

# GRADU AMAIERAKO LANA

KONPUTAGAILUEN INGENIARITZA

---

## BCI SISTEMA RASPBERRY PI ETA EMOTIV KASKOAREKIN

---

Iñigo Erquicia Garcia

### **Zuzendariak:**

Antonio Arruti Illarramendi  
Javier Muguerza Rivero

---

2019ko Iraila



## Laburpena

Gradu amaierako lan honen funtsa, *Brain-Computer Interface* (BCI) sistema bat inplementatzea da. Garun jarduerarekin zerikusia duten seinaleak prozesatuko dira, buru-azalean ipintzen diren elektrodoei esker (EEG). Emotiv kaskoak eskaintzen dizkigun elektrodo hauekin, erabiltzailearen pentsamendu kontzienteari buruzko informazioa eskuratzea da helburua, eta datu hauek, beste gailu batean prozesatu ahal izatea. Emotiv kaskoa zuzenean konektatuko da Raspberry Pi gailu batekin eta honek, informazio hori prozesatu eta sistema bat jarriko da martxan, non seinalearen ezaugarriak erauzi ondoren, kanpo ekintza bat burutuko den, jasotako seinalearen araberakoa izango dena. Seinaleen eskuraketa eta prozesamendua, Python lengoian garatuko da, eta estimuluek sortaraziko dituen uhin seinaleetatik ezaugarri erabilgarriak erauzteko, gaur egungo teknika erabilienetako batekin egingo da, *Steady-State Visual Evoked Potential* (SSVEP) izenekoa.

# Edukien aurkibidea

---

<b>Laburpena</b>	<b>2</b>
<b>Edukien aurkibidea</b>	<b>3</b>
<b>Taulen eta irudien aurkibidea</b>	<b>6</b>
<b>Kodeen aurkibidea</b>	<b>7</b>
<b>1. Kapitulua</b>	<b>9</b>
Aurkezpena	9
1.1 Sarrera	9
1.2 Motibazioa	9
1.3 Proiektuaren deskribapena	9
1.4 Memoriaren edukia	10
1.5 Dokumentuaren idazkera	10
<b>2. Kapitulua</b>	<b>12</b>
Proiektuaren helburuen dokumentua	12
2.1 Helburuak	12
2.2 Plangintza	12
2.2.1 Irismena	12
2.2.2 Atazak	13
2.2.3 Orduen estimazioa	14
2.2.4 Atazen deskonposaketa	16
2.2.5 Kronograma	16
2.2.6 Komunikazio plana	18
2.2.7 Kalitate plana	18
2.2.8 Lan metodologia	18
2.2.9 Arriskuen plana	18
2.3 Plangintzan desbiderapenak	19
<b>3. Kapitulua</b>	<b>21</b>
Erabilitako teknologiak	21
3.1 Sarrera	21
3.2 Emotiv Epoc+	21
3.2.1 Kontzeptu nagusiak	22
3.2.1.1 Giza garuna	22
3.2.1.2 EEG (Elektroentzefalograma)	23
3.2.3 Materiala	24
3.2.4 Arkitektura	25
3.2.4.1 EEG sentsoreak	25

3.2.4.2 EEG seinaleak	26
3.2.4.3 Konexioa	26
3.2.4.4 Detekzio posibleak	26
3.2.4.5 Ezaugarri gehigarriak	26
3.2.5 Erabilpen gida	26
3.3 Raspberry Pi	28
3.3.1 Materiala	28
3.3.2 Arkitektura	29
3.3.3 Softwarea	30
3.3.3.1 Spyder	30
3.3.3.2 Raspbian sistema eragilea	31
3.3.4 Komunikazioa	31
3.3.4.1 SSH (Secure SHell)	31
3.3.4.2 SCP (Secure Copy Protocol)	32
<b>4. Kapitulua</b>	<b>33</b>
Brain-Computer Interface	33
4.1 Sarrera	33
4.2 ERP (Event-Related Potential)	33
4.2.1 SSVEP	34
4.2.2 P300 uhina	35
4.3 Metodologia	35
4.3.1 Sistemaren deskribapena	36
4.3.2 Ingurunearen deskribapena	39
4.3.2.1 Estimuluak	39
4.3.2.2 Ekintzak	40
<b>5. Kapitulua</b>	<b>41</b>
Ingurunearen implementazioa eta muntaia	41
5.1 Sarrera	41
5.2 Estimuluen interfazea	41
5.3 LED-aren muntaia	44
<b>6. Kapitulua</b>	<b>46</b>
Algoritmoaren inplementazioa	46
6.1 Sarrera	46
6.2 Erabilitako paketeak	46
6.3 Algoritmoa	47
6.3.1 Eskema	47
6.3.2 Azalpena	48
6.4 Feedback	58
<b>7. Kapitulua</b>	<b>60</b>
Probak eta emaitzak	60

7.1 Sarrera	60
7.2 Garapenean egindako probak	60
7.3 Amaierako emaitzak	61
<b>8. Kapituluak</b>	<b>63</b>
Ondorioak	63
8.1 Sarrera	63
8.2 Zailtasunak	63
8.3 Proiektuaren ondorioak	64
8.4 Hobekuntza posibleak	64
<b>Bibliografia</b>	<b>66</b>
<b>Eranskinak</b>	<b>67</b>
Erabilpen gida	67
Bilera aktak	69
Konstituzio bilera	69
Lehenengo bilera	69
Bigarren bilera	69
Hirugarren bilera	70
Laugarren bilera	70
Bosgarren bilera	70
Seigarren bilera	70
Zazpigarren bilera	71
Zortzigarren bilera	71
Bederatzigarren bilera	71
Hamargarren bilera	71
Hamaikagarren bilera	72
Itxiera bilera	72

## Taulen eta irudien aurkibidea

---

### Taulak

<b>2.1 Taula:</b> Ataza bakoitzaren ordu estimazioak.	15
<b>2.2 Taula:</b> Atazen orduen desbiderapenak.	20
<b>3.1 Taula:</b> Elektrodoen identifikazioa kokapenaren baitan.	23
<b>3.2 Taula:</b> Garun erritmo desberdinen frekuentziak.	24
<b>7.1 Taula:</b> LED-a 5 alditan kommutatzeko denborak.	62

### Irudiak

<b>2.1 Irudia:</b> LDE diagrama.	16
<b>2.2 Irudia:</b> Gantt diagrama.	17
<b>3.1 Irudia:</b> Emotiv Epoc+.	22
<b>3.2 Irudia:</b> Garunaren lobulu nagusien banaketa.	22
<b>3.3 Irudia:</b> Emotiv Epoc+ materiala.	25
<b>3.4 Irudia:</b> Sentsoreen eta erreferentzien kokapenak.	25
<b>3.5 Irudia:</b> Elektrodoen kokapen egokia.	27
<b>3.6 Irudia:</b> "Cortex UI" aplikazioaren sentsore kalitatearen interfazea.	28
<b>3.7 Irudia:</b> Raspberry Pi 3 Model B+.	28
<b>3.8 Irudia:</b> Raspberry Pi-arekin lan egiteko postua.	29
<b>3.9 Irudia:</b> Raspberry Pi 3 B+ arkitektura.	30
<b>4.1 Irudia:</b> SSVEP adibide baten ezaugarriak.	34
<b>4.2 Irudia:</b> P300 uhinaren adibide bat <i>Oddball</i> estrategia erabiliz.	35
<b>4.3 Irudia:</b> Sistemaren urratsak.	37
<b>4.4 Irudia:</b> Estimuluen interfazea.	40
<b>5.1 Irudia:</b> LED-aren zirkuitua.	45
<b>6.1 Irudia:</b> Algoritmoaren funtzioen eskema.	48
<b>6.2 Irudia:</b> Exekuzioaren <i>feedback</i> -a.	59
<b>7.1 Irudia:</b> 10Hz-ko estimuluaren grafikoak.	60
<b>7.2 Irudia:</b> 12Hz-ko estimuluaren grafikoak.	61
<b>7.3 Irudia:</b> Bideorako lagungarria.	62

## Kodeen aurkibidea

---

<b>3.1 Kodea:</b> Spyder instalazioa.	31
<b>3.2 Kodea:</b> SSH bidezko urruneko-konexioa Raspberry Pi-an.	31
<b>3.3 Kodea:</b> SCP erabilera fitxategiak kopiatzeko.	32
<b>5.1 Kodea:</b> Estimulu interfazeko aurkezpena.	42
<b>5.2 Kodea:</b> Estimulu interfazeko erabilpen gida.	43
<b>5.3 Kodea:</b> Estimuluen implementazioa.	45
<b>6.1 Kodea:</b> pyusb instalazioa.	46
<b>6.2 Kodea:</b> numpy instalazioa.	46
<b>6.3 Kodea:</b> scipy instalazioa.	47
<b>6.4 Kodea:</b> matplotlib instalazioa.	47
<b>6.5 Kodea:</b> pycrypto instalazioa.	47
<b>6.6 Kodea:</b> Funtzio eraikitzailea.	49
<b>6.7 Kodea:</b> Emotiv-en usb gailuarekin konexioa.	50
<b>6.8 Kodea:</b> Datuak deszifratzeko gakoak.	51
<b>6.9 Kodea:</b> Main funtzioa.	53
<b>6.10 Kodea:</b> Datu paketeak lortzeko funtzioak.	54
<b>6.11 Kodea:</b> Kalibrazioa egiteko funtzioa.	57
<b>6.12 Kodea:</b> Datu paketeak prozesatu.	57
<b>6.13 Kodea:</b> Estimulua begiratu den zehazteko kode zatia.	58
<b>6.14 Kodea:</b> LED-aren funtzioak.	58





## Aurkezpena

---

### 1.1 Sarrera

Gizakion gorputzaren kontrola nagusia, burmuinean gauzatzen da eta gaur egun asko dira bertan sortzen diren uhin hauek aztertzen dituen zientziak. Lan honetan, burmuinak sortutako uhinak irakurtzen eta interpretatzen saiatuko gara. Horretarako, sistema bat diseinatu eta inplementatuko dugu. Erabiltzailearen pentsamenduarekin egingo dugu lan, estimulu desberdinen aurrean aurreikusitako emaitzak lortzea da helburua.

Kostu baxuko *Emotiv* kasko bat erabiliko da hau dena gauzatzeko. Gailu honek, garunean ipinitako elektrodoetatik jasotako informazioa eskainiko digu eta Raspberry Pi batean prozesatuko dugu.

### 1.2 Motibazioa

Irakasgai batean emandako gaiei esker seinaleen inguruko mundua interesgarria iruditu zitzaidan. Gainera, bertako irakasleek biomedikuntza ikerketa talde batean egonik, burmuineko seinaleekin zerikusia duen proiektua aurkeztu zidaten eta ez nuen zalantzarik eduki. *Emotiv Epoc+* kaskoari esker gizakion burmuinarekin eta pentsamenduarekin egin zitekeen guztia ikusi ondoren (robotak mugitu, ekintza desberdinak gauzatu, gaixo diren pertsonak lagundu etab.), proiektu hau ekiteko dudan motibazioa handia da.

### 1.3 Proiektuaren deskribapena

Hasieran aipatu den bezala, proiektu honen izena BCI sistema baten inplementazioa da. BCI (*Brain Computer Interface*) sistema baten funtsa, ordenagailu eta burmuinaren arteko elkarrekintza gauzatzea da. Gaur egun badira burmuineko jarduera irakurtzeko gai diren kostu baxuko tresnak. *Emotiv Epoc+* dispositiboak esaterako, ez dituzte 800 euroak gainditzen eta hauei esker, elektroentzefalograma (EEG) fidagarriak eskura ditzakegu kanpoko gailu batean tratatu ahal izateko.

Proiektuaren ideia nagusia, sistema baten kontrola, gure burmuinarekin eta pentsamenduekin gauzatzea da. Bestalde, estimuluen aurrean sortzen diren seinaleak ulertu eta prozesatu ahal izateko SSVEP teknika bat erabiliko da, aurrerago azalduko dena. Proiektu honen garapena, bi atal nagusitan bana daiteke:

- Informazio eskuraketa: Emotiv Epoc+ kaskoa Raspberry Pi batekin konektatu ahal izatea eta *Bluetooth* bidez, kaskoak sortutako informazioa Raspberry Pi-an jaso ahal izatea. Pausu honetan, kanpoko gailu guztien muntaketa eta prestakuntza egin beharko da. Raspberry Pi bat hasieratik konfiguratu, *Bluetooth* bidez kaskoa konektatu eta software edo programa baten laguntzaz kaskoko datuak jaso.
- Informazio tratamendua: Behin Raspberry Pi-Emotiv elkarrekintza gauzatzen dela ziurtatzean, jasotako datuak tratatuko ditugu. Horretarako SSVEP teknika erabiliko da eta estimulu baten aurrean jasotako seinalearekin, sistema bat jarriko da martxan. Kasu honetan, erabiltzaileari ausazko periodo batekin argitzen diren hainbat objektu erakutsiko zaizkio pantaila batean, eta pausu honen helburua, estimulu horien aurrean gauzatuko den SSVEP ezaugarriak identifikatzea eta aginduren bat gauzatzea izango da.

## 1.4 Memoriaren edukia

Memoria honen edukia, kapitulu desberdinetan banatzen da. Lehenengo kapitulua proiektuaren aurkezpena egiteko erabili da, eta dokumentuaren xehetasunak azaltzeko. Bigarren kapituluan helburuen dokumentua burutuko da, plangintza batean oinarrituz. Hirugarren kapituluan, erabilitako teknologien azalpen sakona egingo da, eta haiekin zerikusia duten arlo teorikoak landuko dira ere. Hurrengo kapituluetan, garatutako sistemarekin zerikusia izango duten azalpenak emango dira. Hasteko, Garun-Ordenagailu Interfazeen inguruko datu teoriko esanguratsuak emango dira eta proiektu honetan garatu den sistemaren deskribapena egingo da jarraian. Ondorengo kapituluan, sistemaren ingurunearen muntaia nolakoa izan den azalduko da. 6. kapituluan, sistemaren atalik garrantzitsuenaren azalpena emango da, programen inplementazioak eta programazioak hain zuzen. Azken kapituluetan, sistemaren probak dokumentatuko dira eta proiektuaren amaierako ondorioak aztertuko dira. Azkenik, bibliografia eta eranskin gehigarriak memoriaren amaieran azalduko dira.

## 1.5 Dokumentuaren idazkera

Dokumentu osoan zehar errespetatu diren idazkerarekin zerikusia duten xehetasunak eta estiloak hauek dira:

- Letra motari dagokionez, dokumentazioaren zati handiena “Arial” letra motan egin da, salbuespenak salbuespen. Izan ere, Linux sistema eragilean exekutututako komando eta lerroak adierazteko “Ubuntu Mono” estiloa erabili da. Sistemaren garapenaren kapituluan aldiz, HTML kodea “Courier New” letrarekin azaltzen da.
- Testu osoa, 11-ko letra tamainarekin idatzi da. Kapituluaren tituluak, 18 tamaina eta letra lodiz idatzi dira. Kapituluaren izenburua eta bertako atalen tituluak 16 letrarekin egin dira. Bestalde, azpiatalen tituluetan 14 eta 12 letra tamainak erabili dira.
- Irudien, taulen eta kode zatien oinetan, haien identifikazio zenbakia eta titulua idatzi dira 10 letra tamaina erabiliz. Era berean, *CopyRight* bidez babestuta dauden irudien

kasuetan haien iturria gehitu da letra etzanez. Adibidez: **1.1 Irudia:** Adibidea. *Iturria:* [www.adibidea.eus](http://www.adibidea.eus). Garapenean inplementatu diren programen izenak ere, letra etzanez adierazi dira, *Estimuluak.html* esaterako.

- Erabilitako Ingeleseko hitzak, letra etzanez idatzi dira esaterako *Bluetooth* edo *Brain-Computer Interface*.
- Siglak erabiliko diren kasutan, hasieran sigla hauen hitz osoa agertuko da, eta parentesi artean, dokumentuan zehar ager litekeen siglak. Adibidez: Elektroentzefalograma (EEG) edo *Brain-Computer Interface (BCI)*.
- Dokumentazio garaian beste informazio iturri baten laguntza edo erreferentziak izan badira, testu hau kortxete arteko zenbaki batekin amaituko da, eta honen erreferentziaren esteka amaierako bibliografian jasoko da. Adibidez: Adibidea[0]. [0][www.adibidea.eus](http://www.adibidea.eus).

## 2. Kapitulu

---

### Proiektuaren helburuen dokumentua

---

Kapitulu honetan, proiektuaren helburuen nondik norakoak azalduko dira. Hasieran helburu nagusiak zein diren zehazki aipatuko dira. Ondoren, plangintzaren azalpen sakon bat emango da. Bertan irismena zein den azaldu, garatuko diren atazen deskribapena eta haien deskonposaketa erakutsiko da irudi batean. Gauzatu beharko diren orduen estimazio bat egingo da, proiektua aurrera eramateko atazen deskonposaketa bat egingo da eta haien arteko menpekotasunak azalduko dira. Azkenik, gerta litekeen arazoen plana azalduko da. Atazak burutzeko beharrezko denboraren estimazio bat egingo da eta Gantt taula batean kronologikoki erakutsi. Proiektua garatzeko kalitate, komunikazio eta arriskuen plana zein den laburki esan eta lan metodologia zein izango den azalduko da. Azkenik, proiektuaren amaieran, hasierako plangintzarekiko zein desbiderapen izan diren azalduko dira.

#### 2.1 Helburuak

Proiektu honen helburu aipagarrienak honako hauek dira:

- Elektroentzefalogramen inguruan sakontzea eta gaur egungo egoera aztertzea.
- Hardware tresna berriekin lan egitea eta haiekin trebatzea, Raspberry Pi eta Emotiv Epoc+ hala nola.
- Raspberry Pi-Emotiv Epoc+ elkarrekintza martxan jartzea edozein erabiltzaileren jarduera neuronal tratatu ahal izateko.
- SSVEP teknika ikertzea eta hau nola tratatu ikastea.
- Aurredefinitutako estimulu baten aurrean gauzatutako uhinak jaso eta ekintza jakin bat gauzatzea.

#### 2.2 Plangintza

Atal honetan, proiektuaren plangintzaren nondik norakoak azalduko dira.

##### 2.2.1 Irismena

Proiektu honen helburu nagusia, gizakion pentsamenduarekin zerikusia duen sistema bat martxan jartzea izango da. EEG seinaleak jasotzeko Emotiv Epoc+ kaskoa erabiliko da eta hauek prozesatzeko Raspberry Pi bat. Erabiltzaileari bi irudi erakutsiko zaizkio eta hauei begiratzean, ekintza ezberdinak burutuko dira. Estimulu bat begiratzean LED bat piztuko da, eta bestea begiratzean, LED-a itzali. Ibilbide guzti horretan, hau da Epoc+ kaskotik seinalea

eskuratu eta seinale hori ekintza bilakatzeko bidean, seinalea prozesatu beharko da. Laginketa bat ezarri beharko zaio, eta seinalearen ezaugarriak erazteko hau filtratu beharko dugu, erabaki ahal izateko eskuratzen dugun seinalea, zein egoeratan sortu den, zein estimulu begiratu duen erabiltzaileak hain zuzen.

## 2.2.2 Atazak

### A1 Kudeaketa

#### A1.1 Plangintza

- A1.1.1 Irismena zehaztu
- A1.1.2 Atazak zehaztu
- A1.1.3 Orduen estimazioa egin
- A1.1.4 Atazak deskonposatu
- A1.1.5 Kronograma sortu
- A1.1.6 Komunikazio plana zehaztu
- A1.1.7 Kalitate plana
- A1.1.8 Lan metodologia
- A1.1.9 Arriskuen plana

#### A1.2 Kudeaketa bilerak

- A1.2.1 Hasiera bilera
- A1.2.2 Jarraipen bilera
- A1.2.3 Itxiera bilera

#### A1.3 Jarraipen kontrola

- A1.3.1 Desbiderapenak kontrolatu

### A2 Analisia eta prestakuntza

#### A2.1 Proiektua planteatu eta helburuak zehaztu

- A2.1 Proiektua zehaztu
- A2.2 Helburu nagusiak azaldu

#### A2.2 Informazio bilketa

- A2.2.1 Emotiv Epoc+
- A2.2.2 EEG
- A2.2.3 *Brain-Computer Interface*
- A2.2.4 SSVEP eta P300

#### A2.3 Ingurunearen diseinua

- A2.3.1 Emotiv Epoc+ kaskoaren erabilpena
- A2.3.2 Erabiltzaileak begiratu beharreko estimuluak sortu

#### A2.4 Instalakuntzaren muntaia

- A2.4.1 Raspberry Pi ekipo bat martxan jarri
- A2.4.2 Emotiv Epoc+ kaskoa martxan jarri
- A2.4.3 Kanpo ekintzaren ingurunea muntatu.
- A2.4.4 Sistemaren aplikazioa martxan jarri

#### A2.5 Softwarearen instalazioa

- A2.5.1 Emotiv Epoc+ kaskoa kontrolatzeko Emotiv Emokit paketeak instalatu
- A2.5.2 Python lengoaiaren programatu ahal izateko paketeak instalatu

#### A2.6 Softwarea eta hardwarea erabiltzen ikasi

A.2.6.1 Emotiv Emokit paketeak eskeintzen dituen baliabideak erabiltzen ikasi

A.2.6.2 Emotiv Epoc+ eta Raspberry Pi gailuak erabiltzen ikasi

### A3 Garapena

A3.1 Raspberry Pi-Emotiv Epoc+ elkarrekintza gauzatu

A.3.1.1 USB hartzaile bidez konexioa gauzatu

A3.2 Emotiv kaskotik informazio egokia jaso

A.3.2.1 Programa baten bitartez elektrodoek jasotako seinalea lortzen dela ziurtatu

A3.3 Jasotako informazioaren tratamendua

A.3.3.1 Interesatzen zaizkigun elektrodoen seinalea filtratu

A.3.3.2 SSVEP gertaera detektatu estimuluen aurrean

A3.4 Sistemaren garapena

A.3.4.1 Garatutako aplikazioa martxan jarri

A.3.4.2 Aplikazioak gauzatuko dituen estimuluen aurrean ekintza bat burutu

Emotiv kaskotik jasotako informazioa tratatu ondoren

### A4 Egiaztapena

A4.1 Probak

A.4.1.1 Sistema, erabiltzaile desberdinekin probatu emaitzak egiaztatzeko

A4.2 Ondorioak

A.4.2.1 Proiektuaren ondorioak atera

### A5 Dokumentazioa

A5.1 Memoria

A.5.1.1 Memoria idatzi

A5.2 Aurkezpena

A.5.2.1 Aurkezpenerako materiala prestatu

A.5.2.2 Defentsa prestatu

## 2.2.3 Orduen estimazioa

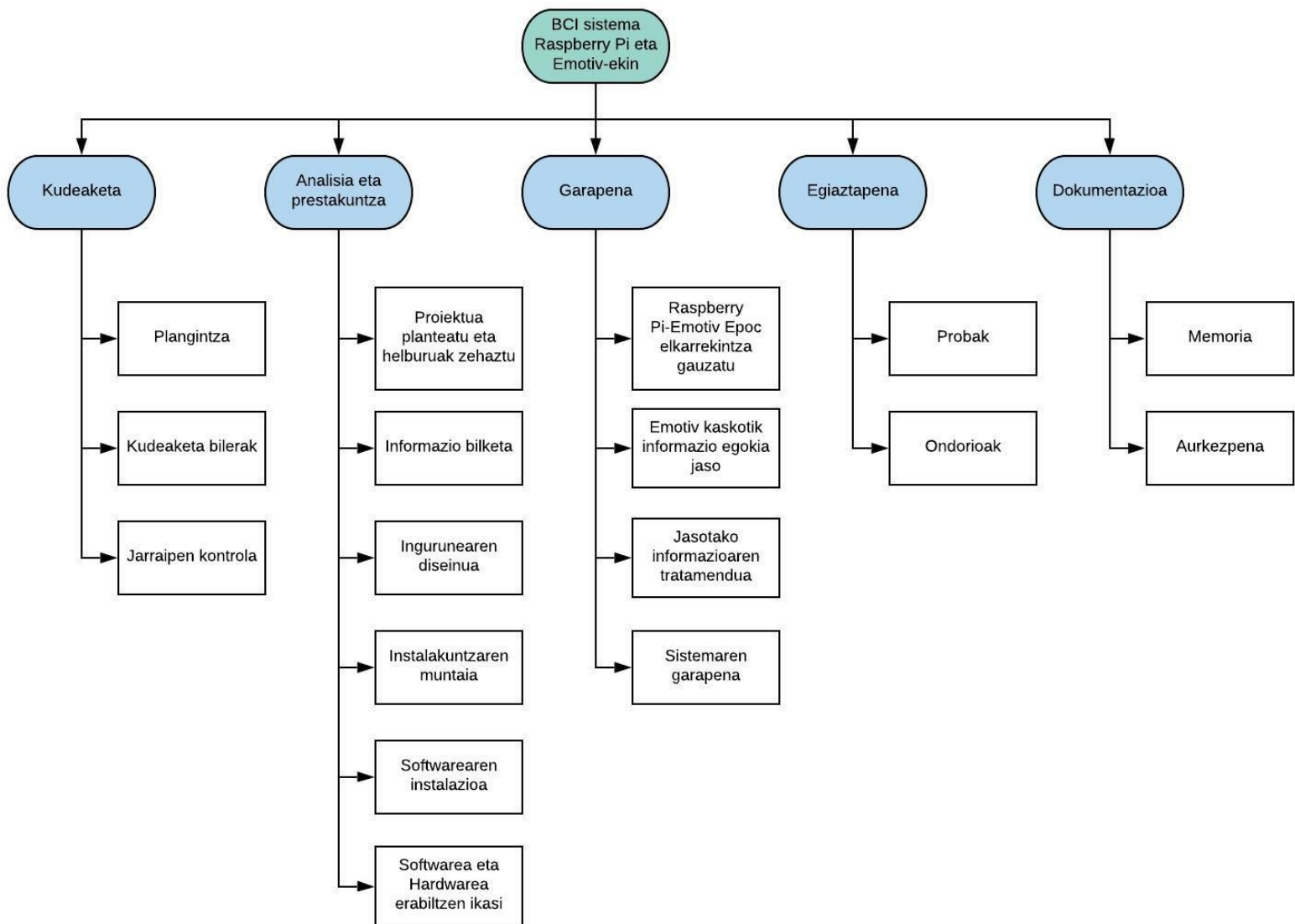
Atazak	Burutu beharreko lanak	Orduen estimazioa
Kudeaketa	Plangintza	20h
	Kudeaketa bilerak	3h
	Jarraipen kontrola	6h
Analisia eta prestakuntza	Proiektua planteatu eta helburuak zehaztu	3h
	Informazio bilketa	40h
	Ingurunearen diseinua	3h

	Instalakuntzaren muntaia	30h
	Softwarearen instalazioa	4h
	Softwarea eta Hardwarea erabiltzen ikasi	20h
Garapena	Raspberry Pi-Emotiv Epc elkarrekintza gauzatu	30h
	Emotiv kaskotik informazio egokia jaso	40h
	Jasotako informazioaren tratamendua	30h
	Sistemaren garapena	45h
Egiaztapena	Probak	4h
	Ondorioak	3h
Dokumentazioa	Memoria	72h
	Aurkezpena	15h
<b>GUZTIRA</b>		<b>336h</b>

**2.1 Taula:** Ataza bakoitzaren ordu estimazioak.



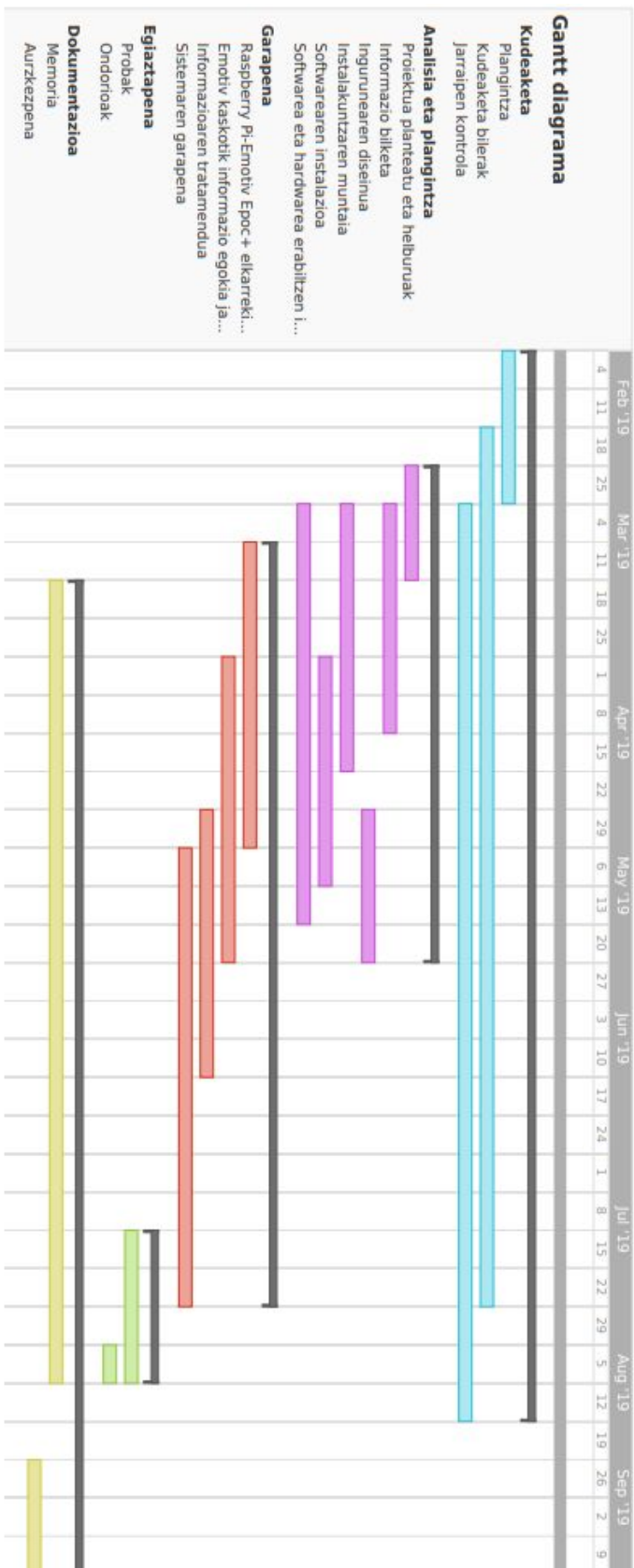
## 2.2.4 Atazen deskonposaketa



2.1 Irudia: LDE diagrama.

## 2.2.5 Kronograma

2.2 irudiko Gantt diagraman ikus dezakegu atazen eta lan paketeen banaketa zein izan den denboran zehar.



2.2 Irudia: Gantt diagrama.

## 2.2.6 Komunikazio plana

Zuzendarien eta nire arteko komunikazioa EHU korreo bitartez egingo da. Edozein zalantza edo bilera zehazteko bertatik egingo da. Bestalde, EHU-k eskaintzen duen GordeTaldea plataforman, direktorio bat edukiko dugu elkarrekin. Bertan, proiektuaren prozesuan egindako dokumentuak, aplikazioak etab. partekatuko dira.

## 2.2.7 Kalitate plana

Atal honetan, proiektuaren kalitatearen plana aztertuko da. Amaierako produktuak bete beharreko oinarriko betekizunak honako hauek dira:

- Raspberry Pi-Emotiv Epoc+ elkarrekintza gauzatzea.
- Emotiv Epoc+ kaskotik garun jarduera jasotzea.
- SSVEP gertaera baten aurrean ekintza bat gauzatzea.
- Amaierako helburua, estimulu bat arretaz begiratzean LED bat piztea eta beste estimulu bat begiratzean LED-a itzaltzea izango da.

## 2.2.8 Lan metodologia

Proiektu hau garatzeko fakultatean 3. solairuko laborategi bateko postu bat egokitu zait. Bertan, tutoreek utzitako materialarekin (PC bat, Raspberry Pi bat eta Emotiv Epoc+) lan egingo da. Bestalde, memoriaren dokumentazioa etxean egiteko aukera ere izango dut. Proiektuaren iraupeneko lehen 3 hilabeteetan, praktiken ondorioz proiektuari eskainiko diodan denbora kopurua egunean 2 ordukoa izango da. Apirilaren 18tik aurrera, praktikak amaitzean, egunean 7 ordukoa izango da.

Bestalde, kaskoarekin edozein proba egiteko, fakultateko postuan egingo da lan. Ordu gehienak bertan pasako dira, bai sistemaren muntaketa, bai informazio bilaketa eta baita memoriaren dokumentazioan ere. Halaber, Raspberry Pi-an edozein instalazio gauzatzeko, etxean egingo da lan eta memoriaren zatirik handiena ere bertan egingo da.

Memoriaren dokumentazioa, informazio bilketa, irudien eta dokumentuen biltegiratzea, Google Drive plataforman gauzatuko da. Bertan, eskuragarri izango dut edonoiz eta edonon proiektuaren jarraipenerako beharko ditudan dokumentuak eta informazioa. Hilabetean behin gutxi gora behera, Drive plataformako direktorioaren segurtasun kopia bat egingo da ekipo lokalean, edozein arrisku ekiditeko eta galerak txikitzeko, informazioa galduz gero.

## 2.2.9 Arriskuen plana

Proiektu honetan aurreikusi diren arrisku nagusiak eta hauek ekiditeko hartu daitezkeen erabakiak hauek dira:

- Raspberry Pi-a martxan jartzean zailtasunak azaldu daitezke bai periferikoen konexioan edota bai sistema eragilearen martxa jartzean. Kasu hauetan, hartu daitezkeen neurriak hauek dira:

- Beste periferikoekin probak egitea edota etxeko materialarekin konektatu. Arazoa mantentzen bada, urruneko konexioa gauzatzea posible izango litzateke, nahiz eta deserosoagoa izan.
- Sistema eragilearekin arazoak izanez gero, beste sistema eragile bat proba liteke edota NOOBS micro-SD txartela desberdin batekin saiatu.
- Emotiv Epoc+ kaskoarekin elkarrekintza gauzatzeko zailtasunak izatea. USB bidezko konexioa gauzatzeko orduan edota seinale ona jasotzeko momentuan arazoak ager daitezke.
  - USB Dongle bidezko konexioa gauzatzea ezinezkoa bada, *Bluetooth* bidez konektatzeko aukera izango da.
  - Behin konektatuta seinale kalitatea egokia lortzen ez bada, kokapen okerra dela eta adibidez, Epoc kasko zaharrarekin probak egiteko aukera dago, kalitate ona lortzeko.
- Programazioan zailtasunak izatea: sistema garatzeko orduan, edozein arazo gerta daiteke.
  - Jada garatuta dauden proiektuak hartuko dira abiapuntu bezala, proiektu hauek, USB Dongle-Epoc+ elkarrekintza gauzatzeko beharrezko kodea izango dute, eta besterik gabe, elektrodoen balioak zein diren lortuko dute.
  - SSVEP teknikarekin zailtasunak badaude, P300 teknika erabiltzeko aukera dago.

Halaber, arazo guzti hauek konpontzeko zailtasunak badira, tutoreekin hitz egingo da beste irtenbide bat bilatzeko.

## 2.3 Plangintzan desbiderapenak

Aurreko atalean, lan atazen denbora estimazio bat aurreikusi da. Hurrengo taulan, ataza bakoitzean sartu diren ordu errealak azalduko dira, konparaketa eginez. Ondoren, desbiderapen hauen arrazoiak azalduko dira.

Atazak	Burutu beharreko lanak	Orduen estimazioa	Egindako orduak
Kudeaketa	Plangintza	20h	22h
	Kudeaketa bilerak	3h	5h
	Jarraipen kontrola	6h	3h
Analisia eta prestakuntza	Proiektua planteatu eta helburuak zehaztu	3h	4h
	Informazio bilketa	40h	50h
	Ingurunearen diseinua	3h	2h
	Instalakuntzaren	30h	25h

	muntaia		
	Softwarearen instalazioa	4h	6h
	Softwarea eta Hardwarea erabiltzen ikasi	20h	25h
Garapena	Raspberry Pi-Emotiv Epoc elkarrekintza gauzatu	30h	30h
	Emotiv kaskotik informazio egokia jaso	40h	60h
	Jasotako informazioaren tratamendua	30h	20h
	Sistemaren garapena	45h	25h
Egiatapena	Probak	4h	7h
	Ondorioak	3h	2h
Dokumentazioa	Memoria	72h	75h
	Aurkezpena	15h	5h
<b>GUZTIRA</b>		<b>336h</b>	<b>366h</b>

**2.2 Taula:** Atazen orduen desbiderapenak.

## 3. Kapitulu

---

### Erabilitako teknologiak

---

#### 3.1 Sarrera

Kapitulu honetan, proiektuan erabili diren teknologien azalpen sakon bat egingo da. Hasteko Garun jarduera neurtzeko erabilitako kaskoaren aurkezpena egingo da, Emotiv Epoc+. Bertan, elektroentzefalografien azalpen teoriko bat egingo da eta Epoc+ bertsioak eskaintzen dizkigun baliabide eta material guztiak azalduko dira. Ondoren, seinale eta datu prozesaketarako erabili den Raspberry Pi-a esplikatu da eta bertan erabilitako softwarea.

#### 3.2 Emotiv Epoc+

“Emotiv Inc.” bioinformatikaren eta teknologiaren inguruan zerikusia duen produktuak garatzen dituen enpresa pribatu bat da. Kable gabeko elektroentzefalografiako produktuak egiteaz gain, hauekin lan egiteko beharrezko softwarea ere garatzen dute. 2011ean Tan Le eta Geoff Mackellar-ek sortu zuten eta Estatu Batuetan, San Frantziskon hain zuzen, dute haien egoitza.

Gure kasuan erabiliko dugun gailua Epoc+ izango da. Gaur egun merkatuan 800 dolar ingurutan aurki dezakegu. Giza garunaren jardueraren ikerketarako sortua izan da, kalitate altuko garun informazioa eskuratzeko gai da modu azkar batean. Bestalde, *Brain-Computer Interface* aplikazio desberdinak jorratzeko aukera ematen digu.

Kapitulu honetan, proiektuan garrantzi gehien edukiko duen gailuaren aurkezpena egingo da. Lan honetan zehar behin eta berriro azalduko diren kontzeptu teknikoen azalpen bat egingo da hasieran. Ondoren Emotiv Epoc+ kaskoaren arkitektura erakutsiko da eta erabiliko den materiala azalduko da. Azkenik kaskoaren erabilpen egokia izateko gida bat egingo da. 3.1 irudian ikus dezakegu Emotiv Epoc+ -a.



**3.1 Irudia:** Emotiv Epoc+.

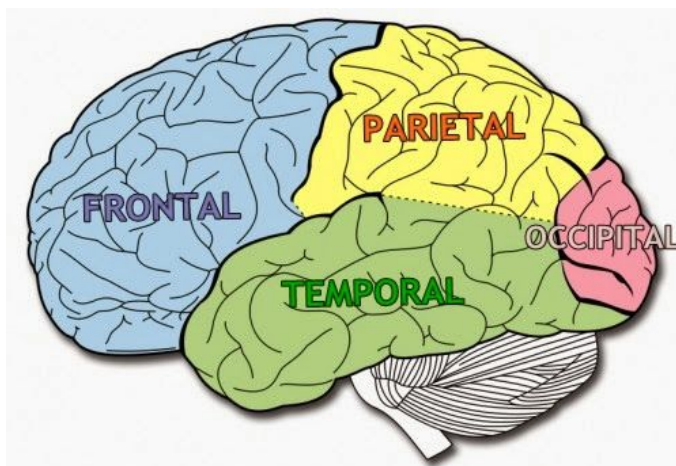
Iturria: <https://www.emotiv.com/product/emotiv-epoc-14-channel-mobile-eeq/>

### 3.2.1 Kontzeptu nagusiak

Atal honetan Emotiv Epoc+ kaskoarekin zerikusia duten kontzeptuak aztertu eta azalduko dira.

#### 3.2.1.1 Giza garuna

Garuna entzefaloaren atalik garrantzitsuen eta handiena da. Gizakietan bereziki, konplexutasun handia duen organoa da, nerbio-sistemaren zuzendaria hain zuzen. Milioika zelulaz osatuta dago eta bi motatakoak izan daitezke: neuronak eta neuroglia. Neuronak bata besteari lotzen zaizkio, neurona-zirkuituak sortzeko. Neuroglia berriz, neuronak elikatu, homeostasia mantendu eta babesa emateaz arduratzen da. Garuna, 6 lobulutan banatzen da zonalde bakoitzak duen funtzioaren arabera. 3.2 irudian ikus ditzakegu lobulu garrantzitsuenen kokapenak. Gehiago sakondu gabe, elektroentzefalogramari emango diogu garrantzia, honen azterketa egitea da eta gure helburua.



**3.2 Irudia:** Garunaren lobulu nagusien banaketa.

Iturria: <https://lamenteesmaravillosa.com/lobulos-cerebrales-caracteristicas-y-funciones/>

### 3.2.1.2 EEG (Elektroentzefalograma)

Elektroentzefalograma (EEG), garunean sortzen den jarduera elektrikoaren azterketa bat da. Hau, buru-azalean ipintzen diren plaka metaliko batzuei esker egiten da, elektrodoak direnak. Burmuineko zelulak etengabe komunikatzen dira pulsu elektrikoen bitartez, uneoro daude aktibo baita lo gaudenean. Jarduera hau, seinale baten bidez adierazi ohi da elektroentzefalografiako erregistro batean [1].

Elektroentzefalograma bat, gai da garun jardueran gerta zitekeen edozein aldaketa sumatzeko. Biomedikuntza munduan oso erabiliak dira garun desorekak antzemateko, esaterako epilepsia, garun tumoreak etab. Halaber, egingo dugun ikerketa, bioinformatikaren inguruan izango da.

Garun jarduera eskuratzeko orduan, elektrodoetara iristen den seinalearen diferentzia potentziala eta haren frekuentzia lortzen da besteak beste. Prozesu hori oso konplexua da, konexio neuronal ugari baitaude eta elektrodoen kokapenak eta pertsona berak aldaketak sor ditzakeelako seinalearen lortze prozesuan.

Elektrodoen kokapenaren garrantzia handia denez, 10-20 izeneko sistema da nazioartean estandarra dena, FISE (*Federación Internacional de Sociedades de Electroencefalografía*) erakundeak definitutakoa. Sistema honek, elektrodoen kokapena deskribatzen du entzefalogramaren neurketa egokia egin ahal izateko [2]. "10" eta "20" zenbakiek, bi elektrodoen arteko distantzia zehazten du, hau da, burezurreko ezker eta eskuinaren arteko distantzia totalaren %10 eta %20 bitarteko distantzia mantendu behar da elektrodo bizilagunen artean. 3.1 taulan ikus dezakegu elektrodoen kokalekuaren baitan hauei ematen zaien identifikazioak zein diren. Epoc+-aren arkitekturaren atalean azalduko da zehazki gure gailuak erabiltzen dituen elektrodoak zein diren.

Elektrodoaren identifikazioa	Lobulua
F	Frontala
T	Tenporala
C	Zentrala
P	Parietala
O	Okzipitala

**3.1 Taula:** Elektrodoen identifikazioa kokapenaren baitan

Amaitzeko, garun erritmo desberdinen azterketa labur bat egingo da. EEG seinaleak, frekuentziaren arabera banatzen dira taldetan [3]. Banda frekuentzia hauek, ikerketa psikologiko eta fisiologiko baten ondorioz definitu ziren. 3.2 taulan ikus dezakegu garun



erritmo desberdinen izenak eta hauen frekuentziak. Ondoren, erritmo hauek zein egoeratan sor daitezkeen azalduko dugu.

Izena	Frekuentzia
Delta	0.1 - 3.5 Hz
Theta	4 - 7.5 Hz
Alpha	8 - 13 Hz
Beta	14 - 30 Hz
Gamma	> 30 Hz

**3.2 Taula:** Garun erritmo desberdinen frekuentziak.

- Delta erritmoa ( $\Delta$ ): lo sakonean gaudenean sortzen den erritmoa da. 20 eta 200  $\mu\text{V}$  tarteko anplitudea duen seinalea da, 0.5-3.5 Hz inguruko frekuentziarekin.
- Theta erritmoa ( $\theta$ ): lobulu tenporalean sortzen den erritmoa da. 4-8 Hz bitarteko frekuentziarekin azaltzen da eta 20-100  $\mu\text{V}$  inguruko anplitudea izaten du. Erritmo hau, umeetan eta estres handia duten helduetan azaltzen da maiz.
- Alfa erritmoa ( $\alpha$ ): erritmo hau, erlaxazio egoerarekin bat azaltzen da. Pertsonak, buru-jarduera gutxi edota begiak itxita izaten ditu. Erritmo hau desagertzen joaten da kontzentrazio handiagoko jardueretan hasten denean. 20 eta 60  $\mu\text{V}$  bitarteko anplitudea izaten du.
- Beta erritmoa ( $\beta$ ): 2-20  $\mu\text{V}$  arteko anplitudea eta 14-26 Hz-ko frekuentzia izaten du erritmo honek. Gorputz-adarren eta muskuluen mugimenduarekin erlazionatuta dago. Kontzentrazio egoeratan ere azaldu daiteke eta lobulu zentral eta frontalean aurkitzen da.
- Gamma erritmoa ( $\square$ ): 30 Hz-ko frekuentzia gainditzen duten erritmoak dira. Frekuentzia altuenak eta anplituderik baxuenak izaten dituzte eta jarduera biziko egoeratan azaltzen dira.

### 3.2.3 Materiala

Emotiv Epoc+ kaskoak ekartzen duen materiala honakoa da:

- 14 kanaletako EEG mugikorra.
- 16 unitateko sentsore paketea.
- Salina likidoa.
- Komunikaziorako USB Dongle bat.
- Bateria kargatzeko USB kablea.



### 3.3 Irudia: Emotiv EPOC+ materiala.

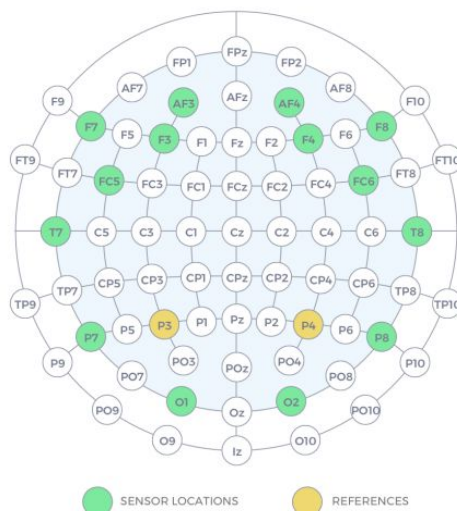
Iturria: <https://www.emotiv.com/product/emotiv-epoc-14-channel-mobile-ee/>

## 3.2.4 Arkitektura

Atal honetan Emotiv EPOC+ kaskoaren arkitekturaren xehetasun aipagarrienak azalduko dira. [4]

### 3.2.4.1 EEG sentsoreak

- Guztira 14 kanal eskaintzen dizkigu: AF3, F7, F3, FC5, T7, P7, O1, O2, P8, T8, FC6, F4, F8 eta AF4. 3.3 irudian ikus dezakegu kanal hauen kokalekua zein den.
- 2 erreferentzia: Buruaren erreferentzia izateko bi sentsore erabiltzen ditu, P3 eta P4 sentsoreetan kokatuak.
- Sentsore materiala: sentsoreetan kokatu beharreko salina likidoa. EEG seinaleen eskuratzea errazteko erabiltzen da.



### 3.4 Irudia: Sentsoreen eta erreferentzien kokapenak.

Iturria: <https://www.emotiv.com/product/emotiv-epoc-14-channel-mobile-ee/>

#### 3.2.4.2 EEG seinaleak

- Laginketa metodoa: Analogiko-digital bihurketa egiten duen laginketa sekuentziala.
- Laginketa frekuentzia: 128-256 laginketa segundoro (erabiltzaileak alda dezake).
- Bereizmena: 16 edo 14 bit.
- Banda-zabalera: 0.16-43 Hz.
- Tarte-dinamikoa: 8400  $\mu$ V(pp).
- Iragazketa: 5. ordenako Sinc iragazketa.

#### 3.2.4.3 Konexioa

Epoc+ bertsioan, 2 motatako konexioa gauzatu daiteke. Kontsumo baxuko *Bluetooth* bidez konekta daiteke edota USB hartzaile baten bidez ere. USB bidezkoa, zuzenean gauzatzen da, hau da PC edo Raspberry-ko USB portu batean hartzailea konektatu eta kaskoa piztean konexioa gauzatzen da. *Bluetooth* bidezko konexioa aldiz, kaskoa piztuta dagoela gauzatu behar da eta ekipoa kaskora konektatu. Horretarako *Bluetooth* 4.0 bertsioa beharrezkoa izango da.

#### 3.2.4.4 Detekzio posibleak

- Emozio neurketa: Emozio desberdin asko detekta ditzake, esaterako zirrara, konpromisoa, erlaxazioa, interesa, estresa, arreta...
- Aurpegi-espresioa: kliska, begi-keinua, harridura, irribarrea, barrea...

#### 3.2.4.5 Ezaugarri gehigarriak

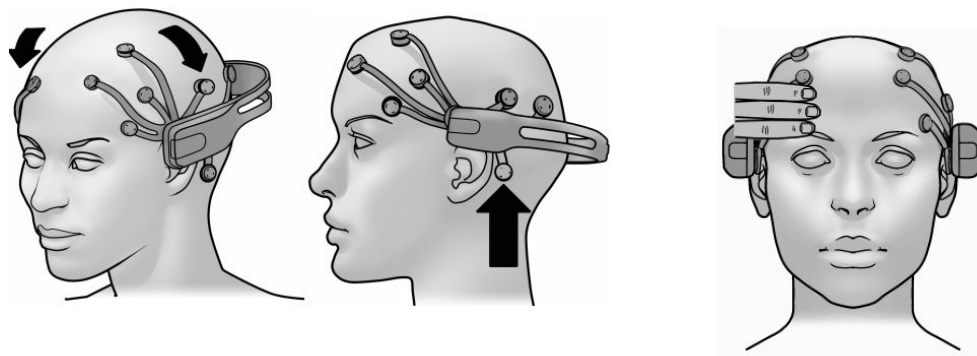
- Lithium Polymer motako 640mAh-ko bateria dauka.
- USB bidezko konexioarekin 12 orduko autonomia du bateriak eta *Bluetooth* bidezkoarekin 6 ordukoa.
- 1,19 Kg-ko pisua du.
- 9x15x15 cm-ko dimentsioa du.

### 3.2.5 Erabilpen gida

Emotiv Epoc+ kaskoaren erabilpen egokia ziurtatzeko, hurrengo urratsak jarraitu:

1. Hasierako karga gauzatzeko, atzean duen botoia *Off* posizioan dagoela USB kablea kaskoan konektatu eta 100-250V-ko korrante elektriko batean entxufatu karga hasteko. %100-eko karga izateko 4 ordu inguru beharko ditu. Kaskoaren karga osatu dela jakiteko, LED bat dauka atzeko aldean, berdez pizten dena guztiz kargatu denean.
2. Sentsoreak kaskoan ipini baino lehen, hauek hidratatzea oso garrantzitsua da. Horretarako salina likidoko tanta batzuk bota sentsore bakoitzari.
3. Behin sentsoreak ongi bustiak daudela, kaskoko kanal bakoitzean sentsoreak ipini behar dira. Horretarako kanalean ipini eta erlojuaren norabidean buelta laurden batekin "click" bat egin beharko luke ongi ipinita dagoela ziurtatzeko.

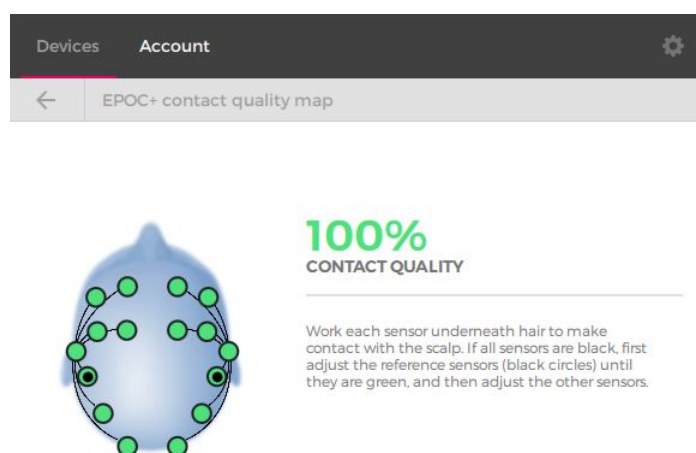
4. Ondoren, USB bidezko konexioa gauzatu nahi bada, USB Dongle-a PC-ra konektatu eta kaskoa piztu. *Bluetooth* bidez egin nahi bada, konexioa gauzatu beharko da kaskoa piztuta dagoela.
5. Kaskoa buruan ongi kokatzea oso garrantzitsua da seinale egokia jaso ahal izateko. Bi eskuen laguntzaz, buruaren goiko aldetik kaskoa ipini. 3.5 irudian ikus dezakegu kaskoaren kokapen egokia zein den.



### 3.5 Irudia: Elektrodoen kokapen egokia.

Iturria: <https://www.emotiv.com/files/Emotiv-Epoc-Quick-Start-Guide-2015.pdf>

6. Kaskoaren kokapena ongi dugula ziurtatzeko, Emotiv-ek “Cortex UI” izeneko software librea eskaintzen digu, orrialde ofizialetik jaitsi dezakeguna. Proba hau egiteko, Windows edo iOS sistema eragilea beharko dugu. PC-ra kaskoaren USB-a konektatzean eta kaskoa piztean, programa honetan kasko horren berri agertuko da. Bertan elektrodoen seinaleen kalitatea neur dezakegu. Seinale kalitate onena izateko, irudiko argi guztiak berdez egon beharko dira. Behin elektrodo guztiak berdez daudela, garun jarduera jasotzeko prest gaude. 3.6 irudian ikus dezakegu “EMOTIV App” programak erakusten digun sentsoreen kalitatearen interfazea.



### 3.6 Irudia: “Cortex UI” aplikazioaren sentsore kalitatearen interfazea.

## 3.3 Raspberry Pi

Proiektu honetan, hasieran esan bezala, Raspberry Pi baten laguntzaz gauzatuko dugu datuen prozesaketa. Raspberry Pi bat, tamaina txikiko eta kostu baxuko ordenagailu bat da, zirkuitu-plaka bakarrekoa. Mikro-ordenagailu hauen agerpena, iraultza izugarria izan da, prezio baxu batekin, beste edozein ordenagailu garestiagoaren gauza berak egin ahal ditzakeelako. Gainera, bere lehen bertsioaren irteeratik, 2012tik hain zuzen, bertsio hobekoak garatzea lortu dute prezioa mantentzen, gaur egungo azken bertsioa arte, Raspberry Pi 3 B+. Raspberry Pi Foundation-ek garatzen ditu Estatu Batuetan, eskoletan konputazio zientzien irakaskuntza sustatzeko asmoarekin. Informatika edo elektronikako proiektuetan erabili daitezke, eta mahaigaineko ordenagailu baten funtzio berdinak burutu ditzakete: kalkulu-orriak, bideojokoak, bideoen erreproduktzioa, testu prozesua etab.

Gure kasuan erabiliko dugun plaka, Raspberry Pi 3 Model B+ izango da. Gaur egun 40€ ingurutan aurki dezakegu merkatuan. Raspberry-aren helburua, Emotiv EPOC+ kaskoaren datuak jasotzea eta prozesatzea izango da. Horretarako, sistemaren garapenerako beharrezkoak zaizkigun programak eta paketek instalatu beharko ditugu bertan. Aurrerago azalduko da garapenerako erabiliko den softwarea zein izango den. 3.6 irudian ikus dezakegu erabiliko dugun Raspberry Pi modeloa.



**3.7 Irudia:** Raspberry Pi 3 Model B+.

### 3.3.1 Materiala

Raspberry Pi 3 B+ plakarekin lan egiteko, EHU-k eskaini digun materiala honakoa izan da:

- Raspberry Pi Model B+ plaka.
- Raspberry-a gordetzeko karkasa.
- 5V - 2.5A-eko elikadura kablea.
- Sistema eragilea instalatzeko NOOBS micro-SD txartela.

Hau guztiaz gain, fakultateko laborategian Raspberry-arekin lan egin ahal izateko eskura izan ditugun periferikoak hauek izan dira: teklatura, sagua, monitorea eta VGA-HDMI egokitzaile bat. 3.8 irudian ikus dezakegu Raspberry Pi-arekin fakultateko laborategian lan egiteko erabilitako periferikoak.



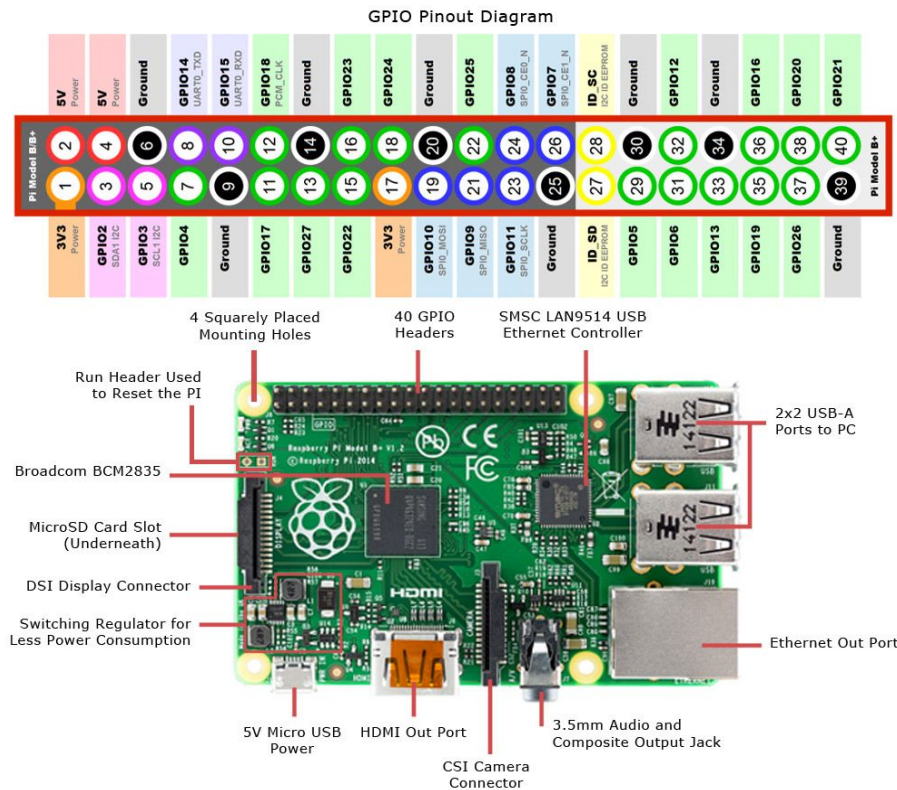
3.8 Irudia: Raspberry Pi-arekin lan egiteko postua.

### 3.3.2 Arkitektura

Erabiliko dugun modeloaren arkitekturaren eta hardwarearen ezaugarri garrantzitsuenak azalduko ditugu atal honetan. Hauek dira Raspberry Pi 3 B+ modeloaren prestazio eta ezaugarriak [5]:

- CPU + GPU: Broadcom BCM2837B0, Cortex-A53 (ARMv8) 64-bit SoC @ 1.4GHz.
- RAM: 1GB LPDDR2 SDRAM.
- Wi-Fi + *Bluetooth*: 2.4GHz y 5GHz IEEE 802.11.b/g/n/ac, *Bluetooth*
- 4.2, BLE
- Ethernet: Gigabit Ethernet konexioa USB 2.0-rekin (300 Mbps)
- GPIO: 40 pin.
- HDMI.
- USB 2.0 4 portu.
- CSI eta DSI portuak kamara eta ukimen-pantaila konektatzeko.
- Audio estereo eta bideo irteerak.
- Micro-SD.

- Power-over-Ethernet (PoE).



### 3.9 Irudia: Raspberry Pi 3 B+ arkitektura.

Iturria: <http://electronicayciencia.blogspot.com/2016/11/conexion-gpio-de-raspberry-pi-3.html>

### 3.3.3 Softwarea

Raspberry Pi hutsa bagenu, sistema eragilearen instalazioa egin beharko genukeen lehendabizi. Gure kasuan aldiz, NOOBS micro-SD txartela dugunez, sistema eragilea bertan instalatuta dator eta Raspberry Pi-ko micro-SD sarreran txartela sartzea besterik ez dugu. Konektatzen dugun lehen aldiari, datu batzuk sartu beharko ditugu (data, ordua, etab.) eta hortik aurrera, zuzenean ingurune grafikoan hasiko da Raspberry Pi-a pizten dugun momentuan. Micro-SD txartelarekin erabili den sistema eragilea Raspbian da.

Bestalde, sistemaren garapena eta implementazioa Python lengoia egin denez, programa bat instalatu da ediziorako, Spyder hain zuzen.

#### 3.3.3.1 Spyder

Spyder Pythonen idatzitako ingurune zientifiko oso ahaltsua da eta zientzialariek, ingeniariak eta datu-analisten diseinatua izan da. Edizio, analisi, arazketa eta profilen funtzionalitate bateratuaren konbinazio paregabea eskaintzen du datuen esplorazioa, exekuzio interaktiboa, ikuskapen sakona eta pakete zientifiko baten bistaratze ahalmen handiarekin. Ezaugarri integratuetatik haratago, bere gaitasunak are gehiago hedatu daitezke bere plugin-sistemaren eta API-aren bidez.

Proiektu honetan, “Spyder Python 2.7” bertsioa erabili da programa guztien ediziorako. Raspberry Pi-an bertsio honen instalazioa gauzatzeko, hurrengo lerroa terminalean exekutatzea besterik ez da.

```
sudo apt-get install spyder
```

**3.1 Kodea:** Spyder instalazioa.

### 3.3.3.2 Raspbian sistema eragilea

Raspbian Foundation erakundearen sistema eragile ofiziala eta librea da. Debian izeneko GNU/Linux banaketa batean oinarritzen da. NOOBS bidez instalatu liteke edo bestela, web-orri ofizialean eskuragarri dago instalazioa. Raspbian sistema eragilea, hezkuntza, programazioa edota erabilpen arrunterako lagungarriak diren hainbat software-rekin dator, esaterako Python, Scratch, Sonic Pi, Java etab.

### 3.3.4 Komunikazioa

Raspberry Pi-arekin komunikazioa gauzatzeko, bi modu erabili dira. Fakultatean, periferiko guztiak eskura izanda, beste monitorea, teklatu eta sagu baten bitartez egin da lan. Halaber, fakultatean Raspberry Pi ekipoa ezin izanenez sarera konektatu (ez dagoelako erregistratuta), beharrezko instalazioak etxean egin dira. Bertan, urruneko-komunikazioa erabili da ssh bidez. Bestalde, fitxategiak Raspberry Pi-tik PC-ra pasa ahal izateko, SCP erabili da, edozein momentutan, bi ekipoetan fitxategiak behar izanez gero.

#### 3.3.4.1 SSH (Secure SHell)

SSH, *Secure SHell*, bezero-zerbitzariaren arkitektura bat erabiliz bi sistemen arteko komunikazio segurua errazten duen protokolo bat da, eta erabiltzaileari *host* batera urrunetik konektatzeko aukera ematen dio. FTP edo Telnet bezalako urruneko beste komunikazio protokoloak ez bezala, SSH-k konexio saioa enkriptatzen du eta ezinezkoa da norbaitek enkriptatutako pasahitzak lortzea.

Komunikazioa hasteko, lehen pausoa Raspberry Pi-ari esleitu zaion IP helbidea zein den ikustea da, periferiko guztiak konektatuta ditugula. Terminalean `ifconfig` exekutatu eta `eth0` interfazearen IP helbidea gordeko dugu. Bestalde, Raspberry Pi-an SSH zerbitzaria aktibatu beharko dugu komunikazioa gauzatu baino lehen. Ondoren, Ethernet kable batekin Raspberry Pi-a nire etxeko eramangarrira konektatu dut beharrezko instalazioak egiteko. Nire ekipoaren Ubuntu 16.04 bertsioan terminal berri bat ireki eta hurrengo lerroa exekutatzean, Raspberry Pi-aren terminala irekiko da, pasahitza sartu ondoren.

```
ssh pi@RASPBERRY_IP_HELBIDEA
```

**3.2 Kodea:** SSH bidezko urruneko-konexioa Raspberry Pi-an.



### 3.3.4.2 SCP (Secure Copy Protocol)

SCP, *Secure Copy Protocol* edo *Simple Communication Protocol*, bi ekipoen artean fitxategiak edo direktorioak kopiatzeko erabiltzen den protokolo segurua da. Komunikazioa zifratu egiten denez, eraso posibleetatik babestuta dago. SCP erabili den kasu guztietan, Raspberry Pi-tik PC-ra direktorioak kopiatzeko izan da, eta hurrengo komandoa exekutatu da Ubuntu 16.04 sistema eragilean:

```
scp (-r) fitx_edo_direktorioa helburuko_erabiltzailea@helburuko_IP:/kokalekua
```

**3.3 Kodea:** SCP erabilera fitxategiak kopiatzeko.

Direktorioren bat kopiatu nahi badugu, -r gehitu behar da hasieran errekursiboki dena kopiatzeko. Fitxategi bakar baten kasuan ez da beharrezkoa ezer gehitzea.

### *Brain-Computer Interface*

---

#### 4.1 Sarrera

Pentsamenduaren bitartez kanpoko munduarekin elkarreragina izatea, azken urteotako ikerkuntza gai garrantzitsu bat izan da bai teknologia eta bai biologia munduan. Teknologia mota honen atzean dagoen ideia oso sinplea da eta edonoren eskura dago: helburua gure pentsamenduak inguruneko ekintza errealetan bilakatzea da. Ekintza hauek, argiak piztu eta itzali bezalako sinpleak edota gurgil-aulkiak maneiatzeko ekintza konplexuagoak izan daitezke. Ideia erraza da baina erronka teknologikoa izugarria da diziplina anitzeko taldeak nahasten baitira; neurozientzia, ingeniari-tza biomedikoa eta informatikoa esaterako. Ideia hauek errealitatera eramanez dira garun-ordenagailu interfazeei esker. Normalean, ingelesezko kontzeptua erabiltzen da, *Brain-Computer Interface* (BCI) eta sigla hauek erabiliko dira dokumentuan zehar. [6]

BCI sistema baten errendimendua eta zuzentasuna, faktore ugariaren menpe dago. Faktore hauek honakoak izan daitezke: seinaleak prozesatzeko algoritmoak, seinaleen ezaugarri esanguratsuak lortzeko metodoak, ezaugarri hauek ekintza edo komando bilakatzeo metodoak, komando hauek exekutatzeko tresnak etab. [7]

Sistema hauetan erabil daitezkeen seinale mota desberdin asko daude, esaterako elektroentzefalograma (EEG), elektromiografia (EMG), gorputz keinuak... Gure proiektuaren azterketa aldiz, EEG seinaleetan sakonduko da. Bestalde, EEG seinaleen azterketak egiten direnean, estimulu eta gertaerekin zerikusia duten ezaugarriak erauzteko orduan teknika ezberdinak erabiltzen dira. Ondorengo atalean, gertaeren aurreko potentzia aldaketen laburpen bat egingo da eta hauek antzemateko dauden teknika erabilienak azalduko dira.

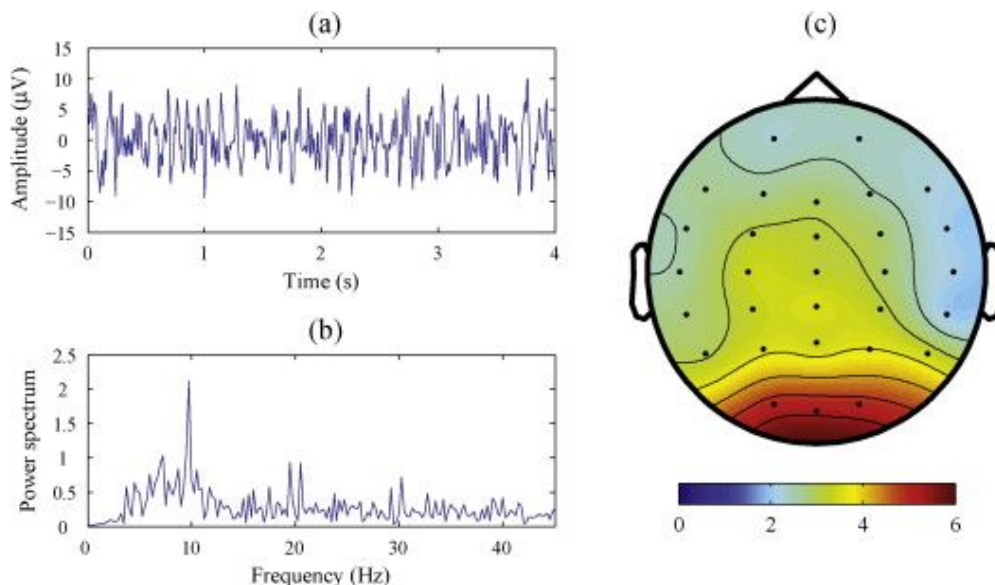
#### 4.2 ERP (Event-Related Potential)

*Event-Related Potential (ERP)*, gertaera edo estimulu baten aurrean garunean sortzen den potentzia aldaketa txiki bat da. Potentzia hauek, zentzumen, mugimendu edota aldaketa kognitiboekin zerikusia duten gertaeretan ageri daitezke. ERP bidez agertzen diren uhin mota asko daude baina gure azterketa, ikusmenarekin lotuta sortzen diren seinaleetan sakonduko dugu. Ikerketa hauetan erabili ohi diren teknika ezagunenak SSVEP eta P300 uhinak dira.

### 4.2.1 SSVEP

*Steady-State Visual Evoked Potential (SSVEP)*, gutxi gora behera 6-60 Hz bitarteko frekuentziarekin keinu egiten duen estimulu bat arretaz begiratzean sortzen den potentzia aldaketa positiboa da[8]. Frekuentzien potentziaren aldaketaz gain, armonikoak ere agertzen dira seinale hauen espektroetan. Teknika hau oso erabilia da BCI sistematan, seinalearen potentzia gehiena estimuluaren frekuentzian biltzen delako. Ezaugarri honi esker, oso erraza da potentzia jakin baten frekuentzia aldaketa hau bereiztea, eta beraz, estimulua begiratzen ari dela ziurtatzea. Potentzia aldaketa hau, lobulu okzipitalean ageri ohi da eta beraz, ezaugarri hau bereizi ahal izateko erabili beharko liratekeen elektrodoak O1 eta O2 dira nagusiak eta P7 eta P8 elektrodoetan ere ageri ohi da. Laburbilduz, erabiltzaileari, frekuentzia desberdinetan keinu egiten duen duten irudiak erakusten zaizkio, non estimulu bakoitza kanpo-ekintza bati lotuta dagoen [9]. Erabiltzaileak estimulu horietako bat arretaz begiratzen duenean, kanpo-ekintza hori burutzen da.

4.1 irudian ikus dezakegu SSVEP teknikaren adibide bat. (a) grafikoan denboraren menpeko seinalea ikus dezakegu. (b) irudian, potentziaren espektroa erakusten duen grafikoan argi ikus daiteke erabiltzaileak 10 Hz inguruko frekuentziaz keinu egiten duen irudi bat begiratu duela, eta armonikoak ageri direla frekuentzia honen multiploetan, 20, 30 eta 40 Hz-tan hain zuzen. Azkenik (c) irudian potentzia aldaketa hau ageri den garuneko atala erakusten da.



**4.1 Irudia:** SSVEP adibide baten ezaugarriak.

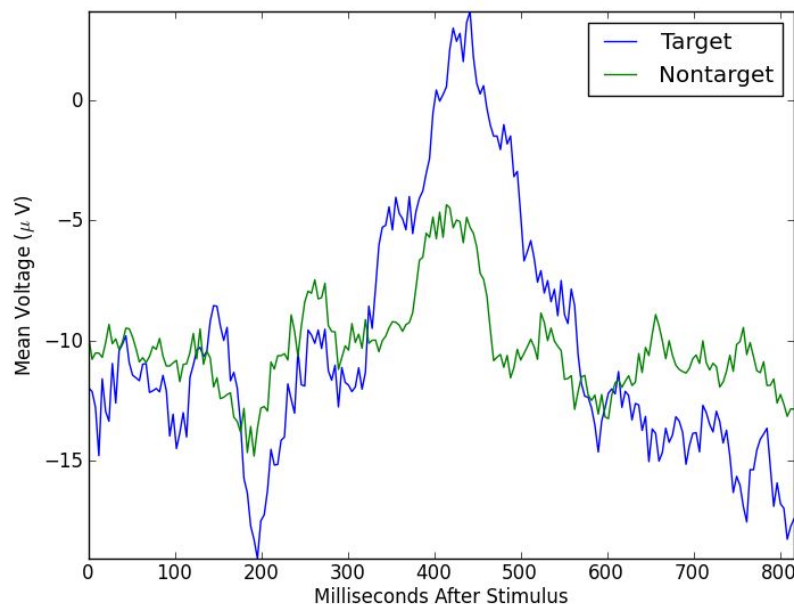
*Iturria:* <https://www.sciencedirect.com/science/article/pii/S016502701400106X>

SSVEP teknika, estimulu gutxi behar dituen BCI sistematan erabiltzea izango litzateke egokiena, erabiltzaileak frekuentzia desberdinetan keinu egiten duten irudi asko baditu pantailan, ez litzatekeelako oso eroso izango. Estimulu asko behar izanez gero, P300 uhinaren teknika erabiltzea gomendatzen da. Gure sisteman, bi estimulu besterik ez ditugu erabiliko eta beraz, SSVEP teknika erabiltzea erabaki da.

## 4.2.2 P300 uhina

P300 uhina, denboran zehar potentziaren anplitudean nabari ikus daitekeen aldaketa bat da. Erabiltzaileari esanguratsu egiten zaion estimulu bisual bat erakutsi behar zaio uhin hau agertzeko. Estimulua agertzen den unetik 300 ms-tara sortzen da potentzia aldaketa eta honen anplitudea nabarmen handitzen da. Aldaketa hau 300-1000 ms bitarteko tartean ikus daiteke. Uhin honen agerpena garuneko atal okzipitalean gauzatzen da SSVEP-ren antzera. Horretarako, azterketa egiteko orduan erabili beharko liratekeen elektrodoak O1, O2, P7 eta P8 dira. [10]

P300 uhina antzemateko erabiltzen den estrategia bi aukeren bereizketa paradigma egitea da. Aukera bat "target" da, non estimulua agertzeko probabilitatea baxua den. Bestalde, "non-target" diren estimuluen agertzeko aukera guztien probabilitatea altuagoa da. Bi klase hauen bereizketa egitean, ikus daiteke "target" seinalean P300 uhin bat agertzen dela erabiltzailearentzat arrotza edo desberdina den estimulu bat agertu delako. Estrategia honi, *Oddball* esaten zaio ingelesez. 4.2 irudian ikus daiteke nola "target" seinalean, hau da estimulu arrotz bat agertzen den seinalean, estimulua agertzetik gutxi gora behera 300 ms-tara potentzia aldaketa nabaria den.



**4.2 Irudia:** P300 uhinaren adibide bat *Oddball* estrategia erabiliz.

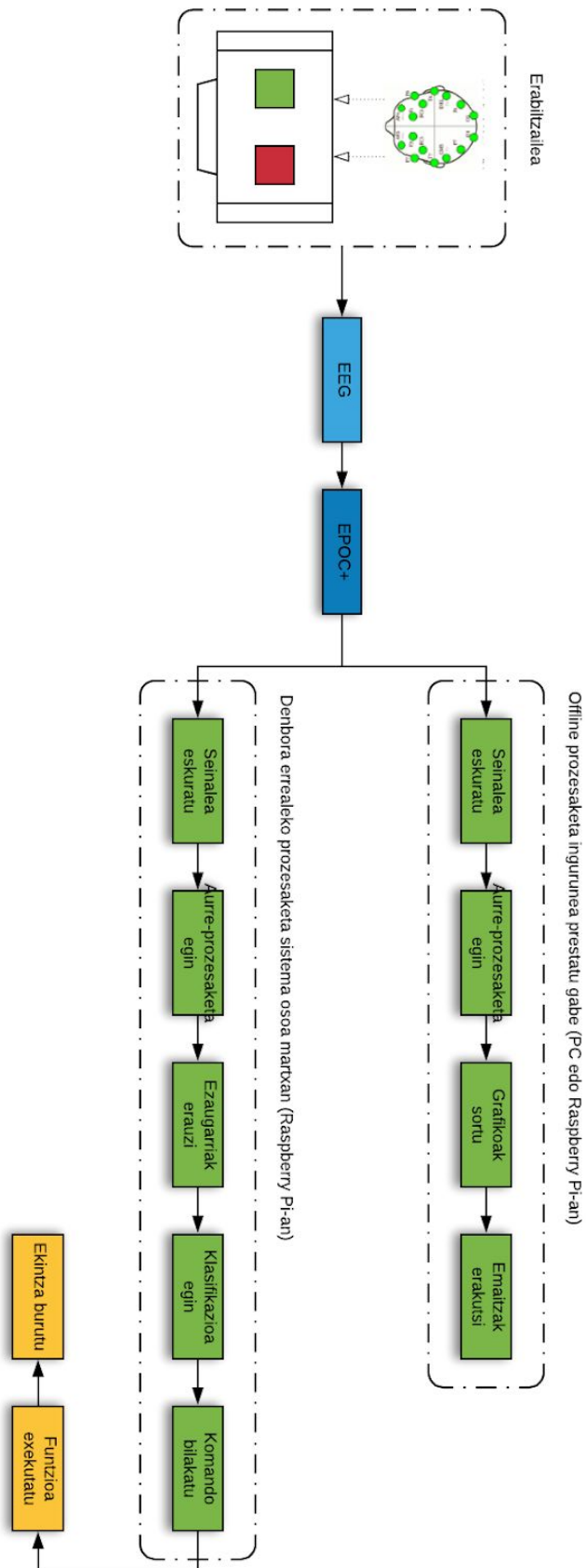
Iturria: [http://www.cs.colostate.edu/eeg/data/json/doc/tutorial/\\_build/html/p300\\_single\\_subject.html](http://www.cs.colostate.edu/eeg/data/json/doc/tutorial/_build/html/p300_single_subject.html)

## 4.3 Metodologia

Atal honetan, garatutako sistemaren metodologia azalduko da. Burutu den BCI-aren deskribapen sakona egingo da eta erabiliko den ingurunearen azalpena emango da. BCI sistemaren garapenaren nondik norakoak, hau da inplementazioak, funtzioak, ingurunearen programazioa, muntaia etab. hurrengo kapituluaz azalduko dira xehetasun handiz.

### 4.3.1 Sistemaren deskribapena

Garatu den sisteman, hainbat pauso jarraitu behar dira hasieran kaskotik jasotako elektrodoen balioak, amaierako ekintza bihurtu ahal izateko. Garatu den sistemaren helburu nagusia, LED bat itzali eta piztea da. Horretarako, erabiltzaileak estimulu keinukari jakin batzuk begiratu beharko ditu arretaz. Lortutako seinalea prozesatu beharko da eta programak jakin beharko du zein estimulu begiratu duen ekintza zuzena burutzeko, argia piztu edo itzali hain zuzen. Hori gauzatu ahal izateko programa SSVEP teknikan oinarritu da, non seinalearen espektroa ikusiz, jakin daiteke zein estimulu begiratu duen erabiltzaileak. Halaber, prozesu guzti horretan, hainbat pauso jarraitu behar dira, oso garrantzitsuak direnak garapenerako eta helburua lortzeko. Pauso hauen deskribapena egin ahal izateko, irudi baten laguntzaz egingo dugu.



4.3 Irudia: Sistemaren urratsak.

Irudian ikusten dugun bezala, sistema bi atal nagusitan banatu da. Egin diren proba batzuk, ingurunea prestatu gabe egin dira, hau da ez dira estimuluak erabili ezta ekintzak burutu ere. Atal honen helburua, kodearen garapenean zehar probak egitea da, emaitzak zein diren axola gabe. Proba hauek, Raspberry Pi-an edo PC-an egin dira eta urrats hauetan banatzen dira:

- Epoc+ kaskotik seinalea eskuratu. Pauso honetan, USB-aren eta kaskoaren arteko elkarrekintza gauzatzen dela ziurtatzen da. Lagin bakoitzean, elektrodo bakoitzaren seinalea, kalitatea, bateria portzentajea etab. lortzen ditugu.
- Ondoren, aurre-prozesaketa egitean, sisteman erabili nahi diren elektrodoen balioak lehenesten dira. Leiho bat erabiliz, laginketa bakoitzean elektrodo jakin baten edo batzuen balioak gordetzen ditugu.
- Leihoa betetzean, erabakitako elektrodo bakoitzaren balioen bektoreekin, nahi ditugun grafikoak sor ditzakegu. Kasu honetan, erabilgarriak eta informazio esanguratsua erakusten diguten grafikoak, denboraren menpeko seinalea, espektrograma eta potentziaren espektroa dira.
- Emaitzak pantailaratzean, ikus dezakegu ea lortutako seinaleak eta grafikoak zentzua duten, eta honen aurrean, ingurunea prestatu eta proba errealak egiten has gaitzke.

Bestalde, sistema osoa martxan dugunean eta ingurunea prest, denbora errealean probak egin dira. Erabiltzaileari estimuluak erakutsi eta prozesaketarekin has gaitzke. Atal honetan egiten diren urratsak honako hauek dira:

- Bestean bezala, Epoc+ kaskotik seinalea jasoko dugu lehendabizi. Horretarako USB eta kaskoaren arteko elkarrekintza zuzena dela ziurtatu beharko dugu noski. Seinalearen kalitatea egokia izan beharko du kasu honetan. Kalitate egokia lortzen dugula ziurtatzeko, Cortex UI programa erabil dezakegu probak egiten hasi baino lehen. Ikusi Emotiv Epoc+-aren erabilpen gida azalpen gehiago izateko.
- Oraingoan ere, aurre-prozesaketa egin beharko dugu, SSVEP teknika erabili baino lehen erabilgarriak izango zaizkigun elektrodoen balioak lortu beharko dira. Kasu honetan, erabiliko den leihoa, dinamikoa izango da. Denbora errealean lortuko dugu seinalea eta beraz, *overlapping* erabiliz, leihoa denboran zehar desplazatuko da eta seinalearen balioak gorde ahalko ditugu uneoro, ondoren ezaugarrien erauzketa egin ahal izateko.
- Ondoren, SSVEP teknikan lagungarri zaizkigun ezaugarriak erauzi beharko ditugu. Urrats honetan, seinaleari filtro bat ezarriko diogu frekuentzia jakin batzuen seinalea soilik eskuratu ahal izateko. Bestalde, hasieran inongo estimulurik eduki gabe, atalase-mailak kalkulatu dira, hauek erabiliko baitira ondoren gailurra dagoen edo ez erabakitzeko orduan.
- Filtroa pasa ondoren, klasifikazioa egin behar da. Ikusi beharko dugu seinalearen leiho bakoitzean, zein frekuentzia gailentzen den eta erabaki zein estimulu begiratu den. Hau da, jarri dezagun kasu bat non estimulu batek 10 Hz-tan keinu egiten duen eta beste estimulu batek 12 Hz-tan. Seinalearen filtroa zenbait leihori pasa ondoren ikusten badugu leiho guztien potentziaren espektroan 10 Hz-ko frekuentzian gailur nabari bat dagoela, esan nahiko du erabiltzailea estimulu horri begira dagoela. Bestalde, gailurra 12 Hz-tan aurkitzen badugu, estimulu horri begira egongo da. Garapenarekin amaitzeko, jasotako ezaugarri bakoitza komando batekin lotu

beharko dugu. Aurreko adibidea jarraituta, 10 Hz-ko frekuentzian gailur bat antzematean, argi bat pizteko komandoa exekutatu, eta 15 Hz-ko frekuentzian gailur bat ikustean, argia itzaltzeko komandoa.

- Azkenik, komandoa exekutatu eta funtzioak ekintza burutu duela ikusi beharko dugu, eta hau izango da erabiltzaileak jasoko duen *feedback-a*.

### 4.3.2 Ingurunearen deskribapena

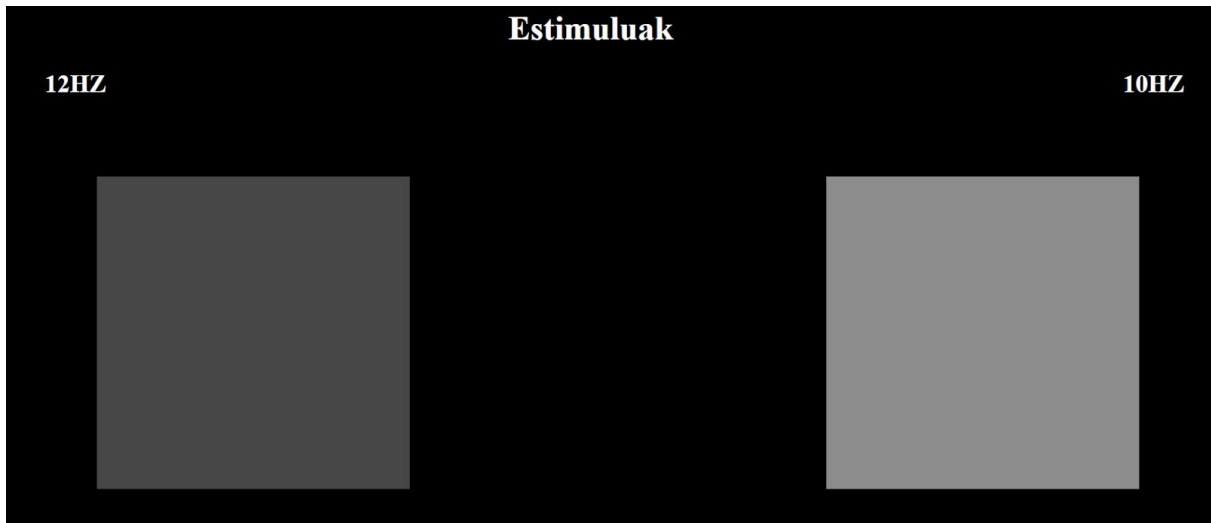
Atal honetan, BCI sisteman erabili den ingurunearen azalpena egingo da. Kasu honetan, bi azpiataletan banatuko da deskribapena. Hasteko, erabiltzaileari erakutsiko zaizkion estimuluak zein eta nolakoak diren azalduko da, eta zein funtzio izango duten. Ondoren, aurreko atalean azalduko urrats guztiak egindakoan burutu nahi den ekintzaren inguruko deskribapena egingo da, hau da zein izango den amaierako ekintza fisikoa eta honen xehetasunak.

#### 4.3.2.1 Estimuluak

Garatutako BCI sisteman, bi estimulu erabili dira, kolore zurkiko bi karratu hain zuzen. Bakoitzak, frekuentzia desberdinetan egiten dute keinu eta noiznahi aldatu ditzakegu frekuentzia hauek. Interfaze hau, HTML erabiliz egin da. Estimuluek seinalearen prozesaketarekin inongo erlaziorik ez duenez, hau da prozesaketa eta estimuluaren interfazea guztiz independenteak direnez, Python ez den beste lengoia batekin egin liteke estimuluen interfaze hau eta ez da beharrezkoa Raspberry Pi-an edukitzea. HTML lengoaiak, nahi dugun estimulua eta frekuentzia lortzeko erraztasuna ematen digu, eta txukunagoa geratzen da estimuluen interfaze hau.

SSVEP-ren azalpena egin denean, ikusi dugu teknika honetan gomendatzen den estimuluen keinu egitearen frekuentzia 6 eta 60 Hz bitartekoa dela, eta beraz, tarte horretako frekuentzia desberdinak erabil ditzakegu. 10Hz-ko karratuari begira gaudenean, kaskoa ipinita eta sistema martxan noski, LED bat piztea izango da helburua, horretarako, arretaz begiratu beharko dugu estimulua segundo batzuetan. Bestalde, 12Hz-ko estimulua begiratzean, LED-a itzali beharko da. 4.4 irudian ikus dezakegu estimuluen interfazea nolakoa den. Irudian efektuak ikusten ez diren arren, hurrengo kapituluan azalduko dugu interfaze honen inplementazioa nola izan den eta era berean, esteka batean eskuragarri izango dugu interfaze hau, probak egin nahi ezkerro edota efektua ikusi nahi ezkerro.





**4.4 Irudia:** Estimuluen interfazea.

#### 4.3.2.2 Ekintzak

Lehenago esan bezala, sistema honen ekintza Led bat kommutatzea da, itzali eta piztu. Horretarako, Raspberry Pi plakak eskaintzen digun sarrera-irteerako pinak erabili dira. Zirkuitua martxan dagoela, sistemak jasotako seinalea prozesatzean eta estimulu bat begiratu duela ziurtatzean, LED-ari esleitutako GPIO pinaren balioa aldatuko da, dagokion ekintza burutzeko.

---

# Ingurunearen implementazioa eta muntaia

---

## 5.1 Sarrera

Aurreko kapitulan, erabilitako ingurunearen deskribapena egin ondoren, kapitulu honetan ingurune horren implementazioaren eta muntaia nolakoa dela azalduko dira. Hasteko estimuluaren interfazearen implementazioa azalduko dugu eta amaitzeko kanpo ekintzaren, gure kasuan LED-aren muntaia nolakoa izan den argituko dugu.

## 5.2 Estimuluaren interfazea

Estimuluaren interfazea egiteko, deskribapenean esan bezala HTML erabili da, eskaintzen digun erraztasuna eta txukuntasunagatik. Hasierako web-orrian, aurkezpen bat egiten da, interfazean dituen aukerak eta azalpenak egiteko. Emotiv Epoc+ kaskoaren erabilpenarekin arazo edo zalantza bat badauka erabiltzaileak, kaskoaren erabilpen egokia izateko gida bat eskaintzen zaio, hasierako orritik eskuragarri duena botoi bat sakatuz. Bestalde, probarekin hasi nahi ezker, jarraitu beharko dituen urratsak azaltzen zaizkio, proba ahalik eta egokien egin ahal izateko. Proba hasteko beste botoi bat izango du aurkezpenaren web-orritik. Hurrengo kode zatian ikus dezakegu *Aurkezpena.html*-ren implementazioa.

```
<!DOCTYPE html>
<html>
<style>
.button {
  background-color: #4CAF50;
  border: none;
  color: white;
  padding: 15px 32px;
  text-align: center;
  text-decoration: none;
  display: inline-block;
  font-size: 16px;
  margin: 4px 2px;
  margin-top: 100px;
  cursor: pointer;
}
</style>
<meta name="viewport" content="width=device-width, initial-scale=1">
  <h1 style="color:white; text-align: center;"><u>BCI Sistemaren
estimuluak</u></h1>

  <h2 style="color:white; text-align: center;"> Kaixo, BCI sisteman
probak egin ahal izateko, estimuluak erakutsiko zaizkizu hemen.
```

Kaskoaren erabilpen eta kokapenaren inguruan zalantzarik baduzu, erabilpen gida irakur ezazu probarekin hasi baino lehen. Kaskoaren kokapena egokia denean, probarekin hasteko prest zaude. Proba hasteko botoia klikatzean, bi karratu agertuko dira. Ezkerrekoa 10Hz-tan klikatzen duena eta eskuinekoa 12Hz-tan. LED-a piztu nahi baduzu, eskuineko karratuari arretaz begiratu beharko diozu eta itzali nahi ezkerreko, ezkerreko karratuari. Sakatu botoia proba hasteko. </h2>

```
<body style="background-color:black; text-align: center;">
<button class="button"
onclick="window.location.href='Erabilpena.html';">ERABILPEN GIDA</button>
<button class="button"
onclick="window.location.href='Estimuluak.html';">PROBA HASI</button>
</body>
</html>
```

### 5.1 Kodea: Estimuluen interfazearen aurkezpena.

Bestalde, aipatu bezala Emotiv Epoc+ kaskoaren erabilpen gida bat eskuragarri dago hasierako web-orritik. Bertan, kaskoaren erabilpen eta kokapen egokia izateko jarraitu beharreko pausoak gomendatzen dira. Orri honetatik, zuzenean proba hasteko aukera izango du erabiltzaileak, kaskoaren erabilera zuzena izatean. 5.2 kodean ikus dezakegu HTML honen inplementazioa.

#### Erabilpena.html

```
<!DOCTYPE html>
<html>
<style>
.button {
  background-color: #4CAF50;
  border: none;
  color: white;
  padding: 15px 32px;
  text-align: center;
  text-decoration: none;
  display: inline-block;
  font-size: 16px;
  margin: 4px 2px;
  margin-top: 10px;
  cursor: pointer;
}
</style>
<meta name="viewport" content="width=device-width, initial-scale=1">
<h1 style="color:white; text-align: center;"> <u>EMOTIV EPOC+
ERABILPEN GIDA</u></h1>

<h2 style="color:white; text-align: left; margin-left:40px">
Kaskoaren kokapen eta erabilpen egokia ziurtatzeko hurrengo urratsak
jarraitu: </h2>
<h3 style="color:white; text-align: left; margin-left:60px"> 1.
Ziurtatu kaskoa kargatuta dagoela.</h3>
<h3 style="color:white; text-align: left; margin-left:60px"> 2.
Sentsoreak kaskoan ipini baino lehen, hauek hidratatzea oso garrantzitsua
```

da.

Horretarako salina likidoko tanta batzuk bota sentsore bakoitzari.</h3>

<h3 style="color:white; text-align: left; margin-left:60px"> 3. Kaskoko kanal bakoitzean sentsoreak ipini behar dira.</h3>

<h3 style="color:white; text-align: left; margin-left:60px"> 4. Ondoren, USB bidezko konexioa gauzatu nahi bada, USB Dongle-a PC-ra konektatu eta kaskoa piztu.

Bluetooth bidez egin nahi bada, konexioa gauzatu beharko da kaskoa piztuta dagoela.</h3>

<h3 style="color:white; text-align: left; margin-left:60px"> 5. Kaskoa buruan ongi kokatzea oso garrantzitsua da seinale egokia jaso ahal izateko.

Irudian ikus dezakezu kokapen egokia.</h3>



<h3 style="color:white; text-align: left; margin-left:60px"> 6. Seinalearen kalitatea neurtu nahi baduzu proba hasi baino lehen, "EMOTIV app" software ofiziala erabili.</h3>

<body style="background-color:black; text-align: center;">

<button class="button"

onclick="window.location.href='Estimuluak.html';">PROBA HASI</button>

</body>

</html>

## 5.2 Kodea: Estimulu interfazeko erabilpen gida.

Azkenik, proba egin ahal izateko estimuluak, *Estimuluak.html* orrian inplementatu ditut. Bertan, bi karratu erakusten dira. Bakoitzaren keinu egiteko frekuentziak, kodeko css estiloan aldatu ahalko dira, proba desberdinak egin behar izanez gero. Bestalde, karratu bakoitzaren estiloa definitzen duen css atalean, karratu hauen tamaina alda dezakegu, eta keinu egiteko ezaugarria gehitzen dion estiloaren atalean, frekuentzia aldatzeaz gain, karratu hauen koloreak alda ditzakegu nahi ezker. 10Hz-ko karratuaren tamaina eta keinu egiteko frekuentzia *.box* eta *.ten* klaseetan jasotzen da eta 12Hz-ko karratuarena *.box* eta *.twelve* klaseetan. Bestalde, keinu egiteko trantsizio koloreak (zuria eta beltza), *@-webkit-keyframes* klaseetan deskribatzen dira, 10-eko kasuan *glowing* eta 12-koaren kasuan *flashing*. Hurrengo kode zatian ikus dezakegu *Estimuluak.html* orrialdearen inplementazioa.

### Estimuluak.html

```
<!DOCTYPE html>
<html>
<style>
  h2{
    word-spacing: 970px;
  }
  .box{
    width:300px;
    height:300px;
  }
  .ten{
    background:#f00;
```

```

margin-top: 75px;
margin-bottom: 120px;
margin-right: 150px;
margin-left: 700px;
-webkit-animation: glowing 100ms infinite; <!-- frekuentzia : 10
hz-->
}
.twelve{
background:#0f0;
margin-top: 10px;
margin-bottom: 120px;
margin-right: 150px;
margin-left: 150px;
-webkit-animation: flashing 83.3ms infinite; <!-- frekuentzia : 12
hz-->
}
@-webkit-keyframes glowing {
0% { background-color: #FFF; }
100% { background-color: #000000; }
}
@-webkit-keyframes flashing {
0% { background-color: #FFF; }
100% { background-color: #000000; }
}
</style>
<meta name="viewport" content="width=device-width, initial-scale=1">
<h1 style="color:white; text-align: center;">Estimuluak</h1>
<body style="background-color:black;">
<h2 style="color:white; text-align: left; margin-left: 100px;">12HZ
10HZ</h2>
<div id="twelve" class="box twelve" >
<div id="ten" class="box ten" ></div>
</div>
</body>
</html>

```

### 5.3 Kodea: Estimuluen implementazioa.

Web-orrialde hau, hurrengo estekan eskuragarri egongo da edonorentzat, edozein momentutan probak egin nahi ezker.

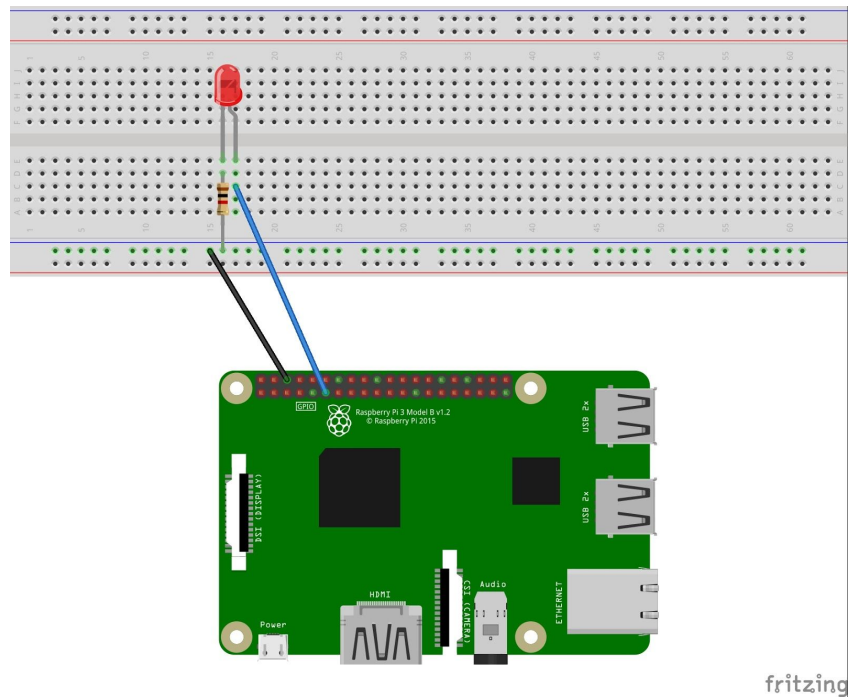
<https://estimuluak.000webhostapp.com/estimuluakWeb/Aurkezpena.html>

## 5.3 LED-aren muntaia

LED-aren zirkuitua muntatu ahal izateko erabili den materiala hau da:

- Raspberry Pi 3 Model B+.
- LED gorri bat.
- Ar-eme bi kable.
- 1 K $\Omega$ -eko erresistentzia bat.

Material horrekin, LED-aren zirkuitu sinplea muntatu ahalko dugu, Horretarako, Raspberry Pi-ko GPIO pin bat erabaki beharko dugu irteera moduan, gure kasuan GPIO17 izango dena. Zirkuitua, hurrengo irudian ikus dezakegu:



**5.1 Irudia:** LED-aren zirkuitua.

### Algoritmoaren inplementazioa

---

#### 6.1 Sarrera

Kapitulu honetan, garatu den algoritmoa azalduko da. Python-ek eskaintzen dizkigun hainbat funtzio erabil ahal izateko eta dependentziak saihesteko, hainbat pakete instalatu behar izan dira lehendabizi, eta hauek zein diren adieraziko da. Ondoren, algoritmoaren deskribapen sakona egingo da, funtzio bakoitzak zertarako erabili diren azalduz. Azkenik, algoritmoarekin lortzen dugun *feedback*-a erakutsiko da, hain zuzen seinaleen grafikoak, espektrogramak, datuak, etab.

#### 6.2 Erabilitako paketeak

Algoritmoaren garapenean zehar beharrezkoak izan diren paketeak hauek dira:

- USB gailuekin komunikatzeko eta haiekin tratatu ahal izateko, Python-ek `pyusb` paketea eskaintzen digu. Pakete honi esker, algoritmoak USB gailu guztien artean Emotiv Epoc+ kaskoaren USB hartzailea detektatu ahal izango du eta bertatik datuak jasotzen hasi. Raspberry Pi-an pakete hau instalatzeko hurrengo lerroa exekutatu terminal batean:

```
sudo pip install --pre pyusb
```

**6.1 Kodea:** `pyusb` instalazioa.

- Dimentsio desberdineko bektore eta matrizeekin lan egin ahal izateko, `numpy` lan paketearen laguntza izango dugu. Erabilpen zientifikoa du hala nola aljebra lineala edota Fourierren transformazioa bezalako azterketan egitea ahalbidetzen digu. Gure kasuan, kaskotik jasotako datuak matrize batean jasoko ditugu, non dimentsioak (elektrodo kopurua) $\times$ (lagin kopurua) diren. Bestalde, elektrodo jakin batzuen seinalearen datuei Fourierren transformazioa aplikatu da, haien frekuentzien analisia egiteko asmoz. Terminalean hurrengo exekutatu paketea instalatzeko:

```
sudo pip install numpy
```

**6.2 Kodea:** `numpy` instalazioa.

- Matematika, zientzia edota ingeniariatza bezalako arlotan lan egiteko, `scipy` paketea eskaintzen digu Python-ek. Pakete honek, `numpy` paketearekin menpekotasuna du,

hain zuzen, numpy paketeak erabiltzen dituen matrize eta bektoreak manipulatzeko edota eragiketak egiteko erabiltzen baita. Instalaziorako lerroa hau exekutatu:

```
sudo apt-get install python-scipy
```

**6.3 Kodea:** scipy instalazioa.

- Jasotako datuak grafikoki erakutsi ahal izateko eta emaitzak pantailaratu ahal izateko, matplotlib paketea erabili da. Numpy paketearekin sortu diren bektore eta matrizeak grafikoki erakutsi daitezke. Kasu honetan, seinalea denboraren menpe adierazteko grafikoak, espektrogramak edota potentzien grafikoak pantailaratzeko erabili da. Hurrengo lerroa terminalean exekutatu instalatu ahal izateko.

```
sudo apt-get install python-matplotlib
```

**6.4 Kodea:** matplotlib instalazioa.

- Azkenik, pycrypto paketea erabili da datuen segurtasuna bermatzeko. Pakete honek, zenbait hash funtzio (SHA256 eta RIPEMD160) eta enkriptatze algoritmo (AES, RES, RSA...) eskaintzen dizkigu. Honi esker, kaskoaren eta Raspberry Pi-aren arteko komunikazio segurua bermatuko dugu, datuak zifratuta doazela. Bestalde, datuak deszifratzeko beharrezkoa izango den funtzioa erabiliko dugu. Instalaziorako lerro hau exekutatu:

```
sudo pip install pycrypto
```

**6.5 Kodea:** pycrypto instalazioa.

Python pakete guzti hauek instalatuta, hurrengo atalean azalduko den algoritmoa arazorik gabe exekutatu ahalko dugu.

## 6.3 Algoritmoa

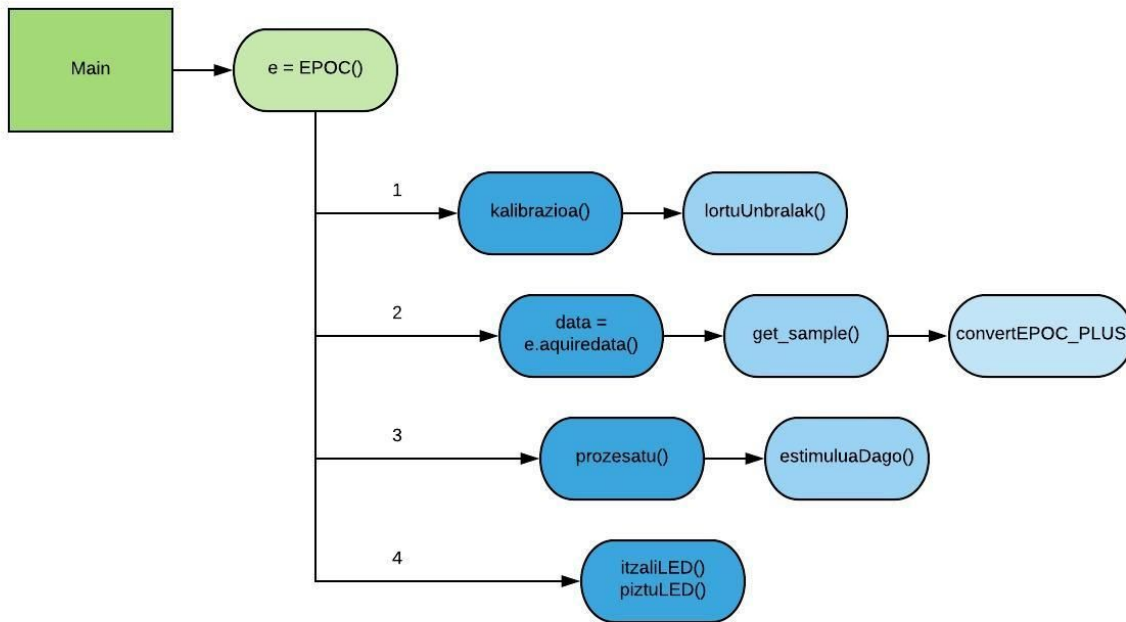
Algoritmoaren inplementazioa egiteko orduan, GitHub plataformako proiektu baten kodean oinarritu naiz. Kode honek, Emotiv Epoc bertsio zaharra erabiltzen zuen eta hasieran egin behar izan ditudan aldaketak, Epoc+ bertsioan funtzionatu ahal izateko aldaketak izan dira. Aurrerago azalduko dira aldaketa hauek zein izan diren eta haien xehetasunak. Esteka honetako proiektua hartu dut oinarri bezala:

<https://github.com/ozancaglayan/python-emotiv>

### 6.3.1 Eskema

Algoritmoaren azalpena ahalik eta errazen eta ulergarrien egiteko, algoritmoan erabilitako funtzio eta dei garrantzitsuenen eskema bat azalduko da. Hurrengo irudian ikus dezakegu, algoritmoak, main funtziotik abiatuta erabiltzen dituen azpifuntzio esanguratsuak.





**6.1 Irudia:** Algoritmoaren funtzioen eskema.

Hasteko, main funtzioak parametro eta hasieraketa guztiak egin ondoren, EPOC objektu bat sortzen du, non konektatuta dugun kaskoarekin konexioa gauzatzen den. Jarraian kalibrazioarekin hasten da, 10 segundoz datuak jaso eta erabiliko diren estimuluaren atalase-mailak kalkulatu dira. Ondoren, estimuluak aktibatuta ditugula, datuak jasotzen hasiko gara. Aurrez adierazitako denbora tartean, laginak jasotzen egongo da eta konbertsio baten ondoren, datu gordina lortuko dugu. Jasotako datu gordin hau prozesatu egin beharko dugu, erabakitzeke ea estimuluren bat begiratzeko egon den laginketaren denbora tartean. Azterketa honen ondoren eta jarraian hiru laginketetan ageri bada estimulu jakin bat begiratu dela, dagokion funtzioari (piztu edo itzali) deituko zaio algoritmoaren zikloa amaitzeko.

### 6.3.2 Azalpena

Hasteko, kodearen main funtzioan EPOC+ kaskoaren objektu bat sortu beharko dugu, dagozkien atributuekin. Ondoren erabiliko diren parametro eta funtzio gehienak, objektu honen gainean egingo baitira. Objektu honen eraikitzailean, ondorengo kode zatian ikus dezakegun parametroak definituko dira.

```

def __init__(self, method="libusb", serial_number=None, enable_gyro=True):
    self.vendor_id = None
    self.product_id = None
    self.decryption = None
    self.decryption_key = None
    self.headset_on = False
    self.counter = 0
    self.method = method

    #Kaskoaren serie zenbakia
  
```

```

self.serial_number = serial_number

# libusb device eta endpoint
self.device = None
self.endpoint = None

#Kaskoaren kanalen maskara bat
self.channel_mask = self.channels

#Funtzio bat USB gailuak zerrendatzeko
self.enumerate()

```

### 6.6 Kodea: Funtzio eraikitzailea.

Objektua sortzeko garaian, pausu esanguratsu bat USB Dongle-arekin konexioa gauzatzea da. Pausu honetan, Raspberry Pi-an konektatuta dauden kanpoko gailuak zerrendatzen dira(1) eta Emotiv Epoc+ kaskoa aurkitzen badu(2) konexioa gauzatzen da. Horretarako ondorengo funtzioak erabiltzen dira, enumerate funtzioa gailuak zerrendatzeko erabiltzen da eta is\_epoc ea gailua Epoc+ kaskoa den ziurtatzen du. Funtzio hauek, github-etik eskuratutako proiektutik mantendu ditut:

```

def _is_epoc(self, device):
    """Custom match function for libusb."""
    try:
        manu = usb.util.get_string(device, device.iManufacturer) (2)
    except usb.core.USBError, usb_exception:
        # If the udev rule is installed, we shouldn't get an exception
        # for Emotiv device.
        print usb_exception
        return False
    else:
        if manu and manu.startswith(self.MANUFACTURER_PREFIX):
            print manu
            return True

    def enumerate(self):
        """USB gailuak zerrendatu eta haien ezaugarriak begiratzen ditu"""
        devices = usb.core.find(find_all=True, custom_match=self._is_epoc) (1)
        #Lortu Epoc+ diren gailuak, _isepoc funtzioari deia eginez.

        if not devices:
            raise EPOCHNotPluggedError("Emotiv EPOCH not found.")

        for dev in devices: #Gailuaren ezaugarriak gorde
            serial = usb.util.get_string(dev, dev.iSerialNumber)
            if self.serial_number and self.serial_number != serial:
                # If a special S/N is given, look for it.
                continue

            # Record some attributes

```

```

        self.serial_number = serial
        self.vendor_id = "%x" % dev.idVendor
        self.product_id = "%x" % dev.idProduct

        if self.method == "libusb":
            # Last interface is the one we need
            for interface in dev.get_active_configuration():
                if
dev.is_kernel_driver_active(interface.bInterfaceNumber):
                    # Detach kernel drivers and claim through libusb

dev.detach_kernel_driver(interface.bInterfaceNumber)
                    usb.util.claim_interface(dev,
interface.bInterfaceNumber)

                self.device = dev

                self.endpoint = usb.util.find_descriptor(
                    interface, bEndpointAddress=usb.ENDPOINT_IN | 2)
            elif self.method == "direct":
                if os.path.exists("/dev/emotiv_epoc"):
                    self.endpoint = open("/dev/emotiv_epoc")
                else:
                    raise EPOCDeviceNodeNotFoundError(
                        "/dev/emotiv_epoc doesn't exist.")

            # Return the first Emotiv headset by default
            break

        self.setup_encryption()
        # Attempt to see whether the headset is turned on
        try:
            self.endpoint.read(32, 100)
        except usb.USBError as ue:
            if ue.errno == 110:
                self.headset_on = False
            print "Setup is OK but make sure that headset is turned on."
        else:
            self.headset_on = True

```

### 6.7 Kodea: Emotiv-en usb gailuarekin konexioa.

Beste funtzio garrantzitsu bat, objektua sortzeko garaian, eskuratuko ditugun datuak deszifratzeko gakoa lortzeko metodoa da. Metodo hau, desberdin egiten zen Epc bertsiio zaharrean eta egindako aldaketa ondorengo kode zatian ikus dezakegu.

```

def setup_encryption(self):
    """ Dekripzio gakoa sortzeko metodoa.
        Gako hau, kaskoaren serie zenbakitik lortzen da.
        Gako honi esker, kaskotik zifratuta lortuko ditugun
        datuak desenkriptatu ahalko ditugu.
    """

    sn = self.serial_number

```

```

k = ['\0'] * 16

k =
[sn[-1], sn[-2], sn[-2], sn[-3], sn[-3], sn[-3], sn[-2], sn[-4], sn[-1], sn[-4], sn[
-2], sn[-2], sn[-4], sn[-4], sn[-2], sn[-1]]
key = ''.join(k)

self.decryption_key = ''.join(k)
self._cipher = AES.new(self.decryption_key)

```

### 6.8 Kodea: Datuak deszifratzeko gakoa.

Objektua sortzeko orduan esanguratsuak diren atazak azalduta, ondoren datuen eskuraketa eta prozesamenduari zerikusia duen implementazioa dator. Datuak eskuratzeko, *main* funtzioa bi zati nagusitan banatzen da. Lehenengo zatian, 4 segundoko irakurketa(3) egiten da, exekuzioaren hasieran, eta 4 segundoko pakete hau prozesatzen(4) da ondoren. Jarraian, bigarren zatian, aurreko paketeko azken 3 segundoko datuak erabiltzen dira eta bertan gehitzen dira segundo bateko irakurketa berri baten datuak. Honi esker, *overlapping* izeneko teknika erabiltzen da, non leiho bat erabiliz, datuen paketeak desplazatu egiten diren denboran zehar. Bigarren zati honetan ere, prozesaketa gauzatzen da, aurrekoan bezalaxe. Hurrengo kode zatia da *main* funtzioa, hasieran parametro lokalak definituko dira eta ondoren lehen aipatutako prozesaketarekin hasiko da:

```

def main():

    e = EPOC()
    LED.setupLED()
    LED.itzaliLED()

    oldata = []
    o2data = []
    p7data = []
    hiruak = []
    media = 0
    seconds = 4
    kont = 1
    sampling_rate = 128
    batFreq = 10
    biFreq = 12
    batHZkop = 0
    biHZkop = 0

    print "Kalibrazioaren hasiera..."
    batUnbral, biUnbral = kalibrazioa(e, batFreq, biFreq)
    print batUnbral, biUnbral
    print "Kalibrazioa amaitu da, aktibatu estimuluak."

    while 1:
        if kont == 1:
            oldata = []
            o2data = []
            p7data = []
            hiruak = []

```

```

media = 0
try:
    data = e.acquire_data(seconds) (3)

    if data is not None:

        for i,channel in enumerate(e.channel_mask):
            if channel == "O1":
                o1 = i
            if channel == "O2":
                o2 = i
            if channel == "P7":
                p7 = i

        for j in range(0, (seconds * sampling_rate)):
            o1data.append(data[j][o1])
            o2data.append(data[j][o2])
            p7data.append(data[j][p7])

        for i in range(0, len(o1data)):
            media = (o1data[i] + o2data[i] + p7data[i]) / 3
            hiruak.append(media)

except EPOCHTurnedOffError, etc:
    print etc
except KeyboardInterrupt, ki:
    e.disconnect()
return 0

```

```

    batHZkop, biHZkop = prosesatu(hiruak, sampling_rate, seconds,
    batUnbral, biUnbral, batFreq, biFreq, batHZkop, biHZkop) (4)

```

```

kont += 1
oldata = oldata[127:(seconds*sampling_rate)-1]
o2data = o2data[127:(seconds*sampling_rate)-1]
p7data = p7data[127:(seconds*sampling_rate)-1]
hiruak = hiruak[127:(seconds*sampling_rate)-1]

else:
    overlap = 1
    leihoa = seconds - overlap
    try:
        data = e.acquire_data(overlap)
        if data is not None:
            for j in range(0, (overlap * sampling_rate)):
                o1data.append(data[j][o1])
                o2data.append(data[j][o2])
                p7data.append(data[j][p7])

            for i in range(leihoa*sampling_rate, len(oldata)):
                media = (o1data[i] + o2data[i] + p7data[i]) / 3
                hiruak.append(media)

```

```

    except EPOCHTurnedOffError, etc:
    print etc
    except KeyboardInterrupt, ki:
    e.disconnect()
    return 0

    batHZkop, biHZkop = prozesatu(hiruak, sampling_rate, seconds,
batUnbral, biUnbral, batFreq, biFreq, batHZkop, biHZkop)

    if batHZkop > 2:
    print str(batFreq) + "BEGIRATUTA LED-a PIZTU DA"
    LED.piztuLED()
    elif biHZkop > 2:
    print str(biFreq) + "BEGIRATUTA LED-a ITZALI DA"
    LED.itzaliLED()

    oldata = oldata[127:(seconds*sampling_rate)-1]
    o2data = o2data[127:(seconds*sampling_rate)-1]
    p7data = p7data[127:(seconds*sampling_rate)-1]
    hiruak = hiruak[127:(seconds*sampling_rate)-1]

```

### 6.9 Kodea: Main funtzioa.

Aurreko kode zatian ikusten den bezala, pakete baten irakurketa `acquire_data(seconds)` funtzioaren deiarekin egiten da. Funtzio honek, nahi diren segundo kopuruak emanda, datu pakete bat itzultzen du. Adibidez, laginketa frekuentzia 128Hz izanik, demagun 4 segundoko irakurketa egiten dugula. Funtzioaren deiak, matrize bat itzuliko liguke non dimentsioak  $(4 \times 128) \times (\text{sentsoreKop})$  diren. Aldi berean, `acquire_data` funtzioak `get_sample` funtzioari `dei(5)` egiten dio eta honek lagin baten bektore bat itzultzen digu. Funtzioak USB Dongle-etik datu gordinak irakurtzen(6) ditu eta datu horien konbertsio bat egin ondoren bektore batean itzultzen ditu. Hurrengo kode zatietan ikus ditzakegu hiru funtzio hauek.

```

def acquire_data(self, duration):
    """Epoc+ kaskotik informazio jasotzeko metodoa.
    Iraupena adierazita, (iraupena * laginketa frekuentzia) lagin
    sortuko ditu"""

    total_samples = duration * self.sampling_rate #Lagin kopuru totala
    definitu
    _buffer = np.ndarray((total_samples, len(self.channel_mask) + 1),
        dtype=np.uint16) #Matrize moduko buffer bat sortu, non
    dimentsioak (laginKop * sentsoreKop) den
    ctr = 0

    while ctr < total_samples: #Kontagailu bat lagin guztiak lortzeko.
        # Datuak lortu get_sample() funtzioari dei eginez.
        data = self.get_sample() (5)

    if data: #Pakete hutsa ez bada
        _buffer[ctr] = np.insert(np.array(data), 0, self.counter)

```

```

#Bufferrean sentsoreen paketea gehitu.
    ctr += 1
    return _buffer #Bufferra itzuli

def get_sample(self):
    """Datu paketeak itzultzen dituen metodoa."""
    try:

        packet_data = [] #Paketea hasieratu
        task = self.endpoint.read(32) #Lehen definitutako gailutik,
32 Byte irakurri
        data = self._cipher.decrypt(task) #Jasotako datuei, lehen
sortutako desenkriptazio gakoa aplikatu (6)

        if( ord(data[0]) < 128 ): #Lehen Byte-a kontagailua da. 1-127
bitartekoa dena.

            #Bigarren Byte-ak, adierazten digu datorren paketea zein
motatakoa den.
            #16 bada, datorren paketeak, sentsoreen balioen informazioa
edukiko du. 32 bada, gyro-ei buruzko informazioa duen paketea izango da.

            if(ord(data[1]) == 16): #Guri interesatzen zaiguna soilik
sentsoreen balioak dira.

                for i in range(2,16,2): #2-tik 16-rako Byte-ak lehen
7 sentsoreen datuak dira. Sentsore bakoitzari 2 Byte dagozkio.
                    packet_data.append(convertEPOC_PLUS(ord(data[i]),
ord(data[i+1]))) #Sentsore bakoitzaren bi Byte-ekin konbertsioa egin

                    #Ondorengo 2 Byte-ak, kaskoaren bateria adierazteko
erabiltzen dira.
                    for i in range(18,len(data),2): #Azken 7 Byte-ak
gainotzeko sentsoreen datuak dira.
                        packet_data.append(convertEPOC_PLUS(ord(data[i]),
ord(data[i+1]))) #Konbertsioa egin (6)

        #Datu paketea itzuli
        return packet_data
    else:
        # Set a synthetic counter for this special packet: 128
        self.counter = 128
        # Parse battery level
    except usb.USBError as usb_exception: #Salbuespena kaskoaren
konexioarekin arazoren bat gauzatu bada
        if usb_exception.errno == 110:
            self.headset_on = False
            raise EPOCTurnedOffError(
                "Make sure that headset is turned on")
        else:
            raise EPOCUSBError("USB I/O error with errno = %d" %
                usb_exception.errno)

def convertEPOC_PLUS(value_1, value_2):

```

```

"""Datu gordinari gauatzen zaion konbertsioa uV eskalara pasatzeko"""
edk_value = (((int(value_1) * .128205128205129) + 4201.02564096001 ) +
((int(value_2) - 128) * 32.82051289))
return edk_value

```

#### 6.10 Kodea: Datu paketeak lortzeko funtzioak.

Datuen prozesaketa hasi baino lehen, hasieran kaskoaren kalibrazio(7) bat egiten da, aurrerago erabiliko diren bi datu garrantzitsu lortzeko. Datu hauek, bi frekuentzien atalase-mailak(8) dira. Horretarako, 10 segundotan, inongo estimulurik begiratu gabe, bi frekuentzien inguruko potentziak jaso eta haien mediana(9) kalkulatu da, eta mediana hau izango da frekuentzia bakoitzak erabiliko duen atalase-maila ondorengo prozesaketetarako. Kalibrazio hau, datuen eskuraketa bezala gauzatzen da, leiho labainkor bat erabiliz. Kalibrazio hau hurrengo kode lerroetan definituta dago:

```

def kalibrazioa(e, batFreq, biFreq):
"""Atalase mailak kalkulatzeko kalibrazioa. 10 segundotan irakurketak
egiten dira eta leiho labainkorra erabiliz.
Aztertu nahi diren frekuentzien inguruko potentzien balioak lortu eta haien
mediana erabiliko da atalase moduan."""
    sampling_rate = 128
    batUnbralTot = []
    biUnbralTot = []
    seconds = 4
    o1data = []
    o2data = []
    p7data = []
    hiruak = []

```

```

#Lehenengo irakurketa 4 segundokoa da.
kalibrazioa = e.acquire_data(seconds) (7)

```

```

if kalibrazioa is not None:

for i,channel in enumerate(e.channel_mask):
    if channel == "O1":
        o1 = i
    if channel == "O2":
        o2 = i
    if channel == "P7":
        p7 = i

for j in range(0, (seconds * sampling_rate)):
    o1data.append(kalibrazioa[j][o1])
    o2data.append(kalibrazioa[j][o2])
    p7data.append(kalibrazioa[j][p7])

for i in range(0, len(o1data)):
    media = (o1data[i] + o2data[i] + p7data[i]) / 3
    hiruak.append(media)

```



```
batUnbral, biUnbral = lortuUnbralak(batFreq, biFreq, hiruak,
sampling_rate, seconds)#lortuUnbralak deiarekin, definitutako bi
frekuentzien behin behineko atalase mailak kalkulatzeko dira uneko
irakurketarako. Kalkulua, inguruko frekuentziek duten potentzien
balioen batz besteko bat eginez egiten da. (8)
```

```
batUnbralTot.append(batUnbral)
biUnbralTot.append(biUnbral)
```

```
oldata = oldata[127:(seconds*sampling_rate)-1]
o2data = o2data[127:(seconds*sampling_rate)-1]
p7data = p7data[127:(seconds*sampling_rate)-1]
hiruak = hiruak[127:(seconds*sampling_rate)-1]
```

```
#Gainontzeko 6 irakurketak segundo batekoak dira, aurreko
laginketako datuekin elkartzen delako, leiho labainkorra erabiliz.
```

```
for i in range(0, 6):
kalibrazioa = e.acquire_data(1)
```

```
if kalibrazioa is not None:
```

```
for j in range(0, (1 * sampling_rate)):
oldata.append(kalibrazioa[j][o1])
o2data.append(kalibrazioa[j][o2])
p7data.append(kalibrazioa[j][p7])
```

```
for i in range((seconds - 1)*sampling_rate, len(oldata)):
media = (oldata[i] + o2data[i] + p7data[i]) / 3
hiruak.append(media)
```

```
batUnbral, biUnbral = lortuUnbralak(batFreq, biFreq, hiruak,
sampling_rate, seconds)#lortuUnbralak deiarekin, definitutako bi
frekuentzien behin behineko atalase mailak kalkulatzeko dira uneko
irakurketarako. Kalkulua, inguruko frekuentziek duten potentzien balioen
batz besteko bat eginez egiten da.
```

```
batUnbralTot.append(batUnbral)
biUnbralTot.append(biUnbral)
```

```
oldata = oldata[127:(seconds*sampling_rate)-1]
o2data = o2data[127:(seconds*sampling_rate)-1]
p7data = p7data[127:(seconds*sampling_rate)-1]
hiruak = hiruak[127:(seconds*sampling_rate)-1]
```

```
stdBat = np.std(batUnbralTot)
stdBi = np.std(biUnbralTot)
```

```
batUnbral = np.median(batUnbralTot) + stdBat
biUnbral = np.median(biUnbralTot) + stdBi (9)
```

```
#Irakurketa guztien atalase mailak kalkulatu, guzti hauen mediana
erabiltzen da desbiderapen estandarra gehituz, behin betiko atalase maila
bezala.
```

```
return batUnbral, biUnbral
```

### 6.11 Kodea: Kalibrazioa egiteko funtzioa.

Behin kalibrazioan atalase-mailak definituta, gainontzeko datuen prozesaketa egingo da, non erabiltzaileari, bi estimulu desberdin begiratzeko aukera emango zaion. Atal honen datuen prozesaketan, kalibrazioaren ataleko eskema berdina jarraitzen da. Kasu honetan aldiz, aztertzen diren frekuentzien potentziaren balioak ea atalase-mailak gainditzen dituen begiraten da(10). Frekuentzia baten potentziak ondoz-ondoko 3 laginetan gainditzen baditu honen atalase-mailaren bikoitza, frekuentzia horri dagokion estimulua begiratu dela suposatzen da eta estimulu horrek esleituta duen ekintza gauzatzen da. Kalkulu hori funtzio honetan egiten da:

```
def prozesatu(ydata, sampling_rate, seconds, batUnbral, biUnbral, batFreq,
biFreq, batHZkop, biHZkop):

    win = seconds * sampling_rate
    freqs, psd = signal.welch(ydata, sampling_rate, nperseg=win)
    #Funtzio hauekin lortuko dugu ea definitutako frekuentzietan
    atalase-mailak nabarmen gainditzen diren.

    batHZ = estimuluaDago(freqs, psd, batFreq - 0.5, batFreq + 0.5,
batUnbral)
    biHZ = estimuluaDago(freqs, psd, biFreq - 0.5, biFreq + 0.5,
biUnbral) (10)

    #Frekuentziaren batean atalase-maila gainditzen bada, haren
    kontagailua inkrementatu.
    if batHZ and not biHZ:
        print (str(batFreq) + " Hz-ko estimulua begiratu duzu")
        batHZkop += 1
        biHZkop = 0
    if biHZ and not batHZ:
        print (str(biFreq) + " Hz-ko estimulua begiratu duzu")
        biHZkop += 1
        batHZkop = 0
    if batHZ and biHZ:
        batHZkop += 1
        biHZkop += 1
    if not batHZ and not biHZ:
        print "Ez duzu estimulurik begiratu"
        batHZkop = 0
        biHZkop = 0

    return batHZkop, biHZkop
```

### 6.12 Kodea: Datu paketeak prozesatu.

Azkenik, lehen main funtzioan ikusi dugun bezala, prozesaketaren ondoren frekuentziaren baten kontagailua 3-ra iristen bada, hau da, ondoz-ondoko 3 laginketa eta prozesaketaren ondoren frekuentzia batek atalase-mailaren bikoitza gainditu duenean, frekuentzia horri dagokion ekintza burutzea besterik ez da falta. Ondorengo lerroetan definitzen da ekintza gauzatu behar den edo ez:

```

if batHZkop > 2:
    print str(batFreq) + "BEGIRATUTA LED-a PIZTU DA"
    LED.piztuLED()
elif biHZkop > 2:
    print str(biFreq) + "BEGIRATUTA LED-a ITZALI DA"
    LED.itzaliLED()

```

**6.13 Kodea:** Estimulua begiratu den zehazteko kode zatia.

LED-aren ekintza gauzatzeko, zirkuitua muntatuta dugula, LED-ari esleitu diogun GPIO pina 1 jartzea piztu nahi bada eta 0 jartzea itzali nahi bada besterik ez da. Ondorengo funtzioak soilik erabili dira LED-arekin elkarrekintza gauzatzeko.:

```

import RPi.GPIO as GPIO
def setupLED():
    GPIO.setmode(GPIO.BCM)
    GPIO.setup(17, GPIO.OUT) #GPIO 17 irteera moduan

def itzaliLED():
    GPIO.output(17, False)

def piztuLED():
    GPIO.output(17, True)

def cleanup():
    GPIO.cleanup()

```

**6.14 Kodea:** LED-aren funtzioak.

Honekin amaitzen da algoritmoaren azalpen sakona.

## 6.4 Feedback

Erabiltzailea proba egiten hasten bada, hasieran kalibrazioa egiteko eskatuko zaio, non inolako estimulurik begiratu gabe egon beharko duen segundo batzuez. Kalibrazioaren amaieran, estimuluak aktibatzeke eskatuko zaio erabiltzaileari. Hemendik aurrera, monitorean segundoro azalduko zaio ea estimuluren bat begiratu duen edo ez duen estimulurik begiratu. Laginketa bakoitzaren prozesaketaren ondoren jasoko duen *feedback*-a izango da. Adibidez, hurrengo irudian ikus dezakegun moduan, guztiz zehatza ez den arren, tarte batzuetan jasotzen ditugun emaitzak, estimulu bat bera begiratzen dugula dira. Lehen zatian 10 Hz-ko estimulua begiratu dugula antzeman dezakegu eta bigarren zatian 12 Hz-koa. Izan ere, estimulu horiek begiratu ziren denbora tarte horietan.



# 7. Kapitularia

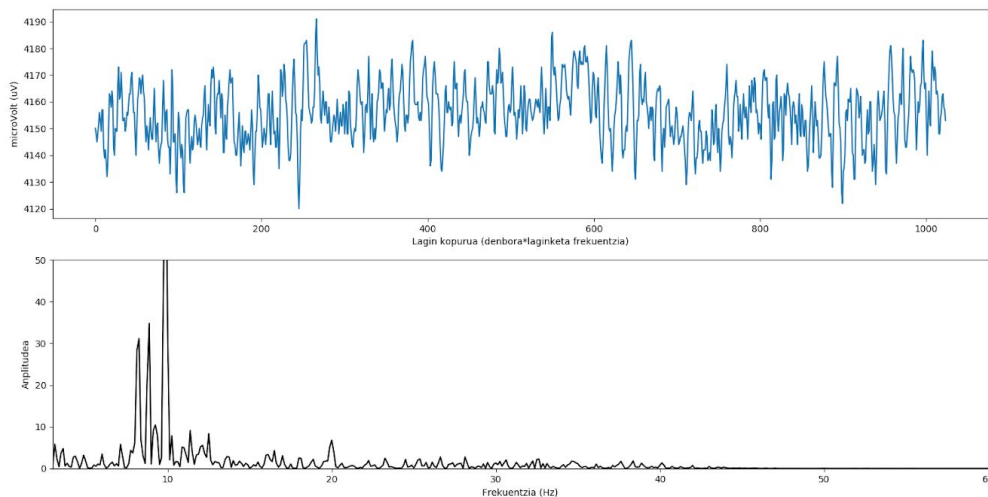
## Probak eta emaitzak

### 7.1 Sarrera

Kapitulu honetan, algoritmoaren garapenean zehar egin diren probak eta amaierako emaitzak azalduko dira.

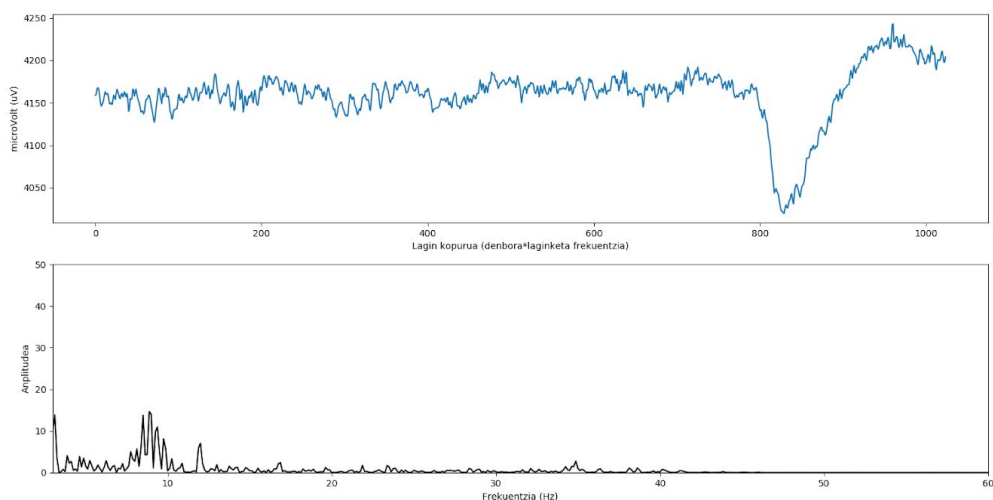
### 7.2 Garapenean egindako probak

Algoritmoaren garapenean zehar, zailtasun asko izan ondoren, lehenik egindako probetan seinalearen grafikoak pantailaratu dira. Honi esker, ziurtatu ahal izan da jasotako seinaleak zentzua zuela eta estimuluak begiratzuz, ikusi zitekeen nola frekuentzia horretan gailur bat agertzen zen seinalearen potentziaren espektroan. Hurrengo irudian adibidez, ikus daiteke nola 10 Hz-ko gailurra oso nabaria dela eta bere lehen armonikoa, 20 Hz-koa hain zuzen ere antzeman daiteke.



**7.1 Irudia:** 10Hz-ko estimuluaren grafikoak.

Esan beharra dago, inongo estimulurik eduki gabe, 7-10Hz bitarteko frekuentziak direla gehien nabarmentzen direnak, hasierako kapituluan azaldu den bezala, egoera lasai batean agertzen diren frekuentziak alfa erritmoari dagozkie nagusiki. Beraz tarte horretako frekuentzia baten estimuluak begiratzean, sortzen den gailurra oso nabaria dela. Izan ere, 12 Hz-ko estimulua ikustean sortzen diren grafikoak ez dira oso esanguratsuak, 7.2 irudian ikus dezakegun bezala.



**7.2 Irudia:** 12Hz-ko estimuluaren grafikoak.

Grafiko hauek ikusita, argi ikus dezakegu frekuentzia baten gailurraren maximoa bestearen oso desberdina dela. Arazo hau ekiditeko, algoritmoaren azalpenean ikusi dugun kalibrazioa egiten da hasieran, non frekuentzia bakoitzak, bere atalase-maila propioa kalkulatu duen, eta kalibrazio hori eginez jakin dezakegu noiz nabarmentzen den frekuentzia baten potentzia-maila, egoera normalarekiko konparazioa eginez.

### 7.3 Amaierako emaitzak

Algoritmoaren azken bertsioan, ez dira grafikoak erakusten eta erabiltzaileak soilik 6.4 atalean azaldutako *feedback*-a jasotzen du. Honen harira, LED-aren zirkuitua martxan dagoela, sortu dugun BCI sistemaren eraginkortasuna kalkulatu da, 5 probetan zehar LED-a 5 aldiz konmutatzeko (3 alditan piztu eta 2 alditan itzali) behar izan den denbora kalkulatu. Hurrengo taulan jasotzen dira emaitza hauek:

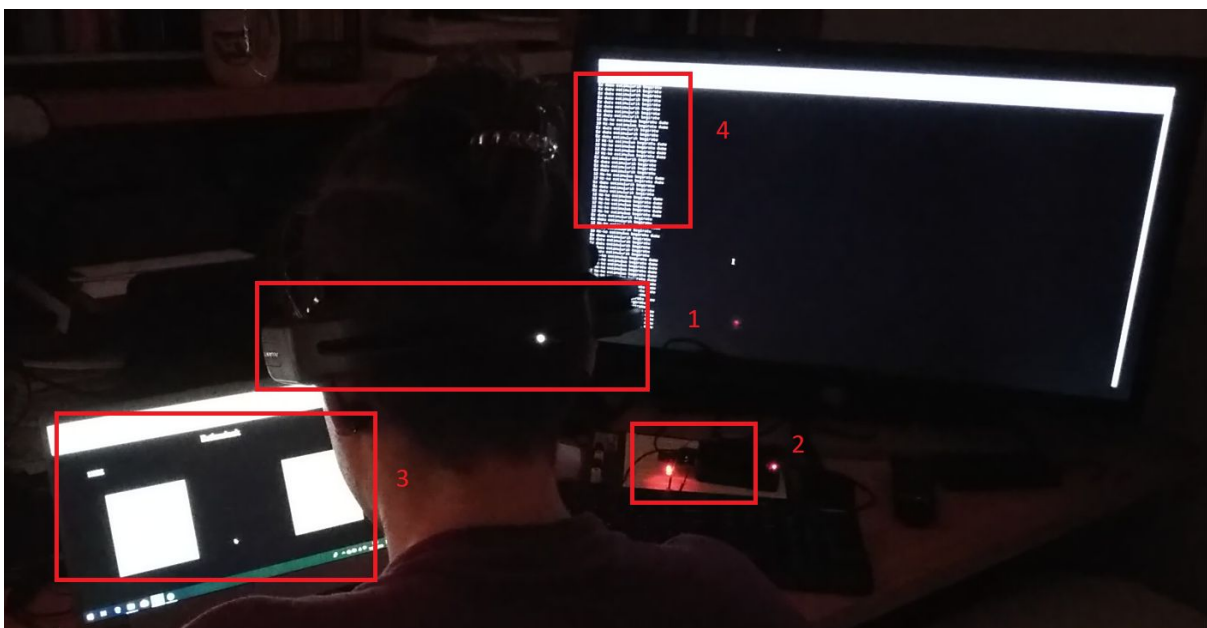
5 ALDITAN KOMMUTATZEKO DENBORAK	
Proba	Denbora
1	26s
2	22s
3	24s
4	20s
5	27s
Bataz bestekoa	23.8s

**7.1 Taula:** LED-a 5 alditan konmutatzeko denborak.

Algoritmoaren azalpenean esan bezala, frekuentzia baten estimulua begiratu dela erabakitzeke, segundo bateko 3 laginketan zehar azaldu behar da frekuentzia hori begiratu dela. Hori horrela izanik, lortu dezakegun eraginkortasun ideala, 3 segundoro kommutazio bat gauzatzea da, hau da 5 kommutazioak 15 segundotan. Gure probetan zehar lortu dugun eraginkortasuna 0.63-koa izan da. Kontuan izanda faktore askok eragin dezaketela denboren oszilazioak, esaterako kaskoaren posizioa, sentsoreen kalitatea, sortutako estimuluen zuzentasuna, ingurunearen argitasuna, algoritmoaren eraginkortasuna, edota erabiltzailearen erantzute-denbora, lortu diren emaitzak ez dira bat ere txarrak. Azkenik, egindako proba bat hurrengo estekako bideoan eskuragarri dago:

<https://www.youtube.com/watch?v=ky07jOAWFTA&feature=youtu.be>

Lagungarri moduan, bideotik ateratako ondorengo irudian, markaturik ikusten diren atalen azalpen bat egingo da.



**7.3 Irudia:** Bideorako lagungarria.

1. Kaskoa zuzen kokatuta dugula bermatu behar dugu proba kasua hasi baino lehen.
2. Raspberry Pi-a beharrezko periferikoetan konektatuta dugula, LED-aren zirkuitua muntatuko beharko dugu plaka batean. Ikusten den argi gorri handia da LED-ak sortutakoa.
3. Beste ekipo batean estimuluak aktibaturik izango ditugu. Eraginkorragoa izango da ingurune ilun batean gauzatzen bada esperimientua.
4. Raspberry Pi-an exekuzioa egitean, erabiltzaileak jasoko duen *feedback*-a edukiko dugu Raspberry Pi-ra konektatu dugun monitorean.

### Ondorioak

---

#### 8.1 Sarrera

Azken kapitulu honetan, proiektuan zehar izandako zailtasun garrantzitsuenak azalduko dira eta amaitzeko, proiektuaren amaierarekin pertsonalki ateratako ondorioak eta ikuspuntuak azalduko ditut.

#### 8.2 Zailtasunak

Proiektuaren garapenean zehar, asko izan dira bidean suertatu diren zailtasunak, eta hauek konpontzea ez da lan erraza izan.

Hasteko, kaskoarekin trebatzeko hainbat saiakera behar izan dira, bai kokapen egokia lortzeko baita elektrodoen kalitate ona lortzeko ere. Azkenean, proba askoren ondoren, errazagoa egin zitzaidan kaskoa ongi kokatzea eta kalitate ona bermatzea.

Bestalde, aurrera ateratzeko denbora gehien behar izan duten zailtasuna, kaskotik datu eta pakete egokiak lortzea izan da. Lehen esan bezala, erabilitako kaskoaren bertsioa berria izanik, uste nuen datu paketeen egitura bertsio zaharraren berdina izango zela, baina ez zen horrela. Hainbat proba egin ondoren konturatu nintzen jasotako seinaleak ez zuela inongo zentzurik eta beraz, bertsio berriaren inguruko informazioa bilatzen hasi nintzen. Halaber, interneten oso urria da bertsio berri honi buruz aurki dezakegun informazioa. Azkenik, ikertzaile batekin hizketan egon ondoren eta foro zein *blog* askotan informazioa bilatu ondoren, bertsio berriaren datu-paketeen egitura zein den jakitea lortu nuen.

Azkenik, behin seinaleen datuak lortzeko metodo egokia erabiliz, zailtasun berriak sistemaren proba kasutan suertatu ziren. Hasieran, SSVEP detekzio egokia lortu ahal izateko 2 frekuentzia esanguratsu lortu behar ziren. 10Hz-ko frekuentzia oso nabaria zenez hau erabiltzea erabaki zen. Halaber, beste gainontzeko frekuentzietan, detekzioa ez zen hain erraza eta esanguratsua. Proba kasu asko egin ondoren eta hainbat aldaketa egin ondoren, 10 eta 12 frekuentziak erabiltzea eraginkorrena zela erabaki zen. Bestalde, estimuluei dagokienez, hasieran koloretako karratuak erabiltzen hasi nintzen baina hauek zuriak izanda, ohartarazi nintzen errazagoa zela SSVEP gertaera detektatzea.

Zailtasun guzti hauen ondorioz, espero zena baino denbora gehiago pasa behar izan da hauek konpontzeko, baina behin oztopo guztiak gaindituta, sistema egonkor eta eraginkor bat sortzea lortu da.



## 8.3 Proiektuaren ondorioak

Hasteko, proiektuaren helburuei dagokionez, hasiera batean planteatutako guztiak bete direla esan beharra dago. Halaber, denboren estimazioetan desbiderapen batzuk egon dira lehen aipatutako zailtasun guztien ondorioz batik bat.

Proiektuaren gaiari buruz, esan beharra dut oso interesgarria iruditu zaidala. Ikerketa azterketa asko egin ondoren, biomedikuntzaren munduan eta batez ere EEG seinaleen inguruan dauden proiektu eta ikerkuntza ugari daudela ikusi dut, eta oso gai interesgarriak jorratzen dituzte. Gainera, Emotiv Epop+ kaskoa nahiko berria izanik, nik egin behar izan dudana azterketa eta proiektua "aitzindaria" dela esan daiteke, kaskotik jasotako datuen egiturari dagokionez.

Bestalde, aurreko ataletan aipatu bezala, BCI sistemarekin lortu den eraginkortasuna %60-koa izan da gutxi gora behera. Eraitza hau halaber, egokia bezala interpreta genezake, kontuan izanda zenbat faktorek eragin dezaketen eraitza eta proba kasuen aldakuntzak.

Azkenik, proiektu eta ikerketa hau oinarri bezala hartuz, hainbat aplikazio eta hobekuntza egin litezke. Hurrengo atalean azalduko dira hobekuntza posible batzuk.

## 8.4 Hobekuntza posibleak

Etorkizun batean, proiektu hau oinarri bezala hartuz, hainbat aplikazio edota hobekuntza egin daitezke. Adibidez, nire kasuan soilik bi estimulu erabiltzen dira eta beraz, bi ekintza soilik gauza daitezke, argia piztu edo itzali kasu honetan. Halaber, estimulu gehiago erabiliz, frekuentzia desberdinetakoak, beste hainbeste ekintza burutu ahalko genuke, esaterako hainbat LED erabili, joko bateko pertsonaia bat kontrolatu, musika erreproduzio aplikazio bat kontrolatu etab.

Bestalde, proiektu honen hobekuntza posible batzuk ere bururatzen zaizkit, lortutako eraginkortasuna hobetzea dutenak helburu. Adibidez, segundo bateko laginketak egin ordez, segundo erdikoak edo gutxiagokoak egin litezke. Bestalde, frekuentzia bateko estimulua begiratu dela erabakitzeko orduan, aurreko kapituluan zehaztu dugu 3 lagin jarraian ageri beharko litezkeela frekuentzia horri dagokionez. Ezaugarri hau murrizt genezake ondoz ondoko 2 laginetara, baina hau arriskutsua izan daiteke faltsu positiboak agertzeko probabilitatea handitzen delako.



## Bibliografia

---

- [1] Elektroentzefalografiaren informazioa: "Electroencefalografía(EEG)"  
<https://hardzone.es/2018/07/02/review-raspberry-pi-3-modelo-b/>
- [2] 10-20 sistemaren informazioa: Homan, Richard W., John Herman, and Phillip Purdy. "Cerebral location of international 10–20 system electrode placement." *Electroencephalography and clinical neurophysiology* 66.4 (1987): 376-382.
- [3] Fernando Lopes da Silva Donald L. Schomer. *Niedermeyer's Electroencephalography: Basic Principles, Clinical Applications, and Related Fields*. LWW, Sixth edition, 2010.
- [4] Emotiv Epoc+ buruzko informazioa:  
<https://www.emotiv.com/product/emotiv-epoc-14-channel-mobile-eeeg/#tab-description>
- [5] Raspberry Pi 3 B+ buruzko informazioa:  
<https://hardzone.es/2018/07/02/review-raspberry-pi-3-modelo-b/>
- [6] BCI sarrerako informazioa: [https://webdiis.unizar.es/~jminguez/Sesion001\\_UJI.pdf](https://webdiis.unizar.es/~jminguez/Sesion001_UJI.pdf)
- [7]  
[http://www.schalklab.org/sites/default/files/misc/BCI2000%20-%20A%20General-Purpose%20Brain-Computer%20Interface%20\(BCI\)%20System.pdf](http://www.schalklab.org/sites/default/files/misc/BCI2000%20-%20A%20General-Purpose%20Brain-Computer%20Interface%20(BCI)%20System.pdf)
- [8] SSVEP detekziorako frekuentzia banda:  
<https://journals.plos.org/plosone/article?id=10.1371/journal.pone.0077536>
- [9] SSVEP azalpenak:  
<https://journals.plos.org/plosone/article?id=10.1371/journal.pone.0159988#sec002>
- [10] ERP detekziorako sentsore egokiak:  
<https://pdfs.semanticscholar.org/24d9/4568b1925cfbe98bb5c88c1193bb91806247.pdf>

## Erabilpen gida

---

Atal honetan, sistema osoa martxan jartzeko eta probatu ahal izateko eman beharreko pausuak azalduko dira.

1. Lehen pausoa kaskoaren sentsoreak ongi hidratatzea da. Haien kaxan daudela, kaskoak ekartzen duen likidoko pare bat tanta bota sentsore bakoitzari. Geroz eta bustiagoak egon, oraindik eta kalitatea hobea lortuko duzu.



**E1 Irudia:** Sentsoreak hidratatu.

2. Behin sentsoreak hidratatuta daudela, kaskoaren elektrodoen tokitan ipini.



**E2 Irudia:** Sentsoreak kaskoak kokatu.

3. Ondoren buruan ipini eta kalitate egokia lortu duzula bermatzeko, "EMOTIV app" aplikazioa erabili Windows edo iOS ekipo batean. Sistemak erabiliko dituen elektrodoak alde okzipitaleko hiru direnez, hauen kalitatea ona izatea gomendagarria da, besteenak ez dute garrantzi handirik.



**E3 Irudia:** Kaskoa zuzen kokatu eta kalitate ona bermatu..

4. Kalitate egokia dagoenean, ingurunea prestatzen has gaitzke. Lehenik, LED-aren zirkuitua muntatuko dugu. 5.3 atalean azaldu den bezala, zirkuitua oso sinplea da eta hori egindakoan, probekin has gaitzke.



**E4 Irudia:** Raspberry Pi-a periferikoetara konektatu eta LED zirkuitua muntatu .

5. Raspberry Pi-a beharrezko periferikoetara konektatu eta terminal batetik gure aplikazioaren karpetan kokatu.
6. Beste ekipo batean, estimuluak prest edukiko ditugu, exekuzioa hasten denerako. Eraginkortasuna hobetzeko asmoz, toki ilun batean gauzatzea gomendatzen da. Hurrengo estekan daude eskura estimuluak:  
<https://estimuluak.000webhostapp.com/estimuluakWeb/Aurkezpena.html>
7. Programa exekutatzean, hasierako 10 segundoak kalibrazioa egiteko erabiltzen da. Segundo hauek oso garrantzitsuak dira eta beraz, guztiz geldirik eta puntu finko batera begiratzea izango litzateke hoberena.
8. Kalibrazioa amaituta, estimuluak begiratzten has zaitezke eta zirkuituan muntatutako LED-a piztu eta itzali zenezake.

---

## Bilera aktak

---

### Konstituzio bilera

**Data:** 2019-02-18 15:30.

**Bilerako partaideak:** Andoni Arruti, Javier Muguerza eta Iñigo Erkizia.

**Helburuak:** Proiektuaren nondik norakoak azaldu eta GAUR-en proiektua esleitu.

**Egin beharrekoak:**

- Gaiaren inguruko materiala aztertu.
- Teoria irakurtzen hasi ideia bat izateko.

### Lehenengo bilera

**Data:** 2019-03-11 14:00.

**Bilerako partaideak:** Andoni Arruti, Javier Muguerza eta Iñigo Erkizia.

**Helburuak:** Fakultateko laborategi batean lanpostua ezarri. Ekiporako erabiltzaileak sortu eta laborategira sartzeko baimena lortu.

**Egin beharrekoak:**

- Raspberry Pi-a martxan jarri beharrezko periferikoekin.
- Marko teorikoa lantzen jarraitu.
- GordeTalde plataformara sarbidea fakultateko ekipoen ahalbidetu dokumentuak partekatzeko.
- Memorian proiektuaren helburuen kapitulua dokumentatzen hasi.

### Bigarren bilera

**Data:** 2019-04-12 12:30.

**Bilerako partaideak:** Andoni Arruti, Javier Muguerza, Iñigo Erkizia eta Axel Garcia.

**Helburuak:** Proiektuaren jarraipena egin eta dudak argitu.

**Egin beharrekoak:**

- Emotiv Epoc+ kaskoa konektatzen saiatu *Bluetooth* edo USB bidez.
- Memorian, proiektuan erabilitako teknologien kapitulua dokumentatu.

## Hirugarren bilera

**Data:** 2019-04-29 15:30.

**Bilerako partaideak:** Andoni Arruti, Javier Muguerza, Iñigo Erkizia eta Axel Garcia.

**Helburuak:** Proiektuaren jarraipena egin eta zalantzak argitu.

**Egin beharrekoak:**

- Emotiv Epoc+ kaskotik jasotako seinale kalitate ona eta egonkorra lortzea.
- Sistemaren implementazioa lantzen hasi.
- Memorian, sistemaren inguruko marko teorikoa dokumentatu.

## Laugarren bilera

**Data:** 2019-05-06 15:30.

**Bilerako partaideak:** Andoni Arruti, Javier Muguerza eta Iñigo Erkizia.

**Helburuak:** Proiektuaren jarraipena egin eta zalantzak argitu.

**Egin beharrekoak:**

- Programan leiho labainkorra implementatzea seinalearen eskuratzeko prozesuan.
- Seinaleari zein filtro ezarri behar diogun aztertzea.
- Memorian, sistemaren metodologia dokumentatzea.

## Bosgarren bilera

**Data:** 2019-05-13 15:30.

**Bilerako partaideak:** Andoni Arruti, Javier Muguerza, Axel Garcia eta Iñigo Erkizia.

**Helburuak:** Proiektuaren jarraipena egin eta zalantzak argitu.

**Egin beharrekoak:**

- FFT aplikatu leiho bakoitzari.
- Filtroren bat aplikatu.
- Ikusi jasotako seinaleak zentzua duen.

## Seigarren bilera

**Data:** 2019-05-20 15:30.

**Bilerako partaideak:** Andoni Arruti, Axel Garcia eta Iñigo Erkizia.

**Helburuak:** Proiektuaren jarraipena egin eta zalantzak argitu.

**Egin beharrekoak:**

- Arazoaren agerpena: erabilitako algoritmoak ez ditu Epoc+ kaskoaren ezaugarriak errespetatzen; bereizmena, gakoaren lorpena, paketeen egitura...

## Zazpigarren bilera

**Data:** 2019-05-27 15:30.

**Bilerako partaideak:** Andoni Arruti, Javier Muguerza, Axel Garcia eta Iñigo Erkizia.

**Helburuak:** Proiektuaren jarraipena egin eta zalantzak argitu.

**Egin beharrekoak:**

- Arazoa konpontzen jarraitu, proiektu gehiago bilatu interneten ea arazoa konpontzeko laguntza dagoen.

## Zortzigarren bilera

**Data:** 2019-06-3 11:30.

**Bilerako partaideak:** Andoni Arruti, Javier Muguerza, Axel Garcia eta Iñigo Erkizia.

**Helburuak:** Proiektuaren jarraipena egin eta zalantzak argitu.

**Egin beharrekoak:**

- CyKit proiektu batekin probak egin, Epoc+ bertsioaren ezaugarriak kontuan hartzen direlako.
- Saiatu CyKit proiektuko beharrezko kodea nire algoritmoan ezartzen.

## Bederatzigarren bilera

**Data:** 2019-06-10 11:30.

**Bilerako partaideak:** Andoni Arruti, Javier Muguerza, Axel Garcia eta Iñigo Erkizia.

**Helburuak:** Proiektuaren jarraipena egin eta zalantzak argitu.

**Egin beharrekoak:**

- Arazoa konponduta, estimuluekin probak egin.
- Estimuluen interfazeaz aldaketak egin SSVEP gertaera nabaria izan dadin.

## Hamargarren bilera

**Data:** 2019-06-17 12:00.

**Bilerako partaideak:** Andoni Arruti, Javier Muguerza eta Iñigo Erkizia.



**Helburuak:** Proiektuaren jarraipena egin eta zalantzak argitu.

**Egin beharrekoak:**

- Overlapping aplikatu datu irakurketan.
- Unbrala kalkulatzeko kalibrazioa egin.
- Probak egin.

## Hamaikagarren bilera

**Data:** 2019-06-24 11:30.

**Bilerako partaideak:** Andoni Arruti, Javier Muguerza eta Iñigo Erkizia.

**Helburuak:** Proiektuaren jarraipena egin eta zalantzak argitu.

**Egin beharrekoak:**

- Probak egin eta emaitzak atera

## Itxiera bilera

**Data:** 2019-07-25 11:00.

**Bilerako partaideak:** Andoni Arruti, Javier Muguerza eta Iñigo Erkizia.

**Helburuak:** Proiektuaren amaierako xehetasunak zehaztu eta udararako lanak planteatu.

**Egin beharrekoak:**

- Udan memoria amaitu.