

UNIBERTSITATEKO MASTERRA: TELEKOMUNIKAZIO INGENIARITZA

MASTER AMAIERAKO LANA

**ARRISKUPEKO TALDE SOZIALENTZAKO APLIKAZIO
MUGIKORRA, POSIZIOAREN MONITORIZAZIORAKO
NB-IOT SAREA ERABILITA**

Ikaslea	<i>Prieto, Murga, Irati</i>
Zuzendaria (1)	<i>Huarte, Arrayago, Maider</i>
Zuzendaria (2)	<i>Astorga, Burgo, Jasone</i>
Saila	<i>Komunikazioen Ingeniaritza</i>
Ikasturtea	<i>2021-2022</i>

Bilbo, 2022ko irailaren 18a

Aurkibidea

AURKIBIDEA	2
LABURPENA HIRU HIZKUNTZATAN	4
TAULEN ZERRENDA	6
IRUDIEN ZERRENDA	7
GRAFIKOEN ZERRENDA	9
AKRONIMOEN ZERRENDA	10
1. SARRERA	12
2. TESTUINGURUA.....	13
2.1. TESTUINGURU SOZIALA.....	13
2.2. HARIRIK GABEKO KOMUNIKAZIOAK.....	16
2.2.1. IoT (Internet of Things).....	16
2.2.2. Komunikazio estandarrak.....	17
2.3. GAUR EGUNGO PROIEKTUAK	20
2.3.1. Haurren lokalizaziorako gailuak.....	20
2.3.2. Emakumeen babeserako APPak.....	20
3. LANAREN HELBURUAK ETA IRISMENA.....	24
3.1. HELBURU NAGUSIA.....	24
3.2. BIGARREN MAILAKO HELBURUAK	24
4. LANAK DAKARTZAN ONURAK	26
4.1. ONURA SOZIALAK.....	26
4.2. ONURA TEKNIKOAK	26
4.3. ONURA EKONOMIKOAK.....	26
5. AUKEREN ANALISIA	27
5.1. DATUEN BILTEGIATZEA.....	27
5.1.1. Biltegiatze metodoa.....	27
5.1.2. Arkitektura.....	29
5.2. APLIKAZIOA.....	30
5.2.1. Aplikazio mota.....	31
5.2.2. Lengoia	32
5.3. KOMUNIKAZIO PROTOKOLOA.....	32
6. ARRISKUEN ANALISIA.....	35
6.1. ARRISKUAK.....	35
6.2. KONPARAKETA.....	36
6.3. KONTINGENTZIA PLANA.....	37
7. PROPOSATUTAKO IRTENBIDEAREN DESKRIBAPENA.....	38
7.1. PLANTEATUTAKO SOLUZIOA.....	38
7.2. DATU BASEAREN DISEINUA.....	39

7.2.1.	<i>Firestore Authentication</i>	40
7.2.2.	<i>Firestore Realtime Database</i>	41
7.3.	APLIKAZIOAREN DISEINUA.....	43
7.3.1.	<i>Aplikazioaren funtzionamendua</i>	43
7.3.2.	<i>Aplikazioaren garapena</i>	48
7.4.	NB-IOT GAILUAREN DISEINUA.....	60
7.4.1.	<i>Erabilitako gailuak</i>	60
7.4.2.	<i>Garapen prozesua</i>	64
8.	EGIN BEHARREKOEN DESKRIBAPENA	73
8.1.	LAN TALDEA ETA BALIABIDE MATERIALAK.....	73
8.2.	FASEEN DESKRIBAPENA.....	74
	<i>LP1 – Proiektuaren kudeaketa eta dokumentazioa</i>	74
	<i>LP2 – Proiektuaren definizioa</i>	74
	<i>LP3 – Soluzioaren diseinua</i>	75
	<i>LP4 – Soluzioaren garapena</i>	76
	<i>LP5 – Lanaren baliozkotzea</i>	77
9.	GANTT-EN DIAGRAMA	77
10.	KOSTUAK	79
10.1.	BARNE-ORDUAK.....	79
10.2.	AMORTIZAZIOAK.....	79
10.3.	GASTUAK.....	80
10.4.	PROIEKTUAREN KOSTU TOTALA.....	80
11.	ONDORIOAK	82
	ERREFERENTZIAK	83
I.	ERANSKINA. KODEA	86
	APLIKAZIOAREN KODEA.....	86
	GPS – NB-IOT GAILUAREN KODEA.....	86

Laburpena hiru hizkuntzatan

4G da gaur egungo errealitatean zabalduen eta erabilgarrien dagoen sarea: hau horrela, estaldura ona ez duten zonaldeetan arriskupeko talde sozialen segurtasuna bermatzea, esaterako, emakumeena, gaur egun existitzen diren tresnak erabilia, ez da bat ere erraza. Hori dela-eta, uneoro eta edonon lokalizagarri izan ahal izateko tresna bat garatzea da proiektu honen helburua.

Helburu hau lortzeko, alde batetik, NB-IoT bidez denbora guztian GPS posizioa bidaltzen duen aplikazio bat sortu da, kontzeptu-proba modura Raspberry gailu bat erabilia datuak hodeian dagoen datu base batean gordetzen dituen. Beste alde batetik, erabiltzaileari dagokion informazio guztia kudeatzeko mugikorrerako Android aplikazio natibo bat garatu da, lokalizazioen historiala mapa gainean kontsultatzeaz gain, posizio seguru bat ezartzeko aukera ematen duena, eta baita hainbat lagunen historialak mapa batean ikusteko hauta ere. Honez guztiaz gain, uneoro erabiltzailearen pribatutasuna eta segurtasuna bermatzeko neurriak ere hartu dira.

Hitz gakoak: segurtasuna, NB-IoT, Android aplikazio natiboa, datu basea.

La red más extendida y utilizable en la realidad de hoy en día es la red 4G. Siendo esto así, garantizar la seguridad de los grupos sociales en peligro, como pueden ser las mujeres, en zonas con cobertura 4G insuficiente, mediante las herramientas disponibles en la actualidad, no resulta fácil. Por ello, el objetivo de este proyecto es desarrollar una herramienta que permita al usuario ser localizable en todo momento y en cualquier lugar.

Para conseguir este objetivo, se ha ideado, por un lado, una aplicación que envía la posición GPS constantemente a través de NB-IoT utilizando como prueba de concepto un dispositivo Raspberry, y que, a su vez almacena los datos en una base de datos situada en la nube. Por otro lado, para gestionar toda la información relativa al usuario, se ha desarrollado una aplicación nativa para dispositivos Android, que, además de consultar el historial de localizaciones sobre mapa, permite establecer una posición segura e incluso ver los historiales sobre mapa del posicionamiento de otros usuarios amigos. Además de todo lo mencionado, también se han tomado medidas para garantizar la privacidad y seguridad del usuario en todo momento.

Palabras clave: seguridad, NB-IoT, aplicación nativa Android, base de datos.

The most widespread and usable network in today's reality is the 4G network. That said, ensuring the safety of endangered social groups, for instance, women's, in areas with insufficient 4G coverage, using the tools currently available, is far from easy. Therefore, the aim of this project is to develop a tool that allows the user to be reachable at any time and in any place.

To achieve this goal, on the one hand, an application has been devised to send the GPS position constantly through NB-IoT using a Raspberry device as a proof of concept, and which, furthermore, stores the data in a database located in the cloud. On the other hand, to manage all the information related to the user, a native application has been developed for Android devices, which, in addition to consulting the location history on a map, allows to establish a secure position and even view the positioning histories of other befriended users on the map. In addition to all of the above mentioned, measures have also been taken to ensure user privacy and security at all times.

Key words: safety, security, NB-IoT, native Android application, database.

Taulen zerrenda

1. Taula. Bluetooth klaseak.....	17
2. Taula. Data Warehouse eta Data Lake arteko konparaketa.....	29
3. Taula. Datu-biltegitratze metodoen aukeren analisia.....	29
4. Taula. Datu-biltegitratze arkitekturen aukeren analisia.....	30
5. Taula. Aplikazio moten aukeren analisia.....	31
6. Taula. Aplikazioaren programazio lengoaiaren aukeren analisia.....	32
7. Taula. Komunikazio protokoloen aukeren analisia.....	34
8. Taula. GPS bidez lortutako informazioaren formatua.....	65
9. Taula. Proiektuaren lan-taldea.....	73
10. Taula. Proiektuan erabilitako baliabide materialak.....	73
11. Taula. Barne-orduen kostua.....	79
12. Taula. Amortizazioen kostua.....	80
13. Taula. Gastuak.....	80
14. Taula. Proiektuaren kostu totala.....	81

Irudien zerrenda

1. Irudia. Emakumeei ziztadak Espainian 2022ko abuztu hasieran.....	15
2. Irudia. IoTren oinarritzko arkitektura.....	17
3. Irudia. ZigBee protokolo-pila.....	18
4. Irudia. AlertCops aplikazioaren pantaila printzipala.....	21
5. Irudia. EraStop aplikazioaren funtzionaltasunak.....	22
6. Irudia. BINDI proiektua.....	23
7. Irudia. BINDI proiektuaren eskumuturrekoa.....	23
8. Irudia. Sigfox, LoRa eta NB-IoT arteko konparaketa.....	34
9. Irudia. Arriskuen probabilitate-eragin matrizea.....	36
10. Irudia. Planteatutako soluzioaren eskema orokorra.....	38
11. Irudia. Firebase-n autentikazio metodoa.....	41
12. Irudia. Datu basearen arkitektura.....	42
13. Irudia. Users taularen arauak.....	42
14. Irudia. Sessions taularen arauak.....	43
15. Irudia. Aplikazioaren autentikazio-erlaitza, pasahitza berreskuratzea eta kontua sortzea.....	44
16. Irudia. Aplikazioaren hasierako menuaren erlaitza.....	44
17. Irudia. Aplikazioaren posizioen historialaren mapa.....	45
18. Irudia. Aplikazioaren ezarpenen erlaitza.....	46
19. Irudia. Aplikazioaren posizio segurua gordetzeko erlaitza.....	46
20. Irudia. Aplikazioaren lagunen zerrendaren erlaitza.....	47
21. Irudia. Aplikazioaren lagunen historiala ikusteko maparen erlaitza.....	48
22. Irudia. Aplikazioaren kodearen antolaketa.....	49
23. Irudia. Aplikazio mailako dependentziak Firebase-ren inplementaziorako.....	50
24. Irudia. Proiektu mailako Gradle fitxategia Firebase-ren inplementaziorako.....	50
25. Irudia. Saioa hasteko botoiaren funtzioa.....	51
26. Irudia. Saioa hasteko autentikazio funtzioa.....	52
27. Irudia. Hasierako orrira joateko funtzioa.....	52
28. Irudia. Firebase-etik erabiltzailearen datuak jasotzeko funtzioa.....	53
29. Irudia. XML fitxategiaren etiketen adibideak.....	54
30. Irudia. Ezarpenen orria kargatzean exekutatutako kodea.....	55
31. Irudia. Pantailako datuak aldatzeko kodea.....	55
32. Irudia. Erabiltzailearen ezarpenak datu basean gordetzeko kodea.....	56
33. Irudia. Strings.xml fitxategiko testuen erabilera.....	57
34. Irudia. Hizkuntzaren detekzioa.....	57
35. Irudia. Aplikazioaren konfigurazioan egindako hizkuntza aldaketa.....	57
36. Irudia. Manifestuan API_KEYa gehitzea.....	58
37. Irudia. Mapa inplementatzeko fragment-a.....	58
38. Irudia. Mapan inplementatutako funtzionaltasunak.....	59
39. Irudia. Ekintza periodikoa definitzeko kodea.....	59
40. Irudia. Raspberry Pi 3 Model B gailua.....	61

41. Irudia. Waveshare SIM7000E NB-IoT HAT gailua.....	62
42. Irudia. Raspberry eta HATaren arteko interkonezioa.....	63
43. Irudia. Amarisoft AMARI Callbox mini base-estazioa.....	64
44. Irudia. Portu serie bidezko komunikazioaren ezarpena.....	65
45. Irudia. GPSaren konfigurazioa.....	65
46. Irudia. GPStik jasotako informazioa irakurri eta prozesatzea.....	67
47. Irudia. Pyrebase-aren konfigurazioa Python-en.....	68
48. Irudia. Erabiltzailearen saioa hasita Raspberry-tik.....	68
49. Irudia. Datu basean GPS datuak idaztea.....	68
50. Irudia. 30 segundoro errepikatutako kodea.....	69
51. Irudia. NB-IoT gailuaren lan-maiztasun bandak.....	69
52. Irudia. NB-IoT-n lan egiteko konfigurazioa.....	70
53. Irudia. Operadorea aukeratzeko komandoa.....	70
54. Irudia. APNaren konfigurazioa.....	70
55. Irudia. Erabiltzailearen ekipoaren konfigurazio zuzenaren ziurtatzea.....	70
56. Irudia: Ekipoaren IParen esleipena.....	71
57. Irudia. Google-era egindako PING zuzena.....	71
58. Irudia. NB-IoT sarea USB bidez jasotzeko konfigurazioa.....	72
59. Irudia. Sarea era egokian konfiguratu dela bermatzea.....	72
60. Irudia. Gantt-en diagrama.....	78

Grafikoen zerrenda

1. Grafikoa. 016ra egindako deien eboluzioa (Espainiar Estatuan)..... 14
2. Grafikoa: Genero indarkeriagatik egindako salaketa kopuruak (EAEn)..... 15

Akronimoen zerrenda

API	<i>Application Programming Interface</i>
BLE	<i>Bluetooth Low Energy</i>
BPSK	<i>Binary Phase-Shift Keying</i>
CDMA	<i>Code Division Multiple Access</i>
CSS	<i>Chirp Spread Spectrum</i>
DBAO	<i>Datuak Babesteko Araudi Orokorra</i>
DBPSK	<i>Differential BPSK</i>
EDGE	<i>Enhanced Data-rate for Global Evolution</i>
FDMA	<i>Frequency Division Multiple Access</i>
FEC	<i>Forward Error Correction</i>
FFD	<i>Full Function Device</i>
FHSS	<i>Frequency Hopping Spread Spectrum</i>
GMSK	<i>Gaussian minimum shift keying</i>
GPS	<i>Global Positioning System</i>
GSM	<i>Global System for Mobile communication</i>
HAT	<i>Hardware Attached on Top</i>
HDOP	<i>Horizontal Dilution of Precision</i>
HPA	<i>Horizontal Pod Autoscaling</i>
IoT	<i>Internet of Things</i>
ISM	<i>Industrial, Scientific and Medical</i>
JVM	<i>Java Virtual Machine</i>
LoRa	<i>Long Range</i>
LoRaWAN	<i>Long Range Wide Area Networks</i>
LPWAN	<i>Low-Power Wide Area Networks</i>
LTE	<i>Long Term Evolution</i>
M2M	<i>Machine to Machine</i>
MOE	<i>Munduko Osasun Erakundea</i>
MSL	<i>Mean Sea Level</i>
NBE	<i>Nazio Batuen Erakundea</i>
NB-IoT	<i>Narrow Band Internet of Things</i>
NoSQL	<i>Not Only SQL</i>

PDOP	<i>Position (3D) Dilution of Precision</i>
PK	<i>Primary Key</i>
QAM	<i>Quadrature Amplitude Modulation</i>
RFD	<i>Reduced Function Device</i>
RFID	<i>Radio Frequency Identification</i>
SQL	<i>Structured Query Language</i>
TDD	<i>Time Division Duplex</i>
TDMA	<i>Time Division Multiple Access</i>
VDOP	<i>Vertical Dilution of Precision</i>
VPA	<i>Vertical Pod Autoscaling</i>
WSN	<i>Wireless Sensor Network</i>

1. Sarrera

IoT (*Internet of Things*) gailu konektatuen sarea da, gailuen eta hodeiaren arteko eta baita gailuen beren arteko komunikazioa errazten duen teknologiarri erreferentzia egiten diona. Gailu hauek, orokorrean, kostu baxukoak eta gaitasun urrikoak izaten dira, sentore ezberdinak dituztenak. Mugatutako gaitasun hauek orotarikoak dira: memoria, energia iturria, prozesadorea... eta honen arabera, erabilera bat edo beste eman dakieke.

Gaur egun, IoT gero eta gehiago erabiltzen da, gero eta gailu gehiago daude interkonektatuta, eta, are gehiago, kopuru honek datozen urteetan gora egiteko joera du. Goranzko joera honek, erabileraren zabalkuntza du oinarri; izan ere, gero eta bizitzaren eremu gehiagotan erabiltzen dira IoT gailuak, esate baterako, autoak konektatzeko, etxeetan, *smart city* edo hirietan, eta beste hainbestetan.

Hau horrela, proiektu honetan, IoT gailu bati beste helburu bat eman zaio. Hain zuzen ere, arriskuan senti daitezkeen herritarrak, edo euren eguneroko bizitzan segurtasun gehiago nahi duten pertsonen lokalizatorako gailu bat eskaini zaie. Kasu honetan, gainera, egun erabileran dagoen 4G estaldurarik ez dagoen toki eta uneetan ere, konexioa izango dela bermatu da. Honen arrazoa, batez ere, jende asko pilatzen den tokietan, existitzen diren babes- eta segurtasun-mekanismoek funtzionatzen ez dutela da. Hau da, adibidez, festa giroan, sarri entzuten da erasoren bat egon dela, izan ere, segurtasunerako aplikazioek interneterako konexiorik gabe ez dute funtzionatzen, eta aplikazio horien erabiltzaileek (esaterako, emakumeek) ez daukate abisu emateko modurik.

Proiektuaren zergatia azalduta, nondik norakoak deskribatuko dira jarraian. Beraz, aurkitutako arazoari konponbidea emateko, bi ataletan banatu da garatutako soluzioa. Alde batetik, Raspberry bat erabilia, GPS eta NB-IoT modulu bat konektatuta duelarik, uneoro erabiltzailearen kokapena datu base batera bidaltzen duen sistema bat garatu da. Beste alde batetik, lortutako datu horiek kudeatzeko Android aplikazio natibo bat sortu da. Aplikazio honek, gaur egungo segurtasun aplikazio edota gailuek dituzten funtzionaltasun antzekoak eskaintzen ditu; esate baterako, puntu seguru bat ezartzea eta bertatik urrunduta automatikoki posizioa gordetzea edota lagun zerrenda bat konfiguratu ahal izatea, batak bestearen lokalizazioen historiala ikusi ahal izateko mapa batean. Azken funtzionaltasunari dagokionez, erabiltzailearen segurtasuna soilik ez, bere datuen segurtasun eta konfidentziasuna ere hartu behar izan da kontuan, berak erabakitako pertsonen bakarrik izan dezaten datu pribatuak eskuratzeko aukera.

Laburbilduz, proiektu honetan LTE teknologia alde batera utzita, IoTren gaur egungo merkatuan koka daitezkeen produktu bat diseinatu da, erabilerraza eta segurua, jadanik existitzen diren erramintak kontuan hartu eta pauso bat aurrerago emanda sortua.

2. Testuingurua

Puntu honetan proiektua bere testuinguruan jarri da, oinarri duen gaia kokatzeko, eta baita bere garapen eta ulermenerako ezagutu beharreko oinarritzko kontzeptuak azaltzeko ere. Lehenik eta behin, lan hau gauzatzeko izan den testuinguru soziala komentatuko da. Jarraian, teknologiaren egoera zein den azalduko da, bai programazioari dagokionez, bai komunikazio protokoloari dagokionez. Azkenik, egun existitzen diren erlazionatutako beste proiektu batzuk aurkeztuko dira.

2.1. Testuinguru soziala

Arriskupean dagoen hori nolabaiteko askatasun-falta duen oro da. Hau horrela izanik, zenbait pertsona edo talde sozial bereziki kalteberak dira eta babes gehigarria behar dute [1]. Arrisku-faktoreak hiru dira:

- **Faktore-pertsonalak:** adina, generoa, ikasketa maila, herritartasuna, osasun fisiko eta mentala, egoera legal edo ekonomikoa, bizitakoak edota edonolako traumak.
- **Ingurumen-faktoreak:** Osasun-laguntzarako sarbidea eta eskumena, gizarte-eta lege-zerbitzuak, pribilegio-sistema informalak, etab.
- **Faktore soziokulturalak:** gizartearen eta komunikabideen jarrera, estigmatizazioa, ikusezintasun soziala, gutxiengoekiko jarrera eta beste hainbeste.

Hau guztia kontuan izanik, sei dira bereizten diren talde handienak:

- Emakumeak.
- Haurrak eta nerabeak.
- Lesbianak, gayak, bisexualak, transgeneroak edo intersexualak (LGTBI).
- Desgaitasun fisiko edo mentalen bat dutenak.
- Atzerritarrak.
- Gutxiengo etnikoetako edo herri indigenetako pertsonak.

Jarraian emakumeen egoera azalduko da zehazkiago. Emakumearen aurkako indarkeria herrialde, klase sozial eta gizartearen alor guztietan gertatzen den fenomeno da. Nazio Batuen Erakundeak (NBE) horrela definitzen du: "emakumeari min fisiko, sexual edo psikologikoa eragiten dion edozein motatako genero indarkeria

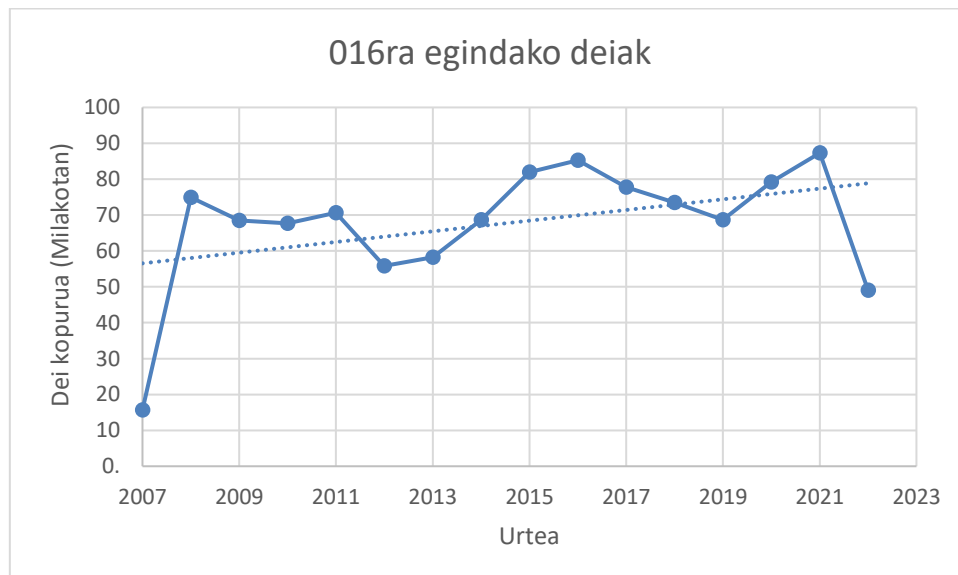
ekintza, eta baita hauen mehatxu, behartze edota askatasun-gabetze arbitrarioa ere; bai bitzta publikoan, bai pribatuan gertatzen direnak”[\[2\]](#).

Gainera, Munduko Osasun Erakundearen (MOE) arabera, indarkeria mota hau emakumearen giza eskubideen bortxaketa gorria izateaz gain, osasun publikoari dagokion arazo larria ere bada [\[3\]](#).

Hau horrela, eta gaitzaren norainokoak argiago ikusteko, MOEk 2000 eta 2018 urteen artean, 161 herrialdetan, analisia egin zuen. Datuen arabera, ia hiru emakumetik batek (% 30) nolabaiteko indarkeria fisiko edota sexuala jasan du; bikotekideen artean, Europan, emakumeen % 25ek indarkeria jasan duela estimatzen da; eta hildako emakumeen % 38 beren bikotekideek hil dituzte. Gainera, mundu osoko emakumeen % 6k beren bikotekideaz bestelakoan sexu-erasoak jasan dituela, nahiz eta honen inguruko datuak mugatuagoak diren.

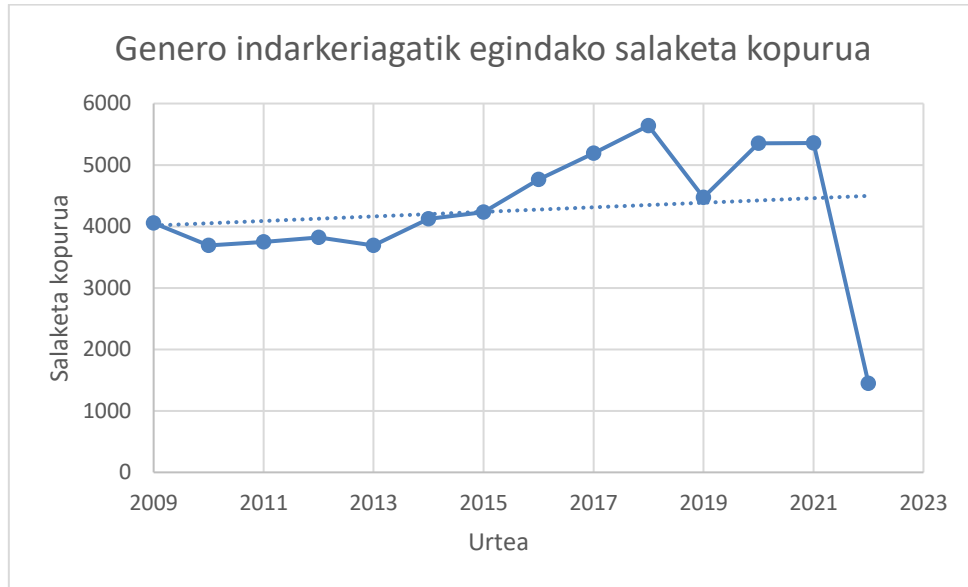
Ikus daitekeenez, munduan zehar hainbat azterketa egin eta egiten dira, baina bertoko egoera zein den argiago ikusi ahal izateko, hurbilagoko datuak aztertu dira, Espainiako Gobernuaren Portal Estatistikoko Genero Indarkeriaren Aurkako Deleagaritzatik ateratako zenbait datu, hain zuzen ere.

- **016 telefono zenbakira egindako deiak.** Espainiar Estatuko indarkeriaren aurkako telefono zenbakia da. Informazioa eta laguntza eskaintzen ditu, genero indarkeriaren alorrean. Osasun ministerioak, Zerbitzu Sozialek eta Berdintasun Zerbitzuak koordinatzen dute. 1. Grafikoan ikus daitekeenez, batezbestekoa 70,000 dei ingurukoa da, eta goranzko joera du [\[4\]](#).



1. Grafikoa. 016ra egindako deien eboluzioa (Espainiar Estatuan).

- **Genero indarkeriatik egindako salaketak:** Mota honetako salaketak ere, gero eta gehiago dira Euskal Autonomia Erkidegoan, 2. Grafikoan ikus daitekeen joera lerroaren maldaren arabera [\[5\]](#).



2. Grafikoa: Genero indarkeriagatik egindako salaketa kopuruak (EAEn).

Hau guztia gutxi balitz, 2022ko udan, beste hainbat herrialdeetatik mugitu ostean, Espainian ere emakumeei egindako zitzadak modan jarri dira. Zitzada hauek hainbat kasutan sumisio kimikorako bidea izan dira, baina beti eraso, eta hamarnaka egon dira Espainia osoan, herrialde osoko emakumeak beldurtuz eta 112 telefonora deiak eta larrialdietara bisitak areagotuz. Jarraian ikus daitezkeen mapan, abuztu hasieran zenbatutako zitzadak ikus daitezke, autonomia erkidegoka. [6]



1. Irudia. Emakumeei zitzadak Espainian 2022ko abuztu hasieran.

2.2. Haririk gabeko komunikazioak

Haririk gabeko teknologiak gailu bat edo gehiago konexio fisikorik gabe komunikatzea ahalbidetzen du. Hau horrela, haririk gabeko teknologiaren eta hari bidezko teknologiaren aplikazioen arteko ezberdintasun handiena, hurrengoa da: lehenengoak hari gabeko maiztasuneko transmisio-gailu bat erabiltzen du datuak transmititzeko; bigarrenak, berriz, kable bat behar du transmisio- eta jasotze-bide gisa.

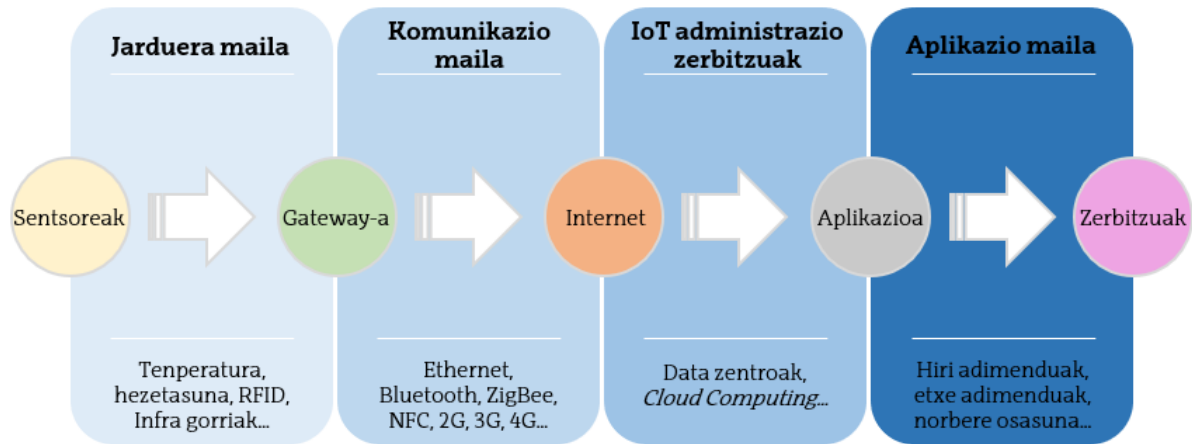
Oro har, hari gabeko sentso-reseak (WSN – *Wireless Sensor Network*) jadanik existitzen diren teknologiak eta estandarrak erabiliz diseinatu dira. WSN sare hauetan, beraz, estandar egokiena, aplikazio espezifikoaren eskakizunen arabera aukeratu da, eta, ondoren, beraz inplementatzeko eman beharreko diseinu-urratsak ematen dira. Aplikazio hauen eskakizunak orotarikoak izan daitezke: banda-zabalera handia, gailuen bateriaren iraupen luzeagoa edota estaldura hobea.

2.2.1. IoT (*Internet of Things*)

Gauzen Interneta edo IoT hainbat modutan definitu daiteke. Beste hainbaten artean, gailuak eta objektuak sare baten bidez, pribatua edo Interneta bera izan daitekeena, elkartu eta interkonektatzea. Sare horretara konektatu daitezkeen gailuei dagokionez, edozein izan daiteke, sentso-rese edo gailu mekanikoetatik hasi eta eguneroko objektuetara arte, hozkailua edo arropa, esaterako. Edozer konektatu liteke Internetera eta gizakiaren esku-hartzerik gabe funtzionatu; beraz, helburua makina-makina (M2M – *Machine to Machine*) interakzioa da. Gainera, sentso-rese teknologian egiten diren aurrerapenek IoT-ren zabalkuntzan laguntzen dute.

IoT-aren oinarritzko arkitekturari dagokionez, lau mailatan banatu da, 2. Irudian ikus daitezkeen bezala. Sentso-rese dira jardura mailan ikusten den lehen elementua. Aplikazioen arabera, hainbat sentso-rese izan daitezke, tenperatura edo hezetasun-sentso-reseak, RFID (*Radio Frequency Identification* – Irrati-maiztasun identifikatzaileak) eta beste hainbeste. Komunikazio mailak, *Gateway*-ek osatua, sentso-rese datuak tratatu eta zenbait interfaze estandar ematen ditu (Ethernet, Bluetooth...). Interfaze haiek mailetakosoki osagai guztiak konektatzen dituzte. IoT-k bai IPv4 eta IPv6 onartzen ditu, eta sortutako azpiegiturari IP-n oinarritutako azpiegitura deritzaio.

Hainbat aplikazio teknologikotarako, sentso-rese-nodoetako datuak hartzen dira, *Cloud Computing*-erako eta datuak analizatzeko, IoT administrazio zerbitzuak liratekeenak. Hauen guztien gainetik, aplikazio maila dago, eta honen periferikoak IoT aplikazioen arabera aldatzen dira [7].



2. Irudia. IoTren oinarritzko arkitektura.

2.2.2. Komunikazio estandarrak

Azpiatal honetan helmen laburreko eta datu-tasa txikiko zenbait komunikazio teknologia azalduko dira. Jarraian, helmen eta abiadura txikiko bi estandar hedatuena azalduko dira; hauek 2.4 GHz-ko ISM bandan egiten dute lan haririk gabeko komunikazioetan.

- **Bluetooth**

Distantzia laburreko haririk gabeko komunikazioetarako erabiltzen da. Esan bezala, 2.4 GHz-ko ISM bandan egiten du lan, eta aldiberekotasuna ekiditeko, *Frequency Hopping Spread Spectrum*-a (FHSS) erabiltzen du, aldi berean 1 MHz-ko banda zabalerako 79 kanal eratzea, alegia.

Teknologiari dagokionez, Bluetooth-a klaseka eta bertsioka sailka daiteke. Klaseei erreparatuz, 4 dira, eta helmenaren eta transmisio potentziaren arabera antolatzen dira, hurrengo taulan ikus daitekeenez.

1. Taula. Bluetooth klaseak.

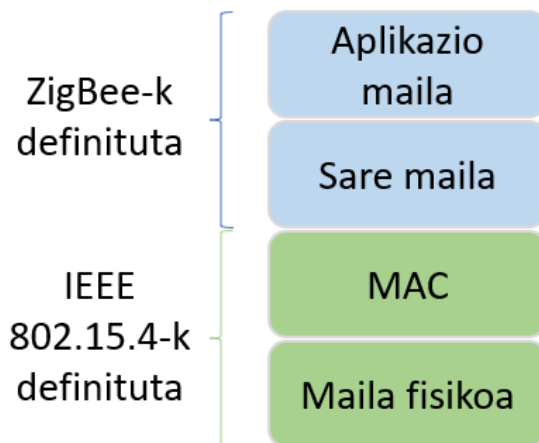
Bluetooth klasea	Helmena (m)	Transmititutako potentzia (mW)
1. maila	100	100
2. maila	10	2.5
3. maila	0.1	1
4. maila	0.05	0.5

Bertsioei dagokienez, ordea, Bluetooth 1-etik Bluetooth 5-era doaz, bigarren hau izanik bertsiorik berriena. Ateratako bertsio bakoitzak, aurrekoak baino

ezaugarri gehiago ditu; esate baterako, azken bertsioek (*Bluetooth Low Energy* (BLE) eta Bluetooth 5) potentzia txikirako eta IoT aplikazioetan erabiltzeko daude gehienbat diseinatuta.

- **ZigBee**

ZigBee potentzia eta abiadura baxuko protokoloa da, ausazko tarteetan datu segida luzeak transmititzeko bereziki diseinatua. Gainera, amatatuta ematen duen denbora-tartea egokituz, nodoaren bizitza-iraupena luza daiteke. ZigBee sareei dagokienez, zuhaitz, izar edo *mesh* topologiak erabil ditzakete, eta helmena bideragailuak erabiliz handitu daiteke. Bere protokolo-pilari dagokionez, hurrengo irudian ikus daitekeena da:



3. Irudia. ZigBee protokolo-pila.

Behin estandar hauek ezagututa, jarraian potentzia txikiko eta helmen handiko (*Low-Power Wide Area Networks - LPWAN*) estandarrak azalduko dira. Protokolo hauek IoT-ren hedapenarekin hazi dira, izan ere, gauzen internetak era egokian funtzionatzeko, potentzia txikia behar duten baina distantzia handiagoak estal ditzaketen estandarrak behar dira. Jarraian azaldukoak dira zenbait:

- **LoraWAN**

LoRa Semtech korporazioak garatutako berezko PHY protokoloa da. LoraWAN protokoloak, 1 GHz baino gutxiagoko irrati bandan modulatu, 10 km-ra arteko estaldura eskaintzen du, potentzia txikia kontsumituz.

Sarearen funtzionamenduari dagokionez, data transmisioa era asinkronoan egiten da. Hau horrela, pakete bera hainbat aldiz birtransmititu daiteke eta erreduantzia sortu. Zerbitzari zentralak bikoiztutako paketeak ezabatzen ditu eta segurtasuna bermatu. Datu trafikoa oso handia ez denean, era egokian funtzionatzen du; hala ere, hau handituz doan heinean, bere errendimendua degradatu egiten da.

- **Weightless**

Weightless hainbat LPWAN teknologiaren batura da, oinarri estazio baten eta milaka nodoren arteko komunikazioaz arduratzen dena. Hiru estandar bereizten dira, bakoitza konfigurazio ezberdinekin:

- *Weightless-W*: Telebistako espazio hutsen bandan egiten du lan eta hauen hedapen ezaugarriak erabili. Oinarri estaziora bidalitako informazioa *narrow band*-ean bidaltzen da, kontsumoa txikitzeko.
- *Weightless-N*: Amaierako nodoetatik base-estaziorainoko noranzko bakarreko komunikazioa ahalbidetzen du, beraz, energetikoki efizienteena da. Nolanahi ere, noranzko bakarreko konfigurazioa izateak, bere funtzionaltasuna murrizten du.
- *Weightless-P*: Bi norabidetako komunikazioa ahalbidetzen duen protokoloa da

Azkenik, sare mugikorretarako erabiltzen diren edo erabili diren zenbait estandar dira aztergai:

- **2G**

Global System for Mobile Communications (GSM) da bigarren belaunaldiko (2G) komunikazioetarako gehien erabiltzen den teknologia, bere sinpletasuna dela-eta. FDMA (*Frequency Division Multiple Access*) eta TDMA-ri esker (*Time Division Multiple Access*) hainbat erabiltzailek aldi berean erabil dezakete kanal bera.

- **3G**

Hirugarren belaunaldiko (3G) teknologia, sare operadore publiko edo pribatuen bidez, lurreko eta satellite bidezko sistemei abiadura handiko datu transmisioa emateko sortu zen. Bi sarbide-teknologia erabiltzen ditu: eboluzio globalerako datu-tasa indartua (EDGE – *Enhanced Data-rate for Global Evolution*) eta banda zabaleko CDMA (W-CDMA – *Wideband-CDMA*).

- **4G**

Laugarren belaunaldiko epe-luzeko eboluzioak (4G-LTE – *Long Term Evolution*) 50 Mbps-eko *uplink*-a du, eta 150 Mbps-ko *downlink*-a. Hala ere, behar den transtzeptorea oso garestia da, eta kontsumo handikoa; IoT-ren eskakizunak ez ditu betetzen, alegia.

Hau horrela, 3GPPren 12. Bertsioan, IoT sistemetarako pentsatutako kategoria berri bat gehitzen da, Cat 0. Kasu honetan, *uplink* eta *downlink* datu-tasak 1 Mbps-ra mugatzen dira. Gainera, LTE banda berean funtziona dezakeen arren, banda zabalera 1.4 MHz-ra murrizten da. Azkenik, aipatzekoa da 4G estaldurarik ez dagoen puntuetan, 3G eta 2G teknologia erabiltzen dela, sistemaren sentikortasuna txarragotuz.

2.3. Gaur egungo proiektuak

Segurtasunerako gailu eta aplikazioei dagokienez, ugari dira merkatuan gaur egun aurki daitezkeenak. Hala ere, bi taldetan banatu dira, egiteko antzekoa duten arren, helburuaren arabera. Alde batetik, haurrak lokalizatuta egoteko GPS gailuak daude, guraso edo tutoreek mugikorrean duten aplikazio batera konektatzen direnak, haurra bera uneoro non dagoen jakiteko. Beste alde batetik, emakumeen babeserako aplikazioak daude, lagunekin lokalizazioa partekatu edo larrialdietara zuzenean deitzeko aukera ematen dutenak.

2.3.1. Haurren lokalizaziorako gailuak

Haurren lokalizaziorako gailuak, hurrek gainean eramateko gailu txikiak dira, bai motxilan, arropan edo jantzita eraman ditzaketenak, hainbat erabileradunak [8]. Gailu hauek gurasoek mugikorrean daramaten aplikazio batera konektatzen dira, eta bertatik hainbat funtzionalitatetarako aplikatu ditzakete. Gailu hauen ezaugarri eta funtzionaltasun ohikoenak ondokoak dira:

- Haurrak gurasoen ondotik aldentzean abisua jasotzea.
- Haurrak zonalde segurutik aldentzean abisua jasotzea.
- Haurren kokapena momentu oro ezagutzea.

Beste askok, gainera, funtzionaltasun aurreratuagoak ere badituzte, esate baterako:

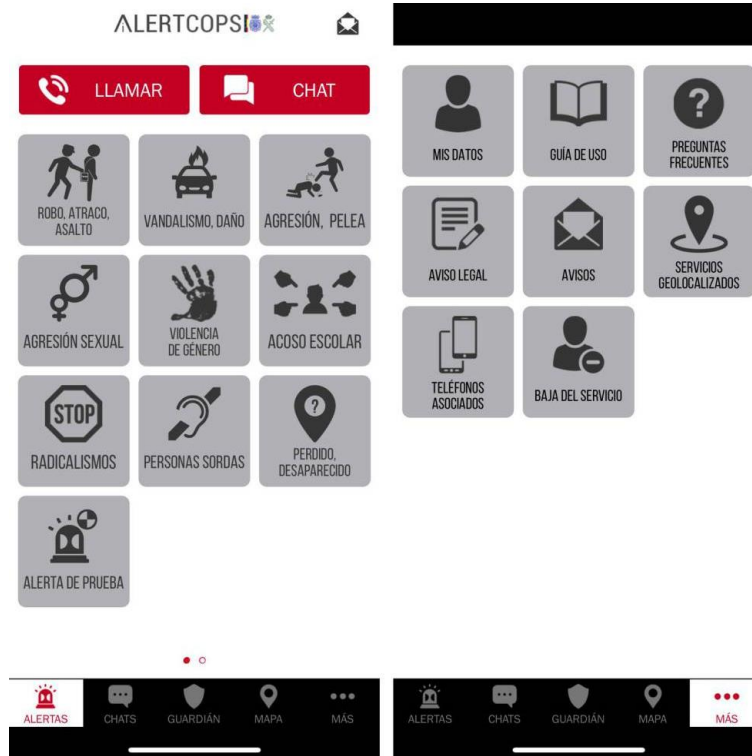
- Haurrarekin harremanetan jarri ahal izatea.
- Haurrak larrialdietako dei bat egin ahal izatea.
- Haurraren gailuan kamera edo mikrofono bat piztea ikusten edo entzuten ari dena zuzenean jasotzeko.

Gailu hauez gain, posible da, baita ere, mugikorrek dituzten haurren lokalizazioa ere lortzea. Horretarako, bai gurasoen eta bai haurren mugikorretan deskargatzen diren aplikazioak erabiltzen dira. Hauek ere, aurretik aipatutako funtzionaltasun berdinak edo oso antzekoak dituzte [9].

2.3.2. Emakumeen babeserako APPak

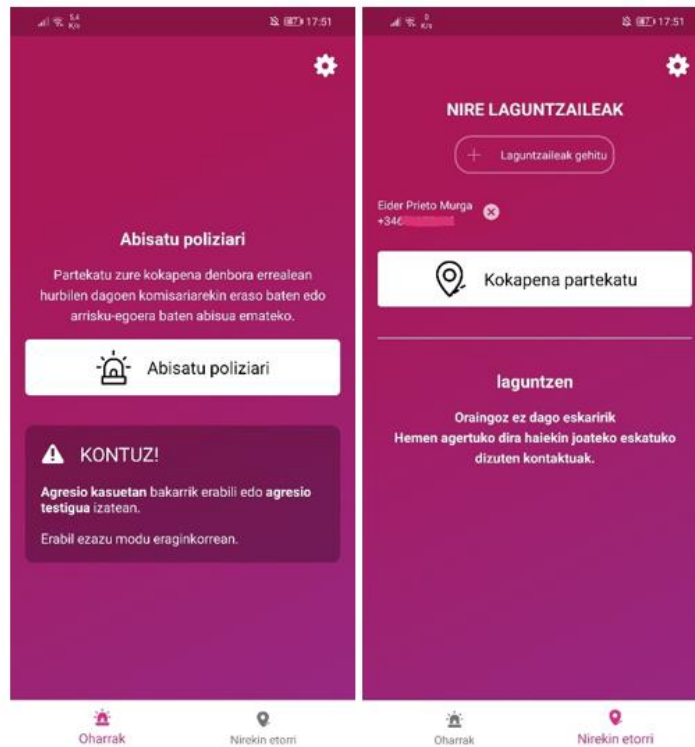
Emakumeen babeserako gaur egun existitzen diren aukerak, mugikorrean instala daitezkeen aplikazioak dira. Aplikazio hauek, lurraldeko segurtasun zerbitzuen aplikazioak izan daitezke, atal batean emakumeen babeserako funtzionalitateak dituztenak; mota honetakoa da, esate baterako, Espainia mailan, polizia nazionala abisatzeko *AlertCops* aplikazioa [10]. APP honetan, hurrengo irudian ikus daitekeenez, beste hainbat botoiren artean, sexu-erasoa jasan edota ikusi dela jakinarazi daiteke txat bidez, eta mugikorraren geolokalizazioa ere bidaliko da. Honez gain, alerta botoi bat ere badu, segundo gutxitan hainbat aldiz sakatuta, hurbilen dagoen komisaldegia abisatu

eta 10 segundoko audioa grabatuko duena, abisuarekin batera bidaltzeko, poliziak egoeraren seriotasunaren berri izan dezan.



4. Irudia. AlertCops aplikazioaren pantaila printzipala.

Beste alde batetik, emakumeentzako bereziki bideratutako aplikazioak daude. Honen adibide da hainbat euskal udalek garatutako *EraStop* APP-a [11]. Aplikazio honekin, hainbat datu pertsonal jaso ostean, erabiltzailearen geolokalizazioa lor dezake. Hainbat funtzionaltasunen artean, aipatzekoak dira Ertzaintzarekin komunikazio zuzena ematen duen botoi bat, eta mugikorrean dagoen eta aplikazioa instalatuta duen edozein kontaktu edonora laguntzeko edo laguntzeko eskatzeko aukera. Honetarako, lagunak denbora errealean lokalizazioa ikusi ahal izango du mapan, ezer arrarorik somatzekotan, abisua eman ahal izateko.



5. Irudia. EraStop aplikazioaren funtzionaltasunak

Azkenik, UC3M4Safety taldeak garatutako BINDI proiektua ere da aipatzekoa. UC3M4Safety, Emakumeen aurkako Indarkeriaren Detekzio, Prebentzio eta Borrokarako diziplina anitzeko taldea da, sei ikerketa taldek eratua, bost Madrilgo Carlos III Unibertsitatekoak (UC3M) eta bat Madrilgo Unibertsitate Politeknikokoa (UPM). Lan taldeak, Eskola Politekniko eta UC3Mko Genero Ikasketen Institutukoak dira [12].

Politeknikoko Seinalearen Teoria eta Komunikazioen Saila, Ingeniaritza Telematikoko Saila eta Teknologia Elektronikako Saila, sentsore adimendu berriekin, ezagutze algoritmoekin eta hari gabeko sareetako segurtasunean ari dira lanean, biktimak babestuko dituen proposamena sortzeko.

Indarkeria ekiditeko eta erasotik babesteko, gailuak, sistema "ikusezin" batekin, gizakiaren aldarreak, objektu arriskutsuak edota erasotzaileak detektatuko lituzke. Honetarako, sistema hau Genero Ikasketen Institutuak ebaluatzen du, eta, gainera, Elektronika Industrialeko Zentroak (UPM-CEI), makinaren algoritmo ikasketen eta sentsoreen datuen tratamenduaren inguruko esperientziarekin laguntzen du, hari gabeko sentsore-sareetan adituak dira-eta.



6. Irudia. BINDI proiektua.

Hau jakinda, BINDI, genero indarkeria ekiditeko soluzio adimenduna da, eramangarria, eskumuturreko edo iduneko bat izan daitekeena. Bi gailuz dago ostatua, eta, hauek, estetikoak izateaz gain, monitore adimendunak dira, erabiltzaileen aurkako eraso sexual edo indarkeria aurreikus dezaketenak. Gailuak, hari gabeko sistemen bidez konektatzen dira haien artean, eta baita telefono mugikorrarekin ere, Bluetooth Low Energy komunikazio protokolo fidagarri eta arinaren bidez.

Honez gain, bikoiztutako paniko botoi bat ere badauka, arrisku egoeran erabiltzaileak aktiba dezakeena. Gainera, sistemaren berri garrantzitsuena, aldagai fisiologiko garrantzitsuenen eta ahots antzemate sentsoreen erabilera da, paniko edota beldur edo estres handiak detekta ditzakeenak erabiltzailean, botoia zapaltzeko ez baldin bada gai.

BINDIren bi gailuak batera edo banaka erabil daitezke, mugikorrarekin daukaten konexioari esker; ziurtasun gehien, biak batera erabiltzeak ematen duen arren. Hala ere, bietako baten erabilerak segurtasun maila nahikoa eskain dezake, mugikorrarekin batera, betiere.



7. Irudia. BINDI proiektuaren eskumuturrekoa.

3. Lanaren helburuak eta irismena

3.1. Helburu nagusia

Lan honen helburu nagusia estaldura onik ez dagoen lekuetan ere erabil daitekeen segurtasun mekanismo bat garatzea da, erabiltzailearen kokapenaren informazioa edonon eta uneoro gordetzeko ahalmena izango duena. Horretarako, hainbat funtzionaltasun diseinatu eta garatu behar dira. Alde batetik, erabiltzailearentzako interfaze grafiko bat garatu behar da, informazioaren kudeaketa egokia egin dezan. Kontuan hartu behar da interfaze hau ahalik eta erabiltzerrazena izan behar dela, egunerokotasunez erabiltzeko pentsatuta baitago.

Beste alde batetik, mugikorretarako estaldurarik gabeko toki edo uneetan ere funtzionamendua bermatzeko, komunikazio-protokolo egokia zehaztu behar da.

3.2. Bigarren mailako helburuak

Helburu nagusiaz gain, bigarren mailako zenbait helburu ere definitu dira, jarraian deskribatzen direnak:

- **Soluzioaren diseinua.**

Soluzioa bera osatzen duten bloke funtzionalen diseinua, interfaze grafikotik hasita, biltegitratze mekanismora.

- **Soluzioaren implementazioa.**

Diseinatutako bloke funtzionalen garapen eta inplementazioa, eta baita funtzionamendu egokiaren bermatzea ere. Ondokoak dira proposatutako bloke funtzionalak:

- Gorde beharreko informazioa era egokian antolatuko duen datu-biltegi bat prestatzea.
- Erabiltzaileen lokalizazioa jaso eta biltegitratuko duen aplikazio bat garatzea.
- Erabiltzaileen datuen kudeaketarako interfaze grafiko baten garapena. Erabiltzaile hauek adituak izango ez direla hartu behar da kontuan, aurretik aipatu bezala. Interfaze honek, beren datuak ikustea ez ezik, zenbait parametro konfiguratzeko aukera eman beharko du, eta baita beste profil batzuen datuak ikusteko aukera ere, aurrerago azalduko den bezala.

- **Erabiltzaileen segurtasuna bermatzea.**

Proiektu honen nondik norakoak kontuan izanik, bi dira bermatu beharreko segurtasun motak. Alde batetik, erabiltzaileen datuen babesa dago; hau da,

diseinatutako soluzioak era egoki eta seguruan tratatu eta gorde behar ditu jasotako datuak, beti Datuak Babesteko Araudi Orokorraren (DBAO) arabera [\[13\]](#), zerbitzuak eskaintzean erabilitako datu pertsonalen segurtasuna bermatzen duena.

Gainera, erabiltzailearen datuen segurtasuna ez ezik, bere segurtasun pertsonala ere bermatu behar du, bere datuetara atzipena hautatutako pertsonak soilik izan dezaten. Puntu hau oso garrantzitsua da, batez ere, erabiltzaileak inorengandik nolabaiteko kontrolik jasaten badu, kontrol hori oraindik ere gehiago ez areagotzeko.

- **Funtzionaltasunaren bermatzea**

Behin proiektua amaituta, eta interfaze guztiak martxan daudela, zenbait proba egingo dira sistema osoaren funtzionamendu egokia bermatzeko.

4. Lanak dakartzan onurak

Proiektuaren garapenak izan dezakeen interesa ezagutzeko, izan ditzakeen onuren analisia egin da. Puntu honetan hiru dira aztertutako onura motak: teknikoak, sozialak eta ekonomikoak.

4.1. Onura sozialak

Proiektu honen onura nagusia soziala da. Izan ere, proiektuaren beraren oinarria erabiltzaileei segurtasun erraminta bat eskaintzea da, arriskuak ekiditeko, eta, behar izatekotan, arazoei konponbide errazagoa jartzeko. Esaterako, norbait galdu edo desagertzekotan, bere lokalizazioen historialera jo ahal izango da, eta gailuak martxan jarraitzekotan, bere uneko kokapenera ere bai.

Aplikazio honek, gainera, jendeari segurtasuna eskaintzeaz gain, arriskuak ere murriztuko ditu. Izan ere, Droga eta Delituaren aurkako Nazio Batuen Bulegoak (UNODC) dioenez, delituak ekiditeko erarik onena proaktiboki jardutea da [14]; hau da, biktima non dagoen jakin ahal bada, erasotzaileari gehiago kostatuko zaio erasotzea, berau harrapatzeko arriskua handiagoa baita.

4.2. Onura teknikoak

Proiektu honen onura teknikoak ugari diren arren, aipagarriena hurrengoa da: sare mugikorretatik haratago dauden beste sare batzuen erabilera frogatu da. Gaur egun zerbitzu mugikor gehienak 4G sarearen bidez eskaintzen diren arren, badaude kasuak hauen erabilera ezinezkoa dena (esaterako, jende asko pilatzen baldin bada), eta ikusi denez, soluzioa badago, eta egingarria da.

4.3. Onura ekonomikoak

Proiektu honen helburu nagusia soziala den arren, onura ekonomikoak ere badakartza. Alde batetik, eta inpaktu ekonomiko handiena suposatuko lukeena, gisa honetako proiektu batek lor ditzakeen diru-laguntzak dira. Ugari baitira proiektu sozialak garatzeko laguntza ekonomikoak eskaintzen dituzten gobernu, fundazio edota banketxeak.

Beste alde batetik, proiektuan garatu beharreko bi atalak hartu behar dira kontuan. Aplikazioari dagokionez, deskarga kopuruaren arabera, monetizatu egin ahal da, iragarkiak erabilia, edo deskargarako prezioa ezarriz.

GPS eta komunikaziorako gailuari dagokionez, ordea, produktuaren beraren salmentagatik dator irabazi ekonomikoa.

5. Aukeren analisisa

Lanaren atal honetan, proiektua garatzeko beharrezkoak diren baliabideen aukeraketa egin da. Aukeraketa hau egiteko, egindako analisi bakoitzean erabilitako irizpideak oso antzekoak izan dira, gutxi gora behera guztietarako baitira erabilgarriak. Hurrengoak dira analisi bakoitzean erabilitako irizpide amankomunak:

- **Erabiltzeko erraztasuna (%30).** Aukera erabiltzeko erraztasuna definitzen du puntu honek, ikaskuntza-kurba zein izango den ere kontuan hartuta.
- **Euskarria (%30).** Eskuragarri dagoen ikas-materiala, dokumentazio ofiziala eta eskura daitekeen laguntza hartzen ditu kontuan puntu honek.
- **Errendimendua (%20).** Aukeratutako alternatibak eskaintzen duen egokitasun eta fidagarritasuna, proiektua garatzerako orduan.
- **Kostua (%20).** Erabili beharreko materialen prezioak eta lizentziak hartzen ditu kontuan, eta baita aukeratutako alternatibaren inguruan ikastearen inguruko kostuak ere.

Analisi bakoitzari dagokion erabakia hartzeko, balio guztiak hamarretik (10) ebaluatzen dira, eta ezarritako portzentaiaren arabera ponderatu. Proiektu jakin honen ezaugarrien ondorioz, erabilerarako erraztasunari eta euskarrien eskuragarritasunari eman zaie garrantzia gehien.

5.1. Datuen biltegitratzea

Puntu hau proiektuaren garapenerako garrantzitsua da, izan ere, datuak era seguruan gordetzeaz gain, bai aplikaziotik eta bai gailutik eskuragarriak izan beharko dira. Hau horrela izan dadin, biltegitratze metodoaz gain, erabiliko den arkitektura ere ezarri beharko dira.

5.1.1. Biltegitratze metodoa

Bi dira existitzen diren datuen biltegitratze metodo nagusiak:

- **Data Lake**

Datu-gordin multzoa gordetzen duen biltegia da, oraindik datuen helburua zehaztu gabe duena. Teorian, ez dago datuak biltegitratzeko gaitasunean mugarik, eta edozein datu sartu dakioke. Hau horrela, *Data Lake*-ek aukera ematen dute datu mota eta egiturari erreparatu gabe informazioa gordetzeko.

- **Data Warehouse**

Datu-multzo egituratu eta iragazia gordetzen duen biltegia da, helburu jakin batekin tratatua. Gainera, aurretiaz ondo definituriko funtzio erregulatzaileak erabiltzen dira.

Bien arteko ezberdintasunak ezartzeko erabiltzen diren kontzeptu garrantzitsuenak hurrengoak dira [\[15\]](#):

- **Datua:** Biltegiratzeko informazioaren edukia, egoera eta egitura.
- **Prozesamendua:** Bi era daude biltegiratutako datuen trataerarako:
 - *Schema-on-write:* Beharrezkoa da datu-basearen xedea eta egitura kargaren aurretik definituta egotea, datuak egitura horrekin baliozkotzen dira-eta.
 - *Schema-on-read:* Datu-basearen xedea eta egitura kargaren aurretik definituta egotea ez da beharrezkoa, beraz, ez dago egiturarekin baliozkotzerik. Egitura datuen irakurketan definitzen da, irakurketa bakoitzean lortu beharreko datuak oinarri hartuta aldaketak egiteko malgutasunarekin.
- **Kostua:** Aukeraren kostua zein den, lizentzien eta *framework*-en arabera.
- **Bizkortasuna:** Puntu honek hainbat alderdi hartzen ditu kontuan:
 - Kontsultak egiteko behar den denbora.
 - Datuen egituran aldaketak egitea.
 - Datu kopuru handien analisia.
- **Segurtasuna:** Datuen babesa adierazten du, biltegiratze metodoaren arabera.
- **Erabiltzaileak:** Biltegi mota bakoitza gehienbat nori zuzenduta dagoen adierazten du.

Puntu hauek guztiak kontuan hartuta, hurrengo taulan ikus daiteke *Data Lake* eta *Data Warehouse* biltegiratze metodoen arteko konparaketa:

2. Taula. Data Warehouse eta Data Lake arteko konparaketa.

Irizpidea	Data Warehouse	Data Lake
Datua	Egituratua, prozesatua	Edozein (egituratua, erdi-egituratua, prozesatu gabea, gordina)
Prozesamendua	<i>Schema-on-write</i>	<i>Schema-on-read</i>
Kostua	Garestia, fidagarria	Kostu baxua
Bizkortasuna	Txikiagoa, konfigurazio ezarria	Handia, konfigurazio malgua
Segurtasuna	Garatua, fidagarria	Garapen prozesuan
Erabiltzailea	Enpresa profesionalak	Datu-zientzialariak

Proiektu honetan, gorde beharreko datuen egitura aurretik ondo definituta dago, gainera, ez da datu kopuru oso handia gordeko, eta erabiltzailea ez da datu-zientzialaria. Gainera, kontuan hartu behar da existitzen direla datu bilketa txikietarako dohaineko *Data Warehouse* -ak. Hau horrela, eta 3. Taulari erreparatuz, ikusi da *Data Warehouse* bat erabiltzea dela kasu honetan aukera egokiena.

3. Taula. Datu-biltegitratze metodoen aukeren analisi.

	Data Warehouse	Data Lake
Erraztasuna (%30)	9	7
Euskarria (%30)	9	9
Errendimendua (%20)	8	9
Kostua (%20)	10	10
Guztira	9	8.6

5.1.2. Arkitektura

Behin biltegitratze metodoa aukeratuta, ze zerbitzu eta arkitektura erabiliko diren aukeratu da. Bi aukera planteatu dira:

- **Firestore**

Google-ek eskaintzen duen plataforma mugikorra da, kalitate handiko app-ak

sortzeko hainbat funtzio eskaintzen dituena, haien artean konbinatuta edozeinen beharretara molda daitekeena. Funtzio horien artean bi datu base eskaintzen dira, bata datu-base tradizionala, eta bestea fitxategien biltegitratzea egitekoa. NoSQL lengoia erabiltzen du [16].

- **PGAdmin – PostgreSQL**

PGAdmin paketea, PostgreSQL interfazeak kudeatzeko doako eta kode irekiko tresna grafikoa da.

Proiektu honetarako Firebase plataforma erabiltzea erabaki da. Alde batetik, aplikazioetan inplementatzeko duen erraztasunagatik; beste alde batetik, datu-basearen beraren muntaia eta segurtasunaren bermatzea modu sinplean egin daitekeelako, eta, azkenik, plataforma berak erabiltzaileen autentikazioa egiteko tresnak eskaintzen dituelako, PGAdmin-a erabilia, beste pauso edo erraminta bat erabili beharko litzatekeela. Gainera, aipatzekoa da, baita ere, Firebase Raspberry-tik eskuragarria dela, kokapenari dagokion informazioa ere bertan gorde ahal izango dela bermatzen duena.

Hurrengo taulan ikus daiteke bien aukeren arteko konparaketa, hasieratik ezarritako parametroak erabilia:

4. Taula. Datu-biltegitratze arkitekturen aukeren analisia.

	Firestore	PGAdmin
Erraztasuna (%30)	9.5	8
Euskarria (%30)	10	9
Errendimendua (%20)	9.5	10
Kostua (%20)	10	10
Guztira	9.75	9.1

5.2. Aplikazioa

Proiektuan garatutako aplikazioari dagokionez, bi dira aztertu beharreko aukerak; alde batetik zer aplikazio mota garatuko den, eta beste alde batetik zer programazio lengoia erabiliko den.

5.2.1. Aplikazio mota

Hiru dira existitzen diren aplikazio motak: Web aplikazioak, aplikazio natiboak eta aplikazio hibridoak; hala ere, kasu honetan web aplikazio bat garatzeak ez du zentzu handirik, gailu mugikorretan erabiltzea baita asmoa. Hau horrela, bi aukera dira aztertu beharrekoak:

- **Aplikazio natiboa**

Sistema eragile mugikor bakar baterako garatutako aplikazio mugikorra da. Esaterako, Android-erako garatutako aplikazio bat ezin da iOS-en erabili, eta alderantziz. Gehienbat, errendimendua eta API ezberdinen erabilera nabarmentzen dira aplikazio natiboen abantailen artean. Hala ere, desabantaila nagusia, sistema eragile bakoitzeko aplikazio ezberdin bat garatu behar dela da.

- **Aplikazio hibridoak**

Sistema eragile ezberdinetan erabil daitezkeen aplikazioak dira. JavaScript, CSS eta HTML lengoaiak erabilia garatzen dira, eta, funtsean, web aplikazioak dira, edukiontzi natibo baten barruan sartuta. Garatzeko errazak dira, baina zenbait API lortzea zailagoa izan daiteke, ez dute errendimendu onik eta diseinuan arazoak egon daitezke, plataforma batetik bestera alda daitezkeelako.

Hau guztia kontuan hartuta, bi aukerak erabiltzea posible dela ondoriozta daiteke. Hala ere, proiektuaren garapena hobetzeko asmoz, Android aplikazio natibo bat garatzea erabaki da, batez ere erabili beharreko APIak lortzeko erraztasunagatik, eta eskuragarri dagoen informazio gehigarria aintzat hartuta. Proiektua merkaturatu nahi izatekotan, erabiltzaile gehiagorengana heldu ahal izateko, hibridora pasatu edo bigarren aplikazio bat garatu beharko litzateke, lehenengoaren oso antzekoa.

Hurrengo taulan ikus daiteke egindako konparaketa, hasierako parametroak aztertuta:

5. Taula. Aplikazio moten aukeren analisia.

	Aplikazio natiboa	Aplikazio hibridoak
Erraztasuna (%30)	9	7
Euskarria (%30)	10	9
Errendimendua (%20)	9	7
Kostua (%20)	10	9
Guztira	9.5	8

5.2.2. Lengoia

Bi aukeren artean egin da alternatiben analisia, programazio lengoaiari dagokionez:

- **Java**

Existitzen diren programazio lengoaietatik, ezagunenetako bat da, GitHub-en hirugarren erabiliena. Objektuetara orientatutako lengoia da. Sinplea eta sendoa, Android aplikazioetarako oso egokia.

- **Kotlin**

Java programazioa produktiboagoa izateko sortu zen lengoia da, JVM (*Java Virtual Machine*) eta Android-en erabiltzeko pentsatuta. Orokorrean, kode gutxiago idatzi behar da, baina lengoia askoz berriagoa da, horrek dakartzan eragozpenekin; esate baterako, dokumentazio eta erabiltzaile gutxiago edota desagertzeko arrisku gehiago.

Hau guztia kontuan hartuta, Kotlin erabiltzea erabaki da, batez ere, Java klaseak ere erabiltzea onartzen duelako, eta zenbait APIren erabilera errazten duelako.

Hurrengo taulan ikus daiteke egindako konparaketa, nahiz eta kasu honetan bi lengoaien erabilera guztiz onargarria zatekeen:

6. Taula. Aplikazioaren programazio lengoaien aukeren analisia.

	Java	Kotlin
Erraztasuna (%30)	9.5	10
Euskarria (%30)	10	9.5
Errendimendua (%20)	9	9.5
Kostua (%20)	10	10
Guztira	9.65	9.75

5.3. Komunikazio protokoloa

Uneoro erabiltzailearen lokalizazioa bidaltzen duen gailuari dagokionez, informazio hori nola transmitituko den erabaki behar da, hau da, zein komunikazio teknologia edo zein komunikazio protokolo erabiliko den.

Komunikazio protokoloari dagokionez, aipatzekoa da, lehenik eta behin, 4Garen erabilera hasieratik utzi dela alde batera, izan ere, beronen estaldurarik ez dagoen

tokietan ere erabili ahal izatea da helburua; esate baterako, jende asko pilatzen den jai edo kontzertu giroan. Hau horrela, hiru dira planteaturiko protokolo aukerak: LoRa, Sigfox eta NB-IoT. Hirurak dira LPWAN (*Low Power Wide Area Network*) sareak, distantzia luzeak, efizientzia energetiko handiak eta kostu baxukoak. Hala ere, badaude haien artean ezberdintasunak [\[17\]](#):

- **Sigfox**

LPWAN sare operadore bat da, bere patentatutako teknologien arabera, *end-to-end* IoT konektibitatea eskaintzen duena. Base estazio propioak ditu, IP sare baten bitartez zerbitzarietara konektatzen direnak. Lizentziarik gabeko bandetan funtzionatzen du.

Amaierako gailuak BPSK (Binary Phase-Shift Keying) modulazioaren bidez konektatzen dira base estaziora, *ultra-narrow band*-ean (100 Hz). Horrek efizientzia handia, zarata maila txikia, kontsumo baxua dakartza. Hala ere, *uplink*-ak 12 byteko 140 mezu onartzen ditu egunean, eta *downlink*-ak 8 byteko 4 mezu. Aipatzekoa da, baita ere, fidagarritasun handiko protokoloa dela, bukaerako gailutik bidaltzen den mezu bakoitza hiru aldiz transmititzen baita, hiru maiztasun kanal ezberdinetan, mezuaren harrera bermatzeko.

- **LoRa**

Maila fisikoko teknologia da, seinaleak *chirp spread spectrum* (CSS) modulazio teknika propioa erabiliz modulatuak dituena. Komunikazioa bidirekzionala da, eta seinaleek zarata maila baxua dute, interferentziekiko erresilientzia handia eta detektatzeko zailak dira. Lizentziarik gabeko bandetan funtzionatzen du.

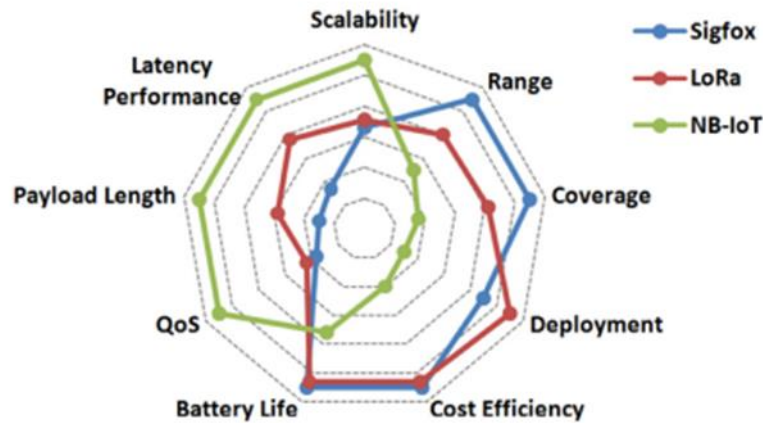
Erabiltzen den hedapen-faktorearen arabera, datu-tasa gero eta txikiagoa da. Hala ere, mezuen luzera maximoa 243 bytekoa da. Aipatzekoa da, baita ere, hedapen-faktore ezberdinekin transmititutako seinale ezberdinak hartzailer berean jaso daitezkeela.

- **NB-IoT**

GSM eta LTEekin batera existitzen den *Narrow Band IoT* teknologia da, lizentziadun bandetan funtzionatzen duena. Hau horrela, eta 200 kHz-ko banda zabalera okupatzen duela jakinda, hiru eratarik funtziona dezake: *Stand-alone operation* (GSM maiztasun-bandetan), *Guard-band operation* (LTE garraiatzaileen guardako bandek erabili gabe uzten dituzten bandetan) edo *In-band operation* (LTE garraiatzailearen *resource-block*-etan).

LTEn dago oinarrituta eta 100 k gailu konekta dakizkioke zeldako. *Data rate*-a mugatuta dago, 200 kbps *downlink*-ean eta 20 kbps *uplink*-ean. Mezu bakoitzaren tamaina 1600 byte da, gehienez, eta bateria bakoitzak 10 urte inguru iraun ditzake.

Hau guztia kontuan hartuta, hiruren arteko konparaketa grafikoa, jarraian ikusten den irudikoa da [17]:



8. Irudia. Sigfox, LoRa eta NB-IoT arteko konparaketa.

Kasu honetan, garrantzia gehien mezuen luzerak eta bidalketa maiztasunak izan dute, baita eskalagarritasunak eta kostuak ere; izan ere, Sigfox erabiltzeak kostu handiagoak suposatuko lituzke. Bere aldetik, LoRa-rekin egin daitezkeen eragiketak mugatuagoak dira. Jarraian ikus daiteke hasieratik aurkeztutako parametroekin egindako analisia:

7. Taula. Komunikazio protokoloen aukeren analisia.

	Sigfox	LoRa	NB-IoT
Erraztasuna (%30)	7	9	9
Euskarria (%30)	7	9	9
Errendimendua (%20)	8	6	8
Kostua (%20)	6	8	9
Guztira	7	8.2	8.8

6. Arriskuen analisia

Puntu honetan, proiektuaren garapenaren etapa ezberdinetan ager daitezkeen arriskuak aztertu eta baloratu dira, proiektua bera atzeratzea edo helburua lortzea galaraz dezaketenak, hain zuzen ere. Hau horrela, gomendagarria da arrisku bakoitza banan-banan aztertzea, baten bat suertatzekotan, neurri egokiak hartzeko, edo, behintzat, prebentzio-neurriak izateko.

6.1. Arriskuak

Jarraian, aztergai diren arrisku guztiak azaldu dira. Bakoitza era egokian baloratzeko, gertaera-probabilitatea eta proiektuan izango lukeen eragina aztertu dira.

A1 – Atzerapenak

Proiektuan gerta daitezkeen arrisku nagusietako bat, garatu beharreko atazaren batean epeak ez betetzea da, ostean egin beharrekoetan ere atzerapenak suposatuko lituzkeena. Hau horrela balitz, proiektu osoak ere atzerapena jasan lezake. Beraz, ondokoak dira gertatzeko probabilitatea eta eragina:

- Gertatzeko probabilitatea: % 70.
- Eragina: % 30.

Gisa honetako proiektu batean atazaren bat epez kanpo amaitzea ez da arraroa, baina kontrakoa gertatzea ere normala da. Sarritan, programatzerako orduan denborak ondo planifikatzea ez da erraza, uste baino arazo gehiago edo gutxiago agertu ahal baitira.

Hasieratik era egokian planifikatuta eta babes-epeak erabilia, suposa lezakeen eragina gutxituko litzateke. Hala ere, eragina bera ertaina da, eta gertatzeko probabilitatea nahiko altua.

A2 – Diseinuan akatsak

Diseinuaren edozein puntutan akatsen bat egoteak arazo larriak ekar ditzake, denboraz detektatzen ez badira. Esaterako, datu basearen egitura zuzena ez izatekotan, datuak galdu daitezke, edo aplikazioaren diseinua ondo planteatuta ez baldin badago, egindako lan guztian aldaketak egin behar izatea gerta daiteke, denbora asko galduz. Hau horrela, ondokoak dira gertatzeko probabilitatea eta eragina:

- Gertatzeko probabilitatea: % 30.
- Eragina: % 90.

Ikus daitezkeenez, diseinu akatsen bat izateak proiektuan eragin handia izango luke, baina gertatzeko probabilitatea ez da handia.

A3 – Arazo teknikoak

Arrisku honen barnean sartzen dira proiektu osoan zehar ager daitezkeen arazo tekniko guztiak, bai garapenean sortutako barne-arazoak edo kontrolatu ezin diren kanpo-arazoak.

Adibidez, kontrolatu ezinako arazo teknikoen barruan egongo litzateke, erabilitako APIren batek euskarria izateari uztea. Ordea, barne-arazoak lirateke, esaterako, elementuren baten haustura, edota gailu ezberdinen arteko integrazio falta.

Guzti hau kontuan hartuta, ondokoak dira gertatzeko probabilitatea eta eragina:

- Gertatzeko probabilitatea: % 70.
- Eragina: % 70.

Kasu honetan, gertatzeko probabilitatea altua da, eta proiektuan eragin handia izango luke, baita ere.

6.2. Konparaketa

Jarraian ageri den irudian arriskuen probabilitate-eragin matrizea ageri da. Matrize honekin, aurretik azaldutako arriskuak era grafikoagoan konpara daitezke, proiektuan eragin handiena izan dezaketenak zeintzuk diren baloratu ahal izateko.

		Eragina				
		% 10	% 30	% 50	% 70	% 90
Probabilitatea	% 10					
	% 30					A2
	% 50					
	% 70		A1		A3	
	% 90					

9. Irudia. Arriskuen probabilitate-eragin matrizea.

Taula hau ikusita, esan daiteke A3 arriskua dela proiektuaren garapenean garrantzia handiena eman beharko zaiona; hau da, arazo teknikoak, kontrolatu ahal direnak behintzat, ahalik eta gutxien izatea izan beharko da helburua. Gainontzeko arriskuak txikiagoak dira, atal hori eta laranjan kokatzen dira-eta; nolana ere, hauek saihesten saiatzea ere ezingo da alde batera utzi.

6.3. Kontingentzia plana

Aurretik aipatutako arriskuak ahalik eta gehien saihesteko, garrantzitsua da bakoitzarentzat kontingentzia plan bat planteatzea. Hau horrela, ondokoak dira arrisku bakoitzarentzat ezarritako neurriak:

A1 – Atzerapenak

Proiektuaren epe mugan atzerapenik egon ez dadin, aurretik aipatu bezala, ezinbestekoa da hasieratik planifikazio on bat izatea, egin beharreko atazak ondo definituz eta aurretiaz, ataza bakoitzean egin beharrekoa gainbegiratuta. Baliteke kanpo-faktoreak direla medio, atzerapenen bat suertatzea, baina, hala ere, lana era egokian antolatu bada, aukera egon beharko litzateke hau arintzeko, beste lan batzuk arinduta, edota denbora- babes-tarteak aprobetxatuta.

A2 – Diseinuan akatsak

Diseinuan akatsik egongo ez dela ziurtatzeko, lehenengo eta behin, planteatutako diseinua aztertu behar da, aurrerago akatsik suposatuko ez duela bermatzeko. Honez gain, garrantzitsua da, baita ere, garatutako ataletako bakoitza amaituta, egindakoa, eta aurretik frogatutako guztia, berriro ere konprobatzea, inolako arazorik ez dagoela bermatzeko, eta, egotekotan, prozesuan ahalik eta goizen atzeman ahal izateko,

A3 – Arazo teknikoak

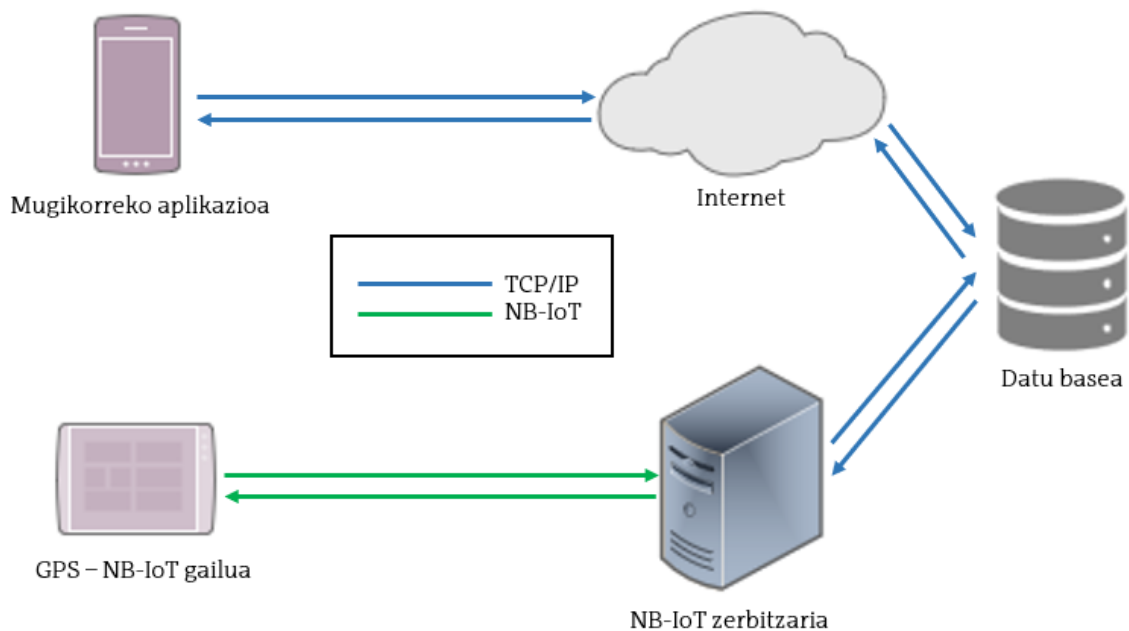
Mota honetako arazoak saihesteko, erabilitako tresnen egoera eta funtzionamenduaren jarraipena egiteaz gain, aurretiaz ziurtatuko da, ahal den heinean, erabilitako erramintak proiektuak irauten duen denboran, behintzat, eskuragarri egongo direla. Inolako apurketarik baldin balego, ordea, ordezkapen bat izatea bermatuko da.

7. Proposatutako irtenbidearen deskribapena

Puntu honetan, proiektua garatzeko diseinatutako soluzioa deskribatzen da zehazki. Horretarako, lehenik eta behin, planteatutako soluzioa bera zein den azaltzen da, amaierako produktuaren funtzionaltasun guztiak barne. Ondoren, diseinu honetan parte hartzen duten hiru atal nagusien diseinuak aztertzen dira, banan-banan. Hau da, datu-basea, aplikazioa eta NB-IoT gailu eta sarea, jarraian ikusiko den bezala.

7.1. Planteatutako soluzioa

Planteatutako soluzioa azaltzeko, 10. Irudiko eskema erabiliko da.



10. Irudia. Planteatutako soluzioaren eskema orokorra.

Eskema honetan, hiru dira ezberdindu beharreko atalak:

- 1. Datu basea.** Datuak gordeko dituen biltegia. Gainontzeko atal biek datu base berdinen kontra bidaliko dituzte eskaerak. *Firestore* datu base bat izango da, zuzenean autentikatzeko aukera emango duena, erabiltzaileen datuak modu seguruan gordeta.
- 2. Mugikorrerako aplikazioa.** Datu basera Internet bidez konektatzen den Android aplikazioa, Kotlin programazio lengoia erabilia garatutakoa. Aplikazio honetan, email eta pasahitz bidez autentikatutako erabiltzaile bakoitzak profil bat izango du. Profil horren bidez, hainbat dira egin ahalko diren eragiketak:
 - **Lokalizazio seguru bat ezarri**, esate baterako etxea edo lantokia izan daitekeena.

- **Uneko lokalizazioa gordetzen** hasteko agindua eman. Hau bi erataraino egin daiteke:
 - Hasierako pantailan dagoen botoi baten bidez.
 - Lokalizazio segurutik distantzia jakin bat urrunduta. Distantzia hau konfiguragarria izango da.
- Momentura arteko lokalizazioen **historiala ikusi**, mapa batean.
- **Lagunak gehitu eta ezabatu**. Lagun hauek erabiltzailearen lokalizazioen historiala ikusi ahalko dute, eta alderantziz; hau da, erabiltzaileak lagunak mugimenduak jarraitu ahal izango ditu.

3. GPS – NB-IoT gailua. Gailu hau erabiltzaileak gainean eraman beharko lukeen IoT gailu txiki bat izango litzateke, beharrezko funtzionaltasunak betetzen dituen. Hala ere, kontzeptu-proba moduan, proiektua aurrera eramateko, Raspberry bat izango da, GPS eta NB-IoT funtzionaltasunak ematen dizkion *hat* edo plaka bati lotuta. Gailu honen helburua, erabiltzailearen lokalizazioa datu basera uneoro bidaltzea da. Datu bidalketa hau, Internet bidez izan beharrean, aplikazioaren kasuan bezala, NB-IoT bidez egin da.

7.2. Datu basearen diseinua

Datu basearen diseinurako, Google-en *Firebase* plataforma digitala erabili da. Plataforma hau web aplikazioak eta aplikazio mugikorak era azkar eta efizientean garatzeko dago diseinatuta. Modulu ezberdinak inplementa ditzake, horrela aplikazioen beren errendimendua hobetzeko asmoz. Gainera, hodeian dago gordeta, eta Android, iOS eta web teknologietarako dago eskuragarri.

Firebase-k eskaintzen dituen moduluak dagokienez, hiru talde handitan bereiz daitezke [\[18\]](#):

- **Konpilaziorako moduluak.** Aplikazioaren *backend*-a zerbitzariak kudeatu gabe abiarazteko aukera ematen du, eta baita aplikazioa hazten doan heinean eskalatzeko aukera ere. Hurrengoak dira modulu garrantzitsuenak:
 - *Authentication*. Aplikazioetan erabiltzailea autentikatzeko erabiltzen den moduluak. Kasu honetan, zuzenean *backend* zerbitzuak eskaintzen ditu, erabiltzaileak hainbat mekanismo erabilita autentikatzeko: pasahitzak, telefono zenbakiak, Google, Facebook, Twitter...
 - *Realtime Database*. Hodeian gordetako NoSQL datu base bat da. Datuak JSON formatuan gordetzen dira, eta konektatuta dauden bezero guztiekin denbora errealean sinkronizatzen dira. Gainera, aplikazioak interneterako konexiorik ez badu ere martxan mantentzen da.
 - *Cloud Storage*. Aplikazioan erabiltzeko edo aplikazioaren bidez erabiltzaileak sortutako eduki astunagoa (argazkiak edo bideoak) gorde

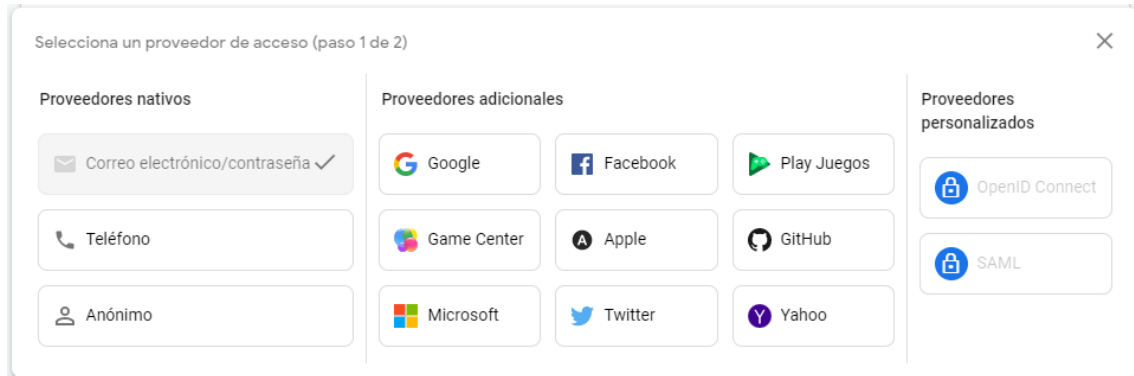
edo emateko balio duen biltegitratze zerbitzua. Modulu honek, gainera, automatikoki eskalatzen du, beharrianen arabera.

- **Merkaturatze eta gainbegiratze moduluak.** Aplikazioaren merkaturatzea kontrolatzea ahalbidetzen du, eta baita bere funtzionamendu egokia gainbegiratzea ere. Hurrengoak dira modulu garrantzitsuenak:
 - *Crashlytics.* Denbora errealean erabiltzaile ezberdinen aplikazioetan gertatzen diren akatsei buruzko informazioa gordetzen duen erraminta, hauek ahalik eta lasterren konpondu ahal izateko. *Dashboard* ezberdinak ditu, erroreak nola gertatu diren jakiteko, eta baita kodearen zein lerrotan gertatu diren gordetzeko.
 - *Performance Monitoring.* Denbora errealean aplikazioaren errendimenduari buruzko informazioa gordetzen duen erraminta.
- **Parte-hartzerako moduluak.** Erabiltzaileek aplikazioa nola erabiltzen duten ulertuta, erabiltzaile kopurua handitzeko aukera ematen duten erramintak. Esaterako, erabiltzaile mota ezberdinentzako pertsonalizatuta, funtzionaltasun berriak probatzeko esperimenduak eginda... Hurrengoak dira modulu garrantzitsuenak:
 - *Analytics.* Aplikazioen erabilerari eta erabiltzaileen parte-hartzeari buruzko estatistikak eskaintzen dituen erraminta da. SDK soilik integratuta, orotariko informazioa jaso daiteke.
 - *Remote Config.* *Firebase Console*-tik aplikazioaren itxura eta portaera aldatzea ahalbidetzen duen erraminta, aurretik zenbait parametro ezarrita.

Kasu honetan, bi dira erabilitako erramintak. Alde batetik, autentikaziorako modulua (*Authentication*), eta, beste alde batetik, NoSQL datu basea (*Realtime Database*). Hau horrela, hurrengo bi ataletan azaldu da bakoitza nola diseinatu den.

7.2.1. *Firebase Authentication*

Autentikazioari dagokionez, kasu honetan sarbide-metodoa aukeratu behar izan da. Hain zuzen ere, email eta pasahitz bidezko sarbidea aukeratu da. Hau egiteko, *Firebase*-ren kontsolatik, *Authentication* atalean, metodoa gehitu da, hurrengo irudian ikus daitekeenez.



11. Irudia. *Firebase-n autentikazio metodoa.*

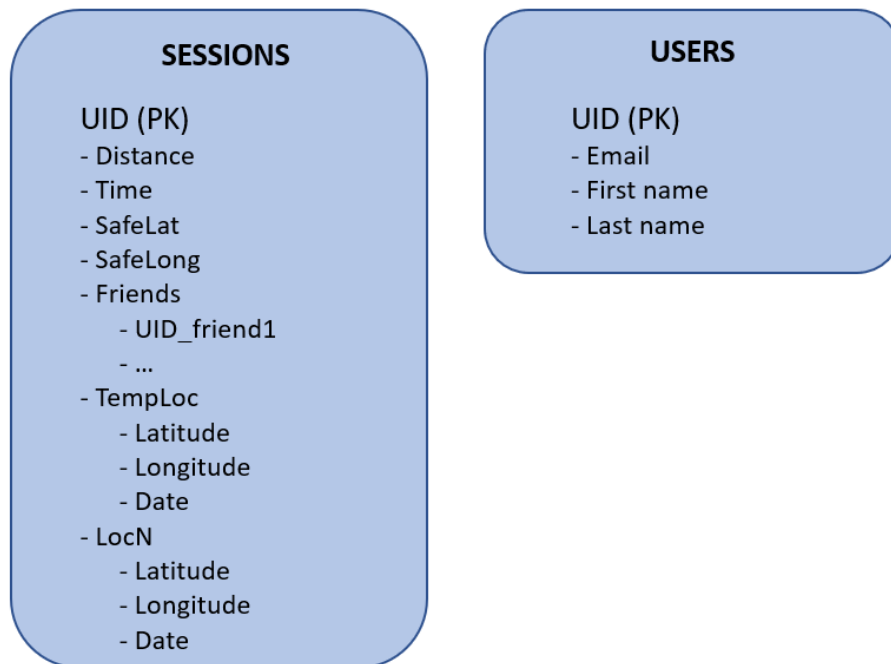
Metodo hau erabiltzeak, proiektuaren datu-basera bai aplikaziotik eta bai Raspberry-tik sartzeko aukera emango du. Gainera, aplikazioan, gakoa ahaztekotan, aldatu ere egin ahalko da. Hau egiteko, mezu elektronikoa bat bidaltzen da kontua sortzeko erabiltzeko helbidera, eta bertan aldaketa egiteko link-a jasotzen da.

7.2.2. *Firebase Realtime Database*

Datu base honetan, erabiltzaileei buruzko informazioa gorde da. Hau era egokian egin ahal izateko, bi dira egindako lanak:

1. Datu basearen **arkitektura** definitu.
2. Datu basearen **arauak** definitu.

Arkitekturari dagokionez, bi taula bereizi dira; alde batetik, erabiltzaileei dagozkien datuak gordetzen dituen taula (email-a, izena eta abizena), eta beste alde batetik, aplikazioaren eta gailuaren funtzionaltasunak betetzeko beharrezkoak diren sesioen datuak. Banaketa hau, horrela egin da, alde batetik taulei egitura zuzenagoa emateko, erabiltzaileak eta sesioak banatuta, eta, beste alde batetik, lagunen funtzionaltasuna era egokiagoan garatzeko; hau da, konputazionalki, errazagoa izateko. Hau horrela, hurrengoa da diseinatutako arkitektura:



12. Irudia. Datu basearen arkitektura.

Aplikazioaren diseinuaren atalean azalduko da zehazki eremu bakoitza zertarako erabili den.

Datu basearen arauak definitzeari dagokionez, *Firestore*-k berak badu datuen erabilerarako arauak kudeatzeko erraminta. Hau horrela, taula bakoitza idatzi eta irakurtzeko arauak zeintzuk diren definitu da:

- **Users taula.** Datuen irakurketa eta idazketa autentikatutako erabiltzaileek baino ezin dute egin. Gainera, datu berriak idazteko, sarreraren PKa, erabiltzailearen IDa (UID) izan behar da, eta sarrera horrek bi eremu izan behar ditu beteta: email-a eta izena, biak String motakoak. Jarraian ikus daitezke taula honen arauak, azaldutako baldintzak betetzeko:

```

"users":{
  ".read": "auth != null",
  ".write": "auth != null",
  "$uid":{
    ".validate":"newData.hasChildren(['email','firstName'])",
    "email":{".validate":"newData.isString()"},
    "firstName":{".validate":"newData.isString()"}
  },
  ".indexOn": ["email","firstName"]
}
  
```

13. Irudia. Users taularen arauak.

- **Sessions taula.** Aurreko taulan betetako baldintza berak aplikatu dira; hau da, edozein datu idatzi edo irakurtzeko, autentikatutako erabiltzaile batek egin behar du. Honez gain, sarreraren IDa, erabiltzailearen IDa izan behar da, eta sarrera berri bat gehitzerakoan, distantzia, denbora eta posizio segurua gehitu behar dira. Jarraian ikus daiteke baldintza hauek betetzeko konfiguratutako arauen zerrenda:

```

"sessions":{
  ".read": "auth != null",
  ".write": "auth != null",
  "$suid":{
    ".validate":"newData.hasChildren(['distance','time','safeLong','safeLat'])",
    "distance":{".validate":"newData.isNumber()"},
    "time":{".validate":"newData.isNumber()"},
    "safeLong":{".validate":"newData.isNumber()"},
    "safeLat":{".validate":"newData.isNumber()"}
  },
  ".indexOn": ["distance","time","safeLong","safeLat"]
}

```

14. Irudia. Sessions taularen arauak.

7.3. Aplikazioaren diseinua

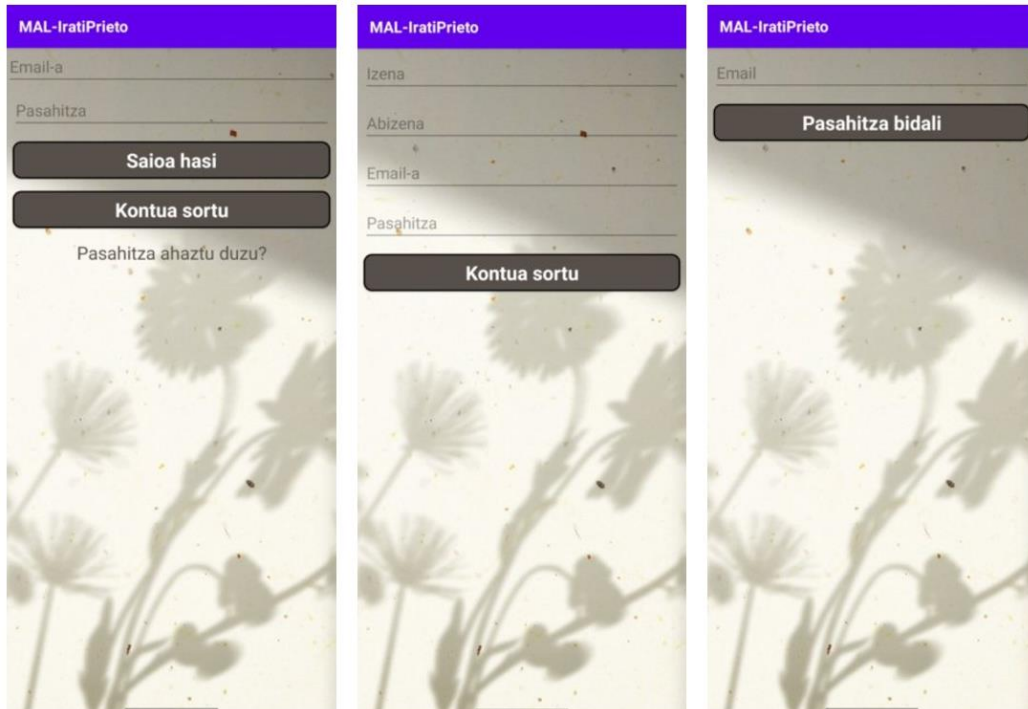
Proiektu honetan garatutako aplikazioa, Android gailuetan soilik erabil daitekeen APP bat da, izan ere, soinean eramateko gailu batekin erabiltzeko pentsatuta dagoenez, ez dauka zentzurik web bertsio bat sortzeak, aukeren analisisan azaldu den legez.

Proiektu honetarako sortutako Android aplikazioa zelan egituratu eta garatu den hobeto ulertzeko, interesgarria da, lehenik eta behin, aplikazioa bera ezagutzea. Hori dela eta, puntu hau bi ataletan banatuko da; lehen puntuan funtzionamendua bera ikusiko da, dauzkan pantailak erakutsita, eta, bigarrenean, garapen prozesua nolakoa izan den azalduko da.

7.3.1. Aplikazioaren funtzionamendua

Jarraian, banan-banan aplikazioak dituen erlaitz guztiak ikusiko dira, bakoitzean inplementatuta dauden funtzionalitateak azalduz:

1. **Autentikazio-erlaitza.** Aplikazioa zabaltzean ateratzen den lehen pantaila da. Bertatik, hiru ekintza posible daude: email eta pasahitzarekin aplikaziora sartu, pasahitza ahaztekotan, berri bat lortu, eta, azkenik, konturik ez izatekotan, berri bat sortu.



15. Irudia. Aplikazioaren autentikazio-erlaitza, pasahitza berreskuratzea eta kontua sortzea.

2. **Hasierako erlaitza.** Hasierako menua duen orria da. Bertatik, aplikazioak eskuragarri dituen funtzionaltasun guztietara hel daiteke: posizioen mapa (*Mapa ikusi*), ezarpenak, eta lagunen historialak. Gainera, orri pantaila honek edozein momentutan erabiltzailean posizioa gordetzen hasteko botoia du.



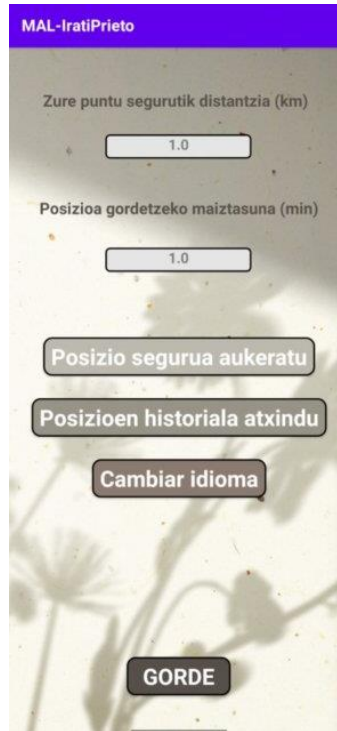
16. Irudia. Aplikazioaren hasierako menuaren erlaitza.

3. **Erabiltzailearen posizioen historiala.** Orri honek Google Maps-en API bat erabiltzen du, datu basean erabiltzaileak gordeta dituen posizio guztiak ikusi ahal izateko. Horretarako, pantailan bertan agertzen den botoia sakatuta, mapa bera hurbiltzen du lokalizazio guztiak dauden zonaldera; hauek gorritz agertuko dira. Mapa honetan, gainera, erakusle more batekin, posizio segurua zein den ere erakusten da. Azkenik, aipatzekoa da, erakusle bakoitzean koordinatuak eta data ikus daitezkeela, gainean sakatuz.



17. Irudia. Aplikazioaren posizioen historialaren mapa.

4. **Ezarpenak.** Erlaitz honetan aplikazioak dituen funtzionaltasun konfigurarri guztiak alda daitezke. Hau da, posizio segurua ezarri, posizio segurutik zenbat distantzia urruntzean hasiko den posizioa gordetzen, zenbat minuturo gordeko den posizioa eta posizioen historiala ezabatzea. Honez gain, aplikazioaren hizkuntza ere alda daiteke orri honetatik; euskara eta gaztelania daude momentu honetan eskuragarri.



18. Irudia. Aplikazioaren ezarpenen erlaitza.

5. **Posizio segurua aukeratzea.** Pantaila honetan mapa bat agertuko da, erakusle batekin. Erakusle hau sakatuta mantenduta, mapatik mugitu daiteke. Behin toki egokian kokatu denean, *Gorde* botoiarekin posizio segurua gordeko da.



19. Irudia. Aplikazioaren posizio segurua gordetzeko erlaitza.

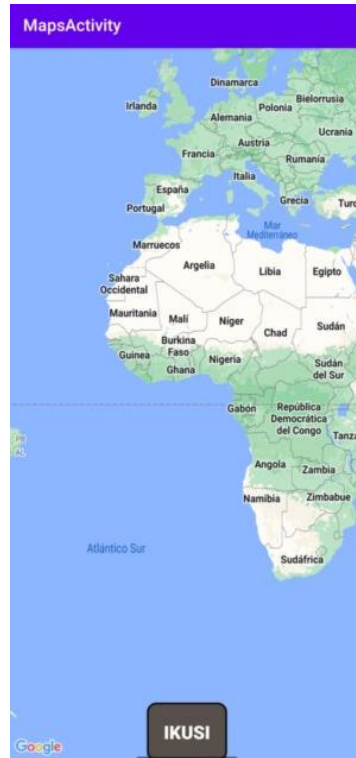
6. **Lagunen zerrenda.** Erlaitz honetan gehienez bost lagun gehitzeko aukera ematen da. Gehienez 5 lagun izatea erabaki da, egoera edo momentu ezberdinetan baten bat libre egoteko nahikoa delako, baina ez segurtasun-arrakala bat izateko beste; hau da, lagun bat zerrendan egoterik nahi ez baldin bada, ez dira nahikoa zerrendan daudela ez konturatzeko.

Lagun berri bat gehitzeko, bere email-a sartu eta gehitzeko botoiari eman behar zaio; kontuan hartu behar da, lagun horrek aplikazioan kontu bat izan behar duela sortuta, bestela ezin izango da gehitu. Ordea, lagun bat ezabatzeko, email-aren ezkerrean ageri den gurutzearekin kentzen da. Lagun bat gehitu denean, bere historiala ikusteko aukerei dagokienez, bi aukera daude: sartutako email-ak erabiltzailearen email-a bere lagun artean badu, eskuinaldean botoi bat agertuko da, bere posizioa ikusi ahal izateko. Bestalde, erabiltzailea ez baldin badago lagun artean, ez da inolako botoirik agertuko eta ezin izango da historiala ikusi. Hau horrela garatu da, erabiltzaileak kontrola izan dezan bere posizioa nork ikus dezakeen inguruan.



20. Irudia. Aplikazioaren lagun zerrendaren erlaitza.

7. **Lagunen historiala.** Norbere historialaren erlaitza bezalakoa da orri hau. Posizioak ikusteko botoiarekin mapan agertuko dira lagunaren posizio guztiak. Gainera, kasu honetan ere, posizio hauen markagailuetan momentuaren eta koordinatuen inguruko informazioa ikusi ahal izango da.



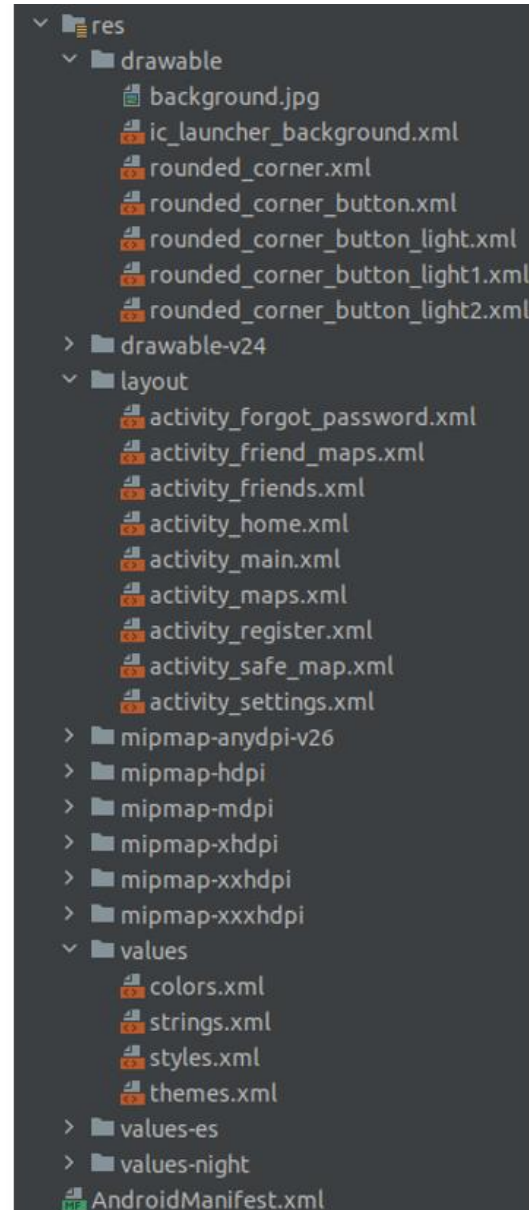
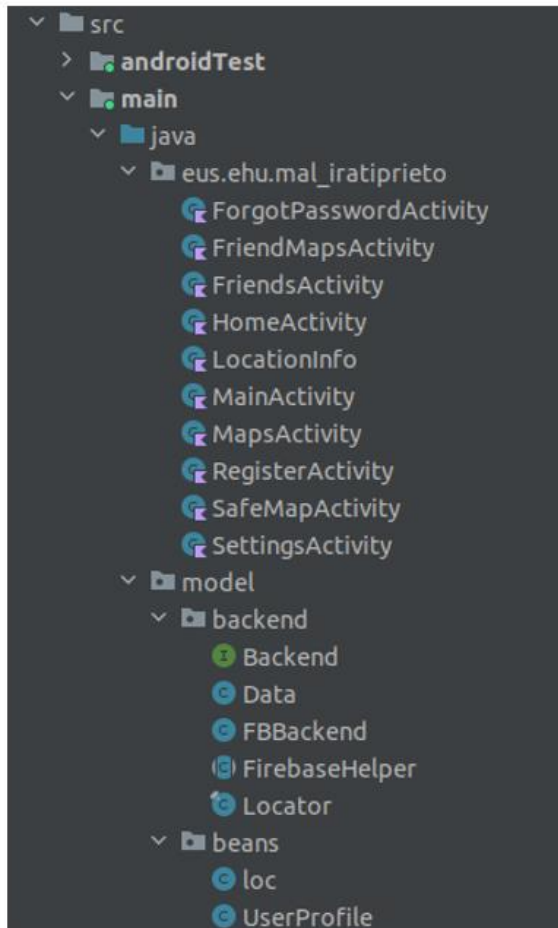
21. Irudia. Aplikazioaren lagunen historiala ikusteko maparen erlaitza.

7.3.2. Aplikazioaren garapena

Aplikazio hau garatzeko egindako antolaketa 22. irudian ikus daitekeena da. Alde batetik, Kotlin eta Java programazio lengoaiak erabilia garatutako jarduera edo *Activity*-ak daude, eta beste alde batetik baliabide edo *Resource*-ak dituzten xml fitxategiak.

Android aplikazioetan, Jarduera edo *Activity*-ek Erabiltzailearen Interfazeak (UI – *User Interface*) irudikatzen dituen leihoak hornitzen dituzte. Normalean, jarduerak pantaila osoa betetzen dute, baina posible da txikiagoa izatea eta beste orri batzuen gainean ere agertzea. Hau horrela, normalean, jarduera bakoitzak erlaitz ezberdin bati egiten dio erreferentzia eta bi atalez dago osatua: atal logikoa eta atal grafikoa. Atal logikoari dagokionez, java edo, kasu honetan, .kt fitxategi bat da, *activity*-a manipulatzeko eta berarekin komunikatzeko kodea duena. Ordea, alde grafikoa, XML fitxategi bat da, HTMLk dituen antzeko etiketak erabiliz, pantailaren diseinua egiteko erabiltzen dena.

Hau horrela, *activity* edo erlaitz bakoitzarentzat Kotlin fitxategi bat eta XML fitxategi bat sortu dira. Horrez gain, datu basearekin komunikazioa ezartzeko funtzio guztiak batzeko, eta kode osoan erabili ahal izateko aldagai globalak sortzeko, zenbait klase gehiago ere sortu dira, *model* karpeta daudenak.



22. Irudia. Aplikazioaren kodearen antolaketa.

XML fitxategiei dagokienez, erlaitzen diseinuak dituztenez gain, zenbait estilo global definitu dira, *drawable* karpetan gorde direnak. Azkenik, hizkuntza aldatzeko funtzionaltasuna inplementatzeko, *strings.xml* fitxategian, bi hizkuntzetako String-ak gorde dira, aukeraketa egiterakoan, multzo bat edo beste aplikazioan kargatzeko.

Hainbat puntu dira aztertzeko garrantzitsuak: datu basearekin komunikatzeko erabilitako modua, orri ezberdinak eta zenbait konfigurazio, jarraian banan-banan ikusiko direnak.

Datu basearekin komunikazioa

Aplikazioa Firebase datu basearekin konektatzeko, Android Studiok zuzenean ematen du aukera automatikoki sortutako aplikazioa Firebase kontu batera lotzeko. Behin hori eginda, Google zerbitzuen fitxategia gehitzen da proiektura (*google-services.json*), eta, azkenik, Firebase-ren SDK gehitzen da proiektu eta app mailako dependentzietan, jarraian ikus daitekeenez:

```
dependencies {  
  
    implementation 'androidx.core:core-ktx:1.7.0'  
    implementation 'androidx.cardview:cardview:1.0.0'  
    implementation 'androidx.appcompat:appcompat:1.4.1'  
    implementation 'com.google.android.material:material:1.5.0'  
    implementation 'androidx.constraintlayout:constraintlayout:2.1.3'  
    implementation 'com.google.android.gms:play-services-maps:17.0.1'  
    implementation 'com.google.firebase:firebase-database-ktx:20.0.3'  
    testImplementation 'junit:junit:4.13.2'  
    androidTestImplementation 'androidx.test.ext:junit:1.1.3'  
    androidTestImplementation 'androidx.test.espresso:espresso-core:3.4.0'  
    // Import the Firebase BoM  
    implementation platform('com.google.firebase:firebase-bom:29.0.4')  
    implementation 'com.google.firebase:firebase-core:9.6.1'  
    implementation 'com.google.firebase:firebase-analytics'  
    implementation 'com.firebaseui:firebase-ui-auth:7.2.0'  
    implementation 'com.google.firebase:firebase-database'  
    implementation 'com.google.firebase:firebase-storage'  
    implementation 'com.google.android.gms:play-services-location:17.0.0'  
  
}  
  
apply plugin: 'com.google.gms.google-services'
```

23. Irudia. Aplikazio mailako dependentziak Firebase-ren implementaziorako.

```
buildscript {  
    ext.kotlin_version = '1.6.20'  
    repositories {  
        // Check that you have the following line (if not, add it):  
        google() // Google's Maven repository  
    }  
    dependencies {  
        // Add this line  
        classpath 'com.google.gms:google-services:4.3.10'  
        classpath 'com.android.tools.build:gradle:3.4.1'  
        classpath "org.jetbrains.kotlin:kotlin-gradle-plugin:$kotlin_version"  
    }  
}  
  
plugins {  
    id 'com.android.application' version '7.1.1' apply false  
    id 'com.android.library' version '7.1.1' apply false  
    id 'org.jetbrains.kotlin.android' version '1.6.21' apply false  
    id 'com.google.android.libraries.mapsplatform.secrets-gradle-plugin' version '2.0.0' apply false  
}
```

24. Irudia. Proiektu mailako Gradle fitxategia Firebase-ren implementaziorako.

Behin proiektua Firebase-ekin konektatuta dagoela, FBBackend.java klasean gehitu dira datu-basearekin egin beharreko funtzio guztiak. Horrez gain, *activity* bakoitzean, funtzio hauek erabiliko dira datuak gorde edo bistaratzeko. Jarraian funtzionaltasun baten adibide bat azalduko da, saioa hastearena, hain zuzen ere; gainontzekoen funtzionamendua oso antzekoa da, beraz, bibliografian dagoen 22. Estekan ikus daiteke.

Adibidea azaldu baino lehen, aipatu behar da, funtzio guztiak dituen klaseaz gain, hainbat klase gehiago erabili direla, kodea errazteko. Alde batetik, FirebaseHelper klaseak datu basearen erreferentzia eskuratzen du, eta, beste alde batetik, Backend klaseak FBBackend-ean garatzen diren funtzio guztiak definitzen dituen interfazea du.

Saioa hasteko lehenengo eta behin, erabiltzailearen email eta pasahitza jasotzen dituen pantailan, saioa hasteko botoia sakatzeko exekutatu beharreko funtzioa egin da. Funtzio honek, saioa hasteko funtzioari egiten dio dei, 25. Irudian ikus daitekeenez:

```
fun login(view: View) {  
    loginUser()  
}
```

25. Irudia. Saioa hasteko botoiaren funtzioa.

Saioa hasteko funtzioa 26. Irudian agertzen dena da. Funtzio honek, erabiltzaileak sartutako email eta pasahitza jaso, eta, hutsik ez badaude, Firebase-k eskaintzen duen metodoa erabiltzen du, email eta pasahitzarekin saioa hasteko (*signInWithEmailAndPassword(email, password)*). Funtzio honek itzultzen duen ataza ondo egin bada (*task.isSuccessful*), hasierako menura joateko funtzioari deitzen dio, bestela, erabiltzaileari jakinarazten dio autentikazioa ez dela ondo egin.

```

private fun loginUser() {
    //Email-a eta pasahitza jaso
    email = etEmail.text.toString()
    password = etPassword.text.toString()
    //Ezin dira hutsik egon
    if (!TextUtils.isEmpty(email) && !TextUtils.isEmpty(password)) {

        //Email eta pasahitzarekin saioa hasi
        mAuth.signInWithEmailAndPassword(email, password)
            .addOnCompleteListener(this) {
                //Ondo egin da
                task ->
                if (task.isSuccessful) {
                    //Ondo egin bada, datuak jaso eta hasierako orrira

                    goHome()
                } else {
                    //Bestela arazoa egon dela abisatu
                    Toast.makeText(this, "Authentication failed.",
                        Toast.LENGTH_SHORT).show()
                }
            }
    } else {
        Toast.makeText(this, "Enter all details", Toast.LENGTH_SHORT).show()
    }
}

```

26. Irudia. Saioa hasteko autentikazio funtzioa.

Hasierako orrira joateko funtzioak, behin Firebase-ko saioa hasita dagoela, erlaitza aldatzeaz gain, erabiltzailearen datuak jasotzen ditu datu basetik. Hau 27. Irudian ikus daitekeen kodearekin egin da.

```

private fun goHome() {
    val user = FirebaseAuth.getInstance().currentUser!!
    data = Data.getInstance()
    data.user.id = user.uid
    data.user.name = user.displayName

    var backend: Backend

    Locator.getBackend().userData
        .addOnCompleteListener { task: Task<UserProfile?> ->
            if (task.isSuccessful) {
                val intent = Intent(this, HomeActivity::class.java)
                intent.addFlags(Intent.FLAG_ACTIVITY_CLEAR_TOP)
                startActivity(intent)
            }
            finish()
        }
}

fun login(view: View) {
    loginUser()
}

```

27. Irudia. Hasierako orrira joateko funtzioa.

Azkenik, 28. Irudian ageri den kodearekin, datu basetik datuak nola jaso ikus daiteke. Funtzio honetan, lehenengo eta behin, autentikatuta dagoen erabiltzailea jasotzen da, eta saioa hasita ez badauka, ez du aurrera jarraitzen utziko. Behin hori eginda, datu basean, *sessions* taulan, erabiltzaileak duen erreferentzia jasoko da, bere UIDa erabilia. Azkenik, taulako sarrera horrek dituen seme ezberdinek dituzten datuak jasoko dira, eta aldagai globalean gorde, kodearen edozein puntutatik eskuragarria izateko.

```

public Task<UserProfile> getUserData() {
    FirebaseUser fireUser = FirebaseAuth.getInstance().getCurrentUser();
    if( fireUser == null )
        return Tasks.forException(new Exception("Not logged"));
    return dbRef("sessions", fireUser.getUid()).get()
        .continewithTask( task -> {
            DataSnapshot data = task.getResult();
            if( data.hasChildren() ) {
                i=0;
                while(data.child("loc"+i).exists()){
                    loc location = new loc();
                    location.setDate(data.child("loc"+i).child("date").getValue(String.class));
                    location.setLatitude(data.child("loc"+i).child("latitude").getValue(Double.class));
                    location.setLongitude(data.child("loc"+i).child("longitude").getValue(Double.class));
                    data1.getUser().addLocations(location);
                    i++;
                }
                if(data.child("friends").exists()){
                    for(DataSnapshot ds : data.child("friends").getChildren()) {
                        data1.getUser().addFriend(ds.getKey());
                        getEmail(ds.getKey());
                    }
                }
                data1.getUser().setTime(data.child("time").getValue(Double.class));
                data1.getUser().setDistance(data.child("distance").getValue(Double.class));
                data1.getUser().setSafeLat(data.child("safeLat").getValue(Double.class));
                data1.getUser().setSafeLong(data.child("safeLong").getValue(Double.class));

                return task;
            } else return dbRef("sessions", 1).get().continewith(subtask -> {
                DataSnapshot session = subtask.getResult();
                return null;
            });
        }).continewith(task -> {
            task.getResult(); // To propagate possible error
            return data1.getUser();
        });
}

```

28. Irudia. *Firestore*-etik erabiltzailearen datuak jasotzeko funtzioa.

Aplikazioaren erlaintzak

Erlaitz ezberdinen garapena egiteko, aurreko adibidean azaldutakoaren prozesu antzekoa egin da. Lehenik eta behin, XML fitxategian orriaren diseinua egin da, eta botoiei dagokien funtziora deia esleitu zaie. Jarraian, .kt fitxategian orriaren funtzionaltasunak definitzeko funtzioak sortu dira. Azkenik, eta behar izatekotan, datu basean idazteko edota irakurtzeko funtzioak garatu dira. Prozesu hau ulertzeko, ezarpenen erlaintza aztertuko da zehatzago; nolana ere, eranskinean dago kode osoa eskuragarri.

Esan bezala, lehenengo orriaren diseinua egin da XML fitxategian. Horretarako, elementu ezberdinak definitu dira, etiketa ezberdinak erabiliz. Elementu hauei, gainera, ID ezberdinak eman zaizkie, aurrerago kodetik eskuragarriak izateko. Hurrengo irudian ikus daitezke, adibidez, zenbait elementuren etiketak. Irudi honetan hiru etiketa mota ageri dira: botoi bat, pantailan idatzita ageri den testu bat eta idazteko gelaxka bat. Hirurak daude orrian finkatuta (*layout_constraint*) eta diseinua ezberdinak dituzte ezarrita.

```

<Button
    android:id="@+id/button_ChangeLang"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginTop="28dp"
    android:background="@drawable/rounded_corner_button_light"
    android:onClick="changeLang"
    android:text="@string/changeLanguage"
    android:textAllCaps="false"
    android:textColor="@color/white"
    android:textSize="28dp"
    android:textStyle="normal"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@+id/button_DeleteHist" />

<TextView
    android:id="@+id/textView2"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="@string/chooseDistance"
    android:textColor="#57504A"
    android:textSize="20dp"
    android:textStyle="bold"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.392"
    app:layout_constraintStart_toStartOf="parent"
    tools:layout_editor_absoluteY="64dp" />

<EditText
    android:id="@+id/textViewDistance"
    android:layout_width="182dp"
    android:layout_height="30dp"
    android:background="@drawable/rounded_corner"
    android:hint="@string/distance"
    android:inputType="number"
    android:paddingHorizontal="5sp"
    android:paddingVertical="5sp"
    android:textAlignment="center"
    android:textColor="@color/black"
    android:textColorHint="#7C7C7C"
    android:textSize="18dp"
    android:textStyle="bold"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.497"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@+id/textView2"
    app:layout_constraintVertical_bias="0.042"
    tools:ignore="DuplicateIds" />
  
```

29. Irudia. XML fitxategiaren etiketen adibideak.

Behin orria diseinatuta dagoela, Kotlin fitxategia sortu da, orriaren funtzionamenduaren kudeaketarako. Kasu honetan, orria kargatzen denean exekutatzeko den funtzioa 30. Irudian ikus daiteke.

```

override fun onCreate(savedInstanceState: Bundle?) {
    super.onCreate(savedInstanceState)
    setContentView(R.layout.activity_settings)
    setData()

    saveSettings_btn = findViewById(R.id.button_SaveSettings)
    saveSettings_btn.setOnClickListener {

        getData()
        var backend: Backend

        Locator.getBackend().setUserSettings()
        .addOnCompleteListener { task: Task<DataSnapshot?> ->
            if (task.isSuccessful) {
                Toast.makeText(this, "Settings saved", Toast.LENGTH_LONG).show()
                startActivity(Intent(this, HomeActivity::class.java))
            }
            finish()
        }
    }
}

```

30. Irudia. Ezarpenen orria kargatzean exekutatuako kodea.

Funtzio honekin, lehenengo eta behin, datu basetik irakurritako datuak kargatzen dira orrian; hau da, zenbat minuturo gorde nahi den posizioa eta posizio segurutik zenbat urrundu ahal den erabiltzailea posizioa gorde gabe. Hau egiteko, aurretik diseinuan etiketei esleitutako Idak erabiliz, elementuen testua aldatzen da, 31. Irudian ikusten denez.

```

private fun setData(){
    data = Data.getInstance()
    timeID = findViewById(R.id.textViewTime)
    distanceID = findViewById(R.id.textViewDistance)

    if(data.user.time!=0.0)
        timeID.hint = data.user.time.toString()
    if(data.user.distance!=0.0)
        distanceID.hint = data.user.distance.toString()
}

```

31. Irudia. Pantailako datuak aldatzeko kodea.

Honez gain, gordetzeko botoia sakatzen denean exekutatzeko duen kode bat ere badu, *setOnClickListener* erabilita implementatua. Funtzio honek, datu basean gordetzen dituen erabiltzaileak sartutako datuak, hurrengo irudian ikus daitezkeen kodea implementatuta. Kode honetan, 28. Irudian ikusi denez, erabiltzailearen instantzia lortzen da lehenengo; jarraian, dagokion taularen erreferentzia lortzen da eta, azkenik, *time* eta *distance* balioak aldatzen dira, *setValue* funtzioa erabilita.

```

@Override
public Task<DataSnapshot> setUserSettings() {
    FirebaseAuth fireUser = FirebaseAuth.getInstance().getCurrentUser();
    if( fireUser == null )
        return Tasks.forException(new Exception("Not logged"));
    // user = new UserProfile();

    DatabaseReference dbRef = FirebaseDatabase.getInstance().getReference();

    DatabaseReference sessionRef = FirebaseDatabase.getInstance().getReference("sessions/" + data1.getUser().getId());
    return dbRef.child("sessions").child(data1.getUser().getId()).get().addOnCompleteListener(new OnCompleteListener<DataSnapshot>() {
        @Override
        public void onComplete(@NonNull Task<DataSnapshot> task) {
            if (!task.isSuccessful()) {

            } else {
                DataSnapshot dataSnapshot = task.getResult();
                dbRef.child("sessions").child(data1.getUser().getId()).child("time").setValue(data1.getUser().getTime());
                dbRef.child("sessions").child(data1.getUser().getId()).child("distance").setValue(data1.getUser().getDistance());
            }
        }
    });
}

```

32. Irudia. Erabiltzailearen ezarpenak datu basean gordetzeko kodea.

Amaitzeko, ezarpenen erlaintzari dagokionez, gainontzeko botoiek antzeko funtzioak erabiltzen dituzte, datu basea ezabatzeko, beste erlaintz batera pasatzeko edota hizkuntza aldatzeko. Azken hau hurrengo puntuan azalduko da.

Konfigurazio garrantzitsuak

Garatutako aplikazioari dagokionez, hiru dira aipatu beharreko konfigurazio bereziak. Alde batetik, hizkuntzaren aldaketa, beste alde batetik, Google Maps-en APIaren erabilera, eta, azkenik, periodikoki exekutatzen den funtzio baten inplementazioa:

- **Hizkuntza aldaketa.**

Hizkuntza aldaketa era egokian ahalbidetzeko, lehenik eta behin, aplikazio osoan dauden hitz eta esaldi guztiak baliabideetan dagoen strings.xml fitxategian sortu behar dira; fitxategi honek, String bakoitza hizkuntza ezberdinetan gordetzeko aukera ematen du, horrela, aplikaziotik hizkuntza aukeratuta, zerrenda bat edo beste erabiliko baita. Kasu honetan, euskara eta gaztelera dira erabilgarri dauden hizkuntzak. Aplikazioak berez mugikorrek erabiltzen duen hizkuntza hartuko du oinarri, eta hori izango da lehenengo kargatuko duena. Hau eginda, diseinu orri guztietan, eskuz testuak idatzi beharrean, fitxategiari egingo zaio erreferentzia, eta zein String den adierazita, aplikazioak hizkuntza batean edo bestean erakutsiko du. Hurrengo adibidean, *chooseDistance* IDa duen testua bistaratzen da:

```
<TextView  
    android:id="@+id/textView2"  
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"  
    android:text="@string/chooseDistance"
```

33. Irudia. Strings.xml fitxategiko testuen erabilera.

Hizkuntza aldatzeari dagokionez, ezarpenen orritik egin daiteke. Orri horretan dagoen botoia sakatuta, Kotlin bidez programatutako funtzioak aplikazioa zein hizkuntzatan dagoen detektatzen du, eta, horren arabera, dagokiona ezarri. Funtzio hau jarraian ikus daitekeena da.

```
fun changeLang(view: View) {  
    saveSettings_btn = findViewById(R.id.button_ChangeLang)  
  
    if(saveSettings_btn.text.equals("Hizkuntza aldatu")){  
        setLocale("eu")  
        startActivity(Intent(this, HomeActivity::class.java))  
    }else{  
        setLocale("es")  
        startActivity(Intent(this, HomeActivity::class.java))  
    }  
}
```

34. Irudia. Hizkuntzaren detekzioa.

Hizkuntza detektatuta, aplikazioaren konfigurazioa aldatzen da, bere baliabideak aldatuta. Gainera, aplikazio osoan ezarpen berriak erabiliko direla adierazten da, editorean egindako aldaketak aplikatuz, 35. Irudian ikus daitekeenez.

```
private fun setLocale(lang: String?) {  
    val locale = Locale(lang)  
    Locale.setDefault(locale)  
    val config = Configuration()  
    config.locale = locale  
    baseContext.resources.updateConfiguration(config, baseContext.resources.displayMetrics)  
    val editor = getSharedPreferences("Settings", MODE_PRIVATE).edit()  
    editor.putString("My_Lang", lang)  
    editor.apply()  
}
```

35. Irudia. Aplikazioaren konfigurazioan egindako hizkuntza aldaketa.

- **Google Maps-en APIaren implementazioa.**

API hau mapak erakusten dituzten erlaitz guztietan inplementatu da. Horretarako, Google Cloud-etik, erabiliko den APIa aukeratu da. Behin hori eginda, aurretik aplikazioan gehitutako *google-services.json* fitxategia birkargatu behar da, ezarpen berriekin, eta, Firebase inplementatzeko egin den bezala, Gradle fitxategietan dependentzia berriak gehitu dira. Oraingoan, gainera, aplikazioaren manifestua ere aldatu behar izan da.

Android Manifest-a, aplikazioaren ezinbesteko informazioa duen fitxategia da; esaterako, bertan definitzen dira aplikazioaren paketearen izena, jarduerak, baimenak edota instalatu ahal izateko aplikazioak izan beharreko funtzioak. Kasu honetan, Google Maps Platform-etik, APIa erabiliko dela zehaztu denean jasotzen den API_KEYa gehitu behar da manifestuan, jarraian ikus daitekeenez:

```
<meta-data
    android:name="com.google.android.geo.API_KEY"
    android:value="AIzaSyA074IGIMBFZ0SNDWxq-eFusjmiss_IGX0" />
```

36. Irudia. Manifestuan API_KEYa gehitzea.

Behin konfigurazio guztia gehituta, XML eta .kt fitxategiak sortu dira. Diseinuari dagokionez, *fragment* bat baino ez da gehitu behar XML fitxategian, honek zuzenean inplementatzen baitu mapa pantailan. Hurrengo irudian ikus daiteke *fragment* honek duen itxura.

```
<fragment
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:map="http://schemas.android.com/apk/res-auto"
    android:name="com.google.android.gms.maps.SupportMapFragment"
    android:id="@+id/map"
    android:layout_width="match_parent"
    android:layout_height="match_parent"/>
```

37. Irudia. Mapa inplementatzeko *fragment*-a.

Funtzioei dagokienez, Google-ek hainbat funtzionaltasun eskaintzen ditu, bere mapekin lan egin ahal izateko. Aplikazio honen kasuan, erlaitz ezberdinetan inplementatutakoak hainbat dira: markagailuen kolorea aldatzea, markagailua mapatik mugitu eta bere posizioa jasotzea, markagailuetara irudia hurbiltzea eta, azkenik, markagailuetan informazioa gehitzea. 38. Irudian ikusten diren kode lerroetan aipatutako funtzionaltasunetatik hiru gehitzen dira. Alde batetik, markagailuaren izenburuan (*title*) gehitutako informazioa ageri da; beste alde batetik, kolorea aldatu zaio ikonoaren aukeretan (*icon*). Azkenik, kamera mugitu da latitude eta longitudea non dauden arabera (*moveCamera*).


```
val locationLat = data.user.safeLat
val locationLong = data.user.safeLong
// check if the latitude and longitude is not null
if (locationLat != null && locationLong != null) {
    // create a LatLng object from location
    val latLng = LatLng(locationLat, locationLong)

    //create a marker at the read location and display it on the map
    map.addMarker(MarkerOptions().position(latLng)
        .title("" + locationLat + ", " + locationLong)
        .icon(BitmapDescriptorFactory.defaultMarker(BitmapDescriptorFactory.HUE_VIOLET)))
    val update = CameraUpdateFactory.newLatLngZoom(latLng, 16.0f)
    //update the camera with the CameraUpdate object
    map.moveCamera(update)
```

38. Irudia. Mapan inplementatutako funtzionaltasunak.

- **Maiztasun jakinarekin exekutatutako funtzioa.**

Garatutako aplikazioaren funtzionaltasunen artean, badago bat periodikoki exekutatu beharrekoak, posizioa hainbat minuturo gordetzekoa, hain zuzen ere. Hau egiteko, Javak eskaintzen duen API bat erabili da, *ExecutorService*. API honek zeregin asinkronoak kontrolatzen ditu, hariak automatikoki sortuz eta maneiatuz.

Zerbitzu hau erabilia, erabiltzaileak ezarritako denboraren arabera errepikatzen den zeregin bat definitu da. Zeregin honek etengabe konprobatzen du ea distantziari dagokion baldintza betetzen den. Hau da, erabiltzailearen posizioa eta posizio seguruaren arteko distantzia berak ezarritakoa baino handiagoa bada, datu basean gordeko du, posizio berri bezala (locN) Raspberry-aren bidez lokalizazio tenporalean gordetako balioa.

39. Irudian ageri da *ExecutorService*-a erabilia definitutako ekintza periodikoa, *checkGuardado()* funtzioa exekutatzen duena erabiltzaileak adierazitako periodoarekin (*time*).

```
val executorService = Executors.newScheduledThreadPool(10)
executorService.scheduleAtFixedRate({
    checkGuardado()}, 0, time, TimeUnit.MINUTES)
```

39. Irudia. Ekintza periodikoa definitzeko kodea.

7.4. NB-IoT gailuaren diseinua

Erabiltzailearen posizioa NB-IoT bidez bidaliko duen gailuari dagokionez, bi dira izan beharreko funtzionaltasunak. Alde batetik, periodikoki erabiltzailearen lokalizazioa jaso behar du eta Firebase-era bidali, eta, beste alde batetik, komunikazio guztia NB-IoT bidez egin behar du.

Garatutako diseinua azaltzeko, hainbat atal ezberdinduko dira. Lehenengo eta behin, erabilitako gailuak berak zeintzuk diren azalduko da, eta hauen arteko konexioa nola egin den. Ostean, garapen prozesua bera deskribatuko da, hiru ataletan banatuta: GPS koordenatuen jasotzea, Firebase-arekin komunikazioa eta NB-IoT bidezko komunikazioa.

7.4.1. Erabilitako gailuak

Proiektu honetan, hiru dira erabilitako gailuak:

- **Raspberry Pi 3 Model B.**

40. Irudian ikus daitekeena da. Gailu hau tamaina txikiko ordenagailu bat da, prozesatze eta komunikazio ahalmen handiak dituena. Hainbat motatako proiektuak egiteko erabiltzen da, esaterako, elektronika edo robotikakoak, hala ere, beste hainbat funtzio ere izan ditzake, adibidez, domotika, eta web zerbitzari edota datu baseen sorrera [\[19\]](#).

Bere ezaugarriei dagokienez, hurrengoak dira garrantzitsuenak:

- Prozesadorea: Broadcom BCM2837.
- Erlojuaren maiztasuna: 1.2 GHz.
- Nukleoak: 4.
- RAM: LPDDR2 1 GB.
- Erresoluzioa: 1080 p 30 fps-ra.
- Konektibitatea:
 - 4 GHz IEEE 802.11.b/g/n.
 - Bluetooth 4.1.
- Portuak:
 - 40 GPIO pin.
 - HDMI.

- 4 x USB 2.0.
 - Audioa: 3.5 mm-ko Jack-a.
 - Micro USB-a elikadurarako.
 - Ethernet 10/100 base T.
- Elikadura: 5 V, 1 A.
 - Memoria: MicroSD.



40. Irudia. Raspberry Pi 3 Model B gailua.

Gailu honek, gainera, Raspberry Pi OS sistema eragilearekin egiten du lan, Debian sistema eragilean oinarritua. Pi OS kode irekikoa da, eta erabiltzen duen *software*-a ere kode irekikoa da.

- **Waveshare SIM7000E NB-IoT HAT.**

41. Irudian ikus daitekeena da. HAT (*Hardware Attached on Top*) bat, GPIO pinak erabilia, beste Raspberry-aren gainean muntatzen den beste gailu bat da. Kasu honetan erabilitakoa, potentzia txikiko *Narrow Band* IoT komunikaziorako modulu bat da, NB-IoT, Cat-M, EDGE, GPRS eta GNSS posizionamendua onartzen dituen. Abantailen artean, tamaina eta atzerapen txikia eta estaldura handia dira aipagarrienak [20]. Ezaugarriei dagokienez, hurrengoak dira garrantzitsuenak:

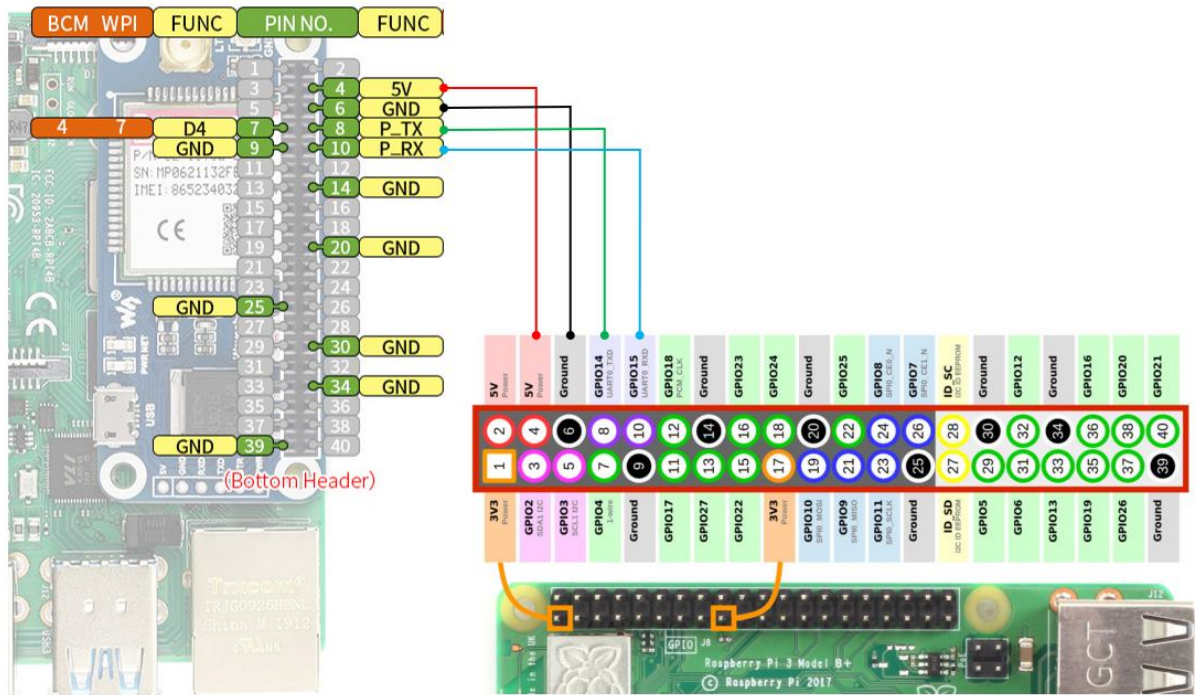
- Bandak:
 - NB-IoT/Cat-M: FDD-LTE B3/B8/B20/B28.

- GPRS/EDGE: 900/1800 MHz.
- Satellite sistemak: GPS, BeiDou, GLONASS, Galileo.
- Hartzailea: 16 kanal, C/A kodifikazioa.
- Sentikortasuna:
 - Jarraipenean: -162 dBm (GPS) / -157 dBm (GLONASS).
 - *Cold start*: -148 dBm.
- Zehaztasuna: < 2.5 m.
- Elikadura: 5 V.
- Funtzionamendu-tentsioa: 5 V / 3.3 V.



41. Irudia. Waveshare SIM7000E NB-IoT HAT gailua.

Bi gailu hauen konexioa portu serie bidez egin da. Horretarako, Raspberry-aren UART-a GPIO pinetan egon dadin konfiguratuta da, eta ez Bluetooth-ean, jatorriz datorren bezala. Hau eginda, hurrengo da egindako interkonexioa:



42. Irudia. Raspberry eta HATaren arteko interkonezioa.

- **Amarisoft AMARI Callbox mini**

43. irudian ikus daitekeena da Amarisoft base-estazioa [21]. Gailu honek, NB-IoT sareak testatzeko erabiltzeaz gain, CATM1, CAT0 eta CAT4 testatzeko ere balio du. 500 erabiltzailerera arte konekta dakizkioke, 200 Mbps-ko *uplink*-a eta 75 Mbps-ko *downlink*-a ditu eta 20 MHz-ra arteko banda zabalera du. Amarisoft-arekin lan egiteari dagokionez, aipatzekoa da berezko software bat dakarrela, berau konfiguratu ahal izateko. Beste hainbaten artean, ondokoak dira ezaugarri aipagarrienak:

- eNodeB/ng-eNodeB:
 - TDD eta FDD maiztasun banda guztiak sub-6 GHz bandan.
 - 1,4, 3, 5, 10, 15 eta 20 MHz-eko banda zabalerak LTeN eta 200 kHz NB-IoT-ko lan modu guztietan.
 - Antenak: 1etik 10era (MIMO 2x2).
 - NB-IoT azpi-garraiatazaileen espaziamendua: 15 kHz eta 3.75 kHz.
- gNodeB:
 - FDD/TDD FR1 (< 6 GHz).
 - 20 MHz-ra arteko banda zabalerak.

- Modu guztiak ahalbidetu daitezke.
- 2x2 MIMO-ra arte.



43. Irudia. Amarisoft AMARI Callbox mini base-estazioa.

7.4.2. Garapen prozesua

Raspberry eta HATak osatutako gailuaren funtzionaltasunak inplementatzeari dagokionez, hiru etapatan banatu da prozedura, aurretik aipatu den legez. Lehenengo eta behin, bi gailuen arteko komunikazioa bermatuta, HATetik GPS koordenatuak irakurtzea inplementatu da Raspberry-an. Behin hori eginda, Firebase-arekin komunikazioa ezarri da, eta irakurritako lokalizazioa bertan idaztea ere. Azkenik, WiFi bidez egindako prozesu guztia NB-IoT bidez egitera pasatu da.

Aipatzekoa da, baita ere, prozesu guztia Python lengoaiari idatzitako *script* bat erabiliz garatu dela, beronek eskaintzen dituen abantaila eta liburutegiak aprobetxatu ahal izateko. Gainera, AT (*Attention*) komandoak erabiliz garatu da komunikazioa. Komando hauek normalean M2M komunikazioa erabiltzen duten gailuetan erabiltzen dira, eta oinarrizkoenak Hayes komando bezala ere ezagutzen dira. Batez ere, GSM, GPRS edota mugikor MODEMetan erabiltzen dira.

1. Pausoa. GPS koordenatuen irakurketa

SIM7000E-tik bidalitako koordenatuak irakurtzeko, esan bezala, portu seriea ahalbidetu da. Hori eginda, portu seriea Raspberry-aren `/dev/serial0` portuan dago eskuragarri.

Kontuan hartu behar da, baita ere, transmisio-abiadura zein den, komunikazioa era egokian inplementatu dela bermatzeko; kasu honetan, *baud rate*-a 115200 Bd-koa da, 44. Irudian ikus daitekeenez.

```

##START SERIAL COMMUNICATION
ser = serial.Serial("/dev/serial0", 115200)
ser.timeout=2
ser.flushInput()
ser.flushOutput()
  
```

44. Irudia. Portu serie bidezko komunikazioaren ezarpena.

GPS koordenatuak irakurtzeko erabilitako AT komandoa *AT+CGNSINF* da. Hala ere, hau exekutatu aurretik, GNSSa aktibatu behar da (*AT+CGNSPWR=1*) eta ATaren beraren egoera (*AT*) egokia dela ere bermatu behar da. Hau 45. Irudian ikus daitezkeen komandoen bidez egin da.

```

#command echo activation to check sent commands
ser.write("ATE1\r\n".encode('utf-8'))
print("read> "+ser.read(100).decode('utf-8'))
#AT estate
ser.write("AT\r\n".encode('ascii'))
print("read> "+ser.read(100).decode('utf-8'))

#turn on GNSS
ser.write("AT+CGNSPWR=1\r\n".encode('utf-8'))
sleep(0.01)
cgns = ser.read(25).decode('utf-8')
print("read>"+cgns)
  
```

45. Irudia. GPSaren konfigurazioa.

Guztia martxan dagoela, GPSaren informazioa lortu da. Lortutako datuak, koma bidez banatutako string luze batean jasotzen dira, hurrengo taulan ikus daitekeen formatuarekin.

8. Taula. GPS bidez lortutako informazioaren formatua.

Indizea	Parametroa	Unitateak	Heina
1	GPSaren egoera	--	0-1
2	Egoera finkoa	--	0.1
3	UTC data eta ordua	yyyyMMddhh mmss.sss	yyyy: [1980,2039] MM: [1,12] dd: [1,31] hh: [0,23]

Indizea	Parametroa	Unitateak	Heina
			mm: [0,59] ss.sss:[0.000,60.999]
4	Latitudea	±dd.ddddddd	[-90.000000,90.000000]
5	Longitudea	±dd.ddddddd	[-180.000000,180.000000]
6	MSL altituea	metro	
7	Abiadura	Km/ordu	[0,999.99]
8	Ibilbidea	Gradu	0,360.00]
9	Modu finkoa	--	0,1,2
10	Erreserbatuta I		
11	HDOP	--	[0,99.9]
12	PDOP	--	[0,99.9]
13	VDOP	--	[0,99.9]
14	Erreserbatuta II		
15	GPS satelite erabilgarriak	--	[0,99.9]
16	GNSS satelite erabiliak	--	[0,99.9]
17	GLONASS satelite erabilgarriak	--	[0,99.9]
18	Erreserbatuta III		
19	C/N0 max	dBHz	[0,55]
20	HPA	metro	[0,9999.9]
21	VPA	metro	[0,9999.9]

Hau horrela, proiektuaren garapenerako beharrezkoak diren datuak hiru dira, latitudea, longitudea eta data/ordua, 3, 4 eta 5 errenkadak, batik bat. Datu hauek lortzeko, jasotako string-a banatu egin da, eta lortutako array-tik, dagozkion posizioak gorde. Honez gain, aipatzekoa da, UTC data jasotzen denez, ordua bi ordu aurreratu behar dela. Hau

egiteko, Python-ek eskuragarri duen *timeDelta* funtzioa erabili da, bi orduko aldea gehitzeko. Prozesu hau 46. Irudian ikus daitekeen kodearekin egin da.

```
#Get all GPS info
ser.write("AT+CGNSINF\r\n".encode())
gps = ser.read(1000).decode('utf-8')
#Split and get necessary data
gpsList = gps.split(",")
print(gpsList)

latitude = gpsList[3]
longitude = gpsList[4]
date = gpsList[2]
#Divide date and time
year = date[0:4]
month = date[4:6]
day = date[6:8]
hour = date[8:10]
minute = date[10:12]
second = date[12:14]
dString = year+"/"+month+"/"+day+" "+hour+": "+minute+": "+second
t = datetime.strptime(dString, "%Y/%m/%d %H:%M:%S")
#Add two hours
t1 = t + timedelta(hours=2)
#Convert into string
t1String = t1.strftime("%Y/%m/%d %H:%M:%S")
```

46. Irudia. GPStik jasotako informazioa irakurri eta prozesatzea.

2. Pausoa. Firebase-ekin komunikazioa

Raspberry-a Firebase-arekin komunikatzeko, Pythonen *Pyrebase* liburutegia erabili da. Liburutegi honekin, eta Firebase-en kontsolatik lortzen den proiektuaren konfigurazioarekin, zuzenean konektatzen da garatutako *script*-a sortutako Firebase aplikazioarekin, 47. Irudian ikus daitekeenez. Irudi honetan, konfigurazioaz gain, datu basearen eta autentikatutako profilen erreferentzia nola lortu ere ikus daiteke.

```
config = {  
    "apiKey": "AIzaSyDOFUKWSW8J_MCSbSU4-o5f38XUfy5W8V8",  
    "authDomain": "mal-iratiprieto.firebaseio.com",  
    "databaseURL": "https://mal-iratiprieto-default-rtdb.europe-west1.firebaseio.com",  
    "storageBucket": "mal-iratiprieto.appspot.com"  
}  
  
firebase = pyrebase.initialize_app(config)  
db = firebase.database()  
auth = firebase.auth()
```

47. Irudia. Pyrebase-aren konfigurazioa Python-en.

Behin datu basearekin komunikazioa ezarrita dagoela, erabiltzailearen email eta pasahitza eskatu dira, saioa hasi ahal izateko; horrela, erabiltzailearen UIDa jaso ahalko da, datu basearen zein errenkada aldatu behar den jakiteko. Kontuan hartu behar da ezin dela Raspberry-tik konturik sortu ez pasahitzik aldatu; bi funtzionaltasun hauek APPTik baino ezin dira bermatu.

```
email = input("Enter your email adress: \n")  
passw = getpass("Enter your password: \n")  
  
login = auth.sign_in_with_email_and_password(email, passw)  
  
print("sucess");  
  
userID = auth.current_user['localId']
```

48. Irudia. Erabiltzailearen saioa hastea Raspberry-tik.

Erabiltzailea autentikatuta dagoela, datu basean idatzi dira aurreko puntuan lortutako balioak, ezarritako arkitekturaren, mailaz maila sartuz, hurrengo irudian ikus daitekeenez. Puntu honetan, aipatzekoa da, denbora osoan *tempLoc* datua idazten dela datu basean, eta aplikazioak bertatik irakurriko duela beharrezkoa denean. Hau da, jarraian azalduko denez, 30 segundoro GPStik lortutako datuak idatziko dira datu baseko datu berean, eta, aplikazioak detektatzen duenean ezarritako distantzia baino urrunago dagoela edota erabiltzaileak posizioa gordetzeko botoia sakatu duenean, datu hau irakurri, eta *locN* datuetan idatziko du.

```
db.child("sessions").child(userID).child("tempLoc").child("date").setValue(t1String)  
db.child("sessions").child(userID).child("tempLoc").child("latitude").setValue(latitude)  
db.child("sessions").child(userID).child("tempLoc").child("longitud").setValue(longitude)
```

49. Irudia. Datu basean GPS datuak idaztea.

Azkenik, 30 segundoro errepikatu beharreko funtzionaltasuna implementatu da, 50. Irudiko pseudo-kodea erabilia. Bertan ikus daitekeenez, begizta infinitu bat ezarri da, 27

segundoro errepikatzen dena. 27 segundo dira, eta ez 30, izan ere, burutu beharreko lanak gutxi gorabehera 3 segundoko iraupena du.

```
#EVERY 30 SECONDS, READ GPS INFO AND UPDATE FB
while True:
    #Code to repeat

    sleep(27)
```

50. Irudia. 30 segundoro errepikatutako kodea.

3. Pausoa. NB-IoT bidezko komunikazioa

Raspberry-aren eta Firebase-aren arteko komunikazioa NB-IoT bidez egiteko, lehenengo eta behin, GPIO bidez soilik ez, USB bidez ere konektatu dira Raspberry-a bera eta NB-IoT *hat*-a. Behin hori eginda, sarearen konfigurazioa egin da, AT komando bidez, berriro ere.

Puntu honetan, aipatzekoa da, NB-IoT sarea sortzeko, Amarisoft enpresaren *AMARI Callbox mini* base estazioa erabili dela. Hau konfiguratzeko, lehenik eta behin, zein maiztasun bandatan ezarri behar den jakin behar da. Horretarako, *AT+CBANDCFG* komandoa erabili da, gailuak zein bandatan lan egin ahal duen jakiteko. Honen erantzuna hurrengo irudian ikus daitekeena da:

```
AT+CBANDCFG?
+CBANDCFG: "LTE",3,5,8,20,28
+CBANDCFG: "NB-IoT",3,5,8,20,28
```

51. Irudia. NB-IoT gailuaren lan-maiztasun bandak.

Hau horrela, 20. maiztasun bandan transmititzeko konfiguratu da, 6300 ARFCN-arekin (*Absolute Radio Frequency Channel Number*), hain zuzen ere. Behin hori eginda, base-estazioa konfiguratuta dagoenez, gailua bera konfiguratu da, hurrengo pausoak jarraituta:

- Gailua NB-IoT-n lan egiteko konfiguratu. Hau egiteko, bi komando erabili dira, 52. Irudian ikus daitezkeenak:
 - *AT+CMNB*. CAT-M eta NB-IoT arteko aukeraketa egiteko balio du; kasu honetan, 2 balioa ezarri zaio (NB-IoT).
 - *AT+CNMP*. Funtzionamendu modua aukeratzeko, GSM eta LTE artean. Kasu honetan, NB-IoT-k LTEren banda berean lan egiten duenez, LTE aukeratu da (38).

```
AT+CNMP=38  
OK  
AT+CMNB=2  
OK
```

52. Irudia. NB-IoT-n lan egiteko konfigurazioa.

- Gailuak erabiliko duen operadorea aukeratu, *AT+COPS* komandoak emandako aukeren artean. Kasu honetan, Amarisoft-ak sortutako sarea aukeratu da, eta 53. Irudiko komandoarekin zehaztu. Operadorea zein den galdetzen denean, gainera, 9 zenbakia azaltzen da, NB-IoT sare bat dela bermatzen duena.

```
AT+COPS=1,2,"00101"  
OK  
  
AT+COPS?  
+COPS: 1,2,"00101",9
```

53. Irudia. Operadorea aukeratzeko komandoa.

- APN-aren (*Access Point Name*) konfigurazioa. Konfigurazio hau egiteko, kontuan izan behar da Amarisoft-ak "internet" izena duela. Hau horrela, hurrengo irudian ikus daitekeenez konfiguratu da:

```
AT+CSTT="internet",,
```

54. Irudia. APNaren konfigurazioa.

Behin konfigurazio guztiak eginda, erabiltzailearen ekipoaren informazioa jaso da, sarera ondo konektatu dela ziurtatzeko. Jarraian ikus daitekeenez, *AT+CPSI* komandoaren erantzunean, NB-IoT sarera konexioa agertzen da, "00101" MCC+MNC kodearekin, eta 6300 ARFCNarekin, konexioa egokia dela zehaztuz.

```
AT+CPSI?  
+CPSI: LTE NB-IOT, Online, 001-01, 0x0002, 277447554, 37, EUTRAN-BAND20, 6300,0,0,-3,-56,-59,20
```

55. Irudia. Erabiltzailearen ekipoaren konfigurazio zuzenaren ziurtatzea.

Guztia ondo egin dela bermatuta, konexioa martxan jarri da *AT+CIICR* komandoa erabilia, ekipoak automatikoki IP bat jaso dezan. Kasu honetan, ekipoaren IPa 192.168.3.2 da, Amarisoft-aren IP tarte posibleen barruan dagoena, hain zuzen ere. Gainera, APN egokian dagoela ere ziurtatu da, *AT+CGPADDR* komandoa erabilia, konfiguraturutako APNa 1a da-eta.

```
AT+CIICR
OK

AT+CIFSR
192.168.3.2

AT+CGPADDR
+CGPADDR: 1,192.168.3.2
+CGPADDR: 13,0.0.0.0
```

56. Irudia: Ekipoaren IParen esleipena.

Puntu honetara helduta, ekipoaren konektibitatea ziurtatzeko, PING egin da Google-era, eta jarraian ikus daitekeenez, era egokian burutu da:

```
at+cipping="8.8.8.8"
+CIPPING: 1,"8.8.8.8",2379,114
+CIPPING: 2,"8.8.8.8",122,114
+CIPPING: 3,"8.8.8.8",171,114
+CIPPING: 4,"8.8.8.8",148,114

OK
```

57. Irudia. Google-era egindako PING zuzena.

Puntu honetara helduta, NB-IoT modulua era egokian dago konfiguratuta, beraz, Raspberry-a konfiguratzea baino ez da falta, izan ere, USB bidezko interfazea detektatu arren, ez dago ondo konfiguratuta, ez baitauka IPrik. Hau horrela, NB-IoT sarea USB bidez jaso dezan, eta berau sare-interfaze bezala erabili ahal izateko. Konfigurazio hau burutzeko komandoak 58. Irudian ikus daitezkeenak dira:

```

root@raspberrypi:/home/pi# qmicli --device=/dev/cdc-wdm0 --device-open-proxy --get-wwan-iface
wwan0
root@raspberrypi:/home/pi# qmicli --device=/dev/cdc-wdm0 --get-expected-data-format
802-3
root@raspberrypi:/home/pi# qmicli --device=/dev/cdc-wdm0 --device-open-proxy --wda-get-data-format
error: couldn't create client for the 'wda' service: QMI protocol error (3): 'Internal'
root@raspberrypi:/home/pi# route
Kernel IP routing table
Destination      Gateway          Genmask         Flags   Metric  Ref    Use Iface
default          0.0.0.0         0.0.0.0        U       204    0      0   wwan0
link-local      0.0.0.0         255.255.0.0    U       204    0      0   wwan0

root@raspberrypi:/home/pi# ip link set dev wwan0 down
root@raspberrypi:/home/pi# echo Y > /sys/class/net/wwan0/qmi/raw_ip
root@raspberrypi:/home/pi# ip link set dev wwan0 up

root@raspberrypi:/home/pi# qmicli --device=/dev/cdc-wdm0 --device-open-proxy --wds-start-networks="ip-type=4,apn=internet" --client-no-release-cid
[/dev/cdc-wdm0] Network started
Packet data handle: '2201541024'
[/dev/cdc-wdm0] Client ID not released:
Service: 'wds'
CID: '1'

root@raspberrypi:/home/pi# udhcpc -q -f -n -i wwan0
udhcpc: started, v1.30.1
No resolv-conf for interface wwan0.udhcpc
udhcpc: sending discover
udhcpc: sending select for 192.168.3.2
udhcpc: lease of 192.168.3.2 obtained, lease time 7200
  
```

58. Irudia. NB-IoT sarea USB bidez jasotzeko konfigurazioa.

Konfigurazio honekin, jadanik ekipoak IP bat jaso du, 192.168.3.2, hain zuzen ere. Guztia ondo egin dela ziurtatzeko, bideraketa taulak eta interfazeen konfigurazioa aztertu dira, 59. Irudian ikus daitekeenez.

```

root@raspberrypi:/home/pi# route
Kernel IP routing table
Destination      Gateway          Genmask         Flags   Metric  Ref    Use Iface
default          192.168.3.1     0.0.0.0        UG      204    0      0   wwan0
link-local      0.0.0.0         255.255.255.252 U       204    0      0   wwan0

root@raspberrypi:/home/pi# ifconfig -a
wwan0: flags=4163<UP, BROADCAST, RUNNING, MULTICAST> mtu 1500
    inet 192.168.3.2 network 255.255.255.252 destination 192.168.3.2
    unspec 00-000-00-00-00-00-00-00-00-00-00-00-00-00-00-00 txqueuelen 1000 (UNSPEC)
    RX packets 2 bytes 604 (604 B)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 60 bytes 9923 (9.6 KiB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
  
```

59. Irudia. Sarea era egokian konfiguratu dela bermatzea.

Konfigurazio hau guztia burututa, Raspberry-ak NB-IoT sarea erabiltzen du, beraz, Python *script*-a exekutatzean, interfaze honetatik egingo da, eta ez WiFi bidez, orain arte bezala.

8. Egin beharrekoen deskribapena

Atal honetan proiektuaren garapenerako egindako plangintzaren deskribapena egin da. Horretarako, proiektua bera fasetan banatu da, lan-pakete deiturikoak (LP). Honez gain, eta faseak banan-banan azaldu baino lehen, lan taldea zein izan den definitu da, eta erabilitako baliabide materialak zeintzuk izan diren ere aipatu da. Azkenik, plangintza osoari dagokion Gantt-en diagrama egin da.

8.1. Lan taldea eta baliabide materialak

Jarraian datorren taulan ikus daiteke lan-taldea osatzen duten ingeniarien multzoa:

9. Taula. Proiektuaren lan-taldea.

Kodea	Izena	Erantzukizuna	Rola
I1	Maidier Huarte	Senior ingeniaria	Proiektua gainbegiratu eta zuzendu
I2	Jasone Astorga	Senior ingeniaria	Proiektua gainbegiratu eta zuzendu
I3	Irati Prieto	Ingeniaria	Proiektua burutu

Lan-taldeaz gain, ondokoak dira proiektua gauzatzeko behar diren baliabide material guztiak:

10. Taula. Proiektuan erabilitako baliabide materialak.

Kodea	Baliabidea	Kopurua
PC	Ordenagailua	1
AM	Android mugikorra	1
RB	Raspberry Pi	1
WS	Waveshare SIM7000X NB-IoT hat	1
AS	Amarisoft AMARI Callbox mini	1

8.2. Faseen deskribapena

Jarraian azaltzen direnak dira proiektuan zehar garatutako fase edo lan-paketeak.

LP1 – Proiektuaren kudeaketa eta dokumentazioa

Fase honek proiektuak irauten duen beste iraun du. Lan pakete honen barnean sartzen dira bai proiektuaren kudeaketari dagozkion lan guztiak, bai garatu beharreko dokumentazioari dagozkionak. Gainera, plangintza era egokian jarraitzen dela bermatuko da, batzarrak eginez, jarraipen egokia bermatzeko. Bi ataza nagusi bereizten dira lan-pakete honetan:

- **A1.1. Proiektuaren jarraipena.** Proiektuan zehar egin beharreko ataza guztien jarraipena egiteko ataza, bakoitzarentzat ezarritako epeak betetzen direla ziurtatzeko, atzerapenik ez egoteko asmoz.
Iraupena: 50 ordu.
- **A1.2. Dokumentazioa.** Bukaeran entregatu beharreko memoria garatu ahal izateko, prozesu osoa dokumentatzeko ataza. Dokumentatzeaz gain, ataza honen barnean egongo da, baita ere, dokumentua bera idaztea. Proiektuan zehar garatuko da, gainontzeko faseekin batera.
Iraupena: 100 ordu.

Fase hau garatzeko beharrezkoak diren giza-baliabide eta baliabide materialen lan-ordu kopuruak ondokoak dira:

- **I1, I2:** 20 ordu bakoitzak.
- **I3:** 150 ordu.
- **PC:** 150 ordu.
- **RB, WS:** 20 ordu bakoitzak.

LP2 – Proiektuaren definizioa

Bigarren lan pakete hau hasierako faseari dagokiona da. Honetan, espezifikazioak ezarriko dira eta proiektuaren atal ezberdinei dagokien aukeren analisia egin da. Era berean, helburuak zeintzuk diren ere ezarri da.

Lan pakete honi dagozkion atazak ondokoak dira:

- **A2.1. Proiektuaren helburuen eta espezifikazioen definizioa.** Proiektuaren funtzionaltasunak zeintzuk diren definitzeko ataza. Honez gain, proiektuaren beraren helburuak eta irismena zeintzuk diren ere ezarri da.
Iraupena: 10 ordu.
- **A2.2. Proiektuaren garapenerako beharrezkoak diren kontzeptuen ikasketa.** Proiektua era egokian garatzeko ezagutu beharreko kontzeptuak

ikasteko ataza. Ataza hau ezinbestekoa da, izan ere, hainbat gai argi izan behar dira bai aukeren analisia era egokian burutzeko, eta baita proiektuaren garapena bera ahalik eta ondoen egin ahal izateko. Kontzeptu hauen ikasketan sartzen dira, esaterako, NB-IoT-ren eta Raspberry-aren funtzionamendu eta Kotlin programazio-hizkuntzaren ezaguera.

Iraupena: 50 ordu.

- **A2.3. Aukeren analisia.** Proiektuaren garapenari dagozkion atal ezberdinetan egon daitezkeen alternatiben analisia burutzeko ataza, kasuan kasu aukerarik egokiena erabiltzeko.

Iraupena: 50 ordu.

- **A2.4. Arriskuen analisia.** Proiektuaren garapenean gerta daitezkeen analisia egiteko ataza. Ataza hau betetzeko, arrisku bakoitzari probabilitate eta eragina esleitzeaz gain, ekiditeko eta gertatzekotan, aurre egiteko mekanismoak ere definitu dira.

Iraupena: 10 ordu.

Fase hau garatzeko beharrezkoak diren giza-baliabide eta baliabide materialen lan-ordu kopuruak ondokoak dira:

- **I1, I2:** 5 ordu bakoitzak.
- **I3:** 120 ordu.
- **PC:** 100 ordu.

LP3 – Soluzioaren diseinua

Behin ezagutu beharreko informazio guztia ikasita eta alternatibak baloratuta, garatuko den soluzioaren diseinu teoriko bat egin behar da, proiektuaren beharrak beteko dituen. Lan pakete honi dagozkion atazak ondokoak dira:

- **A3.1. Diseinu orokorra.** Proiektuak bere osotasunean zelan funtzionatuko duen definitzen da, funtzionaltasunak, eta aplikazioaren nondik norakoak barne.

Iraupena: 20 ordu.

- **A3.2. Datu basearen diseinua.** Datu basean zein datu gordeko den definituko da, eta baita nola antolatuko den ere, aplikazioaren funtzionamendua optimizatzeko.

Iraupena: 10 ordu.

- **A3.3. Aplikazioaren diseinua.** Aplikazioaren funtzionaltasunak betetzeko hau nola egituratuko den definitzen da. Horretarako, pantaila ezberdinak definitu eta diseinatu dira, bakoitzak beteko duen funtzioa zehaztuz.

Iraupena: 30 ordu.

- **A3.4. Gailuaren diseinua.** Raspberry-arekin beteko diren funtzionaltasunen

diseinua, NB-IoT bidezko komunikazioa barne hartuta.

Iraupena: 20 ordu.

Fase hau garatzeko beharrezkoak diren giza-baliabide eta baliabide materialen lan-ordu kopuruak ondokoak dira:

- **I1, I2:** 5 ordu bakoitzak.
- **I3:** 80 ordu.
- **PC:** 80 ordu.

LP4 – Soluzioaren garapena

Fase honetan, aurrekoan egindako diseinua oinarri hartuta, soluzioa bera garatu da, erraminta funtzional bat lortu arte. Lan pakete honi dagozkion atazak ondokoak dira:

- **A4.1. Datu basearen muntaia.** Datu baseak izan beharreko taula ezberdinak sortzen dira, eta baita aplikazioa lehen aldiz funtzionarazteko beharrezkoa den proba-erabiltzailea. Honez gain, aplikazioan bertan gehitu beharko den konfigurazioa kargatzen da, hurrengo atazarako prest izateko. Konfigurazio honekin, aplikazioa datu basera konektatu ahal izango da.
Iraupena: 10 ordu.
- **A4.2. Aplikazioaren garapena.** Aplikazioaren pantaila eta funtzionaltasun guztiak programatzen dira. Gainera, datu basearekin komunikazioa egokia dela ere bermatzen da.
Iraupena: 200 ordu.
- **A4.3. Gailuaren programazioa.** GPS funtzioa egingo duen gailua programatzen da, eta NB-IoT bidez datu basera datuak bidaltzen dituela bermatzen da.
Iraupena: 60 ordu.

Fase hau garatzeko beharrezkoak diren giza-baliabide eta baliabide materialen lan-ordu kopuruak ondokoak dira:

- **I3:** 270 ordu.
- **PC:** 270 ordu.
- **AM:** 150 ordu.
- **RB:** 60 ordu.
- **WS:** 50 ordu.
- **AS:** 20 ordu.

LP5 – Lanaren baliozkotzea

Azkenik, garatutako produktuak era egokian funtzionatzen duela bermatuko da. Kasu honetan hiru dira lan-pakete honi dagozkion atazak:

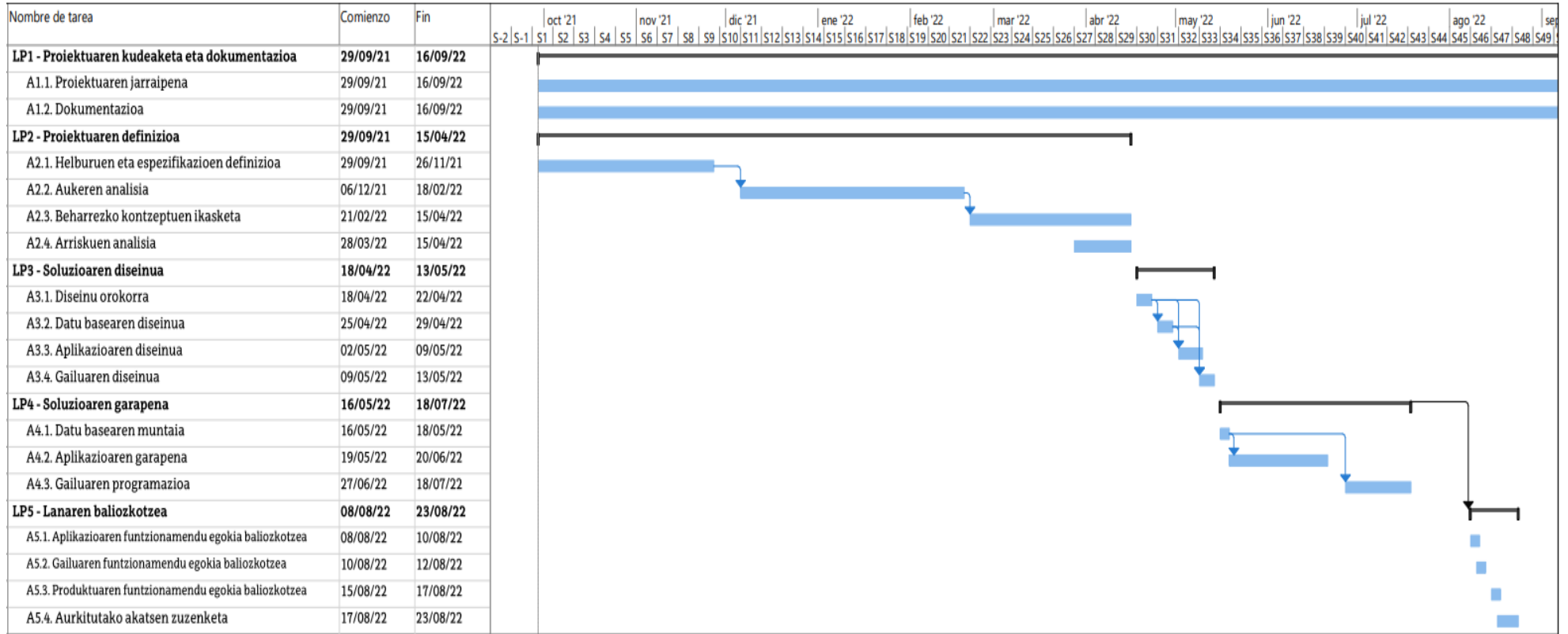
- **A5.1. Aplikazioaren funtzionamendu egokia baliozkotzea.** Aplikazioaren funtzionaltasun guztiek era egokian funtzionatzen dutela bermatuko da.
Iraupena: 5 ordu.
- **A5.2. Gailuaren funtzionamendu egokia baliozkotzea.** Garatutako gailuaren funtzionaltasuna egokia dela bermatuko da.
Iraupena: 5 ordu.
- **A5.3. Produktuaren funtzionamendu egokia baliozkotzea.** Aplikazioa eta gailua aldi berean erabiltzen direnean, funtzionamendua egokia dela bermatuko da.
Iraupena: 5 ordu.
- **A5.4. Aurkitutako akatsen zuzenketa.** Aurreko hiru atazetan detektatutako akats txikiak zuzenduko dira, eta behin hori eginda, funtzionamendua egokia dela balioztatuko da berriro ere.
Iraupena: 20 ordu.

Fase hau garatzeko beharrezkoak diren giza-baliabide eta baliabide materialen lan-ordu kopuruak ondokoak dira:

- **I1, I2:** 5 ordu bakoitzak.
- **I3:** 35 ordu.
- **PC:** 35 ordu.
- **AM:** 30 ordu.
- **RB:** 30 ordu.
- **WS:** 30 ordu.
- **AS:** 20 ordu.

9. Gantt-en diagrama

Jarraian proiektuaren hasieratik amaierara egindako lanaren Gantt-en diagrama ikus daiteke. Kontuan hartu behar da, proiektua 2021-2022 kurtso akademikoan garatu dela, baina, garapena apiriletik aurrera egin dela, nahiz eta proiektuaren definizioa data horren aurretik egin.



60. Irudia. Gantt-en diagrama

10. Kostuak

Hurrengo atal honetan proiektuaren alderdi ekonomikoak aztertuko dira, hau da, proiektuaren kostu totala. Hiru azpi-ataletan banatuko da: barne-orduak, amortizazioak eta gastuak, jarraian banan-banan zehaztuko direnak.

10.1. Barne-orduak

Proiektu hau garatzeko hiru ingeniariaren lana behar izan da, barne-orduetan kontuan hartu beharrekoak: proiektua bera garatu duen ingeniari juniorra eta bi ingeniari senior, zuzendaritza-lanak egiteaz gain, erabakiak hartzen ere lagundu dutenak.

Lan paketeetan ezarritako lan-orduak erabilita, hurrengo taulan ikus daiteke barne-orduen dagokien guztizkoa:

11. Taula. Barne-orduen kostua.

Kodea	Kargua	Ordu kopurua (h)	Orduko tasa (€/h)	Kostua (€)
I1	Ingeniari seniorra	35	50	1 750
I2	Ingeniari seniorra	35	50	1 750
I3	Ingeniaria	655	20	13 100
GUZTIZKOA				16 600 €

10.2. Amortizazioak

Proiektu honi dagozkion amortizazioei dagokienez, erabilitako ekipoa dira kontuan hartuko direnak, izan ere, erabilitako software guztia doakoa da, baita datu baserako erabilitako plataforma ere.

Lan paketeetan ezarritako erabilera-orduak erabilita, hurrengo taulan ikus daiteke amortizazioei dagokien guztizkoa:

12. Taula. Amortizazioen kostua.

Kodea	Erosketa prezioa (€)	Bizitza erabilgarria (h)	Erabilera orduak (h)	Kostua (€)
PC	800	20 000	635	25.4
AM	300	20 000	180	2.7
RB	100	10 000	110	1.1
WS	45	10 000	100	0.45
AS	20 000	10 000	40	80
GUZTIZKOA				109.65 €

10.3. Gastuak

Berrerabili ezin diren elementuen gastuak ere aurrekontuan kontuan hartu behar dira, barne-orduen eta amortizazioen gisan. Gastuak dira, esaterako, elektrizitatea edota Interneterako konexioa.

Hurrengo taulan ikus daitezkeenak dira proiektu honetarako egin diren gastuak:

13. Taula. Gastuak.

Kontzeptua	Kostua (€)
<i>Argindarra</i>	30
<i>Interneterako konexioa</i>	70
<i>Bestelako materiala</i>	20
GUZTIZKOA	120 €

10.4. Proiektuaren kostu totala

Proiektuaren kostu totala, orain arte kalkulaturako kostu guztien batura da; hau da, barne-ordua, amortizazioak eta gastuak. Kasu honetan ez da inolako zerbitzurik azpikontratatu, hortaz, kostu hori ez da batu behar. Azkenik, kostu ez-zuzenentarako eta ezustekoentarako % 10 gehituko da, kasuan kasu.

14. Taula. Proiektuaren kostu totala.

Kontzeptua	Guztizkoa (€)
<i>Barne-orduak</i>	16 600
<i>Amortizazioak</i>	109.65
<i>Gastuak</i>	120
AZPITOTALA I	16 829.65
<i>Ez-zuzenak (% 10)</i>	1 682.97
AZPITOTALA II	18 512.62
<i>Ezusteak (%10)</i>	1 851.26
GUZTIZKOA	20 363.88 €

11. Ondorioak

Proiektu hau burututa, hasieratik definitutako helburuak lortu direla esan daiteke. Alde batetik, gaur egun gehien erabiltzen den 4G estaldurarik ez dagoen lekuetan ere erabil daitekeen soluzio bat diseinatu da, izan ere, nahiz eta NB-IoT-k LTEren banda berdinetan lan egin, tarteko bandak ere erabiltzen dituzenez, estaldura berma daitekeela esan daiteke. Beste alde batetik, erabiltzailearentzako interfaze grafiko bat garatu da, aplikazio bat, hain zuzen ere, informazioaren kudeaketa egin dezan, eta bere kokapenaren informazioa gordetzen duen gailu baten kontzeptu-proba bat ere bai, kasu honetan Raspberry-arekin eta bere *hat*-arekin lortu dena. Bi hauek erabiltzailearentzako erabilerrazak eta sinpleak dira. Azkenik, erabiltzailearen eta bere datuen segurtasuna bermatu da uneoro, erabiltzaile autentikatuak eta lagun-zerrendak erabilia.

Hau guztia esanda, aipatu beharra dago gaur egun ere horrelako proiektuen beharra badagoela. Testuinguruan aipatu bezala, oraindik ere arriskupean badaude talde sozial asko eta orotarikoak, eta hauek ahalik eta seguruen egon eta senti daitezen behar beste tresna eskaini behar zaizkie. Gainera, garatutako proiektu hau egun pil-pilean dagoen IoT erabilia garatu da, eta 5G sarearen iritsierarekin ere, guztiz moldagarria izango da, betiere kontuan izanik mugikorrerako estaldura onik ez dagoen momentu eta zonaldeetan ere erabilgarria izan behar duela.

Erreferentziak

- [1] Asociación para la prevención de la tortura. *Grupos en situación de vulnerabilidad*. <<https://www.apt.ch/es/centro-de-conocimiento/detention-focus-database/grupos-en-situacion-de-vulnerabilidad>> [Azken kontsulta: 2022ko irailaren 13a]
- [2] Nazio Batuen Erakundea. (1993). *Emakumearen aurkako indarkeria ezabatzeari buruzko adierazpena*. New York, Nazio Batuak.
- [3] MOE. *Violencia contra la mujer. Datos y cifras* <<https://www.who.int/es/news-room/fact-sheets/detail/violence-against-women>> [Azken kontsulta: 2022ko abuztuaren 15a]
- [4] Gobierno de España – Ministerio de Igualdad. *Portal estadístico de la Delegación del Gobierno contra la Violencia de Género*. <<http://estadisticasviolenciagenero.igualdad.mpr.gov.es/>> [Azken kontsulta: 2022ko abuztuaren 15a]
- [5] EpData. *País Vasco - Datos y estadísticas sobre violencia de género por comunidad autónoma*. <<https://www.epdata.es/datos/datos-graficos-violencia-genero/49/pais-vasco/308>> [Azken kontsulta: 2022ko abuztuaren 15a]
- [6] El Español. *El mapa de los pinchazos en España: 63 denuncias y ninguna prueba de sumisión química*. <https://www.elespanol.com/reportajes/20220804/pinchazos-espana-denuncias-ninguna-prueba-sumision-quimica/692681045_0.html> [Azken kontsulta: 2022ko irailaren 13a]
- [7] Bhojar, P., Sahare, P., Dhok, S. B., & Deshmukh, R. B. (2019). *Communication technologies and security challenges for internet of things: A comprehensive review*. AEU - International Journal of Electronics and Communications, 99, 81-99. doi:<https://doi.org/10.1016/j.aeue.2018.11.031>
- [8] Nunca Me Pierdo. *Localizadores para niños*. <<https://nuncamepierdo.com/infantil/>> [Azken kontsulta: 2022ko abuztuaren 17a]
- [9] El Mundo, Sapos y Princesas. *4 Apps para localizar a nuestros hijos si se pierden en vacaciones*. <<https://saposyprincesas.elmundo.es/consejos/recomendaciones-tecnologia/apps-para-localizar-a-los-ninos/>> [Azken kontsulta: 2022ko abuztuaren 17a]
- [10] AlertCops. <<https://alertcops.ses.mir.es/mialertcops/>> [Azken kontsulta: 2022ko abuztuaren 17a]
- [11] EraStop. <<https://agrestop.mrsystem.es/>> [Azken kontsulta: 2022ko abuztuaren 17a]
- [12] UC3M. *Uc3m4safety - Equipo multidisciplinar para la detección, prevención y combate de la violencia contra las mujeres*. <http://portal.uc3m.es/portal/page/portal/inst_estudios_genero/proyectos/UC3M4Safety_es> [Azken kontsulta: 2022ko abuztuaren 20a]

- [13] Europako Parlamentuaren (EB) 2016/ 679 Erregelamendua - 2016ko apirilaren 27koa - datu pertsonalen tratamenduari eta datu horien zirkulazio askeri dagokienez pertsona fisikoak babesteari buruzkoa, eta abenduaren 5eko 3/2018 Lege Organikoa - datu pertsonalen babesari eta eskubide digitalen bermeari buruzkoa.
- [14] UNODC. (2011). *Manual sobre la aplicación eficaz de las Directrices para la prevención del delito*. New York, Nazio Batuak.
- [15] Rhine, P. P., & Wang, Z. S. (2018). *Data lake: A new ideology in big data era*. *ITM Web of Conferences*, 17, 3025. doi:10.1051/itmconf/20181703025.
- [16] IEBS. *Firebase, qué es y para qué sirve la plataforma de Google*. <<https://www.iebschool.com/blog/firebase-que-es-para-que-sirve-la-plataforma-desarrolladores-google-seo-sem/>> [Azken kontsulta: 2022ko abuztuaren 20a]
- [17] Mekki, K., Bajic, E., Chaxel, F., & Meyer, F. (2019). *A comparative study of LPWAN technologies for large-scale IoT deployment*. *ICT Express*, 5(1), 1-7. doi:<https://doi.org/10.1016/j.ict.2017.12.005>.
- [18] Seidor nts. *¿Qué es Firebase? ¿Qué ventajas ofrece en 2022 a nuestras apps?* <<https://www.nts-solutions.com/blog/firebase-que-es.html>> [Azken kontsulta: 2022ko abuztuaren 25a]
- [19] Raspberry Pi. *Raspberry Pi 3 Model B for home*. <<https://www.raspberrypi.com/products/raspberry-pi-3-model-b/>> [Azken kontsulta: 2022ko abuztuaren 26a]
- [20] Waveshare. *SIM7000E NB-IoT / Cat-M / EDGE / GPRS HAT for Raspberry Pi, GNSS, for Europe, Africa, Australia, Southeast Asia*. <<https://www.waveshare.com/sim7000e-nb-iot-hat.htm>> [Azken kontsulta: 2022ko abuztuaren 26a]
- [21] Amarisoft. *Amari Callbox mini*. <<https://www.amarisoft.com/products/test-measurements/amari-lte-callbox/>> [Azken kontsulta: 2022ko irailaren 10a]
- [22] Section. *Creating a Location-tracking App using Firebase and Google Maps in Android*. <<https://www.section.io/engineering-education/creating-a-location-tracking-app-using-firebase-and-google-maps-in-android/>> [Azken kontsulta: 2022ko ekainaren 10a]
- [23] Github. MAL-IratiPrieto. <<https://github.com/IratiPrieto/MAL-IratiPrieto>>

ERANSKINAK

I. ERANSKINA. Kodea

Memoriaren atal honetan erabilitako kode guztia biltzen da, bi ataletan banatuta: aplikazioari dagokion kodea alde batetik, eta GPS – NB-IoT gailuari dagokiona, bestetik.

Aplikazioaren kodea

Aplikazioaren kodea, bibliografiaren [23. Sarreran](#) aurki daiteke osorik, Github-eko biltegi publiko batean, izan ere, kodeak berak klase eta fitxategi gehiegi dituzenez, ez da posible era argi eta ordenatuan txostenean bertan bere osotasunean sartzea.

GPS – NB-IoT gailuaren kodea

```
from getpass import getpass

import pyrebase

import serial, time, sched

import datetime

from datetime import datetime, timedelta

from time import sleep

config = {

    "apiKey": "xxxx",

    "authDomain": "mal-iratiprieto.firebaseio.com",

    "databaseURL": "https://mal-iratiprieto-default-rtdb.europe-

west1.firebaseio.com",

    "storageBucket": "mal-iratiprieto.appspot.com"

}

firebase = pyrebase.initialize_app(config)

db = firebase.database()

auth = firebase.auth()
```

```
email = input("Enter your email adress: \n")
passw = getpass("Enter your password: \n")

login = auth.sign_in_with_email_and_password(email, passw)
print("sucess");

userID = auth.current_user['localId']
print("UID> "+userID)

##START SERIAL COMMUNICATION
ser = serial.Serial("/dev/serial0", 115200)
ser.timeout=2
ser.flushInput()
ser.flushOutput()
ser.write("ATE1\r\n".encode('utf-8'))
print("read> "+ser.read(100).decode('utf-8'))
ser.write("AT\r\n".encode('ascii'))
print("read> "+ser.read(100).decode('utf-8'))

ser.write("AT+CGNSPWR=1\r\n".encode('utf-8'))
sleep(0.01)
cgns = ser.read(25).decode('utf-8')
print("read>"+cgns)

#EVERY 30 SECONDS, READ GPS INFO AND UPDATE FB
```

```
while True:

    ser.write("AT+CGNSINF\r\n".encode())

    gps = ser.read(1000).decode('utf-8')

    gpsList = gps.split(",")

    print(gpsList)

    latitude = gpsList[3]

    longitude = gpsList[4]

    date = gpsList[2]

    year = date[0:4]

    month = date[4:6]

    day = date[6:8]

    hour = date[8:10]

    minute = date[10:12]

    second = date[12:14]

    dString = year+"/"+month+"/"+day+"
"+hour+": "+minute+": "+second

    t = datetime.strptime(dString,"%Y/%m/%d %H:%M:%S")

    t1 = t + timedelta(hours=2)

    t1String = t1.strftime("%Y/%m/%d %H:%M:%S")

    print("lat> "+latitude+" long> "+longitude+" date>
"+date)

    db.child("sessions").child(userUID).child("tempLoc").ch
ild("date").set(t1String)
```

```
db.child("sessions").child(userID).child("tempLoc").child("latitude").set(latitude)
```

```
db.child("sessions").child(userID).child("tempLoc").child("longitude").set(longitude)
```

```
sleep(27)
```

```
ser.close()
```