



Universidad del País Vasco Euskal Herriko Unibertsitatea



Bilboko Industria Ingeniaritza Teknikoko Unibertsitate Eskola Escuela Universitaria de Ingeniería Técnica Industrial de Bilbao

KUDEAKETAREN ETA INFORMAZIO SISTEMEN INFORMATIKAREN INGENIERITZAKO GRADUA

GRADU AMAIERAKO LANAREN MEMORIA

Kanpoko kirolentzako *LiveTrack* sistema baten garapena

Egilea: Ander Orbegozo Igartua

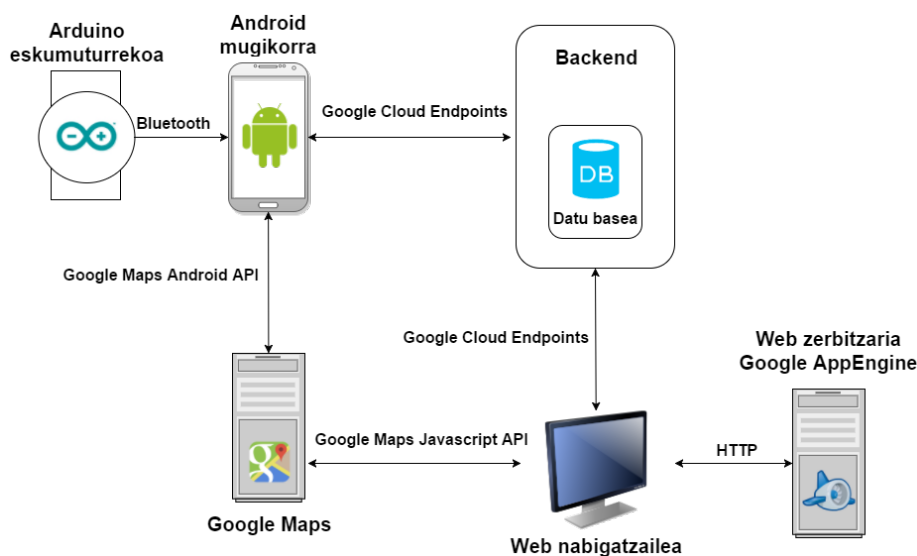
Tutorea: Oskar Casquero Oyarzabal

Iraila - 2015

Laburpena

Kanpoko kirolari baten jarraipena denbora errealean egiteko sistema oso baten garapena osatu da. Sistema honek hiru zati nagusi ditu(irudia 1):

- GPS-dun eskumuturrekoa. Hau kirolariak eramango du gainean bere posizioa lortzeko.
- Android aplikazioa. Kirolariak eramango du eta eskumuturrekoaren datuak internet bidez datu basera igoko ditu.
- Web zerbitzaria. Kirolariaren jarraipena egiteko balio du. HTML 5 estandarrean oinarrituta dago.



Irudia 1: Proiektuaren eskema orokorra

Eskumuturrekoa, kirolariak gainean eramango du. Honen lehenengo prototipoa Arduino Uno baten bidez egin da. Prototipo hau, eskumuturreko bat izateko handiegia da baina garatze prozesuan tarteko pauso bat da. Horrela, errazago manipulatu daitezke konexioak. Erabilitako GPS eta Bluetooth moduluak serial bidez konektatzen dira. Arduino-ra konektatu baino lehen ordenagailuan konprobatu dira Serial-USB kable bat erabiliz.

GPS moduluak, NMEA sententziak bidaltzen ditu serial konexioaren bidez. NMEA sententzia mota ezberdinak daude (GLL, GGA, MSS...). Hauek dira NMEA sententzien adibide batzuk:

- \$GPGLL,4916.45,N,12311.12,W,225444,A,*1D
- \$GPVTG,054.7,T,034.4,M,005.5,N,010.2,K*48
- \$GPWPL,4807.038,N,01131.000,E,WPTNME*5C

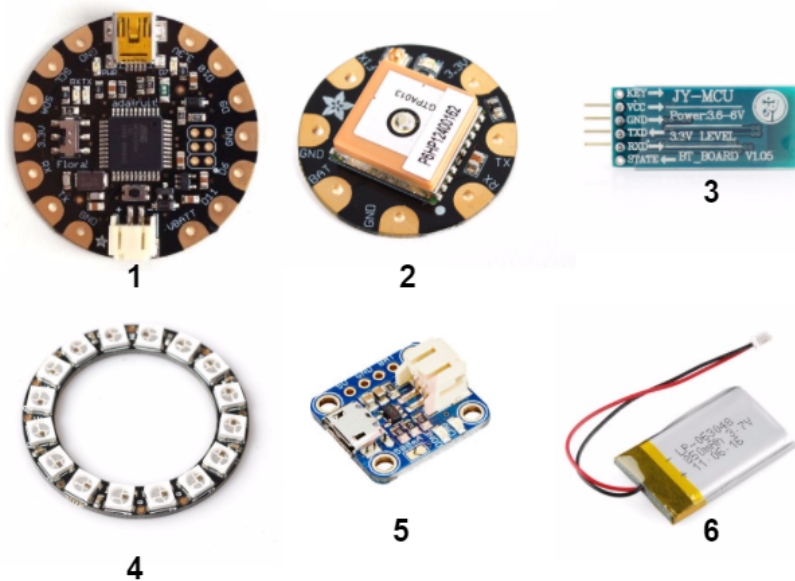
Sententzia horiek tinyGPS++ liburutegiaren bidez kudeatzen dira mikro-kontroladorean.

Behin datuak kudeatuta, datuak Bluetooth moduluaren bidez mugikorrera bidaliko dira. Hori egiteko protokolo propio bat definitu da. Protokolo hone-tan, datuak “,” batez bananduta bidaltzea erabaki da. Lerroa non hasten eta bukatzen den jakiteko lerroaren hasieran “#” bat jarri da eta “@” bat bukaeran. Bidaltzen den lerroak hurrengo egitura du:

- #accuracy-valid,accuracy,location-valid,latitudea,longituda,speed-valid,abiadura,bearing-valid, bearing, altitud-valid,altituda@

Behin pieza handiekin bukatuta, pieza txikiagoetara aldatu da (irudia 2):

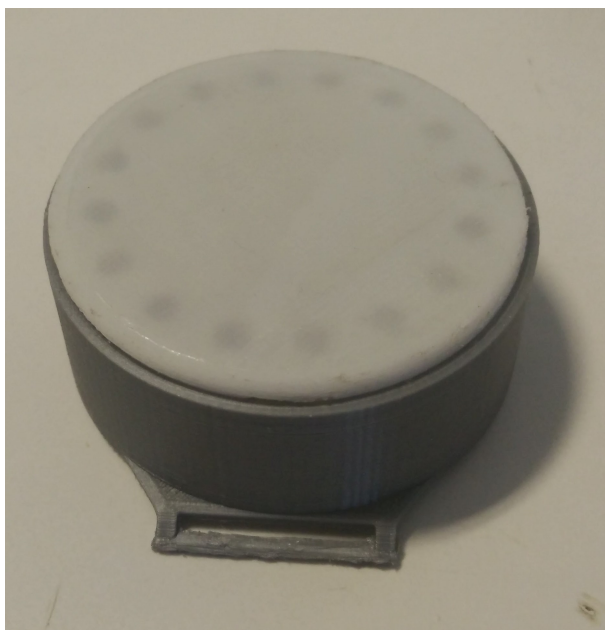
- Adafruit Flora (1)
- Flora Wearable Ultimate GPS Module (2)
- HC-06 Bluetooth (3)
- NeoPixel Ring 16 (4)
- Adafruit MicroUSB - USB LiIon/LiPoly charger (5)
- Lithium Ion Polymer Battery - 3.7v 150mAh (6)



Irudia 2: Eskumuturrekoan erabilitako piezak

GPS moduluak ordua jasotzen duenez, zirkulu bat osatzen duten NeoPixel Ring led-en bitartez, eskumuturrekoak ordua adierazten du. Led horiek GPS-ak konexioa duen edo ez adierazteko ere erabiltzen da kolore gorria eta berdearen bitartez.

3 dimentsiotako inprimagailu baten bidez egindako karkasa bat (irudia 3) egin da eskumuturrekoa bertan sartu ahal izateko. Honek, 20mm-ko korrea arruntak erabiltzeko aukera ematen du. Diseinua paperean egin da eta ondoren SketchUp programara pasa da inprimatua izateko. Karkasak bi zati ditu. Alde batetik, tapa. Honek sakonera gutxiago dauka led-ak doazen lekuetan argia pasa dadin. Gainera, zulo batzuk egin dira beste zatiarekin lotzeko. Bestetik, karkasaren zati nagusia dago. Zati honek bateriarendako zulo bat du eta patilla batzuk ditu taparen zuloetan sartzeko eta horrela bi piezak lotzeko.



Irudia 3: 3D-tan inprimatutako karkasa

Android mugikorrerako aplikazioa sortu da. Aplikazio honek, eskumuturrekoak Bluetooth bidez bidalitako datuak jasoko ditu. Eskumuturreko gabe funtzionatzeko aukera dago mugikorraren GPS-a erabiliz. Ibilbide berri bat hastean, datuak mapan agertuko dira gure posizioa eta egin dugun ibilbidea marraztuz. Mapa mota ezberdinak aukeratu daitezke (Mapa arrunta, satelite irudiak, hibridoa). Mapa Android-eko Google Maps API-aren bidez kargatzen da. Abiadura eta altuera datuak, grafiko batean ikusten dira.

Honetaz gain backend-arekin komunikatuko da datu hauek datu basera igotzeko. Mugikorrak ibilbidea hastean eta amaitzean interneteko konexioa edukitzea beharrezkoa da datuak igotzeko. Tartean posible da internet konexiorik ez izatea. Kasu honetan datuak memorian gordeko dira beranduago igotzeko.

Backend-ak hainbat metodo ditu datu basera datuak igo ahal izateko edo datuak jasotzeko. Backend hau Google Cloud Endpoints erabiliz garatu da. Datu baseko edozein operazio egiteko, backend-aren metodoak erabiltzen dira:

- `trackBerriaHasi()`. Ibilbide berria sortuko du eta ibilbide horren identifikadorea bueltatuko du. Saioa hasita edukitzea beharrezkoa izango da.
- `wayPointAdd()`. Ibilbide batera puntu bat igoko du bere informazioarekin (latitueda, longitueda, abiadura...). Saioa hasita edukitzea beharrezkoa izango da.
- `getTrack()`. Ibilbide bat lortuko du bere datu guztiekin. Ibilbidea pribatua bada saioa hasita egon beharko da.
- `editTrack()`. Ibilbidearen aukerak aldatuko ditu (izena, pribatua...). Saioa hasita egon behar da.
- `deleteTrack()`. Ibilbidea ezabatuko du. Saioa hasita edukitzea beharrezkoa izango da.
- `myTracks()`. Erabiltzailearen ibilbide guztiak bueltatuko ditu. Saioa hasita edukitzea beharrezkoa izango da.
- `endTrack()`. Ibilbidea amaituta dagoela markatuko du. Saioa hasita edukitzea beharrezkoa izango da.

Backend-ak, erabiltzaileak baimenduta dagoen ere kontrolatuko du. Baimenduta ez badago errore bat bueltatuko da hori adierazteko. Erabiltzaileak OAuth bidez hasiko dute saioa Google kontu batekin bai webgunean, bai android aplikazioan.

Datu baseak ibilbideak gordeko ditu. Datu basea nosql motakoa da. Ondorioz, etorkizunean eskakizunak aldatzen badira, eskema aldatzen joan daiteke. Ibilbide bakoitza erabiltzaile batekin lotuta egongo da eta hainbat WayPoint izango ditu. WayPoint bakoitzak, GPS-ak momentu batean jaso dituen datuak gordetzen ditu (latitueda, longitueda, abiadura, altuera...). Ibilbide bakoitzak identifikadore bat izango du eta identifikadore horretaz gain erabiltzailearen email helbidea beharrezkoa da ibilbide konkretu bat sisteman identifikatzeko. Ibilbideak amaituta dagoen edo ez gordeko du. Amaituta badago internet konexioa eta bateria aurrezteko lortu daiteke petizio berriak ekiditen. Honez gain, ibilbidea markatzerakoan puntuak erabili beharrean jasotako posizio bakoitzeko, oraingo posizioa markatzen da eta ibilbidea marrarekin bakarrik marrazten da. Honi esker aplikazioaren askoz fluidoago funtzionatzen du eta mapan zoom egitean edo desplazatzean ez da motel nabaritzen.

Aplikazioak, ibilbidea partekatzeko aukera du. Honek ibilbidearen URL-a sortuko du eta bidaltzeko erabili nahi den aplikazioa eskatuko du (Whatsapp, email, twitter, facebook...). Behin URL hori jasota irekitzen saiatzen denean aplikazioa instalatuta badago URL-a aplikazioarekin irekitzeko aukera dago. Aplikazioarekin irekitzen bada denbora errealean ikusiko du kirolariaren posizioa mapan eta grafikoa aurretik aipatu den moduan.

Aplikazioa ez badago instalatuta edo ordenagailu batean egonez gero, web nabigatzailean zabalduko da url-a. Webgunea, JSP, JavaScript eta CSS fitxategiekin osatuta dago. JSP fitxategiak url-aren parametroak prozesatuko ditu

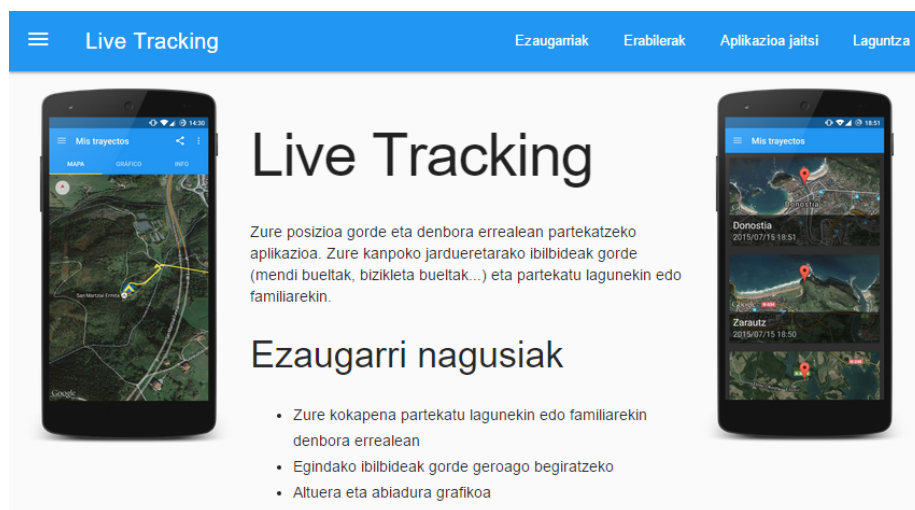
(ibilbidearen identifikadorea eta sortu duenaren jabea). Ondoren web nabigatzaileari prozesaturiko HTML kodea, Javascript eta CSS-ak bidaliko dira. Javascript bidez, Google Maps zerbitzuari deiak egingo dira eta mapa kargatuko da. Gainera, backend-ari deiak egingo dizkio ibilbidearen identifikadorea eta sortu duenaren email-arekin, ibilbidearen datuak lortzeko. Behin datuak lortuta, ibilbidea eta posizioa mapan marraztuko dira. Backend-ari deiak errepikatu egingo dira datu berriak lortzeko ibilbidea oraindik ez bada amaitu.

Ibilbide bat pribatua edo publikoa den aukeratu daiteke aplikaziotik. Publikoa bada, url-a duen edonork ikusi dezake ibilbidea. Pribatua bada, defektuz ibilbidearen jabeak bakarrik ikusi ahalko du baina baimendutako email helbideak gehitu daitezke horiek ere ikus dezaten.

Aplikazioan gure ibilbide guztien zerrenda ikusten da. Zerrendako elementu bakoitza ibilbide bat da eta bere izena, hasi zen data eta ordua eta mapatxo batean hasi zen lokalizazioa agertzen da. Hemen, aurretik egindako ibilbideak kudeatu daitezke. Ibilbideak ezabatu, partekatu edo berrikusi daitezke.

Webguneak proiektuaren inguruko informazio orrialde batzuk ere ditu (irudia 4). Material Design Lite liburutegia erabili da itxura hobea izan dezan. Aplikazioaren inguruko informazioa dago. Webgunea hurrengo helbidean aurkitu daiteke:

- <http://live-tracking.appspot.com/>



Irudia 4: Webgunearen kaptura

Aplikazioa, PlayStore dendan aurkitu daiteke aorbegozo005 bilaketa eginez edo hurrengo helbidean sartuz:

- <https://play.google.com/store/apps/details?id=com.aorbegozo005.livetrack>

Aplikazioa hainbat erabiltzailerekin probatu da eta jasotako feedback-ari esker erroreak zuzendu eta funtzioak hobetu dira.

Aurkibidea

1	Sarrera	1
2	Hasierako planteamendua	2
2.1	Arriskuak	2
2.2	Denbora plangintza	2
2.3	Tresnak	6
3	Aurrekarien azterketa	9
3.1	Garmin Livetrack	9
3.2	LocaToWeb	10
3.3	Strava	10
4	Eskakizunen kaptura	11
4.1	Ibilbidea ikusi	12
4.2	Erregistratu	12
4.3	Saioa hasi	12
4.4	Erregistratu gabeko erabiltzailea	12
4.5	Erabiltzaile erregistratua	12
4.6	Ibilbide berria sortu	12
4.7	Nire ibilbideak kudeatu	12
5	Azterketa eta diseinua	13
5.1	Eskumuturrekoa	15
5.2	Backend-a	15
5.3	Datu basea	16
5.4	Mugikorreko aplikazioa	17
5.4.1	Sekuentzia diagrama	19
5.5	Webgunea	21
6	Garapena	22
6.1	Arduino eskumuturrekoa	22
6.1.1	Lehenengo prototipoa	22
6.1.2	Bigarren prototipoa	26
6.1.3	Karkasa	31
6.2	Android aplikazioa	33
6.2.1	Lehenengo bertsioa	34
6.2.2	Bigarren bertsioa	34
6.2.3	Hirugarren bertsioa	36
6.2.4	GPS iturriak	38
6.2.5	Listener-ak	39
6.2.6	Abiadura eta altueraren grafikoa	40
6.2.7	GPX	41
6.2.8	Distribuzioa	42
6.3	Backend-a eta datu basea	42
6.4	Webgunea	42
6.4.1	Lehenengo bertsioa	43
6.4.2	Bigarren bertsioa	43
6.4.3	Hirugarren bertsioa	43

6.4.4	Twitter Cards	44
6.4.5	Open Graph Protocol	45
6.5	Promozioa	45
7	Egiaztapena eta ebaluazioa	47
7.1	Backend-a eta datu basea	47
7.2	Webgunea	47
7.3	Erabiltzaileekin probak	47
7.3.1	Lehenengo erabiltzailea	47
7.3.2	Bigarren erabiltzailea	48
7.3.3	Hirugarren erabiltzailea	48
7.3.4	Laugarren erabiltzailea	48
8	Ondorioak eta etorkizuneko lanak	49
8.1	Balorazioa pertsonala	49
8.2	Erabiltzaileen harrera	49
8.3	Lanaren ustiapena	51
8.4	Etorkizunerako lana	51

Irudiak

1	Proiektuaren eskema orokorra	i
2	Eskumuturrekoan erabilitako piezak	ii
3	3D-tan inprimatutako karkasa	iii
4	Webgunearen kaptura	v
5	Gantt deskribapena	4
6	Gantt 1.zatia	5
7	Gantt 2.zatia	5
8	Gantt 3.zatia	6
9	Gantt 4.zatia	6
10	Aplikazioa instalatua izan den mugikorren modeloak	7
11	Garmin Connect webgune jarraitzilea	9
12	LocaToWeb webgune jarraitzilea	10
13	Strava-ko ibilbide bilaketa	11
14	Erabilpen kasuak	11
15	Proiektuaren arkitektura	13
16	Konexioaren eskema	14
17	Datu basearen diagrama	16
18	Ibilbidea ikusteko pantailaren prototipoak	18
19	Datu basearen diagrama	19
20	GPS datuak jasotzen deneko sekuentzia diagrama	20
21	Boltaje aldatzailea	23
22	Eskumuturrekoaren lehenengo prototipoa	23
23	GPS-ak bidalitako NMEA sententziak Arduinoren IDE-aren serie monitorean	24
24	Bluetooth bidez datuak jasotzen Bluetooth SPP Tools PRO ap- likazioan	26
25	Bigarren prototipoko materiala	27
26	Eskumuturreko bigarren prototipoa	28
27	Prototipoaren konexioen eskema	28
28	NeoGeo watch (https://learn.adafruit.com/assets/11151)	29
29	NeoPixelak 2:40:35 ordua markatzen	30
30	Paperean egindako karkasaren bozetoa	32
31	3 dimentsiotan egindako karkasaren basea eta tapa	33
32	3 dimentsiotan inprimatutako karkasa	33
33	Android aplikazioaren lehenengo bertsioak Bluetooth bidez jaso- tako datuak erakusten zituen.	34
34	Android aplikazioaren bigarren bertsioa posizio puntu bakarrarekin	35
35	Google Developer Console. Beharrezkoak diren ID-ak ateratzen dira	36
36	Ibilbidearen kolorearen konparaketa.	37
37	Mapan puntuak gertu badaude ez da ondo ikusten	38
38	Androideko activity eta fragmenten eskema	39
39	Ezkerraldeko marra desagertuta	41
40	Webgunearen itxura Material Design Lite liburutegia erabilia	43
41	Twitter card erabilia aplikazioa deskargatzeko aukera agertzen da	44
42	Open Graph adibidea Telegram aplikazioan	45
43	Irailaren 4an PlayStore kontsolako datuak	49

44	Appbrain-etik ateratako grafikoa. Espainiako PlayStore-eko kirol sekzioan aplikazioak izandako posizioak	50
45	PlayStore-eko kirol sekzioko aplikazioa berrien top zerrenda . . .	51

Taulak

1	Denbora plangintza	3
2	NMEA adibide sententziaren azalpena	24
3	Bluetooth bidez mugikorrari bidaltzen zaizkion datuak	25

1 Sarrera

GAL hau kanpoko kirolak (txirrindularitza, trekking) egiten dituzten kirolari edo kirolzaleei zuzendua dago. Alde batetik, pertsona hauetako askok bakarka entrenatzen dute, honek beraien segurtasunari begira dakartzan ondorioekin. Bestalde, bakarka entrenatzeak etxeoak arduratuta uzten ditu entrenamendu saioak burutzen dituzten bitartean.

LiveTrack sistemak oso hedatuta daude paketeri garraio eta banaketan, baina ez kirol arloan. Proiektu honen helburua kirolarientzako *Livetrack* sistema bat garatzea da.

Testuinguru honetan, GAL honek *wearable device* edo janzteko gailu batean oinarritutako *LiveTrack* sistema bat garatzea du helburu. Sistemaren egiturari dagokionez, ondorengo atalak aurreikusten dira:

- GPSa duen eskumuturrekoa. Kirolariak eramango du kirola egiten duen bitartean.
- Android sistema eragilearentzako aplikazioa. Aplikazio honek hari gabeko teknologia bitartez eskumuturrekotik GPS datuak aldizka jaso eta datu biltegi batera igoko ditu.
- Webgune jarraitzailea. Webgune honek mapa batean, denbora errealean, ibilbidea jarraitzeko aukera emango du.

Sistema hau garatzeko ideia, nire mapekin lan egiteko interesarekin eta GAL honen zuzendariaren (Oskar Casquero) txirrindularitzarako zaletasunetik sortu da. Interesgarria eta guztiz gaur egungoa ikusten dut mapekin lan egitea eta mugikorreko aplikazio bat garatzea. Gainera, trebakuntza lortu dut hardware eta elektronika aldetik, ezagutza arlo hauetara loturiko konpetentziak, gaur egun “Internet of Things” izenarekin izendatu izan den industria esparru berrian garrantzi handia hartzen dutenak, ez baitira graduan lantzen. Azkenik, aplikazioa appstore batera igo nahi izan da prozesu hau nola egiten den ikasteko, proiektua zabalduz erabiltzaile-multzo handi baten feedback-a edo berrelikadura jasotzeko, eta GAL-etik eratorritako prototipoari “produktu” itxura emateko.

2 Hasierako planteamendua

Proiektu honen helburua denbora errealean kirolari baten jarraipena egiteko aukera ematen duen sistema oso bat garatzea da. “Sistema osoa” sistema integral bezala ulertu behar da, non funtzio jakin bat betetzen duen software bat diseinatu eta garatzeaz aparte, berarekin lan egingo duen hardware-a ere diseinatu eta garatu egiten da, erabiltzaileari bere jardura betetzeko behar duen guztia emanaz. Aurretik aipatutako hiru zatiak garatuko dira proiektu honetan.

2.1 Arriskuak

Proiektu honen garapenaren arriskurik handiena hardware inguruan dago. Kudeaketa eta Informazio Sistemen Informatikaren Ingeniaritzako Graduko ikasleok ez dute elektronikako prestakuntzarik jasotzen, hau GAL honen garapen egokirako arrisku bat delarik. Arazo hau konpontzeko, elektronikazaleen artean eta hezkuntzan arrakasta handia izan duten open source hardware plataforma (Arduino) eta kit-ak (Adafruit eta Sparkfun-ek fabrikatzen dituztenak) erabiliko dira GPS-a duen eskumuturrekoaren muntaia egiteko. [39]

Honetaz gain, proiektuaren hasieran Arduino-ren inguruko prestakuntza egingo da proiektu txiki batzuk eginez GAL honen zuzendariaren gidarekin hurrengo gaietan trebatuz:

- Mikrokontroladore egitura eta programazioa
- Serie komunikazio bus-ak (I2C, SPI)
- Osagai elektronikoko-digitalen arteko konexioa (tentsioak, soldatzea, eta abar)

GAL-aren garapenaren inguruan agertzen den beste arrisku bat informazioaren (kodea edo dokumentazioa) galerarekin erlazionatuta dago, adibidez, ordenagailua apurtzen delako. Kasu honetan, github erabiliz datuak beste ordenagailu batera berreskuratzeak aukera egongo da baina softwarearen instalakuntzak atzerapena ekarriko du. Bestetik, github erabiltzeak zuzendariaren eta beste edozein erabiltzailearekin kodea partekatzeak aukera ematen du. Hau oso onuragarria da arazoak edo zalantzak argitzeko.

2.2 Denbora plangintza

Proiektua iterazio ezberdinetan egiteko antolatu da. Edozein proiektutan, antolakuntza modu hau jarraitzeak garrantzia dauka, baina honetan are eta garrantzi handiago hartzen du hardware atalean eremu ezezagun batean mugitu izan naizelako. Lehenengo, Arduino/GPS, Android eta Web Zerbitzarien inguruko prestakuntza jaso eta/edo sakondu izan behar da. Ondoren, funtzio nagusia betetzen duen bertsio simple bat lortuz eta ondoren erroreak zuzenduz. Azkenik, funtzionalitate berriak gehitu eta interfazea hobetu. Ataza bakoitzean erabiliko den denboraren estimazioak egin dira (taula 1).

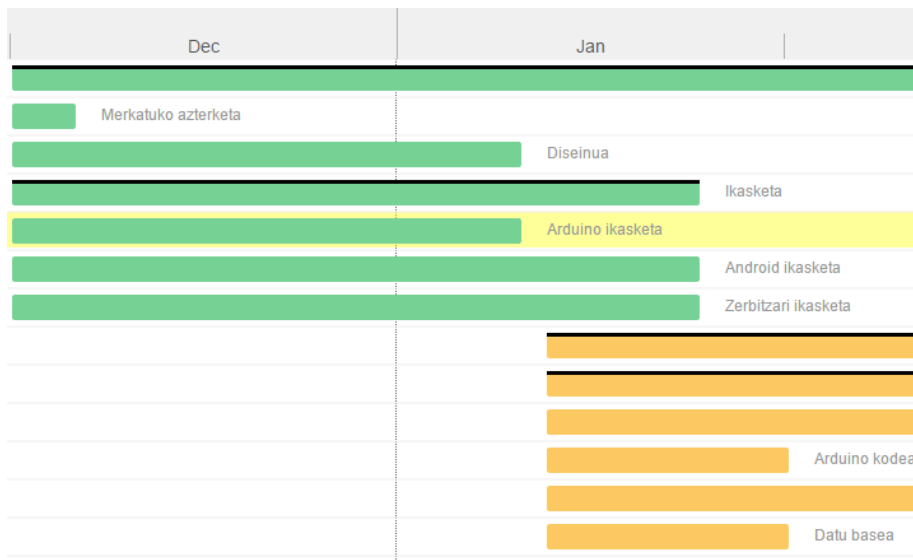
Ataza	Denbora (Orduak)
Dokumentazioa	30
Arduino ikasketa	15
Android ikasketa	15
Merkatuan dauden sistemen azterketa	10
Hasierako diseinua	25
Zerbitzari ikasketa	25
Android aplikazioaren kodea	50
Zerbitzariaren kodea	50
Arduino kodea	25
Eskumuturrekoaren muntaia	25
Android aplikazioaren testeaketa	15
Webgunearen testeaketa	10
Eskumuturrekoaren egiaztapena	10
Guztira	305

Taula 1: Denbora plangintza

Proiektua denboran zehar nola garatuko den ikusteko gantt bat egin da. Gantt honetan (irudia 5), ikusten da lana 5 bloke nagusitan banatu dela. Proiektu osoa 145 egunetan garatzea espero da.

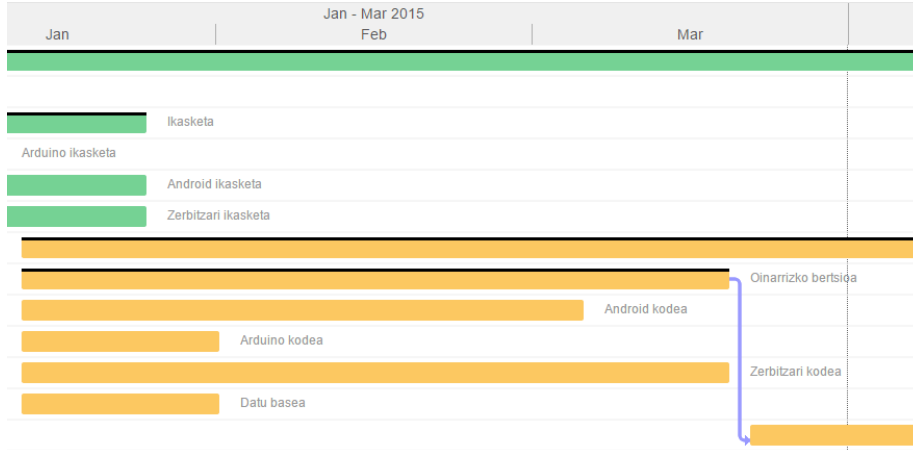
name	start	end	dur.	depe.
[-] Proiektua	12/1/2014	6/19/2015	145	
Merkatuko azterketa	12/1/2014	12/5/2014	5	
Diseinua	12/1/2014	1/9/2015	30	
[-] Ikasketa	12/1/2014	1/23/2015	40	
Arduino ikasketa	12/1/2014	1/9/2015	30	
Android ikasketa	12/1/2014	1/23/2015	40	
Zerbitzari ikasketa	12/1/2014	1/23/2015	40	
[-] Garapena	1/12/2015	6/12/2015	110	
[-] Oinarrizko bertsioa	1/12/2015	3/20/2015	50	
Android kodea	1/12/2015	3/6/2015	40	
Arduino kodea	1/12/2015	1/30/2015	15	
Zerbitzari kodea	1/12/2015	3/20/2015	50	
Datu basea	1/12/2015	1/30/2015	15	
Bigarren bertsioa (Kode hobetu)	3/23/2015	4/17/2015	20	9
Hirugarren bertsioa	5/4/2015	5/29/2015	20	18
[-] Ebaluazioa	4/20/2015	6/12/2015	40	14
Kodearen testeaketa	4/20/2015	6/12/2015	40	

Irudia 5: Gantt deskribapena



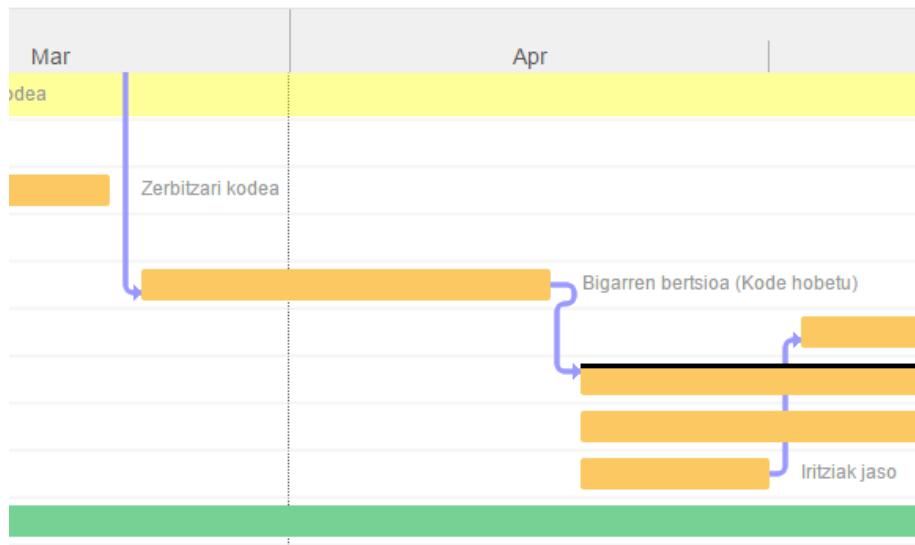
Irudia 6: Gantt 1.zatia

Lehenengo zatian, merkatuko azterketa, diseinua eta ikasketa egingo da. Arduinoko ikasketa besteak baino azkarrago bukatzea espero da beharrezko prestakuntza basikoa delako eta Oskarren laguntzarekin kontaktzen delako (irudia 6).



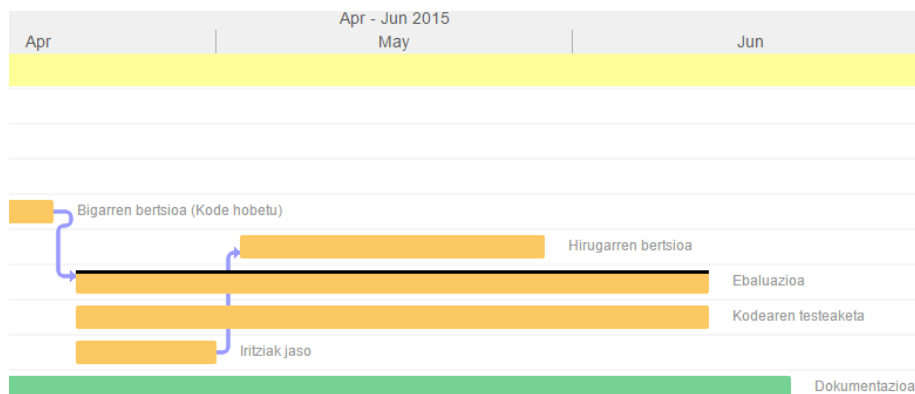
Irudia 7: Gantt 2.zatia

Diseinua amaitu ondoren garapenarekin hasiko da. Datu basea eta Arduinoko kodea azkarrago egingo dela uste da. Android aplikazioa eta zerbitzariak denbora gehiago beharko du (irudia 7).



Irudia 8: Gantt 3.zatia

Lehenengo bertsioa egin ondoren, kodea hobetzen saiatuko da. Ez dira erabilgaitzak erabiliko proiektua probatzeko fase honetan. Errore eta hobekuntzak nik bilatuko ditut fase honetan (irudia 8).



Irudia 9: Gantt 4.zatia

Proiektuaren bigarren bertsioa amaitzean, kodearen egiaztapena hasiko da. Kasu honetan, erabilgaitzak iritziak eta aurkitutako erroreak jasoko dira eta kodea hobetzen eta egiaztatzen jarraituko da (irudia 9).

Dokumentazioa proiektu osoan zehar egiten joango da zikinean. Azkenengo asteetan dena ordenatu eta txukundu egingo da.

2.3 Tresnak

Proiektu hau garatzeko erabili den hardwarea hurrengo hau izan da:

- Ordenagailu eramangarria (Asus K53SV)
- Android Mugikorra (One Plus One)
- 3D-tako inprimagailua (BQ Prusa i3)
- Adafruit Flora
- Flora Wearable Ultimate GPS Module
- HC-06 Bluetooth
- NeoPixel Ring 16
- Adafruit MicroUSB - USB LiIon/LiPoly charger
- Lithium Ion Polymer Battery - 3.7v 150mAh

Hauek izan dira proiektu osoan zehar erabilitako hardware-a, hala ere lagun eta familiarrek euren mugikorretan aplikazioa probatu dute funtzionamendua egokia dela egiaztatzeko. Honetaz gain, aplikazioa PlayStore dendan dagoenez mugikor modelo ezberdin askotan instalatua izan dela jakin daiteke (irudia 10).

INSTALACIONES TOTALES POR USUARIO EL 6/9/2015



		TU APLICACIÓN	
<input type="checkbox"/>	Motorola Moto G (falc...	7	4,19%
<input type="checkbox"/>	Samsung Galaxy S5 (...)	7	4,19%
<input type="checkbox"/>	Google Nexus 5 (ham...	6	3,59%
<input type="checkbox"/>	Samsung Galaxy S4 (...)	5	2,99%
<input type="checkbox"/>	Samsung Galaxy S3 (...)	5	2,99%
<input type="checkbox"/>	Google Nexus 4 (mako)	4	2,40%
<input type="checkbox"/>	Sony Ericsson Xperi...	3	1,80%
<input type="checkbox"/>	Sony Xperia Z3 (D6603)	3	1,80%
<input type="checkbox"/>	Samsung Galaxy S3 ...	3	1,80%
<input type="checkbox"/>	Samsung Galaxy Not...	3	1,80%
<input type="checkbox"/>	Otras	121	72,46%

Irudia 10: Aplikazioa instalatua izan den mugikorren modeloak

Proiektua garatzeko hurrengo software-a erabili da:

- Android Studio. Android aplikazioaren eta zerbitzariaren API-aren garapena egiteko.
- Arduino IDE. Arudino kodea garatzeko.

- Texmaker eta Latex dokumentazioa idazteko.
- Github. Kode bertsio kontrol modura.
- SublimeText HTML, CSS, JSP eta JavaScript kodea editatzeko.
- gantt.twproject.com gantt diagrama egiteko.
- www.draw.io irudi eskematikoak egiteko.
- Chrome nabigatzailea webgunearen garapenerako.
- Apache Tomcat JSP fitxategiak lokalean egiaztatzeko.
- InkScape ikonoak sortzeko.
- Paint.net irudiak eta kapturak editatzeko.
- Google Sketch Up. Karkasaren 3D diseinua egiteko
- realfavicongenerator.net webguneko favicon irudiak eta kode sortzeko.
- AndroidAssetStudio aplikazioaren ikonoak sortzeko.
- allthefreestock.com Creative Commons Zero lizentziadun irudiak aurkitzeko.

3 Aurrekarien azterketa

Azkenaldian kuantifikatzeko eskumuturreko asko aurkitu daitezke merkatuan. Eskumuturreko horietatik gutxi batzuk dute GPSa eta pausuetan oinarritzen dira gehienak posizioan oinarritu beharrean. Gainera GPS-a duten eskumuturrekoen prezioa handia izaten da. [14]

Hurrengo aplikazioak aurkitu dira proiektu honen arazo berdina konpontzen saiatzen direnak:

3.1 Garmin Livetrack

Garmin-ek, LiveTrack sistema bat [18] garatu du bere Garmin Edge 510 eta 810 txirrindularitza produktuentzako. Garmin-en sistemak hiru atal ditu:

- GPSa duen eskulekuko-konputagailua (Garmin Edge 510 edo 810).
- Android eta iOS sistema eragilearentzako Garmin Connect Mobile aplikazioa.
- Garmin Connect webgune jarraitzailea (irudia 11).



Irudia 11: Garmin Connect webgune jarraitzailea

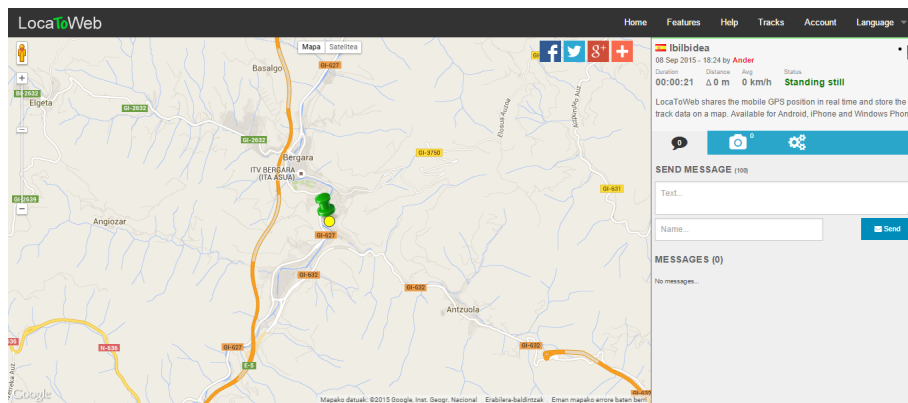
Garmin Fit aplikazioaren bidez, mugikorraren GPS-a erabili dezakegu Garmin-en eskulekuko-komputagailua ordeztuz. Kasu honetan aplikazioa erosi beharko da.

Zerbitzu honek duen desabantaila nagusia bere webgunearen moteltasunean aurkitzen da, mapa aurkezten den bitartean erabiltzailea adierazita agertzen deneko puntua ez baita behar bezalako arintasunarekin mugitzen.

3.2 LocaToWeb

LocaToWeb [26] android, ios eta Windows Phonerako aplikazioak eskaintzen ditu eta jarraipena egiteko webgunea.

Mugikorraren GPSa erabiliz gure posizioa, abiadura... igotzen dira eta argazkiak igotzeko aukera ere ematen du. Ondoren, jarraipena egin daiteke webgunetik (irudia 12).



Irudia 12: LocaToWeb webgune jarraitzailea

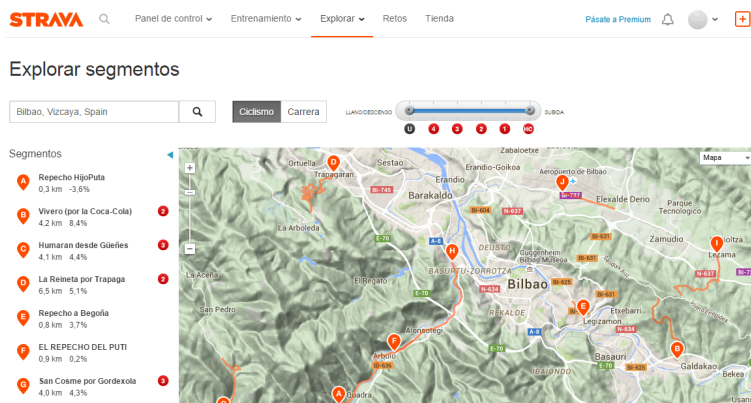
3.3 Strava

Strava [33] webguneak kanpoko kiroltarietarako ibilbideen jarraipena egiteko aukera ematen du. Android eta Iphone mugikorrarako aplikazioaren bidez edo hainbat GPS gailuetatik datuak igotzeko aukera ematen du.

Zerbitzu honek ez du denbora errealean beste batzuek gure posizioaren jarraipena egiteko aukera ematen, hau da, webgunean egindako ibilbidea ikusteko aukera ematen digu baina ibilbidea amaitu ondoren.

Hala ere, aukera asko ditu. Besteak beste, erabiltzaileek ibilbideak partekatu ditzakete eta horrela ezagutzen ez ditugun ibilbide berriak aurkitu aukera ematen du (irudia 13).

Aplikazioa eta funtzio basikoak dohainik erabili daitezkeen arren funtzio batzuk erabiltzeko kuota bat ordaindu behar da.



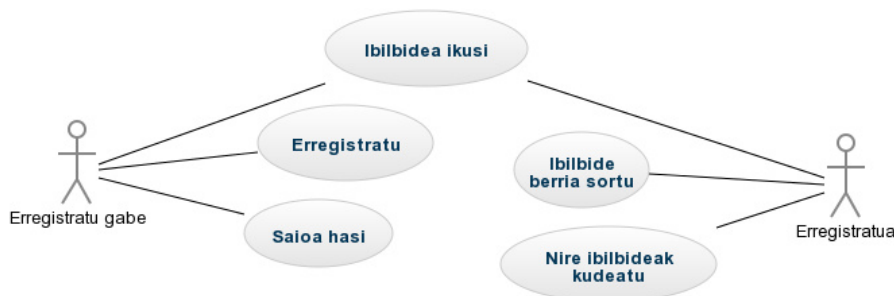
Irudia 13: Strava-ko ibilbide bilaketa

4 Eskakizunen kaptura

Garatuko den aplikazio eta webguneak kanpoko kirolak egiten duten jendeari ez ezik, euren lagun eta familiarrei zuzenduta egongo da ere. Bi erabiltzaile mota desberdinu ditzakegu:

- Alde batetik, kanpoko kirola egiten duen erabiltzailea egongo da. Erabiltzaile honek saioa hasi behar du. Behin saioa hasita, ibilbide berria sortuko du. Aplikazioak, bere posizioa beste datu batzuekin batera erregistratu eta zerbitzarira bidaliko du. Azkenik, bere ibilbidea partekatuko du twitter, whatsapp, email... bidez.
- Beste aldetik, kirolari horren lagun edo familiarrak daude. Erabiltzaile hauek, saioa hasi beharko dute kirolariak ibilbidea pribatu moduan jarri badu bakarrik. Hauek, kirolariak partekatutako ibilbidea ikusiko dute denbora errealean Android aplikaziotik edo webgunetik.

Sisteman hurrengo erabilpen kasuak daude (irudia 14):



Irudia 14: Erabilpen kasuak

4.1 Ibilbidea ikusi

Ibilbide bat ikusteko aukera da. Ibilbide hori erabiltzaile horrena izan daiteke edo beste erabiltzaile batena.

4.2 Erregistratu

Erabiltzaileak kontu bat sortuko du zerbitzuaren funtzionalitate guztiak erabili ahal izateko.

4.3 Saioa hasi

Saioa hasiko du aurretik sortutako kontura sartzeko. Behin saioa hasita erabiltzaile erregistratua izango da.

4.4 Erregistratu gabeko erabiltzailea

Saioa hasi ez duen erabiltzaileak dira. Publiko dauden ibilbideak ikusteko aukera dute url-a badute. Erregistratu eta saioa hasteko aukera dute funtzionalitate gehiago lortzeko.

4.5 Erabiltzaile erregistratua

Saioa hasi duen erabiltzailea. Erabiltzaile honek funtzionalitate gehiago ditu eta ibilbide pribatuak ikusteko aukera du baimenduta badago ibilbide horretan.

4.6 Ibilbide berria sortu

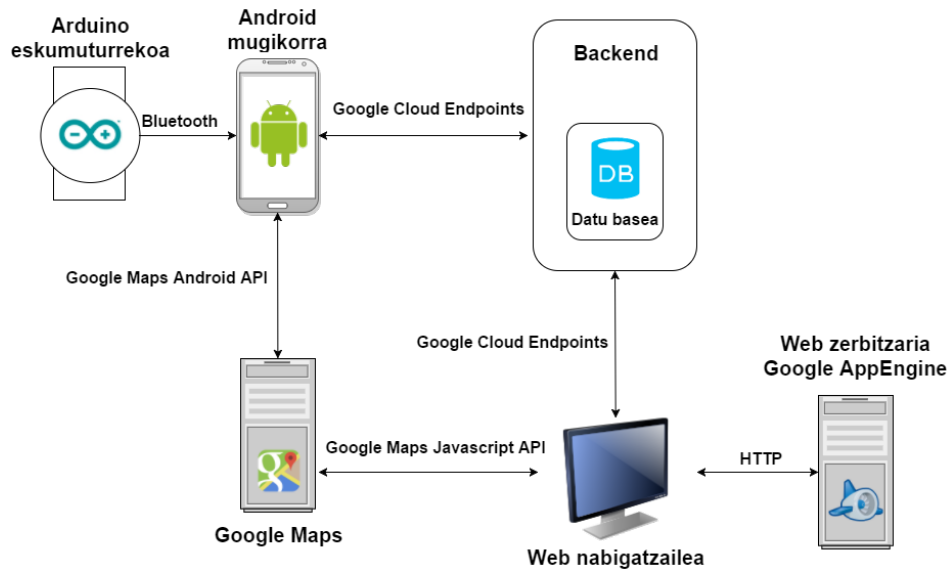
Ibilbide berri bat sortzen da. Posizioaren datuak igoko dira eta beste erabiltzaile batzuekin partekatzeko aukera dago.

4.7 Nire ibilbideak kudeatu

Sortutako ibilbideak kontsultatu ahalko dira. Ezabatu edo pribatua den aldatzeko aukera egongo da. Beste erabiltzaile batzuekin partekatzeko aukera ere egongo da.

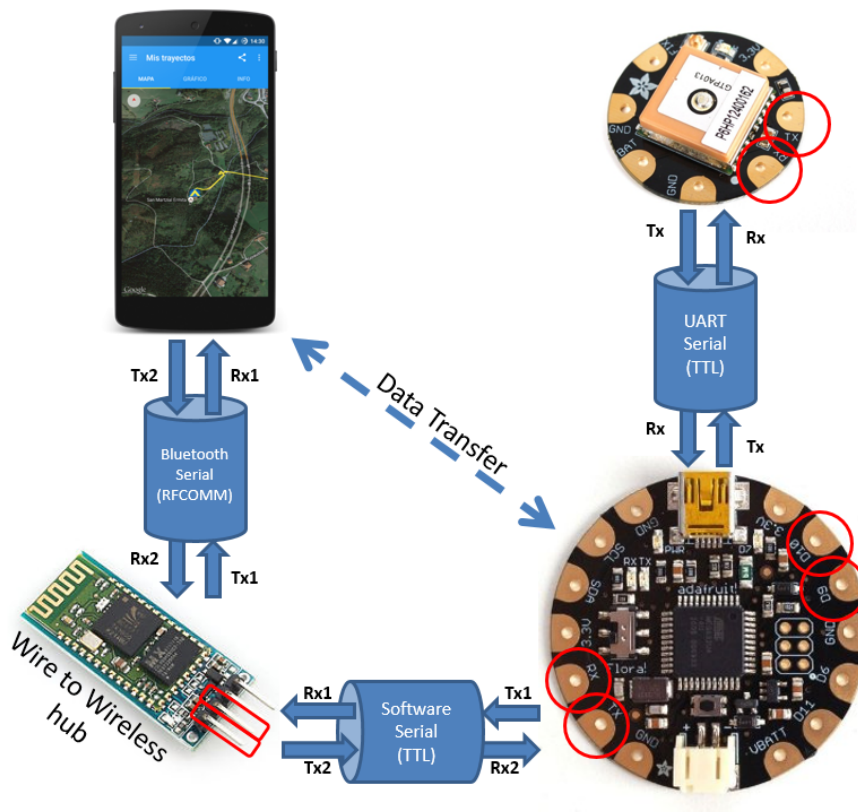
5 Azterketa eta diseinua

Proiektua garatzeko hurrengo banaketa egin da (irudia 15):



Irudia 15: Proiektuaren arkitektura

Alde batetik, eskumuturrekoak Bluetooth 2.1 bidez datuak bidaliko ditu Android mugikorrera (irudia 16). Horretarako SPP (Serial Port Profile) profila erabiliko da. Profil honek bluetooth RFCOMM konexio baten gainean serial konexio bat emulatzeko aukera ematen du. Arduinoren ikuspuntutik SPP erabiltzea oso erreza da bluetooth moduluarekin duen serial komunikazioren kudeaketan oinarritzen delako. Android-en ikuspuntutik, SPP erabiltzea erreza da ere, RFCOMM socket bat sortuta, bere gainean SPP zerbitzua erabili nahi dela adierazi behar da. Zentzu honetan, RFCOMM protokoloa garraio protokolo bat bailitzan ikusi daiteke, TCP protokoaren antzerako funtzioa betetzen duelarik.



Irudia 16: Konexioaren eskema

Bluetooth 4.0 konexioak batera gutxiago erabiltzeko aukera du. Baina proiektua hastean Arduinorentzat Bluetooth 4.0 modulu oso gutxi zeuden eta zegoen euskarria oso txikia zen. Hori kontuan izanda eta aurretik komentatutako hardware aldeko esperientzia txikia, euskarri handia zuen Bluetooth 2.1 erabiltzea erabaki zen.

Android mugikorrak datuak jaso eta Backend-era bidaliko ditu Google Cloud Endpoints bidez. Backend-ak, jasotako datuak, datu basean gordeko ditu. Aldi berean, aplikazioak, datuak mapa batean erakusten ditu ere. Horretarako Google Maps Android API-aren bidez mapa lortuko du.

Beste alde batetik, zerbitzarira konektatu gaitezke nabigatzaile baten bidez HTTP protokoloarekin. Zerbitzariak beharrezko HTML, CSS eta javascript fitxategiak bueltatuko dizkigu. Behin hori izanda, backend-arekin konektatuko da datu baseko datuak lortzeko. Google Maps Javascript API-a erabiliko da mapa kargatzeko.

5.1 Eskumuturrekoa

Aurretik aipatu den moduan, eskumuturrekoa Arduino mikrokontroladorean oinarrituta egongo da. Arduino komunitatea oso handia da eta hardware munduan hasteko aproposa da tutorial askorekin kontaktzen baitu.

Eskumuturrekoaren programazioa egiteko Arduino IDE-a erabiliko da. Programa nahiko sinplea izango da eta bere helburu nagusia GPS datuak Bluetooth bidez bidaltzea izango da, lan handiena mugikorrari utziz. Hau horrela egitea erabaki da Arduino mikrokontroladoreak memoria gutxi eta prozesadore motelak dituzten bitartean mugikorrek gero eta potentzia handiagoa dutelako. Bestalde, Arduinok bete beharreko funtzioak GPS datu errezeptzio, NMEA parser eta mugikorrera bideko berbidalketara mugatzeak eskumuturrekoan batera aurrezteko aukera ematen digu, berbidalketa tartetean Arduinoa “lotan” (kontsumo maila txikian) uzteko aukera posibilitatea aurkezten baita.

Honetaz gain, proiektuaren hasieran Arduinorekin lotutako proiektu txiki batzuk egingo dira prestakuntza lortzeko mikrokontroladorearen egiturari, programazioan eta serie komunikazioan.

5.2 Backend-a

Datu basea eta backend-a oso lotuta egongo dira. Aplikazioak eta webguneak backend-aren metodoak deituko dituzte eta ez dute datu basearekin zuzenean komunikaziorik izango. Beraz, backend-aren helburua datu basea kudeatzea izango da jasotako deien arabera. Honetaz gain erabiltzailearen baimenak kontrolatuko ditu hauek egin dezaketen operazioak baimendu edo ez baimenduz.

Backend-a garatzeko Google Cloud Endpoints [19] erabiltzea erabaki da. Hau erabiltzeak Android Studio-n bertan aplikazioarekin batera garatzeko aukera ematen du eta klaseak eta metodoak partekatu daitezke bien artean.

Honetaz gain, Google kontu batekin OAuth bidez saioa hasteko aukera ematen du [36]. Horrela, ez da pasahitzik gorde behar eta ez dugu erabiltzaileak berez duen Android kontuaz aparte beste kontu berri bat sortzera behartzen. Zentzu honetan, Android aplikazioan gehienetan kontua lehendik sartuta egongo da eta aukeratu besterik ere ez da egin beharko erabiltzaileari saioa hastea erraztuz.

Backend-aren garapena egiteko API bat definitu da. Hauek dira API horrek eskaintzen dituen metodoak:

- `trackBerriaHasi()`. Ibilbide berria sortuko du eta ibilbide horren identifikadorea bueltatuko du. Saioa hasita edukitzea beharrezkoa izango da.
- `wayPointAdd()`. Ibilbide batera puntu bat igoko du bere informazioarekin (latitudea, longitudea, abiadura...). Saioa hasita edukitzea beharrezkoa izango da.
- `getTrack()`. Ibilbide bat lortuko du bere datu guztiekin. Ibilbidea pribatua bada saioa hasita egon beharko da.

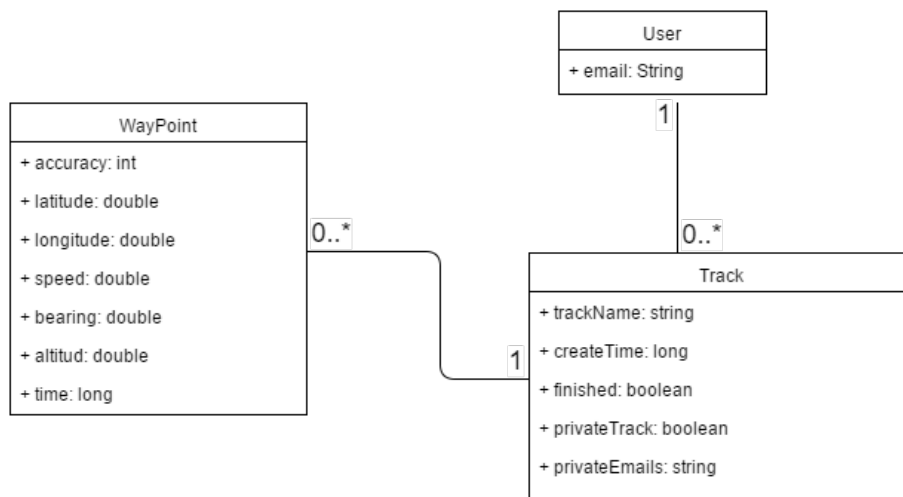
- editTrack(). Ibilbidearen aukerak aldatuko ditu (izena, pribatua...). Saioa hasita egon behar da.
- deleteTrack(). Ibilbidea ezabatuko du. Saioa hasita edukitzea beharrezkoa izango da.
- myTracks(). Erabiltzailearen ibilbide guztiak bueltatuko ditu. Saioa hasita edukitzea beharrezkoa izango da.

Proiektuak aurrera egin ahala metodo berri bat gehitu da, "endTrack()". Hasiera batean, ez da aurreikusi interesgarria izango zela ibilbide bat amaituta zegoen ala ez jakitea. Baina proiektua aurrera joan ahala, datu honen bidez internet erabilera murrizteko aukera ikusi da. Funtzionamendua boolean baten bidez doa. Boolean horrek ibilbidea amaituta dagoen edo ez adierazten du. Amaituta badago, ibilbidea ez denez aldatuko ez dago zertan API-ari aldaketak eskatzen ibili.

5.3 Datu basea

Datu basearentzat, Google Datastore zerbitzua [9] aukeratu da. Zerbitzu hau aurreko backend-arekin integratuta dago eta dohainik da erabilera maila batzuk gainditzen ez diren bitartean. Datu base honen berezitasuna nosql motakoa dela da. Honen hainbat abantaila dakar, hots:

- Aldaketak egiteko arazorik ez dago, eskema dinamikoa baita. Honek malgutasun handia eskaintzen du aplikazioak garatzen eta mantentzen diren bitartean agertzen diren aldaketa eskakizunei.
- Eskalatzea horizontalean egiten denez (zerbitzari gehiago gehituz) datu basea prestatuta dago lan karga handiak jasotzeko zerbitzariak gehituz [30]. Honi esker erabiltzaile kopurua asko handitu daiteke denboran zehar.



Irudia 17: Datu basearen diagrama

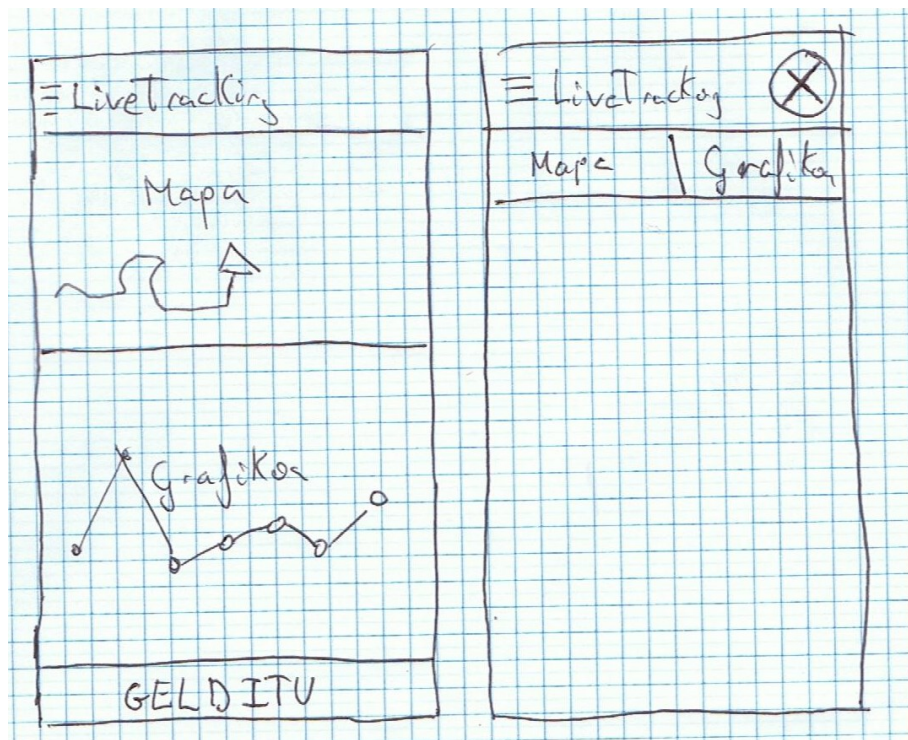
Datu basean (irudia 17) hiru entitate izango ditugu. User, erabiltzailearen email-a gordetzeko erabiliko da. Track bakoitza ibilbide bat izango da eta erabiltzaile batek hainbat Track izan ditzake. WayPoint bakoitzean GPS-ko momentu bateko datuak gordeko ditu. Track bat, hainbat WayPoint-ekin osatuta egongo da.

5.4 Mugikorreko aplikazioa

Mugikorreko aplikazioa Android sistema eragilerako egitea erabaki da. Sistema honek erabiltzaile kopuru izugarria du Europan, eta Espainian oraindik ere handiagoa [4].

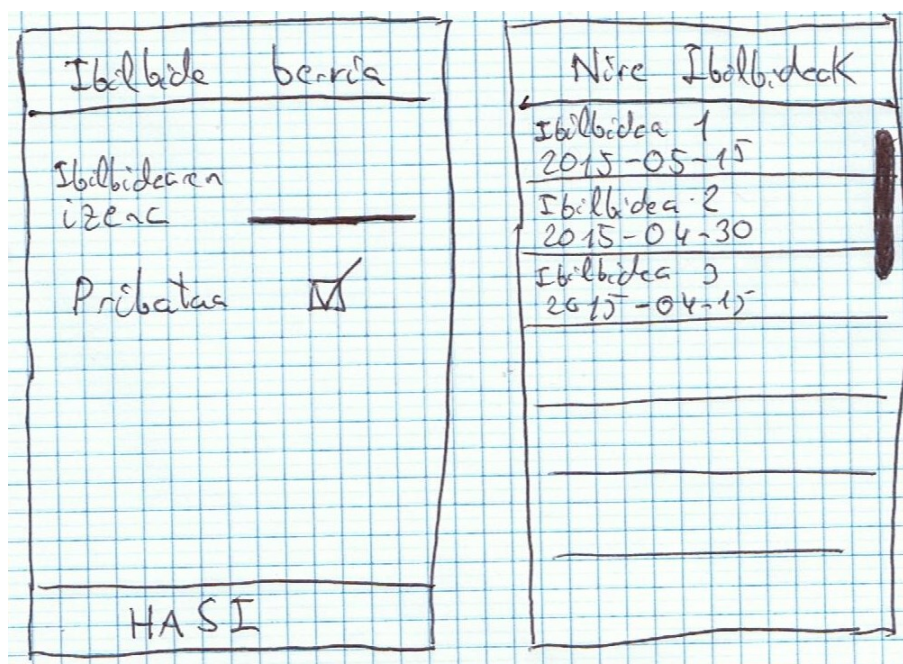
Android-en aplikazioak garatzea ez da hain erreza oinarri finko bat ez badaukagu. Arazo hori konpontzeko Google-ek 2014. urtean Android Studio sortu zuen. Android Studio, Android plataformarentzako garapen ingurune integratua (IDE) da. JetBrains-en IntelliJ IDEA softwarean oinarrituta dago eta kode irekikoa da. Windows, Mac OS X eta GNU/Linux sistema eragileetan lan egiteko prestatuta dago. IDE hau da momentu honetan gomendatua Android aplikazioak garatzeko. Lehen, Eclipse eta plugin batekin egiten zen baina plugin hori ez da mantentzen eta Android Studio-ra pasatzea gomendatzen da. *“If you have been using Eclipse with ADT, be aware that Android Studio is now the official IDE for Android, so you should migrate to Android Studio to receive all the latest IDE updates”* [13]. Beraz, garapena egiteko Android Studio IDE-a erabiliko da.

Material Design [28] gidan oinarrituko da interfazea egiteko. Interfazea erabakitzeke eta garatzean ideia bat izateko paperezko prototipoak egin dira. Zeregin honetarako mockup web-tresna bat erabiltzea aztertu zen [1], baina diseinatu beharreko pantailen sinpletasunak ez zuen trena hauen abantailak usiatzeko justifikaziorik nahikorik aurkeztu. Horrez gain, kasu honetan eskuzko edo paperezko prototipoek lan-malgutasun handia ematen zuten.



Irudia 18: Ibilbidea ikusteko pantailaren prototipoak

Pantaila honetan ibilbide bat ikusteko pantailaren bi prototipo daude (irudia 18). Ezkerrekoan, pantaila berdinean erakusten da mapa eta grafikoa, eskuinaldekoan berriz, mapa eta grafikoa aukeratzeko goiko botoiak erabiltzen dira eta aukeratuta duguna bakarrik ikusten da pantailan. Bigarren hau aukeratu zen edozein pantailatan funtzionatzen duelako. Ezkerrekoak arazoak eman ditzake pantaila txikia bada ez dagoelako biendako espazio nahiko.



Irudia 19: Datu basearen diagrama

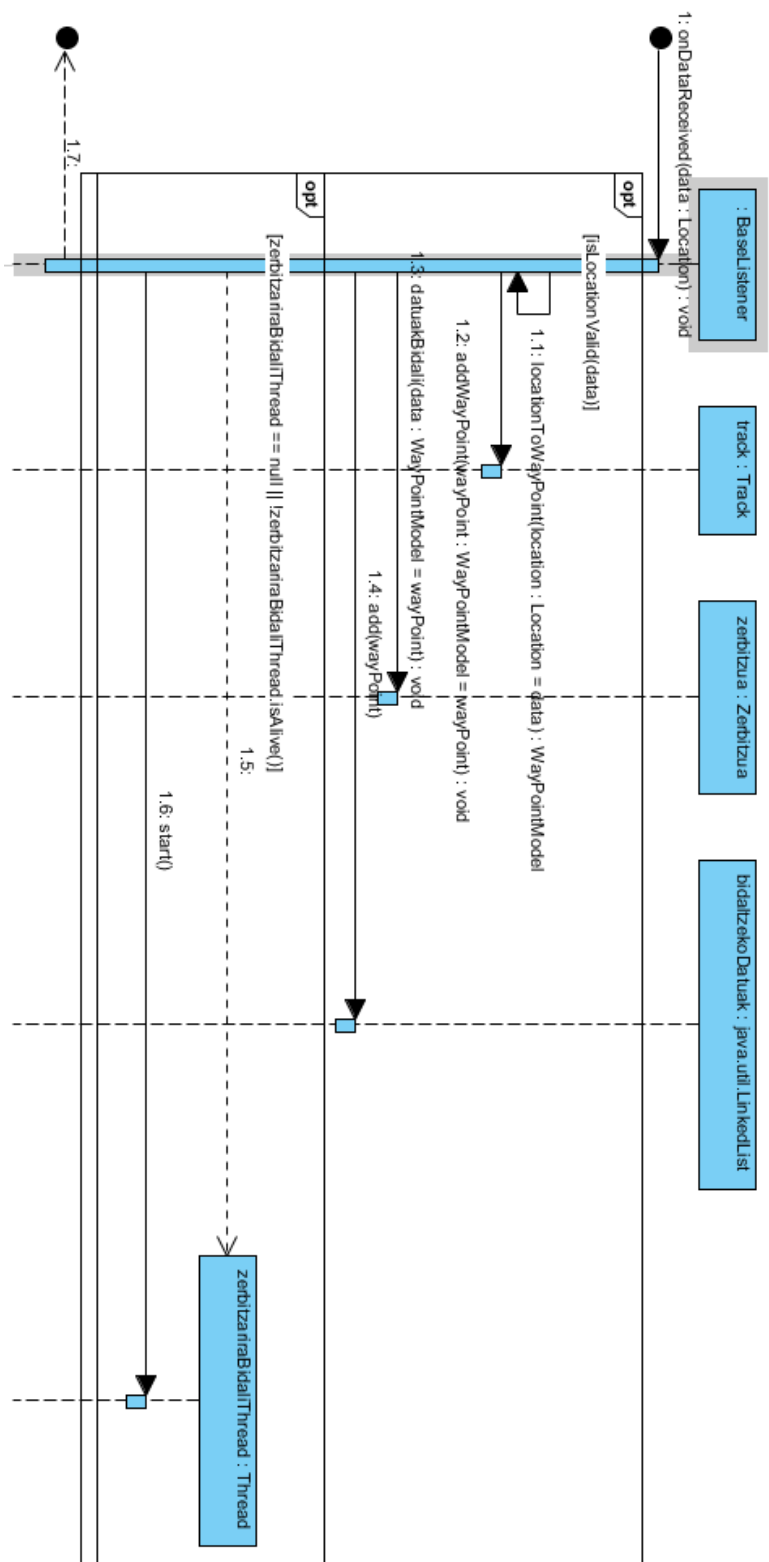
Ezkerraldeko pantailan (irudia 19) ibilbide berria hasteko pantaila dago. Ibilbideari izena jartzen da eta pribatua den edo ez aukeratzeko checkbox bat dago. Hasi botoiak pantailaren zabalera du erabiltzaileak ondo ikusi dezan, akzio garrantzitsua delako.

Eskuinaldeko pantailan (irudia 19) erabiltzailearen ibilbideen zerrenda dago. Zerrendak ibilbidearen izena eta hasi den data eta ordua izango du bilaketa errazteko.

Androiden interfazeak garatzeko bi activity egingo dira. Bat login pantaila izango da. Bestea, login egin ondoren dagoen pantaila izango da. Honek navigation drawer [8] bat izango du eta honen bidez sekzio ezberdinetara nabigatu ahal izango dugu. Nabigazio bakoitza fragment [17] bat izango da eta navigation drawer-eko item bat aukeratzean erakusten ari den fragment-a aldatuko da.

5.4.1 Sekuentzia diagrama

Sekuentzia diagrama honek (irudia 20) GPS datuak jasotzean gertatuko dena deskribatzen du. Kontuan izan sekuentzia diagrama hau azken bertsioa dela eta aldaketak jasan dituela. Aldaketarik handiena zerrenda bat erabiltzea da. Hasiera batean internet konexioa behar zen denbora osoan. Zerrendari esker ezin bada informazioa igo internetik ez dagoelako une horretan, hurrengo datua heltzean horretaz gain igo ezin izan direnak ere igoko dira.



Irudia 20: GPS datuak jasotzen deneko sekuentzia diagrama

Sekuentzian (irudia 20) ikusten den lehenengo metodoa, `onDataReceived`, GPS datuak jasotzean deituko da. Lehenengo, datu horiek duten prezisioa egokia den begiratuko da. Ez bada egokia baztertua izango da. Egokia bada `Location` objektua `WayPointModel` objektura aldatuko da. Aldaketa egin ostean zerbitzura bidaliko da zerbitzura konektatuta egon daitezken fragmentuak informazio hori jaso dezaten. Azkenik, datuak igotzeko haria ez badago martxan berri bat sortuko da eta martxan jarriko da. Honek, zerrendako datuak zerbitzarira igoko ditu.

5.5 Webgunea

Webgunea estandarrak betetzen dituzten nabigatzaile modernoetan jarraipena egiteko balioko du. Mapa batean eskumuturrekoa daraman kirolariaren posizioa denbora errealean ikusteko aukera izango dugu.

Webgunea garatzeko Sublime Text 3 editorea erabili da. Igotzeko backend-ak erabilitako zerbitzu berdina erabiliko da. Hau da, Android Studio bidez igoko dira fitxategiak. Fitxategi hauek Google App Engine zerbitzuan gordetzen dira backend-arekin batera. HTTP eskaera bat jasotzean dagokion fitxategia bidaliko zaio eskaera egin duenari.

Javascript bidez backend-arekin konektatuko da eta mapan datuak kargatuko ditu. Url-ko parametro baten bidez desberdinduko da zein den ibilbidea.

6 Garapena

6.1 Arduino eskumuturrekoa

Eskumuturrekoaren garapena hainbat etapatan banatu da. Kontuan izan behar da ikaslea ez zela arlo honetan aditua eta prestakuntzari dagokionez hutsetik hasi behar izan dela. Horrela, lehenengo etapan, ez da tamaina kontuan hartu hardware-a aukeratzeko orduan, lan egiteko erraztasuna eta funtzionalitate nagusia martxan jartzea helburu nagusiak izan direlarik. Bigarren etapan, eskumuturrekoan sartzeko moduko hardware-a aukeratu da, tamainak garrantzia handia hartuz.

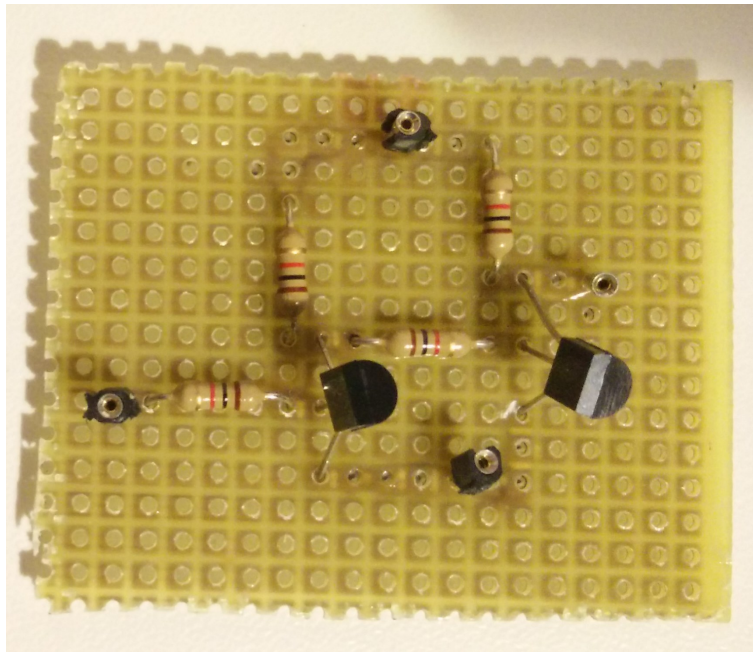
Arduinoz aparte badaude “Internet of Things” arloan erabili daitezkeen beste plataforma batzuk, horien artean Raspberry Pi eta Beaglebone ezagunenak direlarik. Arduino sistema eragilerik gabeko mikrokontroladore baten oinarritzen den bitartean, Raspberry Pi eta Beaglebone-k Linux bertsio berezi bat exekutatzen duen SoC (System-on-Chip) bat dute, bere CPU, GPU eta memoriarekin, honek dituen abantaila guztiekin. Hala ere, Raspberry Pi-a eta Beaglebone-a ez daude GAL honetan darabilgun eskumuturreko bezala sistema txertatu batean sartuak izateko eta modu iraunkorrean lan egiteko aukerarik. Arduinok, ostera, baditu eskumuturrekoaren tamaina eta kontsumora egokitzeko plaka ezberdinak.

6.1.1 Lehenengo prototipoa

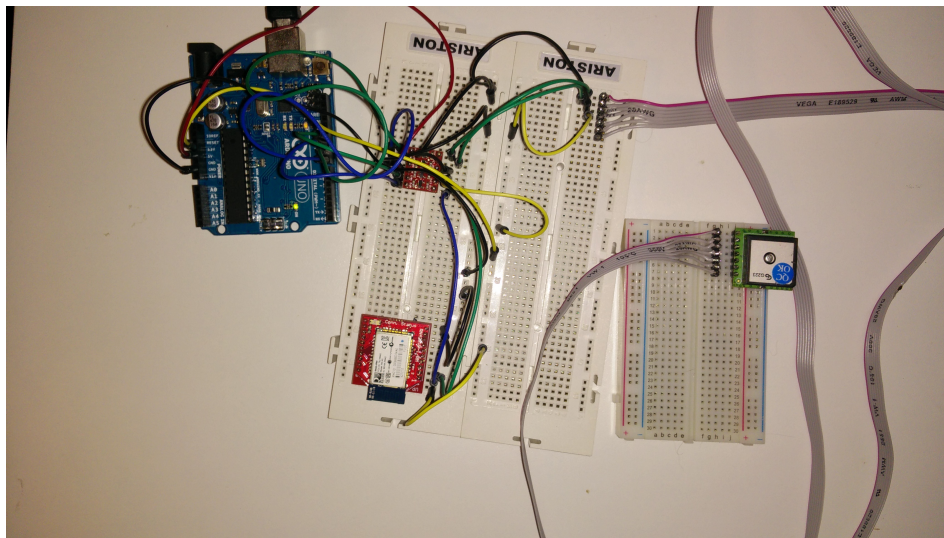
Lehenengo prototipoa (irudia 22) egiteko Arduino Uno mikrokontroladorea aukeratu da elektronikariko arloan esperientzia txikia duten erabiltzaileei elektronikariko eskuragarriagoa egiteko aukerarik gomendagarriena delako bai maila ekonomikoan bai zailtasun mailan [7]. Hasiara baten hau da GAL-aren zuzendariak utzitako materiala:

- Arduino Uno
- RN42XV bluetooth modulua
- GPS-622R GPS modulua

Bluetooth eta GPS moduluak 3.3 voltioarekin funtzionatzen dute. Aldiz, Arduino Uno mikro-kontroladoreak 5 voltioarekin lan egiten du. Ondorioz, tentsio-aldatzaile bat behar da. Horrelako bat erosi aurretik, zuzendariak aukera ona ikusi du lehenengo aldiz estaintzailea erabili ahal izateko eta nire lehenengo soldadurak egin ditut etxe-tentsio-aldatzailea (irudia 21) bat eginez. Etxe-tentsio-aldatzaile hau NOR ate moduan funtzionatzen duten BJT transistore seriean jarriaz lortzen da.

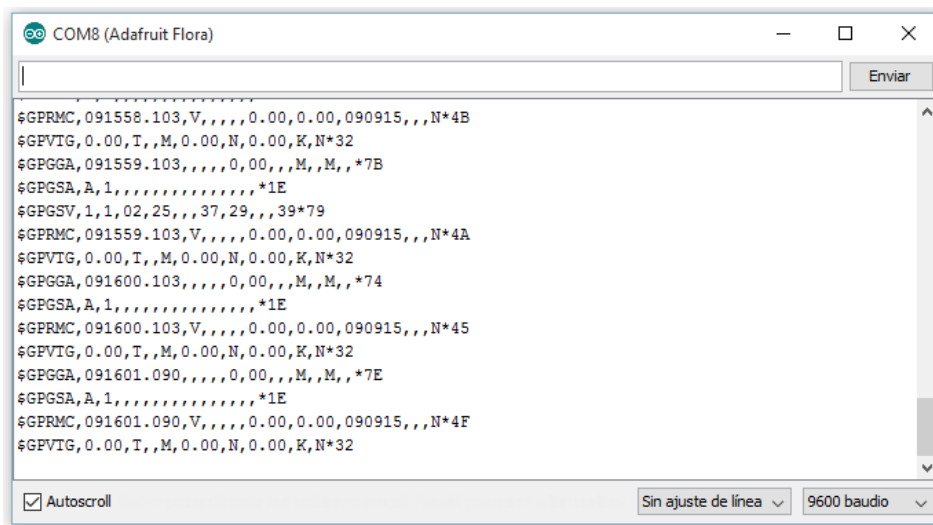


Irudia 21: Boltaje aldatzailea



Irudia 22: Eskumuturrekoaren lehenengo prototipoa

GPS moduluak serial bidez datuak bidaltzen ditu NMEA formatuan. Lehenengo, GPS-aren datuak serial bidez ordenagailura bidali dira eta Arduino IDE-aren serie monitorearen bidez ikusi dira irudia 23.



Irudia 23: GPS-ak bidalitako NMEA sententziak Arduinoren IDE-aren serie monitorean

NMEA sententzia mota ezberdinak daude [10]. Adibide bat hurrengoa izan daiteke:

- \$GPGGA,123519,4807.038,N,01131.000,E,1,08,0.9,545.4,M,46.9,M,,*47

Adibide honetan [10] datuak hurrengoa adierazten dute (taula 2):

Datua	Azalpena
\$	NMEA Sententzia guztien hasieran jartzen da.
GPGGA	Sententzia mota adierazten du. Kasu honetan GGA motakoa da.
123519	Ordua UTC formatuan. 12:35:19
4807.038,N	Latitueda 48 gradu eta 7.038 minutukoa da iparraldera.
01131.000,E	Longitueda 11 gradu eta 31 minutukoa da ekialdera.
1	Seinalearen kalitatea. Kasu honetan sateliteekin konexioa dago.
08	Jarraitzen ari den satelite kopurua. Kasu honetan, 8.
0.9	H-DOP balioa. GPS datuek duten errorea kalkulatzeko balio du [40].
545.4,M	Itsasotik 545,4 metroko altueran dagoela adierazten du.
46.9,M	geoid-aren tamaina WGS84 elipsoidearen gainetik [16].
*47	Aurreko datuen checksum-a. Datuak ondo jaso ditugun jakiteko erabiltzen da.

Taula 2: NMEA adibide sententziaren azalpena

GPS datuak kudeatzeko Arduinorentzako eskuragarri dagoen TinyGPS++ liburutegia [22] erabili da. Honek, NMEA sententziak gestionatzen ditu eta baliozko datuak gordetzen ditu.

Datuak, testu plano bezala bluetooth moduluari bidaltzen zaizkio serial konexioaren bitartez. Taulan (taula 3) dauden datuak bidaltzea erabaki da.

Datua	Azalpena
accuracy-valid	accuracy datua baliozkoa den. “valid” bada datua zuzena da.
accuracy	GPSak ematen digun posizioaren errorea metrotan. Gure benetakoa posizioa datu honen erradioa duen eta eman digun posizioan zentratutako zirkuluaren barruan egongo da.
location-valid	latitude eta longitude datuak baliozkoak diren. “valid” bada datua zuzena da.
latitudea	latitudea gradutan.
longitudea	longitudea gradutan.
speed-valid	abiadura baliozkoa den. “valid” bada datua zuzena da.
abiadura	Eskumuturrekoak daraman abiadura m/s-tan.
bearing-valid	bearing baliozkoa den. “valid” bada datua zuzena da.
bearing	Eskumuturreko mugitzen ari den norabidea gradutan.
altitud-valid	altitua baliozkoa den. “valid” bada datua zuzena da.
altitua	altitua metrotan.

Taula 3: Bluetooth bidez mugikorrari bidaltzen zaizkion datuak

Datuak “,” batez bananduta bidaltzea erabaki da. Hau honela egitean arazo bat sortu da. Ezin da jakin lerroa non hasten eta bukatzen den datuak mugikorrean jasotzerakoan. Hori konpontzeko, lerroaren hasieran “#” bat jarri da eta “@” bat bukaeran. Laburbilduz, datuak bidaltzeko diseinatu den protokoloak ondorengo egitura dauka:

- #accuracy-valid,accuracy,location-valid,latitudea,longitudea,speed-valid,abiadura,bearing-valid, bearing, altitud-valid,altitua@

Adibide bat hurrengoa da:

- #valid,20,valid,43.2632053,-2.9496959,valid,3,invalid,0,invalid,0@

Probak egiterako orduan, Android bidez datuak ondo jasotzen direla jakiteko Bluetooth SPP Tools PRO aplikazioa [24] erabili da. Honen bidez, Arduinora konektatuz jasotako datuak mugikorrean ikusten dira eta ondo bidaltzen ari dela ziurtatu da (irudia 24).

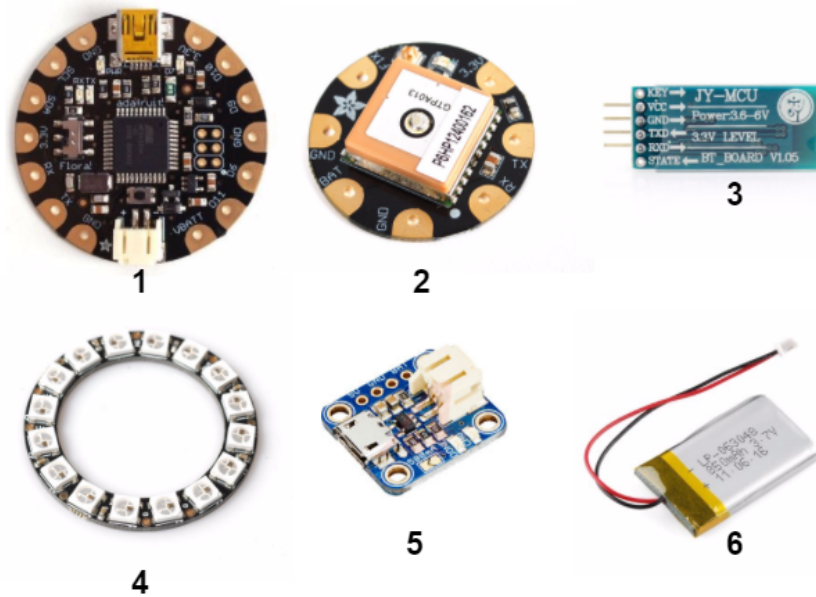
```
98.90@#valid,72,valid,
43.112693786621,-2.414933443069,invalid,-1,invalid,-1,valid,98.90@#valid,
72,valid,43.112686157226,-2.414933443069,invalid,-1,invalid,-1,valid,
98.90@#valid,72,valid,
43.112686157226,-2.414933443069,invalid,-1,invalid,-1,valid,98.90@#valid,
72,valid,43.112678527832,-2.414928436279,invalid,-1,invalid,-1,valid,
98.90@#valid,72,valid,
43.112678527832,-2.414928436279,invalid,-1,invalid,-1,valid,98.90@#valid,
72,valid,43.112674713134,-2.414925098419,invalid,-1,invalid,-1,valid,
98.90@#valid,72,valid,
43.112674713134,-2.414925098419,invalid,-1,invalid,-1,valid,98.90@#valid,
72,valid,43.112659454345,-2.414920091629,invalid,-1,invalid,-1,valid,
98.90@#valid,72,valid,
```

Irudia 24: Bluetooth bidez datuak jasotzen Bluetooth SPP Tools PRO aplikazioan

6.1.2 Bigarren prototipoa

Bigarren prototipoa (irudia 26 eta irudia 27) egiteko hardware aukeraketa berria egin behar da (irudia 25). Oraingoan tamainak garrantzia handia du:

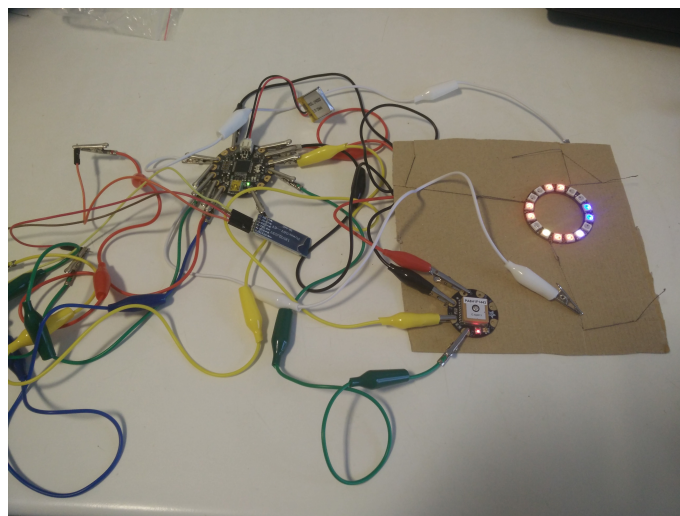
- Adafruit Flora (1)
- Flora Wearable Ultimate GPS Module (2)
- HC-06 Bluetooth (3)
- NeoPixel Ring 16 (4)
- Adafruit MicroUSB - USB LiIon/LiPoly charger (5)
- Lithium Ion Polymer Battery - 3.7v 150mAh (6)



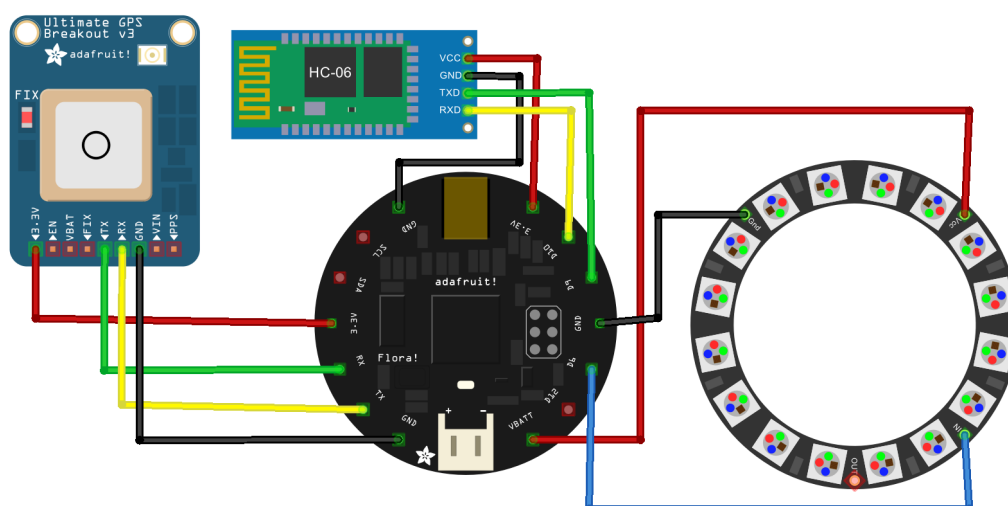
Irudia 25: Bigarren prototipoko materiala

Hardware aldetik nire ezagutza txikia izanik, Adafruit-eko produktuak erabiltzea erabaki da ondorengo arrazoiengatik:

- FLORA, Adafruit enpresaren plataforma elektronikoa wearable-a (janzgarria) da. Adafruit komunitateak zerotik diseinatutako plataforma da prototipo janzgarriak egiterako orduan esperientziarik hobea eskaini ahal izateko. Eskumuturrekoa wearable edo janzgarri bat denez, FLORA proiektu honentzako oso egokia da.
- Plaka oso txikia da (1,75"-ko diametroa eta 4,4g-ko pisua) eta eskumuturrerako egokia den forma borobila dauka. Horrez gain, Adafruit-ek plaka honentzat bereziki prestatutako sensore, GPS eta LED neopixel moduluak dauka. Plakak programaziorako mini-USB ataka dauka, beraz ez da programatzeko kable ezta plaka osagarriarik behar. 3.7V-ko LiPo bateria bat zuzenean konektatzeko JST konektore bat dauka.
- Plaka hau ATmega32U4 mikroprozesadorean oinarritu egiten da eta Arduinorekin bateragarria da. Beraz, lehenengo prototipoan garatutako kode guztia zuzenean berrerabiltzeko aukera ematen digu.
- FLORA-k euskarri zabala aurkezten du izan ere Adafruit web gunean plaka honi buruzko tutorial eta proiektu anitzak aurki daitezke [3], horietako batzuk GPS aplikazioen inguruan.



Irudia 26: Eskumuturreko bigarren prototipoa



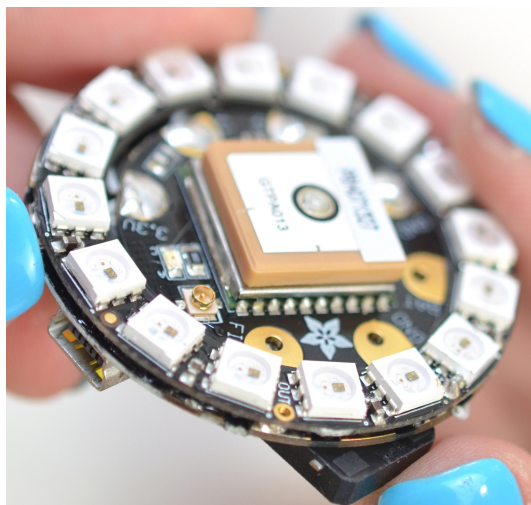
Irudia 27: Prototipoaren konexioen eskema

Eskumuturrekoa funtzionatzen dagoen jakiteko adierazleren bat erabiltzea gomendagarria zela aurreikusi zen. Helburu hori betetzeko led batzuk ipintzea pentsatu zen. Zentzu honetan, Adafruit-eko NeoGeo Watch proiektuan oinarriturik, [6] GAL honetako eskumuturrekoaren argi adierazleak programatu dira. Gainera, irudian (irudia 28) ikus daitezkeen bezala, GPSa, Flora eta neo-pixelak leku txiki batean sartu daitezke, lekua oso ondo aprobeztatuz.

GPS konexioaren informazioa emateaz gain NeoPixel-ak erabiliz ordua ematea ere erabaki da. Arazoa da erloju batean 12 zenbaki daudela eta 16 neopixel

daudela zirkulu bat osatzen. Honen konponbidea 12, 3, 6, eta 9 zenbakiei (ardatzetako erreferentzia diren zenbakiak) dagozkien neopixelak bakarrak izan beharrean bi izatea da.

Aipatutako zenbakiak beti piztuta egongo dira eta GPS konexioa dagoen edo ez adieraziko dute. Konexioa badago berde egongo dira eta konexiorik ez badago gorri. Goiko aldea zein den jakiteko, hamabiei dagozkion bi pixelak kolore gorri edo berde argiago batez egongo dira. Ordua adierazteko argi zuria erabiltzen da, minutuentzako argi urdina eta segundoentzako horia. Hurrengo irudian (irudia 29) ikusten da funtzionamendua.



Irudia 28: NeoGeo watch (<https://learn.adafruit.com/assets/11151>)

GPS konexioaren informazioa emateaz gain ordua ematea ere erabaki da NeoPixel-ak erabiliz. Arazoa da erloju batean 12 zenbaki daudela eta 16 neopixel daudela zirkulu bat osatzen. Honen konponbidea 12, 3, 6, eta 9 zenbakiei dagozkien neopixelak bakarrak izan beharrean bi izatea da.

Aipatutako zenbakiak beti piztuta egongo dira eta GPS konexioa dagoen edo ez esango dute. Konexioa badago berde egongo dira eta konexiorik ez badago gorri. Goiko aldea zein den jakiteko, hamabiei dagozkion bi pixelak gorri edo berde argiago batez egongo dira. Ordua adierazteko argi zuria erabiltzen da, minutuak urdina eta segundoak horia. Hurrengo irudian (irudia 29) ikusten da funtzionamendua.



Irudia 29: NeoPixelak 2:40:35 ordua markatzen

Neopixel horien (irudia 29) bidez GPS seinalerik ez duela jakin daiteke 3, 6 eta 9-ri dagozkion argiak gorritz daudelako. Goiko erreferentzia (12 dagozkion argiak), gorri argiangoan dago (arrosa). Argi zuriak, ordu biak direla adierazten du. Argi urdinak, 40 minutuen inguruan dagoela dio, 12 argi ditugunez ezin da 60 minutuak prezisio guztiz jakin. Argi horiak segundoak adierazten ditu, 35 inguruan kasu honetan.

NeoPixel-ak kontrolatzeko Adafruit-ek eskaintzen duen liburutegia [2] erabili da. Liburutegi honek pixel bakoitzari zenbaki bat ezartzen dio 0-tik 15-era erlojuen orratzen kontratan. Kodea egiterakoan, liburutegiaren erraztasunari esker, argiak kontrolatzea erraza izango zela uste zen baina pixelak erlojuen orratzen kontrara zenbatzeak eta hainbat orduri bi pixel ezartzeak gauzak konplikatu ditu. Lana errazteko hurrengo metodoa egin da:

```

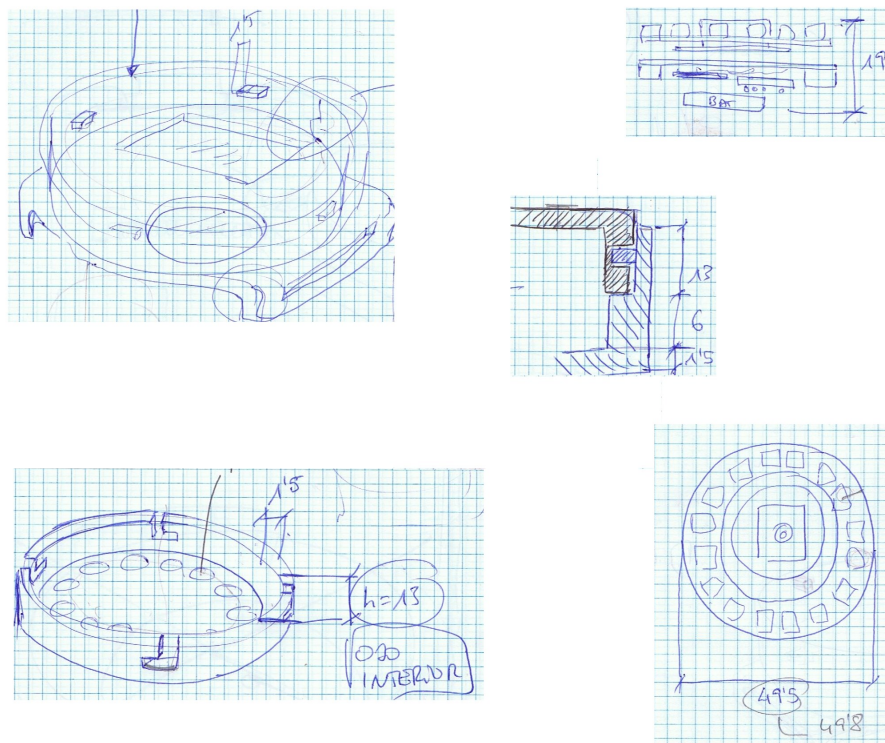
1 // Ordu horri dagokion pixelak piztuko ditu pasatutako
   koloreekin
2 void setPixelsTime(int hour, int R, int G, int B) {
3     int pixel = hour;
4     if (hour > 0) {
5         pixel++;
6         if (hour > 3) {
7             pixel ++;
8             if (hour > 6) {
9                 pixel++;
10                if (hour > 9) {
11                    pixel ++;
12                }
13            }
14        }
15    }
16    pixel = 15 - pixel;
17    setPixelColor(pixel, R, G, B);
18    if (hour % 3 == 0) {
19        setPixelColor(pixel - 1, R, G, B);
20    }
21 }

```

Metodoari ordua pasatu behar zaio. Ordua erloju analogiko baten balioa izango da, 12 izan ezik 0 moduan kontsideratuko delarik. Minutuak edo segundoak izanez gero (minutuak edo segundoak)/5 pasatzeko aukera dago. Beste hiru parametroak nahi den RGB kolorea adierazten dute. Metodoak, kontuan izango du zenbat 2 pixeldun (0, 3, 6, 9 orduak) pasatu diren, indizea batean handituz pasa den bakoitzeko. Ondoren, pixelen indizeak erlojuen kontrako norabidean daudela kontuan hartuta indizea alderantzikatu behar da 15 - indizea eginez. Orain indize egokiarekin ezarritako kolorean piztuko da. Azkenik, bi pixeldun zenbakia bada 3-rekin zatitzearen hondarra 0 izango da eta daukan indizearen aurreko pixela ere piztu beharko da.

6.1.3 Karkasa

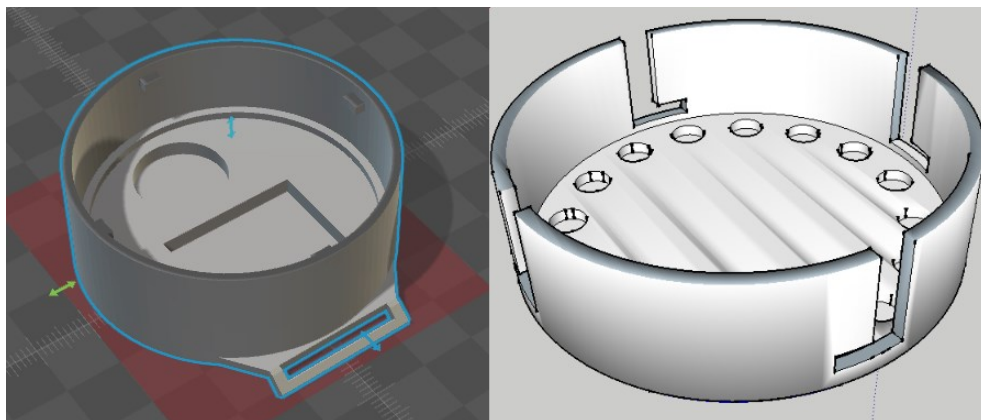
Karkasa egiteko, Sistemen Ingeniaritza eta Automatika saileko tekinkariak (Cesar Pérez) emandako laguntza handia aipatu behar da. Hasiera batean paperean neurriak hartu eta gutxi gora beherako krokis bat marraztu da (irudia 30).



Irudia 30: Paperean egindako karkasaren bozetoa

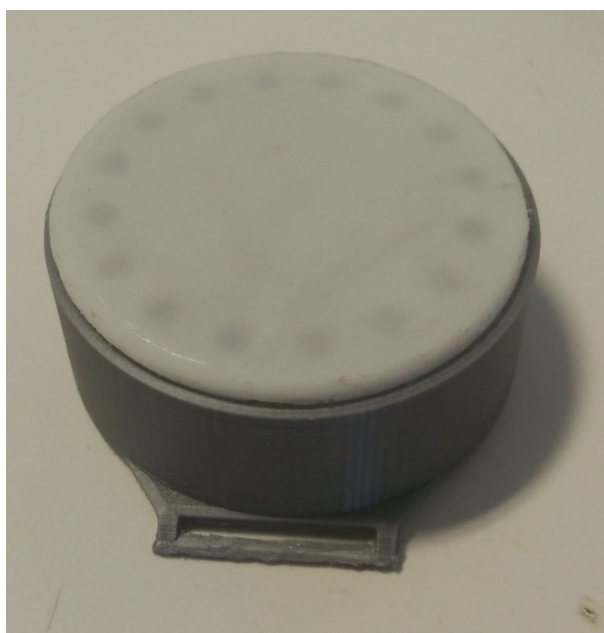
Karkasa bi zatitan egitea erabaki da. Alde batetik, uhala sartzeko pieza nagusia bi zulorekin eta 4 patillarekin. Bigarren zatia tapa bat da. Honek 4 zulo ditu aurretik aipatutako patillak sartu eta lotzeko. Horrez gain, tapan neopixelak doazen tokian doazen tokian lodiera gutxitu egin da argiak ondo ikus daitezen.

Ondoren Google SketchUp programa erabiliz ordenagailuan 3 dimentsiotako modeloa sortu da (irudia 31) papereko datuak jarraituz.



Irudia 31: 3 dimentsiotan egindako karkasaren basea eta tapa

Behin 3D modeloa izanda inprimatua izan da (irudia 32).



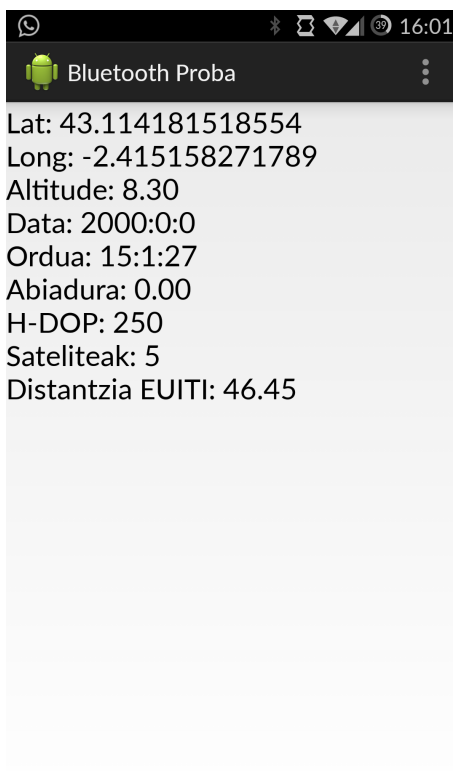
Irudia 32: 3 dimentsiotan inprimatutako karkasa

6.2 Android aplikazioa

Android aplikazioa garatzeko Android Studio softwarea erabili da. Garapen osoan zehar Google-ek garatzaileendako eskaintzen duen orrialdea [11] oso garantzitsua izan da, garatzeko bertako adibideak hartu direlarik.

6.2.1 Lehenengo bertsioa

Lehenengo bertsioan Bluetooth bidez datuak jasotzeaz eta pantailan ipintzeaz bakarrik arduratu da. Horretarako, Bluetooth bidez jasotako datuak TextView batean erakusten dira (irudia 34). Ordua bidaltzea erabaki da segundoa begiratzuz datuak ondo jasotzen ari diren aukera izateko posizioz mugitu ez arren.



Irudia 33: Android aplikazioaren lehenengo bertsioak Bluetooth bidez jasotako datuak erakusten zituen.

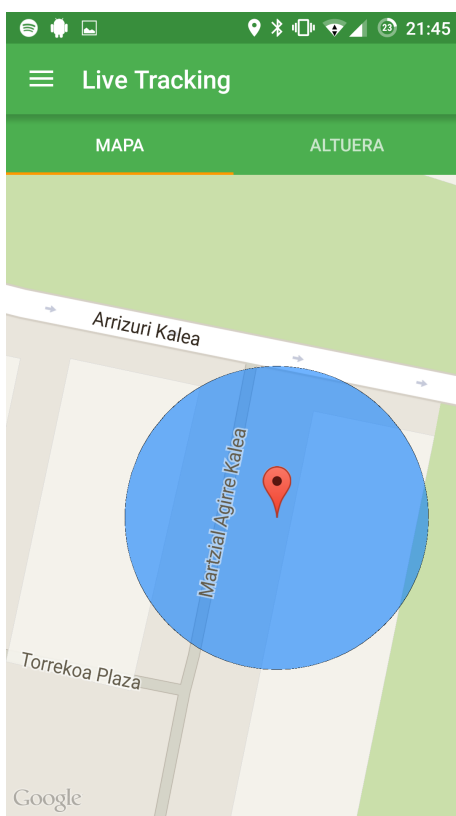
Android-en, Main thread izeneko hari bat dago. Hari hau, pantailako elementuak eguneratzeaz arduratzen da eta lan gutxi egin behar da bertan aplikazioak fluido funtzionatzeko. Hari horretan 5 segundo baino gehiago tardatzen duen edozer egin ezker, Androidek mezu bat erakusten dio erabiltzaileari aplikazioa ixteko aukerarekin.

Aurrekoa kontuan izanda, Bluetooth konexioa hari berri batean egin behar da Main thread-a blokeatu ez dadin.

6.2.2 Bigarren bertsioa

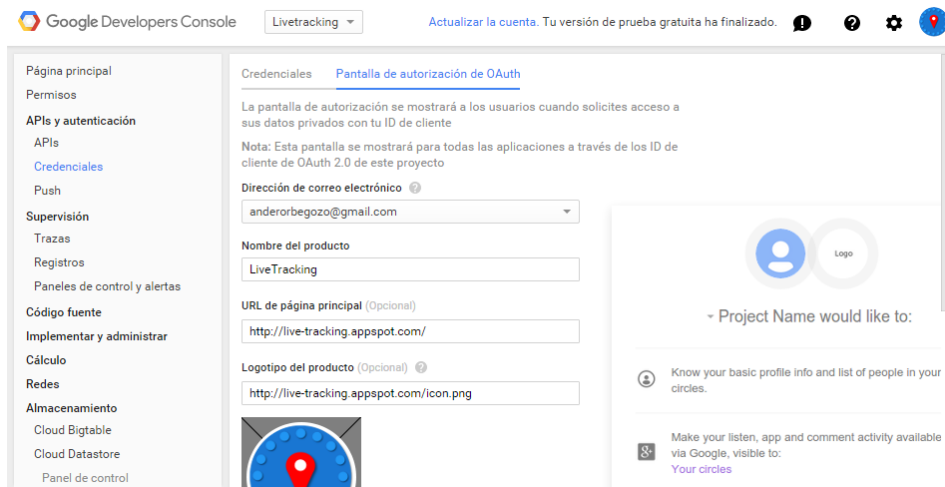
Bigarren bertsio honen helburua zerbitzariarekin konexioa egitea da, aurreko bertsioan Bluetooth bidez lortutako datuak zerbitzarira igoz. Honela, aplikazioaren basea lortzen da konexio guztiak eginez.

Gauzak simple izaten jarraitzeko, alde batetik konexioan funtzionamendu egokia eta zuzena bermatzeko erabiltzaile ezberdinak egongo direla ez zen kontuan hartu; beste alde batetik, puntu bakar bat gordetzen zen, hau da, posizio berria lortzean aurrekoa gainidazten da erabiltzaile guztientzat puntua berdina delarik (irudia 34).



Irudia 34: Android aplikazioaren bigarren bertsioa posizio puntu bakarrarekin

Backend-arekin konexioa egiteko Google-ek eskainitako jarraibideak [38] erabili dira. Proiektua, Google Developer Console orrialdean sortu da eta beharrezko identifikadoreak hortik atera dira (irudia 35). Google Maps erabiltzeko ere orrialde berean aktibatu da.



Irudia 35: Google Developer Console. Beharrezkoak diren ID-ak ateratzen dira

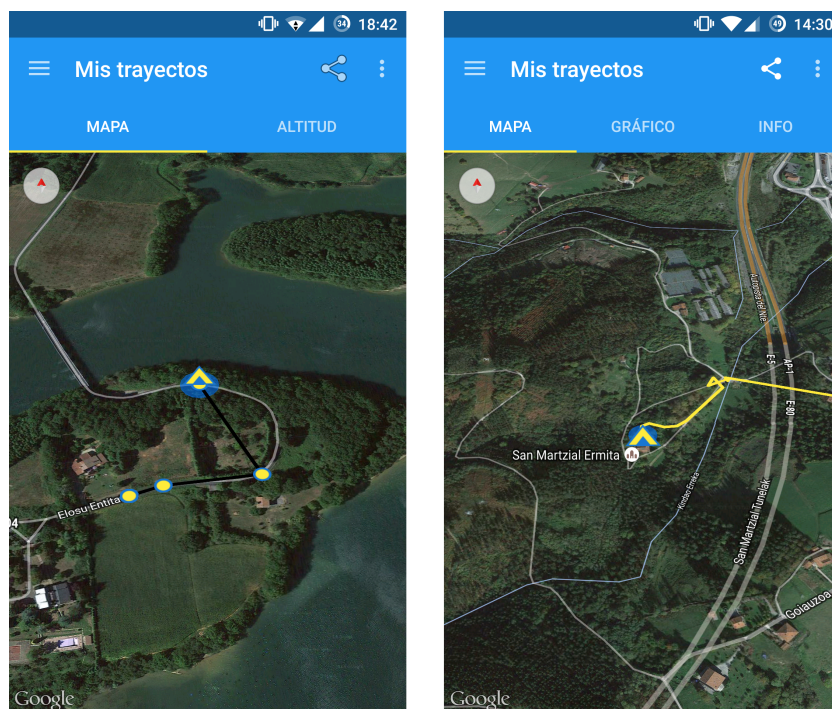
Aplikazioak datu basera informazioa igo dezan lortzen da baina bakarrik aplikazioa pantailan bistaratzen den bitartean. Mugikorra blokeatuz gero edo beste aplikazio batera mugituz gero datuen igoera gelditzen da. Hau konpontzeko, Androideko zerbitzu bat [32] sortu da atzeko planoan funtzionatzen duena. Horrela, zerbitzuak martxan jarraitzen du memoria kritikoki baxua ez den bitartean.

6.2.3 Hirugarren bertsioa

Hirugarren bertsio honetan erabiltzaileen saioa kontuan hartu da eta funtzionalitate gehiago sartu dira. Kasu honetan Bluetooth bidez jasotako datu guztiak taula 3 zerbitzarira bidaltzen hasi da.

Login pantaila bat sortu da. Login pantaila honetan gure kontua aukeratu da Android-ek gordeta dituen kontuen artean. Berri bat gehitu edo sortzeko aukera dago. Behin kontua aukeratuta SharedPreferences bidez gordeko da. Horrela, kontua ez da berriz eskatuko saioa itxi arte.

Mapan ibilbidea zuzenen bidez erakusten da. Hasiera batean zuzenak beltzak ziren baina kolorea aldatu da kontrastea hobetzeko. Orain horiak izanda kasu gehienetan askoz hobeto ikusten dira (irudia 36).



Irudia 36: Ibilbidearen kolorearen konparaketa.

Horretaz gain, puntuak kentzea erabaki da. Ezkerreko irudian ikus daiteke (irudia 36) hainbat puntu daudela GPS posizio bat jaso den bakoitzean. Puntuak oso gertu daudenean, pantailan ez dira ondo ikusten (irudia 37). Horrez gain, saio luze batean puntu asko gorde eta grafikoki adierazi behar direnez, eskumuturrekoa adierazten duen puntuaren eguneratzea moteldu egiten da. Bi arrazoiak kontuan izanda azkenengo puntua bakarrik adieraztea erabaki da. Erabaki honek aplikazioaren errendimendua modu erabakigarrian handitu egiten du, erabiltzaile batzuek hau nabarmenki azpimarratu dutelarik.



Irudia 37: Mapan puntuak gertu badaude ez da ondo ikusten

6.2.4 GPS iturriak

Proiektua aurrera doan bitartean aplikazioak kasu guztietan ondo funtzionatzen zuela ziurtatu nahi da. Momentuz GPS-a leiho ondoan bakarrik konprobatu da Arduino eskumuturrekoaren muntaia osoa eginda ez dagoelako eta kable guztiekin kaletik ibiltzea ez delako aukerarik onena. Muntaia bukaerarako uztea pentsatuta dago behin denak ondo funtzionatzen duenean.

Hori ikusita, mugikorraren GPS-a erabiltzeko aukera garatzea erabaki da. Honi esker, aplikazioaren funtzionamendua egiaztatu daiteke eskumuturrekoa bukatu aurretik eta gainera aplikazioaren publikoa handitzen da eskumuturrekoa ez dutenek ere erabiltzeko aukera izanez.

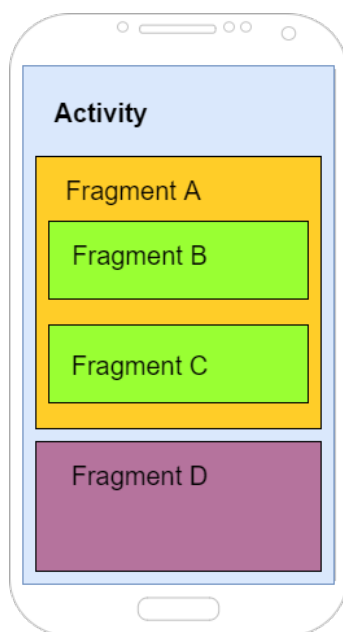
Klase abstraktu berri bat sortu da hau egiteko: [Base Listener](#)

[Klase honetatik](#) sortu dira bai [mugikorreko GPS-tik](#) bai [Bluetooth-etik](#) datuak jasotzen dituzten klaseak. Klase horiek Location motako objektua sortu behar dute jasotako datuekin eta euren aitaren (BaseListener-en) onDataReceived metodoari deitu behar diote Location objektua pasatuz. Honi esker desberdintasun bakarrak bi seme klase horietan dago, hortik aurrera berdina da datuak nondik datozen, kode berdina erabiliko da.

Aplikazioak funtzionatzeko ibilbide berria hastean eta amaitzean beharrezkoa da Internet konexioa izatea. Baina tartean konexioa joaten bada jasotako informazioa bidaltzeko datuak zerrendan gordeta gelditzen dira. Datu berri bat jasotzean, zerrendan dauden datu guztiak igotzen saiatuko da. Zerbitzariak, Way-PointModel motako datua espero duenez locationToWayPoint metodoa dago aldaketa hori egiteko.

Kontuan izan location data baliozkoa den jakiteko metodoaren (isLocationValid-
ren) arabera datu batzuk baztertuko direla. Kasu honetan, prezisioa 70 metro baino txikiagoa izan behar da datua onartzeko.

6.2.5 Listener-ak



Irudia 38: Androideko activity eta fragmenten eskema

Androideko pantaila bat activity bat da (irudia 38). Activity horrek fragmentak izan ditzake (A eta D fragmentuak). Fragment bakoitzak interfazearen zati bat izango da eta horren kontrola eramango du. Gainera, fragment batek bere seme fragmentak ere izan ditzake (B eta C fragmentuak A fragmentuaren semeak dira) [17].

Fragmentuak bere aita fragmentua eta activity-aren erreferentzia bat dute defektuz. Hala ere batzuetan komenigarria da aita edo activity-tik seme fragmentu bateko metodoren bati deitzea. Kasu honetan ez dago defektuz inongo erreferentziarik. Fragmentuak setArguments metodoa dute datua primitiboak (boolean, int, float...) pasatu ahal izateko. Ondorioz, beste fragmentu baten erreferentzia pasatzeko metodo bat sortu daiteke edo eraikitzailea aldatu. Arazoa dago baina gailua biratzean Android sistemak fragmentuak desegin eta

berriz sortzen dituela eta ez duela sortu dugun metodo hori edo eraikitzaile hori erabiliko. Ondorioz, erreferentzia null izango da.

Komunikazioa egiteko modu bat aurkitu da aitaren erreferentzia erabiliz. Fragment-ak aitari edo activity-ari bere erreferentzia pasatzen dio sortzean eta desegitean kendu. Horrela fragmentu aitak edo activity-ak beste fragmentuen erreferentziak ditu hauen metodoak deitu ahal izateko.

Fragmentu batek ibilbide bat kargatu nahi duenean [WayPointReceiver interfazea](#) implementatuko du. Adibidez, mapa eta grafikoaren fragmentuak implementatzen dute.

[DataListener-a](#), datuak jasotzen duen fragmentuan da (zerbitzutik martxan dagoen ibilbidea edo internetetik ikusten ari den ibilbidea). Fragmentu honek hainbat fragmentu seme edukiko ditu [WayPointReceiver](#) implementatzen dutenak (mapa eta grafiko fragmentuak). Seme horiek `addWayPointReceiver` eta `removeWayPointReceiver` bidez datuak jaso nahi duten edo ez adieraziko dute. [DataListener-ak](#) zerrenda batean gordeko ditu `add` egiten duenean eta `remove` egitean kendu egingo ditu zerrendatik. Listener hau [WayPointReceiver-etik](#) sortu denez honen metodoak ditu. Metodo horietan datuak jasotzean aipatutako zerrendan dauden semeei deitu behar die datuak pasatzeko. Honen bidez fragmentuen arteko komunikazioa egitea lortu da.

6.2.6 Abiadura eta altueraren grafikoa

Grafikoa erakusteko `MPAndroidChart` [23] liburutegia erabili da. Liburutegi honek grafiko berdinean bi datu ezberdin erakusteko aukera ematen du eta beste askok ez bezala zoom egiteko aukera dago datuak hobeto ikusteko.

Liburutegi hau erabiltzean arazo batekin aurkitu da. X indizeko balioak `String` zerrenda moduan pasatu behar dira eta ezin da `int` balioak zuzenean erabili. Kasu honetan datuen arteko tartea ez da beti berdina. Adibidez, datuak 10m, 20m eta 40m-tan izanez gero, grafikoan 10-20 distantzia eta 20-40 distantzia berdina da. Hasiera batean, hau konpontzeko zerrenda oso sortzea erabaki da metroen balio guztiak erabiliz eta zerrendari balioak gehitzen joatea erabiltzaileak metroak egiten dituen bitartean. Adibidez, erabiltzaileak 10 kilometro egin baditu, zerrendak 100000 string ditu ("1m", "2m"..., "10000m"). Arazorik ez dago balioak txikiak direnean baina ikus daiteke balioa oso handia egiten denean horrenbeste string memorian izatea ez dela egokia eta aplikazioa ixten hasten da.

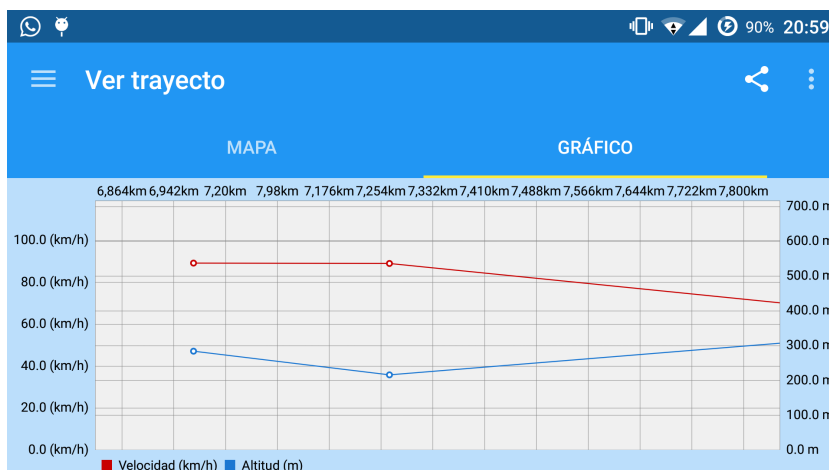
Memorian horrenbeste string ez izateko, zerrenda simulatzea erabaki da. Honi esker string-ak behar direnean sortzen dira. Grafikoak balio gutxi batzuk erakusten ditu (8-10 artean). Beraz, memorian horiek bakarrik mantentzen dira. Efektu hau lortzeko klase hau sortu da:


```

1 public class MaxList implements List<String> {
2     @Override
3     public String get(int i) {
4         if (i < 1000) {
5             return String.valueOf(i) + "m";
6         } else {
7             return String.valueOf(i / 1000) + "," + String.
                valueOf((i / 10) % 100) + "km";
8         }
9     }
10 }

```

Egindako aldaketek funtzionatzen duten arren, beste arazo bat sortu dute. Balioak bakarrik puntu batzuetan daude eta liburutegiak puntu guztietan balioa izatea espero du. Ondorioz, zoom egitean pantailatik kanpo gelditzen diren puntuen arteko marrak desagertzen dira (irudia 39).



Irudia 39: Ezkerraldeko marra desagertuta

Errore honen soluzio bat aurkitu da github-en [12]. Soluzio honekin, liburutegiak puntu guztiak kontuan hartzea egiten du. Metodo honek funtzionatzen du baina ez da optimoa pantailan ez dauden puntu guztiak kontuan hartzen baitu. Base hori hartuta, [beste soluzio bat egin da](#). Zerrenda batzuetan datuen indizeak gorde eta honela kalkulatu zein indize tarte minimo hartu behar den marrak ondo marrazteko efizientea izaten jarraituz.

6.2.7 GPX

GPX formatua GPS datuak gordetzeko erabiltzen da [15]. Formatu hau, XML testu bat da eta OpenStreetMap-eko adibidea [21] jarraituz garatu da. Aukera honi esker erabiltzaileak aplikaziotik ibilbideak esportatu dezakete gus-tuko duten programa batean erabili ahal izateko. Adibidez Google Earth edo RubiTrack programak erabili daitezke ibilbidea bistaratzeko.

6.2.8 Distribuzioa

Aplikazioa PlayStore eta Amazon AppStore dendetara igo da dohainik deskargatua izan ahal izateko:

- <https://play.google.com/store/apps/details?id=com.aorbegozo005.livetrack>
- <http://www.amazon.es/dp/B01498W3ZS>

Dendetara igotzeko email bat sortu da dendetan publiko dagoena: livetrackingapp@gmail.com. Aplikazioaren kapturak, deskripzioa eta bideo txiki bat egin da dendetako datuak betetzeko.

PlayStore dendara igotzeko garatzaile kontu bat sortu behar da. Horretarako 25\$ ordaindu behar izan dira. Beste dendetan dohainik da.

Behin garatzaile kontua izanda, hainbat datu ikusteko aukera ematen du, besteak beste, erabiltzaileek jaso duten erroreak. Erabiltzaileek aplikazioa erabiltzen ari diren bitartean erroreren bat gertatzen bada igotzeko aukera dute. Honi esker, pare bat errore detektatu eta konpondu dira errore mota eta kodeko zein lerrotan gertatu den informazioa izanez.

Aplikazioa aurretik aipatutako dendetara bakarrik igo den arren beste denda batzuetan agertu dela ikusi da. Hauek dira aurkitu diren dendak baina posible da gehiagotan egotea:

- <http://www.appjenny.com/Android/App/1918310/com.aorbegozo005.livetrack>
- <http://www.appbrain.com/app/live-tracking/com.aorbegozo005.livetrack>
- http://www.similarplay.com/live_tracking/live_tracking/apps/com.aorbegozo005.livetrack
- <http://apk.1mobile.com/m/live-tracking-apk.html>
- <http://applion.jp/android/app/com.aorbegozo005.livetrack/>
- <http://www.mobomarket.club/app/com.aorbegozo005.livetrack.html>

6.3 Backend-a eta datu basea

Webguneak eta aplikazioak deiak egiten dute modulu honetara. Garatzeko Android Studio erabili da. Honek abantaila handi bat dakar, android aplikazioa eta backend-a editore berdinean garatzea eta ondorioz aplikaziotik backend-ak dituen metodoak ikustea.

Datastore zerbitzuak eskema dinamikoa du. Ondorioz ez da datu basearen eskema sortzeko pausurik egon.

6.4 Webgunea

Webgunearen garapena hiru fasetan egin da. Lehenengo, mapen erabileran trebatu naiz, bigarrean, backend-arekin konexioa izan da garrantzizkoa eta azkenik, hirugarren fasean, erroreak zuzendu, interfazea eta itxura hobetu eta informazio orrialde batzuk sortu dira.

6.4.1 Lehenengo bertsioa

Lehenengo bertsioan, Google Maps Javascript V3 API-a [20] erabiltzen ikasi da. Orrialdeak, mapa bat erakusten du eta marker bat ipintzen da mapan. Ondoren ibilbide bat marraztu da. Posizio datuak kodean daude sartuta momentu honetan ez dagoelako backend eta datu basearekin konexiorik.

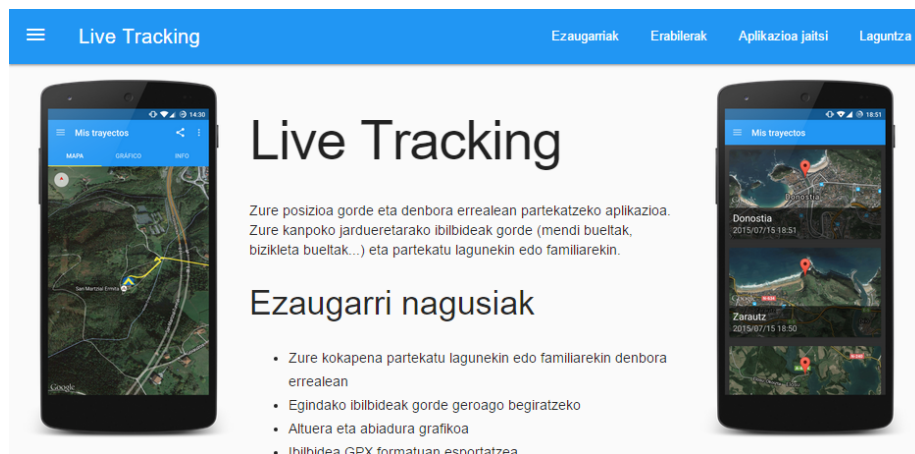
6.4.2 Bigarren bertsioa

Webgunea Html, CSS eta javascript bidez egin da. Orrialde bakarra egin da bertsio honetan, ibilbide bat ikusteko orrialdea. Interfazeari garrantzia handirik eman gabe mapak ia pantaila oso hartzen du eta botoi bat dago saioa hasteko.

Aurreko bertsioan ikasitakoarekin mapa gehitu da eta backend-arekin konexioa egin da datuak lortzeko. Hori javascript bidez egin da Google-ek eskainitako adibideak eta instrukzioak [37] jarraituz.

6.4.3 Hirugarren bertsioa

Hirugarren bertsioan webgune osoa egin da. Webgunearen helbidea hau da: <http://live-tracking.appspot.com/>. Orrialde guztiak egiten hasi da. Itxura hobea lortzeko Material Design Lite liburutegia [29] erabili da. Honek webgunearen itxura hobetzeko aukera ematen du modu azkar batean. Honetaz gain, Android-eko aplikazioa Material Design-en oinarrituta dagoenez erabiltzailearentzat antzeko interfazea izango du bai aplikazioan bai webgunean erabilerraztasuna hobetuz (irudia 40).



Irudia 40: Webgunearen itxura Material Design Lite liburutegia erabiltuta

Material Design Lite erabiltzean ikusi da orrialde gehienek kodea partekatzen dutela. Goiko menuaren kodea eta beheko informazioa orrialde guztietan berdina da. Kodea ez errepikatzeko, HTML-tik JSP-ra aldatzea erabaki da. Honi esker berrirabiltzen den kodea fitxategi batean jarri eta kode hori erabiltzen duten orrialde guztietan include bat besterik ez da egin behar:

```
1 <%@include file="./head.jsp" %>
```

Hasiera batean hau gauzatzeak lana izan arren, beranduago asko eskertu nuen menuko elementu bat aldatzen nuen bakoitzean ez nuelako fitxategi guztietan aldatu beharrik.

6.4.4 Twitter Cards

Twitter Cards [35] bidez webguneko url-ak eta aplikazioa konektatzea ahalbidetzen du. Hori lortzeko orrialdeetan meta etiketa batzuk ipintzen dira:

```
1 <meta name="twitter:card" content="app">  
2 <meta name="twitter:site" content="@livetrackingapp">  
3 <meta name="twitter:app:id:googleplay" content="com.  
  aorbegozo005.livetrack">
```

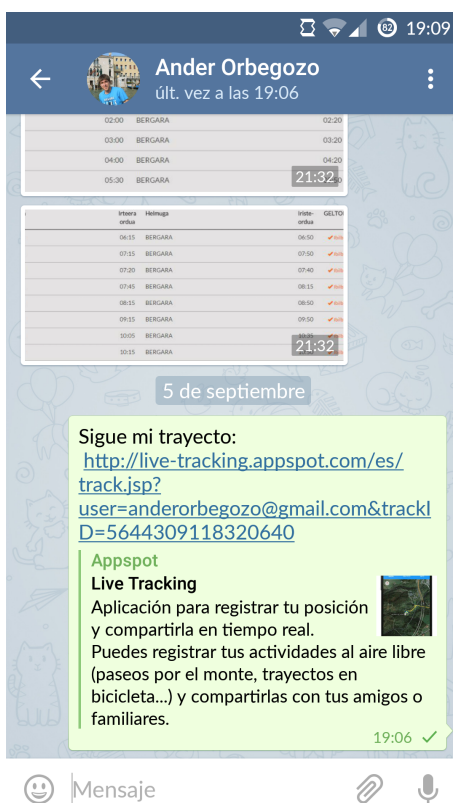
Meta etiketa hauek PlayStore-eko identifikadorea zein den esaten dute. Horri esker tweet batean webguneko url bat badago PlayStoretik deskripzioa, balorazioak... lortzen ditu eta tweet-arekin batera erakusten du. Gainera, android-eko twitterretik ikusten badugu aplikazioa instalatzeko botoi bat ere agertuko da (irudia 41).



Irudia 41: Twitter card erabilia aplikazioa deskargatzeko aukera agertzen da

6.4.5 Open Graph Protocol

Protokolo honek [34] Twitter Card-en antzera meta etiketa batzuen bitartez informazioa emateko aukera ematen du. Honen bidez url-a Facebook, Telegram, Google Plus... bidez partekatzean irudi bat eta deskripzio bat agertzen da gure mezuarekin batera (irudia 42).



Irudia 42: Open Graph adibidea Telegram aplikazioan

6.5 Promozioa

Aplikazioa dendetan igota zegoela hainbat mezu jaso dira aplikazioa promozionatzeko aukerak eskainiz ordainduz gero. Hala ere, aplikazioaren promozioa ordaindu gabe egitea erabaki da.

Nire lagun eta familiarrei deskargatzeko eskatu diet eta euren iritzia emateko eskatu. Honetaz gain, Twitter bidez zabaltzen saiatu gara.

Hainbat blog-ekin kontaktuan jarri nintzen. Blog batzuk androideko aplikazioak argitaratzen dituzte eta beste batzuk kanpo kirolen inguruan idazten dute. Gehienek ez dute erantzun baina batzuek Twitter-en zabaldu dute:

- <https://twitter.com/rodadas/status/628808697209352192> (irudia 41)

- <https://twitter.com/androidappsrevw/status/631187708430446592>

7 Egiaztapena eta ebaluazioa

Egiaztapena zati ezberdinetan banatu da proiektuaren banaketaren ondorioz.

7.1 Backend-a eta datu basea

Datu baseko datuak ikusteko [Google-ek eskaintzen duen kontsola](#) erabili da. Honen bidez datuak editatu eta ikusteko aukera izan dut eta nire aplikazioak datuak ondo igotzen dituela ziurtatu.

Backend-a ondo funtzionatzen duela egiaztatzeke aurreko orrialdeaz gain [Googlek eskaintzen duen api explorer](#) zerbitzua erabili nuen. Honi esker, nire backend-aren metodo guztiak probatzeko aukera ematen du nahi den parametroekin. Honetaz gain erabiltzailea logeatuta dagoen edo ez aukeratzeko aukera du. Honi esker, ibilbide pribatuak direla egiaztatu nuen eta ez dela posible hauek logeatu gabe edo baimendutako kontuan logeatu gabe eskuratzea.

7.2 Webgunea

Webgunearen egiaztapena egiteko W3C organizazioak eskaintzen duen webgunea [27] erabili da. Webgune honen bidez nire webguneko orrialdeak HTML 5 estandarra betetzen dutela ziurtatu da eta gaizki itxitako hainbat etiketa aurkitu nituen.

7.3 Erabiltzaileekin probak

Aplikazioaren erabilera hainbat erabiltzailerekin probatu da eta euren iritzia emateko eskatu zaie. Prozesu honi esker hainbat arazo aurkitu dira eta gainera hainbat hobekuntza gehitu dira aplikazioan.

7.3.1 Lehenengo erabiltzailea

Erabiltzailea astean behin korrika egitera irteten da.

Diseinua gustatu zaio eta fluido doala uste du baina ibilbidea amaitzeko botoia argiago jartzea gomendatzen du. Aldaketa hau egin da eta orain botoiak ikonoa izan beharrean Amaitu testua du eta beheko aldean agertzen da.

SD txartelean instalatzeko aukera eskertuko lukeela esaten du mugikorraren memoria ez erabiltzeko. Aukera hau gehitu da aplikazioaren manifestean aldaketa txiki bat gehituz Android Developer [5] orrialdean oinarrituta.

Azkenengo iradokizuna nire ibilbideak pantailaren aldaketak izan dira. Ibilbide batean klik luzea egiten partekatzeko aukera izatea nahi du ibilbidera sartu gabe. Iradokizun hau inplementatu egin da. Horretaz gain, etiketak ezartzeko aukera izan eta ordena aldatzeko aukera eskatu du ibilbideak azkarrago bilatu ahal izateko.

7.3.2 Bigarren erabiltzailea

Erabiltzaile hau mendira joaten da hilabetean behin eta paseo batzuk ematen ditu herrian bertan.

Nire ibilbideetako ibilbideak ezabatzeko aukera nahi du. Inplementatua izan da eta ibilbidean klik luze bat eginez ezabatzeko aukera agertuko da.

7.3.3 Hirugarren erabiltzailea

Erabiltzailea mendira joaten da astean behin eta perretxikoak hartzera ere joaten da.

Grafikoan m/s tan abiadura km/h-tara aldatzea gomendatzen du. Aldaketa hau egin da km/h ezagunagoa egiten delako eguneroko bizitzan oso erabilia delako (Trafiko seinaleak...).

Ibilbide berri bat hastean GPS seinalea hartu bitartean Bergara agertzen da mapan. Hasiera batean uste zuen aplikazioak Bergaran zegoela markatzen ari zela eta oraindik kargatzen dagoela adierazten duen adierazle bat jartzea gomendatzen du. Hau konpontzeko GPS seinalea hartu bitartean maparen gainean kargatzen ari dela adierazteko zirkulu bat jarri da (ProgressBar bat).

Aurreko erabiltzailearen moduan ibilbideak ezabatzeko aukera eskatu du.

Mapan klik egitean abiadura eta altueran koma eta gero zenbaki gehiegi zeudela komentatu du. Adibidez, abiadura 4.574757574467657452 km/h izan daiteke. Aldaketa egin da koma eta gero bi zenbaki bakarrik azaltzeko abiaduraren kasuan eta altuera metrotan jarri da komarik gabe.

Azkenik errore bat aurkitu du. Ibilbidea berria martxan badaukagu eta saioa ixten badugu datuak igotzen jarraituko du. Hau konpondu egin da eta orain saioa ixtean ibilbidea ere amaitu egingo da.

7.3.4 Laugarren erabiltzailea

Erabiltzaile honek egunero paseo txiki bat ematen du eta hilabetean bi aldiz mendira joaten da.

Gomendatzen du defektuz ibilbidea publikoa izatea. Aldaketa egin da eta pribatua botoia defektuz desaktibatuta dago orain.

8 Ondorioak eta etorkizuneko lanak

Proiektua garatzean hainbat material (kodea, dokumentazioa, webgunea...) sortu da. Hurrengo helbideetan aurkitu daiteke material guzti hau:

- [Kodea eta dokumentazioa Github webgunean](#)
- [Webgunea: http://live-tracking.appspot.com/](http://live-tracking.appspot.com/)
- [Aplikazioa PlayStore dendan](#)
- [JavaDoc dokumentazioa](#)

8.1 Balorazioa pertsonala

Landutako lanarekin eta emaitzarekin pozik egon arren, argi dago planifikazioa ez dela guztiz egokia izan. Ondo ezagutzen ez diren materietako denborak kalkulatzeko zaila da (Arduino, Android garapena...) eta ondorioz denbora desbideraketak espero nituen. Baina ez nuen espero bigarren kuatrimestrean izan dudana lan karga. Aurreko kuatrimestrearekin konparatuz, azterketak oso garrantzi gutxi izan dute baina kuatrimestre osoan zehar lan karga handia izan da. Ondorioz, bi hilabetetan zehar proiektua ia geldirik egon zen eta planifikazioa guztiz aldatzera behartu ninduen. Galdutako denbora hori udan errekuperatu da.

Bestalde, ikuspegi akademiko batetik, GAL honek karrerako ezagupen ezberdinak praktikan jartzeko aukera ezin hobea eman dit, web sistemak eta aplikazio mugikorak oinarri izan direlarik. Horrez gain, “Internet of Things” arloan erabiltzen diren elektronikako ezagutzak lortzea ere ahalbideratu dit. Azkenik, “produktu” baten garapen oso edo integral bat egitea (softwarea eta hardwarea) oso suspergarria izan da.

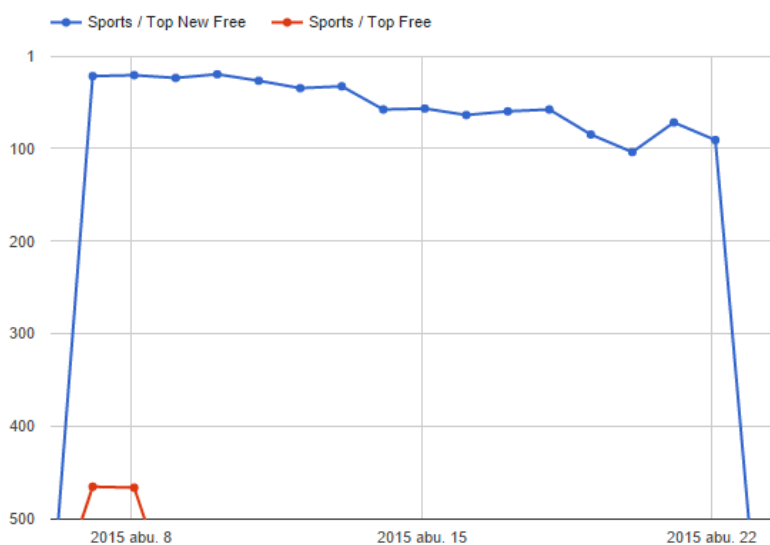
8.2 Erabiltzaileen harrera

Aplikazioak espero zen baino deskarga eta erabilpen handiagoa izaten ari da irudia 43. Irailaren 8an PlayStore dendan aplikazioak 171 deskarga zituen, horietako 96-ek instalatuta izaten jarraitzen zuten eta 12 pertsonak baloratu zuten aplikazioa. Guztiek 5 izar eman diote aplikazioari.

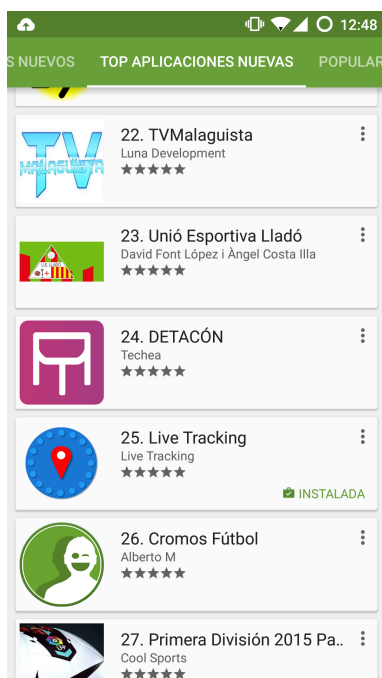
NOMBRE DE LA APLICACIÓN	PRECIO	INSTALACIONES ACTUALES/TOTALES	VALORACIÓN MEDIA / TOTAL
 Live Tracking 1.4	Gratuita	96 / 171	★ 5,00 / 12

Irudia 43: Irailaren 4an PlayStore kontsolako datuak

PlayStore-en aplikazioa igo eta lehenengo 30 egunetan aplikazio berria sekzioan agertzeko aukera dago [25]. Aplikazioa lehenengo 30 egunetako tarte horretan, Espainiako dendan, kirol aplikazio berrien top zerrendan egon da hainbat egunetan irudia 44. Egun onenean 21. postuan egon da eta hainbat egunetan 30. postua baino beherago egon da irudia 45. Estatu gehienetako aplikazio berrien sekzioan egon da posizio handiagoetan izan arren (Suitza, Alemania, Frantzia, Erresuma Batua, Indonesia, India, Mexiko, Herbehereak, Errusia, Saudi Arabia eta Estatu Batuak). Honetaz gain, pare bat egunetan kirolen top zerrenda arruntean egon zen, berria zela kontuan hartu gabe (465 eta 466 posizioetan).



Irudia 44: Appbrain-etik ateratako grafikoa. Espainiako PlayStore-eko kirol sekzioan aplikazioak izandako posizioak



Irudia 45: PlayStore-eko kirol sekzioko aplikazioa berrien top zerrenda

8.3 Lanaren ustiapena

Proiektua mantentzeko asmoa dut izan duen harrera ona kontutan izanda. Zerbitzaria dohainik da muga batzuk gainditzen ez diren bitartean [31] eta ongi etorri moduan kreditu batzuk ematen zutenez muga horrek gainditu arren mantentzeko aukera dut. Feedback asko jaso dut erabiltzaileetatik eta ikusten da proiektua erabiltzen ari dela.

Aplikazioaren harrera oso ona izan denez bi lehiaketatan parte hartzeko asmoa daukat:

- <http://gipuzkoapps.eus/eu/>
- <https://www.europe-samsung.com/smsdev/Home/Articulo/267>

8.4 Etorkizunerako lana

Proiektua hobetzeko badago lana. Hurrengoak dira pentsatu diren hobekuntza batzuk:

- Webguneak altuera eta abiadura grafikoa izatea.
- Webgunetik ibilbideak kudeatzea (Pribatua den edo ez aldatu, ezabatu edo izena aldatu).
- Google ez den beste Oauth kontu identifikazio batzuk onartzea (Twitter, Facebook...).

Bibliografia

- [1] *9 Excellent Tools for Design Mockups*. 2012. URL: <http://mashable.com/2012/06/07/mockup-tools>.
- [2] Adafruit. *Adafruit NeoPixel Library*. URL: https://github.com/adafruit/Adafruit_NeoPixel.
- [3] *Adafruit Learn*. URL: <https://learn.adafruit.com/search?q=flora>.
- [4] *Android dominates EU, WP closes gap to iOS*. 2013. URL: http://www.gsmarena.com/android_dominates_top_5_european_countries_with_more_than_70_usage-news-6300.php.
- [5] *App Install Location*. URL: <http://developer.android.com/intl/es/guide/topics/data/install-location.html>.
- [6] Tyler Cooper Becky Stern. *FLORA NeoGeo Watch*. 2015. URL: <https://learn.adafruit.com/flora-geo-watch/overview>.
- [7] Codeduino. *Which Arduino Should I Buy?* 2013. URL: <https://codeduino.com/information-and-news/hardware/which-arduino-should-i-buy/>.
- [8] *Creating a Navigation Drawer*. URL: <https://developer.android.com/intl/es/training/implementing-navigation/nav-drawer.html>.
- [9] *Datastore Concepts Overview*. 2015. URL: <https://cloud.google.com/datastore/docs/concepts/overview>.
- [10] Dale DePriest. *NMEA data*. URL: <http://www.gpsinformation.org/dale/nmea.htm>.
- [11] *Developer Android*. URL: <http://developer.android.com/intl/es/index.html>.
- [12] Stoyan Dimitrov. *Line connecting two points disappears when one of them is off-screen*. 2015. URL: <https://github.com/PhilJay/MPAndroidChart/issues/801#issuecomment-120079477>.
- [13] *Download Android Studio and SDK Tools*. URL: <https://developer.android.com/intl/es/sdk/index.html>.
- [14] Jill Duffy. *Best Fitness Trackers for 2015*. 2015. URL: <http://www.pcmag.com/article2/0,2817,2404445,00.asp>.
- [15] Dan Foster. *GPX: the GPS Exchange Format*. URL: <http://www.topografix.com/gpx.asp>.
- [16] Witold Fraczek. *Mean Sea Level, GPS, and the Geoid*. 2003. URL: <http://www.esri.com/news/arcuser/0703/geoid1of3.html>.
- [17] *Fragments*. URL: <http://developer.android.com/intl/es/guide/components/fragments.html>.
- [18] Garmin. *Garmin LiveTrack*. URL: <http://www8.garmin.com/livetrack/>.
- [19] *Google Cloud Endpoints Overview*. 2015. URL: <https://cloud.google.com/appengine/docs/java/endpoints/>.
- [20] *Google Maps Javascript API*. URL: <https://developers.google.com/maps/documentation/javascript/>.

- [21] *GPX*. URL: <http://wiki.openstreetmap.org/wiki/GPX>.
- [22] Mikal Hart. *Tiny GPS++*. 2014. URL: <http://arduiniana.org/libraries/tinygpsplus/>.
- [23] Philipp Jahoda. *MPAndroidChart*. 2015. URL: <https://github.com/PhilJay/MPAndroidChart>.
- [24] Jerry.Li. *Bluetooth spp tools pro*. 2014. URL: https://play.google.com/store/apps/details?id=mobi.dzs.android.BLE_SPP_PRO&hl=en.
- [25] *La oportunidad de Google Play*. URL: <https://developer.android.com/intl/es/distribute/googleplay/about.html>.
- [26] *LocaToWeb*. URL: <http://locatoweb.com/>.
- [27] *Markup Validation Service*. URL: <https://validator.w3.org/>.
- [28] *Material Design*. URL: <https://developer.android.com/intl/es/design/material/index.html>.
- [29] *Material Design Lite*. URL: <http://www.getmdl.io/>.
- [30] EILEEN MCNULTY. *SQL VS. NOSQL- WHAT YOU NEED TO KNOW*. 2014. URL: <http://dataconomy.com/sql-vs-nosql-need-know/>.
- [31] *Quotas*. URL: <https://cloud.google.com/appengine/docs/quotas>.
- [32] *Services*. URL: <http://developer.android.com/intl/es/guide/components/services.html>.
- [33] *Strava*. URL: <https://www.strava.com/>.
- [34] *The Open Graph protocol*. 2014. URL: <http://ogp.me/>.
- [35] Twitter. *Twitter App Card*. URL: <https://dev.twitter.com/cards/types/app>.
- [36] *Using Auth with Endpoints*. 2015. URL: <https://cloud.google.com/appengine/docs/java/endpoints/auth>.
- [37] *Using Endpoints in a JavaScript Client*. URL: https://cloud.google.com/appengine/docs/java/endpoints/consume_js.
- [38] *Using Endpoints in an Android Client*. URL: https://cloud.google.com/appengine/docs/java/endpoints/consume_android.
- [39] *What is an Arduino?* URL: <https://learn.sparkfun.com/tutorials/what-is-an-arduino>.
- [40] Samuel J. Wormley. *GPS Errors and Estimating Your Receiver's Accuracy*. 2010. URL: http://www.edu-observatory.org/gps/gps_accuracy.html.