

GRADUA: Kudeaketaren eta Informazio Sistemen  
Informatikaren Ingeniaritzako Gradua

## GRADU AMAIERAKO LANA

**@IKASTENBOT, GRAL BAT GARATZEKO  
LAGUNTZAILEA**

**Ikaslea:** de Pablo, Ruiz, Amaia

**Zuzendaria (1):** Pereira, Varela, Juan Antonio

**Zuzendaria (2):** Armendariz, Leuna, Ana Jesus

**Ikasturtea:** 2017-2018

**Data:** Bilbo, 2018, 07, 18



## Laburpena

Proiektu honen helburua ikasleei Gradu amaierako lan (GRAL) baten garapean zehar laguntza eskainiko dien bot bat garatzea da. Telegram bat-bateko mezularitzarako aplikaziorako bot bat izango da.

Bot-ak ikasleek izan ditzaketen galderak argitzen lagunduko du, deialdien datei buruzko abisuak emango ditu eta GRALen memoriak aztertuko ditu ortografia eta gramatika erroreak bilatzeko eta plagioa dagoen aztertzeko.

Beste alde batetik, administrazio panel bat garatu da bot-aren informazioa kudeatzeko. Panel honen bitartez, deialdien datak eguneratu, abisuak kudeatu eta galderak gehitu edota aldatu ahalko dira.

Bot-a PHPn garatutako php-telegram-bot liburutegia erabiliz garatu da eta administrazio panela Python-en idatzitako Django framework-a erabiliz. Bi tresnek MySQL datu-base bera erabiltzen dute.

**Hitz gakoak:** bot, Telegram, GRAL laguntzailea, Gradu Amaierako Lana, PHP, administrazio panela, Django



## Resumen

El objetivo de este proyecto es desarrollar un bot para asistir a los estudiantes durante el desarrollo de su Trabajo Fin de Grado (TFG). Este bot estará disponible en la aplicación de mensajería Telegram.

El bot responderá a las posibles preguntas que puedan tener los alumnos, enviará avisos de los plazos de las convocatorias y analizará las memorias de los TFG para buscar errores ortográficos y gramaticales y analizar posibles plagios.

Por otro lado, se ha desarrollado un panel de administración para gestionar la información del bot. A través del panel se podrá actualizar las fechas de las convocatorias, gestionar los avisos y añadir o modificar las preguntas.

El bot se ha desarrollado utilizando la librería php-telegram-bot en PHP y el panel de administración utilizando el framework Django escrito en Python. Las dos herramientas comparten una base de datos MySQL.

**Palabras clave:** bot, Telegram, asistente TFG, Trabajo Fin de Grado, PHP, panel administración, Django



## Abstract

The objective of this project is to develop a bot to assist students during the development of their Final Degree Project (FDP). This bot will be available in the Telegram messaging application.

The bot will answer the questions that the students may have, send notifications of the deadlines of the calls and analyze the report of the FDP to find spelling and grammar errors and analyze possible plagiarism.

Besides, an administration panel has been developed to manage bot information. Through the panel you can update the dates of the calls, manage the notifications and add or modify the questions.

The bot has been developed using the php-telegram-bot library in PHP and the administration panel using the Django framework written in Python. Both tools share a MySQL database.

**Keywords:** bot, Telegram, FDP assistant, Final Degree Project, PHP, administration panel, Django





# Gaien Aurkibidea

<b>1</b>	<b>Terminoak</b>	<b>15</b>
<b>2</b>	<b>Sarrera</b>	<b>17</b>
2.1	Proiektuaren jatorria eta deskribapena . . . . .	17
2.2	Proiektuaren aukeraketaren zergaitia . . . . .	18
<b>3</b>	<b>Hasierako planteamendua</b>	<b>19</b>
3.1	Helburuak . . . . .	19
3.2	Erabilitako tresnak . . . . .	20
3.3	Norainokoak . . . . .	22
3.3.1	Kudeaketa . . . . .	24
3.3.2	Ikasketa . . . . .	25
3.3.3	Analisia eta Diseinua . . . . .	26
3.3.4	Inplementazioa . . . . .	26
3.3.5	Testak . . . . .	28
3.3.6	Dokumentazioa . . . . .	30
3.4	Denbora planifikazioa . . . . .	31
3.5	Analisi ekonomikoa . . . . .	34
3.6	Arriskuen analisia . . . . .	36
<b>4</b>	<b>Betekizunen bilketa</b>	<b>39</b>
<b>5</b>	<b>Tresnen hautapena</b>	<b>41</b>
5.1	Programazio lengoaiak . . . . .	41
5.2	Beste software eta liburutegiak . . . . .	41
<b>6</b>	<b>Analisi eta diseinua</b>	<b>43</b>
<b>7</b>	<b>Arkitektura</b>	<b>47</b>
7.1	Telegram bot . . . . .	47
7.2	Administrazio panela . . . . .	48
<b>8</b>	<b>Garapena</b>	<b>51</b>
8.1	Prototipoen garapena . . . . .	52
8.1.1	1. prototipoa . . . . .	52
8.1.2	2. prototipoa . . . . .	55
8.1.3	3. prototipoa . . . . .	56
8.1.4	4. prototipoa . . . . .	59
8.1.5	5. prototipoa . . . . .	60
8.2	Administrazio panela . . . . .	60
8.3	Izandako arazo batzuk . . . . .	65
<b>9</b>	<b>Testak</b>	<b>69</b>
9.1	1. prototipoa . . . . .	69
9.2	2. prototipoa . . . . .	70

9.3	3. prototipoa . . . . .	71
9.4	5. prototipoa . . . . .	72
9.5	Administrazio panela . . . . .	73
<b>10</b>	<b>Ondorioak</b>	<b>75</b>
10.1	Helburuak . . . . .	75
10.2	Denboraren planifikazioa . . . . .	75
10.3	Etorkizunerako lana . . . . .	75
10.4	Hausnarketa pertsonala . . . . .	77
<b>A</b>	<b>I. Eranskina: Instalazio gida</b>	<b>81</b>
A.1	Aurrebaldintzak . . . . .	81
A.2	Bot-aren instalazioa . . . . .	81
A.3	Datu-basea . . . . .	83
A.4	Administrazio panela . . . . .	83
<b>B</b>	<b>II. Eranskina: Bot-aren erabiltzaile gida</b>	<b>85</b>
<b>C</b>	<b>III. Eranskina: Administrazio panelaren erabiltzaile gida</b>	<b>89</b>

## Irudien Zerrenda

1	Proiektuaren LDE diagrama, bloke nagusiekin eta hauen deskonposaketarekin. . . . .	23
2	Proiektuaren Gantt diagrama . . . . .	33
3	Bot-a DBaren diseinua. . . . .	45
4	Aplikazioaren arkitektura. . . . .	49
5	<i>KeyboardButton</i> motako hiru botoi bata bestearen gainean, mugikorraren teklatuaren tokian. . . . .	52
6	<i>InlineKeyboradButton</i> motako botoiak matrize itxuran kokatuta, bot-aren mezuen azpian. . . . .	52
7	/faq komandoaren lehen inplementazioa, galdera bakoitza komando batean. . . . .	53
8	/faq komandoaren bigarren inplementazioa, elkarrizketa erabiliz. . . . .	54
9	/faq komandoaren azken inplementazioa, <i>InlineKeyboardButton</i> ak erabiliz. . . . .	55
10	GRAL baten erregistroa eta bertsio berri baten bidalketa . . . . .	56
11	Bot-ak GRALaren zuzenketa erakusten du, non erroren orrialde ezberdinak ikus daitezkeen mezu berean. . . . .	59
12	/imagesTFG komandoa exekutatzeko jasotzen den erantzun baten adibidea, non entitatea eta url posible batzuk agertzen diren. . . . .	61
13	Administrazio paneleko pantaila nagusia, non kudeatu daitezkeen ereduak agertzen dira . . . . .	63
14	DBan FAQ galderen taulan agertzen diren elementuen zerrenda. . . . .	64
15	DBan FAQ galdera baten informazio eta honek izan ditzakeen erantzunen informazioa . . . . .	65
16	Galdera baten testua aldatzeko bi leiho, ezkerrean galdera eta eskuinean testua . . . . .	66
17	GRALaren zuzenketa, erroreak gaizki markatuta . . . . .	68
18	/faq komandoaren adibidea. . . . .	85
19	GRAL baten erregistroa eta bertsio berri baten bidalketa . . . . .	86
20	GRAL baten erroreak ikusteko pantaila . . . . .	87
21	Interneten aurkitu den irudia . . . . .	88
22	Administrazio paneleko login pantaila . . . . .	89
23	Administrazio paneleko pantaila nagusia . . . . .	90
24	FAQ galderen zerrenda . . . . .	91
25	FAQ galdera bat gehitu/aldatu . . . . .	92
26	Deialdi baten data bat gehitu/aldatu . . . . .	93
27	Testu bat gehitu/aldatu . . . . .	93
28	Abisu berezi bat gehitu/aldatu . . . . .	94
29	Abisu bat gehitu/aldatu . . . . .	94



## Taulen Zerrenda

1	Jardueren denbora estimazioa . . . . .	32
2	Proiektuaren kostuak . . . . .	36
3	Jardueren denbora estimazioa eta errealak . . . . .	76



# 1 Terminoak

Proiektuaren azalpenekin hasi baino lehenago, dokumentu honetan azalduko diren termino eta sigla batzuen esanahia azalduko da. Hona hemen definizio batzuk:

- **API:** ingelesez *Application Programming Interface*, hau da, aplikazioak programatzeko interfazea. Software funtzio edo metodoen multzoak dira, beste software batek liburutegi bezala erabiltzeko.
- **Chatbot/bot:** elkarrizketa bat jarraitzeko gai den programa informatikoa. Proiektuan zehar bot bezala azalduko da.
- **DB:** Datu-basea. Proiektuan datu-basea bere siglaz aipatuko da.
- **FAQ:** *Frequently Asked Questions*-en ingelesezko akronimoa da, euskaraz maiz egindako galderak esan nahi du.
- **Framework:** Software tresna eta moduluak eskaintzen dituen programazio-ingurunea, normalean aplikazioak sortzeko erabiltzen dena
- **GRAL:** Gradu Amaierako Lana. Gradu bat amaitzeko derrigorrezko proiektua, TFG (Trabajo Fin de Grado) gaztelaniaz. GRAL siglaz gain beste batzuk edota beste formatu batean erabiltzen dira: GAL, GrAL.
- **Hash funtzioa:** hash funtzio bat erabiliz probabilitatearen bidez sarre-rako datu bat identifikatzeko String bat sortzeko erabiltzen da. Adibidez, MD5 eta SHA-2 Hash funtzioaren algoritmo mota desberdinak dira.
- **JSON:** *JavaScript Object Notation*en laburdura, datuen elkartrukerako formatu arina da. Testu hutsezko formatu hau datu egitura sinpleak eta array asoziatiboak errepresentatzeko erabiltzen da.
- **Komando:** Aurrez egituratutako eragiketa edo ekintza bati hasiera ematen dion agindu edo informazio berezia. Telegram bot-ekin hitz egiteko komando bitartez egiten da. Telegramen komando bat `/-z` hasten da, adibidez `"/start"`.
- **LDE:** Lan deskonposaketaren egitura. proiektu baten ataza edo zereginen banaketa modu hierarkikoan egitean lortzen den diagrama da.
- **Script:** Programatutako kode bat da, zeinek modu sekuentzialen exekutatzen diren komando edo agindu ezberdinak ditu.
- **URL:** *Uniform Resource Locator* formatu estandar bat betetzen duen karaktere katea da, Interneten dokumentuak eta irudiak bezalako baliabideak izendatzeko balio duena.





## 2 Sarrera

Chatbot bat gizaki batekin elkarrizketa bat simulatzen duen software aplikazioa da. Gaur egun bat-bateko mezularitzarako aplikazioetan asko erabiltzen dira, adibidez, Facebook Messenger, Telegram edo Skype. Bot batek izen ditzakeen funtzionalitateen erabilgarritasuna oso handia izan daiteke eta esparru ezberdin ugarian erabili daitezke, adibidez, euskarri teknikoak, hezkuntza, turismoa, kirolak, medikuntzako aholkaritza, jolasteko...

@ikastenbot bot-a Euskal Herriko Unibertsitateko ikasleen laguntzailea izango da, ikasleei aholkuak emango eta ikasleak izan ditzakeen arazo eta galderei erantzun dituen.

GRAL honen helburua @ikastenbot bot-aren lehen bertsio bat egitea izango da. Lehen bertsio honetan bot-ak Kudeaketaren eta Informazio Sistemen Informatikaren Ingeniaritzako Graduko ikasleei beraien GRALak garatzen lagunduko die. @ikastenbot ikasleek izan ditzaketen zalantzak argituko ditu, GRALaren garapenean zehar dauden daten abisuak emango ditu eta GRALaren memoria aztertuko du erroreak bilatzeko

### 2.1 Proiektuaren jatorria eta deskribapena

Proiektu hau *egelapi*<sup>1</sup> plataformaren bitartez ezagutu nuen. Bertan Kudeaketaren eta Informazio Sistemen Informatikaren Ingeniaritzako Graduko irakasleek GRAL lanen proposamenak argitaratzen dituzte. Proiektu hau, Juanan Pereira irakasleak argitaratutako proposamena izan zen, bere ikerkuntza taldeko ikerketa lerro baten parte izanik.

Unibertsitateko gradu bat amaitzean nahitaezkoa da gradu amaierako lan bat (GRAL) egitea. Kudeaketaren eta Informazio Sistemen Informatikaren Ingeniaritzako Graduan azken urteetan egindako GRALetan ikusi da ikasleek zalantza eta errore berdinak izaten dituztela. Zalantzei dagokienez batzuk GRALaren matrikula eta defentsa kudeaketarekin lotura dute, beste batzuk memoriaren idazketarekin. Komunak diren errore batzuk hauek dira: GRALen memoriak ez dituzte izan beharreko atalak, irudi eta taulen azalpen eta aipamen falta, erabilitako testua erreferentziarik gabe plagio bihurtuz...

@ikastenbot Telegram bat-bateko mezularitzarako aplikaziorako bot bat izango da. Honen bitartez aipatutako erroreak konpondu nahi dira, kalitate handiagoko GRALak lortzeko.

GRAL honetan @ikastenbot bot-aren lehen bertsioa garatuko da. Implementatuko diren funtzionalitateak GRALen memoriak aztertzeke hauek dira: zuzentzaile ortografikoa eta testu eta irudien plagioaren azterketa. Gainera, ikasleen

---

<sup>1</sup><https://egelapi.ehu.eus/?lang=es>

laguntzailea izango da GRALaren garapenean zehar, izan ditzaketen zalantzak argituz edota abisuak emanez, deialdien daten berri emateko adibidez.

## 2.2 Proiektuaren aukeraketaren zergaitia

Gradu amaierako lan hau aukeratzearen arrazoiak honakoak izan ziren:

### Telegrameko Bot-ak

Proiektuaren berri izan nuenean, Telegrameko bot-ak maiz erabiltzen nituen, jolasteko, abisuak jasotzeko edota inkestak egiteko, eta hauen funtzionamendua hobeto ezagutzeko aukera egokia zela pentsatu nuen. Gainera, hezkuntza arloko bot-ik ez nuen ezagutzen eta GRALak garatzen lagunduko zuen bot bat oso erabilgarria izango zela kontuan izanda, proiektuaren ideia asko gustatu zitzaidan.

### Teknologia berriak

Proiektuan erabiliko ziren teknologia gehienak ez nituen ezagutzen, PHP, Telegrameko APIa eta funtzionalitate ezberdinetarako erabili diren gainontzeko API eta liburutegiak. Teknologia berriak ikasteko eta hauekin lan egiteko aukera aproposa zela pentsatu nuen.

### Ikasleei laguntzea

Bot-aren helburuetako bat ikasleei GRALaren garapenean zehar laguntzea da, izan ditzaketen zalantzak argitzeko. Nik ere zalantza asko nituen, eta ezagutzen ez nituen gauza ugari zeuden GRALari buruz. Bot-aren garapenaren bitartez nire moduan dauden beste ikasleei laguntza eskaintzea aukera egokia iruditu zitzaidan.

## 3 Hasierako planteamendua

Atal honetan proiektuaren helburuak azalduko dira. Horrez gain, proiektuaren norainokoak, arriskuak, denboraren planifikazioa, analisi ekonomikoa eta erabilitako tresnak azalduko dira.

### 3.1 Helburuak

Proiektu honen helburu nagusia da ikasleei GRAL baten garapenean zehar laguntza eskaintzen duen bot bat garatzea. Helburu honetaz aparte beste helburu batzuk ere daude proiektu honetan. Jarraian helburu horiek azalduko dira atal ezberdinetan banaturik.

#### Telegram bot-a

Jarraian, Telegram-eko bot-a garatzean bete nahi diren helburuak ageri dira:

- **Tresnak bateratu:** GRAL baten garapenean, ikasle zein irakasleentzat, lagungarriak izan daitezkeen erremintak aplikazio bakar batean izatea da helburu nagusia.
- **Komando argiak eta erraz gogoratzekoak:** Bot-a erabiltzeko komandoak sinpleak izan behar dira. Hitz gakoak laburrak, gogoratzen errazak eta funtzionalitatearekin lotuta egon behar dira.
- **Erabilerraza:** Erabiltzaileak komando bat exekutatzeko duenean, beti izan behar du gertatzen ari denaren berri. Gainera, erabiltzaileak zerbait egin behar badu, botoiren bat sakatu edota fitxategi bat bidali, bot-ak azalpen bat bidali beharko dio.
- **Eguneratuta egotea:** Bot-ak bidaltzen duen informazioa eguneratuta egon behar da. Bot-ak deialdien berri eman behar duenean, informazio egokia bidali behar du eta ez pasatutako deialdien informazioa.

#### Orokorrak

Ondoren azalduko diren helburuak ez dira soilik bot-aren garapeneko helburuak, baizik eta proiektuaren garapen orokorrarekin lotuta dauden helburuak dira:

- **Modifikazio eta funtzionalitate berrietarako prestatua:** Proiektu hau proiektu handiago baten zatia da. GRAL honetan, ikasleei GRALak garatzen lagunduko dien bot baten funtzionalitate batzuk garatuko dira. Baina, GRALa amaitu ostean, norbaitek bot-aren garapenarekin jarraitzen badu behintzat, funtzionalitate gehiago gehitu zaizkio bot-ari.

Bestalde, bot-aren lehen bertsio hau Kudeaketaren eta Informazio Sistemen Informatikaren Ingeniaritzako Graduko ikasleei laguntza eskaintzeko bideratuta egongo da, baina aurrerantzean beste graduetakoko ikasleei laguntzea hedatzea da helburua. Hortaz, modifikazio eta funtzionalitate berrietarako prestatuta egon behar du.

- **Ikastea:** Helburu pertsonala da proiektu honen garapenean zehar ikastea. Normalean graduko ikasketetan zehar egindako proiektuak taldeka egiten dira eta ez dira oso handiak izaten. Beraz, proiektu honen garapenean ikasi nahi dena ez da soilik bot-a garatzeko behar diren teknologiak ikastea, baizik eta honelako proiektu bat bakarka egitean ikasi daitezkeen beste moduko gaitasunak ere, adibidez, planifikazioa egiten edota sor daitezkeen arazoak nola konpondu.

## 3.2 Erabilitako tresnak

Atal honetan proiektua garatzeko erabili diren tresnak azalduko dira. Proiekturen zein ataletan erabili diren ere aipatuko da.

- **Bitbucket:** Berstio kontrolerako Mercurial edo Git sistemak erabiltzen dituzten proiektuentzat ahokalekua eskaintzen duen zerbitzu bat da, zeinek errepositorio pribatuak doan eskaintzen dituen. Kodearen bertsio kontrola web-ean izateko erabiliko da.
- **Cacoo:** Diagramak online egiteko tresna da, bere erabilpena errazten duen interfaze sinple bat dauka. Memorian agertuko diren diagrama batzuk egiteko erabiliko da.
- **Cloud Vision API:** Argazkiak aztertzeke Googleek eskaintzen duen zerbitzua. Proiektu honetan, irudi batzuk interneten dauden jakiteko erabiliko da, baina beste erabilera batzuk ditu, adibidez argazki batean dauden elementuak identifikatzeko, argazki batetik testua ateratzeko edo pertsonen argazki batean beraien emozioak aztertzeke.
- **Composer:** PHP lengoaiarako liburutegi kudeatzailea da. Komando lerrotik exekutatu da eta aplikazio baten mendekotasunak kudeatzeko erabiltzen da.
- **Cron:** Aldizka exekutatu behar diren prozesuak kudeatzeko tresna. Prozesuak eta hauek noiz exekutatu behar diren crontab fitxategian zehaztu behar dira. Proiektu honetan bot-aren abisuak kudeatzeko erabiliko da.
- **Django:** Django Pythonen idatzita dagoen kode irekiko web framework bat da. Djangoren helburua, web orri konplexuen sorrera erraztea da. Proiektu honetan administrazio panel bat egiteko erabiliko da.
- **GanttProject:** Gantt Project proiektuaren plangintzak grafikoki irudikatzea ahalbidetzen du, ataza ezberdinak eransten dira hauen hasiera eta

bukaera datak eta hauen arteko aurrekotasun adieraziz, programak berak denboran zehar irudikatzen ditu grafikoki.

- **Git:** Bertsio kontrolerako software bat da. Bere helburua proiektu bateko fitxategietan egindako aldaketen kontrola eramatea eta partekatutako fitxategietan hainbat pertsonen egin dezaketen aldaketak koordinatzea da. Bitbucketen kodearen bertsio kontrola eramateko erabiliko da.
- **LanguageTool:** Doakoa eta kode irekiko gramatika eta ortografia zuzentzailea da. Gainera, hainbat aplikazioetarako plug-in bat eskaintzen du. GRALen memoriak zuzentzeko erabiliko da.
- **MySQL:** Datu-base erlazionalak kudeatzeko sistema bat da. Proiektuaren datu-basea sortzeko eta kudeatzeko erabiliko da.
- **MySQL Workbench:** MySQL motako datu-baseekin lan egiteko tresna bisuala da. DBak diseinatzeko, kudeatzeko eta mantentze lanak egiteko erabili daiteke. Proiektuaren DBa modu bisualean kudeatzeko erabiliko da.
- **Overleaf:** LaTeX-eko online editore bat da, zeinetan posiblea da aldi berean pertsona batek baino gehiagok proiektu berean lan egitea eta PDF formatura zuzenean konpilatu dezake. Proiektuaren memoria idazteko erabiliko da.
- **pdimages:** Komando lerrotik erabiltzeko kode irekiko tresna PDFetatik irudiak ateratzeko. GRALen memorietatik irudiak ateratzeko erabiliko da.
- **PHP:** interpretatutako programazio lengoiaia bat da, batez ere webgune dinamikoak sortzeko erabiltzen da. Bot-a inplementatzeko erabili den lengoiaia.
- **PhpStorm:** PHP lengoiaian programatzeko ingurunea. Bot-a garatzeko erabiliko da.
- **PowerPoint:** Diapositiba aurkezpenak egiteko software bat da. Defentsaren aurkezpena prestatzeko erabiliko da.
- **Python:** Interpretatutako programazio lengoiaia bat da, ardatz bezala sintaxi garbi, erraz eta irakurgarri bat duelarik. Administrazio panela egiteko erabiliko da.
- **PyCharm:** Python lengoiaian programatzeko ingurunea. Administrazio panela garatzeko erabiliko da.
- **spatie/pdf-to-text:** PDF batetik testua ateratzeko PHPko liburutegia. Memorien testuak ateratzeko erabiliko da.
- **Telegram:** Bat-bateko mezularitzarako aplikazioa da, mugikorreko, mahai-gaineko eta web-eko aplikazio du. Bot-a Telegram aplikazioan erabiliko da.

- **unoconv:** Komando lerrotik erabiltzeko tresna, LibreOfficek inportatu dezakeen edozein formatu LibreOfficek esportatu dezakeen edozein formatura bihurtzeko. doc, docx eta odt fitxategiak pdf bihurtzeko erabiliko da.
- **Visual Paradigm:** UML diagramak egiteko softwarea. Proiektuaren LDEa egiteko erabiliko da.

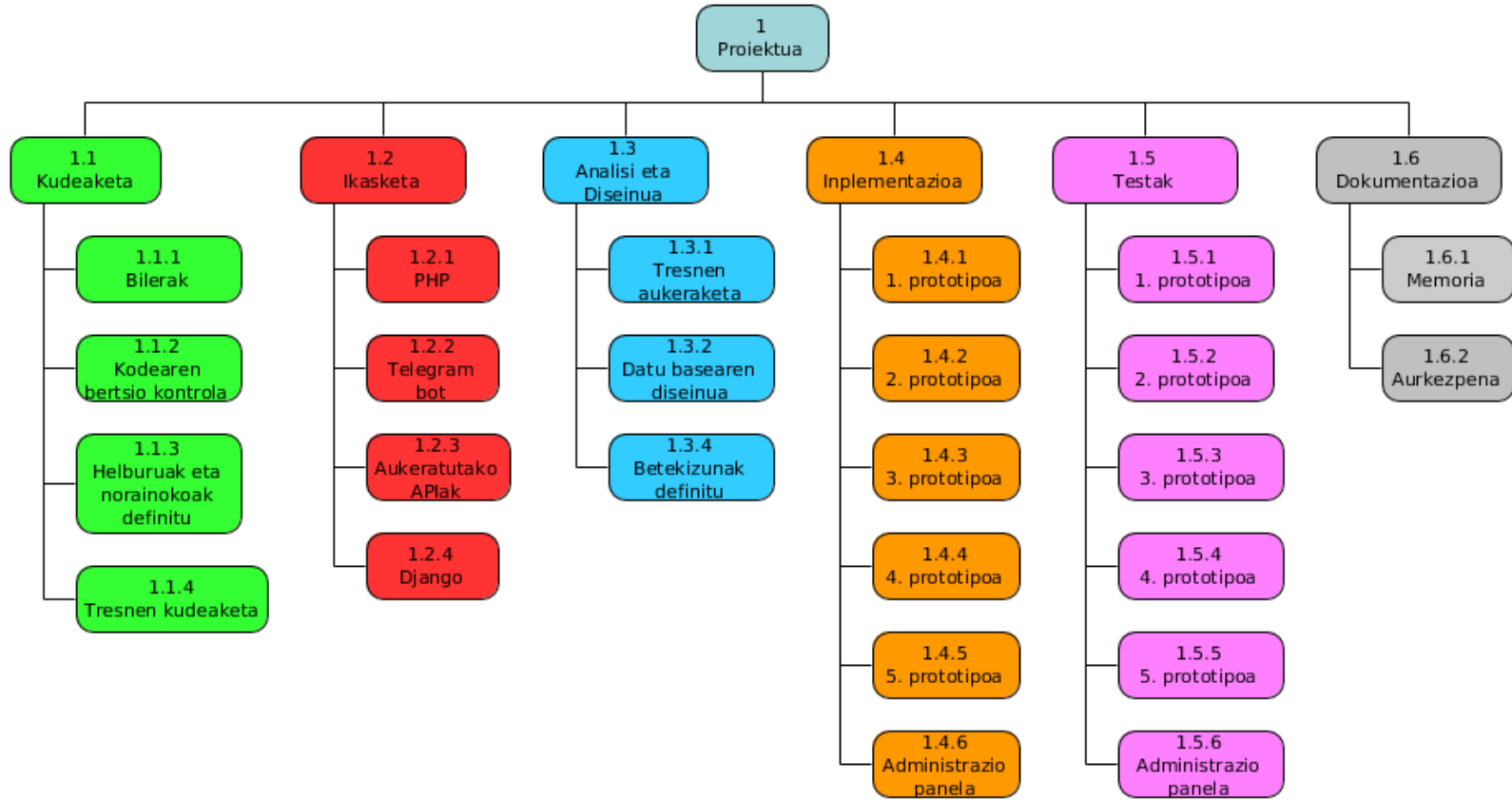
### 3.3 Norainokoak

Proiektuaren norainokoak azaltzeko 1. irudian agertzen den LDE diagrama egin da, proiektuaren bloke eta jarduerak adieraziz.

Bloke nagusiak honako hauek dira:

- **Kudeaketa:** Bloke hau proiektuaren segimenduaz ardurazten da.
- **Ikasketa:** Proiektuan zehar erabiliko diren tresnak nola erabiltzen diren ikasteko prozesuak biltzen dituen blokea da.
- **Analisisa eta diseinua:** Bloke honetan proiektuaren funtzionalitateak eta hauek nola egin adierazten da.
- **Inplementazioa:** Proiektuak izango dituen funtzionalitateak inplementatuko dira.
- **Testak:** Proiektuak izan behar dituen funtzionalitateak ondo dauden frogatuko da bloke honetan.
- **Dokumentazioa:** Proiektuari buruzko dokumentazioaren idazketa. Proiektu osoan zehar egingo da.

Proiektu hau prototipoka garatuko da, hau da, funtzionalitate bakoitza bakarrik balitz bezala tratatuko da. Hortaz, bloke batzuetako jarduerak ez dira behin bakarrik gertatuko, prototipo bakoitzean baizik; adibidez, ikasketa, analisisa eta diseinua. Jarraian dauden azpiataletan, blokeen banaketa sakonago azalduko da.



Irudia 1: Proiektuaren LDE diagrama, bloke nagusiekin eta hauen deskonposaketarekin.

### 3.3.1 Kudeaketa

Kudeaketa proiektuaren garapen guztia iraungo duen blokea da, hau baita proiektua nola egin zehazten duena eta honen jarraipena egiten duena dena ondo joateko. Bloke hau honako jardueratan banatzen da:

<i>Bilerak zuzendariarekin</i>
<b>Lan-blokea:</b> Kudeaketa
<b>Denbora estimazioa:</b> 20 ordu.
<b>Deskribapena:</b> Proiektuaren zuzendariarekin proiektuaren garapenean zehar izandako bilerak, zalantzak argitzeko eta lanaren jarraipena izateko.
<b>Irteerak/Emangarriak:</b> Bileran hartutako erabakiak dituen dokumentua.
<b>Baliabideak:</b> Papera eta boligrafoa.

<i>Kodearen bertsio kontrola</i>
<b>Lan-blokea:</b> Kudeaketa
<b>Denbora estimazioa:</b> 10 ordu.
<b>Deskribapena:</b> Bot-aren kodearen bertsio kontrola Bitbucket web orrian sortu eta kudeatu.
<b>Irteerak/Emangarriak:</b> Ez dago
<b>Baliabideak:</b> Git eta Bitbucket.

<i>Helburuak eta norainokoa definitu</i>
<b>Lan-blokea:</b> Kudeaketa
<b>Denbora estimazioa:</b> 2 ordu.
<b>Deskribapena:</b> Proiektuaren helburuak zehaztu eta hauek betetzeko beharrezkoak diren zereginak definitu.
<b>Irteerak/Emangarriak:</b> Helburu, funtzionalitate eta zereginen zerrenda.
<b>Baliabideak:</b> Papera eta boligrafoa

<i>Tresnen kudeaketa</i>
<b>Lan-blokea:</b> Kudeaketa
<b>Denbora estimazioa:</b> 10 ordu.
<b>Deskribapena:</b> Proiektua garatzeko beharrezkoa izango den softwarea instalatu eta eguneratuta mantendu.
<b>Irteerak/Emangarriak:</b> Software eguneratua.
<b>Baliabideak:</b> Ordenagailua eta internet konexioa.



### 3.3.2 Ikasketa

Bloke honek proiektua garatu ahal izateko beharrezkoak diren tresnak menperatzea du helburu. Tresna batzuen erabilera lehendik dira ezagunak; beraz, atal honetan ezezagunak diren tresnen ikasketa bakarrik ageri da. Bereziki PHP eta Telegram Bot-ak erabiltzeko liburutegiaren ikasketa da garrantzia handiena dutenak eta proiektuaren hasieran egingo direnak. Gainontzekoak jarduera bakar batean bildu dira eta prototipo bakoitzean beharrezkoak izango diren tresnak baino ez dira ikasiko.

<i>PHP ikasi</i>
<b>Lan-blokea:</b> Ikasketa
<b>Denbora estimazioa:</b> 10 ordu.
<b>Deskribapena:</b> PHP lengoaiaren sintaxia eta erabilera ikasi.
<b>Irteerak/Emangarriak:</b> Ez dago
<b>Baliabideak:</b> PHP, PhpStorm eta PHP eskuliburua.

<i>Telegram bot-en eta php-telegram-bot liburutegiaren erabilera ikasi</i>
<b>Lan-blokea:</b> Ikasketa
<b>Denbora estimazioa:</b> 20 ordu.
<b>Deskribapena:</b> php-telegram-bot liburutegiaren erabilera ikasi, honek eskaintzen dituen adibideak begiratzuz eta Telegram Bot APIaren dokumentazio irakurriz.
<b>Irteerak/Emangarriak:</b> Ez dago.
<b>Baliabideak:</b> PHP, PhpStorm, php-telegram-bot liburutegia eta Telegram Bot APIaren dokumentazioa[Telegram, 2018].

<i>Aukeratutako APIak eta liburutegiak erabiltzen ikasi</i>
<b>Lan-blokea:</b> Ikasketa
<b>Denbora estimazioa:</b> 10 ordu.
<b>Deskribapena:</b> Proiektua garatzeko beharrezkoak izango diren APIak (zuzentzeko, plagioak eta irudiak aztertzeko) eta zenbait liburutegi probatu eta dokumentazioa irakurri.
<b>Irteerak/Emangarriak:</b> Ez dago.
<b>Baliabideak:</b> PHP, PhpStorm, erabiliko dire APIen eta liburutegien dokumentazioa.

<i>Django</i>
<b>Lan-blokea:</b> Ikasketa
<b>Denbora estimazioa:</b> 15 ordu.
<b>Deskribapena:</b> Django erabiltzen ikasi administrazio panela egiteko.
<b>Irteerak/Emangarriak:</b> Ez dago.
<b>Baliabideak:</b> Python, PyCharm, Django liburutegia, MySQL Workbench, Djangoren gida[Azzopardi and Maxwell, 2014], ordenagailua.

### 3.3.3 Analisia eta Diseinua

Atal honetan analisi eta diseinuarekin lotuta dauden zereginak ageri dira. Prototipoka lan egingo denez, hauetako jarduera batzuk proiektu osoan zehar egingo dira.

<i>Tresnen aukeraketa</i>
<b>Lan-blokea:</b> Analisia eta Diseinua
<b>Denbora estimazioa:</b> 10 ordu.
<b>Deskribapena:</b> Proiektua garatzeko erabiliko diren programazio lengoaiak, softwarea eta APIak aukeratuko dira, aukera ezberdinak probatuz eta hobere-na hautatuz.
<b>Irteerak/Emangarriak:</b> Erabiliko diren tresnen zerrenda.
<b>Baliabideak:</b> Ordenagailua internet konexioarekin.

<i>Datu-basearen diseinua</i>
<b>Lan-blokea:</b> Analisia eta Diseinua
<b>Denbora estimazioa:</b> 5 ordu.
<b>Deskribapena:</b> Programak beharko duen datu-basearen diseinua egin.
<b>Irteerak/Emangarriak:</b> Datu-basearen diseinua.
<b>Baliabideak:</b> Papera eta boligrafoa.

<i>Betekizunak definitu</i>
<b>Lan-blokea:</b> Analisia eta Diseinua
<b>Denbora estimazioa:</b> 3 ordu.
<b>Deskribapena:</b> Proiektua garatzeko beharrezkoak izango diren funtzionalitateak definitu.
<b>Irteerak/Emangarriak:</b> Betekizunen zerrenda.
<b>Baliabideak:</b> Papera eta boligrafoa.

### 3.3.4 Inplementazioa

Inplementazioan proiektuaren atal nagusia da, non proiektuaren funtzionalitateak garatzeko beharrezkoa den kodea inplementatuko den. Proiektuaren imple-

mentazioa honako ataletan banatzen da:

<b>1. Prototipoa</b>
<b>Lan-blokea:</b> Inplementazioa
<b>Denbora estimazioa:</b> 25 ordu.
<b>Deskribapena:</b> FAQ erantzungo dituen komandoa inplementatu.
<b>Irteerak/Emangarriak:</b> FAQ komandoaren kodea eta hau exekutatzeko beharrezkoak diren datuak.
<b>Baliabideak:</b> PHP, PhpStorm, php-telegram-bot liburutegia, MySQL Workbench, Telegram eta ordenagailua eta mugikorra internet konexioarekin.

<b>2. Prototipoa</b>
<b>Lan-blokea:</b> Inplementazioa
<b>Denbora estimazioa:</b> 20 ordu.
<b>Deskribapena:</b> Bot-aren erabiltzaileei abisuak modu automatizatuan bidaltzea ahalbidetzen duen kodea inplementatu.
<b>Irteerak/Emangarriak:</b> Erabiltzaileei abisuak bidali ahal izatea.
<b>Baliabideak:</b> PHP, PhpStorm, php-telegram-bot liburutegia, MySQL Workbench, Telegram eta ordenagailua eta mugikorra internet konexioarekin.

<b>3. Prototipoa</b>
<b>Lan-blokea:</b> Inplementazioa
<b>Denbora estimazioa:</b> 30 ordu.
<b>Deskribapena:</b> Erabiltzaileak memoriaren bertsio ezberdinak bot-ari bidali eta bot-ak zuzentzea komandoen bitartez.
<b>Irteerak/Emangarriak:</b> Memorien bertsioak kontrolatzen eta zuzentzen dituen kodea.
<b>Baliabideak:</b> PHP, PhpStorm, php-telegram-bot liburutegia, MySQL Workbench, Telegram, LanguageTool eta ordenagailua eta mugikorra internet konexioarekin.

<b>4. Prototipoa</b>
<b>Lan-blokea:</b> Inplementazioa
<b>Denbora estimazioa:</b> 10 ordu.
<b>Deskribapena:</b> Erabiltzaileek bidalitako memorien testua plagioa den ikusten duen komandoa.
<b>Irteerak/Emangarriak:</b> Plagioa den begiratzen duen kodea.
<b>Baliabideak:</b> PHP, PhpStorm, php-telegram-bot liburutegia, MySQL Workbench, Telegram eta ordenagailua eta mugikorra internet konexioarekin.

### ***5. Prototipoa***

**Lan-blokea:** Inplementazioa

**Denbora estimazioa:** 20 ordu.

**Deskribapena:** Erabiltzaileek bidalitako memorian agertzen diren irudiak internet aurkitzen diren ikusten duen komandoa.

**Irteerak/Emangarriak:** Irudiak aztertzen dituen komandoaren kodea.

**Baliabideak:** PHP, PhpStorm, php-telegram-bot liburutegia, MySQL Workbench, Telegram, Cloud Vision API eta ordenagailua eta mugikorra internet konexioarekin.

### ***Administrazio panela***

**Lan-blokea:** Inplementazioa

**Denbora estimazioa:** 30 ordu.

**Deskribapena:** Datu-basean informazioa eguneratuta mantentzeko administrazio panel baten inplementazioa.

**Irteerak/Emangarriak:** Administrazio panelaren kodea.

**Baliabideak:** Python, PyCharm, Django liburutegia, MySQL Workbench, ordenagailua.

### **3.3.5 Testak**

Test fasean inplementatutako prototipoak betekizunak betetzen dituela egiaztatuko da. Modu inkrementalean egingo da, hau da, test bakoitzean prototipo horren funtzionalitateaz gain, lehenagoko prototipoen funtzionalitateak ere ondo daudela egiaztatu behar da.

### ***1. Prototipoa***

<b>Lan-blokea:</b> Testak
<b>Denbora estimazioa:</b> 3 ordu.
<b>Deskribapena:</b> 1. prototipoko funtzionalitateak probatu proba kasu ahalik eta gehienekin.
<b>Irteerak/Emangarriak:</b> Proba kasuen emaitza.
<b>Baliabideak:</b> Telegram eta MySQL Workbench

### ***2. Prototipoa***

<b>Lan-blokea:</b> Testak
<b>Denbora estimazioa:</b> 5 ordu.
<b>Deskribapena:</b> 2. prototipoko funtzionalitateak probatu proba kasu ahalik eta gehienekin eta arinagoko funtzionalitateak ondo jarraizten dutela egiaztatu.
<b>Irteerak/Emangarriak:</b> Proba kasuen emaitza.
<b>Baliabideak:</b> Telegram eta MySQL Workbench

### ***3. Prototipoa***

<b>Lan-blokea:</b> Testak
<b>Denbora estimazioa:</b> 5 ordu.
<b>Deskribapena:</b> 3. prototipoko funtzionalitateak probatu proba kasu ahalik eta gehienekin eta arinagoko funtzionalitateak ondo jarraizten dutela egiaztatu.
<b>Irteerak/Emangarriak:</b> Proba kasuen emaitza.
<b>Baliabideak:</b> Telegram eta MySQL Workbench

### ***4. Prototipoa***

<b>Lan-blokea:</b> Testak
<b>Denbora estimazioa:</b> 3 ordu.
<b>Deskribapena:</b> 4. prototipoko funtzionalitateak probatu proba kasu ahalik eta gehienekin eta arinagoko funtzionalitateak ondo jarraizten dutela egiaztatu.
<b>Irteerak/Emangarriak:</b> Proba kasuen emaitza.
<b>Baliabideak:</b> Telegram eta MySQL Workbench

<i>5. Prototipoa</i>
<b>Lan-blokea:</b> Testak
<b>Denbora estimazioa:</b> 8 ordu.
<b>Deskribapena:</b> 5. prototipoko funtzionalitateak probatu proba kasu ahalik eta gehienekin eta arinagoko funtzionalitateak ondo jarraizten dutela egiaztatatu.
<b>Irteerak/Emangarriak:</b> Proba kasuen emaitza.
<b>Baliabideak:</b> Telegram eta MySQL Workbench

<i>Administrazio panela</i>
<b>Lan-blokea:</b> Testak
<b>Denbora estimazioa:</b> 3 ordu.
<b>Deskribapena:</b> Administrazio panela ono funtzionatzen duela probatu, informazioa DBan sartu, aldatu eta ezabatzen duen.
<b>Irteerak/Emangarriak:</b> Proba kasuen emaitza.
<b>Baliabideak:</b> Administrazio panela eta MySQL Workbench

### 3.3.6 Dokumentazioa

Bloke honetan proiektuaren dokumentazio idatziarekin zerikusi duten jarduerak ageri dira.

<i>Memoriaren idazketa</i>
<b>Lan-blokea:</b> Dokumentazioa
<b>Denbora estimazioa:</b> 75 ordu.
<b>Deskribapena:</b> Proiektuaren memoria idatzi.
<b>Irteerak/Emangarriak:</b> Proiektuaren memoria.
<b>Baliabideak:</b> Overleaf, Visual Paradigm

<i>Aurkezpena prestatu</i>
<b>Lan-blokea:</b> Dokumentazioa
<b>Denbora estimazioa:</b> 15 ordu.
<b>Deskribapena:</b> Defentsarako aurkezpena prestatu.
<b>Irteerak/Emangarriak:</b> Aurkezpenerako diapositibak.
<b>Baliabideak:</b> PowerPoint.

### 3.4 Denbora planifikazioa

Atal honetan proiektuaren denboraren estimazioa totala kalkulatu da aurreko ataletan azaldutako jarduerak kontuan hartuta. Ondoren, jardueren banaketa denboran zehar azalduko da.

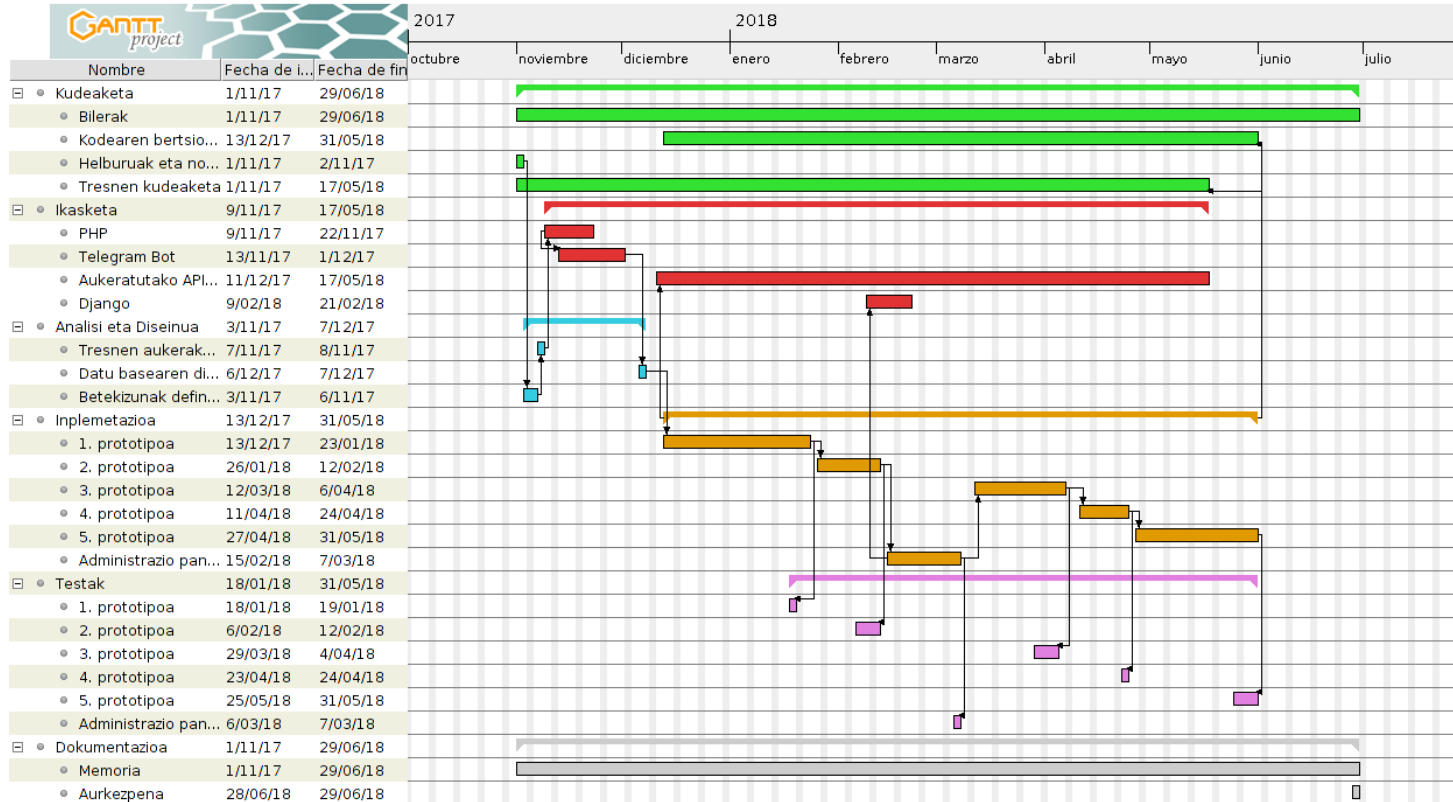
1. taulan proiektuaren denbora estimazioa jarduerako eta batuketa totala agertzen dira. Proiektu osoa garatzeko estimatutako denbora 367 ordukoa da. Denbora gehien behar duten jarduerak implementazio atalekoak dira, 135 ordu, proiektuaren heren bat baino gehiago. Dokumentazioak ere ordu nahiko behar ditu, 90. Ordu gutxien behar duten blokeak Analisia eta diseinua eta testak dira, 18 eta 27 ordu hurrenez hurren.

Proiektuaren banaketa denboran zehar 2. irudian agertzen den Gantt diagraman ikusi daiteke. Planifikazioan proiektuaren hasiera azaroan kokatzen da. Amaiera ekainaren amaieran, GRALeke hirugarren deialdian amaitzeko. Jarduerak, bereziki implementazioarenak, beraien denbora estimatua kontuan izan da diagraman jartzeko, baina baita data horietan lan egiteko denbora kontuan izanda. Adibidez, lehenengo prototipoak denboran hirugarrenak baino zati zabalago hartzen du bere denbora estimazioa txikiago izanda; baina abenduan eta urtarrilean lauhilekoaren amaiera da eta projektuan lan egiteko denbora murriztuko dela suposatzen da.

Taula 1: Jardueren denbora estimazioa

JARDUERAK	DENBORA ESTIMAZIOA
<b>KUDEAKETA</b>	<b>42</b>
Bilerak	20
Kodearen bertsio kontrola	10
Helburuak eta norainokoak definitu	2
Tresnen kudeaketa	10
<b>IKASKETA</b>	<b>55</b>
PHP	10
Telegram Bot	20
Aukeratutako APIak	10
Django	15
<b>ANALISI ETA DISEINUA</b>	<b>18</b>
Tresnen aukeraketa	10
Datu-basearen diseinua	5
Betekizunak definitu	3
<b>INPLEMENTAZIOA</b>	<b>135</b>
1. Prototipoa	25
2. Prototipoa	20
3. Prototipoa	30
4. Prototipoa	10
5. Prototipoa	20
Administrazio panela	30
<b>TESTAK</b>	<b>27</b>
1. Prototipoa	3
2. Prototipoa	5
3. Prototipoa	5
4. Prototipoa	3
5. Prototipoa	8
Administrazio panela	3
<b>DOKUMENTAZIOA</b>	<b>90</b>
Memoria	75
Aurkezpena	15
<b>GUZTIRA</b>	<b>367</b>





Irudia 2: Proiektuaren Gantt diagrama

### 3.5 Analisi ekonomikoa

Proiektu hau unibertsitateko proposamen bat denez, ez da espero inolako ordainsaririk. Hala ere, atal honetan proiektuaren analisi ekonomikoa egingo da, proiektuaren gastuak kalkulatzeko.

#### Eskulana

Programazioko proiektu batean eskulana izaten da gastu gehien sortzen duena. 2018ko martxoaren 6ko BOEaren arabera, programatzaile baten batezbesteko hileko soldata 1500€-koa da gutxi gorabehera. Orduko soldata ondorengo formulen bitartez kalkula daiteke:

$$Soldata/ordu = \frac{Soldata\ hilabetean}{Asteak/Hil.\ * Lanegunak/Astean\ * Lanorduak/egunean} \quad (1)$$

Normalean hilabete batek 4 aste izaten ditu eta aste bakoitzak 5 lanegun 8 ordukoa bakoitza. Beraz, ekuazioa horrela izango litzake:

$$Soldata/ordu = \frac{1500}{4 * 5 * 8} = \frac{1500}{160} = 9,37 \quad (2)$$

Guztira 9,37€ irabaziko da lanordu bakoitzeko.

Proiektua garatzeko estimatutako denbora 367 ordukoa denez:

$$Soldata = Lanorduak * Soldata/orduro = 367 * 9,37 = 3438,79 \quad (3)$$

Eskulanari dagokion gastu ekonomikoa 3438,79€-koa izango litzake.

#### Software

Ez da gasturik egin softwarean. Proiektuan erabiliko diren ia tresna guztiak doakoak dira edo proba epe barruan erabili dira. Googleren Cloud Vision APIa erabiltzea izan daiteke gastu sor dezakeena, baina Googleek doako kreditu kantitate bat ematen du hau erabiltzeko, 300\$ 12 hilabetetan erabiltzeko. Gainera muga batzuen barruan mantenduz gero zerbitzua doakoa izaten jarraitzen du. Hala ere, erabiltzaile askok erabiliko balute kontuan izan beharreko gastu bat izango litzateke, baina proiektua garatzeko ez da beharrezkoa izango.

## Hardware

Proiektua garatzeko erabili den hardwarea ordenagailu eramangarria eta mugikorra izan dira.

Ordenagailuaren ustezko bizi iraupena 4 urtekoa da eta bere kostua 800€. Beraz, ordenagailuaren hilabeteroko amortizazioa ondorengoa da:

$$\text{Amort. hilabetero} = \frac{\text{Prezioa}}{\text{Ustezko bizi iraupena}} = \frac{800}{12 * 4} = 16,67 \quad (4)$$

Hilabeteroko amortizazioa 16,67€-koa da eta eramangarria proiektu osoan erabiliko denez, hau da, 8 hilabete:

$$\text{Amortizazioa} = \text{Amort. hilabetero} * \text{Hilabete kopurua} = 16,67 * 8 = 133,36 \quad (5)$$

Eramangariaren amortizazioa 133,36€-koa izango da.

Mugikorrari dagokionez, bere bizi iraupena 2 urtekoa da eta bere kostua 220€-koa. Beraz, mugikorraren hilabeteroko amortizazioa ondorengoa da:

$$\text{Amort. hilabetero} = \frac{\text{Prezioa}}{\text{Ustezko bizi iraupena}} = \frac{220}{12 * 2} = 9,17 \quad (6)$$

Hilabeteroko amortizazioa 9,17€-koa da eta mugikorra proiektuaren inplementazioan eta testean erabiliko denez 6 hilabetez erabiliko da:

$$\text{Amortizazioa} = \text{Amort. hilabetero} * \text{Hilabete kopurua} = 9,17 * 6 = 55,02 \quad (7)$$

Mugikorraren amortizazioa 55,02€-koa izango da.

Hardware osoaren gastua kalkulatzeko:

$$\text{Hardware amort.} = \text{Eramangarria} + \text{Mugikorra} = 133,36 + 55,02 = 188,38 \quad (8)$$

Guztira, hardwarearen gastua 188,38€-koa izango da.

## Zeharkako gastuak

Proiektua garatzeko beharrezkoak izango diren, baina garapenarekin lotura zuzena ez daukaten gastuak dira.

Bereziki, atal honetan gehitu beharreko gastuak erabiliko den argindarra eta internet konexioa izango dira. Gastu hauek aldakorrak izan daitekeenez kuota finkoa ezarriko da, 30€hilabetean. Hilabete kopurua kontuan izanda:

$$\text{Zeharkako gastuak} = \text{Kuota} * \text{Hilabete kopurua} = 30 * 8 = 240 \quad (9)$$

Zeharkako gastuen kostu totala 240€-koa izango da.

## Gastu guztien batura

Atal honetan, kalkulatu diren gastu guztiak elkartuko dira proiektuaren kostu totala ezagutzeko. 2. taulan gastu guztiak ikus daitezke eta proiektuaren kostu totala 3867,17€-koa dela kalkulatu da.

Taula 2: Proiektuaren kostuak

<i>Gastu motak</i>	<i>Gastua (€)</i>
<b>Eskulana</b>	<b>3438,79 €</b>
<b>Software</b>	<b>0 €</b>
<b>Hardware</b>	<b>188,38 €</b>
<b>Zeharkako gastuak</b>	<b>240 €</b>
<b>Guztira</b>	<b>3867,17 €</b>

## 3.6 Arriskuaren analisia

Proiektu baten garapenean arriskuak sortu daitezke. Beraz, hauek identifikatzea garrantzitsua da, hauek saihesteko edo gertatzen badira, ahalik eta arinen konpontzeko.

### Arazoak ordenagailuarekin

- **Deskribapena:** Ordenagailuaren arazo baten ondorioz egindako lana galitzea, ordenagailua bat-batean amatatzen delako edota guztiz hondatzen delako.
- **Probabilitatea:** Ertaina
- **Eragina:** Handia
- **Prebentzioa:** Ordenagailuaren mantentzea egunean izan eta lanaren babes kopiak egin.
- **Kontingentzia plana:** Beste ordenagailu batean lan egin eta babes kopietatik informazioa berreskuratu.

### Gaixorik egotea

- **Deskribapena:** Edonolako osasun arazoak izatea.
- **Probabilitatea:** Ertaina
- **Eragina:** Handia

- **Prebentzioa:** Osasun egokia izateko ohikoak diren ekintzak egin: errespetatu janari eta lo orduak, atsedenak hartu lan egiterakoan...
- **Kontingentzia plana:** Medikura joan edota beharrezkoa den medikamentua hartu eta beharrezkoa bada lan gutxiago egin.

### Irakasgaietako lana pilatzea

- **Deskribapena:** Beste irakasgaietako proiektuak edota azterketak data batean pilatzea.
- **Probabilitatea:** Handia
- **Eragina:** Handia
- **Prebentzioa:** Ez utzi irakasgaietako lana azken egunetan egiteko.
- **Kontingentzia plana:** Lehenengo irakasgaietako lana egin eta ondoren proiektua. Beharrezkoa bada, proiektuaren amaiera luzatuko da.

### Denbora estimazioa

- **Deskribapena:** Egindako denbora estimazioa baxuegia izatea.
- **Probabilitatea:** Txikia
- **Eragina:** Handia
- **Prebentzioa:** Estimazioa egiterako orduan, pentsatutakoa baino denbora pixka bat gehiago esleitu.
- **Kontingentzia plana:** Lan orduak gehitu edota amaiera data luzatu.

### Aukeratutako API edota liburutegiak

- **Deskribapena:** Aukeratutako API edota liburutegiak aldatzea edo funtzionatzeari uztea.
- **Probabilitatea:** Txikia
- **Eragina:** Handia
- **Prebentzioa:** APIak edota liburutegiak aukeratzeko orduan begiratu dokumentazioan ea aldaketarik jasango duten eta horrela bada beste tresna bat aukeratu edota planifikazioan kontuan izan.
- **Kontingentzia plana:** API edota liburutegien bertsio zaharra funtzionatzen jarraitzen badu ez egin ezer. Bestela, beste tresna bat bilatu.



## 4 Betekizunen bilketa

Atal honetan bot-ak bete behar dituen betekizunak azalduko dira. Betekizun hauek erabiltzaileen premiak asebetetzeko beharrezkoak diren funtzionalitateak aztertuz lortu dira. Hortaz, betekizun bakoitza erabiltzailearen premiez eta bot-ak hauek betetzeko egin beharreko prozesuez baldintzatuta egongo da.

- **Maiz Egiten diren Galderak erantzun:** Bot-ak ikasleek maiz egiten dituzten galderak (FAQ) erantzun beharko ditu. Galdera hauek askotan arautegiarekin edota deialdien informazioarekin zerikusia dute. Beraz, bot-ak ez ditu soilik erantzunak emango, beharrezkoa den dokumentazioa ere bidaliko du.
- **Deialdien datak:** FAQ erantzutean bot-ak deialdien data eguneratuekin, hau da, hurrengo datekin, erantzungo du. Bestalde, deialdi baten data bat hurbil dagoenean, bot-ak erabiltzaileei abisu bat bidali beharko die.
- **GRALak bidali:** Erabiltzaile bakoitzak GRAL bakar bat erregistratu ahalko du bot-ean, baina GRAL horretako bertsio ezberdinak bidali ahalko dizkio, hauek aztertzeko eta ikusteko ea hobetuz doan lana. Gainera, bot-ak hainbat fitxategi mota onartuko ditu: pdf, docx eta odt.
- **GRAL zuzenketa:** GRALen zuzenketa gramatikala eta ortografikoa egingo da API bati deiak eginez. Erabiltzaileari zuzentzaileak aurkitutako erroreak erakutsiko zaizkio.
- **GRALen plagioak:** GRALeko testua eta irudiak originalak diren egiaztatuko da. Plagioren bat aurkitzen bada erabiltzaileari jakinaraziko zaio eta berak erabaki beharko du eduki hori erabili ahal duen, daukan lizenziaren arabera.
- **Hizkuntza:** Bot-a eta bere funtzionalitateak gaztelaniaz egon beharko dute. Hala ere, aurrerago beste hizkuntza batzuk gehitzeko prestatuta egon beharko da.
- **Pantailaratu:** Telegram gailu ezberdinetan erabili daiteke, gehienbat mugikorretan eta ordenagailuetan erabiliko da. Hortaz, mezuak pantailaratzeko orduan, mezuen tamaina kontuan izan behar da, mezua oso luzea bada mugikorrean zailagoa izango baita ondo ikustea.
- **Informazioa eguneratu:** Datu-basea eguneratuta eta aldaketetarako prestatuta egon behar da. Hori errazteko, administrazio panel bat egongo da, non galderak gehitu, kendu edo aldatu daitezke eta deialdietako datak eta abisuak kudeatuko dira.





## 5 Tresnen hautapena

Proiektuaren garapena egiteko erabili diren programazio lengoaiak, liburutegi eta beste software batzuk nola aukeratu ziren azalduko da atal honetan.

### 5.1 Programazio lengoaiak

Proiektuan hartutako lehen erabakia izan zen bot-a programatzeko erabiliko zen lengoia aukeratzea. Juananekin bildu nintzenean, esan zidan bot-a hiru lengoia ezberdinetan programatu ahal nuela, PHP, Python edo NodeJS, baina normalean berak PHPn programatzen duela.

Aipatutako hiru lengoia horretatik, NodeJS ez dut inoz erabili eta PHP oso gutxi, ondo ezagutzen nuen lengoia bakarra Python zen. Baina proiektu honekin, nik GRALa amaitu ostean, jarraitu nahi zuela jakinda eta normalean PHP erabiltzen duela, bot-a inplementatzeko PHP lengoia aukeratu zen. Honekin batera, Juananek bot-ak programatzeko erabiltzen duen PHP liburutegia, php-telegram-bot, aukeratu zen.

Administrazio panel bat beharrezko izango zela erabaki zenean, Juanan komentatu zidan Pythoneko Django liburutegiaren administrazio panel xume bat egitea erraza zela. Django web aplikazioak egiteko liburutegi egokia zela entzunda nuen eta Juananek hau komentatzean Django erabiltzea hautatu zen, liburutegi hau erabiltzen ikasteko gogoia piztu zitzaidan.

### 5.2 Beste software eta liburutegiak

Aukeratu beharreko gainontzeko tresnak bot-aren funtzionalitateekin lotuta daude.

- **Fitxategiak pdf bihurtu:** Bot-aren funtzionalitateak erabili ahal izateko pdf fitxategiak behar dira, baina erabiltzaileak, pdf, doc, docx eta odt fitxategi motak ere igo ditzake. Hortaz, fitxategi horiek pdf bihurtzeko erreminta bat beharrezkoa zen. Aurkitutako tresna gehienek fitxategi mota bakarra bihurtu zezaketen pdf. Beharrezkoak ziren fitxategi mota guztiak pdf bihurtzeko gai zen aurkitutako lehen tresna aukeratu zen, unoconv.
- **Pdf-tik testua atera:** Lehenago garatutako proiektu batetan pdf-etatik testua atera behar izan nuen eta eginkizun hau egiteko pare bat erreminta eta hauen emaitzak nolakoak ziren ezagutzen nituen. Baina proiektu hura Pythonen garatutako zenez, PHPn antzeko tresnak bilatzen, spatie/pdf-to-text PHPko liburutegiaren lortzen ziren testuak onak zirela ikusita proiektuan erabiltzea erabaki zen.

- **Pdf-etik irudiak:** Eginkizun hau burutzeko pdfimages erabili da. Juananek gomendatutako tresna bat izan zen. Komando lerrotik exekutatu da eta oso erabilerraza da.
- **Zuzentzailea:** Gaztelaniazko testua zuzentzeko tresna bat behar zen. Aztertutako zuzentzaile gehien artean arazo bi aurkitu ziren: ez zuten gaztelera zuzentzen edo erreminta erabiltzeko limiteak baxuak ziren. Aukeratutako tresna LanguageTool izan da. Tresna honek ere limiteak ditu, adibidez 20KB testu eskaera bakoitzeko. Baina beraien zerbitzaria erabili ordez, zure zerbitzarian zuzentzailearen instantzia bat instalatzeko aukera dago.
- **Irudiak aztertu:** Irudiak interneten agertzen diren aztertzen duen tresna bat beharrezkoa zen. Googlek irudiak bilatzeko aukera eskaintzen duenez irudiak bilatzeko APIa kontsultatu zen, baina API hau 2011tik gaitzetsita dago eta Custom Search APIa erabiltzea gomendatzen dute. Baina API hau ez zen aproposa. Googleen APIetan bilatuz Cloud Vision APIa aurkitu zen. API honek funtzionalitate ugari ditu argazkiekin lan egiteko, baita proiekturako beharrezkoa zena.

## 6 Analisi eta diseinua

Atal honetan proiektua funtziona dezan beharrezko den datu-basearen diseinua azalduko da.

Proiektuan erabilitako DBak garapenean zehar hainbat aldaketa pairatu ditu, funtzionalitateak pixkana-pixkana gehituz joan direlako. 3. irudian DBaren azken diseinua ageri da. Garrantzitsua da esatea, atal honetan azalduko den DBaren diseinua, bot-ak izan behar dituen funtzionalitateak erabili ahal izateko sortu diren taulak direla. Honetaz aparte, DBan beste taula ugari daude, php-telegram-bot liburutegiak eta Django behar dituztenak, baina beraiek kudeatzen dituztenez ez dira aipatuko.

Ondoren, DBko taulak azalduko banaka-banaka azalduko dira, funtzionalitatearen arabera.

### GRALen kudeaketa

GRALak kudeatzeko beharrezkoak diren taulak azalduko dira:

- **TFG**: GRAL baten informazioa, GRALa erregistratzean gordetzen da. GRALaren izenburua, hizkuntza eta zein erabiltzailearena den adierazten du.
- **TFGversion**: Erabiltzaile batek GRAL baten memoriaren bertsio ezberdinak kudeatzeko. Gordetzen duen informazioa hau da: bertsio zenbakia, zein TFGri dagokion, hash-a eta igondako data. Gainontzeko atributuak zerbitzariko helbide ezberdinekin lotuta daude, jakiteko non gordetzen den memoriaren fitxategiak edota erabiltzaileak komando bat exekutatzeko erantzuna non gorde den adierazteko.
- **TFGimage**: GRAL baten irudien informazioa gordetzen du. Irudia GRAL zein bertsiori dagokio, zerbitzarian non dagoen, hash-a eta Google Cloud Vision APIak itzulitako erantzuna gordetzen du.

### FAQ galderak

Jarraian FAQ galdera eta erantzunak kudeatzeko beharrezkoak diren taulak azalduko dira.

- **faq\_question**: FAQ galderak adierazten dituen taula, galderaren testua eta zein ordenetan agertuko den zerrendan adierazten du.
- **faq\_response**: Galdera baten erantzuna, galdera batek hainbat erantzun izan ditzake. Zein galderari dagokion eta bere testua zein den gordetzen ditu. Gainera, erantzun berezia bada, hau da, erantzunean irudi, bideo,

dokumentu edota data bidaltzen duen, informazio hori gordetzen du. Iru-di, bideo eta dokumentuen zerbitzariko helbideak gordeko dira.

- **messages\_lang:** Hizkuntzak kudeatzeko taula, non FAQ galdera eta erantzunen testuak gordetzen dira. Taulak FAQ galdera eta erantzunen testua gaztelaniaz eta euskaraz gordetzen du. Testu bakoitzari etiketa bat jarri ahal zaio testua errazago aurkitu ahal izateko.

## Abisuak

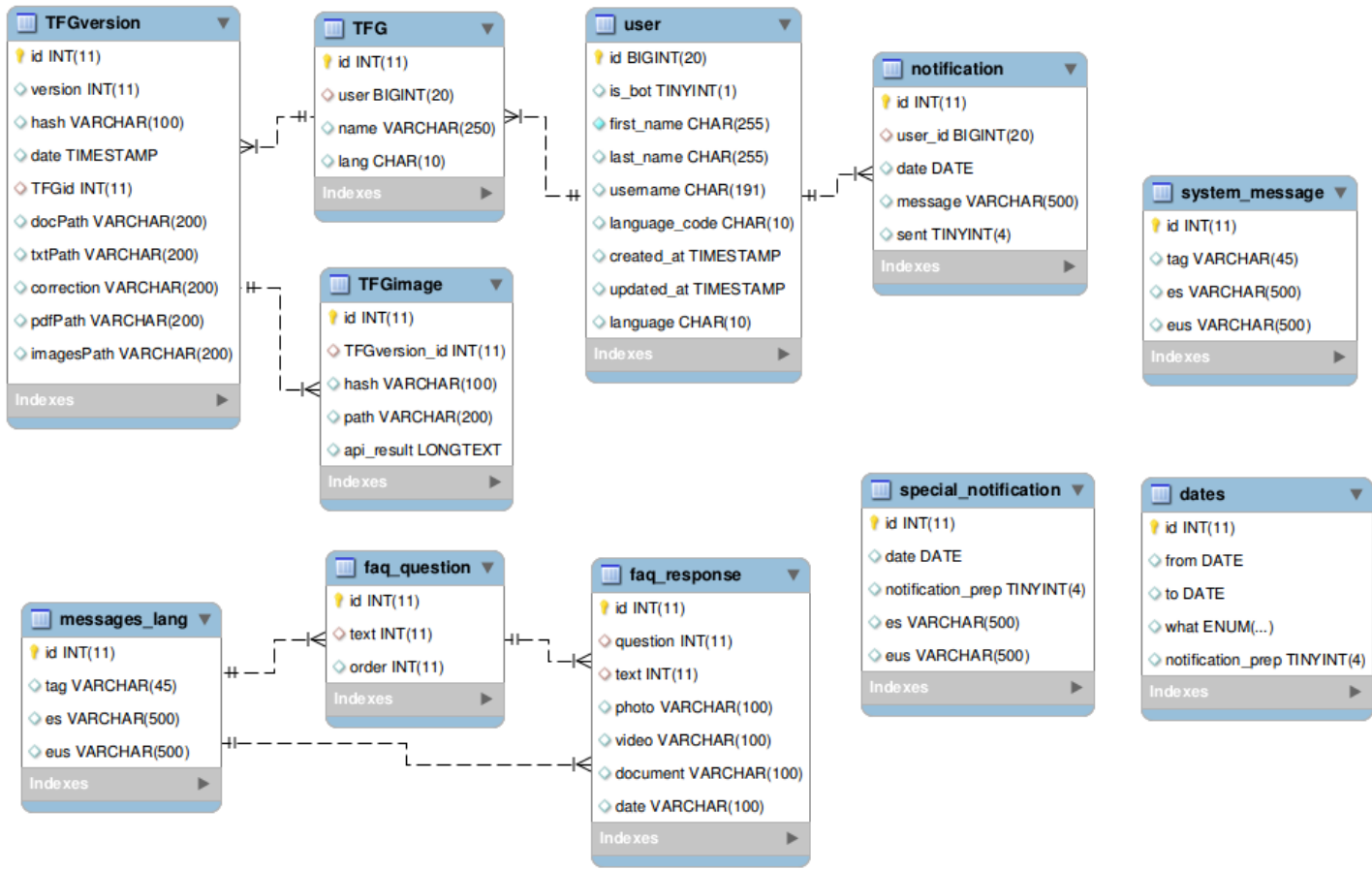
Bot-ak bi abisu mota ditu, deialdien daten abisuak eta abisu bereziak. Hauek kudeatzeko taulak honako hauek dira:

- **dates:** Taula honetan deialdien datak gordetzen dira, noiztik nora eta zer data den, adibidez matrikula data edo defentsa data. Data hauen abisuak eman behar direnez, data baten abisuak prestatu dutenak eta ez dutenak banatzeko aldagai bitar bat dago. FAQ erantzunak data bat dutenean informazioa hemendik hartzen dute.
- **special\_notification:** Taula honetan abisu bereziak gordetzen dira; duen informazioa hau da: abisua bidali behar den data, gaztelaniazko eta euskarazko testua. Gainera, daten bezala, aldagai bitar bat du abisua prestatuta dagoen markatzeko.
- **notification:** Taula honetan bidalitako edo bidaltzeko dauden abisuak gordetzen dira. Bi kasu hauek ezberdintzeko atributu bitar bat du. Taulan gordetzen den informazioa hau da: noiz bidali, nori bidali eta mezuaren testua.

## Bestelakoak

Atal honetan lehen aipatutako funtzionalitate ezberdinekin lotura ez duten DB-ko taulak azalduko dira (ikus 3. irudia).

- **system\_message:** messages\_lang taularen egitura bera du, taula hau ere hizkuntzak kudeatzeko baita. Bi taula berdinean egitearen zergatia da messages\_lang taula administrazio paneletik aldatu ahal dela eta system\_message taula ez, bot-ak funtzionatzeko beharrezkoak diren testuak gordetzen baititu eta ez da nahi administrazio paneletik aldatu ahal izatea.
- **user:** Taula hau php-telegram-bot liburutegiaren taula bat da, non bot-aren erabiltzaileen datuak gordetzen diren. Taula hau aldatu egin da, atributu bat gehitu zaio, erabiltzaileak bot-a zein hizkuntzatan erabili nahi duen jakiteko. Gainera, taula hau beharrezkoa da GRALak erabiltzailearekin lotzeko eta abisuak nori bidali behar zaizkion jakiteko.



Irudia 3: Bot-a DBaren diseinua.



## 7 Arkitektura

Garatu den aplikazioak funtzionatzeko beharrezkoak dituen tresnen eta zerbitzuen arteko informazio aldaketa nola gertatzen den azalduko da atal honetan.

4. irudian jarraian azalduko denaren laburpena modu grafikoan ikus daiteke.

### 7.1 Telegram bot

Bot-a zerbitzari batean kokatuta dago. PHP lengoaiari dagoen `php-telegram-bot` liburutegia erabiltzen du Telegram Bot APIa erabiltzeko. Bot-aren kodean bi motako fitxategiak daude: komandoen fitxategiak, komando bakoitzeko `php` fitxategi bat; eta beste script batzuk, abisuak bidaltzeko adibidez.

Erabiltzaile batek bot-ari Telegram aplikazioaren bitartez mezu bat bidaltzen dionean, mezu hau Telegrameko zerbitzarira heltzen da. Bot-ak mezu hori jasotzeko bi modu daude:

- **Polling:** bot-ak Telegrami berarentzat mezurik duen galdetzean datza. APIko `getUpdates` metodoa erabiliz egiten da, eta mezurik badago erantzunean jasoko dira. Benetako elkarrizketa bat simulatu nahi bada bot-ak etengabe egon beharko da galdetzen. Normalean bot-a garatzen dagoenean erabiltzen da.
- **Webhook:** Telegram APIko `setWebhook` metodoa erabiliz, url bat zehaztu daiteke webhook bezala. Webhookari esker Telegramek bot-arentzat mezu bat jaso bezain laster bot-ari bidaliko dio zehaztutako urla. Modu honi esker, galdezketaren gotea ekidin daiteke.

Behin bot-ak mezua jaso duela, komando baten kodearen bitartez mezuaren araberrako erantzun bat prestatuko du eta Telegrameko zerbitzariari bidaliko dio erabiltzaileari aplikaziora heltzeko.

Bot-aren erantzuna prestatzeko, datu-basea kontsultatu daiteke edota beste software eta zerbitzuei deiak egin beharko zaizkio bot-aren kodetik. Adibidez, erabiltzaileak fitxategi bat bidaltzen badu lehenengo pausua bot-arentzat, Telegrametik fitxategia deskargatzea izango da, Telegram Bot APIa erabiliz. Ondoren erabiliko diren tresna ezberdinak aipatuko dira:

- Fitxategia pdf formatuan ez badago, zerbitzarian dagoen `unoconv` softwarea erabiliko da, pdf formatuan lortzeko.
- Pdf-tik testua lortzeko PHPko `spatie/pdf-to-text` liburutegia erabiliko da.
- Pdf-tik irudiak ateratzeko zerbitzarian dagoen `pdfimages` softwarea erabiliko da.
- Testuaren zuzenketa egiteko, testua `LanguageTool` softwareari HTTP eskaera bat bidaliko zaio. Honek JSON erantzun bat itzuliko du.

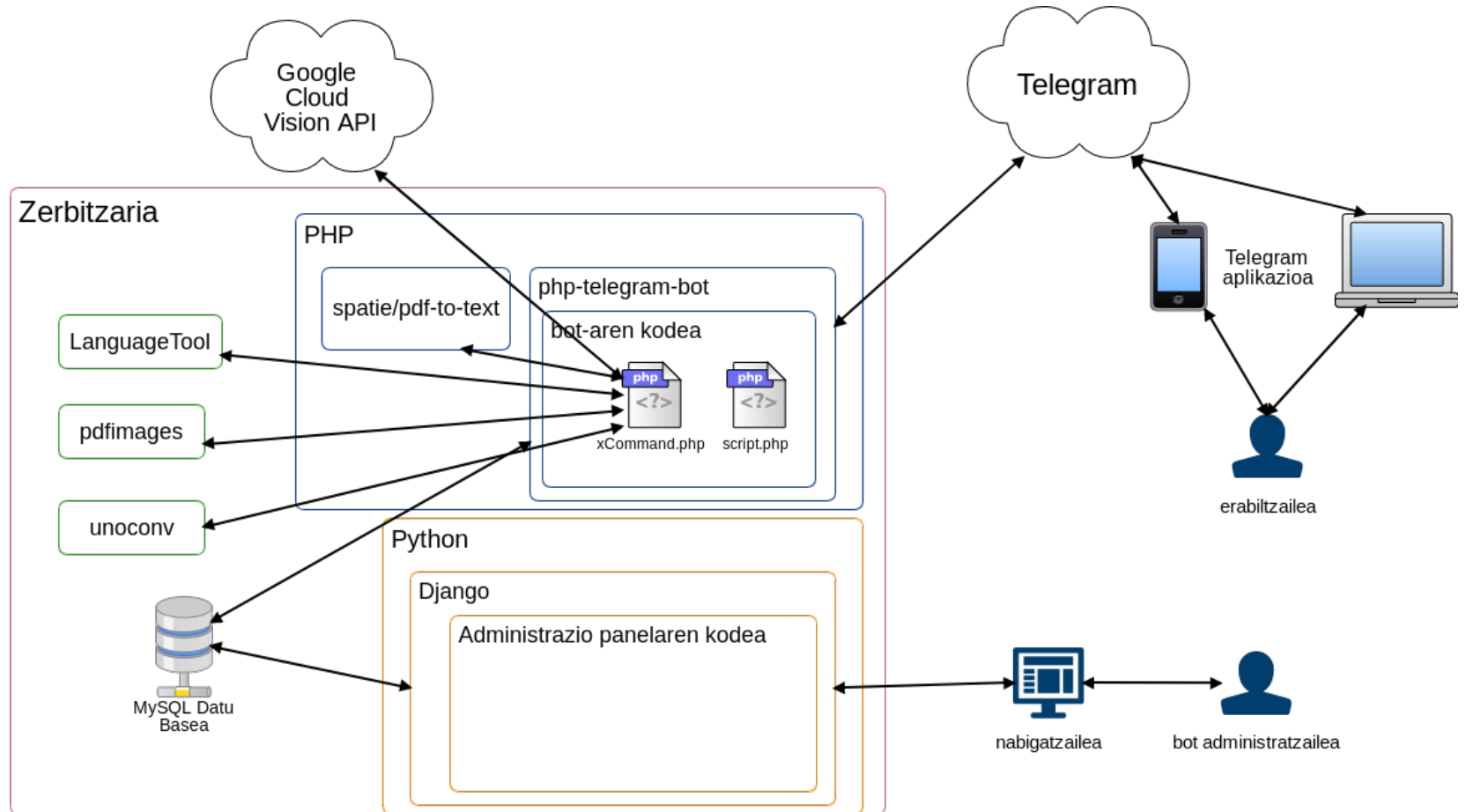
- Irudiak aztertzeko Google Cloud Vision APIa erabiliko da. Hau Googleen zerbitzu bat denez, APIa erabiliz, HTTP POST eskaerak bidaliko dizkiogu bot-aren kodetik. Erantzuna JSON formatuan bidaltzen du.

Bot-aren kodean dauden script batzuk, adibidez abisuenak, ez dira exekutatzen erabiltzaileak bidalitako mezu baten ondorioz, guk nahi dugunean exekutatuko dira. Datu-basean kontsultak egin ditzakete eta erabiltzaileei mezuak bidali Telegram Bot APIaren bitartez.

## 7.2 Administrazio panela

Administrazio panelak Pythonen programatutako Django frameworka erabiltzen du. Bot-aren administratzailea web nabigatzaile baten bitartez administrazio paneleko interfazea ikusi ahalko du, zerbitzariaren helbidera konektatzen bada. Interfazean erakutsiko den informazioa DBtik hartutakoa izango da eta egiten den edozein aldaketa DBan gordeko da.





Irudia 4: Aplikazioaren arkitektura.



## 8 Garapena

Atal honetan proiektuaren funtzionalitateak nola garatu diren eta izandako arazo batzuk azalduko da. Baina garapenarekin hasi aurretik, bot-aren garapenean erabilitako zenbait elementu azalduko dira.

### Elkarrizketa

Komando batek erabiltzailearekin interakzio bat baino gehiago behar duenean elkarrizketa bat sortzen da, adibidez erregistro batetan non erabiltzaileari bere datuak banaka eskatzen zaizkion. Elkarrizketa bat amaituko da erabiltzaileak edo bot-ak gelditzen dutenean. Beraz, elkarrizketa batek hiru egoera ditu:

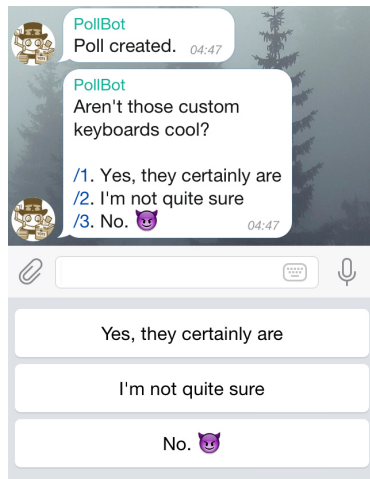
- **Geldituta:** Aktibatuta ez dagoenean bot-ak elkarrizketa amaitu duelako ez delako erabiltzailearekin interakzio gehiago behar. Komandoa era egokian amaitzen denean gertatzen da.
- **Deuseztatuta:** Aktibatuta ez dagoenean erabiltzaileak elkarrizketa amaitu duelako /cancel komandoa erabiliz. Erabiltzaileak komando hori erabiltzen duenean, momentuko elkarrizketa deuseztatzen da.
- **Aktibatuta:** Elkarrizketa egoera honetan dagoen bitartean, erabiltzaileak bot-ari bidaltzen dion mezu bakoitzeko, elkarrizketa hasi duen komandoa exekutatu da.

Elkarrizketa bat aktibatuta dagoenean bitartean bot-ak hainbat zeregin egin ditzake. Jakiteko momentu bakoitzean zer egin behar duen, hau da, elkarrizketaren zein unetan doa, bot-ak DBan elkarrizketaren barruko egoera identifikazioa gordetzen du ohar batean. Gainera, elkarrizketan zehar lortutako informazioa ere gorde daiteke elkarrizketaren oharretan. Modu honetan, elkarrizketaren edozein unetan, lehengo informazio guztia berreskuratu eta erabili daiteke.

### Botoiak

Telegrameko bot-ek bi motatako botoiak erabiltzen dituzte:

- **KeyboardButton:** Botoi hauek programatzaileak zehazten ditu eta mugikorraren teklatua ezkututzen dute, bere tokian jarritik. Mota honetako botoi bat sakatzean, erabiltzaileak mezu bat bidaltzen du botoiak duen testu berarekin. Erabiltzaileak era errazagoan datuak bidaltzeko erabiltzen da, ez baitu ezer idatzi behar, botoi bat sakatu baizik.
- **InlineKeyboardButton:** Botoi hauek bot-ak bidaltzen dituen mezuen tokian agertzen dira. *KeyboardButton* erabiltzea baino konplexuagoak dira. Botoi hauek sakatzean pantailan ez da agertzen inolako mezurik *KeyboardButton*ekin gertatzen den bezala, baizik eta ezkutatuta daukaten programatzaileak esleitutako datuak bidaltzen diote bot-ari, *callback\_data*



Irudia 5: *KeyboardButton* motako hiru botoi bata bestearen gainean, mugikorraren teklatuaren tokian.



Irudia 6: *InlineKeyboardButton* motako botoiak matrize itxuran kokatuta, bot-aren mezuen azpian.

deitutako parametroan dagoen informazioa hain zuzen ere. Informazio hau bot-ari *callback query* deitutako komando batera heltzen zaio eta botoiaren funtzionalitatea kudeatzen du.

5. irudian *KeyboardButton* motako botoiez osatutako teklatu bat agertzen da eta 6. irudian *InlineKeyboardButton* motako botoiez osatutakoa. Botoien kokapena teklatuaren barruan programatzaileak erabakitzen du, hau da, botoi bat gehitzean teklatuari, aurrekoaren ondoan edo hurrengo lerroan kokatu daiteke.

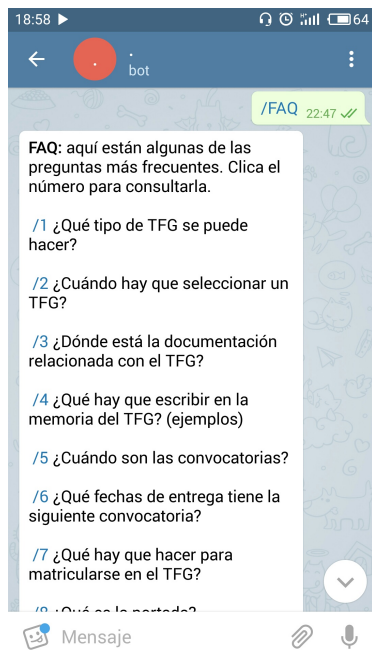
## 8.1 Prototipoen garapena

Hasierako planteamenduan (3. atala) azaldu den bezala, proiektuaren garapena prototipoka egin da, prototipo bakoitzak funtzionalitate berri bat gehituz. Ondoren prototipo bakoitzean egindako garapena azalduko da.

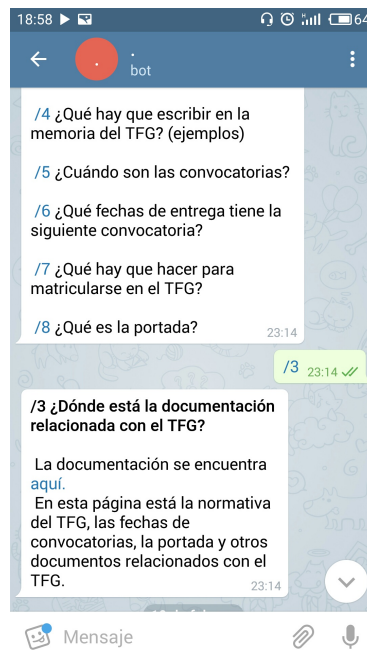
### 8.1.1 1. prototipoa

Lehenengo prototipo honetan, bot-ak maiz egiten diren galderak erantzutea implementatu zen. Horretarako `/faq` komandoa garatu zen. Garapenean aldaketa gehien izan dituen atala izan da, gehienbat programazio lengoiaia eta Telegrameko bot-ek eskaintzen dituzten funtzionalitateak ondo ez ezagutzeagatik.

`/faq` komandoaren funtzionalitatea maiz egiten diren galderen zerrenda bat erakustea da eta erabiltzaileak galdera bat aukeratzen duenean galderaren erantzuz



(a) Galdera zerrenda



(b) Erantzun bat

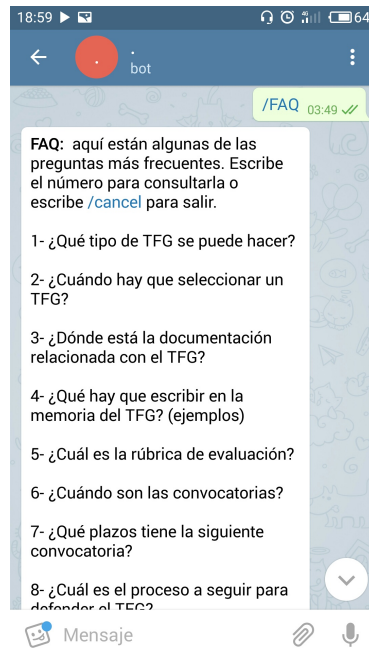
Irudia 7: /faq komandoaren lehen implementazioa, galdera bakoitza komando batean.

nak erakustea. Galdera batek erantzun bat baino gehiago izan ditzake, erantzuna luzea bada eta hainbat mezutan itzuli nahi bada adibidez edo fitxategi ugari bidali nahi badira. Erantzun bezala bot-ak bidali dezakeen fitxategiak hauek dira: argazkiak, bideoak eta dokumentuak.

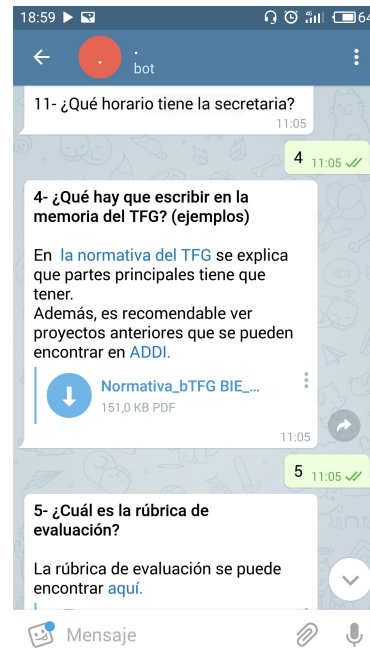
Lehendabizi FAQ zerrenda bat egin zen, arautegia, nik ikasle moduan izan nitzakeen galderak eta Juanan irakasle moduan ezagutzen zituen ohiko galderak kontuan izanik. Behin galderak izanik /faq komandoaren lehen bertsioaren garapena hasi zen.

Lehen saiakeran, galdera bakoitza komando bat bezala tratatu zen, 7. irudian ikusi daitekeen bezala. Komando bat mezu batean agertzen denean klikatu egin daiteke exekutatzeko. Honegatik erabiltzailearentzat nahiko modu eroso da eta programatzen erraza, ez duelako inolako elementu berezirik erabiltzen. Gainera, galdera bakoitzaren erantzuna guztiz kudeatu dezakegu, komando bakoitza fitxategi ezberdin bat baita. Baina honek eragozpen handi bat du, aldaketa bat egitea zailtzen du, fitxategi ugari aldatu beharko litzateke. Proiektuaren helburu bat modifikazioetarako prestatuta egotea denez, soluzio hau alde batera ustea erabaki zen.

Hurrengo saiakeran elkarriketa erabili zen. 8. irudian azaltzen den bezala, /faq



(a) Galdera zerrenda



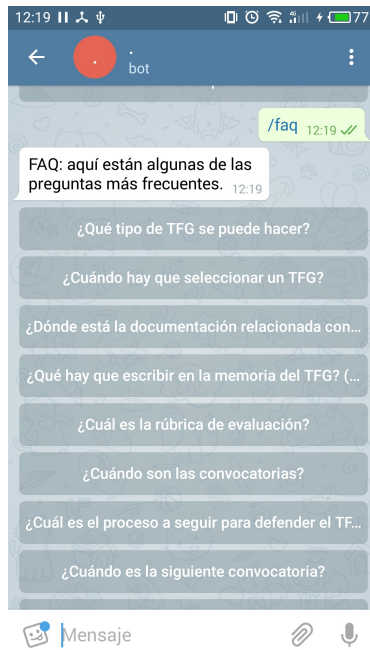
(b) Bi erantzun jarraian

Irudia 8: /faq komandoaren bigarren implementazioa, elkarrizketa erabiliz.

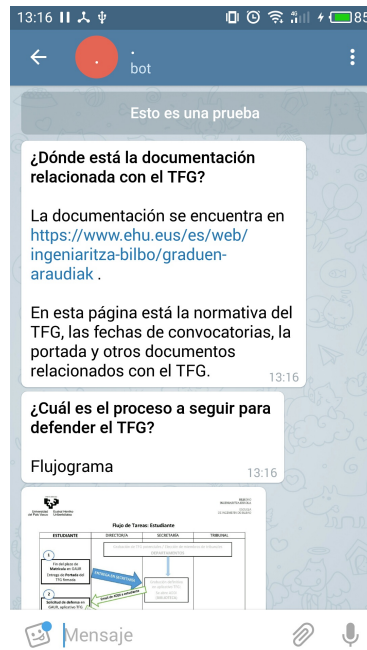
komandoa exekutatzean galdera zerrenda erakusten da eta erabiltzaileari gal-  
detu nahi duen galderaren zenbakia idazteko eskatzen zaio. Erabiltzaileak zen-  
bakia idazten duenean erantzun egiten zaio baina elkarrizketa ez da amaitzen,  
erabiltzaileak beste galdera baten zenbakia idatz dezake edo /cancel komandoa  
erabili elkarrizketa amaitzeko. Modu hau erabiliz komando bakarra dago eta  
galdera bat aldatu edo gehitzeko datu-basean aldaketa eginez nahiko da.

Baina elkarrizketa /cancel komandoaz soilik amaitu daiteke, bot-ak ez du el-  
karrizketa inoiz amaitzen, erabiltzaileak galdera baten zenbakia bidaltzea zain  
gelditzen da. Beste modu bat izango litzake erabiltzaileak galdera baten zenba-  
kia bidaltzen duenean, bot-ak erantzutea eta elkarrizketa amaitzea; baina honek  
eragiten du erabiltzaileak egin nahi duen galdera bakoitzeko /faq komandoa be-  
rriro exekutatu beharra. Horregatik, elkarrizketa behar bada ez da komando  
honetarako soluzio aproposena.

Azken implementazioa *InlineKeyboardButton* erabiliz egin da. 9. irudian azal-  
tzen den bezala, galdera bakoitza botoi bat da eta botoia sakatzerakoan bot-ak  
erantzuna bidaltzen du. Botoi hauek erabiliz, beste bi implementazioetan azal-  
dutako arazoak konpondu dira eta galdera bat aldatu edo gehitzeko datu-basean  
aldaketa eginez nahiko izango da.



(a) Galdera zerrenda



(b) Bot-aren bi erantzun

Irudia 9: /faq komandoaren azken implementazioa, *InlineKeyboardButton* erabiliz.

### 8.1.2 2. prototipoa

Bigarren prototipoan bot-aren erabiltzaileei abisuak modu automatizatuan bidaltzea ahalbidetzen duen kodea implementatu zen. Abisuak bi motatakoak izango dira: deialdien daten abisuak eta abisu bereziak. Deialdien daten abisuen bitartez, datu-basean gordeta dagoen deialdi baten epe bat baino zazpi egun lehenago erabiltzaileei mezu bat bidaliko zaie, deialdiko zein epe eta data adieraziz. Abisu bereziak, administrazio paneletik DBan sartutako mezuak dira eta bertan adierazitako datan bidaliko dira.

Funtzionalitate hau hiru PHP scripten bitartez lortzen da. Script batek deialdien epeak begiratzeko eta epe bat zazpi egunetan amaitzen bada, erabiltzaileko abisu bat sortuko du DBan. Beste script batek berdina egiten du baina abisu bereziekin eta hiru egun falta direnean. Azken scripta bidali gabe dauden eta data egokian dauden abisuak bidaltzeaz arduratzen da.

Abisuen prozesua automatizatzeko cron erabili da. Horretarako crontab fitxategian hiru scriptak aldika exekutatzeko sarrerak gehitu behar dira. Jarraian crontab fitxategian sartu behar diren lerroak ageri dira. Abisuak prestatzeko scriptak orduero exekutatu dira, mezuak bidaltzearena berriz egunean behin

arratsaldeko bostetan.

```
0 * * * * /usr/bin/php /Path/to/broadcast_convocato$
0 * * * * /usr/bin/php /Path/to/broadcast_especial.$
0 17 * * * /usr/bin/php /Path/to/broadcast_enviar.p$
```

### 8.1.3 3. prototipoa

Hirugarren prototipoan GRALak zuzentzeko funtzionalitatea inplementatu da. GRAL bat zuzentzeko erabiltzaileak lehendabizi GRALa erregistratu beharko du bot-ean. Horretarako /registerTFG komandoa dago zeinetan, elkarrizketa baten bitartez, bot-ak GRALaren izenburua eta hizkuntza eskatzen dizkio erabiltzaileari. Ondoren, /addTFG komandoaren bitartez erabiltzaileak GRALaren bertsio berri bat igoko du, fitxategi bat bidaliz. Prozesu honen adibide bat 10. irudian ikusi daiteke.



Irudia 10: GRAL baten erregistroa eta bertsio berri baten bidalketa

Erabiltzaileak bidalitako dokumentua Telegrameko zerbitzarira igotzen da. Beraz, bot-ak Telegrametik fitxategia deskargatu beharko du. Ondoren fitxategiaren mota zuzena dela begiratu behar da, hau da, fitxategiaren luzapena bere benetako media motarekin (*MIME type*), bat datorren. Bot-ak doc, docx, odt eta pdf motako fitxategiak onartzen ditu.



Fitxategiaren mota zuzena bada, bertsio berri dela egiaztatu behar da, hau da, lehenago igondako fitxategi bera ez dela. Horretarako fitxategiari hash funtzio bat - zehazki md5 funtzioa - aplikatzen zaio eta DBan begiratzen da ea existitzen den. Bertsio berria bada, bere informazioa DBan gordeko da eta bot-aren zerbitzarian gordeko da. Gordetako GRAL bertsioak erabiltzaile bakoitzaren Telegrameko identifikatzailea izen moduan duen karpeta baten barruan gordeko da, dagokion GRAL bertsio zenbakiaren barruan. Jarraian honen adibide bat ageri da, non erabiltzailearen identifikatzailea 205754489 da.

```
Files/
├── Download
│   └── 205754489
│       ├── 1
│       │   ├── 0d813d10083a0a5b4a7664cf48899d1e.pdf
│       │   ├── 0d813d10083a0a5b4a7664cf48899d1e.txt
│       │   ├── correction.txt
│       │   └── images
│       │       ├── image-000.ppm
│       │       ├── image-001.ppm
│       │       └── image-002.ppm
│       ├── 2
│       │   ├── 7ce656179369b44cce6fd9c1a958e57e.pdf
│       │   └── 7ce656179369b44cce6fd9c1a958e57e.txt
│       ├── 3
│       │   ├── c71064b21bf75262093eff4cbe5da0fc.odt
│       │   ├── c71064b21bf75262093eff4cbe5da0fc.pdf
│       │   ├── c71064b21bf75262093eff4cbe5da0fc.txt
│       │   └── correction.txt
│       ├── flujograma.jpg
│       └── Normativa_ES.pdf
```

Behin fitxategia daukagula prozesatu egin behar da, zuzenketa egiteko beharrezkoa den informazioa lortzeko.

### Fitxategiaren prozesamendua

Erabili den zuzentzaileak, LanguageTool, testua hartzen du sarrera gisa zuzenketa egiteko. Hortaz, erabiltzaileak bidalitako fitxategitik testua atera behar da. PHP spatie/pdf-to-text paketea erabiliz, pdf-etatik testua atera daiteke. Baina bidalitako fitxategi guztiak ez dira pdf-ak izango. Pdf-ra bihurtzeko gainontzeko fitxategiak unoconv tresnaren bitartez egin da.

Horrela pdf-etako testua lortzen da, baina testu hau ez da oso egokia zuzentzaileari pasatzeko. spatie/pdf-to-text erabiltzean lortzen den testua pdf-an agertzen den ordenean da; orri-oinak, goiburua, orrialde zenbakiak eta horre-

lako elementuak hartzen ditu eta hauek testuaren erdian agertzen dira maiz. Gainera, pdf-eko lerro bakoitzaren ostean lerro-jauzi bat gehitzen du, esaldiak banatuz.

Hori kontuan izanda, testua garbitu beharko da. Helburua testua zuzentzea denez, lortu beharreko testu ahalik eta zuzenen egon beharko da, testu bihurtzaileak sor ditzakeen erroreak minimizatuz, nahiz eta testu originaletik zatiak galdu. Hau egiteko espresio erregularrak erabili dira. Adibidez, `https?:[\ \n][ \n]` espresioa 'http:'-z edo 'https:'-z hasten diren, eta zuriunez edo lerro-jauziz amaitzen diren eta hauen artean zuriunea eta lerro-jauzia ez den edozein karaktere dituen hitzak aurkitzeko balio du, hau da, url-ak aurkitzeko.

Lehendabizi lerro bakoitzaren amaieran gehitutako lerro-jauzia kendu behar da; hau egiteko, lerro-jauzi bakoitzaren ostean minuskulaz hasten bada lerro jauzia kendu da. Ondoren, testuan agertzen diren URLak kendu dira, http-z hasten den hitz oro alegia, hauek errore bezala hartzen baititu zuzentzaileak. Azkenik, zuzenak diren esaldiak hartu dira: maiuskulaz edo esaldi hasierako sinbolo bereziaz (`l,i`) hasten diren, puntuz edo esaldia amaitzeko sinboloz (`?,!`) amaitzen diren eta luzera minimo bat duten esaldiak. Modu honetan goiburuetan eta orri-oinetan agertzen diren testuak ez dira hartzen, normalean lanaren izenburua edo egilearen izen-abizenak agertzen direlako.

Behin testu erabilgarria izanda, zerbitzarian gordeko da eta zuzenketa egin ahal izango da.

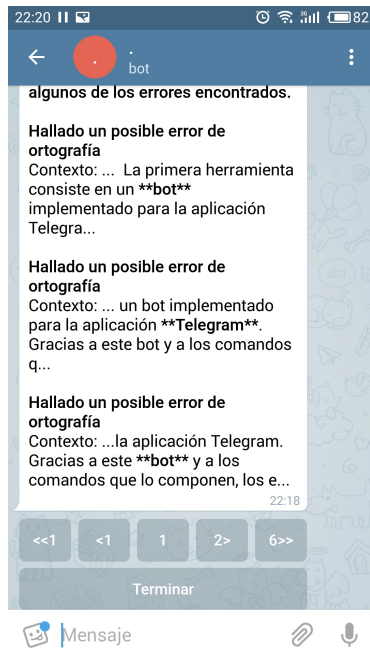
## Zuzenketa

Testuaren zuzenketa egiteko erabiltzaileak `/correctTFG` komandoa exekutatu du. Bot-ak GRALaren azken bertsioa noizkoa den esango dio eta ea hori zuzendu nahi duen. Erabiltzaileak baiezkoa ematean GRALaren testua LanguageTool aplikazioari pasatuko zaio http eskaera baten bitartez. LanguageTool-ek JSON formatuan bidaliko du erantzuna.

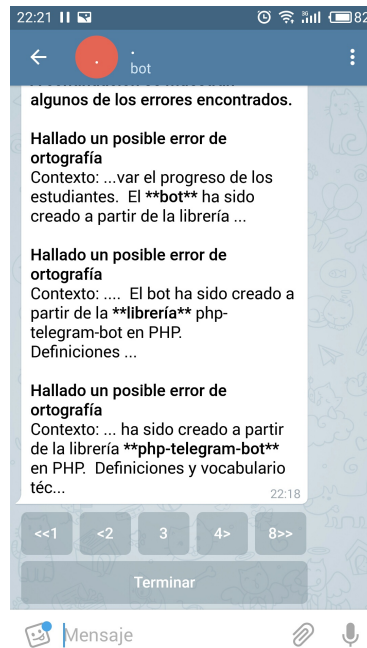
Erabiltzaileak `/correctTFG` komandoa nahi duenean exekuta dezake, baina GRAL bertsioa aldatzen ez denean ez du zentzurik testua berriro LanguageTool-eri bidaltzea, erantzun berdina bidaliko baitu. Horregatik erantzuna DBan gordetzea erabaki zen, baina errore asko aurkitzen dituen erantzunaren tamaina oso handia denez, azkenean testu fitxategi batean zerbitzarian gordetzen da.

Errore kopurua handia izan daitekeenez, erabiltzaileari zatika erakutsiko zaizkio, *pagination* baten bitartez, non orrialdeko elementu kopuru maximoa hiru izango da, gehiago erakutsiz gero mugikorrean ondo ikustea zailtzen baita. Orri batetik bestera pasatzeko *InlineKeyboardButton*en bitartez egiten da. Orrien botoien azpian zuzenketa amaitzeko beste botoi bat dago.

11. irudian erroreen bi orrialde ezberdin ikus daiteke. Botoi bat sakatzerakoan erakusten diren erroreak lehengo mezu berean erakusten dira, hau da, bot-ak ez du mezu berririk bidaltzen, lehengo mezua aldatzen du.



(a) Lehenengo orrialdeko erroreak



(b) Hirugarren orrialdeko erroreak

Irudia 11: Bot-ak GRALaren zuzenketa erakusten du, non erroren orrialde ezberdinak ikus daitekeen mezu berean.

#### 8.1.4 4. prototipoa

Prototipo honen helburua erabiltzaileek bidalitako memorien testua plagioa den ikusten duen komandoa implementatzea zen, baina ez da honetarako software aproposik aurkitu.

Interneten aplikazio ugari daude plagioak bilatzeko baina hauetako asko web bitartez erabiltzekoak dira, ez daukate API egokirik. Gainera, batzuk bot-aren funtzionalitatearen aurkakoa egiten dute, originala den lan bat igotzen da eta aldizka interneten begiratzen da plagiorik dagoen begiratzeko.

Beste batzuk, erantzun anbiguoak ematen dituzte, adibidez, internetetik ateratako testu batek originala den testu batek baino kointzidentzia gutxiago izatea.

Azkenik, egokiak izan daitezkeen erreminta batzuk aurkitu dira, baina hauek erabiltzeko limiteak oso baxuak edota prezioak oso altuak dira. Adibidez, *copy-leaks*<sup>2</sup> aplikazioa dohainik probatzeko hitz limitea 6250 hilabetean da edo *plagscan*<sup>3</sup> aplikazioaren bitartez 20 orri (5000 hitz) aztertzeak 5€-ko prezio du.

<sup>2</sup><https://api.copyleaks.com>

<sup>3</sup><https://www.plagscan.com>

Aipatutako arrazoi guztiak kontuan izanda, funtzionalitate hau ez implementatzea erabaki zen, software egokirik aurkitu ez izanagatik.

### 8.1.5 5. prototipoa

Erabiltzaileek bidalitako memorian agertzen diren irudiak interneten aurkitzen diren ikusten duen komandoa implementatu da prototipo honetan. Funtzionalitate hau erabiltzeko, erabiltzaileak `/imagesTFG` erabili behar du.

Funtzionalitate hau betetzeko lehen pausua dokumentutik irudiak ateratzea da. Irudiak ateratzeko `pdfimages` tresna erabili da. Tresna honen bitartez pdf bateko irudiak atera daitezke eta irudi bakoitza fitxategi batean gordetzen da.

Behin irudiak izanda, irudi bakoitza plagioa izan daitekeen ikusi behar da. Horretarako Googleren Cloud Vision APIa erabili da. HTTP POST eskaera bat egin behar da, irudia eta detektatu nahi den ezaugarriaren informazioa, kasu honetan internet egon daitezkeen antzeko irudiak bilatzea, JSON formatuan bidaliz.

APIak erantzuna JSON formatuan bidaliko du. Erantzunean agertzen diren parametroetatik *webEntities*-en *score* erabiliko da irudiak filtratzeko; *webEntities* batek irudia zein elementurekin lotuta dagoen esaten du, adibidez web-orrialdea, animalia... eta honen *score* elementu hori izateko duen aukera neurtzen du. Probak eginez *score* 0.9 baino handiagoa denean APIak erantzun nahiko ona ematen du.

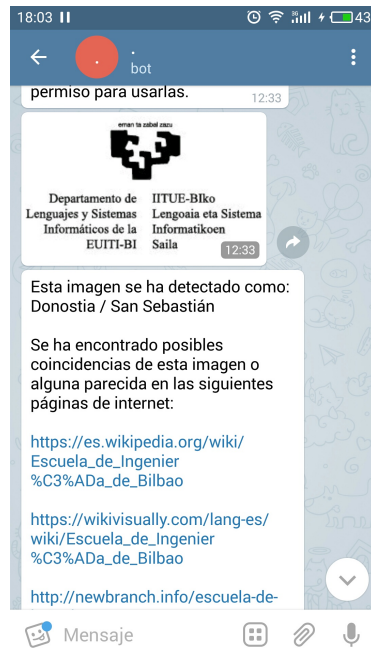
Ondoren erantzunean agertzen diren irudi posibleen url-ak aukeratzen dira eta bot-ak erabiltzaileari irudia eta antzekoak izan daitezkeen irudien url-ak bidaltzen dizkio. 12. irudian komandoaren erantzun baten adibidea ikus daiteke, non irudi bat, bere entitatea eta antzekoak izan daitezkeen irudi batzuen url-ak agertzen dira. Erantzuna oso luzea ez izateko, bidaliko diren irudi kopuru maximoa lau izango da.

Zuzenketan gertatzen den moduan, erabiltzaileak `/imagesTFG` komandoa nahi duenean exekuta dezake, baina GRAL bertsioa aldatzen ez denean ez du zentzurik irudiak berriro aztertzea eta APIari bidaltzea, erantzuna berdina izango baita. Horregatik APIaren erantzuna DBan gordetzen da.

Beste alde batetik, posible da GRALean irudi bat errepikatuta agertzea, adibidez goiburu edota orri-oinetan jartzen direnak. Berdinak diren irudiak ez aztertzeko, DBan irudien hash-a (md5) gordetzen da eta APIari bidali aurretik DBan irudia existitzen den begiratzen da.

## 8.2 Administrazio panela

Datu-basean informazioa sartzeko edo aldatzeko administrazio panel bat egin da. Administrazio panel honen helburua ez da aplikazio osoaren DBa kudeatu



Irudia 12: /imagesTFG komandoa exekutatzean jasotzen den erantzun baten adibidea, non entitatea eta url posible batzuk agertzen diren.

ahal izatea, baizik eta eguneratuta egon behar de informazioa kudeatzea. Panelaren bitartez FAQ galderak eta hauen erantzunak gehitu, aldatu edota ezabatu ahal izango dira, deialdien datak kudeatu eta deialdien abisuak eta abisu bereziak kudeatu.

Administrazio panela Djangoarekin egin da. Django web orrialdeak diseinatzeko erabiltzen da. Baina proiektu honetan bakarrik erabiliko da eskaintzen duen administrazio interfaze automatikoa, helburua informazioa eguneratzea baita eta ez da ezer konplexua egin nahi. Administrazio panela sortzeko jarraitu den prozedura azalduko da.

Administrazio panela egiteko hiru fitxategietan aldaketak egitearekin nahikoa da: settings.py aplikazioaren ezarpenak adierazteko, models.py aplikazioaren datu ereduak gordetzeko eta admin.py ereduak nola erakutsiko diren administrazio panelean erakusteko.

Django instalatuta izan ondoren, egin beharreko lehen pausua proiektu bat sortzea da `django-admin.py startproject ikastenbot_project` exekutatuz. Ondoren, sortutako karpeta berriaren barruan dagoen settings.py fitxategian datu-basearen ezarpenak adierazi behar dira eta `python manage.py startapp ikastenbot` exekutatu aplikazioaren izena adieraziz.

Jarraian models.py fitxategian DBaren taulen ereduak adierazi behar dira, adi-

bidez, jarraian agertzen den zatian *FaqQuestion* taularen ereduak agertzen da. Fitxategi honetan, DBko taularen egitura zehazteaz gain, geroago erakutsiko den informazioa adierazteko funtzioak, atributu bakoitzak izango duen izena edota laguntza testua (*help\_text*) adierazi daiteke.

```
class FaqQuestion(models.Model):
    #DBko taulen zutabeak
        id = models.AutoField(primary_key=True) # AutoField?
        text = models.ForeignKey('MessagesLang', verbose_name="Pregunta",
db_column='text', blank=True, null=True)
        order = models.IntegerField("Orden",help_text="Dejar vacio para
que la pregunta no aparezca en el Bot",blank=True, null=True)

    #informazio ezberdina inpeimatzen laguntzeko metodoak
    def __unicode__(self):
        return self.text.es
    def question_Text_ES(self):
        return self.text.es
    question_Text_ES.short_description = "Pregunta ES"
    def question_Text_EUS(self):
        return self.text.eus
    question_Text_EUS.short_description ="Pregunta EUS"
    class Meta:
        verbose_name = "Pregunta Frecuente"
        verbose_name_plural = "Preguntas Frecuentes"
        managed = False
        db_table = 'faq_question'
```

Normalean models.py fitxategian aldatetak eginez eta *makemigrations* eta *migrate* komandoak erabiliz DBa kudeatu daiteke, models.py sortutako ereduak DBan sortuz. Baina kasu honetan DBa dagoeneko existitzen da eta ez dira makemigrations eta migrate komandoak erabili behar, DBko taulen ereduak adieraztearekin nahikoa da (migrate behin exekutatu behar da Django funtzionatzeko behar dituen taulak sortzeko). Hau egiteko `python manage.py inspectdb` komandoa erabil daiteke, bere irteera models.py idatziz. Komando honek DBko taulak aztertu eta eredu bat sortuko du taula bakoitzeko. Baina aldateta batzuk egitea beharrezkoa da, adibidez auto inkrementalak diren gakoak Integer moduan hartzen ditu baina datu mota zuzena AutoField da. Django DBko taulak ez kudeatzeko `managed = False` erabili behar da ereduak zehaztean.

Behin ereduak zehaztuta daudela, administrazio panelean erakutsi nahi den informazioa zehaztu behar da admin.py fitxategian. Administrazio panelak hiru pantailetan erakusten da ereduaren informazioa: pantaila nagusia, eredu bateko elementuen zerrenda eta elementu baten informazioa aldatzeko pantaila.

Pantaila nagusian, 13. irudia, kudeatu nahi ditugun ereduak agertzen dira.



Irudia 13: Administrazio paneleko pantaila nagusia, non kudeatu daitezkeen ereduak agertzen dira

Hauek pantaila honetan agertzeko admin.py fitxategian honako lerro hauek idatzi behar dira:

```
admin.site.register(Dates,DatesAdmin)
admin.site.register(FaqQuestion,FAQQuestionAdmin)
admin.site.register(MessagesLang,MessagesLangAdmin)
admin.site.register(Notification, NotificationAdmin)
admin.site.register(SpecialNotification,SpecialNotificationAdmin)
```

Goiko kodean agertzen diren bigarren parametroak ez dira beharrezkoak, hurrengo pantailatan erakutsi nahi den informazioa zehazteko erabiltzen da. Parametro bakarra badago, hurrengo pantailatan defektuzko atributuak erakutziko dira.

14. irudian, DBan FAQ galderen taulan agertzen diren elementuen zerrenda agertzen da. Zerrenda honek dituen zutabeak `list_display` parametroan adierazi behar dira. Parametro honetan, ereduko atributuez gain, ereduan definitutako funtzioak erabili daitezke. Jarraian agertzen den kode zatian `list_display` parametroan funtzio bati eta atributu bati erreferentzia egiten zaio.

```
class ResponseInline(admin.StackedInline):
    # galderak modifikaztean erantzunak azpian agertzeko
    model = FaqResponse
    extra = 0

class FAQQuestionAdmin(admin.ModelAdmin):
    # fieldsets = [(None, {'fields': ['order','text']})], aldatu
```



Irudia 14: DBan FAQ galderen taulan agertzen diren elementuen zerrenda.

daitezkeen elementuak

```

readonly_fields = ('text_es', 'text_eus')
def text_es(self, obj):
    return obj.text.es
def text_eus(self, obj):
    return obj.text.eus
inlines = [ResponseInline] #erantzunak erakutsi
list_display = ('question_Text_ES', 'order')

```

admin.py fitxategian ere, DBko taula bateko elementu baten informazioa nola erakutsi adieraz daiteke. Adibidez, `fieldsets` parametroa erabiliz, aldaketa pantailan agertuko diren alda daitezkeen elementuak, eta nola multzokatuta eta zein ordenatan adierazi daiteke. Aldatu ezin daitezkeen elementuak erakutsi ahal dira ere definitu daitezke `readonly_fields` zehaztuz.

15. irudian ikus daitezkeen FAQ galderan bere erantzunak ere ageri dira azpian. Hau gako arrotzei esker egin daiteke. Erantzun bakoitza galdera bakar bati dagokionez, galdera horren aldaketa pantailan agertzea posiblea da, `inlines` parametroa erabiliz.

Gainerako DBko taulekin FAQ galderekin egin den gauza bera egin da, baina askoz ere sinpleagoa, taulako elementuak soilik erakutsi behar direlako, ez da gako arrotzekin ibili behar. Beraz, soilik erabili da `list_display` parametroa, zerrendan agertuko diren zutabeak aldatzeko.



Administración de Django Bienvenido/a, **admin**. [Cambiar contraseña](#) / [Terminar sesión](#)

[Inicio](#) > [Ikastenbot](#) > [Preguntas Frecuentes](#) > ¿Cuál es el proceso a seguir para defender el TFG?

### Modificar Pregunta Frecuente Historico

Pregunta: ID: 12 TAG: faq\_q Text: ¿Cuál es el proceso a seguir para defender el TFG?

Orden: 8  
Dejar vacío para que la pregunta no aparezca en el Bot

Text es: ¿Cuál es el proceso a seguir para defender el TFG?

Text eus: None

**Respuestas**

Respuesta: ¿Cuál es el proceso a seguir para defender el TFG?  Eliminar

Id Texto: ID: 13 TAG: faq\_r Text: Flujograma

Foto: flujograma.jpg

Video:

Documento:

Tipo Fecha:

[+ Agregar Respuesta adicional.](#)

Irudia 15: DBan FAQ galdera baten informazio eta honek izan ditzakeen erantzunen informazioa

### 8.3 Izandako arazo batzuk

Garapenean zehar hainbat arazo sortu dira, gehienak nahiko erraz konpontze-koak, adibidez, bot-ak irudiak bidaltzen dituenean, irudiaren azpian joan daitekeen testuak luzera mugatu du, ondorioz, plagioen irudiaren azpian testua ez zenez sartzen, aparteko mezu batean bidaltzen da informazioa. Baina izandako beste arazo batzuk nahiko zailak izan dira. Jarraian, horietako bi azalduko dira.

#### Administrazio panela eta gako arrotzak

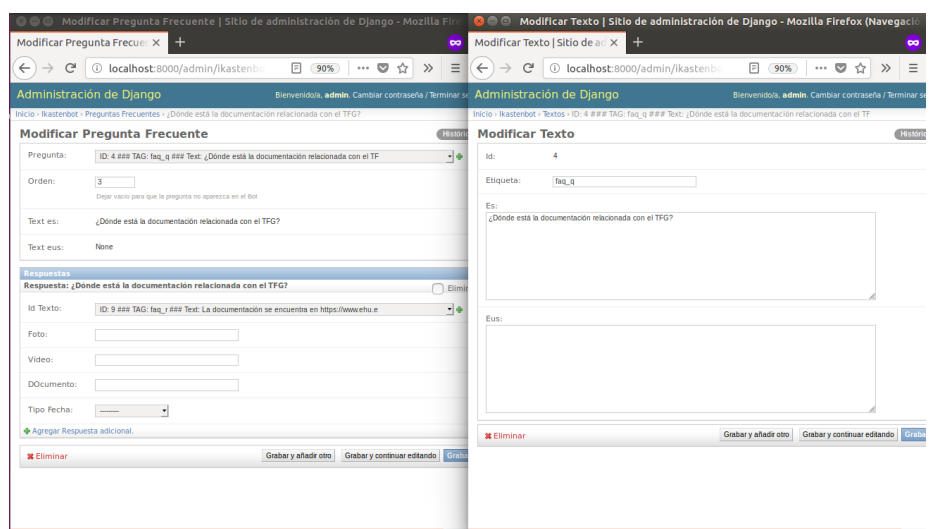
Djangorekin egindako administrazio panela nahiko simplea da. Orokorrean nahiko erraza da administrazio panel bat egitea, baina arazoa gako arrotzak (*ForeignKey* django) erabiltzean sortzen da.

Galdera eta erantzunen testuak beste taula batean gordetzen dira hizkuntzak kudeatzeko, ondorioz galdera eta erantzunek beraien testuaren gako gordetzen dute. Hortaz, galdera eta erantzun bakoitzari testu bat dagokio.

Lehen azaldu den moduan eta 15. irudian ikusi daitekeen bezala, galdera baten azpian honen erantzunak agertzen dira. Erantzun bakoitza galdera bakar bati dagokionez, galdera horren aldaketa pantailan agertzea posiblea da, `inlines` parametroa erabiliz. Baina hizkuntzaren kasuan alderantzizkoa gertatzen da, galdera edo erantzun bakoitzari testu bat dagokio baina testu bat hainbat galdera edo erantzunei lotuta egon daiteke. Hortaz, ezin da testua galdera aldatzeko pantailan aldatu.

Arazo hau konpontzen saiatzen denbora asko erabili da, posibleak zitekeen soluzio ugari probatuz. Administrazio panel erraz bat egiten pentsatutako denbora baino gehiago erabiliz arazo honen ondorioz.

Azkenean lortutako soluziorik aproposena galderen testua irakurketa soilik egiteko erakustea izan da `readonly_fields` parametroan testuen hizkuntza ezberdinetako testuak definituz. Ez da soluzio oso ona baino aurkitutako soluziorik onena da. Testua irakurtzeko soilik denez, aldaketak egin nahi badira, alda-



Irudia 16: Galdera baten testua aldatzeko bi leiho, ezkerrean galdera eta eskuinean testua

ketak egiteko modurik errazena bi leiho aldi berean irekita izatea da: batean aldatu nahi den galdera erakutsiz eta bestean testua aldatzekoa. Hau errazteko, galderan testua agertzen denean bere identifikatzailea agertzen da, horrela errazagoa izango da testuen zerrendan aldatu nahi den testua aukeratzea. 16. irudian honen adibide bat bat ikus daiteke, non ezkerreko leihoan aldatu nahi den galdera agertzen da eta eskuinekoan galderaren testua.

## Testuen kodifikazioa

Gaztelaniazko testua erabiltzean UTF-8 kodifikazioa erabili behar da, bestela ezin izango dira azentudun karaktereak ondo detektatu. Lehendik gaztelaniazko testuekin lan egiteagatik, banekien zer nolako arazoak eman ditzakeen, baina ez PHP lengoaian.

Lehen arazoa PDF-tik testua ateratzean eta espresio erregularrak erabiltzean izan zen. PHPn espresio erregularrak `"/` artean joan behar dute eta hauen amaieran *modifier* bat jarri daiteke, adibidez puntuak lerro jausiak ere kontuan hartu ditzan.

PHPk UTF-8rekin lan egiteko *modifier* bat dauka, `'u'`. *Modifier* hau erabiltzen bada erabilitako espresio erregularra eta espresio erregularra aplikatuko zaion testua UTF-8 moduan tratatzen dira. Beraz, espresio erregularrak `'/<espresioa>/u'` itxura izan behar du.

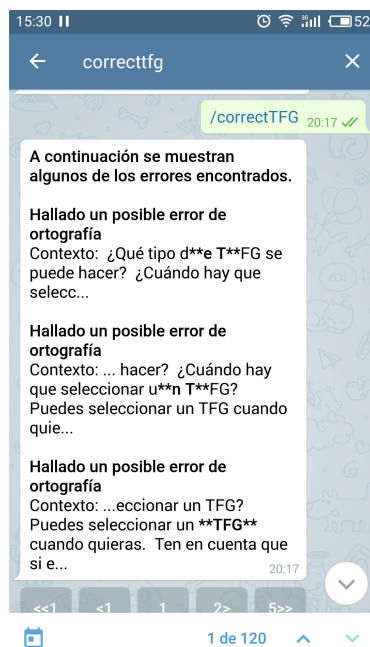
Ez nekienez `'u'` *modifier*-arena hasiera batean pdf-tik lortzen zen testuan karaktere arraroak irteten ziren eta gainera, espresio erregularrak ez ziren ondo aplikatzen.

Kodifikazioarekin izandako beste arazo bat GRALaren zuzenketa erakustean izan da. Erabiltzaileari zuzenketaren mezuan errorea zein hitzetan dagoen markatzen zaio, lehen azaldutako 11. irudian agertzen den bezala. Hitz hau markatzeko zuzentzaileak ematen duen informazioa erabiliz egiten da, testuan errorea non hasten den eta zenbateko luzera duen jakinda.

Errorea markatzeko `**` karaktereak erabili dira eta hauek testuan sartzeko, testu originalari `substr` funtzioa aplikatu zaio, testu originala zatika hartzeko. Hau egitean lortutako emaitza 17. irudian ikusi daitekeena da: errore batzuk ez ditu ondo markatzen, adibidez `**e T**` jartzen du `**TFG**` ordez.

Errore hau gertatzen zen hainbat kasu begiratu ostean, konturatu nintzen `**` karaktereak mugitzen ziren karaktere kopurua, karaktere berezi kopuruaren berdina zela. Hau gertatzen da, PHP-ko `substr` funtzioak byte bakarreko karaktereak hartzen dituelako; hortaz, bi byteko karaktere bat, bi karaktere moduan izango da.

Hau konpontzeko PHP-k eskaintzen duen `mb_substr` funtzioa erabili da. Funtzio hau `substr` funtzioaren berdina egiten du baina multibyte karaktereak kontuan izanda.



Irudia 17: GRALaren zuzenketa, erroreak gaizki markatuta

## 9 Testak

Proiektuaren funtzionalitateak ondo funtzionatzen duten egiaztatzeko hainbat test egin dira. Test bakoitzean prototipo horren funtzionalitateaz gain, lehenagoko prototipoen funtzionalitateak ere ondo daudela egiaztatu da. Hona hemen egindako testak.

### 9.1 1. prototipoa

Lehenengo prototipoan `/faq` komandoa inplementatu da, jarraian egindako probak ageri dira:

<i>/faq komandoa- Galderak</i>
<b>Probaren deskribapena:</b> <code>/faq</code> komandoa bidaltzean bot-ak FAQ galderak erakustea.
<b>Itxarondako emaitza:</b> FAQ galderen zerrenda <i>InlineKeyboradButton</i> botoietan.
<b>Lortutako emaitza:</b> Itxarondako emaitza.

<i>/faq komandoa- Galderen ordena</i>
<b>Probaren deskribapena:</b> DBan galderen ordena aldatu eta galdera bati ordenarik ez ipini.
<b>Itxarondako emaitza:</b> Galderen ordena beste bat izatea eta ordenarik ez duen galdera ez agertzea zerrendan.
<b>Lortutako emaitza:</b> Itxarondako emaitza.

<i>/faq komandoa- Testu erantzuna</i>
<b>Probaren deskribapena:</b> Galdera zerrendatik testua duen galdera bat aukeratu eta bere botoia sakatu.
<b>Itxarondako emaitza:</b> Aukeratutako galderari dagokion testu erantzuna jasotzea.
<b>Lortutako emaitza:</b> Itxarondako emaitza.

<i>/faq komandoa- Argazki erantzuna</i>
<b>Probaren deskribapena:</b> Galdera zerrendatik erantzunean argazkia duen galdera bat aukeratu eta bere botoia sakatu.
<b>Itxarondako emaitza:</b> Aukeratutako galderari dagokion argazkia erantzun bezala jasotzea.
<b>Lortutako emaitza:</b> Itxarondako emaitza.

<i>/faq komandoa- Bideo erantzuna</i>
<b>Probaren deskribapena:</b> Galdera zerrendatik erantzunean bideoa duen galdera bat aukeratu eta bere botoia sakatu.
<b>Itxarondako emaitza:</b> Aukeratutako galderari dagokion bideoa erantzun bezala jasotzea.
<b>Lortutako emaitza:</b> Itxarondako emaitza.

<i>/faq komandoa- Dokumentu erantzuna</i>
<b>Probaren deskribapena:</b> Galdera zerrendatik erantzunean dokumentua duen galdera bat aukeratu eta bere botoia sakatu.
<b>Itxarondako emaitza:</b> Aukeratutako galderari dagokion dokumentua erantzun bezala jasotzea.
<b>Lortutako emaitza:</b> Itxarondako emaitza.

<i>/faq komandoa- Data erantzuna</i>
<b>Probaren deskribapena:</b> Galdera zerrendatik erantzunean data bat duen galdera bat aukeratu eta bere botoia sakatu.
<b>Itxarondako emaitza:</b> Aukeratutako galderari dagokion testua eta data erantzun bezala jasotzea.
<b>Lortutako emaitza:</b> Itxarondako emaitza.

<i>/faq komandoa- Hainbat erantzuna</i>
<b>Probaren deskribapena:</b> Galdera zerrendatik botoi bat baino gehiago sakatu.
<b>Itxarondako emaitza:</b> Aukeratutako galderei dagozkien erantzunak jasotzea.
<b>Lortutako emaitza:</b> Itxarondako emaitza.

## 9.2 2. prototipoa

Bigarren prototipoan abisuak bidaltzeko kodea inplementatu zen. Egindako probak scriptak eskuz eta modu automatikoan Cron-en bitartez exekutatu probatu dira.

<i>Abisuak- Bidali</i>
<b>Probaren deskribapena:</b> DBan bidali gabe dauden abisu batzuk egotea, data ezberdinetakoak, eta abisuak bidaltzeko scripta exekutatu.
<b>Itxarondako emaitza:</b> Bidaltzeko dauden mezuetatik, egunekoak direnak baino ez bidali.
<b>Lortutako emaitza:</b> Itxarondako emaitza.

<i>Abisuak- Datak</i>
<b>Probaren deskribapena:</b> Deialdien daten taulan datuak egotea eta deialdien abisuak prestatzen dituen scripta exekutatu.
<b>Itxarondako emaitza:</b> Epea betetzeko zazpi egun geratzen direnen daten abisuak prestatu.
<b>Lortutako emaitza:</b> Itxarondako emaitza.

<i>Abisuak- Bereziak</i>
<b>Probaren deskribapena:</b> Abisu berezien taulan datuak egotea eta abisu bereziak prestatzen dituen scripta exekutatu.
<b>Itxarondako emaitza:</b> Epea betetzeko baino hiru egun edo gutxiago geratzen diren abisuak prestatu.
<b>Lortutako emaitza:</b> Itxarondako emaitza.

### 9.3 3. prototipoa

Hirugarren prototipoan memoriak zuzentzeko funtzionalitatea implementatu da. Honetarako hiru komando daude: /registerTFG, /addTFG eta /correctTFG.

<i>/registerTFG komandoa- GRAL bat erregistratu</i>
<b>Probaren deskribapena:</b> /registerTFG komandoa exekutatu eta GRAL bat erregistratu izenburua eta hizkuntza zehaztuz.
<b>Itxarondako emaitza:</b> Erregistratutako GRALa DBan agertzen da.
<b>Lortutako emaitza:</b> Itxarondako emaitza.

<i>/addTFG komandoa- GRAL bat erregistratu gabe</i>
<b>Probaren deskribapena:</b> /addTFG komandoa exekutatu lehenago /registerTFG exekutatu gabe.
<b>Itxarondako emaitza:</b> Ezin izango da GRAL baten bertsio bat igo, bot-ak egin beharreko azaldu behar du.
<b>Lortutako emaitza:</b> Itxarondako emaitza.

<i>/addTFG komandoa- Dokumentu egokia bidali.</i>
<b>Probaren deskribapena:</b> /addTFG komandoa exekutatu eta luzapen egokia duen fitxategi bat bidali. Lau fitxategi motarekin probatu: doc, docx, odt eta pdf.
<b>Itxarondako emaitza:</b> Bertsioa DBan erregistratuko da eta bidalitako dokumentuaz aparte, pdf bertsio bat eta fitxategiaren testua duen testu fitxategi bat gordeko dira.
<b>Lortutako emaitza:</b> Itxarondako emaitza.

<i>/addTFG komandoa- Egokia ez den dokumnetu bat bidali.</i>
<b>Probaren deskribapena:</b> /addTFG komandoa exekutatu eta luzapena aldatu zaion fitxategi bat bidali. Adibidez: odt fitxategi bati pdf luzapena jarri. Fitxategi mota eta luzapen ezberdinak erabiliz probatu.
<b>Itxarondako emaitza:</b> Bertsioa ez da gordeko eta erabiltzaileari fitxaegia zuzena ez dela esango zaio.
<b>Lortutako emaitza:</b> Itxarondako emaitza.

<i>/correctTFG komandoa- GRAL bertsiorik ez.</i>
<b>Probaren deskribapena:</b> /correctTFG komandoa exekutatu baina DBan ez izan GRAL bertsiorik.
<b>Itxarondako emaitza:</b> Erabiltzaileari bertsio bat igotzeko eskatuko zaio.
<b>Lortutako emaitza:</b> Itxarondako emaitza.

<i>/correctTFG komandoa- Lehen zuzenketa.</i>
<b>Probaren deskribapena:</b> /correctTFG komandoa exekutatu lehenago zuzendu ez den bertsio batetan.
<b>Itxarondako emaitza:</b> Zuzentzaileak bidalitako erantzuna JSON formatuan zerbitzarian gordeko da eta erabiltzaileari erroreak pantailan erakutsiko zaizkio.
<b>Lortutako emaitza:</b> Itxarondako emaitza.

<i>/correctTFG komandoa- Dagoeneko zuzenduta.</i>
<b>Probaren deskribapena:</b> /correctTFG komandoa exekutatu lehenago zuzendua izan den bertsio batetan.
<b>Itxarondako emaitza:</b> Ez zaio zuzentzaileari deituko, informazioa lehen lortutako fitxategitik hartuko du eta erabiltzaileari erroreak pantailan erakutsiko zaizkio.
<b>Lortutako emaitza:</b> Itxarondako emaitza.

<i>/correctTFG komandoa- Orria aldatu.</i>
<b>Probaren deskribapena:</b> /correctTFG komandoa exekutatu eta erroreak ikusteko mezuan orria aldatzeko botoiak sakatu.
<b>Itxarondako emaitza:</b> Mezua aldatuko da eta beste errore batzuk erakutsiko ditu.
<b>Lortutako emaitza:</b> Itxarondako emaitza.

<i>/correctTFG komandoa- Zuzenketa amaitu.</i>
<b>Probaren deskribapena:</b> /correctTFG komandoa exekutatu eta zuzenketa ikusteko botoia sakatu.
<b>Itxarondako emaitza:</b> Errorrea erakusten duen mezuaren botoiak desagertuko dira.
<b>Lortutako emaitza:</b> Itxarondako emaitza.

## 9.4 5. prototipoa

Prototipo honetan irudiak interneten aurkitzen diren egiaztatzen duen komandoa inplementatu da.

<i>/imagesTFG komandoa- GRAL bertsiorik ez.</i>
<b>Probaren deskribapena:</b> /imagestTFG komandoa exekutatu baina DBan ez izan GRAL bertsiorik.
<b>Itxarondako emaitza:</b> Erabiltzaileari bertsio bat igotzeko eskatuko zaio.
<b>Lortutako emaitza:</b> Itxarondako emaitza.



<i>/imagesTFG komandoa- Lehen exekuzioa.</i>
<b>Probaren deskribapena:</b> /imagesTFG komandoa exekutatu lehenago exekutatu ez den bertsio batetan.
<b>Itxarondako emaitza:</b> Google Cloud Vision APIari deitu zaio irudi bakoitzeko eta datuak DBan gordeko dira. Ondoren erabiltzaileari interneten aurkitutako irudiak bidaliko zaizkio.
<b>Lortutako emaitza:</b> Itxarondako emaitza.
<i>/imagesTFG komandoa- Dagoeneko exekutatu.</i>
<b>Probaren deskribapena:</b> /imagesTFG komandoa exekutatu lehenago exekutatu izan den bertsio batetan.
<b>Itxarondako emaitza:</b> Ez zaio Google Cloud Vision APIari deituko, informazioa DBtik hartuko du eta erabiltzaileari erantzuna erakutsiko zaio.
<b>Lortutako emaitza:</b> Itxarondako emaitza.
<i>/imagesTFG komandoa- Irudirik ez.</i>
<b>Probaren deskribapena:</b> /imagesTFG komandoa exekutatu eta interneten agertzen diren irudirik ez aurkitzea.
<b>Itxarondako emaitza:</b> Erabiltzaileari irudirik aurkitu ez dela esango zaio.
<b>Lortutako emaitza:</b> Itxarondako emaitza.
<i>/imagesTFG komandoa- Lau irudi baino gehiago.</i>
<b>Probaren deskribapena:</b> /imagesTFG komandoa exekutatu eta interneten agertzen diren lau irudi baino gehiago aurkitzea.
<b>Itxarondako emaitza:</b> Erabiltzaileari lau irudi erakutsiko zaizkio.
<b>Lortutako emaitza:</b> Itxarondako emaitza.

## 9.5 Administrazio panela

Administrazio panela Djangok kudeatzen du. Hortaz, atal honetan probatu da panelean egindako edozein aldaketa DBan exekutatzen dela.

<i>Administrazio panela</i>
<b>Probaren deskribapena:</b> Administrazio panelean agertzen den DBko taula bakoitzeko elementuak gehitu, ezabatu eta aldatu.
<b>Itxarondako emaitza:</b> Panelean gehitutako, ezabatutako edota aldatutako informazioa DBan aldatzea.
<b>Lortutako emaitza:</b> Itxarondako emaitza.



## 10 Ondorioak

Azken atal honetan proiektuaren garapenaren eta lortutako emaitzaren hausnarketa egingo da, hasieran planifikatutakoaren konparatuz.

### 10.1 Helburuak

Proiektuaren helburu eta betekizun ia guztiak bete dira. GRALak garatzen laguntzeko bot-a garatu da, nahiko erabilerraza dena, funtzionalitatearekin lortutako komandoekin eta modifikazio eta funtzionalitate berrietarako prestatua. Baina ez da funtzionalitate bat garatu: GRALen testuen plagioak.

Funtzionalitate hori garatu ez izatea tresna egokirik aurkitu ez izanagatik izan da. Tresnak GRAL baten testu bat hartuta internet testuaren zati antzekoak dauden begiratu behar zuen, plagioa den aztertzeko. Aurkitutako tresnak ez zuten erantzun oso egokia edo beraien prezioak nahiko altuak ziren. Hau kontuan izanda, funtzionalitate hori ez garatzea erabaki zen.

### 10.2 Denboraren planifikazioa

Hasieran egindako denboraren planifikazioa eta amaierakoa, ordu kopuruari dagokionez ez dute aldaketari handirik, jarduera batzuetan diferentzia pixka bat handiago da.

3. taulan ikus daitekeen moduan, proiektuaren iraupen totala 367 ordu izatetik 414 ordura pasatu da, planifikatutakoa baino 47 ordu gehiago behar izan dira. Diferentzia honen arrazoi nagusiak garapenean izandako arazoekin lotuta daude, bereziki administrazio panela eta zuzenketak egiteko komandoen garapenarekin, ordu kopuruaren igoera 15 eta 5 hurrenez hurren. Hala ere, inplementazioaren blokearen ordu kopuru totala ez da estimazioaren hain ezberdina 4. prototipoa, plagioarena, ez delako inplementatu. Proporzionalki diferentzia handiena duen jarduera tresnen aukeraketa izan da, estimazioan 10 ordu izatetik errealitatean 20 ordu izatera pasatu da. Honen arrazoi nagusia izan da plagioaren funtzionalitatea garatzeko beharrezkoa zen tresna bat bilatzea.

Beste alde batetik, proiektuaren banaketak denboran zehar aldaketa handiagoa jasan du. Hasiera batean proiektua ekainean amaitzekoa zen. Baina garapenean eta planifikatuta ez nituen betebeharrak pertsonalen ondorioz, proiektuaren amaiera data atzeratu da eta lana amaieran pilatu da.

### 10.3 Etorkizunerako lana

Garatutako proiektua proiektu handiago baten parte denez, etorkizunerako lana oso handia izan daiteke, beste funtzionalitate batzuk gehituz, adibidez, ira-

Taula 3: Jardueren denbora estimazioa eta errealak

JARDUERAK	ESTIMAZIOA	ERREALA
<b>KUDEAKETA</b>	<b>42</b>	<b>49</b>
Bilerak	20	20
Kodearen bertsio kontrola	10	12
Helburuak eta norainokoak definitu	2	2
Tresnen kudeaketa	10	15
<b>IKASKETA</b>	<b>55</b>	<b>60</b>
PHP	10	10
Django	15	15
Telegram Bot	20	20
Aukeratutako APIak	10	15
<b>ANALISI ETA DISEINUA</b>	<b>18</b>	<b>30</b>
Tresnen aukeraketa	10	20
Datu-basearen diseinua	5	7
Betekizunak definitu	3	3
<b>INPLEMENTAZIOA</b>	<b>135</b>	<b>145</b>
1. Prototipoa	25	30
2. Prototipoa	20	15
3. Prototipoa	30	35
4. Prototipoa	10	-
5. Prototipoa	20	20
Administrazio panela	30	45
<b>TESTAK</b>	<b>27</b>	<b>25</b>
1. Prototipoa	3	2
2. Prototipoa	5	7
3. Prototipoa	5	6
4. Prototipoa	3	-
5. Prototipoa	8	7
Administrazio panela	3	3
<b>DOKUMENTAZIOA</b>	<b>90</b>	<b>105</b>
Memoria	75	90
Aurkezpena	15	15
<b>GUZTIRA</b>	<b>367</b>	<b>414</b>

kasleen tutoretzen orduak kontsultatzea edo memoriak izan behar dituen atal guztiak izatea. Baina proiektu honetan garatutako funtzionalitate batzuk ere hobetu daitezke. Jarraian hobekuntza batzuk azalduko dira:

- **Hizkuntza:** Hizkuntzak kudeatzeko DBan beharrezko dena implementatu da, baita DBari egiten zaizkio dei guztiak erabiltzailearen hizkuntza dute kontuan, baina ez da hizkuntza aldatzea posible, ez zen proiektuaren helburua.
- **Zuzenketa:** Zuzenketa egitean batzuetan errore bezala markatzen da ondo dagoen hitz bat, adibidez ingelesez dagoelako edo siglak direlako. Erabiltzaile bakoitzak bere GRALerako salbuespenak gorde ahal izatea eta hauek kudeatzea funtzionalitate interesgarria izan daiteke.
- **Testuaren plagioa:** Proiektu honetan garatu ez den atala da. Proiektuaren planteamenduaren arabera, dagoeneko existitzen diren tresnak bot-ak bateratzea zen helburua. Baina tresna egokirik aurkitu ez denez soluzio posible bat tresna sortzea izan daiteke eta internet guztian bilatu beharrean, bilaketa ADDI-ko memorieta mugatu.
- **Pdf-etatik testua atera:** Pdf-etatik testua ateratzen denean, iragazki generiko batzuetatik pasatzen da testua garbitzeko, baina bidean informazioa galtzen da. Garbiketa prozesu hau nahiko generikoa da, GRAL formatu ezberdinen ondorioz. Baina behar bada posible litzake iragazki espezifikoagoak egitea.

## 10.4 Hausnarketa pertsonala

Proiekturen memoriarekin amaitzeko hausnarketa pertsonal txiki bat egingo da.

Proiektuaren garapenaren lortutako aplikazioa behar bada ez da erabat zuzena izango, baina kontuan izanda proiektuaren hasieran ez nituela erabilitako tresna gehienak ezagutzen, emaitza oso ona lortu da, guztiz funtzionala den bot-a garatzea lortu da. Proiektuaren alde honekin pozik nago eta teknologia berri nahiko erabiltzen ikasi dut.

Beste alde batetik, proiektuaren garapen honekin lehendik nekien zerbait egiaz-tatu dut: neure buruari jartzen dizkiodan epeak betetzen ez naiz oso ona. As-kotan gertatzen zait pertsonala den proiektu bat garatzean epeak ez ditudala errespetatzen, beti egiten dut dena azken momentuan. Baina beste norbaite-kin lan egitean epeak errespetatzen ditut; beraz, proiektua bakarka garatzearen esperientzia, hurrengo proiektuetan jarraitu behar ez dudan prozeduraren adibidea izatea espero dut.

Azkenik, nahiz eta hobetzeko alderdi batzuk egon, proiektua garatzearen espe-rientzia ona izan da, eta gehienbat, ikasteko balio izan dit.



## Erreferentziak

[Azzopardi and Maxwell, 2014] Azzopardi, L. and Maxwell, D. (2014). *Tango With Django*. [tangowithdjango.com](http://tangowithdjango.com). A beginner's guide to web development with Python 2.7 / Django 1.7.

[Telegram, 2018] Telegram (2018). Telegram Bot API. <https://core.telegram.org/bots/api>. [Telegram Bot API 3.6, 2018-07-20].





## A I. Eranskina: Instalazio gida

Eranskin honetan bot-a eta administrazio panela instalatzeko jarraitu beharreko pausuak azalduko dira. Instalazioa Ubuntu zerbitzari batean egin da.

### A.1 Aurrebaldintzak

Bot-a eta administrazio panela instalatu ahal izateko, honako programa hauek instalatuta izatea beharrezkoa da:

- **PHP 7:** Bot-ak behar duen PHP bertsioa da. `sudo apt-get install php7.0` exekutatuz instala daiteke.
- **Composer:** Bot-aren kodearen dependentziak kudeatzeko. `sudo apt install composer` exekutatuz instala daiteke.
- **unzip:** Composer erabiltzeko beharrezkoa izango da. `sudo apt install unzip` exekutatuz instala daiteke.
- **Git:** Errepositorioak klonatzeko erabiliko da.
- **Java:** LanguageTool erabiltzeko beharrezkoa izango da JDK instalatuta izatea.
- **Maven:** Java proiektuak kudeatzeko tresna, LanguageTool erabiltzeko beharrezkoa. `sudo apt install maven` exekutatuz instala daiteke.
- **MySQL:** DBa kudeatzeko erabili den softwarea.
- **Python 2.7+:** Administrazio panelerako beharrezkoa, gutxienez 2.7 bertsioa.
- **Pip:** Pythonen liburutegiak kudeatzeko. `sudo apt install python-pip` exekutatuz instala daiteke.
- **Telegram:** Bot-a sortzeko beharrezkoa izango da.
- **Google kontu bat:** Google Cloud Vision APIa erabiltzeko Google kontu bat erabili behar da.

### A.2 Bot-aren instalazioa

Lehendabizi Bitbucketeko proiektua klonatu behar da:

```
git clone https://amaiadp@bitbucket.org/amaiadp/bot_tfg.git
```

Proiektua klonatu ostean bot\_tfg karpetara mugituko gara. Ondoren proiektuaren dependentziak instalatu behar dira baina hori egin aurretik php-ren zenbait liburutegi behar dira:

```
sudo apt-get install php-curl
```

```
sudo apt-get install php-mbstring
```

```
sudo apt-get install php-mysql
```

Proiektuaren dependentziak instalatu:

```
composer update
```

DBikastenbot.php fitxategia mugitu:

```
mv DBikastenbot.php vendor/longman/telegram-bot/src/Commands/DBikastenbot.php
```

Telegram bot bat sortu @BotFather bot-arekin hitz eginenez Telegram bitartez eta sortutako bot-aren api-key eta bot-aren username config.php fitxategian idatzi.

Google Cloud proiektu berri bat sortu, Cloud Vision APIa aktibatu eta Kreditzialak sortu. Lortutako API Key-a config.php fitxategian idatzi.

Ondoren, bot-ak erabiltzen dituen beste tresna batzuk instalatuko dira.

PDF-ekin lan egiteko tresnen instalazioa:

```
sudo apt-get install poppler-utils
```

```
sudo apt-get install unoconv
```

## LanguageTool

LanguageTool-en zerbitzaria instalatzeko honako pausu hauek jarraitu behar dira.

Lehendabizi proiektua klonatu behar da:

```
git clone --depth 5 https://github.com/language-tool-org/language-tool.git
```

JDK ez badago instalatu:

```
sudo apt-get install openjdk-8-jdk
```

```
JAVA_HOME="/usr/lib/jvm/java-1.8.0-openjdk-amd64"
```

```
export JAVA_HOME
```

Proiektuaren root karpetatik:

```
./build.sh languagetool-standalone package -DskipTests
```

Zerbitzaria martxan jartzeko honako agindu hau erabili beharko da:

```
java -cp languagetool-standalone/target/LanguageTool-4.3-SNAPSHOT/  
LanguageTool-4.3-SNAPSHOT/languagetool-server.jar
```

### A.3 Datu-basea

Bot-aren kodearekin batera sql fitxategi bat dago, ikastenbot.sql. Fitxategi hau MySQL zerbitzarira inportatu behar da komando honen bitartez (zure DBko datuak ipini):

```
mysql -u <username> -p <databasename> < ikastenbot.sql
```

Ondoren, datu-basearen konexiorako beharrezko datuak config.php fitxategian idatzi behar dira.

### A.4 Administrazio panela

Administrazio panela instalatzeko pausuak hauek dira:

Django instalatu:

```
pip install -U django==1.7
```

Proiektua klonatu:

```
git clone https://amaiadp@bitbucket.org/amaiadp/adminpanel.git
```

Liburutegi hauek instalatu:

```
sudo apt-get install python-dev libmysqlclient-dev
```

```
pip install mysqlclient
```

Bot-aren datu-basearen konexiorako beharrezko datuak settings.py fitxategian idatzi.

Zerbitzaria martxan jartzeko honako agindu hau erabili beharko da:

```
python manage.py runserver 0.0.0.0:8000
```

Administrazio panela honelako url batean egongo da (zerbitzariaren url-aren arabera): <http://127.0.0.1:8000/admin>

Defektuzko erabiltzaile izena eta pasahitza: admin, admin.



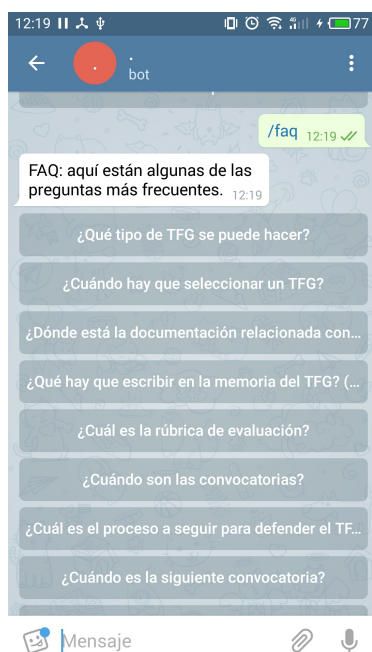
## B II. Eranskina: Bot-aren erabiltzaile gida

Eraskin honetan bot-ak eskaintzen dituen funtzionalitateak nola erabili azalduko da.

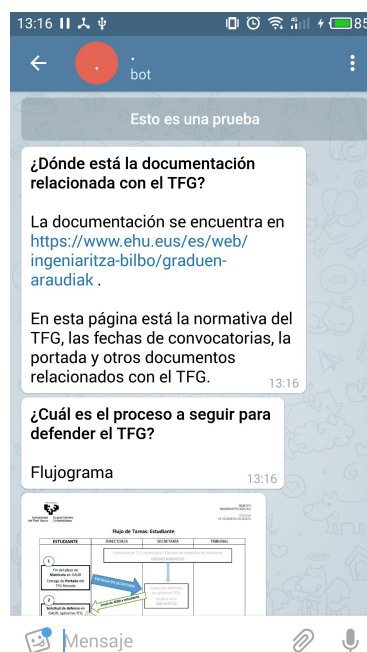
Lehendabizi bot-arekin Telegram aplikazioan elkarrizketa bat hasi behar da. Komando bat exekutatzeko nahikoa da hau mezu bezala bidaltzea.

### FAQ galderak

Bot-ak maiz egiten diren galderak erantzuten ditu. Horretarako /faq komandoa erabili daiteke. Komandoa exekutatzean galderen zerrenda bat lortuko da, galdera bakoitza botoi bat da. Botoi bat sakatzean galdera horren erantzuna bidaliko da. Erantzunean argazkiak, bideoak edo dokumentuak jaso daitezke. Honen adibide bat ikus daiteke 18. irudian.



(a) Galdera zerrenda



(b) Bi erantzun jarraian

Irudia 18: /faq komandoaren adibidea.

### GRAL bat erregistratu

Erabiltzaile bakoitzak GRAL bakar bat erregistratu dezake /registerTFG komandoaren bitartez. GRAL bat erregistratua izate beharrezko izango da gai-

nontzeko funtzionalitateak erabili ahal izateko.

GRAL bat erregistratzean GRALaren izenburua eta hizkuntza zehaztu behar da. Hizkuntza gaztelania (es) edo euskara (eus) aukeratu daiteke baina funtzionalitate batzuk soilik daude gaztelaniazko GRALentzat. 19. irudian GRAL baten erregistroaren adibide bat ageri da.



Irudia 19: GRAL baten erregistroa eta bertsio berri baten bidalketa

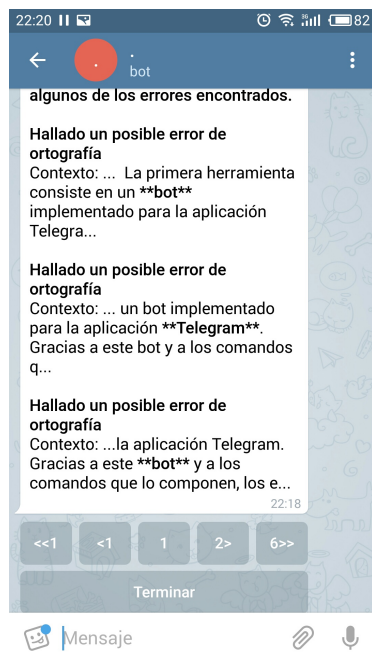
### **GRAL bertsio bat bidali**

GRAL memoria idazten joan ahala, memoriaren bertsio ezberdinak bidali ahal zaizkio bot-ari. Horretarako /addTFG erabili daiteke. Bidaliko den fitxategia doc, docx, odt edo pdf formatuetan egon beharko du. Gogoratu /addTFG komandoa erabili ahal izateko GRAL bat erregistratuta eduki behar dela. 19. irudian /addTFG komandoaren adibide bat ikus daiteke.

### **Memoriaren zuzenketa**

/correctTFG komandoaren bitartez /addTFG komandoaren bitartez igotako azken memoriaren bertsioa zuzenduko da. Funtzionalitate hau soilik dago dago gaztelaniazko GRALentzat prestatuta. Testua zuzentzen pixka bat tardatzen

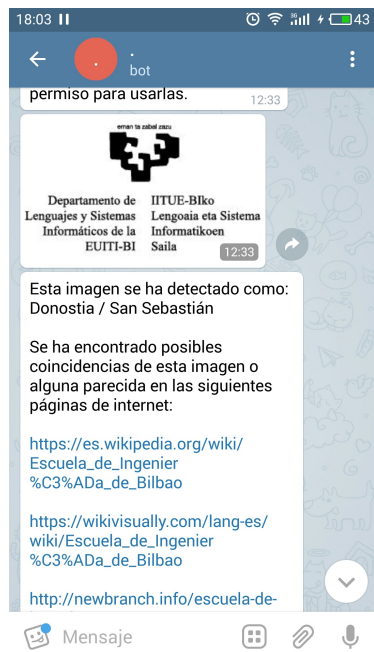
duenez, erantzuna ere pixka bat tardako du. Erantzunean hiru errorealdi berean ikusi daitezke eta beste errore batzuk ikusteko, zenbakidun botoiak sakatu behar dira. Botoi hauek orrialdeak pasatzea moduan da, botoia sakatzean mezuaren testua aldatuko da. Erroreak ikusten amaitzean, "Terminar" botoia sakatu behar da. 20. irudian erroreak nola erakusten diren pantaila ikusi daiteke.



Irudia 20: GRAL baten erroreak ikusteko pantaila

## Irudiak aztertu

Bot-ak memorian agertzen diren irudiak originalak diren aztertu dezake. Horretarako /imagesTFG komandoa erabiltzen da. Komando honen bitartez, bidalitako GRALaren azken bertsioiko irudiak interneten agertzen diren aztertuko du. Irudiren bat interneten agertzen dela detektatzen bada, zein irudi den eta zein url-an aurkitu den jasoko da erantzun gisa. Kontuan izan positibo faltsuak gerta daitezkeela. 21. irudian interneten aurkitu den irudi baten erantzuna nolakoa den ikusi daiteke.



Irudia 21: Interneten aurkitu den irudia



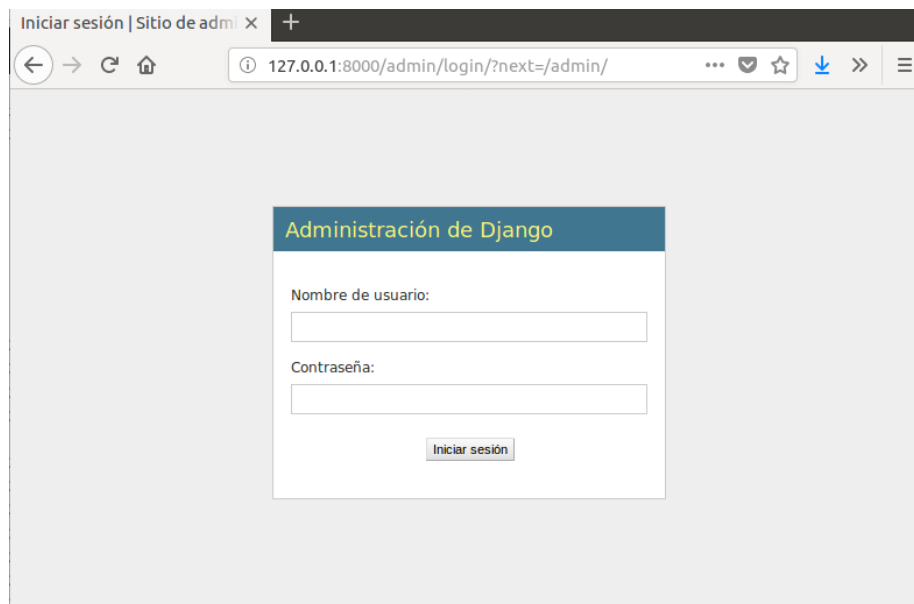
## C III. Eranskina: Administrazio panelaren erabiltzaile gida

Eranskin honetan administrazio panela erabiltzeko zenbait argibide emango dira.

Administrazio panelak, login pantailaz gain, hiru pantaila ezberdin ditu: pantaila nagusia, elementuen zerrenda eta elementu bat aldatzeko edo gehitzeko pantaila.

### Login

Lehen pausua administrazio panela erabiltzeko 22. irudian ikusten den pantailan logeatzea izango da. Aldaketarik egin ez bada, logeatzeko erabiltzaile izena eta pasahitza "admin" eta "admin" dira.



Irudia 22: Administrazio paneleko login pantaila

### Pantaila nagusia

Pantaila honetan kudeatu nahi den elementu mota aukeratu behar da. 23. irudian ikusten den bezala, aukeratu daitezkeen elementu motak hauek dira: deialdien datak, abisuen erregistroa, abisu bereziak, FAQ galderak eta galdera

eta erantzunen testuak. Hauetako baten izenean klikatuz gero, elementu mota horren elementuen zerrendara agertuko da.



Irudia 23: Administrazio paneleko pantaila nagusia

## Elementuen zerrenda

Pantaila nagusian agertzen den elementu mota bakoitzeko, honelako pantaila bat dago. 24. irudian FAQ elementuen zerrendaren pantaila ageri da. Elementuak taula batean agertzen dira eta zutabeen izena klikatzean, elementuen ordena alda daiteke.

Ezker aldean elementuak aukeratzeko checkbox bat ageri da. Elementuak aukeratzeko badira, goian agertzen den menu zabalgarriaren bitartez ezabatu egin daitezke. Elementu berri bat sortzeko goian eskuinean dagoen botoia erabili behar da. Elementu bat aldatzeko, elementu horren urdinez agertzen den testua klikatu behar da. Elementu bat aldatzeko edo gehietzeko pantaila bera erabiltzen da.

## Galdera aldatu/gehitu

Pantaila honetan galdera baten informazioa eta beraren erantzunena aldatu daiteke. 25. irudian pantaila hau bi zatitan banatzen dela ikus daiteke. Goiko aldean galderaren informazioa agertzen da, galderaren testua zein den eta zein ordenetan agertu behar den zehaztuz. Ordena hutsik uzten bada galdera ez da bot-ean agertuko.

Pantailaren beheko aldean, galderaren erantzunak kudeatzeko da. Bertan galderen informazioa ageri da, beraien testua zein den eta erantzunean dokumentu, argazki edo bideorik bidali behar den zehazten da. Gainera, hurrengo deialdiko epe baten data bidali nahi bada erantzun gisa, dataren menu zabalgarriaren bitartez aukeratu daiteke; kontuan izan data hau testuaren amaieran gehitzen dela.

Administración de Django Bienvenido/a, **admin**. [Cambiar contraseña](#) / [Terminar sesión](#)

[Inicio](#) > [Ikastenbot](#) > Preguntas Frecuentes

### Escoja Pregunta Frecuente a modificar Añadir Pregunta Frecuente +

Acción:  Ir seleccionados 0 de 9

<input type="checkbox"/> Pregunta ES	Orden
<input type="checkbox"/> ¿Qué tipo de TFG se puede hacer?	1
<input type="checkbox"/> ¿Cuándo hay que seleccionar un TFG?	2
<input type="checkbox"/> ¿Dónde está la documentación relacionada con el TFG?	3
<input type="checkbox"/> ¿Qué hay que escribir en la memoria del TFG? (ejemplos)	4
<input type="checkbox"/> ¿Cuál es la rúbrica de evaluación?	5
<input type="checkbox"/> ¿Cuándo son las convocatorias?	6
<input type="checkbox"/> ¿Cuál es el proceso a seguir para defender el TFG?	8
<input type="checkbox"/> ¿Cuándo es la siguiente convocatoria?	12
<input type="checkbox"/> Esto es una prueba	100

9 Preguntas Frecuentes

Irudia 24: FAQ galderen zerrenda

## Data aldatu/gehitu

26. irudian, deialdi baten epe baten data aldatzeko pantaila agertzen da. Pantaila honetan deialdiaren epea noiztik nora den eta deialdiaren zein epe den zehazten da. Gainera, daten abisua ematen direnez, agertzen den checkbox-a abisua bidali den jakiteko da.

## Testua aldatu/gehitu

Galdera eta erantzunen testua kudeatzeko pantaila. 27. irudian ikus daitekeen moduan, testu bakoitzak zenbakizko identifikatzaile bat, etiketa bat eta testua gaztelaniaz eta euskaraz gordetzeko esparruak ditu.

## Abisu bereziak aldatu/gehitu

Deialdien datez gain beste abisu bat bidali nahi bada, 28. irudian agertzen den pantailaren bitartez egin daiteke. Abisu berri bat sortzeko nahikoa da zein datatan eta bidali nahi den mezua gaztelaniaz zein euskaraz zehaztea. Pantailan agertzen den checkboxa, bidaltzeko prest dauden abisua prest ez daudenetatik ezberdintzeko da.

Administración de Django Bienvenido/a, **admin**. [Cambiar contraseña](#) / [Terminar sesión](#)

[Inicio](#) > [Ikastenbot](#) > [Preguntas Frecuentes](#) > ¿Dónde está la documentación relacionada con el TFG?

### Modificar Pregunta Frecuente Historico

Pregunta:	ID: 4 ### TAG: faq_q ### Text: ¿Dónde está la documentación relacionada con el TF	<input style="float: right;" type="button" value="+"/>
Orden:	<input type="text" value="3"/>	<small>Dejar vacío para que la pregunta no aparezca en el Bot</small>
Text es:	¿Dónde está la documentación relacionada con el TFG?	
Text eus:	None	

#### Respuestas

Respuesta: ¿Dónde está la documentación relacionada con el TFG?  Eliminar

Id Texto:	ID: 9 ### TAG: faq_r ### Text: La documentación se encuentra en https://www.ehu.e	<input style="float: right;" type="button" value="+"/>
Foto:	<input type="text"/>	
Vídeo:	<input type="text"/>	
DOcumento:	<input type="text"/>	
Tipo Fecha:	<input type="text" value="-----"/>	

Agregar Respuesta adicional.

Irudia 25: FAQ galdera bat gehitu/aldatu

### Abisuak aldatu/gehitu

Pantaila honetan abisu baten informazioa alda daiteke. Abisu batek honako informazioa du: norentzat den abisua, noiz bidaliko den edo bidali den, eta zein da abisuaren testua. Gainera bidalita dauden abisuak bidalita ez daudenetatik ezberdintzeko checkbox bat dago. Kontuan izan bidalita dagoen abisu batean aldaketak egiten badira, bot-aren erabiltzaileari ez zaiola aldaketa hori erakutsiko.

Administración de Django Bienvenido/a, **admin**. [Cambiar contraseña](#) / [Terminar sesión](#)

Inicio > Ikastenbot > Fechas de convocatorias > matricula

### Modificar Fecha de convocatoria Historico

Fecha inicio:  Hoy

Nota: Usted esta a 2 horas antes de la hora del servidor.

Fecha fin:  Hoy

Nota: Usted esta a 2 horas antes de la hora del servidor.

Qué:    
Selecciona una opción

Notificación creada para el envío

[✖ Eliminar](#) [Grabar y añadir otro](#) [Grabar y continuar editando](#) [Grabar](#)

Irudia 26: Deialdi baten data bat gehitu/aldatu

Administración de Django Bienvenido/a, **admin**. [Cambiar contraseña](#) / [Terminar sesión](#)

Inicio > Ikastenbot > Textos > ID: 1 ### TAG: faq ### Text: FAQ: aquí están algunas de las preguntas más frecu

### Modificar Texto Historico

Id:

Etiqueta:

Es:

Eus:

[✖ Eliminar](#) [Grabar y añadir otro](#) [Grabar y continuar editando](#) [Grabar](#)

Irudia 27: Testu bat gehitu/aldatu

Administración de Django Bienvenido/a, **admin**. [Cambiar contraseña](#) / [Terminar sesión](#)

[Inicio](#) > [Ikastenbot](#) > [Notificaciones Especiales](#) > intento de notificación especial

### Modificar Notificación Especial Historico

Fecha:  [Hoy](#) | [📅](#)  
Nota: Usted esta a 2 horas antes de la hora del servidor.

Notificación preparada

Texto ES:

Texto EUS:

[✖ Eliminar](#) [Grabar y añadir otro](#) [Grabar y continuar editando](#) [Grabar](#)

Irudia 28: Abisu berezi bat gehitu/aldatu

Administración de Django Bienvenido/a, **admin**. [Cambiar contraseña](#) / [Terminar sesión](#)

[Inicio](#) > [Ikastenbot](#) > [Notificaciones \(log\)](#) > 2018-04-17

### Modificar Notificación (log) Historico

ID usuario:  ▼

Fecha:  [Hoy](#) | [📅](#)  
Nota: Usted esta a 2 horas antes de la hora del servidor.

Mensaje:

Enviado

[✖ Eliminar](#) [Grabar y añadir otro](#) [Grabar y continuar editando](#) [Grabar](#)

Irudia 29: Abisu bat gehitu/aldatu