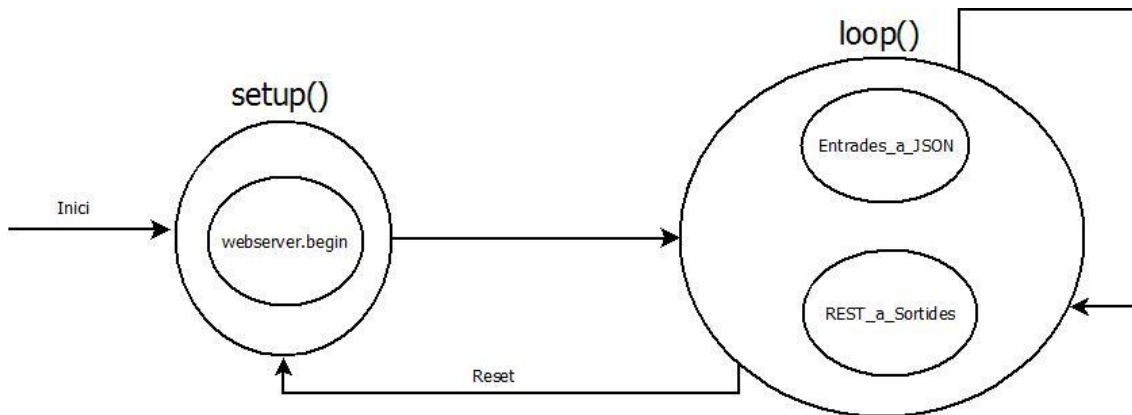
Així doncs, s'han implementat les següents 2 funcions dins d'un mateix sketch Arduino anomenat "**android_REST_JSON.ino**" :

- Entrades a JSON
 - Funció que llegeix totes les entrades digitals (digitalRead) i analògiques (analogRead) de la placa Arduino, les converteix en format JSON i les publica al servidor web de l'Arduino.
- REST a Sortides
 - Funció que processa els paràmetres de les crides REST obtinguts pel servidor web de l'Arduino i els transforma en accions sobre les sortides digitals (digitalWrite) i analògiques (analogWrite) de la placa Arduino.

Per a que les funcions incloses en aquest sketch puguin ser utilitzades des d'altres sketches, només és necessari copiar el fitxer "android_REST_JSON.ino" en el directori on es troben els altres sketches. Llavors a l'obrir l'sketch principal des de l'entorn de programació, ja s'aprecia com estan vinculats. El mateix succeeix al descarregar els sketches, ja que es descarreguen de cop tots els del directori.



Les funcions Entrades_a_JSON i REST_a_Sortides han de ser cridades des del bloc **loop** (s'executa cíclicament) de l'sketch principal.



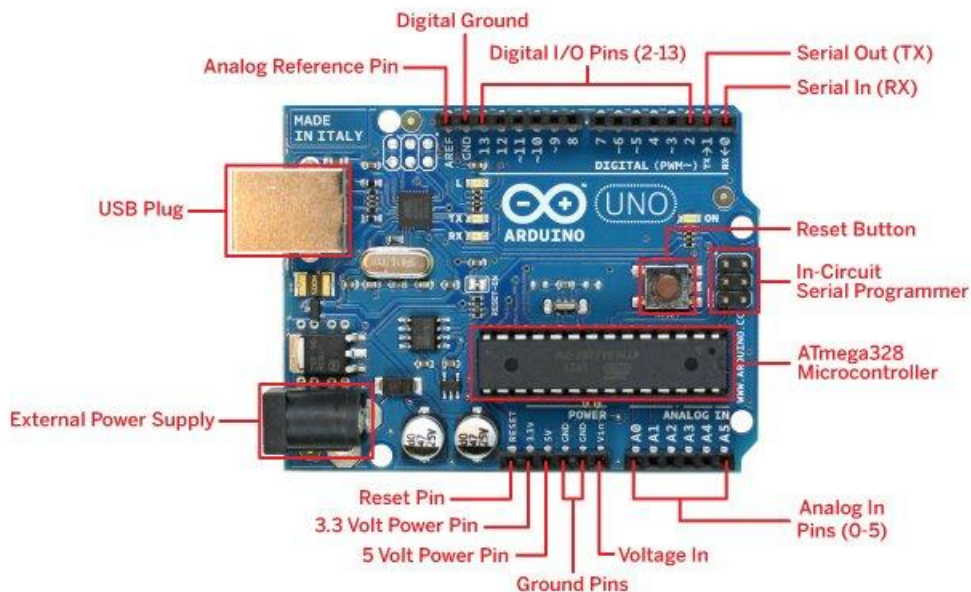
9.4.4 Arquitectura Arduino

Arduino és una plataforma de hardware lliure, basada en un conjunt de plaques microcontroladores, un entorn de desenvolupament i documentació.

9.4.4.1 Hardware

Les configuracions de les plaques varien depenent del model d'Arduino. Així podem trobar diferències en el nombre d'entrades i sortides, en el connector per descarregar els sketch (USB, micro-USB), en el connector per alimentar la placa (micro-USB, jack), en els voltatges d'entrada, en les comunicacions, etc.

A continuació podem veure la configuració més habitual, la de l'Arduino UNO, el model més popular i que és la base d'altres models més complexos.



9.4.4.2 Entorn de programació

El microcontrolador de la placa Arduino es programa mitjançant el llenguatge de programació Arduino (basat en Wiring) y l'entorn de desenvolupament Arduino (basat en Processing). El projectes fets amb Arduino es poden executar sense necessitat de connectar-se a un ordinador.

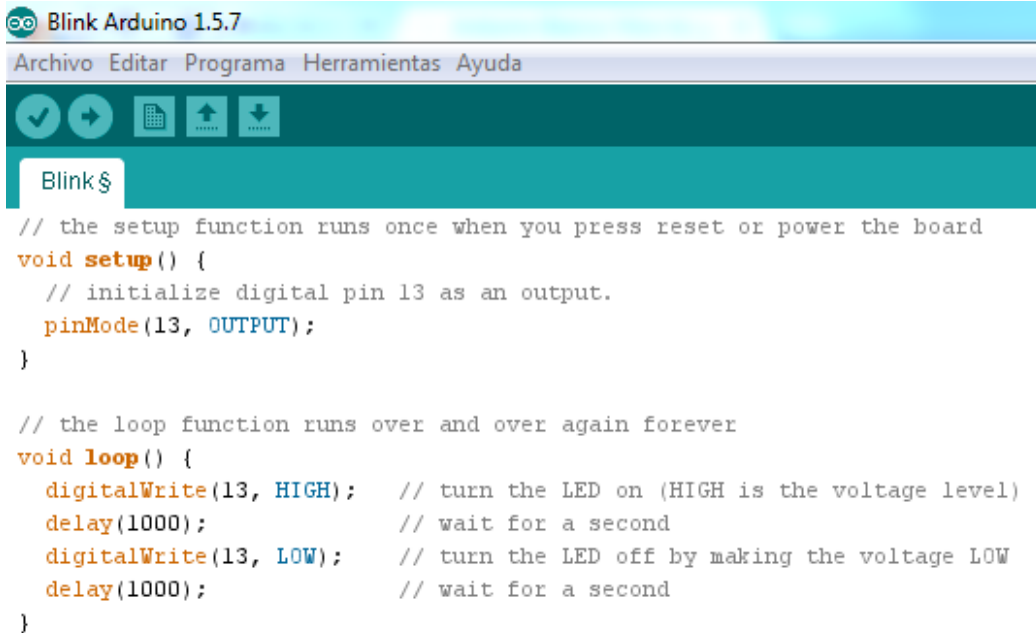
El gran èxit d'aquesta plataforma és degut, a banda del seu baix cost, a la facilitat de programació. Un programa o sketch Arduino necessita obligatòriament de 2 funcions :

- **void setup()**
 - S'executa a l'enegar-se la placa. En aquest punt s'acostumen a col·locar les definicions dels pins i s'inicien els diferents serveis (port sèrie, webserver, etc).
- **void loop()**
 - S'executa després del *setup*. Després de realitzar totes les operacions que conté es torna a executar cíclicament.

Per tal d'interactuar amb les entrades i sortides de les plaques Arduino, es disposa d'unes instruccions senzilles i intuïtives :

- **pinMode(PIN, MODE)**
 - Defineix si un pin actua com a entrada (INPUT) o sortida (OUTPUT).
 - Exemple : `pinMode(2, OUTPUT);` //defineix el pin 2 com a sortida
- **digitalWrite(PIN, VALUE)**
 - Activa (HIGH) o desactiva (LOW) un pin definit com a sortida digital.
 - Exemple : `digitalWrite(2, HIGH);` //activa la sortida digital del pin 2
- **digitalRead(PIN)**
 - Llegeix l'estat d'un pin definit com a entrada digital
 - Exemple : `digitalRead(2);` //llegeix l'estat de l'entrada digital del pin 2
- **analogWrite(PIN, VALUE)**
 - Canvia l'estat d'un pin definit com a sortida analògica
 - Exemple : `analogWrite(9, 255);` //carrega el valor 255 a sortida 9
- **analogRead(PIN)**
 - Llegeix l'estat del pin definit com a entrada analògica
 - Exemple : `analogRead(9);` //llegeix l'estat de l'entrada analògica del pin 9

A continuació podem veure un exemple típic per entendre el funcionament d'un sketch Arduino :

A screenshot of the Arduino IDE interface. The title bar reads "Blink Arduino 1.5.7". The menu bar includes "Archivo", "Editar", "Programa", "Herramientas", and "Ayuda". Below the menu bar is a toolbar with icons for a checkmark, a right arrow, a document, an upload arrow, and a download arrow. The main text area shows the code for the "Blink" sketch. The code is as follows:

```
Blink $  
  
// the setup function runs once when you press reset or power the board  
void setup() {  
  // initialize digital pin 13 as an output.  
  pinMode(13, OUTPUT);  
}  
  
// the loop function runs over and over again forever  
void loop() {  
  digitalWrite(13, HIGH); // turn the LED on (HIGH is the voltage level)  
  delay(1000);           // wait for a second  
  digitalWrite(13, LOW); // turn the LED off by making the voltage LOW  
  delay(1000);           // wait for a second  
}
```

A l'igual que en d'altres plataformes de programació, els sketches Arduino poden ampliar les seves capacitats gràcies a l'ús de llibreries. Aquestes poden ser implementades amb llenguatges com C i C++.

9.4.4.3 Documentació

Com a conseqüència de la gran popularitat de la plataforma Arduino, existeix abundant documentació disponible, sobretot a la xarxa. Així, a banda de la documentació oficial, les grans comunitats d'usuaris d'Arduino i els milers d'entusiastes d'aquesta plataforma posen a disposició de tothom una quantitat enorme d'informació (codi font, tutorials, videos, etc).

9.4.5 Arquitectura lògica d'intercanvi d'informació entre plataformes

Com hem vist anteriorment, l'aplicació mòbil es comunica amb el servidor web de l'Arduino per actuar sobre els elements físics connectats a ell (control) i per obtenir informació de l'entorn a partir dels sensors també connectats a l'Arduino (supervisió).

Per tal d'enviar ordres a l'Arduino, es fan servir crides REST, en les quals s'inclou el pin i el valor que aquest ha de prendre. Aquí podem veure alguns exemples de com es fan les crides, i com les processa posteriorment l'Arduino.

Crida REST	Comanda executada per l'Arduino
http://<IP_ARDUINO:PORT>/arduino/digital/13/1	digitalWrite(13, HIGH)
http://<IP_ARDUINO:PORT>/arduino/analog/2/123	analogWrite(2, 123)

En el cas de la supervisió, el programa de l'Arduino genera cada cop que se li fa una petició, un document en format JSON amb la informació dels sensors. Un cop rebuda per l'aplicació, aquesta la tracta fàcilment gràcies a l'ús d'aquest estàndard.

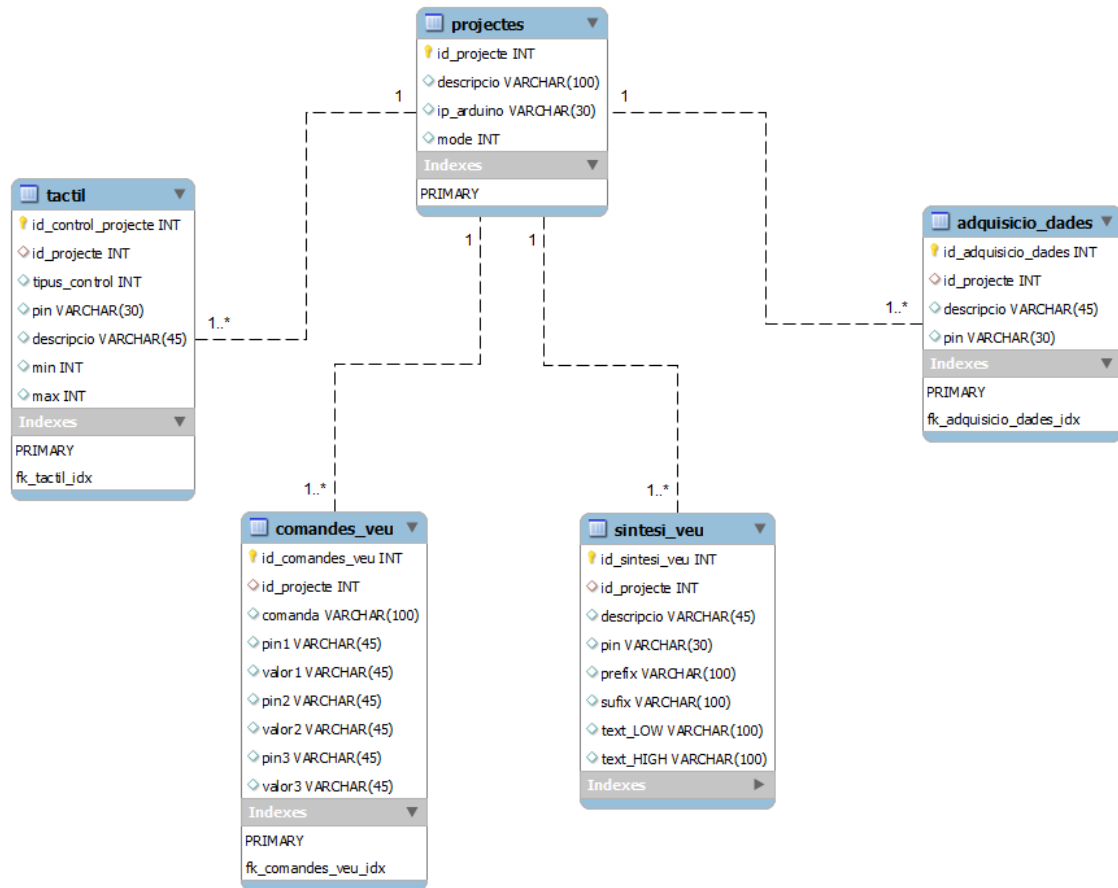
9.5 Arquitectura de base de dades

Els dissenys de les interfícies creades pels usuaris, així com els controls, comandes i les accions associades a aquests, s'emmagatzemen en una base de dades SQLite.

9.5.1 Model relacional de la base de dades

La informació dels dissenys dels usuaris s'emmagatzemen en una base de dades SQLite. Bàsicament és una taula de projectes i les taules de controls que componen la interfície. Degut a les diferències entre la interacció tàctil i la interacció verbal, les taules són diferents per a cada cas.

Per tant, quan un usuari seleccioni un projecte es crearà en temps d'execució una interfície a partir de la informació emmagatzemada a la base de dades.



Entitat lògica	Descripció	Cardinalitat	Entitat relacionada
projectes	Manté la persistència de les dades relacionades amb els projectes dels usuaris	1:N 1:N 1:N 1:N	tactil comandes_veu sintesi_veu adquisicio_dades
tactil	Manté la persistència de les dades relacionades amb els controls que formen les interfícies gràfiques tàctils	1:N	projectes
comandes_veu	Manté la persistència de les dades de la modalitat comandes de veu	1:N	projectes
sintesi_veu	Manté la persistència de les dades relacionades amb la modalitat síntesi de veu	1:N	projectes

adquisicio_dades	Manté la persistència de les dades relacionades amb la modalitat adquisició de dades	1:N	projectes
------------------	--	-----	-----------

9.5.2 Entitat Projectes

L'entitat Projectes té els següents atributs :

Atribut	Descripció	Tipus	Longitud	Clau
id_projecte	Identificador del projecte	Numèric	-	SI
descripcio	Nom per descriure el projecte	Text	45	NO
ip	IP de l'Arduino. Opcionalment es pot afegir un número de port.	Text	30	NO
tipus_projecte	Indica els tipus de projecte (0 : tàctil, 1 : comandes veu, 2 : síntesi veu, 3 : enregistrament de dades)	Numèric	-	NO

9.5.3 Entitat Tàctil

L'entitat Controls té els següents atributs :

Atribut	Descripció	Tipus	Longitud	Clau
id_control	Identificador del control	Numèric	-	SI
id_projecte	Identificador del projecte amb el està relacionat el control	Numèric	-	SI
tipus_control	Indica el tipus de control (0 : Switch, 1 : NumberPicker, etc)	Numèric	-	NO
descripcio	Nom per descriure el control. Apareix com a etiqueta que acompanya al control a la interfície gràfica	TEXT	45	NO
pin	Indica el pin de l'Arduino associat al control (A0,..., A5,2...,13)	TEXT	5	NO

min	Indica el valor mínim que pot prendre un control de tipus analògic	Numèric	-	NO
max	Indica el valor màxim que pot prendre un control de tipus analògic	Numèric	-	NO

9.5.4 Entitat Comandes_Veu

L'entitat Comandes té els següents atributs :

Atribut	Descripció	Tipus	Longitud	Clau
id_comanda	Identificador de la comanda	Numèric	-	SI
id_projecte	Identificador del projecte amb el està relacionat el control	Numèric	-	SI
comanda	Paraula/es associada a una acció o accions	TEXT	100	NO
pin1	Indica el pin de l'Arduino associat a la comanda (A0,..., A5,2...,13)	TEXT	10	NO
valor1	Valor que ha de prendre el pin1	Numèric	-	NO
pin2	Indica el pin de l'Arduino associat a la comanda (A0,..., A5,2...,13)	TEXT	10	NO
valor2	Valor que ha de prendre el pin2	Numèric	-	NO
pin3	Indica el pin de l'Arduino associat a la comanda (A0,..., A5,2...,13)	TEXT	10	NO
valor3	Valor que ha de prendre el pin3	Numèric	-	NO

9.5.5 Entitat Sintesi_veu

Atribut	Descripció	Tipus	Longitud	Clau
id_control	Identificador del control	Numèric	-	SI
id_projecte	Identificador del projecte amb el està relacionat el control	Numèric	-	SI

pin	Indica el pin de l'Arduino associat al control (A0,..., A5,2...,13)	TEXT	5	NO
descripcio	Nom per descriure el control. Apareix com a etiqueta que acompanya al control a la interfície gràfica	TEXT	45	NO
prefix	Text que precedeix la reproducció d'una valor analògic	TEXT	100	NO
sufix	Text que succeeix la reproducció d'una valor analògic	TEXT	100	NO
text_LOW	Text que es reproduueix quan una entrada digital no està activa	TEXT	100	NO
text_HIGH	Text que es reproduueix quan una entrada digital està activa	TEXT	100	NO

9.5.6 Entitat Aquisicio_dades

Atribut	Descripció	Tipus	Longitud	Clau
id_control	Identificador del control	Numèric	-	SI
id_projecte	Identificador del projecte amb el està relacionat el control	Numèric	-	SI
pin	Indica el pin de l'Arduino associat al control (A0,..., A5,2...,13)	TEXT	5	NO
descripcio	Nom per descriure el control. Apareix com a etiqueta que acompanya al control a la interfície gràfica	TEXT	45	NO

9.6 Diagrama de classes

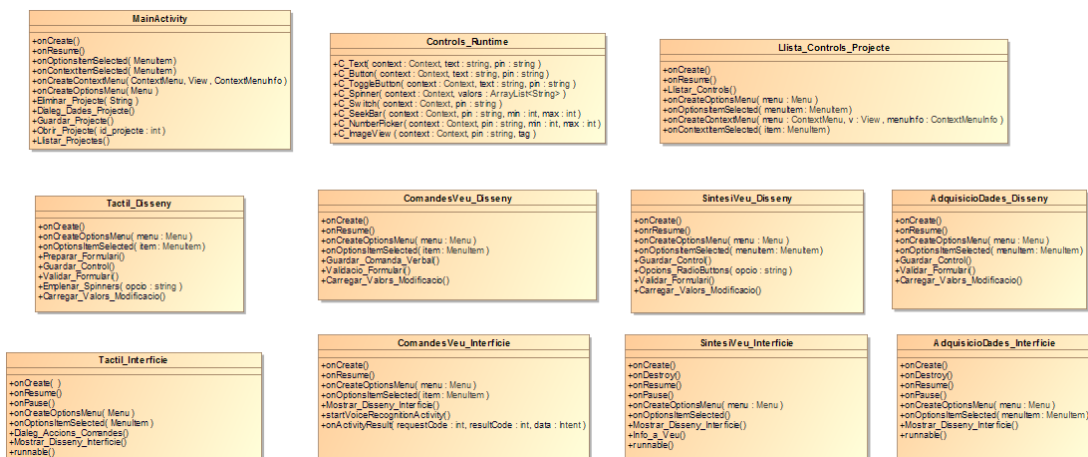
A continuació es presenten els diagrames de classes de l'aplicació mòbil. Els sketch Arduino desenvolupats són molts senzills i com és habitual en aquesta plataforma, no s'han implementat sota el paradigma de l'orientació a objectes.

9.6.1 Capa de presentació

Les classes d'aquesta capa la componen les anomenades Activities, que permeten generar la interfície gràfica. Aquí hi trobem la Activity de la pantalla principal, on apareix el llistat de projectes creats, les Activities per mostrar i dissenyar les interfícies de comunicació – porten el sufix “_interficie” i “_disseny” respectivament - i finalment l'Activity que permet modificar els controls de les esmentades interfícies.

Però com a particularitat important d'aquesta aplicació, gran part de les pantalles són dinàmiques, degut a que es generen a partir dels dissenys emmagatzemats a la base de dades. Aquesta capa conté un classe amb mètodes per a crear controls en temps d'execució i incrustar-los a les Activities amb sufix “_interficie”.

També des d'aquestes classes s'activen els serveis de reconeixement, síntesi de veu i enregistrament de dades.



A continuació es descriuen breument els mètodes de cada classe :

Classe	Mètode	Descripció
MainActivity	onCreate	Inicialització Activity
MainActivity	onResume	Quan Activity llesta, crida Llistar_Projetes
MainActivity	onCreateOptionsMenu	Afegeix icones a la ActionBar
MainActivity	onOptionsItemSelected	Gestiona la pulsació de les icones de la ActionBar
MainActivity	onCreateContextMenu	Afegeix un menú contextual a la llista de projectes
MainActivity	onContextItemSelected	Gestiona la pulsació de les opcions del menú contextual (modificar i eliminar)

MainActivity	Dialeg_Dades_Projecte	Obre una pantalla amb el formulari per donar d'alta un projecte
MainActivity	Obrir_projecte	Obre el disseny d'un projecte en una nova pantalla
MainActivity	Llistar_Projectes	Mostra una llista amb el projectes guardats
MainActivity	Guardar_Projecte	Guarda les dades d'un projecte a la base de dades
MainActivity	Eliminar_Projecte	Elimina un projecte de la base de dades
Tactil_Disseny	onCreate	Inicialització Activity. Capturar valors passats per altres Activities.
Tactil_Disseny	onCreateOptionsMenu	Afegeix icones a la ActionBar
Tactil_Disseny	onOptionsItemSelected	Gestiona la pulsació de les icones de la ActionBar (Guardar control)
Tactil_Disseny	Preparar_Formulari	Tasques per inicialitzar el formulari
Tactil_Disseny	Guardar_Control	Guardar control a la base de dades
Tactil_Disseny	Validar_Formulari	Validació del formulari abans de guardar dades
Tactil_Disseny	Emplenar_Spinners	Canviar el contingut dels desplegable en funció d'altres valors triats
Tactil_Disseny	Carregar_Valors_Modificació	Emplenar el formulari amb les dades d'un control a modificar
Tactil_Interficie	onCreate	Inicialització Activity. Capturar valors passats per altres Activities.
Tactil_Interficie	onResume	Cridar a Mostrar_Disseny_Interficie. Activar timer monitorització.
Tactil_Interficie	onPause	Aturar timer monitorització.
Tactil_Interficie	onCreateOptionsMenu	Afegeix icones a la ActionBar
Tactil_Interficie	onOptionsItemSelected	Gestiona la pulsació de les icones de la ActionBar (Afegir Controls i Llista controls interfície)
Tactil_Interficie	Mostrar_Disseny_Interficie	Obté de la base de dades els controls del projecte i els col·loca al Layout de l'Activity
Tactil_Interficie	Runnable	Procés en segon pla que captura informació de l'Arduino i la presenta a la pantalla
ComandesVeu_Disseny	onCreate	Inicialització Activity. Capturar valors passats per altres Activities.
ComandesVeu_Disseny	onCreateOptionsMenu	Afegeix icones a la ActionBar
ComandesVeu_Disseny	onOptionsItemSelected	Gestiona la pulsació de les icones de la ActionBar (Guardar comanda verbal)

ComandesVeu_Disseny	Guardar_Comanda_Verbal	Guardar comanda verbal i accions a la base de dades
ComandesVeu_Disseny	Validar_Formulari	Validació del formulari abans de guardar dades
ComandesVeu_Disseny	Carregar_Valors_Modificació	Emplenar el formulari amb les dades d'un control a modificar
ComandesVeu_Interficie	onCreate	Inicialització Activity. Capturar valors passats per altres Activities.
ComandesVeu_Interficie	onResume	Cridar a Mostrar_Disseny_Interficie.
ComandesVeu_Interficie	onCreateOptionsMenu	Afegeix icones a la ActionBar
ComandesVeu_Interficie	onOptionsItemSelected	Gestiona la pulsació de les icones de la ActionBar (Afegir comanda i Llista controls interfície)
ComandesVeu_Interficie	Mostrar_Disseny_Interficie	Obté de la base de dades les comandes del projecte i les col·loca al Layout de l'Activity
ComandesVeu_Interficie	startVoiceRecognitionActivity	Activa el sistema de reconeixement de veu
ComandesVeu_Interficie	onActivityResult	Gestiona els resultats obtinguts pel sistema de reconeixement de veu
SintesiVeu_Disseny	onCreate	Inicialització Activity. Capturar valors passats per altres Activities.
SintesiVeu_Disseny	onResume	Emplena el formulari en cas d'edició
SintesiVeu_Disseny	onCreateOptionsMenu	Afegeix icones a la ActionBar
SintesiVeu_Disseny	onOptionsItemSelected	Gestiona la pulsació de les icones de la ActionBar (Guardar control)
SintesiVeu_Disseny	Guardar_Control	Guardar control a la base de dades
SintesiVeu_Disseny	Validar_Formulari	Validació del formulari abans de guardar dades
SintesiVeu_Disseny	Opcions_Radiobuttons	Habilitat unes opcions o unes altres en funció dels radiobuttons seleccionats
SintesiVeu_Disseny	Carregar_Valors_Modificació	Emplenar el formulari amb les dades d'un control a modificar
SintesiVeu_Interficie	onCreate	Inicialització Activity. Capturar valors passats per altres Activities. Inicialització síntesi de veu.
SintesiVeu_Interficie	onResume	Cridar a Mostrar_Disseny_Interficie. Activar timer monitorització.
SintesiVeu_Interficie	onPause	Aturar timer monitorització.
SintesiVeu_Interficie	onDestroy	Aturar la síntesi de veu
SintesiVeu_Interficie	onCreateOptionsMenu	Afegeix icones a la ActionBar

SintesiVeu_Interficie	onOptionsItemSelected	Gestiona la pulsació de les icones de la ActionBar (Afegir Controls i Llista controls interfície)
SintesiVeu_Interficie	Mostrar_Disseny_Interficie	Obté de la base de dades els controls del projecte i els col·loca al Layout de l'Activity
SintesiVeu_Interficie	Runnable	Procés en segon pla que captura informació de l'Arduino i la presenta a la pantalla
SintesiVeu_Interficie	Info_a_Veu	Construeix una cadena de text amb la informació capturada de l'Arduino i la reproduïx
AdquisicioDades_Disseny	onCreate	Inicialització Activity. Capturar valors passats per altres Activities.
AdquisicioDades_Disseny	onResume	Emplena el formulari en cas d'edició
AdquisicioDades_Disseny	onCreateOptionsMenu	Afegiu icones a la ActionBar
AdquisicioDades_Disseny	onOptionsItemSelected	Gestiona la pulsació de les icones de la ActionBar (Guardar control)
AdquisicioDades_Disseny	Guardar_Control	Guardar control a la base de dades
AdquisicioDades_Disseny	Validar_Formulari	Validació del formulari abans de guardar dades
AdquisicioDades_Disseny	Carregar_Valors_Modificació	Emplenar el formulari amb les dades d'un control a modificar
AdquisicioDades_Interficie	onCreate	Inicialització Activity. Capturar valors passats per altres Activities.
AdquisicioDades_Interficie	onResume	Cridar a Mostrar_Disseny_Interficie. Activar timer monitorització.
AdquisicioDades_Interficie	onPause	Aturar timer monitorització.
AdquisicioDades_Interficie	onCreateOptionsMenu	Afegiu icones a la ActionBar
AdquisicioDades_Interficie	onOptionsItemSelected	Gestiona la pulsació de les icones de la ActionBar (Afegir Controls i Llista controls interfície)
AdquisicioDades_Interficie	Mostrar_Disseny_Interficie	Obté de la base de dades els controls del projecte i els col·loca al Layout de l'Activity
AdquisicioDades_Interficie	Runnable	Procés en segon pla que captura informació de l'Arduino i la presenta a la pantalla
AdquisicioDades_Interficie	Guardar_a_Fitxer	Afegiu els valors capturats a un fitxer
Controls_Runtime	C_Text	Control Text per afegir text descriptiu a cada component i visualitzar valors entrades analògiques
Controls_Runtime	C_Button	Control Button per activar o desactivar sortida digital

Controls_Runtime	C_Switch	Control Button per activar o desactivar sortida digital
Controls_Runtime	C_Spinner	Control Spinner per crear desplegable dinàmics
Controls_Runtime	C_SeekBar	Control SeekBar per canviar valors sortides analògiques
Controls_Runtime	C_NumberPicker	Control NumberPicker per canviar valors sortides analògiques
Controls_Runtime	C_ImageView	Control ImageView per mostrar estat d'entrades digitals
Controls_Runtime	Eliminar_Controls	Eliminar controls d'un layout segons Id

9.6.2 Capa de dades

Aquesta capa conté les classes encarregades de la interacció amb la base de dades i 3 classes que actuen com a estructures de dades.

Aquí hi trobem les classes encarregades de la creació de la base de dades SQLite i de gestionar les altes, baixes i modificacions dels projectes i els controls de les interfícies.

S'ha creat un classe per a simplificar l'accés a les dades d'un projecte creat per l'aplicació, des de qualsevol punt de la mateixa. Així mateix, les funcionalitats de síntesi i reconeixement de veu es recolzen en les propietats de classes d'aquesta capa.

```

package Data [ Capa de dades ]

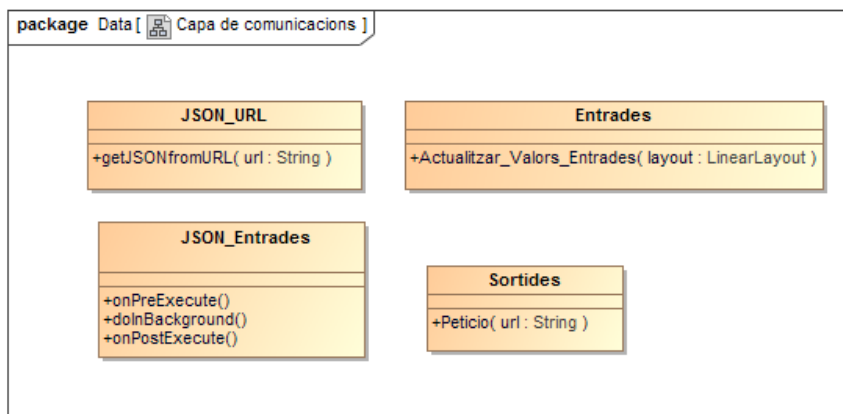
class AdminSQLiteOpenHelper {
+AdminSQLiteOpenHelper( context : Context, nom : String, factory : CursorFactory, versio : int )
+onCreate( db : SQLiteDatabase )
+onUpgrade( db : SQLiteDatabase )
}

class sDades {
+Guardar_Nou_Projecte( context : Context, descriptio : String, ip_arduino : String, mode : int )
+Guardar_Nou_Control( context : Context, id_projecte : int, tipus_control : int, pin : String, descriptio : String, min : int, max : int, descriptio_antic : String )
+Guardar_Comandes_Veu( context : Context, id_projecte : int, comanda : String, pin1 : String, valor1 : String, pin2 : String, valor2 : String, pin3 : String, valor3 : String )
+Guardar_Sintesi_Veu( context : Context, id_projecte : int, pin : String, descriptio : String, prefix : String, sufix : String, text_LOW : String, text_HIGH : String, descriptio_antic : String )
+Guardar_Adduicio_Dades( context : Context, id_projecte : int, descriptio : String, descriptio_antic : String, pin : String )
+SQL_Valors( context : Context, SQL : String )
+Executar_SQL( SQL : String )
}
    
```

Classe	Mètode	Descripció
AdminSQLiteOpenHelper	AdminSQLiteOpenHelper	Constructor de la classe
AdminSQLiteOpenHelper	onCreate	Crea la base de dades SQLite i les taules
AdminSQLiteOpenHelper	onUpgrade	Actualitza les taules
Dades	Guardar_Nou_Projecte	Insereix i actualitza un projecte
Dades	Guardar_Nou_Control	Insereix i actualitza un control
Dades	Guardar_Comandes_Veu	Insereix i actualitza un control
Dades	Guardar_Sintesi_Veu	Insereix i actualitza un control
Dades	Guardar_Adquisicio_Dades	Insereix i actualitza un control
Dades	SQL_Valor	Retorna un valor a partir d'una consulta SQL
Dades	Executar_SQL	Executa una comanda SQL
Projecte	---	---
TipusComanda	---	---
TipusVeu	---	---

9.6.3 Capa de comunicacions

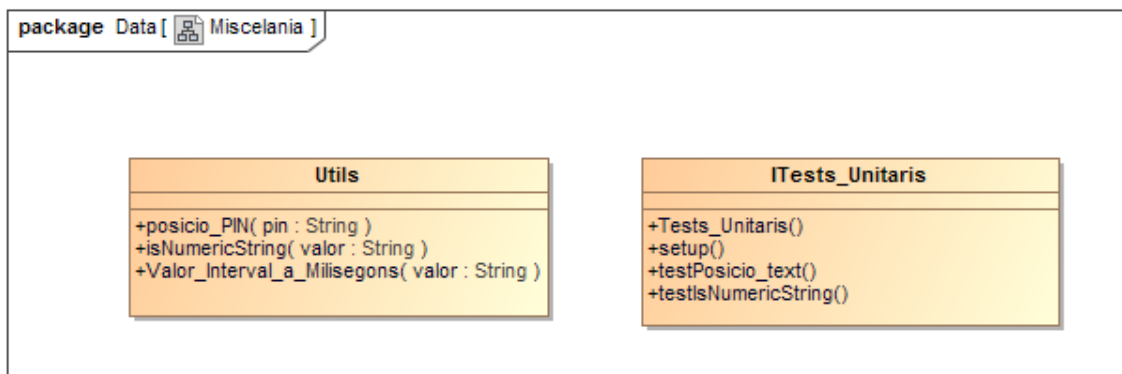
A la capa de comunicacions trobem les classes que intervenen en la comunicació bidireccional amb la placa Arduino. Per una banda tenim les classes encarregades de realitzar peticions al servidor web de l'Arduino mitjançant crides REST. I per l'altra tenim les classes encarregades de capturar la informació publicada al servidor web de l'Arduino i que es troba en format JSON.



Classe	Mètode	Descripció
JSON_URL	getJSONfromURL	Guarda en un objecte JSONObject el contingut d'una URL
JSON_Entrades	doInBackground	Tasca en segon pla que llegeix l'objecte JSONObject i emplena l'array d'entrades
Entrades	Actualitzar_Valors_Entrades	Actualitza els TextViews d'un layout a partir de l'array d'entrades
Sortides	Peticio	Realitza una crida REST

9.6.4 Miscel·lània

En aquest apartat he col·locat una classe que conté funcions que realitzen tasques de suport per d'altres classes i una classe per a la realització de tests unitaris.



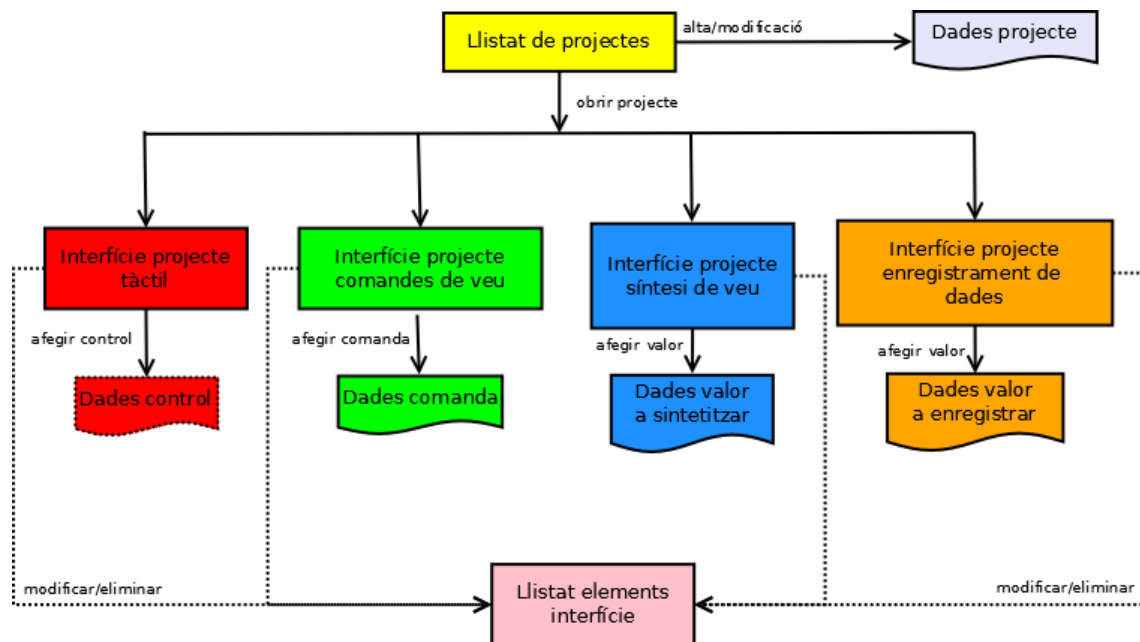
Classe	Mètode	Descripció
Utils	posicio_PIN	Calcula la posició d'un PIN dins l'array de pins
Utils	isNumericString	Valida si un string és numèric
Utils	Valor_Interval_a_Milisegons	Converteix el text d'un desplegable a milisegons
Test_Unitaris	testPosicio_text	Test del mètode posicio_PIN
Test_Unitaris	testIsNumericString	Test del mètode IsNumericString

10. Prototip

L'aplicació parteix d'una pantalla inicial on apareix una llista amb el nom dels projectes creats. Des d'aquesta pantalla es pot cridar una finestra emergent per donar d'alta un projecte nou o bé realitzar diferents tasques amb els projectes : modificar, eliminar i obrir. Si obrim un projecte, l'usuari seguirà per un dels quatre camins paral·lels corresponents als quatre tipus de projecte (tàctil, comandes de veu, síntesi de veu i enregistrament de dades).

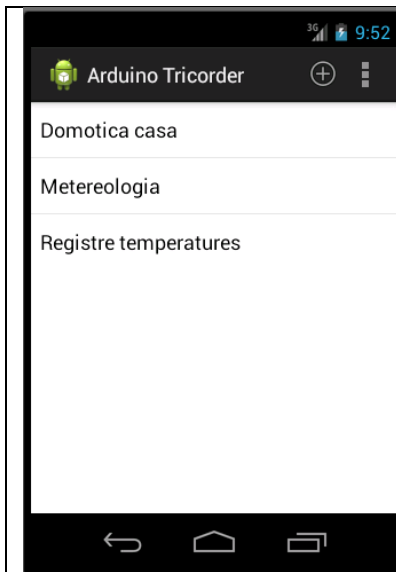
Per a cadascuna d'aquestes vies s'ofereix a l'usuari la possibilitat de dissenyar una interfície d'interacció a mida amb l'Arduino, ja sigui tàctil o verbal. A més a més, totes les interfícies són modificables. L'usuari tant sols ha de triar la descripció del control o comanda de la llista d'elements i modificar els valors desitjats, o bé eliminar el control o comanda.

A la següent figura podem veure el procediment descrit :



A continuació mostrem les pantalles que componen l'aplicació i una descripció de les accions que realitzen . Les captures han estat obtingudes amb l'emulador de l'entorn de programació Eclipse.

Pantalla de projectes

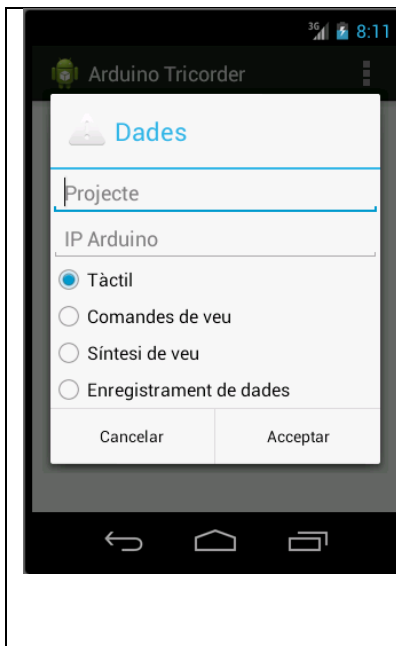


- Pantalla que mostra una llista amb els projectes creats. La icona del davant indica el tipus de projecte (tàctil, reconeixement de veu, síntesi de veu i enregistrament de dades)
- Pulsació curta : Es mostra el disseny del projecte
- Pulsació llarga : Apareix menú contextual per modificar les dades del projecte o be eliminar-lo



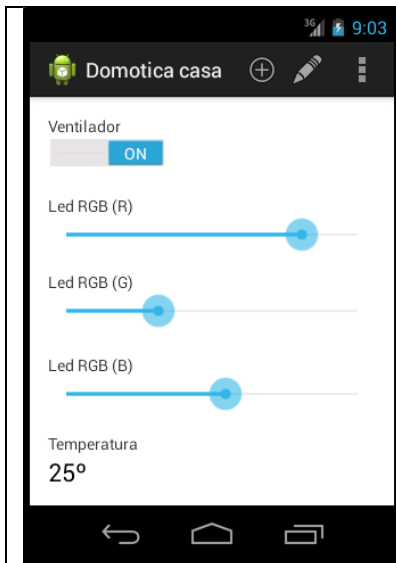
Obre finestra emergent per introduir les dades d'un nou projecte

Pantalla dades projecte



- Pantalla emergent que demana les dades del projecte.
- **Projecte** : nom que identifica el projecte
- **IP Arduino** : IP del microcontrolador Arduino que es vol controlar i supervisar (opcionalment es pot indica el número de port amb el format <IP>:>PORT>)
- Existeixen 4 tipus de projectes
 - **Tàctil** : enviament d'ordres mitjançant controls tàctils i visualització de dades en pantalla
 - **Comandes de veu** : enviament d'ordres mitjançant comandes de veu
 - **Síntesi de veu** : reproducció per audio i visualització en pantalla de dades dels sensors
 - **Enregistrament de dades** : enregistrament en un fitxer CSV de dades dels sensors

Pantalla interfície tàctil



- Inicialment la pantalla està buida. A mesura que s'afegeixen controls, aquests apareixen a la pantalla conformant els disseny de la interfície.
- Mostra els controls que l'usuari ha associat al projecte. Els controls es creen en temps d'execució a partir de la informació emmagatzemada a la base de dades. Aquests permeten interactuar amb l'Arduino enviant ordres i mostrar la informació dels sensors.

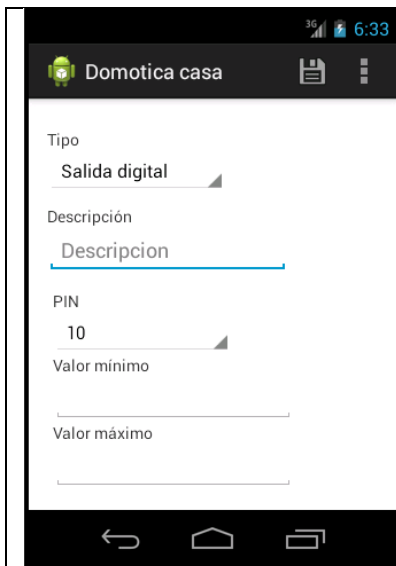


Pantalla per afegir un nou control a la interfície



Pantalla que mostra una llista amb la descripció dels controls per a la seva edició o eliminació.

Pantalla alta/modificació control mode tàctil



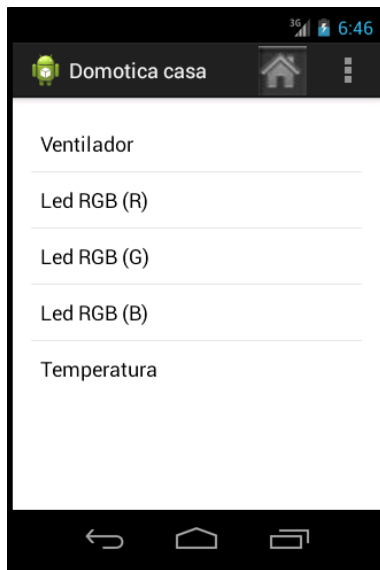
(*) Fent scroll apareix un desplegable amb el tipus de control (Switch, Numberpicker, etc.)

- **Tipo** : entrada(sensor) o sortida(actuador) digital o analògic
- **Descripción** : Nom per descriure el control
- **PIN** : en funció del tipus triat abans es mostra una llista dels pins disponibles per a aquest opció
- **Valor mínimo** : límit inferior per a un control analògic (habilitat si hem triat tipus analògic)
- **Valor máximo** : límit superior per a un control analògic (habilitat si hem triat tipus analògic)
- **Control** : en funció del tipus triat abans es mostren els possibles controls que permeten realitzar l'acció demandada. Així per exemple per activar o desactivar una sortida digital s'ofereix el control "Button" i "Switch", però en canvi per una sortida digital s'ofereix el control "NumbePicker" i "SeekBar".



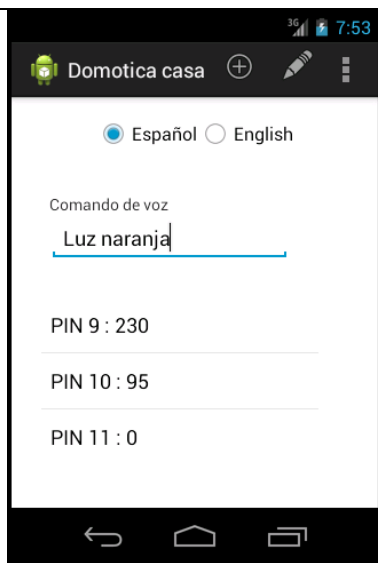
Guarda les dades del control a la base de dades associant-lo al projecte

Pantalla edició/eliminació control o comanda





- Llista amb la descripció dels controls o comandes
- Seleccionant amb una pulsació prolongada un element de la llista apareix un menú contextual que ofereix les opcions per editar o esborrar el control o comanda del projecte obert.

Pantalla disseny interfície comandes de veu




- Mostra una llista amb les comandes de veu associades al projecte. Per a cadascuna de les comandes indica quina operació realitza (pin i valor)
- **Idioma** : Indica l'idioma de la comanda.
- **Comando de voz** : Paraula que al ser reconeguda provocarà una acció



A mesura que es van afegint parelles pin/valor, apareixen en una llista a la part inferior de la pantalla.

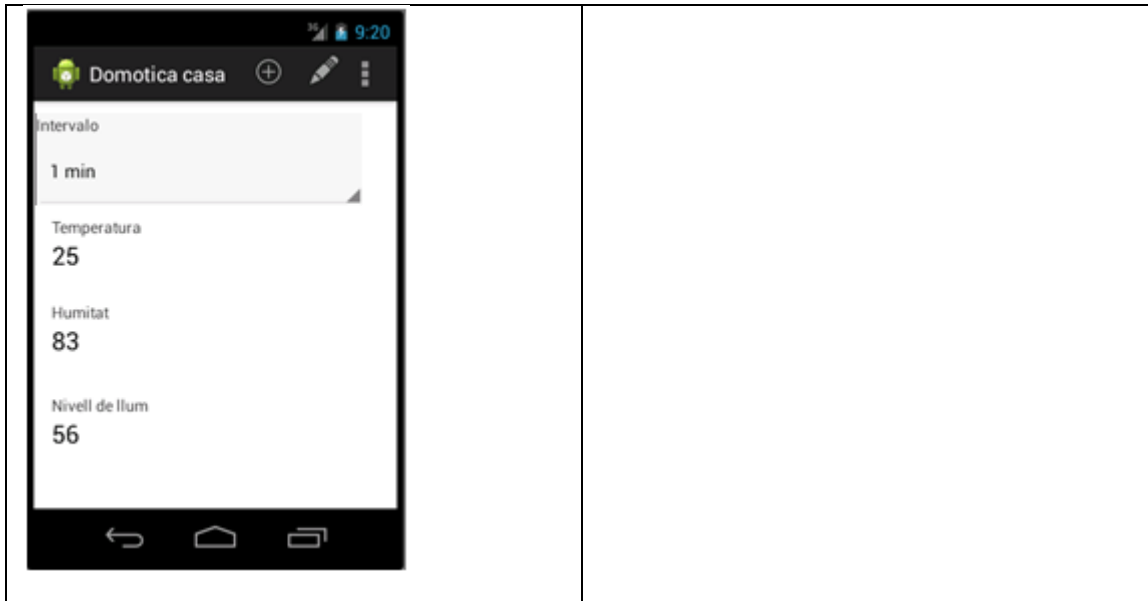
	<p> Afegir una nova parell pin/valor</p> <p> Pantalla que mostra una llista amb la descripció dels controls per a la seva edició o eliminació..</p>
--	---

Pantalla accions associades a comandes de veu

	<ul style="list-style-type: none">• Pin : pin de l'Arduino sobre el que es vol actuar• Valor : valor que prendrà el pin escollit. Pot ser un valor digital (HIGH/LOW) o analògic.
--	--

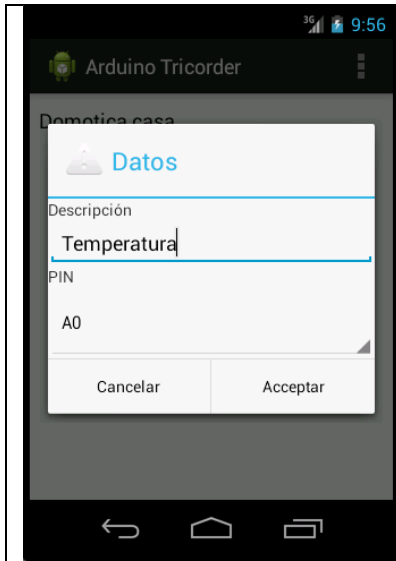
Pantalla disseny interfície monitorització amb síntesi de veu

	<p>Mostra els controls associats al projecte de tipus síntesi de veu obert.</p> <p> Afegir un control per visualitzar i reproduir verbal el valor d'un sensor.</p> <p> Pantalla que mostra una llista amb la descripció dels controls per a la seva edició o eliminació.</p>
--	--



Pantalla de selecció dels valors a monitoritzar

- **Descripción** : nom per descriure el sensor que es vol monitoritzar
- **Pin** : entrada (digital o analògica) que es vol monitoritzar
- **Intervalo**: interval en el que es reproduiran els valors monitoritzats. Expressat en segons.
- **Idioma** : idioma en que es reproduiran els missatges



- **Descripción** : Nom per descriure l'entrada que es vol monitoritzar
- **Pin** : pin de l'Arduino del que es vol obtenir dades

Pantalla disseny interfície enregistrament de dades



Mostra una llista amb les descripcions dels sensors que es vol enregistrar




Afegir un control per visualitzar i enregistrar el valor d'un sensor



Pantalla que mostra una llista amb la descripció dels controls per a la seva edició o eliminació.

Pantalla de selecció dels valors a enregistrar

	<ul style="list-style-type: none">• Descripción : Nom per descriure l'entrada que es vol monitoritzar• Pin : pin de l'Arduino del que es vol obtenir dades
---	---

11. Implementació

En aquest apartat es fa una descripció de les eines, editors i APIs utilitzades en el desenvolupament de l'aplicació mòbil Android i els sketches de suport Arduino.

11.1 Implementació de l'aplicació mòbil Android

Descripció de les fases i dels elements que han intervingut en el desenvolupament de l'aplicació Android.

11.1.1 Consideracions prèvies

Abans d'iniciar la implementació s'han plantejat sobre paper els diferents apartats de l'aplicació, el disseny d'aquests i el sistema de navegació més ràpid i intuïtiu.

Així doncs, s'ha decidit per crear una estructura jeràrquica dividida en projectes, on cada projecte pot ser d'un dels quatre tipus (tàctil, comandes veu, síntesi veu i adquisició de dades) establerts. El disseny triat per a les pantalles és net, amb icones auto-explicatives i fa un ús limitat del colors, en línia amb el que es buscava. S'ha apostat per utilitzar majoritàriament

LinearLayout com a plataforma per a la disposició dels elements gràfics per la seva senzillesa alhora de treballar amb contingut dinàmic.

Cal tenir en compte doncs, que l'aplicació té una part molt important de contingut dinàmic. És el que generen els usuaris al crear les seves interfícies personalitzades. Aquest sense dubte, és un dels punts forts de l'aplicació, però també el que més complexitat ha afegit. I més si és té en compte que no només es tractava de crear controls gràfics en temps d'execució, sinó que s'havia de reconèixer les comandes verbals i sintetitzar les frases proporcionades pels usuaris.

Donada la clara separació en 4 blocs (tàctil, comandes de veu, síntesi de veu i adquisició de dades) i la dualitat entre disseny i interfície, s'ha seguit aquesta estructura per anomenar les 8 Activitats que formen aquest grup.

Per tal de garantir una fàcil revisió del codi, aquest s'ha comentat exhaustivament.

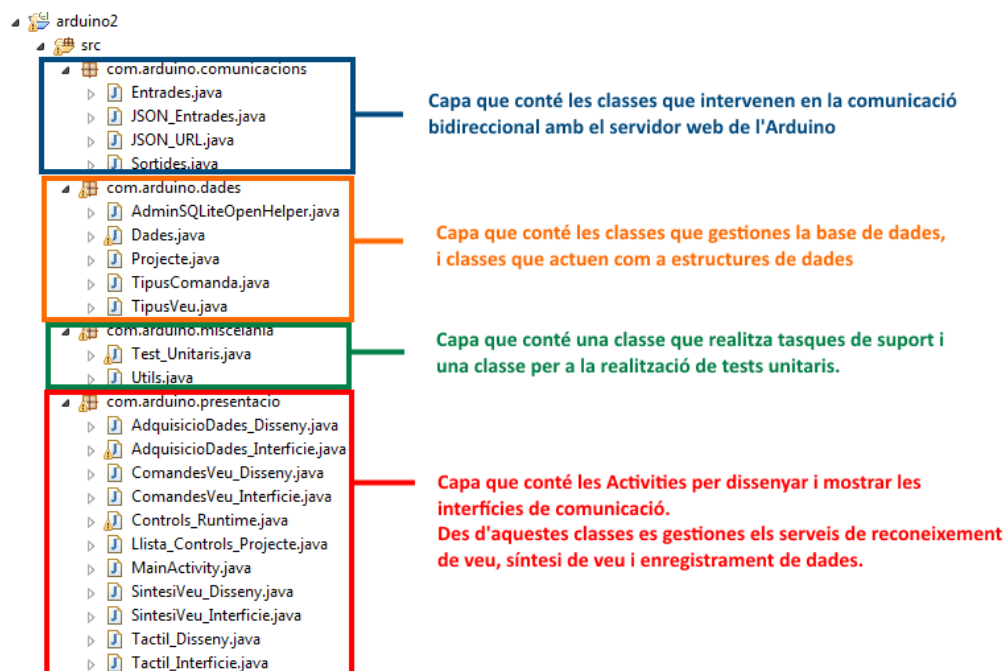
11.1.2 Eines utilitzades

L'eina principal per al desenvolupament de l'aplicació ha estat **Eclipse**. En concret s'ha fet servir l'ADT Bundle (Android Development Toolkit), que és una versió de Eclipse que porta instal·lat el plugin per al desenvolupament d'aplicacions Android.

El motiu de triar Eclipse és que en el moment actual és l'eina més utilitzada per a realitzar aplicacions Android, i el fet de que existeixi una versió amb el SDK d'Android incorporat simplifica la configuració de l'entorn de programació.

11.1.3 Organització de l'arbre de projecte / classes

Aquesta és l'organització del projecte de l'aplicació mòbil en l'entorn de desenvolupament Eclipse.



11.1.4 APIs utilitzades

Les API més destacables que han estat utilitzades són les següents :

- **SQLite API**
 - Utilitzada per emmagatzemar les dades dels diferents dissenys d'interfícies creades pels usuaris. Triada per la seva abundant documentació, facilitat d'ús emprant comandes SQL i eficàcia per a la tasca encomanda.
- **JSON API**
 - Utilitzada per a l'intercanvi d'informació entre l'aplicació Android i els microcontroladors Arduino amb capacitat Ethernet i/o Wifi. Triada per la seva capacitat d'intercanviar dades de forma simple.
- **TTS API (Text To Speech)**
 - Utilitzada per tal de millorar l'ergonomia de l'aplicació, oferint una interfície verbal. Permet reproduir verbalment els valors de les entrades de l'Arduino monitoritzades per l'aplicació.
- **Voice Recognition API**
 - Utilitzada per tal de millorar l'ergonomia de l'aplicació, oferint una interfície verbal. Permet l'enviament de comandes a les sortides de l'Arduino mitjançant comandes de veu.
- **File IO API**
 - Utilitzada per a l'enregistrament a la memòria del dispositiu mòbil de les dades adquirides per l'aplicació provinents de les entrades de l'Arduino.

11.1.5 Base de dades

Per tal d'emmagatzemar a la memòria dels dispositius mòbils les configuracions de les interfícies creades pels usuaris l'aplicació crea una base de dades SQLite a tal efecte. Aquesta base de dades conté 5 taules, una per a desar els projectes i les altres 4 corresponen als 4 tipus de projectes (tàctil, comandes de veu, síntesi de veu i adquisició de dades).

Les majoria d'operacions amb la base de dades s'han implementat dins de la classe **Dades.java**. Aquesta classe allotja les funcions per afegir, modificar i eliminar els projectes i controls de les interfícies, així com l'obtenció de valors concrets a partir d'una consulta SQL.

Per altra banda, s'ha fet necessari incloure funcions d'accés a la base de dades des de les Activities que mostren els controls de les interfícies, ja que a mesura que es recorre el cursor

obtingut a partir d'una consulta SQL, s'han de crear els controls en temps d'execució. Aquestes funcions han pres el nom de **Mostrar_Disseny_Interficie**.

11.1.6 Controls en temps d'execució

En funció de la modalitat de projecte triada per l'usuari i del tipus de senyals de la placa Arduino amb les que vol treballar (entrades i sortides digitals i analògiques) l'aplicació genera dinàmicament els controls necessaris per crear la interfície desitjada.

La classe **Controls_Runtime.java** conté un conjunt de funcions parametritzades per a la creació en temps d'execució dels controls que l'usuari ha definit prèviament. Aquests es col·loquen sobre el layout indicat.

Existeixen diferències entre els paràmetres d'aquestes funcions. Per exemple un control *SeekBar* necessita un valor mínim i un valor màxim, mentre que un control *Switch* no. El primer l'emprarem per a controlar una sortida analògica i el segon per controlar una sortida digital. Per altra banda totes les funcions comparteixen els següents paràmetres:

- Context context : context en el que es crea el control
- String PIN : indica amb quin pin de la placa Arduino està associat un control

A banda de crear i col·locar els controls, també s'han hagut d'implementar els esdeveniments associats a aquests controls (a excepció de *TextView*, *EditText* i *ImageView*). Així per exemple al canviar l'estat d'un *Switch* o al deixar de moure un *NumberPicker*, es capturen aquestes esdeveniments per tal de realitzar les peticions corresponents.

11.1.7 Reconeixement de veu

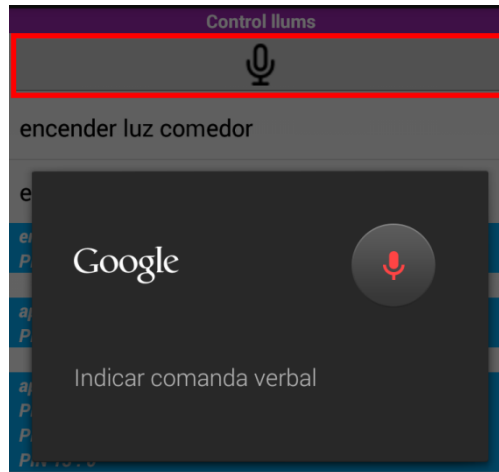
Sense dubte un dels reptes més importants a l'inici del projecte era incorporar un sistema de reconeixement de veu. Alhora era també un factor diferenciador respecte altres aplicacions similars i que aportava atractiu a l'aplicació. Sobretot per que fa que la interfície tingui un altíssim grau d'ergonomia, al límit màxim del que podem trobar amb la tecnologia actual.

La incorporació del reconeixement de veu a l'aplicació ha resultat més senzill del que pensava. Després de validar el funcionament en un projecte Eclipse a banda, la integració amb aquest projecte ha resultat fàcil. La implementació d'aquesta part es troba a la classe **ComandesVeu_Interficie.java**.

L'aplicació està preparada per reconèixer l'idioma que utilitza el dispositiu mòbil. És a dir, que si un usuari té el dispositiu mòbil configurat en Espanyol, l'aplicació reconeixerà les paraules que rebí en aquest idioma, però no les d'un altre idioma. Si un usuari té configurat un altre

idioma, no ha de modificar res per que l'aplicació funcioni. Només ha de tenir en compte fer servir paraules amb l'idioma del dispositiu, que com a norma, serà el més lògic.

Per tal d'activar el sistema de reconeixement de veu, s'ha habilitat un botó a la part superior de la pantalla.



Aquest botó crida al mètode **startVoiceRecognitionActivity** on indiquem una sèrie de paràmetres abans d'iniciar el reconeixement de veu. Un paràmetre important és **EXTRA_MAX_RESULTS**, que en aquest cas pren un valor igual a 2. Aquest paràmetre indica quantes possibles paraules o frases que ha reconegut el sistema ens ha de retornar.

```
private void startVoiceRecognitionActivity() {
    Intent intent = new Intent(RecognizerIntent.ACTION_RECOGNIZE_SPEECH);
    intent.putExtra(RecognizerIntent.EXTRA_CALLING_PACKAGE, getClass().getPackage().getName());
    intent.putExtra(RecognizerIntent.EXTRA_PROMPT, "Indicar comando verbal");
    intent.putExtra(RecognizerIntent.EXTRA_LANGUAGE_MODEL, RecognizerIntent.LANGUAGE_MODEL_FREE_FORM);
    intent.putExtra(RecognizerIntent.EXTRA_MAX_RESULTS, 2);
    startActivityForResult(intent, VOICE_RECOGNITION_REQUEST_CODE);
}
```

Un cop iniciat el sistema de reconeixement de veu, aquest espera l'arribada de comandes verbals.

Quan aquestes arriben, es crida al mètode **onActivityResult**.

```
protected void onActivityResult(int requestCode, int resultCode, Intent data) {

    if (requestCode == VOICE_RECOGNITION_REQUEST_CODE && resultCode == RESULT_OK) {
        ArrayList<String> matches = data.getStringArrayListExtra(RecognizerIntent.EXTRA_RESULTS);
        mList.setAdapter(new ArrayAdapter<String>(this, android.R.layout.simple_list_item_1, matches));

        for (int i = 0; i < comandes.size(); i++) {
            if (matches.contains(comandres.get(i).comanda) == true) {
```

Les frases o paraules reconegudes – limitades a 2 com hem vist – es visualitzen a la pantalla per informar a l'usuari de que és el que ha entès el sistema. Aquestes es comparen amb les comandes especificades prèviament en el disseny de la interfície des de la que s'ha iniciat el reconeixement de veu. En cas de trobar coincidència amb una comanda, es realitza petició corresponent a la placa Arduino.

Per tal de poder encadenar múltiples comandes verbals, al final del mètode `onActivityResult`, es torna a cridar a **`startVoiceRecognitionActivity`** per a iniciar automàticament el sistema de reconeixement de veu.

11.1.8 Síntesi de veu

En línia amb el que s'ha comentat sobre el reconeixement de veu, l'ús de síntesi de veu per a la reproducció de la informació obtinguda de la placa Arduino, afegia certa complexitat a la tasca, però li donava un important valor afegit.

De nou, s'ha optat per fer servir l'idioma per defecte del dispositiu mòbil per reproduir els textos, per ser l'opció més habitual. Per habilitar la funcionalitat de reproducció de textos, creem un objecte de la classe *TextoSpeech* al crear la Activity.

```
mTts=new TextToSpeech(getApplicationContext(),
    new TextToSpeech.OnInitListener() {
        public void onInit(int status) {}
    });
```

El funcionament bàsic és similar al del mode tàctil, on un procés en segon pla obté informació de la placa Arduino. El que s'implementa a la classe **`SíntesiVeu_Interficie.java`**, on es realitza la síntesi de veu, és un procés previ per tal de construir dinàmicament el text a reproduir. Aquest text està condicionat als valors dels sensors connectats a la placa Arduino i capturats per l'aplicació mòbil.

En el cas d'estar monitoritzant una entrada digital, on els valors possibles són 0 o 1, el text a reproduir serà l'indicat prèviament en el disseny de la interfície, és a dir, si el valor és 0 es reproduceix el text associat a aquest valor, i si és 1, doncs el text associat al valor 1.

En canvi, per a un entrada analògica, es concatenen al valor rebut un prefix i un sufix prèviament associats al pin de l'entrada en qüestió. D'aquesta manera s'aconsegueix donar realisme al text reproduït. Per exemple : **“La temperatura es de 20 grados”**. On el text en vermell és el prefix, el text en negreta el valor obtingut i el text verd és el sufix.

Les diferents frases és van concatenant en una sola cadena de text. Per establir una petita pausa entre frase i frase és col·loca la següent cadena **“...”**.


```
for (int k=0; k <= 5; k++) //entrades analògiques
  if (PIN.equals("A"+k)) {
    tt.setText(Projecte.entrades[k]);
    text_a_sintetitzar += textos.get(k).prefix+Projecte.entrades[k]+textos.get(k).sufix+"...";
  }

for (int k = 2; k <= 13; k++) //entrades digitals
  if (PIN.equals(String.valueOf(k))) {
    tt.setText(Projecte.entrades[k+4]);
    if (tt.getText().toString().equals("0"))
      text_a_sintetitzar += textos.get(k+4).text_LOW+"...";
    else
      text_a_sintetitzar += textos.get(k+4).text_HIGH+"...";
  }
```

Un cop obtinguda la cadena final corresponent a totes les entrades a monitoritzar, es sintetitza verbalment el text.

```
mTts.speak(text_a_sintetitzar, TextToSpeech.QUEUE_FLUSH, null);
```

La reproducció és realitza segons l'interval de temps indicat a la part superior de la pantalla, i que es pot modificar en qualsevol moment.

11.1.9 Enregistrament de dades

Un cop implementada la monitorització de la placa Arduino des del mode tàctil i des del mode síntesi de veu ha estat fàcil crear un nou mode que permetés enregistrar la informació rebuda a la memòria del dispositiu.

Com a requeriment important, cal assenyalar que l'escriptura la memòria del dispositiu mòbil requereix de permís, que cal indicar a l'arxiu Manifest.xml, com ja succeeix amb la connexió a internet. En aquest punt s'ha plantejat el dubte d'implementar o no aquesta funció, ja que molts usuaris descarten aplicacions desconegudes que demanen permisos per accedir al seus arxius.

Així doncs, la funcionalitat d'enregistrament de dades es troba implementada a la classe **AdquisicioDades_Interficie.java** . En aquesta classe hi trobem el mètode **Guardar_a_Fitxer** que es crida al mateix temps que es realitza la tasca en segon pla d'obtenció d'informació de les entrades de la placa Arduino. Aquest mètode afegeix cada vegada una nova fila amb els valors obtinguts separats pel delimitador ";" sobre un fitxer amb el nom del projecte obert i amb extensió CSV.

```
FileWriter fstream = new FileWriter(Environment.getExternalStorageDirectory().getPath()+"/"+FITXER+".csv", true);
BufferedWriter fbw = new BufferedWriter(fstream);
SimpleDateFormat sdf = new SimpleDateFormat("dd/MM/yyyy HH:mm:ss");
fbw.append(sdf.format(new Date()).toString()+";");
for (int i = 0; i < pins_escollits.size(); i++) {
  int index = Utils.posicio_PIN(pins_escollits.get(i));
  if (!Projecte.entrades[index].contentEquals("?")) {
    fbw.append(Projecte.entrades[index]+";");
    fbw.flush();
  }
}
fbw.newLine();
fbw.close();
```

11.1.10 Formularis

L'aplicació conté un gran nombre de formularis per tal de crear i modificar projectes i els elements que aquests contenen. Per tant, ha calgut implementar per a cadascun d'ells una funció per validar-los i evitar així l'introducció de valors erronis o repetits, repercutint posteriorment un mal funcionament d'altres funcions de l'aplicació.

Per tal d'agilitzar el procés d'entrada de dades en els formularis per crear els controls de les interfícies, s'ha decidit que després de guardar correctament les dades, aparegui un missatge informant de l'èxit de l'operació i que es netegin tots els camps, però que no redirigeixi a la vista de la interfície del projecte en curs. El motiu d'evitar la redirecció és que així s'evita que l'usuari hagi de tornar a entrar al formulari d'alta, ja que el més habitual serà que l'usuari doni d'alta tots els controls de cop.

Els formularis per gestionar les altes i les modificacions de cada apartat són iguals, l'únic que canvia és si es sol·liciten per afegir o actualitzar un registre a la base de dades. Aquest fet el determina un valor que es passa com a paràmetre al cridar la Activity corresponent.

```
if (intent.getString("DESCRIPCIO_CONTROL_INTENT") != null)
    descripcio_control = intent.getString("DESCRIPCIO_CONTROL_INTENT");

if (!descripcio_control.equals(""))
    Carregar_Valors_Modificacio();
```

Per als casos en que es sol·licita realitzar una modificació, s'han implementat funcions per tal de carregar els valors originals sobre el formulari.

11.1.11 Comunicació amb el servidor web de l'Arduino

Gràcies al servidor web que incorporen algunes plaques Arduino és possible la comunicació remota amb dispositius mòbils Android mitjançant wifi, 3G o 4G. En aquesta aplicació, podem dividir les comunicacions de la següent manera :

- Enviament de peticions : peticions al servidor web de l'Arduino per a que aquest modifiqui el valors de les sortides.
Aquestes peticions es poden realitzar des del mode tàctil o des del mode de reconeixement de veu.
 - Mode tàctil : quan es modifica l'estat d'un control digital (*Switch*) o analògic (*SeekBar*, *NumberPicker*), es realitza una crida REST al servidor web de l'Arduino. Els paràmetres d'aquesta crida determinen l'acció que haurà de portar a terme l'Arduino.

Per al cas d'una sortida digital, el control *Switch* actua de la següent manera quan canvia d'estat :

```
if (isChecked) {
    try {
        w.Peticio("http://" + Projecte.IP + "/arduino/" + tipus + "/" + PIN + "/1");
    } catch (Exception e) {
        Toast.makeText(buttonView.getContext(), "error", Toast.LENGTH_SHORT).show();
        e.printStackTrace();
    }
} else {
    try {
        w.Peticio("http://" + Projecte.IP + "/arduino/" + tipus + "/" + PIN + "/0");
    } catch (Exception e) {
        Toast.makeText(buttonView.getContext(), "error", Toast.LENGTH_SHORT).show();
    }
}
```

I per a una sortida analògica, al deixar anar un control *SeekBar* es realitza una petició d'aquesta forma :

```
public void onStopTrackingTouch(SeekBar arg0) {
    String valor = String.valueOf(seekBar.getProgress());
    w.Peticio("http://" + Projecte.IP + "/arduino/" + tipus + "/" + PIN + "/" + valor);
    Toast.makeText(context, valor, Toast.LENGTH_SHORT).show();
}
```

En el cas d'una interfície amb reconeixement de veu, la petició es realitza quan el sistema reconeix una comanda :

```
if (requestCode == VOICE_RECOGNITION_REQUEST_CODE && resultCode == RESULT_OK) {
    ArrayList<String> matches = data.getStringArrayListExtra(RecognizerIntent.EXTRA_RESULTS);
    mList.setAdapter(new ArrayAdapter<String>(this, android.R.layout.simple_list_item_1, matches));

    for (int i = 0; i < comandes.size(); i++) {
        if (matches.contains(comandes.get(i).comanda) == true) {
            Sortides w = new Sortides();
            w.Peticio("http://" + IP + "/arduino/digital/" + comandes.get(i).pin1 + "/" + comandes.get(i).valor1);
        }
    }
}
```

- Recollida d'informació :

La recollida d'informació sobre l'estat dels sensors connectats a l'Arduino, no es realitza de forma manual, sinó automàtica i en segon pla, a intervals de 3 segons en el cas de la modalitat tàctil, o bé definit per l'usuari a partir d'un desplegable amb diferents intervals de temps, per a les modalitats síntesi de veu i adquisició de dades.

Per a facilitar el tractament de les dades publicades pel servidor web de l'Arduino, aquestes es troben en format JSON.

Les entrades digitals es representen amb una imatge amb un cercle vermell quan l'entrada està inactiva, i un cercle verd quan està activa. Les entrades analògiques fan ús d'un *TextView* per mostrar els valors obtinguts.

Per distingir la captura d'informació, de les peticions que hem vist abans, el format de la crida REST és :

http://[IP_ARDUINO]/arduino/0/0/0

```
protected Void doInBackground(Void... params) {  
    try {  
        JSONObject jsonObject = JSON_URL.getJSONfromURL("http://" + Projecte.IP + "/arduino/0/0/0");  
  
        JSONArray jsonArray = jsonObject.getJSONArray("Arduino");  
        jsonObject = jsonArray.getJSONObject(0);  
  
        Projecte.entrades[0] = jsonObject.getString("A0");  
        Projecte.entrades[1] = jsonObject.getString("A1");  
        Projecte.entrades[2] = jsonObject.getString("A2");  
        Projecte.entrades[3] = jsonObject.getString("A3");  
        Projecte.entrades[4] = jsonObject.getString("A4");  
        Projecte.entrades[5] = jsonObject.getString("A5");  
        Projecte.entrades[6] = jsonObject.getString("2");  
        Projecte.entrades[7] = jsonObject.getString("3");  
        Projecte.entrades[8] = jsonObject.getString("4");  
        Projecte.entrades[9] = jsonObject.getString("5");  
        Projecte.entrades[10] = jsonObject.getString("6");  
        Projecte.entrades[11] = jsonObject.getString("7");  
        Projecte.entrades[12] = jsonObject.getString("8");  
        Projecte.entrades[13] = jsonObject.getString("9");  
        Projecte.entrades[14] = jsonObject.getString("10");  
        Projecte.entrades[15] = jsonObject.getString("11");  
        Projecte.entrades[17] = jsonObject.getString("13");  
    }  
}
```

11.1.12 Permissos

L'aplicació mòbil requereix de 2 permissos per al seu funcionament : connexió a internet i accés d'escriptura a la memòria del dispositiu. Aquests cal especificar-los de la següent manera al fitxer *AndroidManifest.xml* .

```
<uses-permission android:name="android.permission.INTERNET" />  
<uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE" />
```

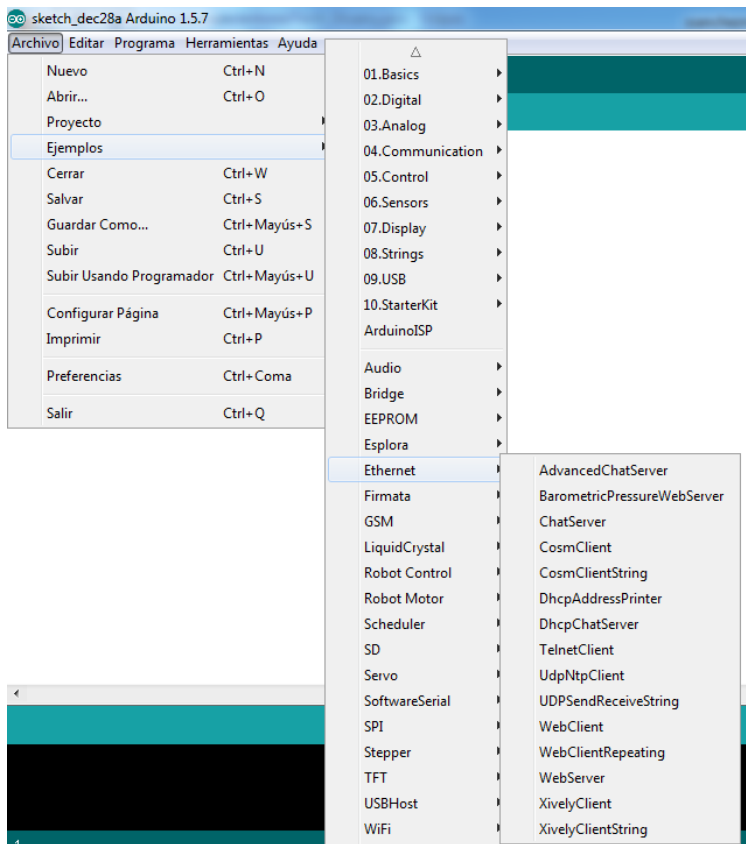
11.2 Implementació dels sketch Arduino

Descripció dels objectius i dels detalls d'implementació dels sketch Arduino per a l'interacció entre una placa Arduino i l'aplicació mòbil Android.

11.2.1 Consideracions prèvies

Partint de la base que el propi entorn de programació de l'Arduino proporciona gran quantitat d'exemples sobre comunicacions IP amb plaques Arduino, s'ha decidit crear un

sketch que complementi a aquests per tal d'assolir els requeriments necessaris per comunicar-se amb l'aplicació Android.



Els objectius principals abans de desenvolupar els sketch han estat que fos portable a tots les plaques que incorporen servidor web, que el codi a incrustar en els sketch dels usuaris fos mínim i finalment que fos fàcil d'utilitzar.

Aquest codi és publicarà en un repositori de GitHub per a que els usuaris el puguin descarregar i utilitzar amb l'aplicació mòbil.

11.2.2 Eines utilitzades

Per al desenvolupament del sketch Arduino s'ha fet servir l'entorn de programació oficial d'Arduino. A banda d'aquest entorn de programació no hi ha gaires més opcions per crear i transferir programes a una placa Arduino. I menys encara que suportin l'Arduino Yún, una de les plaques Arduino més noves i peça clau en aquest projecte. S'ha triat doncs, l'entorn oficial per coneixement de l'eina i fiabilitat.

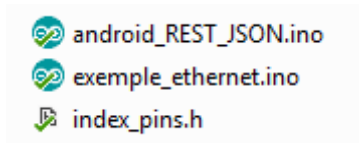
11.2.3 Organització dels sketch Arduino

Els programes o sketch per a les plaques Arduino, són fitxers amb extensió **.ino** . Opcionalment poden incorporar fitxers amb extensió **.h** que incloguin definicions de constants o formin part de llibreries externes. Un projecte Arduino pot incloure diversos fitxers. Un d'ells és el sketch principal i obligatòriament dóna nom al directori que inclou el citat sketch. Els altres sketch del projecte també han d'estar dins d'aquest directori.

Donades les diferències entre l'Arduino Yún i la resta de plaques amb comunicació IP, s'ha optat per crear 2 fitxers diferents. Aquests 2 fitxers són pràcticament iguals, però degut a la impossibilitat de sobrecarregar funcions, i per tal d'homogeneïtzar la crida de les funcions, s'ha optat per aquesta via.

Així doncs s'han implementat 2 sketch : **arduino_tricorder.ino** i **arduino_tricorder_Yun.ino** . A aquests cal sumar un fitxer anomenat **index_pins.h** que inclou definicions de constants.

D'aquesta manera el mínim nombre d'elements que cal carregar a la placa Arduino per realitzar la interacció amb l'aplicació mòbil Android seria el següent :



On **android_REST_JSON.ino** (placa diferent de l'Arduino Yún) i **index_pins.h** són els fitxers que contenen les funcions d'adaptació necessàries i **exemple_ethernet.ino** és el sketch principal, on cal cridar a les citades funcions.

11.2.4 Lectura de les entrades i conversió a JSON

Per a que l'aplicació mòbil pugui rebre informació de l'entorn on està ubicat l'Arduino mitjançant els sensors que té connectats, s'ha implementat un funció que llegeix l'estat de les entrades digitals (**digitalRead**) i analògiques (**analogRead**), transformant aquesta informació en format JSON.

Però en moltes ocasions, els valors recollits són tractats per tal de donar a l'usuari la informació que realment necessita. Per exemple, quan es llegeix l'entrada analògica on està connectat un sensor de temperatura, el valor obtingut està en mV i no en graus. En aquest cas, cal fer una conversió de mV a graus.

```
temperatura = analogRead(pin_sensor_temperatura);  
temperatura = 5.0*temperatura*100.0/1024.0;  
Serial.print(temperatura);  
Serial.println(" oC");
```

Aquest fet complica la recollida d'informació, ja que cal distingir quan es vol la lectura directa de l'entrada o quan es vol la calculada. Aquesta última es guarda en una variable.

Per solucionar això, es defineix dins el programa principal un array de números en punt flotant - ja que les lectures dels sensors són o enteres o decimals - on s'indica si la lectura d'una entrada cal obtenir-la directament de l'entrada física o d'una variable.

```
float entrades[NUMERO_DE_PINS] = (BUIT, BUIT, BUIT, BUIT, BUIT, BUIT, BUIT, BUIT, BUIT, BUIT, BUIT, BUIT, BUIT, BUIT, BUIT, BUIT, BUIT);
entrades[PIN_A2] = temperatura;
```

En aquest cas, estaríem indicant que els valors de les entrades serien els valors obtinguts mitjançant lectures directes (valor BUIT) de les entrades, menys en el cas del pin A2, que caldria agafar-lo de la variable *temperatura*.

Així doncs, després de fer el recorregut per tots els pins i tenint en compte lo comentat anteriorment, es publica la informació en format JSON al servidor web de la placa Arduino.

```
client.println("HTTP/1.1 200 OK");
client.println("Content-Type: text/html");
client.println();
client.println(" ");

client.println("{ 'Arduino':[");
client.println("");

for (int i=0; i <= 5; i++)
  if (entrades[i] == BUIT)
    client.println("\"A"+(String)i+"\":\")+\""+(String)analogRead(i)+"\",");
  else
    client.println("\"A"+(String)i+"\":\")+\""+(String)entrades[i]+"\",");

for (int i=2; i <= 12; i++)
  if (entrades[i+4] == BUIT) //+4 per pin 2 està a la posició 6 = 2 + 4
    client.println("\""+(String)i+"\":\")+\""+(String)digitalRead(i)+"\",");
  else
    client.println("\""+(String)i+"\":\")+\""+(String)entrades[i+4]+"\",");

client.println("\"13\":\")+\""+(String)digitalRead(13)+"\"");
client.println("]");
client.println("]");
```

Tot això dóna com a resultat :

```
{ 'Arduino': [
{
  "A0": "272",
  "A1": "253",
  "A2": "268",
  "A3": "310",
  "A4": "296",
  "A5": "266",
  "2": "99.00",
  "3": "0",
  "4": "0",
  "5": "0",
  "6": "0",
  "7": "0",
  "8": "0",
  "9": "0",
  "10": "0",
  "11": "0",
  "12": "0",
  "13": "0"
}
]}
```

11.2.5 Tractament de les peticions i activació de les sortides

Per a controlar les sortides de la placa Arduino de forma remota, es realitzen peticions HTTP al servidor web de l'Arduino. La següent funció analitza aquestes peticions, i en funció dels paràmetres recollits executa accions sobre les sortides digitals (`digitalWrite`) i analògiques (`analogWrite`) de la placa Arduino.

Un exemple de crida per activar la sortida digital del pin 2 seria :

`http://192.168.1.40/arduino/digital/2/1`

En primer lloc, es guarden els paràmetres rebuts en una variable.

```
String parametres = "";

while (client.connected()) {
  if (client.available()) {
    char c = client.read();
    if (c == '\n')
      break;
    else
      parametres += c;
  }
}
```

Posteriorment, s'analitza si la petició fa referència a una sortida digital o analògica.

```
if (parametres.indexOf("digital") >= 0)
```

Un dels paràmetres és el pin sobre el que s'ha de realitzar l'acció sol·licitada. Aquest s'ha de definir com a sortida.

```
pinMode(pin, OUTPUT);
```

El darrer paràmetre de la petició és el valor que cal aplicar sobre la sortida. La forma d'aplicar-lo és diferent segons si es tracta d'una sortida digital o analògica.

Per a una sortida digital :

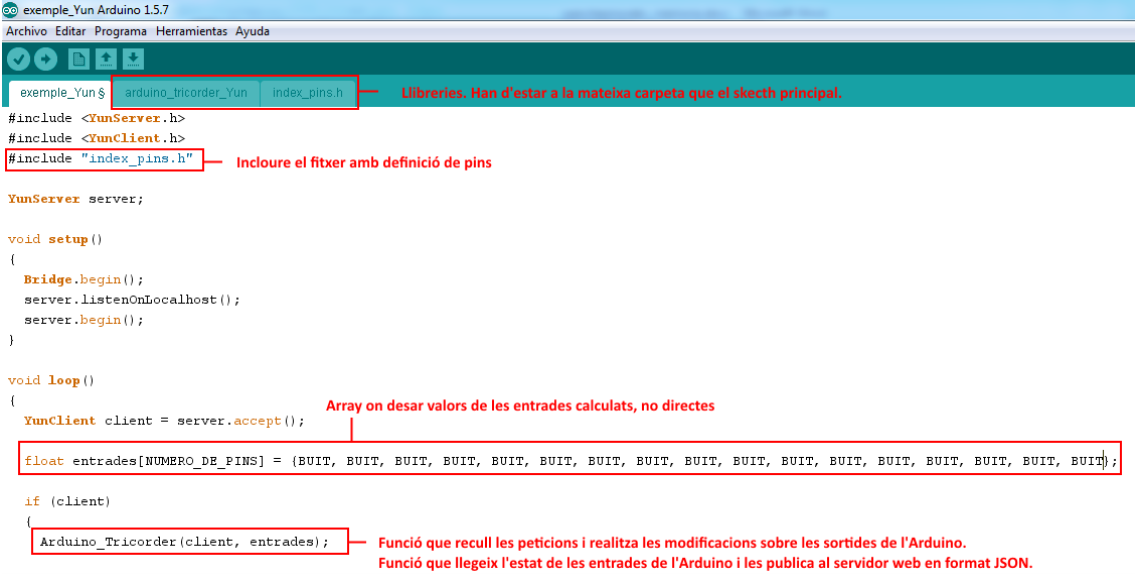
```
if (valor == 1)
  digitalWrite(pin, HIGH);
else
  digitalWrite(pin, LOW);
```

Per a una sortida analògica :

```
analogWrite(motor, 200);
```


11.2.6 Integració dins sketch principal

Amb la idea de simplificar al màxim les modificacions a realitzar en els sketches Arduino dels usuaris, s'ha aconseguit que amb només 3 línies al sketch principal, i 2 fitxers de suport, es pugui fer que un projecte Arduino prèviament creat, interaccioni amb l'aplicació mòbil Android desenvolupada. Sempre que la placa disposi de comunicació IP, és clar.



```
example_Yun$ arduino_tricorder_yun | index_pins.h
#include <YunServer.h>
#include <YunClient.h>
#include "index_pins.h"

YunServer server;

void setup()
{
  Bridge.begin();
  server.listenOnLocalhost();
  server.begin();
}

void loop()
{
  YunClient client = server.accept();
  float entrades[NUMERO_DE_PINS] = {BUILT_IN_PINS};
  if (client)
  {
    Arduino_Tricorder(client, entrades);
  }
}
```

12. Funcionament de l'aplicació

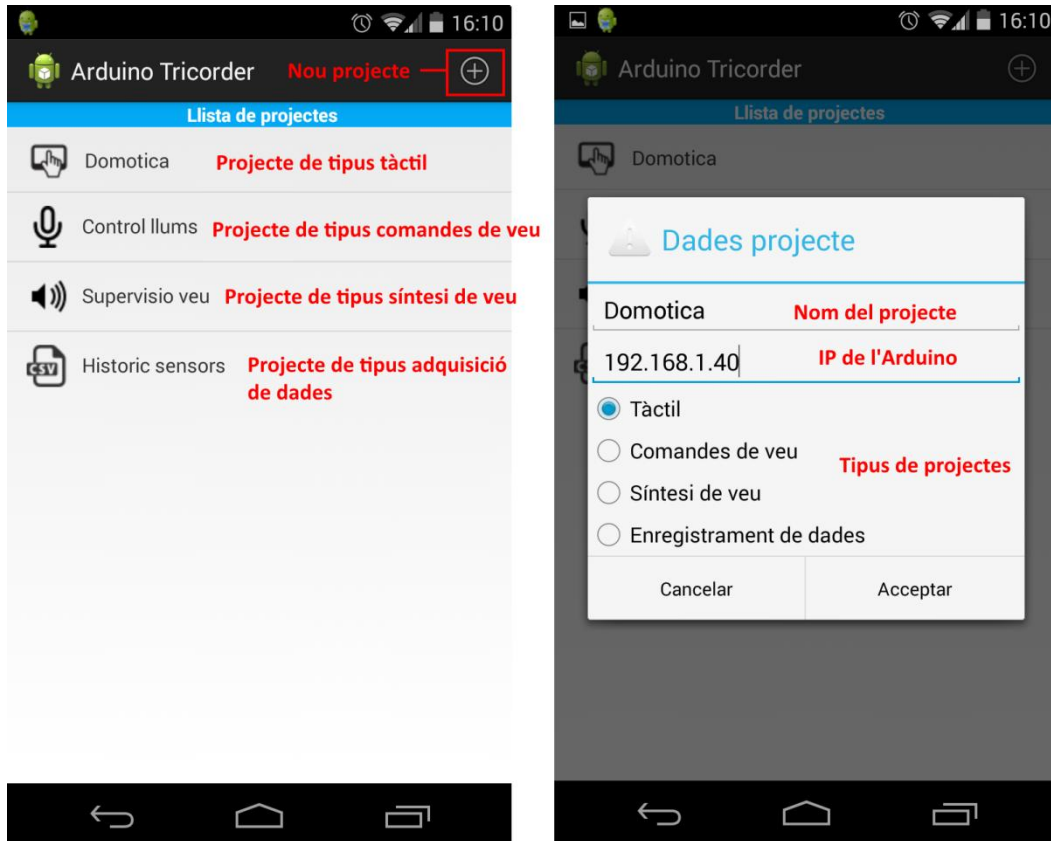
En aquest apartat es mostren una sèrie de captures del funcionament de l'aplicació en un dispositiu mòbil Android. Aquestes mostren el procés de creació d'un projecte en les 4 modalitats disponibles, l'aspecte que ofereixen les interfícies dissenyades pels usuaris, així com la possibilitat d'editar posteriorment tots els elements.

12.1 Creació i llistat de projectes

A l'obrir l'aplicació, apareix el llistat de projectes creats per l'usuari. A mesura que es van creant projectes, aquests apareixen en aquest llistat agrupats per la modalitat del projecte. Una icona representativa de cada modalitat precedeix el nom del projecte. Si es fa una pulsació curta sobre el nom del projecte s'accedeix a la interfície prèviament creada. En

canvi fent una pulsació llarga apareix un menú contextual que permet modificar i eliminar els projectes creats.

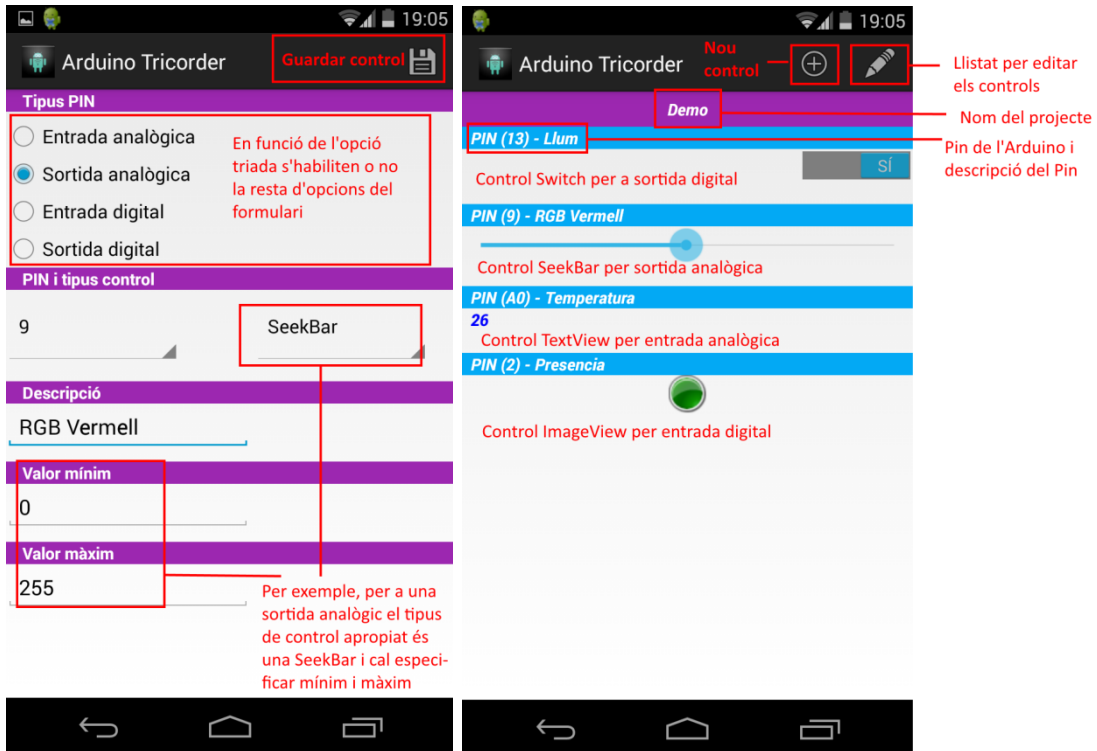
A la part superior dreta de la pantalla hi ha la icona que obre el diàleg per crear un nou projecte. On cal indicar un nom per identificar el projecte, la IP de la placa Arduino amb la que es vol interactuar i el tipus de projecte.



12.2 Modalitat tàtil

La modalitat tàtil és l'única de les 4 disponibles que permet supervisar i controlar una placa Arduino des d'una sola interfície. A la part superior de la pantalla hi ha les icones corresponents a les accions d'afegir un nou control a la interfície i accedir al llistat per editar els controls prèviament afegits.

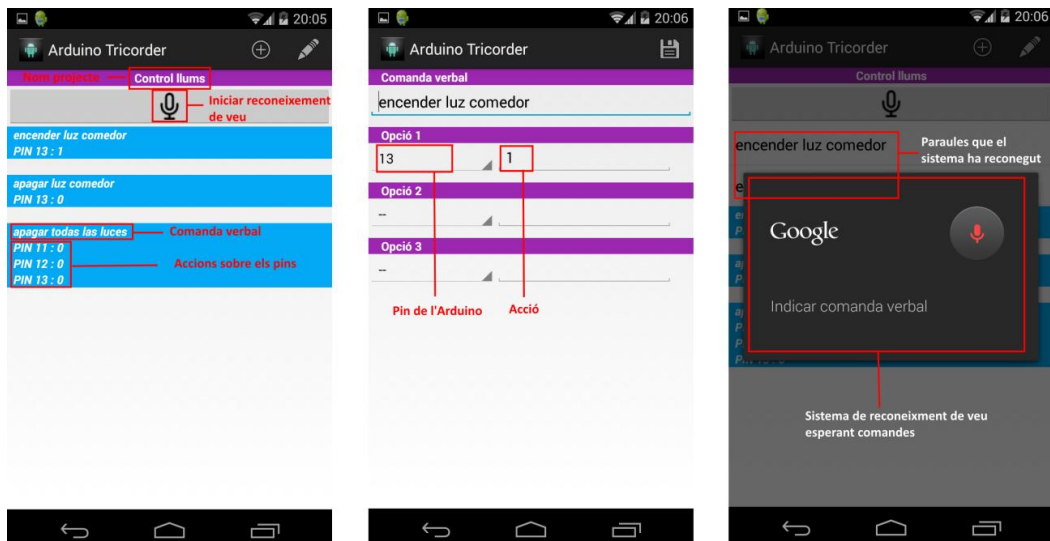
En funció de si es vol supervisar o controlar, i de si la senyal és digital o analògica, es crea sobre la interfície el control corresponent. Per exemple si volem controlar una sortida digital es crearà un control de tipus Switch (on/off). En canvi si la sortida és analògica, es podrà triar entre una SeekBar o un NumerPicker.



12.3 Modalitat comandes de veu

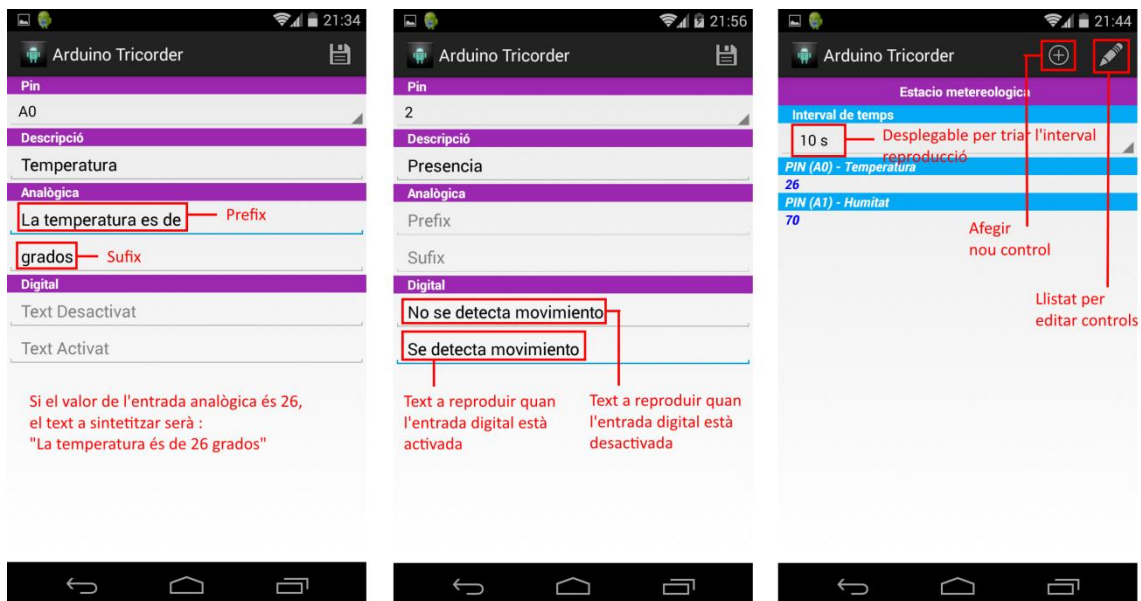
Amb la modalitat de comandes de veu l'usuari pot controlar les sortides d'una placa Arduino mitjançant comandes verbals. Aquestes poden estar formades per una o més paraules. A més l'usuari pot modificar fins a 3 sortides amb una sola comanda.

Per activar el reconeixement de veu cal clicar el botó amb el símbol del micròfon. Les paraules reconegudes apareixen en una llista a sota d'aquest botó, permeten a l'usuari conèixer que ha entès el sistema. Si les paraules reconegudes coincideixen amb una comanda, l'aplicació envia una petició a l'Arduino per a que realitzi la tasca corresponent.



12.4 Modalitat síntesi de veu

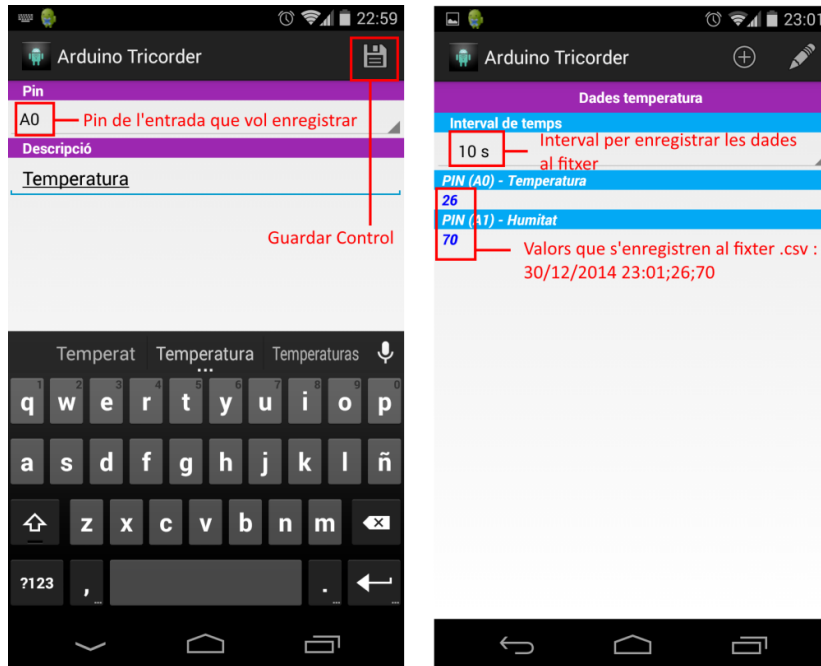
Aquesta modalitat permet a banda de supervisar les dades capturades de la placa Arduino mostrant-les a la pantalla, reproduir-les amb una veu sintetitzada. Per a fer més natural la reproducció d'aquestes dades, cal introduir des del formulari una frase substitutiva en el cas d'entrades digitals, o bé unes paraules davant i darrera del valor obtingut en el cas de les entrades analògiques. Mitjançant un desplegable es pot variar l'interval de reproducció dels missatges.



12.5 Modalitat adquisició de dades

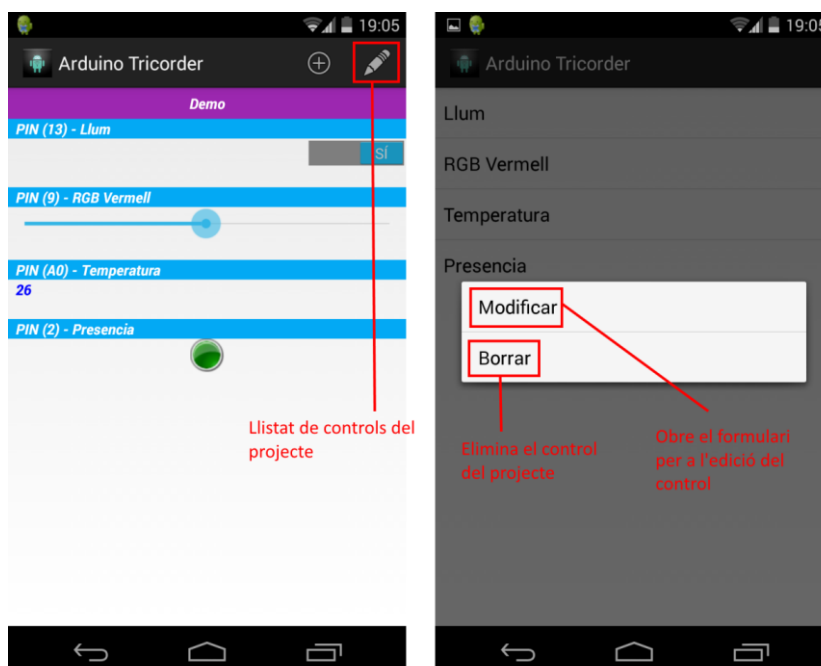
La darrera modalitat d'interfície és la d'adquisició de dades, que es pràcticament igual que l'anterior, permetent la supervisió de les entrades de l'Arduino segons l'interval indicat, però en aquest cas la informació capturada s'enregistra a la memòria del dispositiu mòbil.

Quan s'obre un projecte d'aquesta modalitat, es crea un fitxer amb el nom del projecte – substituint els espais en blanc per el símbol “_” – amb l'extensió CSV. A cada interval de lectura, s'afegeix una línia al fitxer amb la data i hora i els valors capturats, separats amb el delimitador “;”.



12.6 Llistat per editar els controls d'un projecte

Per canviar dades dels controls que s'ha afegit prèviament o per eliminar-los, disposem a cadascuna de les 4 interfícies d'una opció per accedir a un llistat on apareixen les descripcions dels controls afegits. Fent una pulsació llarga sobre una d'aquestes descripcions, apareix un menú contextual que permet eliminar el control, o bé modificar-lo. En aquest cas s'obre el formulari corresponent a cada control amb les dades originals, per a que l'usuari les modifiqui.



13. Conclusions

Amb la realització d'aquest projecte he aconseguit portar a la pràctica el meu desig d'unir dues de les meves passions, la programació i l'electrònica, i més concretament la programació de dispositius mòbils i la plataforma Arduino.

Anteriorment, havia realitzat petits projectes personals d'interacció entre la placa Arduino i dispositius mòbils, però des de plataformes molt més senzilles com PhoneGap, i on les accions estaven definides prèviament.

El repte d'aquesta aplicació era desenvolupar una eina oberta, capaç de permetre als usuaris de la comunitat Arduino incorporar els dispositius mòbils Android en els seus projectes. A més, es volia oferir una aplicació atractiva, amb l'ús de reconeixement i síntesi de veu.

A continuació, passo a concretar les conclusions extretes d'aquest projecte.

13.1 Assoliment d'objectius

El projecte ha donat com a resultat una aplicació que assoleix tots els objectius proposats inicialment. Aquesta aconsegueix, amb el suport dels sketch Arduino desenvolupats, oferir la possibilitat de supervisar, controlar i adquirir dades de qualsevol projecte desenvolupat amb una placa Arduino amb comunicació IP.

A més, s'ha aconseguit quelcom més que un comandament a distància. Amb l'inclusió de reconeixement i síntesi de veu s'ha dotat a l'aplicació de gran ergonomia. Aquest fet sense dubte aporta un important valor afegit respecte a les solucions actuals.

També vull destacar el treball realitzat en la programació dels sketches Arduino, ja que amb molt poc codi s'ha aconseguit implementar la interacció amb l'aplicació mòbil. A més la integració amb els sketch dels usuaris resulta molt senzilla i els exemples facilitats per a les diferents plataformes (Yún i shields) estan llestos per funcionar i són fàcils d'interpretar per a la majoria d'usuaris d'Arduino.

13.2 Desviacions del projecte original

A la conclusió del projecte es pot afirmar que aquest no ha patit desviacions respecte la idea original. En un primer moment, es van plantejar l'assoliment de més objectius, però que ràpidament és van descartar per la limitació temporal, i per tal de garantir la consecució

d'uns objectius més raonables. Al final de projecte s'ha demostrat que aquesta decisió va ser encertada.

13.3 Valoració personal

La meva valoració personal sobre la realització d'aquest projecte és molt bona. Com ja he esmentat abans, en aquest projecte he pogut combinar la programació amb una petita part d'electrònica que són de gran interès per a mi, no només en l'àmbit professional sinó també en el personal. Segurament, en un altre moment hagués decidit desenvolupar aquesta aplicació, però crec que amb menys rigor del que he aplicat al fer el TFG.

Sobretot valoro el fet que en aquest projecte he pogut posar en pràctica un ventall ampli de tecnologies : creació de controls en temps d'execució, tasques en segon pla, comunicacions, bases de dades, escriptura a memòria del dispositiu, síntesi de veu i reconeixement de veu, entre d'altres. Això m'ha permès aclarir molt dubtes que tenia abans de fer projecte sobre quina dificultat tenia treballar amb aquestes tecnologies, i que sincerament crec que après a dominar.

També estic satisfet per haver desenvolupat una aplicació que es diferencia de la majoria d'aplicacions mòbils actuals, al permetre la interacció entre el món virtual i el real. Actualment no hi ha pràcticament aplicacions en aquest àmbit, però que segur que en els propers anys la situació canviarà amb l'aparició de més dispositius connectats com proposa el " l'internet de les coses".

Ara doncs, és el moment de publicar l'aplicació mòbil Android i els sketch Arduino i fer la meva contribució a la comunitat Arduino. Fer una aportació al moviment Maker del que tant he après i amb el qual m'identifico, era una qüestió que tenia pendent de feia temps. Per aquesta raó, aquest projecte no acaba aquí. Tot just ha començat. El següent pas serà consolidar l'aplicació amb les aportacions dels usuaris i implementar noves funcionalitats que permetin millorar-la.

Finalment, valorar l'experiència amb Eclipse i el SDK d'Android com a millorable. La veritat és que he patit alguns contratemps degut a alguns bugs de l'Eclipse. La velocitat i estabilitat de l'emulador també necessita millores. Tot i que he de reconèixer que al final m'he trobat còmode, no descarto provar altres eines com Netbeans i Android Studio.

13.4 Futures millores

Tant abans de començar el projecte, com durant i després del mateix han sorgit un gran nombre de possibles millores. Crec que totes elles són perfectament assumibles, i que

aportaran encara més versatilitat a l'aplicació. També espero rebre *feedback* dels usuaris i afegir millores en les que jo no hi he pensat.

A continuació descriure les que considero més rellevants :

- 1) Afegir suport per a Raspberry Pi : Inicialment havia plantejat fer compatible l'aplicació mòbil amb el famós micro-ordinador de baix cost Raspberry Pi. Degut a les limitacions de temps, i a la diferència en la distribució de pins GPIO dels diferents models existents, ho vaig descartar. Però sense dubte, aplicar el mateix plantejament sobre el Raspberry Pi, ampliaria de forma notable els usuaris potencials de l'aplicació. Això si, caldria rebatejar l'aplicació.
- 2) Peticions i respostes verbals : El projecte actual tot i incorporar reconeixement i síntesi de veu, ho fa de forma separada. Es podria implementar la funcionalitat que permetés demanar a l'aplicació mòbil informació sobre els valors capturats pels sensors de l'Arduino. Per exemple, l'usuari podria demanar “¿ Cuál es la temperatura actual ?”, i l'aplicació podria respondre : “La temperatura es de **26.3** grados”. Caldria deshabilitar el sistema actual que inicia el reconeixement després de rebre una comanda, ja que sinó analitzaria el text reproduït per l'aplicació pensant que és una nova comanda.
- 3) Sistema de notificaciones : Es podria implementar un sistema per rebre notificaciones a partir de la informació capturada per l'Arduino. Per exemple, quan es detecti moviment en un lloc entre unes dates i hores concretes, quan la temperatura sobrepassa un llindar, etc.
- 4) Sistema d'accions condicionades : Aquesta funcionalitat seguiria la filosofia del anomenat IFTTT (If this then that) però amb interacció amb el món real. Per exemple es podria configurar l'aplicació per a que actues en funció de certs esdeveniments :
 - a. Quan el dispositiu mòbil arribi a unes certes coordenades gps (per exemple a prop de casa), activi una sortida de la placa Arduino per obrir porta o llums, calefacció, etc.
 - b. Reproduir un so i vibrar si un sensor de moviment connectat a l'Arduino detecta presència en un horari concret.
 - c. Enviar un correu electrònic o un tweet en funció de valors registrats pels sensors de l'Arduino. Aquesta tasca ja es pot fer des del mateix Arduino, però en aquest cas es tractaria de simplificar el procés a usuaris menys experts en Arduino.
 - d. Definir dia i hora en que s'ha d'executar una acció.

- 5) Aplicació en diferents idiomes : Un cop consolidada la primera versió de l'aplicació, una de les primeres millores és incorporar més idiomes, o com a mínim, l'anglès.
- 6) Incorporar sistema d'autenticació i xifratge : Seria interessant incorporar un sistema d'autenticació, per assegurar que qui controla i supervisa l'Arduino està autoritzat. A més, caldria xifrar l'intercanvi d'informació.

14. Fonts d'informació

La realització del projecte s'ha recolzat en la gran quantitat d'informació disponible a la xarxa. Afortunadament, tant la plataforma Arduino com la plataforma Android compten amb abundant documentació oficial i de particulars.

Les següents fonts han estat consultades entre el 16/09/2014 i el 09/01/2015.

14.1 Fonts sobre Android

En aquestes pàgines he trobat informació i exemples per al desenvolupament de l'aplicació mòbil.

- <http://www.elandroidelibre.com/2011/09/20-horas-de-youtube-para-aprender-a-programar-android.html>
- <http://www.vogella.com/tutorials/Android/article.html>
- <https://www.udemy.com/blog/tutorial-de-android-sqlite-para-principiantes/>
- http://www.sgoliver.net/blog/?page_id=2935
- <http://www.survivingwithandroid.com/>
- <http://developer.android.com/>
- <http://myappandroid.wordpress.com/>
- <http://stackoverflow.com/>
- <http://www.mysamplecode.com/>
- <http://www.androidbegin.com/>
- <http://www.androidhive.info/>
- <http://www.tutorialspoint.com/android/>

14.2 Fonts sobre Arduino

Per al desenvolupament dels sketches Arduino i per a generar la documentació he obtingut informació dels següents enllaços :

- <http://www.arduino.cc/>
- <http://www.instructables.com/>
- <http://blog.bricogeek.com/>