

APLICACIÓ ANDROID PER AL CONTROL I RECEPCIÓ DE DADES D'UNA BOIA KDUINO

SERGI FRUTOS GRACIA

GRAU DE TECNOLOGIES DE LA TELECOMUNICACIÓ

CONSULTOR:

MANEL LLOPART VIDAL

17 de Juny del 2015



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

“Si una persona és perseverant, encara que sigui dura d’enteniment, es farà intel·ligent; i encara que sigui dèbil es transformarà en fort.”

Leonardo Da Vinci

Als meus pares per recolzar-me en tot.

A Aurora per la seva infinita paciència.

A Jaume Píera i Raúl Bardají per deixar-me participar en el projecte.

Al Doctor Joan Ramon Torrellà per animar-me sempre a continuar endavant.

ABSTRACT

Català

Aquest treball consisteix en el disseny i implementació d'una aplicació en llenguatge Java per a un sistema operatiu Android, per al control i la recepció de dades d'una boia científica que mesura la transparència de l'aigua. La boia disposa d'una sèrie de sensors que capten la llum solar atenuada per l'aigua a diferents profunditats. En funció d'aquestes mesures es pot establir analíticament un coeficient d'atenuació que serveix per determinar la qualitat de l'aigua.

El sistema disposa d'un sistema de comunicació per Bluetooth muntat sobre una plataforma Arduino al que es pot connectar el dispositiu Android. L'aplicació reenviarà les dades rebudes per part de la boia a una base de dades per al seu posterior anàlisi de la comunitat científica, per tal d'establir mapes de la qualitat de l'aigua.

El treball forma part d'un dels objectius del Projecte Europeu Citclops i s'ha desenvolupat amb col·laboració amb el Centre Superior D'investigacions Científiques i l'Institut de Ciències del Mar, que és el creador i propulsor del sistema. Aquest treball és un projecte de ciència ciutadana que pretén dissenyar tecnologies de baix cost i metodologia senzilla per a que grups de voluntaris adquireixin dades del mar i les comparteixin amb la comunitat científica.

English

This document consists of a design and an implementation of an application in Java language for an Android operational system, for the control and the reception of data from a scientific buoy that takes measurements to evaluate the transparency of the water. To achieve this goal, the buoy contains sensors that are used to capture the attenuation solar light on the water at different depths. With this measure, it is possible to establish an attenuation coefficient analytically, offering the scientist the possibility of establishing the quality of the water.

The system has a Bluetooth communication system working under an Arduino platform where the Android device can connect with. The application is going to send the data received from the buoy to a data base so that, the scientist community can evaluate it and create maps of the water quality.

This work is part of the objectives of an European Project called Citclops and has been developed in collaboration with the “Centre Superior d’Investigacions Científiques” and “l’Institut de Ciències del Mar” that is the marker and the propeller of the system. This work is a citizen science project that wants to develop low cost and easy technologies for a volunteer groups that help scientists capture measurements from seas and oceans.

Castellano

Este trabajo consiste en el diseño e implementación de una aplicación en lenguaje Java para un sistema operativo Android para el control y la recepción de datos de una boya científica que mide la transparencia del agua. La boya dispone de una serie de sensores que captan la luz solar atenuada por el agua a diferentes profundidades. En función de estas medidas se puede establecer analíticamente un coeficiente de atenuación que sirve para determinar la calidad del agua.

El sistema dispone de un sistema de comunicación por Bluetooth montado sobre una plataforma Arruino al que se puede conectar el dispositivo Android. La aplicación reenviará los datos recibidos por parte la boya a una Base de Datos para un posterior análisis de la comunidad científica, con la finalidad de crear mapas de la calidad del agua.

El trabajo forma parte de uno de los objetivos del Proyecto Europeo Citclops, y se ha desarrollado en colaboración con el Centro Superior de Investigaciones Científicas y el Instituto de Ciencias del Mar, que es el creador y propulsor del sistema. Este trabajo es un proyecto de ciencia ciudadana que pretende diseñar tecnologías de bajo coste y metodologías sencillas para que grupos de voluntarios adquieran datos del mar y los compartan con la comunidad científica para su análisis.

RESUM

El treball consisteix en el disseny i implementació d'una aplicació elaborada amb llenguatge JAVA per a sistemes operatius Android, que permeti el control d'una placa Arduino que a la vegada controla una sèrie de sensors que mesuren la transparència de l'aigua.

El sistema ha de permetre la transmissió i recepció de dades des d'un dispositiu Android a un sistema Arduino que es troba disposat en una boia científica, la qual mitjançant una sèrie de sensors ha de permetre la captació de dades per al posterior anàlisis per part de la comunitat científica. A la Fig.1 es mostra el funcionament del sistema.

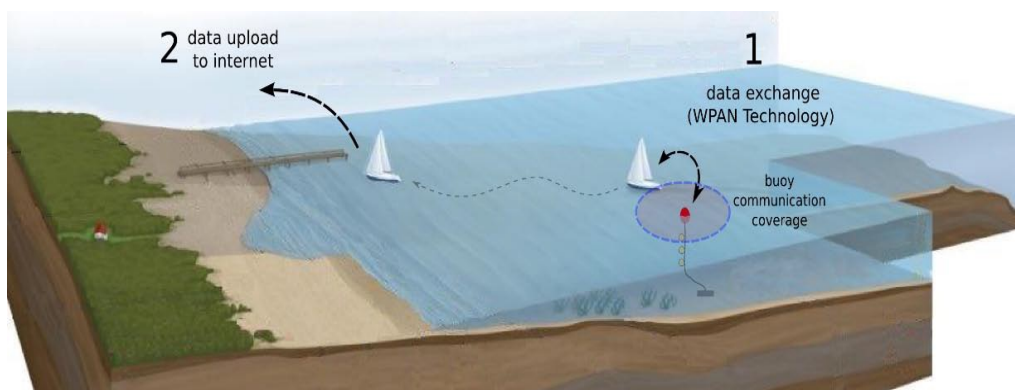


Fig.1 - Sistema de funcionament del projecte KdUINO.

La boia ha de generar una xarxa inalàmbrica d'àrea personal (WPAN) que estarà situada a l'aigua. Quan el dispositiu mòbil de l'usuari estigui dins del radi de cobertura de la WPAN de la boia, aquest ha de poder connectar-se amb la boia per tal de poder llegir les dades mesurades (punt 1). Quan el dispositiu mòbil disposi de les mesures, enviarà les dades a un servidor central a través d'una connexió a internet (punt 2), per tal de disposar d'una base de dades de les diferents mesures realitzades.

El sistema de captació de dades de la boia està format, a més del sistema de control format per la placa Arduino, d'un conjunt de sensors disposats de forma vertical que mesuren una certa quantitat de llum (Fig.2). Mitjançant la quantitat de llum que enregistri cada sensor i emprant la Llei de Beer-Lambert, també coneguda com la Llei de Beer, es pot establir una relació empírica entre l'absorció de la llum i les propietats del material que travessa. En funció de la Llei de Beer i tenint present que el medi material és l'aigua, es pot establir una constant anomenada, coeficient

d'atenuació difusa (k_d), que estima la claredat de l'aigua mesurant la capacitat de penetració de la radiació solar que incideix. D'aquesta manera es pot establir un rang dins del qual la llum a diferents longituds d'ona és atenuada en funció de la profunditat.

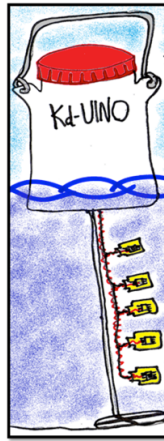


Fig.2 - Sistema de control i captació kduino.

ÍNDEX

1. Introducció	1
1.1 Justificació del Treball	1
1.2 Context	1
1.3 Objectius del Treball	3
1.4 Descripció dels capítols	4
1.5 Planificació temporal del treball.....	5
1.6 Anàlisi del producte obtingut	6
2. Estat de l'art	8
2.1 Perquè es mesura la transparència de l'aigua	8
2.2 Història de la mesura de la transparència de l'aigua	8
2.3 La llei de Beer-Lambert i el coeficient d'atenuació difusa	9
2.4 Mètodes de mesura	9
2.5 L'ús de les noves tecnologies	12
2.6 Projectes actuals en aquest àmbit	13
3. Disseny	14
3.1 Models estàndards de disseny	14
3.2 Procés de disseny	16
3.3 Definició dels requeriments	16
3.3.1 Definició dels requeriments funcionals	16
3.3.2 Definició dels requeriments no funcionals	17
3.3.2.1 Definició dels requeriments tecnològics.....	17
3.3.2.2 Definició dels requeriments de l'usuari	18
3.4 Anàlisi i disseny del software	18
3.4.1 Establiment de les funcionalitats generals	20
3.4.2 Interactivitat amb el dispositiu	23
3.4.3 Interactivitat amb la base de dades.....	25
3.5 Implementació i prova	26
4. Implementació	28
4.1 Estructura dels arxius	28
4.2 Estructura dels layouts	30
4.3 Arxius complementaris	31
4.4 Estructura general de l'aplicació	32
4.5 Menú principal	32
4.5.1 Descripció.....	32

4.5.2 Funcionalitat	33
4.6 Menú make a kduino	35
4.6.1 Descripció.....	35
4.6.2 Funcionalitat	36
4.7 Menú de connexió al kduino.....	38
4.7.1 Descripció.....	38
4.7.2 Funcionalitat	38
4.8 Menú control.....	40
4.8.1 Descripció.....	40
4.8.2 Funcionalitat	41
4.8.2.1 SEAT TIME	41
4.8.2.2 START	42
4.8.2.3 STOP	43
4.8.2.4 GET DATA	44
4.8.2.5 DELETE DATA.....	45
4.8.2.6 CALIBRATION	45
4.8.2.7 READ TIME	47
4.8.2.8 TEST SENSORS	47
4.9 La base de dades	48
5. Proves i avaluació	52
5.1 Entorn de proves	52
5.2 Demostració	53
5.3 Resultat de les proves realitzades	53
6. Conclusions i línies de futur	56
6.1 Conclusions	56
6.2 Línies de futur.....	58
7. Glossari	59
8. Bibliografia	60
9. Annexos	62

ÍNDIX DE FIGURES

FIGURA 1.- Sistema de funcionament del projecte KdUINO	i
FIGURA 2.- Sistema de control i captació kdUINO	ii
FIGURA 3.- Logo del projecte Citclops	3
FIGURA 4.- Distribució de les boies Kduino arreu del món	2
FIGURA 5.- Procés de creació d'una boia Kduino	2
FIGURA 6.- Diagrama de Gantt del projecte	5
FIGURA 7.- Característiques que determinen la qualitat d'un software	6
FIGURA 8.- Disc de Secchi	10
FIGURA 9.- Tub de mesura de la tuerbolesa	10
FIGURA 10.- Perfilador marí	11
FIGURA 11.- Principi de funcionament d'un Nefelòmetre	12
FIGURA 12.- Ús d'imatges per a determinar la transparència de l'aigua	13
FIGURA 13.- Model de cicle de vida en cascada	14
FIGURA 14.- Model de cicle de vida Sashimi	15
FIGURA 15.- Model de cicle de vida en V	15
FIGURA 16.- Procés de disseny del software	16
FIGURA 17.- Versions d'Android i el seu percentatge d'ús	19
FIGURA 18.- Pantalla de selecció de l'API del programari Android Studio	20
FIGURA 19.- Diagrama de blocs del menú principal	21
FIGURA 20.- Diagrama de flux del funcionament del menú makeakduino	21
FIGURA 21.- Diagrama de flux de la connexió Bluetooth	22
FIGURA 22.- Interconnectivitat entre dispositius	23
FIGURA 23.- Send Data del Kduino	23
FIGURA 24.- Start measurements del Kduino	23
FIGURA 25.- Stop measurements del Kduino	24
FIGURA 26.- Send data del Kduino	24
FIGURA 27.- Delete data del Kduino	24
FIGURA 28.- Calibration sensors del Kduino	24
FIGURA 29.- Programació de l'hora i la data del Kduino	24
FIGURA 30.- Test sensors del Kduino	24
FIGURA 31.- Interconnexió Smartphone-DataBase	25
FIGURA 32.- Plana de descàrrega de XAMPP	25
FIGURA 33.- Panel de control de XAMPP	33

FIGURA 34.- Pantalla d'activació del mode desenvolupament d'Android	27
FIGURA 35.- Pantalla d'activació de la implementació en el dispositiu mòbil	27
FIGURA 36.- Estructura general dels arxius del projecte	28
FIGURA 37.- Estructura general de les pantalles del projecte	30
FIGURA 38.- Arxius complementaris	31
FIGURA 39.- Diagrama de navegació de l'aplicació	32
FIGURA 40.- Notificació d'existència de dades per enviar a la base de dades	34
FIGURA 41.- Diagrama de flux del funcionament de la inserció de dades a la DB	35
FIGURA 42.- Notificació de no existència de xarxa per obtenir la localització	37
FIGURA 43.- Missatge de diàleg a l'usuari	37
FIGURA 44.- Estat del Bluetooth	38
FIGURA 45.- Avís de no existència de Bluetooth	39
FIGURA 46.- Activació del Bluetooth	39
FIGURA 47.- Llista de dispositius cercats	39
FIGURA 48.- Avís d'error en la connexió	40
FIGURA 49.- Protocol SEAT TIME	41
FIGURA 50.- Diàleg de confirmació del SEAT TIME	42
FIGURA 51.- Protocol START	42
FIGURA 52.- Diàleg de confirmació del START	43
FIGURA 53.- Protocol STOP	43
FIGURA 54.- Diàleg de confirmació del STOP	43
FIGURA 55.- Protocol GET DATA	44
FIGURA 56.- Diàleg de confirmació del GET DATA	44
FIGURA 57.- Protocol DELETE DATA	45
FIGURA 58.- Diàleg de confirmació del DELETE DATA	45
FIGURA 59.- Protocol CALIBRATION	46
FIGURA 60.- Diàleg de confirmació de calibració	46
FIGURA 61.- Confirmació de gravació de les dades	46
FIGURA 62.- Protocol READ TIME	47
FIGURA 63.- Confirmació de la lectura de la data i l'hora	47
FIGURA 64.- Protocol TEST SENSORS	48
FIGURA 65.- Pantalla Test Results	48
FIGURA 66.- Creació de la base de dades en phpMyAdmin	48
FIGURA 67.- Plana "Chequeo de seguridad"	49
FIGURA 68.- Camps de la taula kduino	49
FIGURA 69.- Obtenció de la ip del servidor	51
FIGURA 70.- Resultat de la inserció de dades.....	51

FIGURA 71.-Tablet Galaxy Tab3	52
FIGURA 72.- Smartphone Galaxy Express 2	52

ÍNDIX DE TAULES

TAULA 1.- Distribució del temps durant el projecte	5
TAULA 2.- Resultats de l'avaluació del producte resultant	54

1. Introducció

1.1 Justificació del Treball

El present treball vol sintetitzar en un projecte real els coneixements adquirits en diverses assignatures realitzades al llarg de la carrera, entorn a un camp d'aplicació sobre el qual vull desenvolupar els meus estudis futurs. Per tant, aquest projecte m'ha de permetre adquirir els coneixements fonamentals per poder seguir adquirint aptituds entorn al disseny de sistemes per a dispositius mòbils.

En segon lloc, i degut a que es troba englobat dins d'una col·laboració amb el Centre Superior D'investigacions Científiques (CSIC), hi ha una motivació especial per la satisfacció de col·laborar en un projecte amb investigadors de nivell Europeu. El qual, ha de permetre que investigadors d'arreu del món puguin utilitzar les dades mesurades per establir mapes de la qualitat de l'aigua i així, establir actuacions per millorar el medi ambient amb l'ajuda de l'aplicació resultant del treball.

1.2 Context

El projecte Citclops [1], desenvolupat per l'Institut de Ciències del Mar (ICM), és un projecte que ha desenvolupat un sistema d'avaluació de l'atenuació de la llum solar a l'aigua, mitjançant una boia anomenada Kduino amb un sistema de sensors per a la captació de la llum solar. Les mesures realitzades han de proporcionar una base de dades per a investigadors d'arreu del món. Això ha de proporcionar la possibilitat de crear mapes de la qualitat de l'aigua, tant en costes com en llacs o estanys.



Fig.3 – Logo del projecte Citclops.

<http://www.citclops.eu>

Encara que sigui un projecte subvencionat per la Comunitat Europea, l'abast és mundial. L'adquisició de dades provenen de diferents llocs del món i continua la

seva expansió. En el mapa següent es pot veure la distribució de les boies disponibles en l'actualitat:



Fig.4 – Distribució de les boies Kduino arreu del món.

http://kduino.cartodb.com/viz/65974162-669f-11e4-a6aa-0e9d821ea90d/embed_map

Un dels objectius principals del projecte és la divulgació de l'ús d'aquests sistemes en la ciutadania de manera que amb un mínim esforç i diners, les persones que desenvolupin activitats en aquests llocs puguin col·laborar amb els investigadors. Amb aquest objectiu la creació de qualsevol sistema dins del projecte ha de ser d'ús fàcil i entenedor. Per tal de facilitar la generalització del seu ús, tant l'aplicació de control com les instruccions de muntatge estan disponibles a la plana web del projecte i permeten a l'usuari poder muntar una boia Kduino de manera fàcil.

A continuació es mostra el procés de creació d'una boia Kduino:

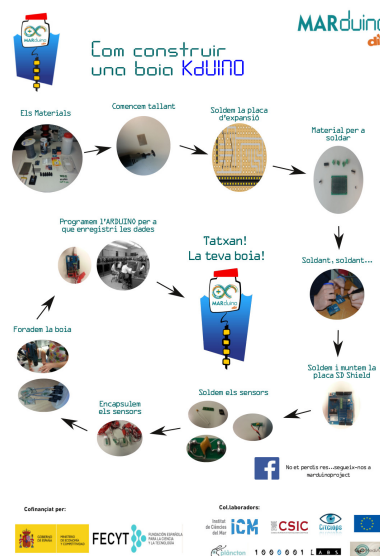


Fig.5 – Procés de creació d'una boia Kduino.

1.3 Objectius del Treball

El propòsit del present treball engloba el disseny d'una aplicació per a dispositius mòbils que proporcioni a l'usuari les funcionalitats de control sobre una boia Kduino.

El sistema Kduino en l'actualitat és un projecte que està en funcionament i per tant ja consta d'una primera versió de software de control. La versió actual del software de control va ser desenvolupada amb l'Appinventor. L'objectiu de la versió que es desenvolupa en aquest treball, és la de proporcionar al projecte d'un software elaborat amb un llenguatge de programació més robust que permeti als investigadors poder seguir desenvolupant-lo en un futur immediat.

Per al disseny de l'aplicació s'ha considerat un programari basat en codi Java i desenvolupat per el software Android Studio que ha de poder ser carregat en un dispositiu mòbil, concretament un Smartphone.

Per tant els objectius principals d'aquest treball són dos:

1. Dissenyar una aplicació mòbil en llenguatge JAVA per a dispositius Android, mitjançant el software Android Studio, que ha de permetre connectar-se a la boia Kduino mitjançant tecnologia Bluetooth, per a rebre dades recollides pels sensors. Així com gestionar el control de la mateixa per a permetre la seva calibració, l'actualització de la data i l'hora, resetejar-la, etc.
2. Crear una base de dades a partir de l'obtenció de les mesures dels sensors de la boia per a que aquestes puguin ser analitzades posteriorment per la comunitat científica.

1.4 Descripció dels capítols

Per a facilitar la lectura de la memòria, l'estructura de la resta del document es detalla a continuació:

1 – Introducció: la finalitat d'aquest capítol és la de posar en context al lector, presentar la problemàtica a resoldre i establir un planning dels processos a seguir per a resoldre aquesta problemàtica.

2 – Estat de l'art: es presenta les línies de coneixement necessàries per a la interpretació dels requeriments de l'aplicació i de les necessitats globals que s'han d'assolir per a englobar l'aplicació dins del projecte global.

3 – Disseny: tot sistema de desenvolupament d'un software implica uns recursos, temps i un conjunt de requeriments. En aquest capítol es tracta el procés de disseny que es du a terme per tal d'assolir els objectius, avaluant els recursos necessaris a partir dels requeriments i es mostra com es planteja la resolució de les diferents problemàtiques que s'han anat plantejant durant el disseny de l'aplicació.

4 – Implementació: es descriu en detall el procés de funcionament, així com es detallen les diferents funcionalitats en cada procés de l'aplicació.

5 – Proves i avaluació: es detallen les proves realitzades, així com el resultat obtingut i es mostren els dispositius que s'han utilitzat per a realitzar-les. També es facilita les adreces web per tal de poder veure uns vídeos per de poder visualitzar el funcionament del programa.

6 – Conclusions i línies de futur: es comenten les conclusions extretes del treball, on s'analitza de manera objectiva el resultat obtingut i el procés de desenvolupament, així com un anàlisi de les competències adquirides i l'experiència de participar en un projecte real. En l'apartat de línies de futur s'analitza el procés que ha de experimentar el programa en un futur, com a conseqüència de ser una eina de treball en un projecte més complex, i quins han de ser els canvis necessaris en el producte obtingut per a ser un programari aplicable en el projecte a curt termini.

1.5 Planificació del Treball

El projecte ha tingut una durada d'uns 4 mesos, en els quals s'han dut a terme les següents tasques:

Fase	Durada	Inici	Fi
Presentació del projecte	10 dies	26/02/2015	11/03/2015
Determinació de l'estat de l'art	15 dies	12/03/2015	01/04/2015
Disseny i implementació	82 dies	2/02/2015	27/05/2015
Entrega del projecte final	15 dies	28/05/2015	17/06/2015
Total	122 dies	26/02/2015	17/06/2015

Taula 1 – Distribució del temps durant el projecte.

La distribució detallada de les tasques i la seva duració es poden apreciar en el següent diagrama:

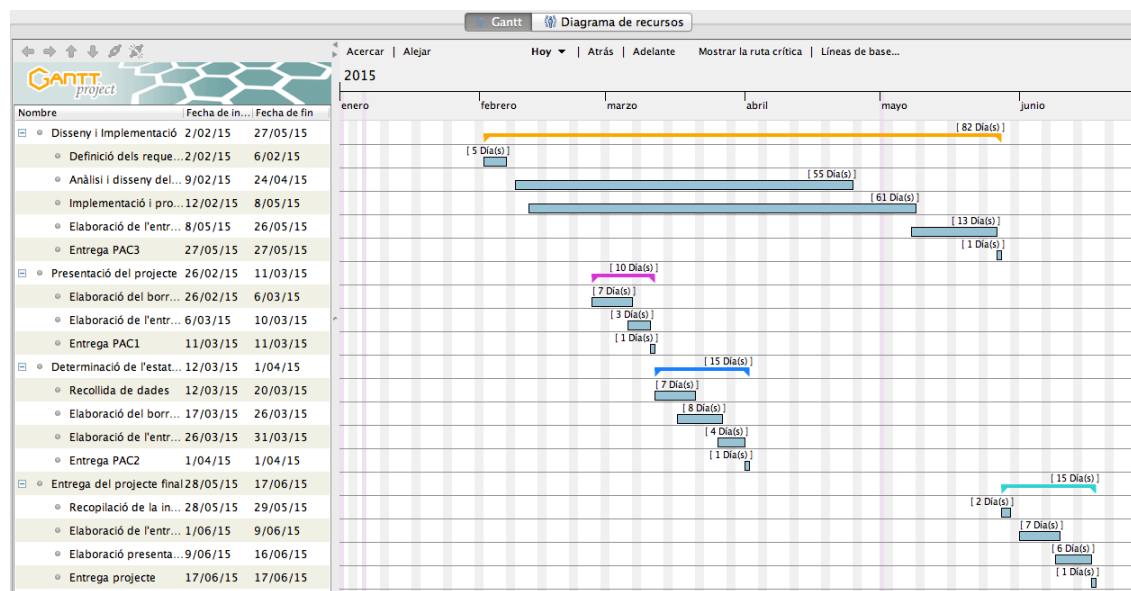


Fig.6 – Diagrama de Gantt del projecte.

1.6 Anàlisi del producte obtingut

Existeixen un gran nombre de models de qualitat. Per tal de determinar un mínim de qualitat en el resultat final, en funció de les característiques que estableix l'ISO-9126 Software Quality Characteristics [2], que proporciona diferents paràmetres d'avaluació per aconseguir aquest propòsit. Aquest paràmetres són els següents:

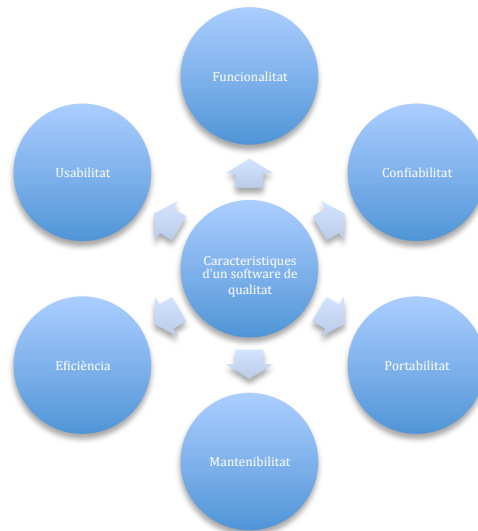


Fig.7 - Característiques que determinen la qualitat d'un software.

En el cas del present projecte s'han determinat les característiques en funció de les anteriors condicions:

1. Funcionalitat: el programari és una eina més dins del sistema d'avaluació de l'atenuació de la llum a l'aigua, per tant ha d'interactuar amb el circuit de control de la boia.
2. Usabilitat: la voluntat del ICM és la de generalitzar l'ús d'aquests sistemes, d'aquesta manera s'ha de tenir present que serà utilitzat per persones que no tenen perquè estar familiaritzades amb el dispositiu, així doncs proporcionar un context de fàcil utilització.
3. Eficiència: proporcionar la màxima eficiència maximitzant l'ús de recursos del sistema, amb l'estructuració i compartiment de funcionalitats comuns entre mètodes diferents.

4. Mantenibilitat: el codi ha estat estructurat i presenta una programació intuïtiva de manera que sigui fàcil la seva comprensió per al seu manteniment, a més de proporcionar facilitats de desenvolupament per a futures versions.

5. Portabilitat: ha de tenir-se en compte que cada persona dispondrà d'un dispositiu mòbil diferent. Per tant, tot i que el programari utilitzat està desenvolupat en un entorn Android, pot haver-hi característiques diferents entre dispositius, que podrien causar un funcionament anòmal o diferent en cada cas. S'haurà d'intentar homogeneïtzar el comportament indistintament del dispositiu.

6. Confiabilitat: el sistema ha de presentar una estructura que presenti robustesa en front a errors i assegurar així el seu funcionament en entorns de llarg períodes de temps i de màxima exigència. Per aquest propòsit es diferencien els errors de funcionament segons si són errors de la pròpia aplicació o externs.

2. Estat de l'art

2.1 - Perquè es mesura la transparència de l'aigua?

La llum solar que incideix a l'aigua dels oceans és fonamental per a l'ecosistema marí. La quantitat de llum que hi incideix, així com la profunditat a la qual arriba, és determinant per a la vida de les plantes i organismes que hi viuen, tanmateix determina aspectes com la distribució, afectant així la vida de moltes comunitats marines.

La mesura de la transparència de l'aigua és fonamental, ja que serveix com a indicador de diferents fenòmens que succeeixen als oceans, com ara: la quantitat de contaminació degut a fenòmens tòxics o la determinació del creixement o disminució natural de certes algues que formen part de l'ecosistema.

L'atenuació de la llum a l'aigua, es degut a la presència de diferents components com: algues que absorbeixen la llum per a fer la fotosíntesi, matèria dissolta o zooplàncton que dispersen la llum. Aquests components són anomenats components òpticament actius.

2.2 - Història de la mesura de la transparència de l'aigua

L'interès de l'esser humà per paràmetres de l'aigua com la transparència, es remunta al segle XVII. Henry Hudson [3] va ser dels primers científics en descriure aspectes de l'aigua com la transparència. Però, el desenvolupament de tècniques de mesures remotes (tècniques que no requereixen la necessitat de disposar de mostres d'aigua) es remunten al segle XIX, on a l'any 1865 Angelo Secchi [4] com a encàrrec del cap de la "Naviera Papal" va inventar un dispositiu per a mesurar la transparència de l'aigua, anomenat disc de Secchi. El primer disc va ser provat des de'l vaixell papal " Immacolata Concezion " el 20 d'Abril de 1865.

Altres intents de quantificar la turbidesa data dels anys 1900 on Whipple i Jackson van desenvolupar un dispositiu anomenat diaphanometer que posteriorment va ser conegut com turbidímetre de vela de Jackson. Aquest dispositiu consisteix en un recipient de cristall i una vela mitjançant la qual, si es projectava la llum sobre el recipient, es podia fer una comparació visual de la suspensió amb una solució d'àcid de silici.

Però no va ser fins a començament dels anys 70 quan es van començar a desenvolupar tècniques amb l'utilització de satèl·lits, aeronaus i dispositius òptics que són els que s'utilitzen actualment, com per exemple els perfiladors marins.

2.3 - La Llei de Beer-Lambert i el coeficient d'atenuació difusa

La llei de Beer-Lambert [5] proporciona un mètode empíric per a relacionar l'absorció de la llum i les propietats del material que travessa. Pel cas de l'aigua, explica que la intensitat decreix de manera exponencial en funció de la profunditat. La manera de calcular-ho és la següent:

$$E = E_0 e^{-K_d Z}$$

on:

E = Irradiació a una profunditat donada

E_0 = Irradiació a la superfície

K_d = Coeficient d'atenuació difusa

Z = Profunditat

El paràmetre K_d , estima la claredat o terbolesa de l'aigua mesurant la capacitat de penetració de la radiació solar que incideix, és a dir, presenta el rang dins del qual la llum a diferents longituds d'ona és atenuada amb la profunditat. Té diverses aplicacions dins de l'òptica dels oceans, com la relació amb la concentració de fitoplàncton i sediments orgànics i inorgànics dissolts a l'aigua marina.

2.4 - Mètodes de mesura

Existeixen un gran nombre de dispositius per a mesurar la transparència de l'aigua, existeixen mètodes basats en la variació de la resistència elèctrica o zona elèctricament sensible, els mètodes que utilitzen sistemes basats en el bloqueig de la llum i aquells que utilitzen un sistema basat en la dispersió de la llum. A continuació es comenten els dispositius més utilitzats per a la mesura de la transparència de l'aigua:

- **Disc de Secchi [6]:** consisteix en un disc d'uns 20cm de diàmetre que penja d'un fil i sobre el qual estan dibuixats quatre quadrants pintats de color blanc i negre, tal i com es pot observar a la Fig.8:



Fig.8 - Disc de Secchi.

El funcionament del disc de Secchi és el següent: el disc es submergeix a l'aigua i es deixa que es vagi enfonsant fins que es deixi de veure, en aquest moment es pren mesura de la profunditat, mesura que es pot prendre a través de marques de longitud que poden estar incorporades en el fil que s'objecta el disc. Un cop realitzada la primera mesura, es torna a pujar el disc fins que aquest torni a ser visible. En aquest punt es torna a mesurar la profunditat a la que es troba. A continuació es procedeix a realitzar un promig d'aquestes dues mesures per obtenir una mesura depenent de la transparència de l'aigua.

Cal destacar que aquest mètode és el més adient per a mesures en llacs, mars i oceans, per la seva senzillesa i efectivitat.

Però aquest sistema també presenta diverses limitacions:

1. Ha de ser utilitzat sempre des de una embarcació o lloc proper a la superfície de l'aigua.
 2. En llocs amb vegetació marina molt densa el disc pot deixar de veure a causa de la vegetació i no perquè la transparència de l'aigua sigui baixa.
- **Tub de terbolesa [7]:** el tub per a la mesura de la terbolesa és un mecanisme com el que es pot veure en la Fig. 9.

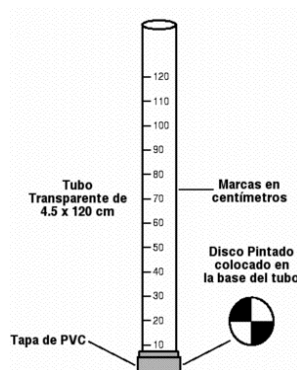


Fig.9 - Tub de mesura de la terbolesa.

El funcionament d'aquest dispositiu és el següent: es verteix l'aigua de la mostra dins del tub fins que la base del tub deixi de ser visible mirant-lo en línia recta des de la part de dalt del tub, arribat aquest punt s'ha de girar el tub per tal d'apreciar si la imatge de la part inferior, és a dir, les parts blanques i negres es poden distingir. Si no es poden distingir, s'anota la quantitat d'aigua que hi ha al tub amb l'ajut de les marques en centímetres. D'aquesta manera es pot realitzar un promig de varies mesures per a determinar la torbesa de l'aigua en funció d'una altre mostra estàndard.

- **Perfiladors marins [8]:** és un instrument que registra diferents paràmetres en una columna d'aigua a través de sensors. Alguns dispositius són capaços de mesurar la radiació solar obtinguda a distintes profunditats. Però aquests aparells són molt costosos i consumeixen molta potència. Per aquest motiu normalment s'acostumen a utilitzar de manera puntual o durant períodes de temps molt reduïts des de vaixells. A la Fig.10 és mostra un perfilador marí:



Fig.10 - Perfilador marí.

De perfiladors trobem dos tipus en funció del mode de funcionament:

- **Cosinus collector:** els quals presenten un sensor direccional. Aquests sensors només capten la radiació que incideix de manera perpendicular al pla del sensor.
- **Scallar:** són aquells perfiladors que presenten una recepció omnidireccional, és a dir capten la radiació en tots els sentits, tant la radiació directa com la radiació provinent de diverses reflexions.

- **Nefelòmetres o turbidòmetre [9]:** és un aparell que serveix per a la mesura de les partícules en suspensió en un líquid. El funcionament d'aquest dispositiu és el que es mostra a la Fig.11:

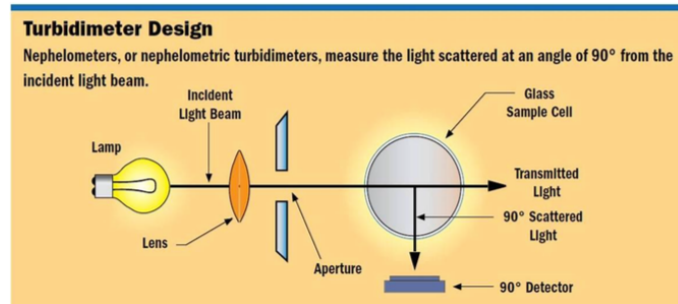


Fig.11 - Principi de funcionament d'un Nefelòmetre.

Com es mostra a la figura anterior, el funcionament es basa en utilitzar una cubeta amb la mostra a analitzar per la qual es fa passar un feix de llum. Amb un angle de 90° respecte de la font lluminosa es col·loca una foto cèl·lula. La quantitat de llum que la foto cèl·lula detecti és llavors, una funció dependent de la forma, el color i la reflectivitat de les partícules. Establint llavors una correlació entre la terbolesa i les partícules suspeses.

2.5 - L'ús de les noves tecnologies

L'ús de les noves tecnologies en aquest àmbit és molt divers i molt beneficiós. L'aplicació de sistemes informàtics permeten analitzar acuradament les dades mesurades. D'aquesta manera es poden establir relacions ràpides i fiables entre les dades mesurades i la teoria que es troba darrere d'aquest àmbit.

Un dels objectius del CSIC en aquest sentit, és generalitzar l'ús dels dispositius que elaboren per tal d'aconseguir que la ciutadania s'impliqui, i així abastar més zones i aconseguir augmentar la base de dades, per establir d'aquesta manera mapes de zones geogràfiques sobre la qualitat de l'aigua. Per aquesta finalitat i per facilitar l'ús dels dispositius a persones que no tenen perquè conèixer la teoria que es necessita, s'integren amb funcionalitats sobre Smartphones per tal d'afavorir la interacció persona-màquina i així fer més funcional aquests sistemes.

Aquests sistemes a més, poden interactuar amb satèl·lits i dispositius GPS per tal d'obtenir dades suplementàries, com ara la localització específica del punt de mesura, dades com l'hora i la data de la mesura, etc.

2.6 - Projectes actuals en aquest àmbit

En l'actualitat existeixen una gran quantitat de projectes relacionats amb la turbidesa i atenuació de l'aigua que engloben diferents àmbits. Entre aquests projectes, es troben estudis de l'efecte de la turbidesa en: l'agricultura, el consum humà, la contaminació d'oceans i mars, efectes sobre la fauna, etc.

Un projecte realitzat per a mesurar l'atenuació de l'aigua en diferents emplaçaments als Estats Units, és el projecte The Secchi Dip-In [10]. En aquest projecte i a través de voluntaris es monitoritza l'atenuació de l'aigua en diferents emplaçaments mitjançant l'utilització del Disc de Secchi, per tal de facilitar una base de dades als investigadors.

També trobem projectes relacionats més propers, entre els quals es troben els estudis que realitza el Centre Superior d'Investigacions Científiques (CSIC) [1]. En aquest sentit el CSIC està duent a terme un parell de projectes relacionats. En tots dos projectes tracten d'elaborar diferents sistemes de mesura, per tal de mesurar adequadament i segons les característiques geogràfiques del lloc, l'atenuació amb el mètode més adient en cada cas.

El primer d'ells és la mesura de l'atenuació de l'aigua mitjançant l'ús de imatges i programes d'anàlisi de les dades. Aquest sistema consisteix en establir una relació entre la distància entre dues fotografies i la relació entre la qualitat d'imatge presa de les dues imatges sota l'aigua, per tal de determinar un indicador de transparència de l'aigua en les tres bandes de colors primaris amb l'ajut d'un software especialitzat.

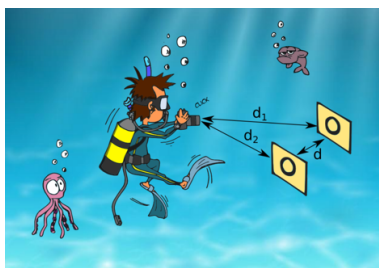


Fig.12 - Ús d'imatges per a determinar la transparència de l'aigua.

Un segon projecte, l'objecte del present treball, és el de determinar l'atenuació de l'aigua amb l'ajut de tres sensors que mitjançant la traducció de la lluminositat captada en energia elèctrica, i sabent la profunditat a la qual es troben així com la distància entre ells, s'aconsegueix determinar la transparència de l'aigua en funció d'una constant.

3. Disseny

3.1 – Models estàndards de disseny

La planificació de qualsevol disseny està determinat tant per les funcionalitats que ha d'assumir el programa com per el sistema de desenvolupament del mateix. Qualsevol enginyeria de software es basa en l'establiment d'etapes, que engloben des de'l concepte inicial fins a l'entrega del producte final. Aquestes etapes determinen les principals fases de desenvolupament, així com proporciona una manera d'administrar el progrés del producte. Aquest procés estableix l'anomenat cicle de vida del software [11]. Existeixen una gran quantitat de models de cicle de vida amb les etapes predefinides per a portar a terme el procés de creació del programari. Els models de cicle de vida més usuals per a projectes petits són els que es mostren a continuació:

1. Model en Cascada: determinada per un procés seqüencial a través de cinc etapes, en la que cada etapa ha d'esperar la finalització de l'etapa anterior (Fig.13).

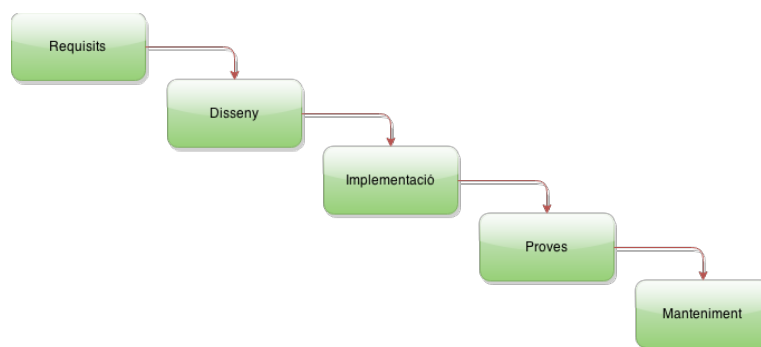


Fig.13 – Model de cicle de vida en cascada.

Aquest model és especialment utilitzat per projectes on els requeriments estan molt definits i no canvien al llarg del procés de desenvolupament. És un procediment en el qual el procés està molt organitzat i no hi ha possibilitat de mesclar les diferents fases. Però, aquest mateix avantatge pot resultar un inconvenient en quant a què, els requisits i la problemàtica de l'etapa futura han d'estar totalment previstos en l'etapa anterior. Per tant, el fet de ser una estructura tant rígida i totalment seqüencial no permet que els problemes no es

facin visibles fins a les darreres etapes en les que es prova el programa, la qual cosa pot suposar un greu problema.

2. Model Sashimi: és un model molt semblant al procés en cascada, però ara s'estableixen sis etapes. L'avantatge principal d'aquest mètode és que els processos estan sobreposats, per tant en qualsevol etapa es pot retrocedir a una etapa anterior per tal de tornar a determinar els requeriments de l'etapa actual. En aquest model doncs, els errors poden ser detectats en etapes inicials i es poden establir les resolucions a mesura que es va desenvolupant el programari.

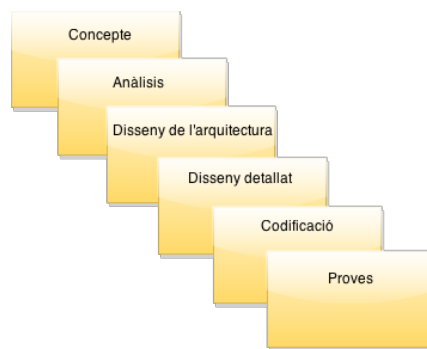


Fig.14 – Model de cicle de vida Sashimi.

3. Model en V: la principal diferència d'aquest model és que inclou en cada fase un procés de verificació i validació com es pot apreciar a la Fig.15. En la part esquerra s'estableixen les etapes del desenvolupament pròpiament dites, i a la dreta es troben les fases de verificació i validació que estableixen per a cada etapa de desenvolupament. El procés segueix una forma de V, per la qual cosa un cop s'arriba a l'etapa de codificació es torna cap a dalt i es torna a verificar que els requeriments de cada etapa s'han assolit favorablement. L'avantatge principal és que en cada fase s'obté un resultat entregable, però un dels desavantatges principals és que quan es troben els problemes no hi ha un camí clar per a resoldre la problemàtica.

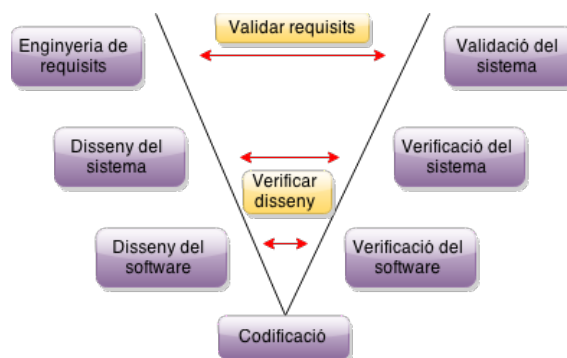


Fig.15 – Model de cicle de vida en V.

3.2 – Procés de disseny

El programa s'ha desenvolupat de manera progressiva en un anàlisi constant dels requeriments que determina el sistema de control, ja que aquest està sent utilitzat en l'actualitat i ha de ser compatible amb el sistema a desenvolupar. El cicle de vida ha estat determinat per un procés en cascada amb retroalimentació constant, per tal d'obtenir el millor de cada model analitzat anteriorment, en el qual, es permet un retrocés a etapes anteriors seguint un procés com el que es mostra a continuació:

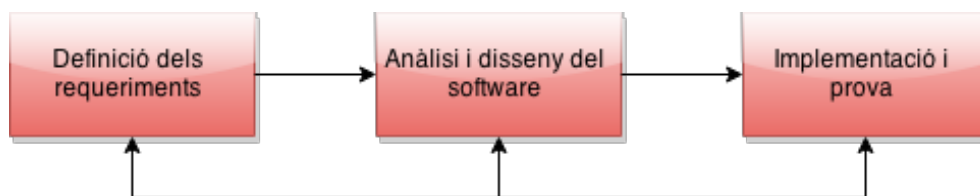


Fig.16 - Procés de disseny del software.

El desglossament de les diferents fases són les que es detallen en els apartats següents.

3.3 - Definició dels requeriments

3.3.1 - Definició dels requeriments funcionals

En aquesta fase s'estableix les funcionalitats que ha d'assolir el programa. Les funcionalitats del sistema en aquest cas estan molt definides. Així doncs s'estableixen les següents funcions:

1. L'aplicació ha de constar d'un menú principal per a que l'usuari pugui escollir si vol enregistrar les dades d'usuari i la boia, si es vol prendre mesures o configurar la boia o només obtenir informació addicional de l'aplicació.
2. Ha de permetre, quan s'hagi seleccionat l'opció de donar d'alta una boia, enregistrar en una base de dades la següent informació:
 - Nom de la boia.
 - Nombre de sensors.
 - Posició de cada sensor.

- Nom de la persona que crea la boia.
- Posició d'on es situa la boia.
- Informació dels sensors.
- Data de l'adquisició de les dades.
- Hora de l'adquisició.

3. El sistema ha de poder connectar-se a la boia a través d'una connexió Bluetooth amb el microcontrolador que aquesta incorpora per permetre les següents funcionalitats:

- Rebre les dades recollides pels sensors.
- Esborrar les dades de la memòria interna de la boia.
- Programar el rellotge intern de la boia.
- Realitzar una mesura.
- Realitzar calibració dels sensors.
- Llegir la data i l'hora programades al Kduino.

4. Dotar el sistema d'interconnectivitat amb una base de dades per enregistrar la informació rebuda de la boia.

3.3.2 - Definició dels requeriments no funcionals

3.3.2.1 - Definició dels requeriments tecnològics

Els requeriments tecnològics requerits per al entorn de desenvolupament de l'aplicació són els següents:

- Mac OS X 10.8.5 o superior.
- Java Runtime Environment (JRE) 6.
- Java Development Kit (JDK) 7.
- Android Studio IDE.
- Android Studio SDK tools Revision 24.2.0.
- MySQL Database Server.
- PhpMyAdmin 4.3.12.
- Smartphone.

3.3.2.2 - Definició dels requeriments de l'usuari

Per a que l'usuari pugui disposar de l'aplicació els requeriments dels quals ha de disposar són els següents:

- Dispositiu amb sistema operatiu Android (recomanable v.4.2 o superior).
- Disposar d'una boia Kduino.
- Disposar d'un ordinador amb connexió a la xarxa d'Internet.
- Tenir instal·lat PhpMyAdmin 4.3.12.
- Tenir instal·lat MySQL Database Server.
- Tenir instal·lat el servidor web local Apache.

3.4 - Anàlisi i disseny del software

La determinació del disseny implica l'estudi de les diferents plataformes possibles tant per al seu desenvolupament, com pel que fa al seu ús en entorns dels dispositius mòbils que hi ha actualment en el mercat. Actualment existeixen dues plataformes iOS i Android. Qualsevol dels dos sistemes operatius permeten desenvolupaments de tercers, però les aplicacions per a un sistema iOS han de ser escrites i compilades especialment per aquest sistema. El sistema Android és una plataforma oberta, més flexible en el disseny i la implementació i d'igual ús que un sistema iOS. A més, utilitza un llenguatge de programació JAVA, que ha estat l'eina de programació de diverses assignatures de la carrera, per tant per tots aquests motius s'ha optat per desenvolupar-lo per aquest sistema.

El sistema operatiu Android [12] desenvolupat per Google ha desenvolupat diverses versions al llarg del temps. La primera versió comercial correspon a la versió 1.0 Apple Pie llançada al mercat al 2008 per al dispositiu HTC Dream. A partir del 2009 les següents versions han anat incorporant millores respecte a la versió anterior i han sigut anomenades amb un nom en clau, sempre relacionat amb dolços i seguint un ordre alfabètic: Apple Pie (v.1.0), Banana Bread (v.1.1), Cupcake (v.1.5), Donut (v.1.6), Éclair (v.2.0/2.1), Froyo (v.2.2.x), Gingerbread (v.2.3.x), Honeycomb (v.3.x), Ice Cream Sandwich (v.4.0.x), Jelly Bean (v.4.1/2/3), KitKat (v.4.4) i Lollipop (v.5.0/5.1).

Qualsevol versió del sistema operatiu Android està identificada per un nivell d'API (Application Programming Interface) que identifica els serveis o funcions que el sistema operatiu ofereix al programador. Així el programador treballa amb una sèrie

de funcions de les quals només coneix els seus paràmetres i els seus valors de retorn, sense hagi de fer ús de la complexitat d'accés al sistema o aplicació, és a dir ofereix un cert nivell d'abstracció que emmascara el sistema. La darrera versió d'Android és la versió 5.1 Lollipop que conté l'API 22. A continuació es mostra el desenvolupament que han seguit les diverses versions d'Android, així com un gràfic que presenta quin és el percentatge d'utilització de cada versió en els dispositius actuals :

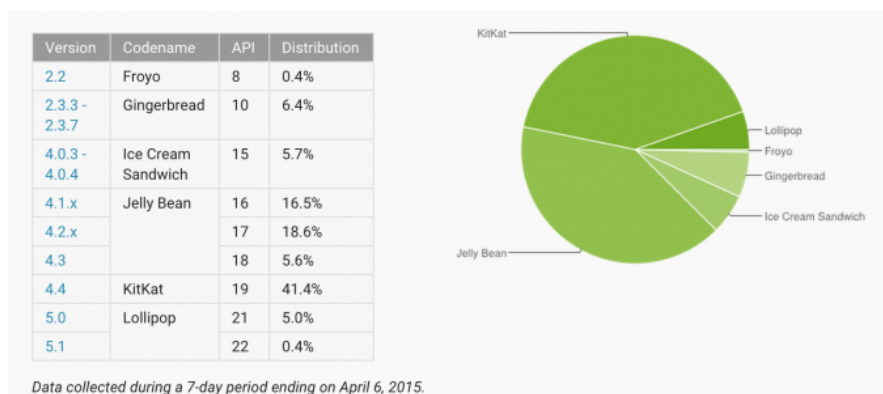


Fig.17 – Versions d'Android i el seu percentatge d'ús.

En la Fig.17 es pot observar com actualment el sistema que presenta un major percentatge d'utilització en els dispositius mòbils actuals és el KitKat, per la qual cosa, si es vol que l'aplicació pugui funcionar en el major nombre de dispositius actuals, s'haurà d'adreçar a usuaris amb sistema operatiu KitKat que correspondrà a una API 19. Aquest fet és essencialment important, perquè com es veurà més endavant, a l'hora de configurar l'entorn de desenvolupament s'haurà d'especificar la versió a la qual va adreçada l'aplicació i el nivell d'API sobre el qual es treballa. Per tant, una elecció inadequada d'aquests factors pot determinar que l'aplicació utilitzi recursos que poden no estar implementats en altres versions de sistemes operatius i presentar llavors problemes de funcionament.

Un cop escollit el sistema operatiu sobre el qual funcionarà l'aplicació, el següent pas és escollir l'entorn del desenvolupament. D'entorns de programació també existeixen diversos, entre els quals els més utilitzats són: Android Studio i Eclipse. Qualsevol dels dos sistemes permet el desenvolupament d'aplicacions per a dispositius mòbils implementades en llenguatge JAVA, per tant qualsevol dels dos sistemes és totalment compatible amb el disseny a realitzar. S'ha escollit l'Android Studio per les facilitats visuals que incorpora.

La Fig.18 detalla el procés de configuració d'un nou projecte del programa Android Studio, per a dispositius amb versió de sistema operatiu KitKat amb API 19:

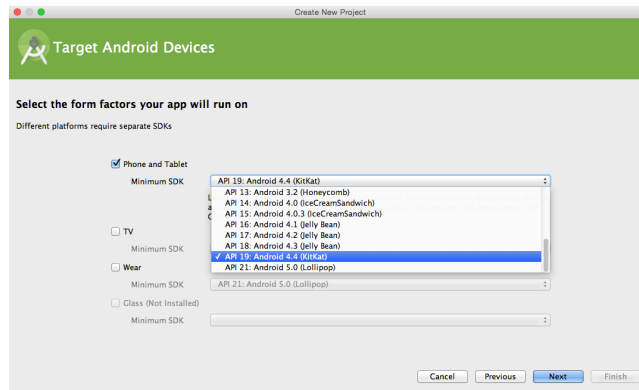


Fig.18 - Pantalla de selecció de l'API del programari Android Studio.

Un cop escollit el sistema operatiu sobre el qual funcionarà l'aplicació, l'entorn de programació i la versió de funcionament, es defineix l'estratègia de desenvolupament del codi. L'estratègia que s'ha dut a terme en la creació de l'aplicació, ha sigut estructurar el funcionament global del sistema en diverses funcionalitats que són les següents:

3.4.1 – Establiment de les funcionalitats generals

El primer pas es establir els principals processos de treball del nostre aplicatiu. Aquests processos han de proporcionar les funcionalitats requerides en un esquema simple d'enteniment. La manera més senzilla és dotar l'aplicatiu d'un menú principal que determini una manera lògica el procés a seguir per l'usuari.

Així doncs, l'aplicatiu ha de constar d'un menú principal que ha permetre a l'usuari les següents funcionalitat:

1. Accedir a un altre menú en el qual l'usuari pugui introduir les dades requerides per l'aplicació.
2. Accedir a un altre menú que proporioni la possibilitat de connectar-se al Kduino i proporioni les funcionalitats requerides sobre el Kduino.
3. Addicionalment ha de permetre a l'usuari accedir a la plana web del projecte.

D'aquesta manera es pot establir d'una manera gràfica el funcionament que hauria de complir cada opció a través d'un diagrama de flux de treball. Per al cas del menú principal, aquest hauria de seguir un funcionament tal i com es mostra en la Fig. 19.

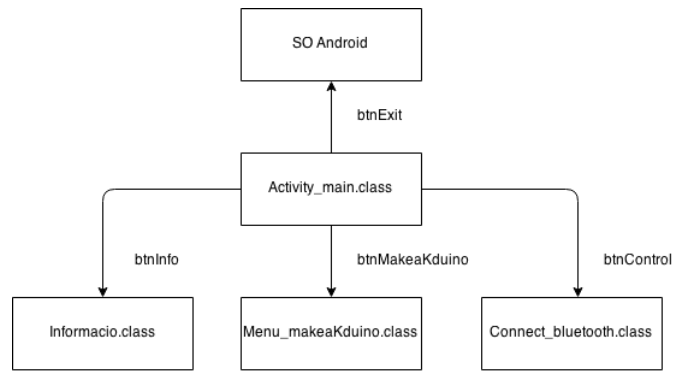


Fig.19 - Diagrama de blocs del funcionament del menú principal.

Una vegada s'accedeix a el menú make a Kduino que ha de permetre a l'usuari introduir les dades requerides, el funcionament de treball hauria de ser el següent:

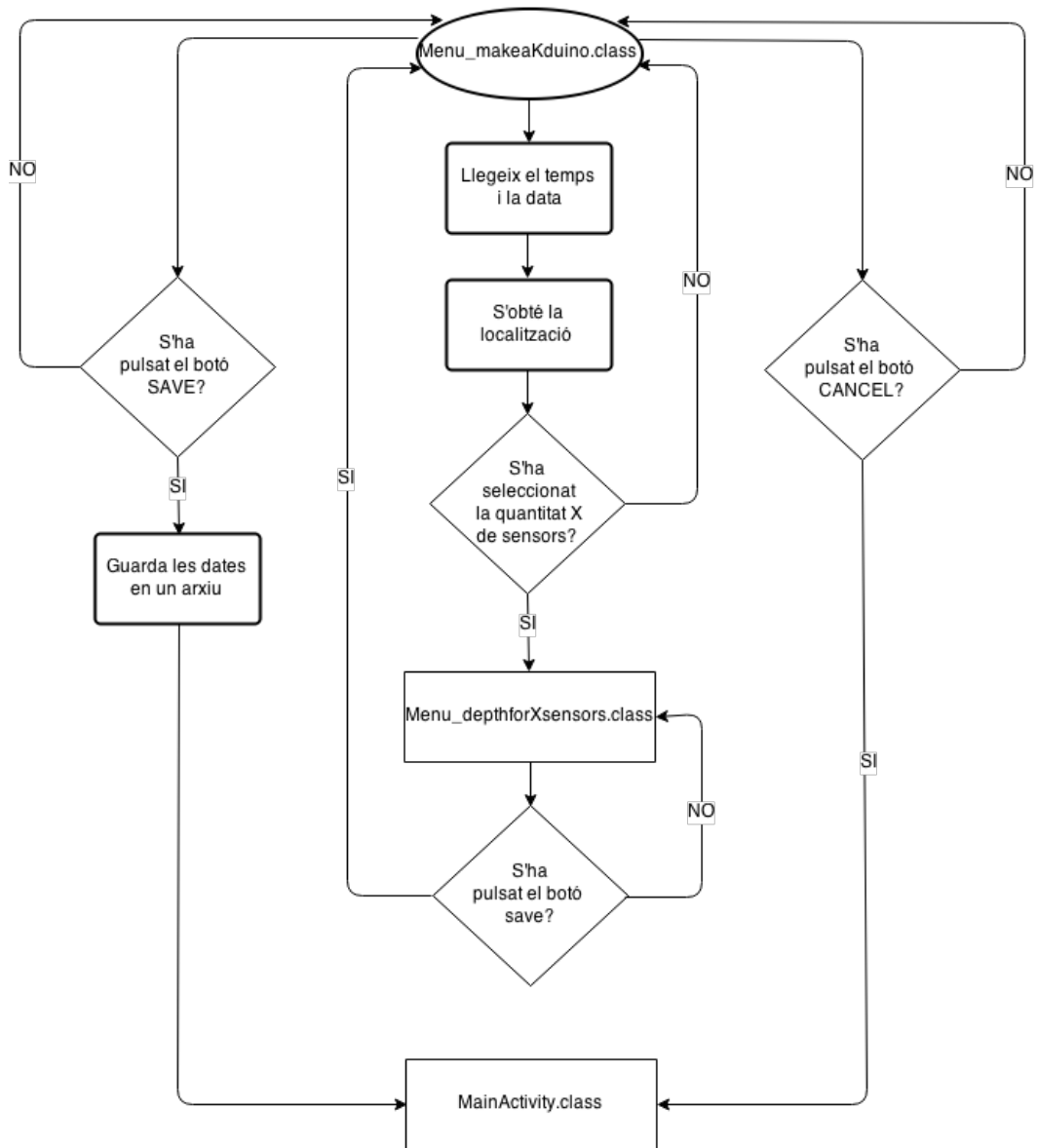


Fig.20 - Diagrama de flux del funcionament del menú makeakduino.

Per a proporcionar connectivitat sobre el Kduino un possible diagrama de treball pot ser el que és mostra a continuació:

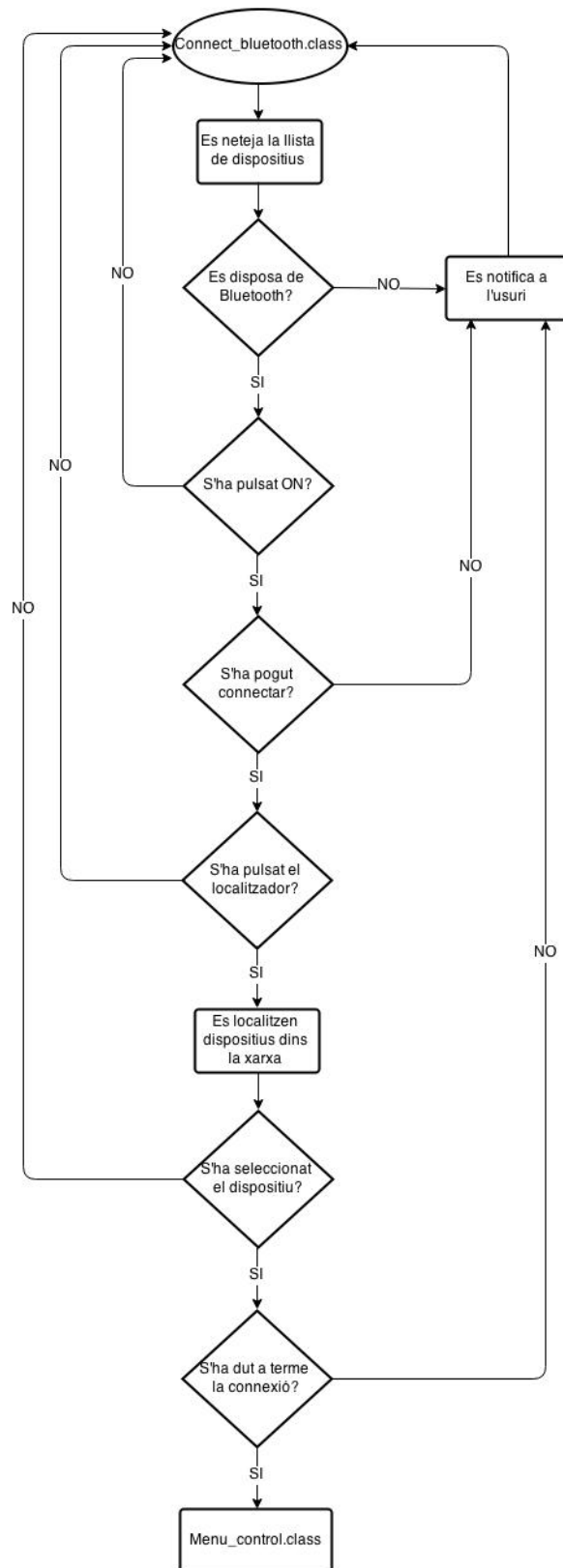


Fig.21 - Diagrama de flux de la connexió Bluetooth.

Una vegada establerts els funcionaments de treball que ha de dur el menú principal i els principals processos de l'aplicació, el següent pas és establir com han de ser els processos de connexió, tant amb el dispositiu Kduino com amb la base de dades.

3.4.2 – Interactivitat amb el dispositiu

Per aquest propòsit, s'estableixen les possibilitats existents com poden ser: Wi-Fi, Bluetooth, Infrarojos, etc. L'Arduino per aquest objectiu presenta un mòdul de transmissió xBee que tradueix un protocol de comunicacions sèrie ASCII a un protocol de comunicacions per radio, a través del estàndar IEEE 802.15.1.

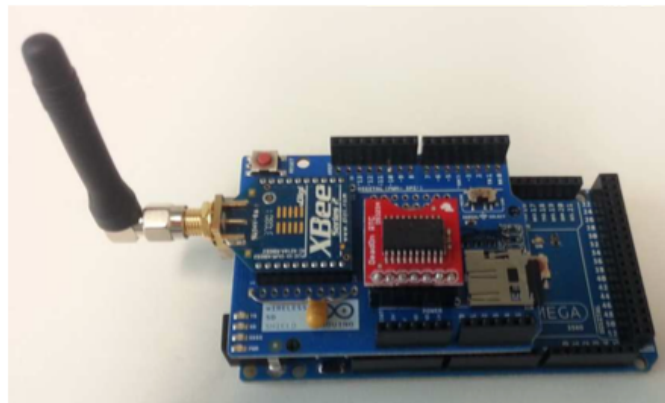


Fig.22 – Interconnectivitat entre dispositius.

Un cop determinat el protocol de comunicació entre ambdós dispositius, el següent pas és establir les funcionalitats que ha de tenir l'aplicació, aquestes venen definides per la programació del Kduino i són:

- Enviar la data i l'hora actual per d'actualitzar aquestes dades en el Kduino:

```
case 'R': // Read time  
  sendTime();
```

Fig.23 – Send Data del Kduino.

- Donar l'ordre de començar a mesurar:

```
case 'S': // Start continuous measurement  
  ContinueMeasurement = true;
```

Fig.24 – Start measurements del Kduino.

- Donar l'ordre de parar les mesures:

```
case 'U': // Stop continuois measurement
ContinueMeasurement = false;
```

Fig.25 – Stop measurements del Kduino.

- Obtenir els resultats de les mesures:

```
case 'D': // Send Data
sendData();
```

Fig.26 – Send Data del Kduino.

- Esborrar les mesures realitzades de la memòria interna del Kduino:

```
case 'Z': // Delete file
deleteFile();
```

Fig.27 – Delete Data del Kduino.

- Calibrar els sensors:

```
case 'C': // Calibration of sensors
CalSensors();
```

Fig.28 – Calibration sensors del Kduino.

- Llegir l'hora i la data de configuració del Kduino:

```
case 'T': // Program time
while (Serial.available() < 12){ delay(5);}
timeProgram();
```

Fig.29 – Programació de l'hora i la data del Kduino.

- Testejar els sensors:

```
case 'Q': // Test of sensors
TestSensors();
```

Fig.30 – Test sensors del Kduino.

3.4.3 – Interactivitat amb la base de dades

Una vegada aconseguida la interactivitat amb el dispositiu i sabent la informació a tractar, el següent pas és establir el mètode de connexió amb la base de dades.

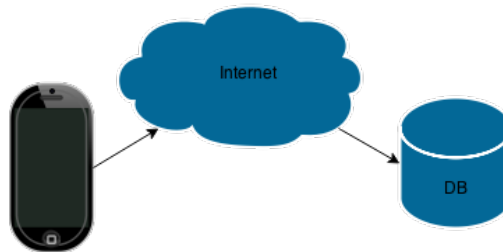


Fig.31 – Interconnexió Smartphone-DataBase.

Els procediments per establir la connexió entre el dispositiu i la base de dades és:

1. Determinar el sistema de gestió de dades (SGBD): Sql Server, MySql, Access, Oracle Sql Server, MySql, Access, Oracle.
2. Dotar de connectivitat a la xarxa a l'aplicatiu.
3. Establir la connectivitat entre l'aplicatiu i el SGBD per interactuar amb la base de dades.

La inserció de les dades a la base de dades, s'ha d'implementar de manera local, això és així, perquè el mode normal de funcionament del programa serà amb la connexió d'una base de dades externa a la qual no es pot accedir. La manera més senzilla de dotar el sistema local d'un sistema de gestió de base de dades, és a través de XAMPP. Aquest paquet totalment gratuït, dota a l'usuari d'un entorn de desenvolupament amb PHP que conté un servidor anomenat Apache i un sistema de gestió de base de dades anomenat MySQL.



Fig.32 – Plana de descàrrega de XAMPP: <https://www.apachefriends.org/es/index.html>

Un cop instal·lat XAMPP, l'aplicació ofereix un panel de control que permet diverses funcions entre les quals per a la implementació de la base de dades les necessàries són: activar i desactivar l'aplicació que es desitgi, accedir als arxius de configuració de l'aplicació corresponent i configurar contrasenyes.

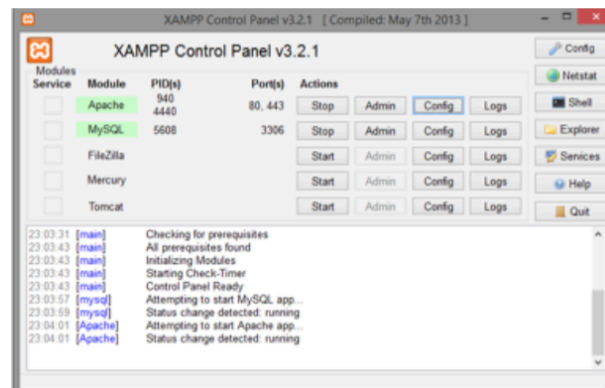


Fig.33 – Panel de control de XAMPP.

Per a interactuar amb la base de dades a través de l'aplicació mòbil, existeixen una gran quantitat d'interfícies de programació d'aplicacions que permeten a aplicacions escrites en diversos llenguatges de programació accedir a les bases de dades MySQL. En aquest cas, per tal d'interactuar amb la base de dades a través de MySQL, s'han de crear una sèrie de scripts en PHP que ha de realitzar les funcions bàsiques sobre la base de dades. Aquests arxius són independents de l'aplicació i al estar ubicats en el servidor són totalment transparents per a l'usuari. D'aquesta manera l'aplicació no té accés directe a la base de dades i queda en mans del gestor quins seran els recursos compartits entre l'usuari i la base de dades.

Per tal de poder accedir a aquests documents des de una aplicació externa, en aquest cas l'aplicació mòbil, es necessita tenir també activat el servidor local. Pel cas de XAMPP el servidor local és l'Apache. Apache és un servidor web que permet allotjar pàgines i aplicacions pròpies de forma que permet donar accés a aquestes, tant localment com des d'Internet. Els documents que Apache permet compartir entre els diferents dispositius han d'estar disposats en la carpeta que Apache té per aquest objectiu. En el cas del sistema operatiu Windows, els arxius han d'estar disposats en la següent adreça: c:\xampp\htdocs.

3.5 - Implementació i prova

Durant el procés de creació s'ha portat a terme un sistema d'implementació i prova constant. Per tal de dur a terme les proves funcionals, el programari Android Studio

disposa d'un emulador que proporciona les funcionalitats d'un dispositiu Smartphone. Tot i que l'ús del simulador ha estat recurrent, per dur a terme les proves d'algunes funcionalitats com poden ser el GPS o la connexió Bluetooth s'han hagut de provar directament sobre un Smartphone.

Per tal d'instal·lar una aplicació en un dispositiu sistema Android, s'han d'activar les opcions de desenvolupament del sistema que per defecte a partir del Android 4.2 (Jelly Bean) venen ocultes. El procés d'activació és fa des de la pantalla que mostra la Fig. 34 sobre la qual s'ha de prémer diverses vegades sobre "Número de compilación" per tal de que el sistema activi el mode.

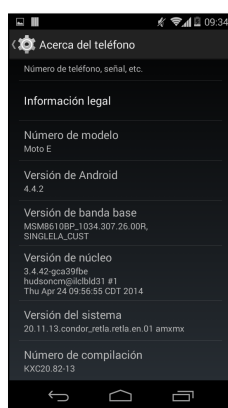


Fig.34 - Pantalla d'activació del mode desenvolupament d'Android.

La primera vegada que es du a terme la connexió entre el dispositiu mòbil i l'ordinador, en el dispositiu apareix una notificació per tal de confirmar la confiança en l'equip Android. Un cop acceptat el permís de depuració a través del port USB el dispositiu mòbil està en disposició de carregar l'aplicació. La càrrega sobre el dispositiu mòbil es du a terme a través del cable de càrrega del dispositiu (cable USB a Micro USB). En el moment de carregar l'aplicació sobre el dispositiu en l'Android Studio apareix la següent pantalla (Fig. 35), en la qual s'escull sobre quin dispositiu es vol fer funcionar l'aplicació, en aquest cas el dispositiu Samsung SM-G3815, o a través de l'emulador que el mateix programa proporciona.

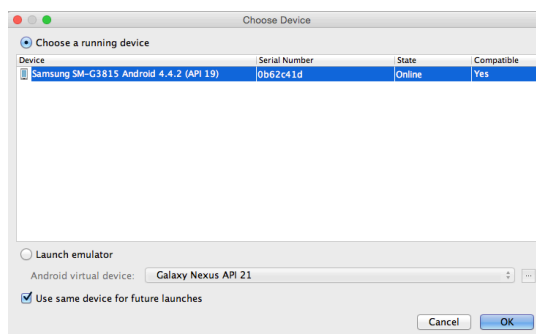


Fig.35 - Pantalla d'activació de la implementació en el dispositiu mòbil.

4. Implementació

4.1 – Estructura dels arxius

El projecte consta de les classes que es mostren en la Fig.36. Cada arxiu té el nom que correspon a les seves funcionalitats, per tant és de fàcil interpretació per a tercers.

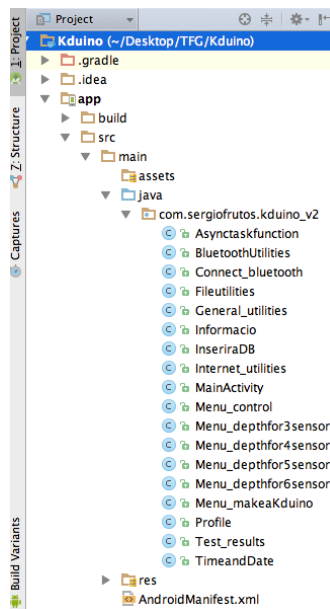


Fig.36 – Estructura general dels arxius del projecte.

A continuació es detallen breument les funcionalitats de cada arxiu:

- Asynctaskfunction: en aquest arxiu es troben les funcionalitats que s'han de realitzar asíncronament, d'aquesta manera es pot realitzar tasques asíncrones tenint accés al procés principal.
- BluetoothUtilities: aquest arxiu recull totes les funcionalitats que s'han de dur a terme sobre la connexió com són la lectura i l'enviament de les dades.
- Connect_bluetooth: recull els mètodes per establir la connexió Bluetooth, la búsqueda dels dispositius i els mètodes d'activació i desactivació del Bluetooth del dispositiu mòbil.

- Fileutilities: es troben els mètodes encarregats de gestionar les funcionalitats sobre els arxius “.txt”, entre els quals es troben la creació dels arxius de text, la gravació i la lectura de les dades sobre els mateixos.
- General_utilities: engloba tots els mètodes que comparteixen tots el arxius de l’aplicació, com poden ser els diàlegs amb l’usuari i els diàlegs d’error.
- Informació: és l’arxiu relacionat amb el layout d’informació (veure Fig.39 – Captura 7), recull els mètodes de retorns al layout principal i la connexió a la plana web del projecte Citclops.
- InserirDB: estableix el mètode de crida a la connexió a la base de dades per tal d’enviar la informació a inserir, que rep com a paràmetres quan es creada la classe.
- Internet_utilities: es troba el mètode per comprovar si el dispositiu té connexió a internet. No s’utilitza ja què, la comprovació es realitza directament a l’arxiu MainActivity, però es deixa implementat per a possibles futures necessitats.
- MainActivity: estableix la relació entre els mètodes i el layout del menú principal (veure Fig.39 – Captura 1). Aquí es troben els mètodes de comprovació de la connexió a internet del dispositiu, de la existència d’arxius de text amb dades per inserir a la base de dades i els mètodes d’accés a les funcions seleccionades per l’usuari a través dels botons del menú.
- Menu_control: proporciona els mètodes per donar funcionalitats que es poden fer sobre la boia i que es troben en el menú control, fent les crides corresponents als mètodes necessaris per a poder dur a terme aquestes funcionalitats.
- Menu_depthforXsensors: (on X determina el nombre de sensors i per tant, la pantalla corresponent) determina les funcionalitats que es presenten a la pantalla corresponent, com són: proporcionar a l’usuari un teclat per a introduir les dades, la lectura de les mateixes i el retorn d’aquestes dades a la classe make a kduino per tal que aquest en gravar el profile insereixi aquestes dades conjuntament amb les altres dades requerides.

- Menu_makeaKduino: proporciona els mètodes que proporcionen a l'usuari la possibilitat d'inserir les dades requerides pel sistema i que han de ser gravades als arxius de text corresponents. A més conté els mètodes d'obtenció de les dades GPS i la crida als mètodes d'obtenció de la data i l'hora del sistema, per a mostrar-los per pantalla.
- Profile: és un objecte que conté totes les dades inserides en el menú make a kduino i ens proporciona la facilitat de poder accedir a les mateixes des de qualsevol mètode de l'aplicació.
- Test_results: conté els mètodes relacionats amb el layout que presenta els resultats del test sensors (veure Fig.39 – Captura 6), mostra els resultats de testejar els sensors i retorna al menú control quan l'usuari vulgui.
- TimeandDate: es troben els mètodes de lectura de la data i l'hora del sistema per retornar-los quan se'ls crida.

4.2 – Estructura dels layouts

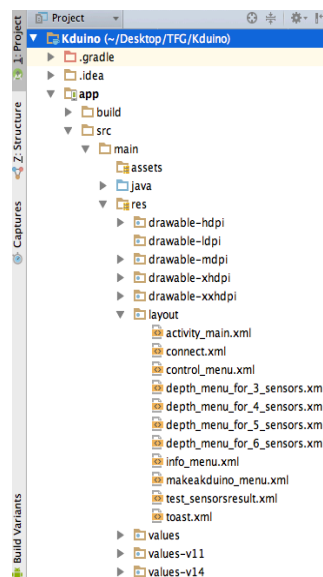


Fig.37 – Estructura general de les pantalles del projecte.

L'estructura de les pantalles que formen l'aplicació es mostra la Fig.37, les quals són analitzades en els posteriors punts del present treball.

4.3 – Arxius complementaris

En la Fig.38 es mostren els arxius complementaris del projecte.

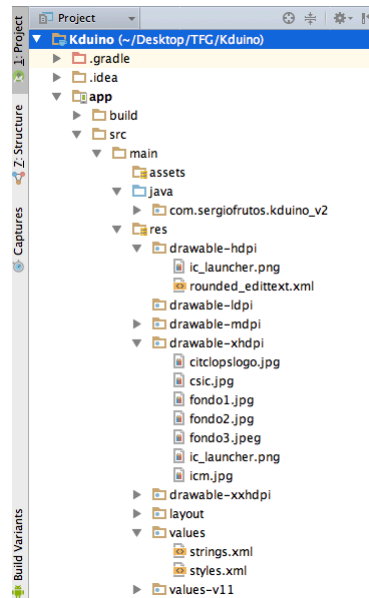


Fig.38 – Arxius complementaris.

Tot arxiu complementari del projecte, com poden ser les imatges, es troben disposats en la carpeta res del projecte. En aquest cas, trobem els arxius d'imatge corresponents als fons de pantalla i així com les imatges que s'utilitzen en l'aplicació. Dins la carpeta drawable-hdpi es troba l'arxiu rounded_edittext.xml, en aquest document s'ha definit les característiques per a un tipus de lletra, per tal de personalitzar el projecte amb una font pròpia.

La carpeta values conté els arxius strings.xml i styles.xml. L'arxiu string.xml conté tots els strings definits pel programador i son textos que poden ser referenciats fàcilment a través d'un nom definit pel programador, d'aquesta manera s'aconsegueix referenciar un string amb facilitat i pot ser utilitzat diverses vegades.

L'arxiu styles.xml és un arxiu que conté el projecte per defecte i conté els diversos estils que utilitza la pròpia aplicació per tal de mostrar els layouts.

4.4 – Estructura general de l'aplicació

La figura següent mostra l'estructura general del funcionament del programa a través de captures:

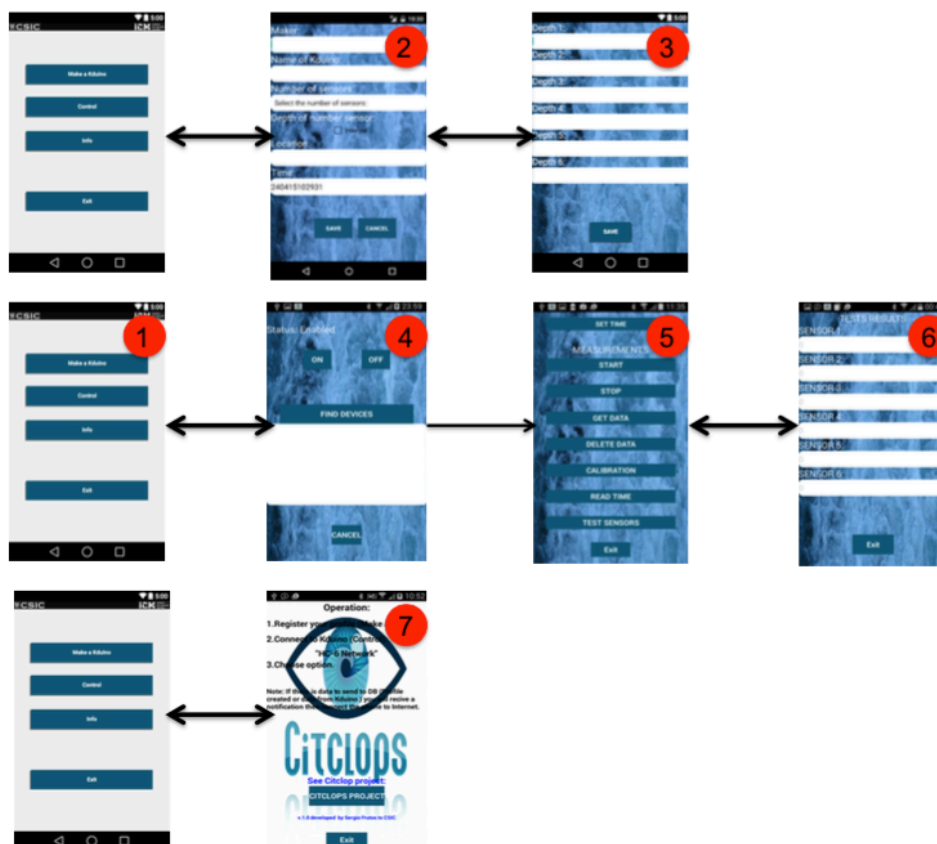


Fig.39 – Diagrama de navegació de l'aplicació.

4.5 – Menú principal

4.5.1 - Descripció

La captura 1 de la Fig.39 presenta la pantalla principal del programa. En aquesta pantalla es disposa de quatre botons que es detallen a continuació:

1. **MAKE A KDUINO:** mostra el menú Make a Kduino (Fig.39 – Captura 2). Menú que permet enregistrar les dades relacionades amb la presa de la mesura, com el nom de la boia, el nom de l'usuari, les dades GPS, la data i l'hora de la mesura i el número de sensors, així com les profunditats de cadascun.

2. CONTROL: mostra les opcions de connexió, a través d'una xarxa Bluetooth, amb la boia (Fig.39 – Captura 5), prèviament establint la connexió amb la boia (Fig.39 – Captura 4).
3. INFO: mostra per pantalla una sèrie de dades relacionades amb: drets d'autor, realitzadors del projecte, funcionament i accés a la plana web del projecte (Fig.39 – Captura 6).
4. EXIT: acaba amb l'execució del programa (Fig.39 – Captura 1).

4.5.2 - Funcionalitat

El funcionament del menú activity main, segueix l'estructura que s'ha dissenyat prèviament en l'etapa de disseny i que és el que mostra la Fig.19, i que respon a les funcionalitats descrites anteriorment.

A més de les funcionalitats que presenta el diagrama de blocs, el menú principal permet la inserció de les últimes dades recollides per l'usuari a la base de dades (DB). Per això, sempre que s'inicialitza la pantalla principal el programa mira si existeix connexió a Internet. Si existeix connexió a Internet, comprova l'existència de l'arxiu de dades anomenat "dades_usuari.txt", que es crearà en el moment en el que l'usuari introdueixi les dades requerides en el menú makeaKduino, i si existeix també algun dels arxius de mesures "dades_boia.txt" o "cal_boia.txt". L'arxiu "dades_boia.txt" conté les dades rebudes de la boia amb les mesures captades pels sensors. L'arxiu "cal_boia.txt" conté les calibracions dels sensors rebudes de la boia. Si es compleixen aquestes tres condicions, el programa llegeix l'arxiu, recull les dades i intenta accedir a la base de dades. Si es pot connectar a la base de dades, envia la informació inserint en aquesta una nova línia de dades. Si la inserció de la informació s'ha dut a terme correctament, el sistema borra l'arxiu de dades d'usuari i l'arxiu/us de dades i continua el seu funcionament. En el cas que no existeixi connexió a Internet o no existeixi l'arxiu, el programa no comença el procés d'inserció de dades a la base de dades i funciona normalment.

Si l'usuari a inserit les dades i es crea l'arxiu "dades_usuari.txt" o qualsevol dels dos arxius de dades "dades_boia.txt" o "cal_boia.txt", i no hi ha connexió a internet per enviar les dades a la base de dades, l'aplicació notifica a l'usuari mitjançant una notificació en la barra de notificacions que existeix informació per enviar a la base de

dades, però com no hi ha connexió a internet no es pot dur a terme la càrrega (Fig.40). D'aquesta manera l'usuari ha de connectar-se a internet per tal que l'aplicació pugui enviar les dades, moment en el qual desapareixerà la notificació.

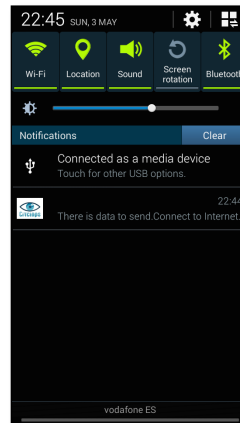


Fig.40 - Notificació d'existència de dades per enviar a la base de dades.

En el cas que hi hagi algun problema en la connexió amb la base de dades, el sistema borra l'arxiu de dades igualment, però sempre existeix una còpia de l'arxiu que es deixa en la memòria del telèfon i al qual només es podrà recuperar de manera manual. En aquest arxiu, anomenat "dades_backup.txt", s'aniran inserint les dades de manera contínua, per disposar sempre d'una còpia de seguretat.

A la Fig.41 es detalla en un diagrama de flux el funcionament de la connexió a la base de dades.

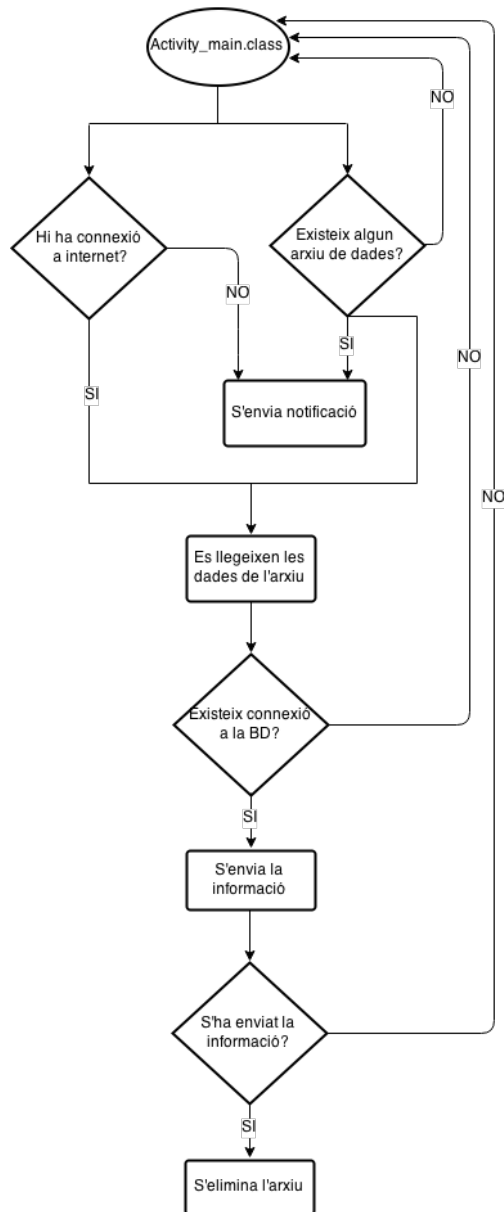


Fig.41 - Diagrama de flux del funcionament de la inserció de dades a la DB.

4.6 - Menú make a kduino

4.6.1 - Descripció

El menú makeakduino, captura 2 de la Fig.39, permet a l'usuari poder introduir les dades necessàries associades a les mesures que es duran a terme. Per aquest propòsit el menú make a kduino disposa dels següents camps:

- MARKER: camp on l'usuari insereix el seu nom per tal d'identificar l'usuari d'una certa boia/es.

- NAME OF KDUINO: camp on l'usuari ha d'inserir el nom de la boia, per identificar la boia. Un usuari pot tenir més d'una boia, per tant s'identificarà cada boia amb un nom diferent.
- NUMBER OF SENSORS: al prémer aquest camp es desplega un llista amb diferents opcions: [3,4,5,6] on l'usuari ha d'escollir quina és la quantitat de sensors que disposa la boia. Existeixen diferents models de boia amb diferents sensors, concretament poden disposar de tres, quatre, cinc o sis sensors segons la profunditat del lloc de la mesura.
Un cop realitzada la selecció, apareix una pantalla que permet la inserció de la profunditat a la qual es troba cada sensor. En la captura 3 de la figura 39 es mostra la pantalla per a l'opció de sis sensors.
- DEPTH OF NUMBER SENSORS: és un checkbox que mostra a l'usuari si la introducció de les diferents profunditats s'ha dut a terme correctament.
- LOCATION: és un camp que mostra per pantalla la localització del dispositiu de forma automàtica, en el moment de fer la captura.
- TIME: en aquest camp es mostra automàticament la data i l'hora de l'enregistrament de les dades.

4.6.2 - Funcionalitat

Seguint el procediment de treball descrit en la Fig.20 el funcionament de del menú make a Kduino és el següent.

Al entrar en aquest menú el programa llegeix automàticament en primer lloc la data i l'hora del sistema per tal de mostrar-la en el camp Time. Tot seguit crida automàticament a un procés fill per tal d'obtenir la localització del dispositiu a través de la xarxa telefònica. Un cop s'obté la latitud i la longitud, el sistema les mostra en el camp Location. En el cas de que el sistema no disposi de xarxa, el sistema ho notificarà a l'usuari (Fig. 42):

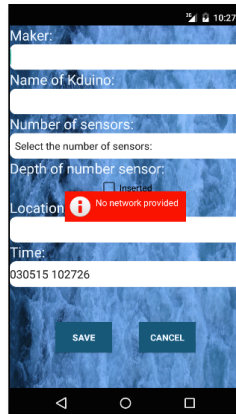


Fig.42 - Notificació de no existència de xarxa per obtenir la localització.

Un cop el sistema ha obtingut la data el temps i la localització romandrà a la espera de que l'usuari insereixi el seu nom en el camp Marker, el nom de la boia en el camp Name of Kduino i seleccioni en la llista desplegable la quantitat de sensors dels que disposa la boia. Una vegada l'usuari seleccioni la quantitat de sensors, el sistema llança una pantalla amb camps en blanc, per tal de que l'usuari insereixi la profunditat de cada sensor (Fig.39 – Captura 3). Quan és prem el botó save del menú depthofXsensors, el programa guarda els paràmetres inserits per l'usuari i retorna al menú make a kduino per tal de seguir amb la introducció de la resta de dades. Al retornar de qualsevol pantalla d'inserció de profunditats, el checkbox Depth of number sensors queda marcat de manera que l'usuari sap que s'han introduït les dades correctament. Un cop totes les dades han estat introduïdes, el sistema espera a que s'hagi polsat el botó save, moment en el qual guardarà totes les dades a l'arxiu "dades_usuari.txt" i "dades_backup.txt".

En el cas que l'usuari premi el botó save sense omplir totes les dades, el programa mostrarà a l'usuari un missatge indicant que ompli totes les dades, tal i com es mostra a continuació:

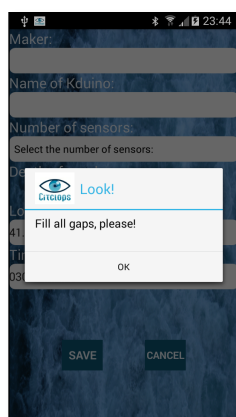


Fig.43 - Missatge de diàleg a l'usuari.

4.7 - Menú de connexió al kduino

4.7.1 - Descripció

Quan l'usuari des de'l menú principal accedeix al menú control, prèviament ha de passar per la pantalla de connexió al Kduino captura 4 de la Fig.39. Des d'aquesta pantalla l'usuari pot connectar-se mitjançant una xarxa Bluetooth amb el Kduino. Per això, el sistema disposa de les següents opcions:

- ON: activa el Bluetooth del telèfon en cas que aquest estigui desactivat.
- OFF: desactiva el Bluetooth del telèfon en cas que aquest estigui activat.
- FIND DEVICES: botó d'activació de la búsqueda de dispositius amb connexió Bluetooth.
- CANCEL: cancel·la la búsqueda en cas que aquesta estigui activa i retorna al menú principal.

4.7.2 - Funcionalitat

A més dels botons principals el sistema proporciona a l'usuari informació de l'estat del Bluetooth, a través del "Status" (Fig. 44):

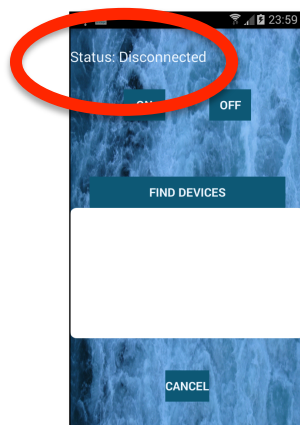


Fig.44 - Estat del Bluetooth.

En el cas que el dispositiu no disposi de Bluetooth el sistema alertarà a l'usuari a través d'un missatge com el que es mostra a continuació (Fig. 45):

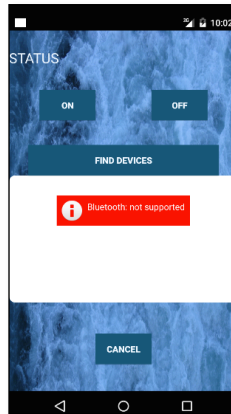


Fig.45 - Avis de no existència de Bluetooth.

Quan el sistema disposi de Bluetooth i l'usuari premi el botó on, el sistema activarà el Bluetooth del sistema (Fig. 46):

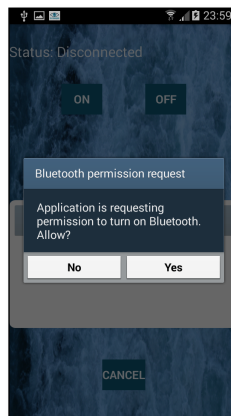


Fig.46 - Activació del Bluetooth.

D'aquesta manera el Bluetooth serà activat, per posteriorment procedir a cercar els dispositius que es troben actius en un radi de 10 metres, els quals es mostren en un llistat sobre el qual l'usuari haurà de seleccionar el Kduino i automàticament el sistema procedirà a connectar-se (Fig. 47).

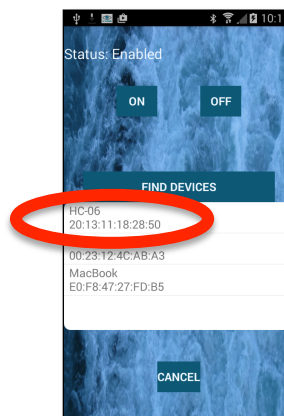


Fig.47 - Llista de dispositius cercats.

En el cas que el sistema no pugui connectar-se amb el dispositiu el sistema mostrarà una advertència a l'usuari per indicar que no s'ha pogut dur a terme la connexió i que ha de provar una altra vegada (Fig. 48):

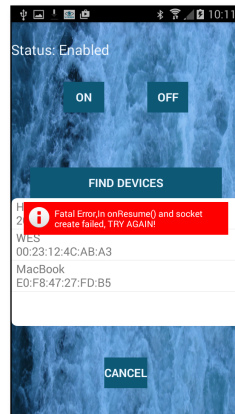


Fig.48 - Avís d'error en la connexió.

4.8 - Menú control

4.8.1 - Descripció

El menú control, captura 5 de la figura 39, permet a l'usuari poder interactuar amb el Kduino. Aquest menú disposa de totes les accions que es poden dur a terme sobre el dispositiu, aquestes accions són:

- SET TIME: envia al Kduino la data i l'hora actuals.
- START: envia una ordre al Kduino perquè aquest comenci a mesurar.
- STOP: envia una ordre al Kduino perquè aquest deixi de prendre mesures.
- GET DATA: rep les dades mesurades pels sensors provinents del Kduino.
- DELETE DATA: envia una ordre al Kduino, per tal que aquest esborri les dades contingudes a la memòria interna.
- CALIBRATION: es reben les dades de les calibracions dels sensors.

- READ TIME: es reben la data i l'hora de la configuració del Kduino.
- TEST SENSORS: es reben les dades resultants de testejar el sensors i es mostren per pantalla.

4.8.2 - Funcionalitat

Cadascuna de les opcions correspon a una funcionalitat concreta que segueix un protocol determinat. Per tant per al disseny de cada funció s'ha hagut de tenir en compte la programació del Kduino. El funcionament de cadascuna es pot resumir amb un protocol de treball, definits en cada cas segons les figures següents:

4.8.2.1 - SEAT TIME:

Quan es prem el botó seat time, el procés que es duu a terme és el següent:

L'aplicació envia una T, de manera que quan el Kduino rep aquest caràcter sap que ha d'esperar una cadena de caràcters enviats en sèrie, que correspon a la data i l'hora del sistema segons la seqüència de caràcters següent:

DDMMYYhhmmss

- on
- DD: és el dia actual
 - MM: és el mes actual
 - YY: és l'any actual
 - hh: és l'hora actual del sistema
 - mm: són els minuts actuals del sistema
 - ss: són els segons actuals del sistema

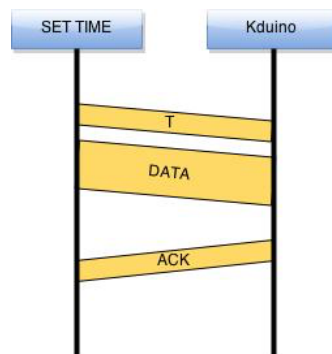


Fig.49 - Protocol SEAT TIME.

Quan el Kduino rebí l'últim caràcter aquest automàticament envia al transmissor, en aquest cas el dispositiu mòbil, un ACK (acknowledgement) si la cadena s'ha rebut correctament. Quan es rep l'ACK el sistema entén que el procés s'ha dut a terme correctament i mostra per pantalla un diàleg a l'usuari indicant que el procés s'ha dut a terme correctament (Fig. 50). En el cas que no es rebí l'ACK el sistema indica a l'usuari que el procés no s'ha pogut dur a terme i que torni a intentar-ho.

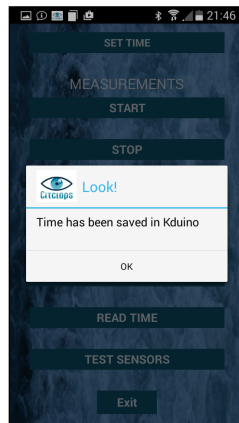


Fig.50 - Diàleg de confirmació del SEAT TIME.

4.8.2.2 - START:

El funcionament del botó START és el que presenta la Fig. 51.

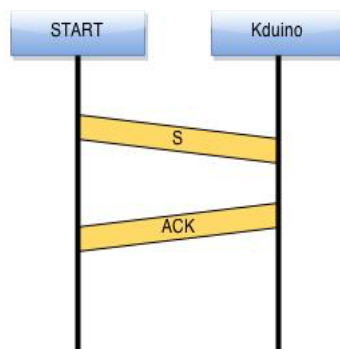


Fig.51 - Protocol START.

Quan es prem el botó START el sistema envia una S. D'aquesta manera s'indica al Kduino que ha de començar a mesurar. Aquest procés pot trigar fins a un minut, per la qual cosa el sistema ha d'esperar fins a rebre el ACK corresponent, que indica que el Kduino ha finalitzat el procés. Quan es rep l'ACK el sistema indica a l'usuari

que el procés s'ha dut a terme correctament (Fig. 52), en cas contrari el sistema també ho posa en coneixement de l'usuari.

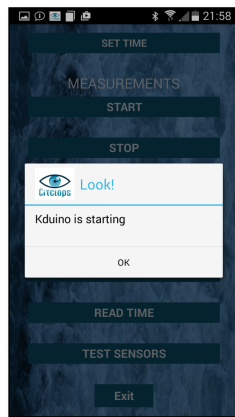


Fig.52 - Diàleg de confirmació del START.

4.8.2.3.- STOP:

Al pulsar el botó STOP el sistema envia una U al Kduino. Aquest en rebre el caràcter, entén que ha de parar les mesures en cas que aquestes s'estiguin produint, moment en el qual envia un ACK al nostre sistema (Fig. 53). En aquest moment és sistema indica a l'usuari que el Kduino ha parat de mesurar (Fig. 54).

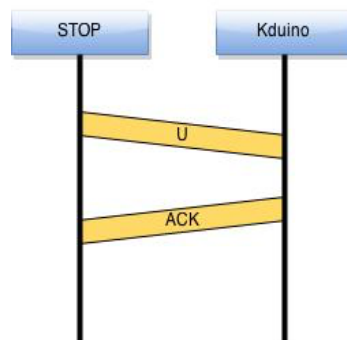


Fig.53 - Protocol STOP.

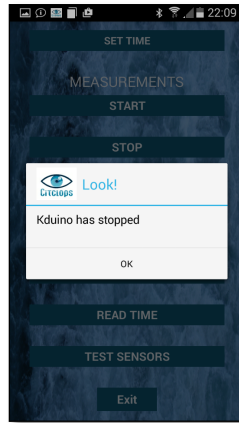


Fig.54 - Diàleg de confirmació del STOP.

4.8.2.4 - GET DATA:

El botó GET DATA envia una D al Kduino, d'aquesta manera el Kduino comença a enviar la informació recollida pels sensors. El programa anirà rebent les dades caràcter per caràcter fins que arribi l'ACK que significarà que l'enviament de dades ha acabat (Fig. 55). Les dades s'aniran guardant en un buffer sobre el qual és recuperaran les dades per gravar-les en dos txt. En el txt "dades_backup", les dades s'insereixen de manera contínua juntament amb la calibració, si existeixen dades de les calibracions. En el "dades_boia.txt" s'insereixen les dades per tal de recuperar-les posteriorment per a introduir-les en la base de dades.

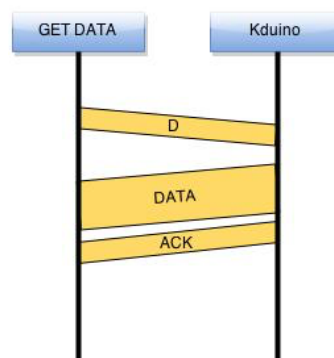


Fig.55 - Protocol GET DATA.

Un cop el sistema rebí l'ACK de finalització de dades, aquest avisarà a l'usuari amb el diàleg següent:

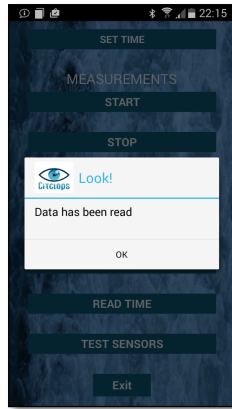


Fig.56 - Diàleg de confirmació del GET DATA.

4.8.2.5.- DELETE DATA:

El funcionament del DELETE DATA és el que mostra la Fig. 57. Com es pot apreciar, el sistema envia una Z per indicar al Kduino que esborri les dades que disposa a la memòria. Un cop es procedeixi a l'esborrament, el Kduino enviarà el corresponent ACK per indicar que s'han esborrat correctament. Moment en el qual s'avisarà a l'usuari amb el missatge corresponent (Fig. 58).

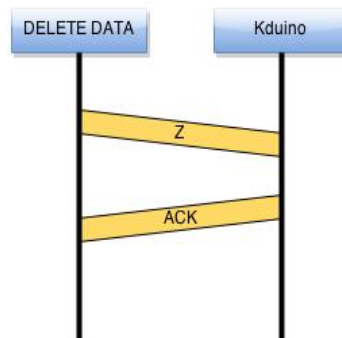


Fig.57 - Protocol DELETE DATA.

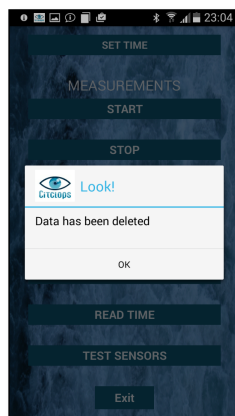


Fig.58 - Diàleg de confirmació del DELETE DATA.

4.8.2.6.- CALIBRATION:

Amb el botó CALIBRATION s'envia una C al Kduino per tal que aquest enviï al sistema les calibracions dels sensors (Fig. 59). Es rebrà una cadena de caràcters els quals el sistema grava també en dos txt. Un dels arxius és "dades_backup.txt", en el qual s'insereixen a continuació de les dades del GET DATA, d'aquesta manera ambdues dades romandran sempre en la memòria, per a recuperar-les en cas de necessitat. I un altre arxiu anomenat "cal_boia.txt", sobre el qual es treballa, en el que s'accedirà per a recuperar les dades i enviar-les a la base de dades.

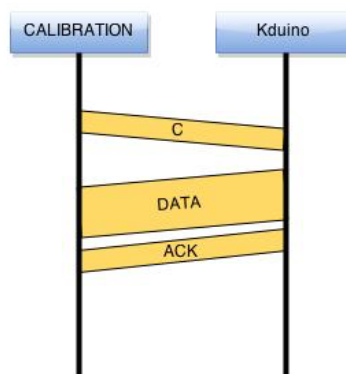


Fig.59 - Protocol CALIBRATION.

Quan es rep l'ACK el sistema entén que l'enviament de dades s'ha acabat i mostra a l'usuari que les dades s'han rebut correctament (Fig. 60), a més es pot procedir a gravar els arxius (Fig. 61).

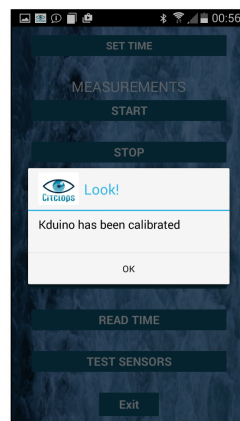


Fig.60 - Diàleg de confirmació de calibració.

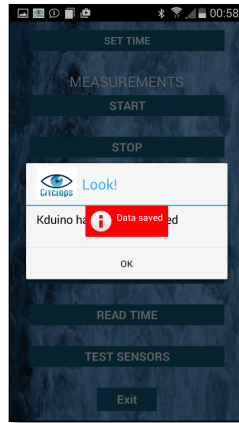


Fig.61 - Confirmació de gravació de les dades.

4.8.2.7 - READ TIME:

El botó READ TIME envia una R al Kduino per tal que aquest enviï les dades de la data i l'hora del sistema Kduino (Fig. 62), les dades rebudes corresponen a una cadena de caràcters sobre la qual no ens demanen que es facin servir. Per aquest motiu, només es mostra a l'usuari el missatge corresponent juntament amb les dades rebudes quan es rebí l'ACK en el cas que s'hagin rebut les dades correctament (Fig. 63).

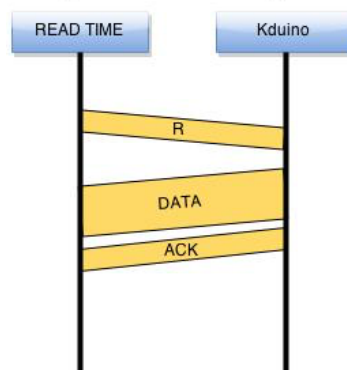


Fig.62 - Protocol READ TIME.

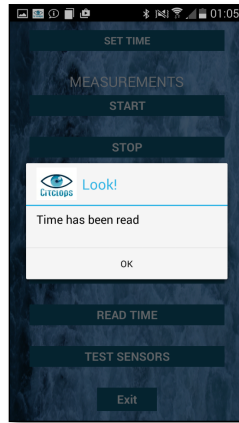


Fig.63 - Confirmació de la lectura de la data i l'hora.

4.8.2.8.- TEST SENSORS:

A la Fig. 64 es mostra el funcionament del TEST SENSORS. El sistema envia una Q al Kduino i aquest envia a continuació una sèrie de dades que indiquen el resultat de testejar els sensors. Les dades rebudes un cop rebut l'ACK final, es mostren en una pantalla diferent, ja que l'usuari ha visualitzar els resultats per tal de prendre nota. D'aquesta manera l'usuari decideix quan deixa de visualitzar els resultats (Fig. 65).

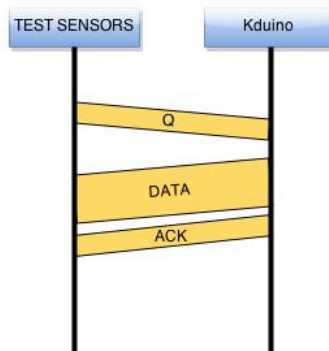


Fig.64 - Protocol TEST SENSORS.

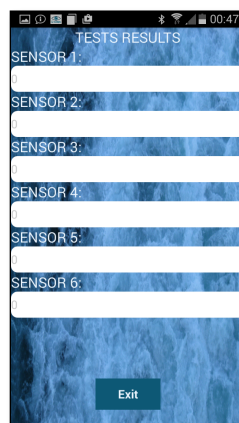


Fig.65 - Pantalla Test Results.

4.9 - La base de dades

Tenint en compte el sistema adoptat en el procés de disseny, l'estructura de funcionament del sistema d'interacció amb la base de dades és el següent:

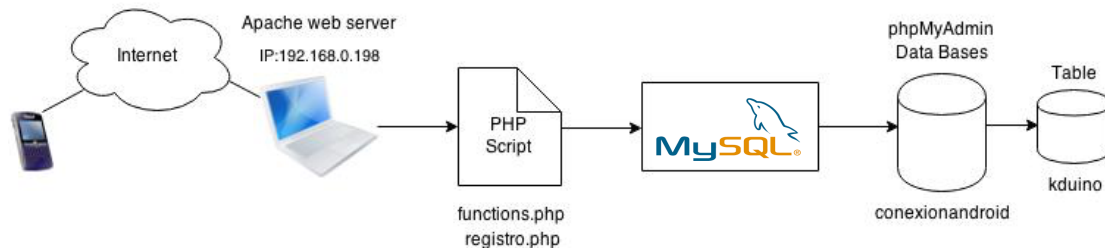


Fig.66 - Creació de la base de dades en phpMyAdmin.

El primer pas per crear la base de dades y poder accedir a phpMyAdmin és activar el mòdul corresponent a MySQL. Un cop activat el mòdul, es configura la contrasenya d'accés polsant Admin des de'l panel de control principal. Posteriorment polsant "Chequeo de seguridad" accedim a la plana de la Fig. 67.



Fig.67 – Plana "Chequeo de seguridad".

En aquesta plana es pot veure la configuració de seguretat de totes les aplicacions de XAMPP, en aquest cas tant MySQL com PhpMyAdmin disposen de password d'accés. Si es vol canviar la contrasenya només cal accedir al link que es troba en la mateixa plana i es podrà canviar segons les necessitats de seguretat. Un cop configurat el password es pot accedir ja a la base de dades de PhpMyAdmin, al qual s'accedeix a través del botó Admin de l'aplicació MySQL. Per poder accedir-hi em d'insertar la contrasenya que hem insertat prèviament al pas anterior.

Un cop s'accedeix a PhpMyAdmin el següent pas és crear una base de dades nova, amb el nom que es vulgui, en aquest cas el nom és "conexionandroid". Dins

d'aquesta base de dades s'ha de crear la taula que conté les dades que s'han d'inserir. En aquest cas, el nostre aplicatiu requereix els següents camps:

#	Nombre	Tipo	Cotejamiento	Atributos	Nulo	Predeterminado	Extra	Acción
1	id	int(2)			No	Ninguna	AUTO_INCREMENT	Cambiar Eliminar Primaria Único Índice Más
2	name	varchar(25)	utf8_spanish2_ci		No	Ninguna		Cambiar Eliminar Primaria Único Índice Más
3	numSensors	int(2)			No	Ninguna		Cambiar Eliminar Primaria Único Índice Más
4	depth1	double			Si	NULL		Cambiar Eliminar Primaria Único Índice Más
5	depth2	double			Si	NULL		Cambiar Eliminar Primaria Único Índice Más
6	depth3	double			Si	NULL		Cambiar Eliminar Primaria Único Índice Más
7	depth4	double			Si	NULL		Cambiar Eliminar Primaria Único Índice Más
8	depth5	double			Si	NULL		Cambiar Eliminar Primaria Único Índice Más
9	depth6	double			Si	NULL		Cambiar Eliminar Primaria Único Índice Más
10	lat	float			Si	NULL		Cambiar Eliminar Primaria Único Índice Más
11	long	float			Si	NULL		Cambiar Eliminar Primaria Único Índice Más
12	marker	varchar(25)	utf8_spanish2_ci		No	Ninguna		Cambiar Eliminar Primaria Único Índice Más
13	data	varchar(50)	utf8_spanish2_ci		Si	NULL		Cambiar Eliminar Primaria Único Índice Más
14	calibrations	varchar(50)	utf8_spanish2_ci		Si	NULL		Cambiar Eliminar Primaria Único Índice Más
15	date	varchar(20)	utf8_spanish2_ci		No	Ninguna		Cambiar Eliminar Primaria Único Índice Más
16	time	varchar(20)	utf8_spanish2_ci		No	Ninguna		Cambiar Eliminar Primaria Único Índice Más

Fig.68 – Camps de la taula kduino.

La variable id serà una variable int que s'ha d'incrementar automàticament en cada inserció d'una nova línia. Està definida com la clau primària (PRIMARY KEY) per tant, aquest valor no pot tenir valors nuls ni repetits. La resta de dades corresponen al tipus de dades que hem de guardar. En les columnes depth1...depth6 es permet tenir valor nuls, perquè si existeixen menys de sis boies la resta de valors fins arribar a sis serà nul. Tanmateix, definim el camp data i calibrations com a null per defecte perquè es pot donar el cas que només s'insereixi una de les dues dades.

La connexió a la base de dades i la inserció de dades es duu a terme a través dels scripts que s'adjunten als annexos, als quals el nostre dispositiu podrà accedir a través del servidor Apache.

L'arxiu functions.php (veure Annex), és l'encarregat de crear la connexió amb la base de dades sota la comanda:

```
$mysqli = new mysqli("localhost","root","contrasenya","conexionandroid");  
$mysqli->query("SET NAMES 'utf8'");
```

En aquest cas concret la connexió es realitza a nivell local, sobre la base de dades conexionandroid, amb nom d'usuari root i el password corresponent.

L'arxiu registro.php (veure Annex), és l'encarregat de inserir les dades a la taula a través de la connexió realitzada prèviament per l'arxiu functions.php. Per tant, es defineixen les variables que han de ser inserides i es relacionen amb les variables que ha de rebre de l'aplicació, amb la comanda:

```
$name=$_GET['name'];
```

Un cop s'han capturat les dades que es reben, s'insereixen a la taula kduino amb la comanda:

```
ejecutarSQLCommand("INSERT INTO  
'kduino'(`name`,`numSensors`,`depth1`,`depth2`,`depth3`,`depth4`,`depth5`,`depth6`  
,`lat`,`long`,`marker`,`data`,`dates`,`time`) values  
('$name','$numSensors','$depth1','$depth2','$depth3','$depth4','$depth5','$depth6','$l  
at','$long','$marker','$data','$dates','$time')
```

Amb la comanda ON DUPLICATE KEY UPDATE es defineix com ha d'actuar el sistema en el cas de trobar una línia duplicada a l'insertar les dades, en aquest cas actualitzarà les dades de la línia prèviament creada.

L'aplicació mitjançant l'adreça IP del servidor es connecta a través de la xarxa d'Internet al servidor web local que en aquest cas correspon l'Apache de l'ordinador de l'usuari.

Així doncs, només cal obtenir l'adreça IP a la qual s'ha d'adreçar l'aplicació per poder ser escoltat per Apache, en el port 80 en aquest cas, i poder accedir als arxius PHP, per obtenir l'adreça, realitzem un ipconfig del sistema per saber l'adreça:

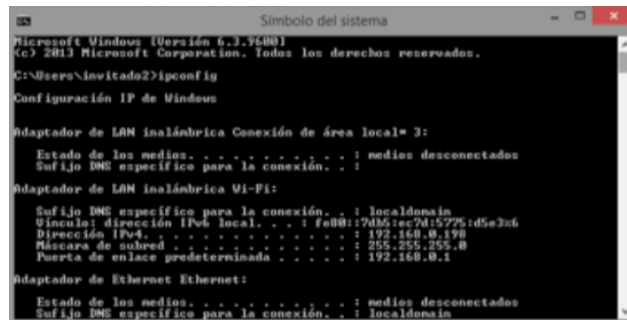


Fig.69 – Obtenció de la ip del servidor.

Finalment, si es procedeix a provar el funcionament, s'obté el resultat següent:

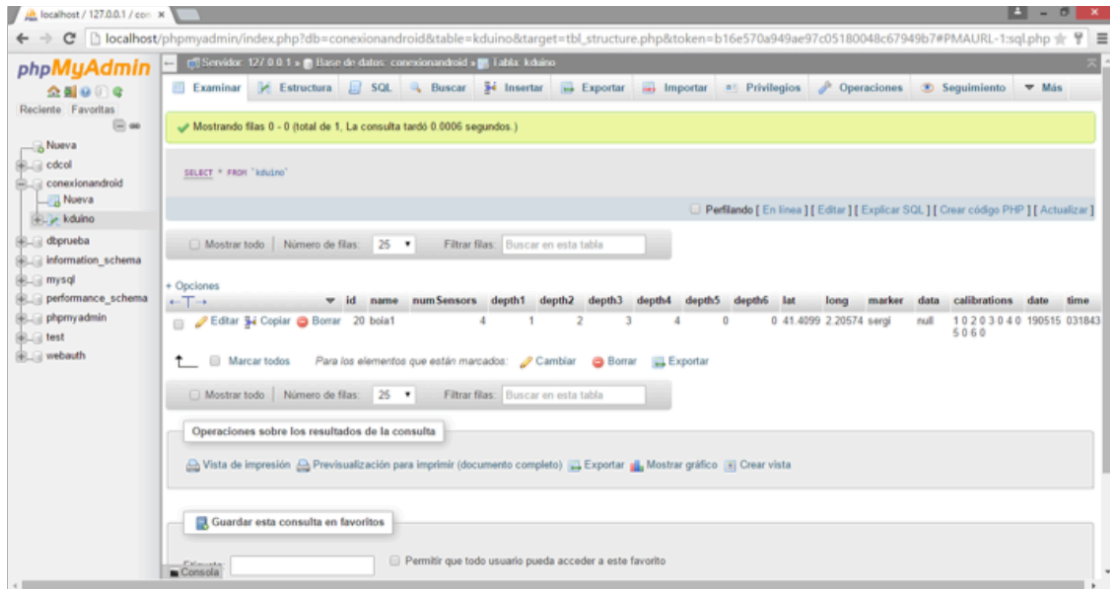


Fig.70 – Resultat de la inserció de dades.

5. Proves i avaluació

5.1 – Entorn de proves

Per tal de comprovar el correcte funcionament de l'aplicació, se han realitzat una sèrie de proves. Aquestes proves s'han dut a terme en dos dispositius diferents que han de ser els sistemes habituals d'ús de l'aplicació, un Smartphone i una Tablet amb les següents característiques:

- Samsung Galaxy Tab3 Lite [13]

Sistema Operatiu: Android, Jelly Bean

Processador: Dual-core 1,2GHz

Memòria interna: 8GB

Memòria externa: microSD fins a 32GB

Wi-Fi: 802.11 b/g/n

Bluetooth v4.0 A2DP

GPS: amb suport A-GPS i GLONASS

Dimensions: 116,4x193,4x9,7 mm

Dimensions pantalla: 1024x600 píxels (7 polsades)

Multitouch: Si

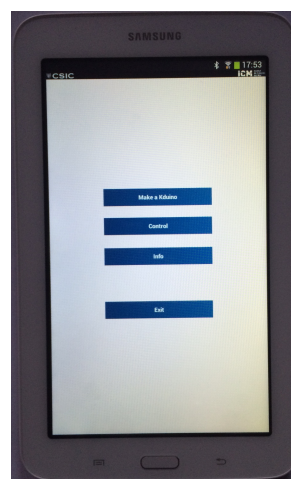


Fig.71 – Tablet Galaxy Tab3.

- Samsung Galaxy Express 2 [14]

Sistema Operatiu: Android, Jelly Bean

Processador: Dual-core 1,7GHz

Memòria interna: 1,5GB

Memòria externa: microSD fins a 64GB

Wi-Fi: 802.11 b/g/n

Bluetooth v4.0

GPS: amb suport A-GPS i GLONASS

Dimensions: 132,4x65,8x9,8 mm

Dimensions pantalla: 540x960 píxels

Multitouch: Si

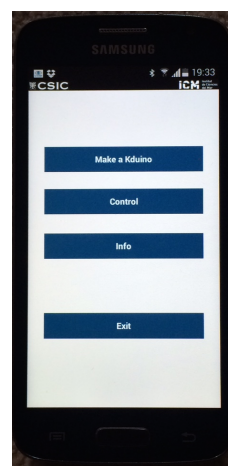


Fig.72 – Smartphone Galaxy Express 2.

5.2 – Demostració

Per a veure el correcte funcionament del programa, es pot accedir a les següents adreces de Youtube, on s'han penjat dos vídeos que mostren totes les funcionalitats de l'aplicació en un entorn real.

Funcionalitat amb el Kduino:

https://www.youtube.com/watch?v=BWJeG_tO_so&feature=youtu.beFuncionalitat

Funcionalitats sobre la base de dades:

<https://www.youtube.com/watch?v=K0n7A-X5pf0>

5.3 – Resultats de les proves realitzades

La taula 2 mostra el resultat de l'avaluació del producte final, en funció de si compleix o no amb els requisits establerts a l'inici del projecte.

	Correcte	No correcte
APIS		
19	<input checked="" type="checkbox"/>	<input type="checkbox"/>
21	<input checked="" type="checkbox"/>	<input type="checkbox"/>
PORTABILITAT		
Smartphone	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Tablet	<input checked="" type="checkbox"/>	<input type="checkbox"/>
FUNCIONALITATS		
KDUINO		
Connexió outdoor	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Connexió indoor	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Set time	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Start measurements	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Stop measurements	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Get data	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Delete data	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Calibration	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Read time	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Test sensors	<input checked="" type="checkbox"/>	<input type="checkbox"/>
BASE DE DADES		
Connexió	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Inserir dades	<input checked="" type="checkbox"/>	<input type="checkbox"/>
BLUETOOTH		
Connexió	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Desconnexió	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Find Devices	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Abast de cobertura:	10m	
INTERNET		
Connexió web	<input checked="" type="checkbox"/>	<input type="checkbox"/>
GENERALS		
Confiançabilitat	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Usabilitat	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Eficiència	<input checked="" type="checkbox"/>	<input type="checkbox"/>

Taula 2 – Resultats de l'avaluació del producte resultant.

6. Conclusions i línies de futur

6.1 – Conclusions

En aquest Treball Final de Grau s'ha realitzat una aplicació per al Sistema Operatiu Android que interactua amb un microcontrolador Arduino contingut en una boia, en funció de les necessitats requerides pel projecte sota el qual es duu terme.

Una vegada realitzada la codificació i portades a terme les proves en un entorn real, es pot comprovar que l'aplicació funciona correctament i s'han assolit els objectius que es van establir al principi del projecte.

Encara així, el sistema no està lliure d'errors, si bé es cert que s'ha intentat minimitzar els errors propis del programari, existeixen errors externs a l'aplicació que poden ser deguts tant a la connexió a la boia a través de la xarxa Bluetooth, com a la connexió amb la base de dades. Altres errors poden provindre de l'enviament i recepció de dades amb la boia, causats per un mal funcionament de la boia, ja que aquesta en condicions normals pot estar en llocs de extrema temperatura i pot causar deficiències en el seu funcionament.

L'objectiu del tractament d'errors ha sigut el de depurar els errors freqüents que poden posar en compromís el funcionament del sistema, i no tant l'eliminació completa dels errors, ja que la correcció d'errors infreqüents no afecta a la fiabilitat percebuda per l'aplicació.

A causa de les necessitats intrínseques del projecte, que com s'ha comentat anteriorment, ha de poder ser utilitzat per persones que no necessàriament han de tenir coneixements tècnics del funcionament de la boia, s'ha obtingut un producte que no ignora la usabilitat i pretén realitzar les funcions d'una manera simplificada i de la manera més lògica possible. Per això s'ha dividit cuidadosament les opcions de navegabilitat i s'organitzen de tal manera que l'usuari interpreti el que es requereix en cada moment.

Per aquest motiu, també proporciona un disseny que no està sobredimensionat, amb un estil gràfic definit i que proporciona alhora una certa sobrietat, si no és així l'usuari podria pensar que l'aplicació és de mala qualitat i no arribi a aconseguir l'objectiu per a la qual va ser creada.

En general, l'aplicació és robusta i presenta senzillesa de funcionament fet que la fa idònia per a la usabilitat que es vol donar.

La implementació de l'aplicació ha estat realitzada amb llenguatge JAVA que resulta familiar, ja que ha estat el llenguatge recurrent de programació de diverses

assignatures de la carrera. Per aquest motiu, tot i que el treball se ha realitzat amb el programari Android Studio, el fet de haver treballat anteriorment amb l'Eclipse en diverses assignatures ha estat essencial per poder extrapolar el funcionament i poder treballar còmodament amb l'Android Studio.

La necessitat de maximitzar l'eficiència del programa i el fet d'haver d'aconseguir unes funcionalitats necessàries, ha portat un reforçament de les tècniques de programació adquirides prèviament i engrandir els coneixements sobre el llenguatge de programació.

El fet d'haver desenvolupat un sistema en totes les seves etapes, ha requerit establir prèviament un model de desenvolupament. Per tal d'adequar els sistemes estàndards al projecte actual, tenint present que el procés s'integra dins d'un projecte global en el qual existeix un equip de treball amb una dinàmica de treball pròpia, s'ha establert un model que pogués permetre establir objectius parcials i demostrables durant tot el procés. El fet de disposar d'una realimentació entre les diferents etapes en aquest cas era indispensable ja que, ha permès en tot moment no allunyar-se dels requeriments exigits pel projecte i a garantir l'èxit de l'aplicació. Aquest fet ha sigut notablement positiu, ja que en cada etapa s'han pogut contrastar els resultats i estructurar així els temps en funció de les entregues.

Aquesta estratègia de desenvolupament ha permès estructurar fàcilment la programació de l'aplicació, establint els passos lògics que ha de seguir l'usuari quan opera amb el programari. A més de facilitar l'estructuració de les pantalles d'una manera lògica, fàcil i intuïtiva per a l'usuari amb un simple menú principal.

La naturalesa del projecte també ha implicat saber fer una selecció adequada de la informació necessària que es requeria per a poder dur a terme la meua tasca i administrar els recursos necessaris, ja que el projecte engloba més complexitat.

En global el present treball ha servit per materialitzar coneixements teòrics adquirits i posar-los en pràctica en un projecte real, en el qual s'ha hagut de donar resposta a una problemàtica que es plantejava, que és la tasca de tot enginyer.

6.2 – Línies de futur

Tot i que els requisits establerts des de'l principi han estat assolits i una vegada contrastat el funcionament amb el sistema que es troba funcionant en l'actualitat, i per tant en condicions de treball normals, és presenta una problemàtica per a ser viable per al seu ús dins del projecte Citclops.

L'aplicatiu per tal de poder ser utilitzat en un entorn real, s'ha de sotmetre a certs canvis que s'engloben dins de l'apartat 4.9 del present treball. En condicions normals la base de dades sobre la que ha de descarregar les dades l'aplicació és privada i externa a l'organització, per tant, per raons òbvies no s'han pogut realitzar en aquest treball i per això, s'ha hagut de implementar de manera local. Això requerirà en un futur immediat que l'aplicatiu sigui sotmès a certs canvis en el programari referent a la inserció de dades a la base de dades.

A mesura que les necessitats del projecte vagin augmentant, augmentarà també les funcionalitats de l'aplicatiu i allargarà el cicle de vida del mateix.

Per aquest motiu el cicle de vida del projecte no acaba amb aquest treball i ha de seguir a poder ser una aplicació funcional dins del projecte. Però, aquests objectius queda fora de l'abast del present treball.

7. Glossari

WPAN	Wireless Personal Area Network
WIFI	Wireless Fidelity
RF	Radio Frequency
CSIC	Centre Superior d'Investigacions Científiques
ICM	Institut de Ciències del Mar
ISO	International Organization for Standardization
API	Application Programming Interface
GPS	Global Positioning System
JRE	Java Runtime Environment
JDK	Java Development Kit
IDE	Integrated Development Environment
SDK	Software Development Kit
IEEE	Institute of Electrical and Electronics Engineers
SGBD	Sistema de Gestió e Base de Dades
USB	Universal Serial Bus
DB	Data Base
IP	Internet Protocol
PHP	Acrònim recursiu de Hypertext Preprocessor

Símbols

K_d	Coeficient d'Atenuació Difúsa
E	Irradiació de la llum a una profunditat donada
E_o	Irradiació de la llum a la superfície
Z	Profunditat

8. Bibliografia

- [1] Institut de ciències del mar (ICM). Projecte MARduino. [Base de dades d'internet]. Catalunya: Centre superior d'investigacions científiques; 2015 [actualitzat 2015; accés 16 Març 2015]. Disponible a: <http://marduino-project.icm.csic.es>

- [2] Wikipèdia. ISO/IEC 9126. [Base de dades d'internet]. Wikipèdia; 2015 [actualitzat 2 Juny 2015; accés 23 Març 2015]. Disponible a: http://en.wikipedia.org/wiki/ISO/IEC_9126

- [3] Wikipèdia. Henry Hudson. [Base de dades d'internet]. Wikipèdia; 2015 [actualitzat 23 Febrer 2015; accés 17 Març 2015]. Disponible a: http://ca.wikipedia.org/wiki/Henry_Hudson

- [4] Wikipèdia. Angelo Secchi. [Base de dades d'internet]. Wikipèdia; 2015 [actualitzat 13 Març 2015; accés 17 Març 2015]. Disponible a: http://en.wikipedia.org/wiki/Angelo_Secchi

- [5] David T.Plummer. Introducció a la bioquímica pràctica. Barcelona: Universitat de Barcelona; 1994. Capítol 7 – “Espectrofotometria”.

- [6] Wikipèdia. Disco Secchi. [Base de dades d'internet]. Wikipèdia; 2015 [actualitzat 24 Desembre 2014; accés 19 Març 2015]. Disponible a: http://es.wikipedia.org/wiki/Disco_Secchi

- [7] Elizabeth Myre & Ryan Shaw. The Turbidity Tube: Simple and Accurate Measurement of Turbidity in the Field. [Base de dades d'internet]. Michigan Technological University; 2006 [actualitzat Abril 2006; accés 19 Març 2015]. Disponible a: <http://www.cas.umn.edu/assets/pdf/Turbidity%20Tube.pdf>

- [8] John H. Morrow, Stanford B. Hooker, Charles R. Booth, Germar Bernhard, Randall N. Lind, and James W. Brown. Advances in Measuring the Apparent Optical Properties (AOPs) of Optically Complex Waters. [Base de dades d'internet]. Michigan Technological University; 2010 [actualitzat 2010; accés 19 Març 2015]. Disponible a:
<http://www.biospherical.com/images/pdf/morrow-215856.pdf>
- [9] Wikipèdia. Nefelòmetro. [Base de dades d'internet]. Wikipèdia; 2015 [actualitzat 25 Maig 2015; accés 23 Març 2015]. Disponible a:
<http://es.wikipedia.org/wiki/Nefelòmetro>
- [10] North American Lake Management Society. [Base de dades d'internet]. Nord Amèrica; 2015 [actualitzat 2015; accés 20 Març 2015]. Disponible a:
<http://www.secchidipin.org>
- [11] Laboratorio Nacional de Calidad del software. Ingeniería del Software. [Base de dades d'internet]. Ministerio de Industria, Turismo y Comercio, Espanya; 2009 [actualitzat Març 2009; accés 23 Març 2015]. Disponible a:
http://datateca.unad.edu.co/contenidos/301569/guia_de_ingenieria_del_softw are.pdf
- [12] Android Developers. [Base de dades d'internet]. [actualitzat 2015; accés 20 Març 2015]. Disponible a: <https://developer.android.com/index.html>
- [13] Samsung. [Base de dades d'internet]. [actualitzat 2015; accés 29 Maig 2015]. Disponible a: <http://www.smart-gsm.com/moviles/samsung-galaxy-tab3-lite>
- [14] Samsung. [Base de dades d'internet]. [actualitzat 2015; accés 29 Maig 2015]. Disponible a:
<http://www.samsung.com/es/consumer/mobiledevices/smartphones/others/S M-G3815ZBAATL>

9. Annexos

Els scripts corresponents a la connexió i inserció de dades en la base de dades corresponen als següents arxius:

Arxiu "functions.php":

```
<?php
header("Content-Type: text/html;charset=utf-8");

function ejecutarSQLCommand($comando){
    $mysqli=new mysqli("localhost","root","contransenya","conexionandroid");
    $mysqli->query("SET NAMES 'utf8'");

    if($mysqli->connect_error){
        printf("Connect Failed: %s\n", $mysqli->connect_error);
        exit(); }

    if($mysqli->multi_query($comando)){

        if($resultset=$mysqli->store_result()){
            while($row=$resultset->fetch_array(MYSQLI_BOTH)){ }

            $resultset->free();}
        }
        $mysqli->close();
    }
    function getSQLResultSet($comando){
        $mysqli = new mysqli("localhost","root"," contransenya ","conexionandroid");
        $mysqli->query("SET NAMES 'utf8'");

        if($mysqli->connect_error){
            printf("Connect failed: %s\n", $mysqli->connect_error);
            exit(); }

        if($mysqli->multi_query($comando)){
            return $mysqli->store_result();}

        $mysqli->close();
    }
?>
```

Arxiu "registro.php":

```
<?php
include ('functions.php');

$name=$_GET['name'];
$numSensors=$_GET['numSensors'];
$depth1=$_GET['depth1'];
$depth2=$_GET['depth2'];
$depth3=$_GET['depth3'];
$depth4=$_GET['depth4'];
$depth5=$_GET['depth5'];
$depth6=$_GET['depth6'];
$lat=$_GET['lat'];
$long=$_GET['long'];
$marker=$_GET['marker'];
$data=$_GET['data'];
$calibrations=$_GET['calibrations'];
$date=$_GET['date'];
$time=$_GET['time'];

ejecutarSQLCommand("INSERT INTO
`kduino`(`name`,`numSensors`,`depth1`,`depth2`,`depth3`,`depth4`,`depth5`,`depth6`,`lat`,`long`,`marke
r`,`data`,`calibrations`,`date`,`time`) values
('$name','$numSensors','$depth1','$depth2','$depth3','$depth4','$depth5','$depth6','$lat','$long','$marker'
,$data','$calibrations','$date','$time')

ON DUPLICATE KEY UPDATE
`name`='$name',    `numSensors`='$numSensors',    `depth1`='$depth1',    `depth2`='$depth2',
`depth3`='$depth3',    `depth4`='$depth4',    `depth5`='$depth5',    `depth6`='$depth6',    `lat`='$lat',
`long`='$long',    `marker`='$marker',    `data`='$data',    `calibrations`='$calibrations',    `date`='$date',
`time`='$time");

?>
```