



Universidad del País Vasco Euskal Herriko Unibertsitatea

# Informatika Fakultatea

## Informatika Ingeniaritzako Gradua

### ■ Gradu Amaierako Proiektua ■

Software Ingeniaritza

InfoTulist: informazio turistikoa kudeatzeko webgune adimentsua.

*[www.InfoTulist.eus](http://www.InfoTulist.eus)*

---

Mikel Urkiaga Aburruza

2016 - ekaina





# Laburpena

---

Euskal Autonomi Erkidegoko turismo informazioa bildu dezaken webgunea inplementatu da. Informazio bulegoko langile eta edozein interesatuk erabil dezaken webgunea, zeinek erabiltzaileen datuak analizatzen dituen gero informazio pertsonalizatua erakusteko.

EAEko edozein herrirentzat izan daiteke informazio turistikoa eskeintzen duen webgunea beren administrari eta guzti. Gainera erabiltzaileek beren *feedbacka* eman dezakete toki nahiz jaietan.

Inplementazioa burutzeko Django lan ingurune aurreratua erabili da eta honekin batera datu meatzaritzako zenbait teknika aplikatu dira erabiltzaile bakoitzari bere gustukoa izan daitekeen informazioa erakusteko.

# Gaien Aurkibidea

---

Laburpena.....	i
Gaien aurkibidea.....	ii
<b>1. Proiektuaren aurkezpena .....</b>	<b>1</b>
1.1- Sarrera.....	1
1.2- Dokumentuaren egitura.....	2
1.3- Ideia eta motibazioa.....	3
1.4- Aurrekariak.....	3
1.4.1- Antzeko sistemak.....	4
1.4.2- Proposatutako soluzioa.....	6
1.4.3- Aurrekariak vs soluzioa.....	9
<b>2. Kudeaketa plana .....</b>	<b>12</b>
2.1- Helburuak.....	12
2.2- Irismena.....	13
2.2.1- Betekizunak.....	13
2.2.2- Mugak.....	14
2.2.3- Produktuaren irismena.....	15
2.2.4- Irismen mailak.....	15
2.2.5- Irismenaren kudeaketa.....	16
2.2.6- LDEa.....	20
2.3- Lan metodologia.....	20
2.4- Emangarri nagusiak.....	21
2.5- Baliabideak.....	22
2.6- Proiektua gauzatzeko plana.....	23
2.6.1- Egutegia.....	23
2.6.2- Gantt diagrama.....	23
2.6.3- Lan karga.....	25
2.7- Kalitate plana.....	25
2.7.1- Gutxieneko kalitatea.....	25
2.7.2- Kalitatearen adierazleak.....	26
2.7.3- Kalitatea ziurtatzea eta kontrola.....	26
2.8- Arriskuak.....	27
2.9- Parte interesatuak.....	28
2.9.1- Komunikazio plana.....	29
<b>3. Erabilitako teknologiak .....</b>	<b>30</b>
3.1- Python.....	30
3.2- Django.....	31
3.3- Datu baseak.....	36

<b>4. Sistemaren funtzionamendua. Betekizunen azterketa egoera erralak erabilia.....</b>	<b>38</b>
4.1- Kasu praktikoa 1: Erabiltzaile berriak egin beharreko lehen pausuak sisteman.....	38
4.2- Kasu praktikoa 2: Familia bat turismo bulegora informazio eske.....	41
4.3- Kasu praktikoa 3: Bizkaitar bat Donostiara hiria ezagutzen.....	47
4.4- Kasu praktikoa 4: Herri txiki bateko kanpin arduraduna inguruko tokiak 'saltzen'.....	50
4.5- Kasu praktikoa 5: Gazte talde bat aste buruan zer egin jakin gabe.....	51
4.6- Kasu praktikoa 6: Turismo bulegoko administrariaren ikuspuntutik.....	52
<b>5. Analisi eta diseinua.....</b>	<b>55</b>
5.1- Erabilpen kasuak.....	55
5.1.1- Lehen rola: Erabiltzaile ez erregistratua.....	55
5.1.2- Bigarren rola: Erabiltzaile erregistratua.....	56
5.1.3- Hirugarren rola: Erabiltzaile administraria.....	57
5.1.4- Laugarren rola: Web administraria.....	57
5.2- Klase diagrama.....	60
5.3- Sekuentzi diagramak.....	62
5.3.1- Tokia ikusi / iritzia idatzi.....	62
5.3.2- Hasiera orriko kontsultaren.....	64
5.3.3- Plana sortu.....	67
5.3.4- Norbere profila kontsultatu.....	69
5.3.5- Bilaketa data, herri edota gertutasunarekiko... .	70
5.3.6- Bilaketa herrian.....	72
<b>6. Inplementazioa.....</b>	<b>75</b>
6.1- Django aplikazioa.....	75
6.1.1- Datu basea.....	75
6.1.2- Administrazio panela.....	77
6.1.3- Txantiloiak.....	81
6.1.4- Erregistroa.....	82
6.1.5- Google Maps.....	85
6.1.6- Ajax deiak.....	86
6.1.7- Hizkuntza ezberdinak.....	88
6.1.8- Segurtasuna administrazio panelean.....	89
6.1.9- Produktuaren egitura.....	90
6.1.10- Probak.....	91
6.2- Datu meatzaritza.....	93
6.2.1- A gidoiak-Tokiak gomendatzeko tokietan.....	94
6.2.2- B gidoiak-Erabiltzaileei arauen bidez gomendatzeko.....	99

6.2.3- C gidoia-Tokiak erabiltzaileei pertsonalizatzeko....	102
6.2.4- D gidoia-Herriko TOP-ak lortzeko.....	104
6.2.5- E gidoia-Antzeko erabiltzaileekin lotzeko.....	105
6.2.6- F gidoia-Antzeko tokiak gomendatzeko.....	109
6.3- Diseinua eta inplementazioaren arteko loturak .....	111
<b>7. Internetera migrazioa.....</b>	<b>114</b>
7.1- Domeinua eta makina birtuala.....	114
7.2- Izandako arazoak.....	115
<b>8. Produktuaren hedapen puntuak.....</b>	<b>117</b>
<b>9. Jarraipen eta kontrola.....</b>	<b>120</b>
9.1- Proiektuaren garapena eta desbideraketak.....	120
9.2- Kalitatearen jarraipen eta kontrola.....	123
9.3- Arriskuen jarraipen eta kontrola.....	125
<b>10. Ondorioak.....</b>	<b>127</b>
<b>Bibliografia.....</b>	<b>128</b>
<b>Eranskinak.....</b>	<b>130</b>
E.1- Automatikoki Django-extensionen bidez sortutako ereduen informazioa.....	130
E.2- Aplikazioko zenbait formulario.....	134
E.3- Ereduen erazagupena.....	136
E.4- Bilera aktak.....	149

# 1

---

---

## Proiektuaren aurkezpena

### 1.1- Sarrera

---

Dokumentu honetan gradu amaierako proiektu baten azalpena egin da. Proiektu honen helburua turismoaren sustapena da, bai turista gehiago erakartzeko eta baita herrikoen parte hartzea ahalbideratzea ere. Webgune adimentsu bat eratuko da non herrietako guneak sartzeko eta baloratzeko aukera egongo den. Gainera *datu meatzaritzaren* laguntzarekin, erabiltzaileei informazio pertsonalizatua erakutsiko zaie. Produktu hau udaletxeetako informazio bulegoei bideratua dago gehienbat (informazio bulegoetako arduraduna) baina edozein herritarrek ere berarentzat interesekoa dena sartu dezake. Sistemari sartzen den informazioa betirako egongo da (informazio bulegoko administrariak ezabatu ezean) eta jendeak bere iritzia emateko aukera izango du informazio horretarako. Bai denboran eta baita geografikoki hedagarria den proiektua da hau, probintzia berriak eta herri berriak sartzeko aukera egongo delako.

Gaur egun Euskal Herriko turismoa gero eta indar gehiago hartzen ari den sektorea da. Herri asko eta askotan dago informazio bulego bat kanpotik datozeinei aholkuak emateko eta inguruan zer egin-ikusi dezaketean adierazteko. Internetaren zabalkuntzarekin batera, jendeak gauza hauek internet bidez kontsultatzeko ohitura hartu du, baina internet oso zabala da eta egon daitekeen informazioa okerra ere izaten da sarritan. Horregatik, jakin egin behar da non begiratu.



## 1.2- Dokumentuaren egitura

---

Dokumentuan proiektuan zehar egindako lana islatuko da, hasieran egindakoa dokumentuan aurrerago ipiniz eta beranduago eginikoak atzerago. Horrela, denboran egindako lana hobeto adierazteko.

Beraz, hau izango da dokumentuaren egitura:

- **Proiektuaren aurkezpena:** Proiektuaren ikuspegi orokorra izateko.
- **Kudeaketa plana:** Proiektua gauzatzeko plana. Atazak, denbora estimazioak, egutegia, ... Noiz zer egin behar den planifikatuko da.
- **Erabilitako teknologiak:** Produkturako erabili diren teknologien aurkezpena egingo da atal honetan. Produktua garatzeko erabili den lan ingurunea eta datu baseen ezaugarri nagusiak.
- **Sistemaren funtzionamenduaren ideia:** Lortu nahi den produktua hobeto ulertzeko, adibide praktiko batzuk ikusiko dira hemen. Produktuak nolako itxura izan nahi duen ikusiko da txantiloien bidez.
- **Analisi eta diseinua:** Produktuaren barne funtzionamendua ulertzeko, erabilpen kasuak zehaztuko dira, rol ezberdinak, datu basearen egitura eta erabilpen kasu garrantzitsuenen sekuentzi diagramak.
- **Implementazioa:** Produktua nola eratu den ikusteko. Bi atal nagusi bereizi dira hemen, alde batetik web aplikazio soila, eta bestetik 'adimena' ematen dioten datu meatzaritzako gidoiak.
- **Internetera migrazioa:** Produktua makina lokaletik Internetera pasatzeko egindako lanak eta sortutako arazoaren aurrean zer erabaki hartu den ikusiko da atal honetan.
- **Produktu finala eta hobekuntzak:** Zer lortu den eta zer hedatu daitekeen ikusiko da, bai aplikazioaren funtzionamenduari begira eta baita datu meatzaritza tekniken hedapenak ere.
- **Jarraipen eta kontrola:** Proiektuaren bizi ziklo guztian eramandako kontrolaren isla. Planifikatutakoa eta errealitatean gertatutakoaren konparazioa.
- **Ondorioak:** Behin proiektua bukatzean egonda idatzitako hausnarketak.
- **Eranskinak:** Proiektuko eranskinetan, dokumentu nagusian sartzea merezi ez dutenak baina kontsultatzeko interesekoak izan daitezkeenak.

## 1.3- Ideia eta motibazioa

---

Motibazioen artean 2 multzo nagusi bereiz daitezke. Alde batetik kanpo motibazioak eta bestetik barne motibazioa edo norberarena.

Kanpo motibazioen artean eta sarreran aipatu bezala, udaletxe askok badaukate jada beren turismo orrialdea interneten baina ez dute aukerarik ematen erabiltzaileek beren iritzia usteko eta beste batzuek ordea, ez daukate ezta webgunerik ere. Interesgarria izango litzateke turismo webgune hauek dinamikoagoak izango balira erabiltzaileen gustuen arabera. Are gehiago, zergatik ez bateratu informazio bulegoetako datuak webgune batean? Erabiltzaileak probintzia edota herrietan bilaketak egiteko aukera ematea. Zer egin daiteke aste buru honetan Gipuzkoan? Eta Donostian? Ba al dago jairik inguruan?

Horretaz gain, produktuak arazo erreal bati erantzuna eman nahi dio. Herri txiki bateko kanpin arduradunak beren ostalariei inguruko toki interesgarrietara ezin bideratzeko arazotik dator ideia hau. Hasierako ideia beraz, edonork toki interesgarriak eta hauekin ibilbideak antolatzea da, gero edozein pertsonak mugikorrarekin eta GPS bidez ibilbide hauek egiteko. Proiektu honetan baina, irismenean ikusiko den bezala ez zaio arazo guztiari konponbidea emango, mugikorreko aplikazioa hedapen puntutzat utziko da eta.

Barne motibazioen artean ordea, benetako arazo bati konponbidea ematen laguntzea eta teknologia berriak ikastea dira. Aurrerago sakonago aipatuko diren *Django* lan ingurunea eta datu meatzaritzako teknika batzuen aplikazioak ikasteko interesa daude barne motibazioen artean.

## 1.4- Aurrekariak

---

Atal honetan aurretik existitzen diren eta antzekoak diren sistemak aipatuko dira, gero egindakoarekin konparatzeko.

Sarreran aipatu bezala badago informazio turistikoa eskaintzen duen webgune fidagarriarik, esaterako Donostiako turismo informazioko webgunean hainbat informazio

aurki daiteke: Zer egin, mapak, non lo egin, non jan, ... Baina informazio hau oso estatikoa da, hau da, orrialde hau bisitatzeko duten guztientzako erakusten du informazio bera. Gainera, ez dio uzten erabiltzaileei beren Donostiako esperientziak kontatzen, iritzien bidez adibidez. Toki finko bat (taberna bat esaterako) aurkitzeko ere ez dago erraztasun handirik eta hainbat 'klik' han eta hemen egin ostean aurkitu daiteke informazio hori.

Beste webgune profesional batzuk ere badaude, zeinetan iritzia eman daitekeen eta hainbat informazio pertsonalizatu erakusten dizuten. Baina hauen helburua normalean 'saltzea' izaten da eta horregatik informazioa bigarren mailan geratzen da sarritan, eta 'plan' edo egiteko gehienak ordaintzekoak izaten dira.

### **1.4.1- Antzeko sistemak**

Aurreko atalean aipatu bezala, merkatuko antzeko bi sistema mota ikusiko dira. Alde batetik udaletxeetako informazio bulegoa interneten eta bestetik, informazio turistikoa 'saltzen' duten webgune profesional horietakoak. Adibide bezala jartzearren, Donostiako 'Kursaal'-aren informazioa bilatuko da webgune mota ezberdinetan eta bertan agertzen diren informazioak konparatuko dira.

- **Informazio bulegoetako webguneak**

Hauetan agertzen den informazioa egiazkoa izaten da eta hainbat informazio aurki daiteke. Baina, sarritan informazio hau nahiko estatikoa izaten da eta ez diete erabiltzaileei beren informazioa ematen uzten. Beraz esan beharrik ez dago, erabiltzaileek ez dutela informazio pertsonalizaturik jasoko.

Mota hauetako webguneen beste ezaugarri bat, agian ahuldura bat da, bertako informazioa ezin dutela edozein erabiltzaile interesatuk eguneratu. Hau da, hotel bateko langileak ezin ditzake hotel inguruko toki interesgarriak webgunean sartu.

Hauen adibide bezala jarri daitezkeen webguneak: *Donostiako turismo webgunea*<sup>1</sup> edota *Bilboko*<sup>2</sup>. Aipatutako ezaugarriak dituzte biek.

---

1 <http://www.sansebastianaturismo.com/>

2 <http://www.bilbaoturismo.net/>

'Kursaal'-aren informazioa bilatzerakoan, bilaketa kutxan 'Kursaal' idatzita ez da lortu nahi izan dugun emaitzarik. Informaziora iristeko, 'Egin' -> 'Artea eta kultura' -> 'Espazio eszenikoak' bidea jarraitu behar izan da. Hemen, hainbat espazio eszenikoko informazioren artean agertzen da Kursaala (ikusi 1.1 irudia). Aukera ematen du bere webgune ofizialera joateko baina bestela ez du beste munduko informaziorik ematen eta esan bezala, gaizki kokatua.

PALACIO DE CONGRESOS – AUDITORIO KURSAAL



Gauez argizatutako Kursaal kuboak

**INTERESGARRIA ZURETZAT**   
→ Kursaalaren webgune ofiziala

Arkitektura garaikidearen adierazgarri handietako bat, eraiki zutenetik Donostiaren bereizgarri izan den artelana, jaialdien eta era guztietako azoken eta bestelako ekitaldien egoitza den kongresu-jauregia.

Hamaika modutan ezagut daiteke Kursaala eta edozein da egokia. Kanpotik zein barrutik, Rafael Moneo artiketktoak diseinatu zituen bi "kubo" horiek hainbat aukera eskaintzen dituzte urteko egun bakoitzerako.

FORUM FNAC GUNEA 

**1.1 Irudia** - Kursaal-aren informazioa Sansebastianturismo.com-en

- **Informazio turistikoa erakusten duten webgune profesionalak**

Webgune hauetako garrantzitsuena aipatzearen *Tripadvisor*<sup>3</sup> da. Webgune honetan hainbat herritako informazio aurki daiteke, eta erabiltzaileek beren iritzia idatz dezake. Jendearen iritzi hauek baloratu edota salatu egin daitezke gainera eta erakusten duen informazioa oso pertsonalizatua izaten da erabiltzaile erregistratu bakoitzarentzat.

Aurreko esperimentuari jarraituz ea zer agertzen den Donostiako 'Kursaal'-aren inguruan. Ezaugarrietan aipatu bezala, jendeak idatzitako hainbat iritzi dago hemen

3 <https://www.tripadvisor.es/>

Kursaalaren inguruan (ikus 1.2 irudia). Gainera, inguruko argazkiak ere agertzen dira maparekin eta Kursaaleko webgunearen loturarekin batera. Baina ez du bestelako deskribapenik ematen tokiaren inguruan. Beraz esan daiteke webgune honetan Donostiako turismo gunean baino informazio gehiago eta baliagarriago dagoela.

Webgune hauetan ordea, herri txikietako informaziorik ez da aurkitzen edo oso informazio murriztua aurkitzen da.

## Kursaal

160 Opiniones | N.º 33 de 93 cosas que hacer en San Sebastián - Donostia

Edificios con valor arquitectónico, Palacios de congresos y convenciones, Recursos para viajeros, Monumentos y puntos de interés

**Resumen** | Opiniones (160) | Preguntas y respuestas | Ubicación

Guardar | Escribe una opinión

¿Te parece esta atracción adecuada para familias?  
 Sí  No  No estoy seguro

Y Cómo llegar

Salamanca Pasealekua, Zurriola Hiribidea, Gran Vía Kalea, Gros, Oihenbe Kalea, Nurtza Kalea, Kolon Pasealekua

Dirección: Avda. de Zurriola 1, 20002 San Sebastián - Donostia, España  
Número de teléfono: 943 00 30 00  
Página web

Mejorar este perfil

Hoy 9:00 - 14:00 **Abierto ahora**  
16:00 - 19:00  
Ver todas las horas

Experiencias destacadas del colaborador de TripAdvisor

Leer las 160 opiniones

Calificación de los visitantes

Excelente	27
Muy bueno	33
Normal	22
Malo	18
Pésimo	6

**“Edificio singular”**  
Palacio de congresos y auditorio oficial del festival de cine de la ciudad, es de arquitectura de línea moderna a base de cristal en sus fachadas, se encuentra junto a la playa y... leer más

Opinión escrita hace 3 días  
jitorreman, Getxo, España

1.2 Irudia - Kursaal-aren informazioa Tripadvisor.es-en

## 1.4.2- Proposatutako soluzioa

Argi dago proposatutako soluzioa ez dela merkatuko erraldoiak bezain beste adimentsua izango, eta ezta ere herrietako turismo webguneek eskaintzen duten beste informazio emango. Baina baditu hauen bien ezaugarri garrantzitsuenetarikoz batzuk, bat eginda. Proposaturiko soluzioa aurreko puntuan ikusitako bi webgune moten tartean kokatu daiteke. Alde batetik emango duen informazioa egokia izango delako eta bestetik, erabiltzaileari bere iritzia ematea ahalbideratuko zaiolako gero

informazio pertsonalizatua emateko. Bestalde, edozein pertsona interesatuk toki interesgarriak sartzeko aukera egongo da, gero administrariak bere onarpena eman dezan. Hiri handiez gain, herri ez hain turistikoak ere agertuko dira.

Webgune honetan, 4 erabiltzaile (rol) mota egongo dira: Erabiltzaile ez erregistratua, erabiltzaile erregistratua, informazio bulegoetako administraria eta sistemaren administraria.

Webgune adimentsua izango da, erabiltzaileengatik informazioa jaso eta hortik jakintza sortuko duelako. Gune hau probintzian banatuko da (Bizkaia, Gipuzkoa eta Araba), hauetan herriak egongo dira eta azken hauetan tokiak eta jaiak. Hainbat herritako edozein toki sartzeko aukera egongo da bere kokapen, izen, deskripzio eta etiketekin. Hauek gero jendeak baloratuko ditu beren iritzia emanaz. Erabiltzaileak bere profila bete duenean (guztiz ez bada ere) sistema gai izango da bere profilari egokitzen zaizkionak gomendatzeko. Sistemak zenbat eta informazio gehiago izan erabiltzaileengandik, are eta jakintsuagoa izango da eta erabiltzaileari erakutsiko dion informazioa erabiltzailearen gustukoagoa izango da.

Erabiltzaileek toki eta jaietan idazten duten iritzia ebaluagarriak izango dira, hau da, gainontzeko erabiltzaileek iritzia egokia edo ez egokia den esateko aukera izango dute, eta iritzia iraingarria den kasuan iritzia salatzeke aukera egongo da, gero administrariak ezabatu dezan.

Erabiltzaileek bilaketa kutxan bilatu nahi duen probintzia, herria, tokia, jaia... idatzita informazioa pantailaratuko zaie. Sistemak erabiltzailearen nahiko informazio badauka, bere profilarantzat gustukoak direnak erakutsiko zaizkio lehenengo postuetan eta gero gainontzekoak. Aipatutako toki hauek ez dute zertan ezagunak izan beharrik. Adibidez, Igeldo mendian bista ederrak dituen toki bat badago eta interesekoa izan daitekeela iruditzen bazaio norbaiti, arazorik gabe sar dezala. Gero jendearen iritziekin joango da postuak irabazten edo galtzen, baina informazioa egon dadila.

Erabiltzaileak tokiak eta jaiak ebaluatzean, sistema beste toki batzuk gomendatzeko gai izango da. Gomendio hauek ez dira sortuko denbora bat igaro arte, aurrerago ikusiko ditugun *script*-ak (gidoiak) exekutatu arte. Hauetan landuko da datu meatzaritza.

Behin datu meatzaritza eginda, sistemak erabiltzaile batzuei gomendioak egiteko gai izango da. Gomendio hauek mota ezberdinetakoak izango dira (aurrerago ikusiko ditugu sakonago), besteak beste, '*x,y eta z tokiak gustuko izan dituzulako, w tokia ere gustuko izango duzu*', edo '*x tokia gustuko izan dutenek w tokia ere gustuko dute*'. Gainera probintzia bakoitzeko herririk baloratuenak eta hauetako toki baloratuenen rankinga egingo da. Gustuko izan dituzten antzeko toki asko dituzten erabiltzaileak ere lotuko dira, horrela bestearen informazioa ikus dezaten (sartutako tokiak eta ibilbideak esaterako).

Ibilbideak sortzeko aukera ere egongo da. Behin ibilbidea sortua dagoela bertan tokiak gehitu edo kentzeko eta beste erabiltzaileek ikus ditzaten publiko jarri al izango dira.

Aurreko ataleko adibidearekin jarraituz, Kursala bilatuta hainbat informazio agertuko da (ikus 1.3 irudia). Alde batetik, informazio orokorra ikusiko da eta balidatutako informazioa dela adierazten duen ikurra eduki dezake. Bestetik, argazki bat eta mapa ere agertuko dira beren tartean puntuazio orokorra izanik. Horretaz gain, tokia kokatuta dagoen herrirako esteka bat egongo da herriaren informazioa erraz ikusteko. Tokiko etiketak ere ikusiko dira eta honen azpian, erabiltzailearen ibilbide lista egongo da, nahi izanez gero tokia ibilbideren batean sartzeko. Hauen azpian tokia sartu duen erabiltzailearen izena (esteka gisa) egongo da (berak sartutako beste toki batzuk ikusi nahi ezker). Behealdean erabiltzaileen iritziak agertuko dira eta azpirago, intereseko beste toki batzuk (uneko tokiarekin lotura izango dutenak).

Laburbilduz, proposatutako sisteman lehentasuna emango zaio informaziora iristeko erraztasunari eta informazio pertsonalizatua erakusteari. Herri baten informazioa erraz bilatzea ahalbideratuko da bilaketa barran herriaren izena idatzi, eta gero, herrian bertan bilaketa eginda adibidez. Aurrerago ikusiko da (4. atalean) sistemaren funtzionamenduaren ideia sakonago.

### Tokia: Donostia Kursaal -Donostia-SS



#### Informazio orokorra:



#### Deskribapena:

Donostiako Kursaal Kongresu Jauregia, Rafael Moneoren obra arkitektoniko ikusgarri bat da. Kantauri itsaso parean kokatuta, hiriko kultura eta kongresu akitibitateen erdigunea da eta urtero 200 ekitaldi eta 500.000 ikusle baino gehiago biltzen ditu. Beria zeharraigisuzko bi kubok osatzen duten Kursoalek punta-puntako teknologiarik inonituzko espazio balio-aniztunak ditu: 1.806 pertsonentzako edukiera daukan Auditorioa, bilera aretoak (15etik 575 pertsonako edukierarekin) eta erakusketa eremuak. Era guztiatiko kongresu, konbentzio, jarduera, azoka, kontzertu eta ikuskizunak antolatzeko toki berezi eta aproposa da. Gainera, 15 urtetako esperientzia dauka dagoneko. Eskaintza bereizgarri eta bitxia sukaldaritza zerbitzua da: taldeko burua Andoni Luis Aduriz da eta arrakasta ziurtatuta dago "Michelin bi izar" dituen sukaldari honen esperientzia eta sormenaren bermearekin.

Herria: Donostia-SS

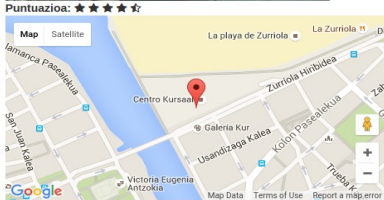
#### Tokiko etiketak:

Hondartza, Jatetxea, Autobusa

#### Gehitu tokia zure ibilbideetara

Ispastar Ibilbidean gorde

Norek sartu: mikel



#### Jendearen iritziak:

JJ (admin) ★★★★★	Ederra benetan Paseoan ibiltzeko toki aparta Baliogarrira Ez egokia Iritziaren puntuak: 1 (bozkek: 1)
USER1 (admin) ★★★★☆	Ondo dago

### 1.3 Irudia - Infoturist-en Kursoala-ri buruzko informazioa

### 1.4.3- Aurrekariak vs soluzioa

Ikusi da aurrekarietan zer aurki daitekeen eta atal honetan proposatutako sistemarekin alderatuko da, ezberdintasunak nabarmenduz.

- Ezberdintasunik handiena informazioa sartzeko erraztasuna da. Proposatutako soluzioan edozein erabiltzaile erregistratuk sar dezake toki interesgarriren bat eta honek, herri txikietako turismoa bultzatzen laguntzen du. Hainbat herritako informazio bulegoko arduradun egon daitezke, bakoitza bere herrialdea(k) edo probintzia(k) administratzen.
- Ez dago erreserbei bideratua. Informazioa ematea da helburu nagusia eta beraz ez da egongo erreserbarik egiteko aukerarik, horrek ez du baztertzen, tokiko webgune ofizialera erreferentzia eginez gero bertatik egin ahal izatea erreserbak.
- Ibilbideak sortzeko aukera egongo da. Ibilbidean tokiak sartu edo kendu ahal izango dira eta ibilbide hauek jendearekin partekatuzeko aukera egongo da.
- Planak sortzeko aukera. Herri bateko toki eta jaiak bueltatuko ditu erabiltzailearen gustuak kontuan izanda. Nahiz eta herriko tokietan iritzirik



emanda ez izan, beste iritzietatik deduzituko dira toki hauek.

- Euskal Autonomia erkidegora mugatuta egongo da. Hiru probintzia egongo dira eta bertako herriak sartuko dira.
- Herrietako jaiak bilatzeko aukera. Herri baten deskribapenetik, bilaketa kutxan jaiaren izena idatzita edota data tarte bat aukeratuta, herrietako jaiak bilatzeko aukera egongo da.
- Gertutasunaren arabera bilaketa. Norbaitek bere inguruan zer toki interesgarri edo zer jai dagoen kontsultatzeko aukera izango du.

Bestalde badira antzekotasunak eta hauek esan daitezke, aurrekarietan ikusitako bi motetan dauden ezaugarriak hoberenak direla.

Aipatutako turismo webguneen berdintasunak:

- Hainbat hizkuntzatan egongo da informazioa eskuragarri (Euskara barne). Informazio turistikoa eskaintzen duen edozein informazio egon behar da hizkuntza ezberdinetan eta horregatik, txantiloietako mezuak hizkuntza ezberdinetan egongo dira eta erabiltzaileei tokien informazioa sartzeko txantiloian eskuragarri dauden hizkuntzetan sartzeko eskatuko zaie beraz.
- Turismo bulegoko arduradunek administratuko dute eta beraz, bertako informazioa fidagarria izango da. Turismo bulegoko langileak erabiltzaile arruntek sartutako tokiek informazio egokia daukatela ziurtatu beharko dute, bakoitzak bere administrazio zonan (herrialdean) noski.
- Herrietako informazio asko aurkitzeko aukera egongo da. Edozein erabiltzaile erregistratuk toki interesgarriak sartu al izango dituen, webguneen informazioa oso anitza izatera irits daiteke.
- Tokien mapak egongo dira. *Google*-ren mapak erabiliko dira tokien kokapena adierazteko eta horrela erabiltzailearen kokapenetik helburura arteko nabigazioa egin ahal izango da.

Turismo orrialde profesionalen berdintasunak:

- Erabiltzaileek iritziak idazteko aukera. Jendearen iritzia oso garrantzitsua da

zeren hauekin datu meatzaritza teknikak aplikatu ahal izango dira eta gainontzeko erabiltzaileek iritziekin tokia gustuko izango duten edo ez jakin ahal izango dute.

- Informazio pertsonalizatua erakutsi. Hau erabiltzaile erregistratuetara soilik mugatuko da, sistemak hauen informazioa izan beharko duelako lehendabizi. Horrela tokiak gomendatzeko gai izango da sistema eta webguneari dinamismoa emango dio.
- Informaziora iristeko erraztasuna. Intuitiboki eta klik gutxitan tokietako informaziora iristea ahalbideratuko da probintzi eta herrien arteko nabigazio erosoarekin.
- Sistemak gomendioak egiten ditu. Aurreko puntu baten esan bezala, erabiltzaile erregistratuei soilik.
- Erabiltzaileek sortutako datuak gordeko dira, gero datu hauekin datu meatzaritza egin ahal izateko.

# 2

---

---

## Kudeaketa plana

### 2.1- Helburuak

---

Proiektuan zehar hainbat helburu daude finkatuak. Haietako batzuk gradu amaierako proiektu guztiek izan behar dutenak eta besteak proiektu hau gauzatzeko beharrezkoak direnak.

Gradu bukaerako proiektua denez, memoria eta defentsa ere izan behar ditu. Hauek dira finkatutako helburuak:

- Web aplikazio funtzional bat egitea
- Egindakoa islatzen duen memoria idaztea
- Egindakoaren defentsa prestatu eta egitea
- Django*<sup>4</sup> lan ingurunea erabiltzen ikastea
- Datu meatzaritzako teknikak datu errealei aplikatzea
- Planifikazio eta kudeaketa egokiak egitea

#### 2.1.1- Produktuaren helburuak

Produktu honen helburua turismoaren sustapena da, bai turista gehiago erakartzeko eta baita herrikoen parte hartzea ahalbideratzea ere. Webgune adimentsu bat eratuko da *Django* lan ingurunea erabiliz, non herrietako guneak sartzeko eta baloratzeko aukera egongo den. Gainera *datu meatzaritzaren* laguntzarekin, erabiltzaileei informazio pertsonalizatua erakutsiko zaie. Produktu hau udaletxeetako informazio bulegoei bideratua dago gehienbat (informazio bulegoetako arduradunei batez ere)

---

4 <https://www.djangoproject.com/>

baina edozein herritarrek ere berarentzat interesekoa dena sartu dezake. Webgune hau, hedagarria izango da. Sisteman sartzen den informazioa betirako egongo da (informazio bulegoko administrariak ezabatu ezean) eta jendeak bere iritzia emateko aukera izango du informazio horretarako.

## **2.1.2- Memoriaren helburuak**

Proiektuaren isla izango da hau. Plangintza, jarraipena eta produktuaren dokumentazioa besteak beste hemen aurkituko dira. Produktuak benetan nola funtzionatzen duen jakiteko ezinbestekoa izango da hau irakurtzea.

## **2.2- Irismena**

---

Proiektu batean egon behar diren atal nagusi guztiak egitea eta funtzionala den produktu bat aurkeztean definitu da irismena. Hau aurrera eramateko, hasieran plangintza bat definituko da eta proiektuaren bizi ziklo guztian jarraipen eta kontrola ere egingo da.

Informazio turistikoa erakusteko gai den webgune bat izango da funtsean. Webgune honek zenbait datu analizatuz erabiltzaileei toki turistikoa gomendatzeko gai izan behar da. Informazio turistikoa ez du sistemaren administratzaileak sartuko, baizik erregistratutako edozein pertsona edota baimendutako erabiltzaile bereziek. Denbora balego, internetetik atzigarri jarriko litzateke webgunea.

### **2.2.1- Betekizunak**

Gradu amaierako lana aurkeztu baino lehen betekizun batzuk bete behar dira. Haietako garrantzitsuena graduko gainontzeko irakasgaiak gaindituak izatea (228 kreditu gaindituak) da, proiektu honek 12 kreditu balio ditu (2 irakasgai bezain beste). Proiektuak graduan zehar hartutako espezialitatearekin bat egin behar du, kasu honetan, software ingeniariarekin.

## 2.2.2- Mugak

Graduan zehar egindako lanekin konparatuz denbora asko daukagu proiektua burutzeko (300 lan ordu). Baina egin beharreko atazek ikasketa eta ikerketa eskatzen dute eta honek atzerapenak ekar ditzake. Proiektua 2016ko uztailean aurkezteko asmoa izanda, 300 lan ordu banatu behar dira 8-9 hilean zehar. Hau kontrolatzeko zenbait barne mugarri jarriko dira benetako mugarriak baino lehen (aurrerago ere ikusiko direnak *Gantt* diagraman).

- **Barne mugarriak**

- Plangintza bukatu (2015-11-1)

- Aplikazioaren analisia eta diseinua bukatu (2015-11-30). Aplikazioaren funtzionalitateak zehaztuta.

- Django* aplikazioa bukatuta (2016-04-25). Aplikazioaren zati ikusgarriena bukatuta beharrezko funtzionalitateekin.

- Datu meatzaritza teknikak (2016-04-20). Aplikazioan informazio pertsonalizatua erakusteko tekniken inplementazioa.

- Memoria (2016-06-15). Proiektuaren memoria idatzita.

- Defentsa (2016-06-30). Defentsa aurkezteko beharrezkoak direnak prestatuta.

- **Kanpo mugarriak**

- Proiektua entregatzeko epea (2016-06-24). Egun honetan memoria eta produktuak prest egon behar dute

- Proiektuaren defentsa (2016-07-08 eta 2016-07-12 bitartean). Egun hauetako batean egin behar da defentsa.

Barne mugarriekin errore margen bat daukagu, hau da, atzeragarriak dira. Kanpo mugarriak berriz, finkoak dira eta nahitaez errespetatu behar dira.

### 2.2.3- Produktuaren irismena

Produktu guztia digitala izango da, ez da produktu fisikorik sortuko (dokumentazioaz aparte). Webgune bat izango da produktuaren oinarria eta hedapen puntu gisa utziko da gero mugikorretarako aplikazioa egitea. Webgune hau, gutxienez lokalki funtzionala izan beharko da eta ez da beharrezkoa izango internet bidez atzigarri egotea bere funtzionalitate guztiekin behintzat.

Webgunea *Django* lan ingurunearen bidez garatuko da eta erabiltzaileei gomendioak emateko, hauek sortutako datuak analizatuko dira automatikoki, behin erabiltzailearen nahiko informazio daukanean.

Webgunean bi motatako informazioa erakutsiko da, alde batetik toki interesgarriak eta bestetik herriko jaiak, printzipioz bakoitzean argazki bat jarri al izango delarik izena eta deskribapentxo batekin. Toki eta jai hauetan erabiltzaileek beren iritzia idazteko aukera izango dute, gero iritzi hauekin erabiltzailearen profilerak egokitzen diren tokiak gomendatu ahal izateko.

### 2.2.4- Irismen mailak

Produktua garatzeko erabiliko diren lan inguruneak ezezagunak zaizkidanez, hauen erabilpenak suposatuko duen denbora kostua zein den estimatzea zaila izango da. Gainera, datu meatzaritza egiteko egin beharrekoak ere ikasteko daude.

Hauek dira definitutako **irismen mailak**:

1. Web aplikazio funtzional soila: Django bidez sortutako web aplikazioa, zeinetan aplikazioaren funtzionalitate gehienak dauden.
2. Web aplikazio funtzional landua: Web aplikazioa zeinek sortutako datuak gorde eta analizatzen dituen.
3. Web aplikazio funtzional bukatua: Hodeian dagoen web aplikazioa. Erabiltzaileen datuak jaso, gorde eta analizatzen dituen aplikazioa.

**Hedapen puntu nagusi** gisa:

4. Mugikorreko aplikazioa: *Ionic* bidez landutako mugikor aplikazioa, toki interesgarriak

kontsultatzeko.

Derrigorrez bete behar diren irismen mailak lehenengo biak dira, bestea osagarri bezala izanik eta azkena hedapen puntu nagusi gisa.

## 2.2.5- Lanaren deskonposaketa egitura (LDE)

Proiektuan zehar egin beharreko atazak, 5 multzotan banatu dira (ikus LDEa 2.1 irudian):

- **Kudeaketa**

Proiektuak irauten duen bitartean egin beharrekoa. Plangintza txostena eta atazen jarraipen eta kontrola biltzen dira hemen. Behin plangintza bat eginda, jarraipen eta kontrola landu behar dira egindako ataza bakoitzarekin.

- **Ikasketa**

Lan inguruneen ikasketa eta datu meatzaritza lantzeko egin beharreko ikasketak biltzen dira hemen. Erabilgarriak suertatu daitezkeen kode zatiak edota tutorialak ere sar daitezke. Oso garrantzitsua den multzoa da hau, garatze prozesuan denbora aurrezteka ekar dezakeelako fase honetan egindako lana ona izan bada.

- **Arkitektura**

Produktua garatzen hasi aurretik, zer nolako arkitektura izango duen erabaki behar da. Produktuaren analisi eta diseinua, erabiliko den testu prozesatzailea, sistema eragilea, ...

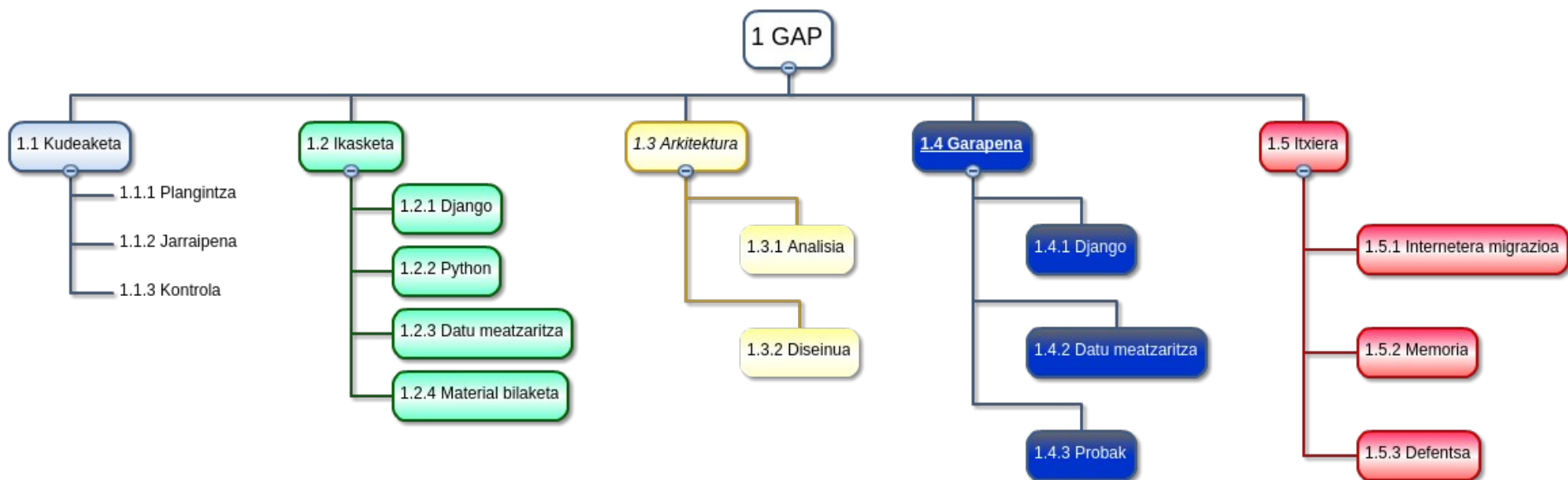
- **Garapena**

Produktua garatzeko egin beharrekoak sartzen dira multzo honetan. Horregatik, dedikazio gehiena izango duen atala da beraz. Gehienbat *Djangon* lan egitea izango da *python* erabiliz. Aplikazioa garatzen den bitartean probak ere egingo dira, egindakoa ondo dagoela ziurtatzeko.

- **Itxiera**

Behin produktua bukatu dela ziurtatu dela, interneterako migrazioa egiteko denbora badago, *hosting* egoki bat bilatu beharko da. Horretaz gain, memoria idazketa egin behar da nahiz eta produktua egiten joan ahala, memorian egongo diren testuak idazten joango diren. Gainera, proiektuaren defentsa ere fase honetan garatuko da.





www.wbstool.com

## 2.1 Irudia - Proiektuko LDEa

Kodea	Ataza	Deskribapena
<b>1-Kudeaketa</b>		
1.1	Plangintza	Proiektuaren plangintza: Atazen zehaztapena, orduen estimazioak, egutegia, arrisku plana, kalitate plana...
1.2	Jarraipena	Proiektuan lan egiten den bakoitzean landu beharreko atala. Proiektuaren egoera eta aurrerakuntza islatuko du. Gainera, zuzendariarekin bilerak ere hemen sartzen dira.
1.3	Kontrola	Jarraipenean identifikaturiko ezbeharrak analizatu eta proiektua bukatzeko egin beharreko aldaketak egingo dira. Dena behar bezala doala ziurtatuko da.
<b>2-Ikasketa</b>		
2.1	Django	Django lan ingurunea nola erabili landuko da hemen.
2.2	Python	Python programazio lengoaiaren ikasketa.
2.3	Datu meatzaritza metodoak	Internetetik datuak jasotzeko metodoak ikasiko dira.
2.4	Material bilaketa	Erabilgarria gerta daitekeen materiala bilatzea (liburutegiak, txantiloiak, etb).
<b>3-Arkitectura</b>		
3.1	Analisia	Proiektuaren espezifikazio nagusiak finkatu eta zein osagai beharko dituen
3.2	Diseinua	Produktuaren analisi eta diseinuak, fluxu diagramak,... sistemako osagaiak nola komunikatuko diren.
<b>4-Garapena</b>		
4.1	Django	Web orrialdearen garapena sartzen da hemen
4.2	Datu meatzaritza	Internetetik/Web orritik datuak jaso, gorde eta analizatzeko egin beharrekoak daude hemen.
4.3	Probak	Egindako inplementazioen probak.
<b>5-Itxiera</b>		
5.1	Internetera migrazioa	Produktua interneten atzigarri jartzeko egin beharrekoak.
5.2	Memoria	Proiektuan zehar egindakoa biltzen duen dokumentua.
5.3	Defentsa	Proiektuaren defentsan erabiliko diren materialen prestaketa.

**2.1 Taula** - LDE eta azpiatza nagusien deskribapena

## 2.2.6- Irismenaren kudeaketa

Hasiera batean finkaturiko irismena aldatzea gerta liteke, atzerapenak gerta daitezkeelako. Hau, jarraipen eta kontrola egitean erabakiko den gauza da. Aldaketaren bat gertatzen bada, jarraipeneko txostenean adierazita geratuko da. Baina gutxienez adierazitako puntuak bete beharko dira.

## 2.3- Lan metodologia

---

Proiektua aukeratutako datan bukatuta egongo dela ziurtatzen duen lan metodologi bat egin behar da. Horretarako, jarraipen egoki bat egitea beharrezkoa da.

- **Lan metodologi pertsonala**

Alde batetik ordutegi bat ezarriko da lan pertsonala aurrera eramateko. Lehenengo lauhilekoan ordu gutxiago sartuko dira proiektuan beste irakasgai kopuruarengatik. Bigarren lauhilekoan berriz, irakasgai kopurua txikiagoa denez, ordu gehiago sartuko da orduan. Bestetik, lan egun bakoitzean jarraipen '*fitxa*' bat bete beharko da ondorengo informazioa jarriz:

1-Zein fasetan lan egin den: Proiektuko LDE-ko zein fasetan eta azpiatazan lan egin den adierazi beharko da hemen.

2-Egindako lana bukatuta geratu den: Lanean aritutako ataza bukatuta geratu den edo ez adierazi behar da.

3- Landutako dedikazioa ordutan: Ataza hori egiten emandako orduak

4-Egiteko geratzen dena: Ataza bukatuta geratu ez bada, zer gehiago egin behar den bukatutzat emateko.

5-Dokumentazioa: Memoriarako baliagarria izango den eguneko dokumentua idaztea.

Aipatutako fitxa hori 'Google forms' bidez beteko da ia bere osotasunean (lehenengo 4 atalak) eta Google-ren kalkulu orrietan gordeta geratuko da beraz. Horrela, lan historia

islatuta izango da. Dokumentazioa lokalean bertan idatziko da eta gero segurtasunagatik Drive-ra igoko da.

Django proiektuan lan egitean *Git*<sup>5</sup> erabiliko da bertsio kontrola izateko eta hauen bertsio berriena interneten ere gordeko da.

Proiektua iterazioka egingo da eta produktua gutxika haziko da. Hau da, produktuaren zati txiki bat egingo da lehenago eta hurrengo iterazioetan aurreko produktua hedatuko da. Iterazio bakoitzean egingo da garapena eta proba kasuak ere, egindakoa ondo dagoela ziurtatzeko.

- **Bilerak**

Zuzendariarekin bilerak egingo dira gutxienez hilean behin proiektuak aurrera nola egiten duen ikusteko eta gure helburura iritsiko garela ziurtatzeko, behar diren aldaketak eginez. Bilera hauek zuzendariaren bulegoan bertan egingo dira eta produktuaren bizi ziklo amaiera hurbildu bitartean bileren maiztasunak handitu egingo dira.

## **2.4- Emangarri nagusiak**

---

Proiektuan hainbat emangarri daude, batzuk barne emangarriak (produktuarekin lotutakoak gehienbat) eta besteak beharrezkoak (behin proiektua bukatuta dagoela).

- **Barne emangarriak**

-Web aplikazioaren prototipoa

-Web aplikazioaren dokumentazio zatiak. Produktua garatzen den bitartean bere dokumentazio zatiak ere Drive-ra igoko dira.

-Datu meatzaritza teknikak aplikatu ostean lortutako emaitzak

- **Kanpo emangarriak**

-Proiektuaren kudeaketa plana (memorian barnean)

---

5 <https://git-scm.com/>

-Produktuaren dokumentazioa (memorian barnean)

-Memoria

-Proiektuaren defentsa

## 2.5- Baliabideak

---

Proiektua aurrera eraman al izateko, hainbat tresna beharko dira. Hauetako gehienak softwareak izango dira baina ikasketarako hainbat dokumentu ere eskuratu beharko dira.

- **Hardwareak**

Proiektua gauzatzeko nahikoa da ordenagailu bat izatea lokalki lan egiteko. Gero Internetera migrazioa egin nahi bada hardwarea alokatu egingo da. Segurtasun kopiak egiteko Googleren biltegiaz aparte pen drive fisiko bat ere erabiliko da.

- **Softwareak**

Atal honetan hainbat liburutegi eta programa beharko dira.

-Garrantzitsuenak Python eta Django dira. Hauek software libreak direnez ez da arazorik egongo makinetan instalatzeko. Pythonen lan egiteko lan ingurune egoki bat ere beharrezkoa da, horregatik *PyCharm Community Edition* erabiliko da.

-Datu base ezberdinak erabiliko dira eta beraz, beharrezkoak izango zaizkigu *MySQL* eta *MongoDB*<sup>6</sup> softwareak eskuratzea. Gainera azken honekin Python erabiltzeko *PyMongo* erabiliko da.

-Facebook eta Google-rekin konexioa egiteko beraien API-ak ere erabiliko direnez, API-kodeak lortu beharko dira.

-Python-eko zenbait liburutegi ere erabiliko dira.

-Sekuentzi diagramak egiteko *StarUML* erabiliko da eta klase diagrama berriz *Diagrams* softwarearekin.

---

<sup>6</sup> <https://www.mongodb.com/>

-Dokumentazioa idazteko LibreOffice erabiliko da.

- **Bibliografia**

Python eta pythonekin datu meatzaritza implementatzeko liburuak

## **2.6- Proiektua gauzatzeko plana**

---

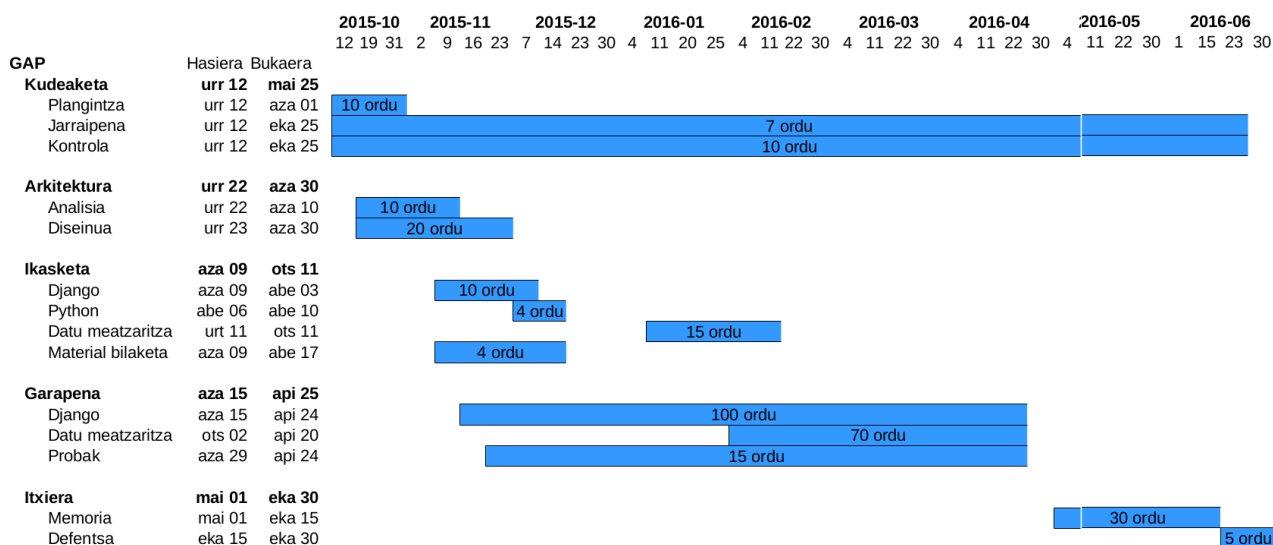
### **2.6.1- Egutegia**

Proiektuak bi irakasgai bezain beste balio du, hau da, 300 lan ordu (12 kreditu). Lan ordu hauek igande arratsaldetik ostegunera bitartean egitea erabaki da, ostiraletik igande goizera proiektuan lanik ez da egingo beraz.

Normala izango da 300 ordu horiek zehatzak ez izatea, hau jarraipen eta kontrola egitean eta proiektua aurrera doan heinean ikusiko da gehiago edo gutxiago den.

### **2.6.2- Gantt diagrama**

Behin egutegia finkatuta, Gantt diagrama sortu da. Hemen, atazak noiz egin behar diren ikusten dira eta beren bizi iraupena. Produktua apirilaren 10ean amaitzea planifikatu denez, Gantt diagraman ere produktuaren azken ataza Apirilaren 10ean amaitzen da.



## 2.2 Irudia - Proiektuko Gantt diagrama

Nahiz eta Gantt diagraman memorian lan egingo dela bukaeran soilik dagoen, atazak egin ahala dokumentatzen joan behar da eta horrela, memoriarako dokumentazio asko eginda egongo da. Beraz, nahiz eta bukaeran soilik agertu, proiektuan zehar memoriarako materialean lan egingo da.

Kodea	Ataza	Estimazioa	Hasiera Data	Bukaera Data	Iraupena
1	Kudeaketa	27 ordu	2015-10-12	2016-06-26	261 egun
1.1	Plangintza	10 ordu	2015-10-12	2015-11-1	21 egun
1.2	Jarraipena	7 ordu	2015-10-12	2016-06-25	260 egun
1.3	Kontrola	10 ordu	2015-10-12	2016-06-25	260 egun
	-Kalitate kontrola	--5 ordu			
	-Arriskuen kontrola	--2 ordu			
	-Bukaera ziurtatu	--3 ordu			
2	Arkitektura	30 ordu	2015-10-22	2015-11-30	38 egun
2.1	Analisia	10 ordu	2015-10-22	2015-11-10	18 egun
2.2	Diseinua	20 ordu	2015-10-23	2015-11-30	37 egun
3	Ikasketa	33 ordu	2015-11-09	2016-2-11	93 egun
3.1	Django	10 ordu	2015-11-09	2015-12-03	26 egun
3.2	Python	4 ordu	2015-12-06	2015-12-10	5 egun
3.3	Datu meataritza	15 ordu	2016-1-11	2016-2-11	31 egun
3.4	Material bilaketa 1	4 ordu	2015-11-09	2015-12-17	39 egun
	-Djangorekin zerikusia duen materiala				

	-Datu meatzaritzarekin zerikusia duen materiala				
4	Garapena	185 ordu	2015-11-15	2016-4-25	163 egun
4.1	Django	100 ordu	2015-11-15	2016-4-24	162 egun
4.2	Datu meatzaritza	70 ordu	2016-2-01	2016-4-20	81 egun
4.3	Probak	15 ordu	2015-11-29	2016-4-25	149 egun
5	Itxiera	35 ordu	2016-5-01	2016-6-30	60 egun
5.1	Memoria	30 ordu	2016-5-01	2016-6-15	45 egun
5.2	Defentsaren prestaketa	5 ordu	2016-6-15	2016-6-30	15 egun

**2.2 Taula** - Gantt-aren isla. Atazak, iraupena eta epea

### 2.6.3- Lan karga

Gantt diagraman ikus dezakegu lan karga handiena izango den asteak Azaroa-Abendua bitartekoak eta Otsaileko lehen asteak direla. Aste horietan 6-7 lan paraleloan egin beharko direla aurreikusten da, ez da zertan egun berean guztiak egin beharrik, baina aste horietan zehar ataza horiek landu beharko dira. Bestalde lan karga gutxien proiektua hastean eta Apiril bukaeran izango dela aurreikusten da, produktua bukatuta egongo dela suposatzen delako eta memorian eta defentsan bakarrik lan egin beharko delako.

### 2.7- Kalitate plana

---

Proiektuan zehar hainbat atazatan egin behar da lan eta ataza hauek elkarrekin lotuta daude nolabait. Ataza hauen kalitate minimoa ezarri behar da eta hau betetzen dela ziurtatu. Hau egiteko, lau lan egingo dira:

- Gutxieneko kalitatea ezarri
- Kalitatearen adierazleak
- Kalitatea ziurtatzea
- Kalitatearen kontrola



## 2.7.1- Gutxieneko kalitatea

Gutxieneko kalitate bat ezarriko zaio proiektuari. Kalitate hau gainditzen bada, proiektua bukatutzat eman daiteke nahiz eta kalitate gehigarriak inplementatu daitezken. Proiektuak gutxienez hau bete behar du:

- Irismenean zehaztutakoa betetzen da.
- Emangarri nagusiak epean emango dira.
- Aplikazioa lokalki (egilearen ordenagailuan) funtzionala da.
- Planifikatutako estimazioak errespetatzen dira desbideraketa txikiekin.
- Erabiltzaileek web orrian sortzen dituzten datuak prozesatzen dira.

## 2.7.2- Kalitatearen adierazleak

Kalitatearen adierazleak asko dira baina garrantzitsu batzuk definituko dira gutxieneko kalitatera iristeko.

- Jarraipen eta kontrola

Jarraipenerako erabiliko den kalkulu orrian egindako lanaren monitorizazioa gordetzen da. Beraz, astetik atera planifikatutako lan orduak egiten direla ikustea adierazle ona da. Bestela, kontrola hartu behar den edo ez ikusi beharko da.

- Kode kopurua

Aplikazioa bere bukaerara iritsi behar bada, kode kopuruak gora egin behar du. Aстетik atera kode kopurua handitzen doala ikustea ere adierazle ona da.

- Kodearen ulergarritasuna

Kodea edonork ulertu badezake, ondo idatzita dagoelako da; komentarioekin, tabulazio egokia, ... Djangok python erabiltzen duenez, tabulazio egokia derrigorrezkoa da eta metodo bakoitzari komentario bat jarriko zaie edota metodoen izenak intuitiboak izango dira. Honek, kalitatezko kodea lortzea ahalbideratuko du.

### **2.7.3- Kalitatea ziurtatzea eta kontrola**

Kalitatea ziurtatzeko aurreko puntuan zehaztutako adierazleak kontuan hartuko dira. Astean zehar gutxienez behin adierazle horiek nola doazen ikusiko da eta horrela, iritsi behar dugun gutxieneko kalitatea ziurtatuko da. Kontrol hau egiteko jarraipeneko kalkulu orria ezinbestekoa da eta proiektuaren 'argazki' orokorra ere. Gainera aldian-aldian tutorearekin bilerak egingo dira proiektuaren egoera ikusi eta egin beharrekoak adosteko.

### **2.8- Arriskuak**

---

Proiektua aurrera doan heinean hainbat arrisku gertatu daitezke bere entrega data atzeratzea ekar dezaketenak. Horregatik, garrantzitsua da arrisku nagusiak identifikatzea eta baten bat gertatu ezkeru zer egingo den planifikatzea.

- Plangintza zehatz mehatz ez jarraitzea

Kasu honetan argi dago jarritako mugarrak ez direla beteko eta beraz, mugarren datak atzeratu egin beharko direla. Hau arazo handia ez izateko jarri dira mugarrak benetako entrega data baino denboran aurrerago. Plangintzan jarritakoa atzeratzeko aukera badago, produktua Apirilaren 10erako bukatuta egongo dela bai dio, eta aurkezpena Uztailean egingo da. Beraz, arrisku honek gertatzeko probabilitate handiak dauzka baina ekar ditzaken ondorioak daten planifikazioa egitean aurreikusten izan denez, mugarrak asko aurreratu dira.

- Lan ingurunea-teknologiek pentsa baino ikasketa gehiago exijitzea

Lan ingurune eta teknologia 'berriekin' egingo da lan produktua aurrera eramateko eta hauen ikasketak uste baino zailagoa izan daiteke. Guzti hauen integrazioak eta ikasketak ezin dira errez auresan zenbateko dedikazioa eskatuko duten, baina arrisku handi bat da benetan lortu nahi dena iristeko ikasketa/inbestigazioa luzatzea. Arrisku honek ere, produktuaren eta beraz proiektuaren bukaera data atzeratzea ekarriko du. Teknologien ikasketa honek arazo handia ematen badu, kalitate minimoa betetzen duen aplikazioa egiten saiatuko da gehigarririk gabe.

- Egindako lana 'galtzea'

Hainbat kode eta dokumentu idatziko dira proiektuan zehar eta ondo babesturik ez badaude, gal daitezke; Ordenagailua apurtu, nahi gabe fitxategiak ezabatu, kode 'ona'-ren gainean lan egin eta emaitza okerragoak lortu, ... Hauetako bat gertatu eta proiektuko katastrofea izan ez dadin, segurtasun kopien ohitura bat hartuko da. Lehenik eta behin, kodearen bertsio-kudeaketa eramateko *Git* erabiliko da. Gainera, lan egiten den bakoitzean dokumentu/kodearen eguneko segurtasun kopia gehigarria egingo da, hau da, egunean bertan egindakoa soilik gehituko da segurtasun kopiara. Astean behin segurtasun kopia totala egingo da, hau da, kode, dokumentu, informazio,... guztiaren segurtasun kopia.

Segurtasun kopia horiek bi tokitan gordeko dira funtsean, alde batetik *Google Drive*-ra eta bestetik *pen drive* batean. Beraz, lana galtzeko hiru tokiek huts egin beharko lukete aldi berean (aipatutakoak + lan egingo den ordenagailua). Praktika onak jarraitzeko, segurtasun kopiak zifratuak gordeko dira pasahitz baten bidez.

- Aplikazioaren garapenean arazoak aurkitzea

Arazo hau ikasketarenarekin erlazionatuta dagoen arren, baliteke aplikazioaren garapenean konturatzea ezin dela helburua lortu. Besteak beste hirugarren batzuek softwareak baliatuko direnez (Facebook, google,..) baliteke hauek arazoak ematea, edota Djangok hiru-lau erabiltzaile mota ez ahalbideratzea. Hau gertatzen bada, aplikazioa birfinkatu beharko da eta beste antzeko produktu bat lortu. Aipatutakoetan hau da arriskurik handiena.

## **2.9- Parte interesatuak**

---

Proiektuko interesatu nagusia proiektuko egilea da. Berak garatu behar duelako produktua eta emangarri guztiak. Gainera, honen gradu bukaerako proiektua denez, gradua bukatzea edo ez adierazten du.

Bestalde, ikasle hau zuzendari baten laguntzarekin dago proiektua aurrera eramaten. Zuzendari hau ere bigarren interesatu nagusia da. Honek proiektua bideratu eta amaieran ebaluatu beharko du.

Proiektua bukatuta dagoenean aurkeztu egin beharko da, eta hemen epaimahaia egongo da (kasu honetan software ingeniartzakoa) eta hauek proiektua ebaluatu behar izango dute aurretik dokumentazioa irakurrita dutelarik. Beraz, hauek ere parte interesatuak dira.

Azken interesatuak *Donostia 2016*<sup>7</sup>-koak dira. Interesatu hauek proiektua martxan zegoela agertu ziren, gradu bukaerako lanen deialdi batean izen ematean. Deialdian parte hartzeko proiektuak kulturarekin zerikusia izan beharra zuen. Proiektua onartuta geratu zen momentutik hauek ere interesatu bihurtu ziren.

### **2.9.1- Komunikazio plana**

Parte interesatuen artean komunikazio bat egongo da proiektuan zehar, defentsako epaimahaia alde batera utzita. Gehienbat ikaslea-zuzendaria bitartekoa izango da komunikazioa eta hau helbide elektronikoko bidez egingo da bilerak hitzartzeko eta gero bileretan bertan ahoz-ahoz.

Epaimahaiari ADDI plataformak jakinaraziko dio noiz den defentsa eguna eta non aurki dezaketen memoria.

---

7 <http://dss2016.eu/>

# 3

---

---

## Erabilitako Teknologiak

Proiektua aurrera eramateko hainbat teknologia ezberdin erabili dira. Horietan nagusia Django lan ingurunea da, web aplikazioa sortzen lagundu duena. Honek, Python programazio lengoia erabiltzen du. Iraunkortasun mailarako bi datu base mota erabili dira, alde batetik MySQL eta bestetik MongoDB. Azken honetako datuekin datu meatzaritza aplikatzea izan da helburu nagusia.

### 3.1- Python

---

Programazio lengoia bezala Python erabiliko da. Hau, interpretatutako programazio lengoia bat da, ardatz bezala sintaxi garbi, erraz eta irakurgarri bat duelarik. Beraz, kodea garbia izan dadin lagunduko du.

Pythonez idatzitako hainbat liburutegi ere erabili dira produktua aurrera eramateko eta garrantzitsuenak aipatuko dira jarraian.

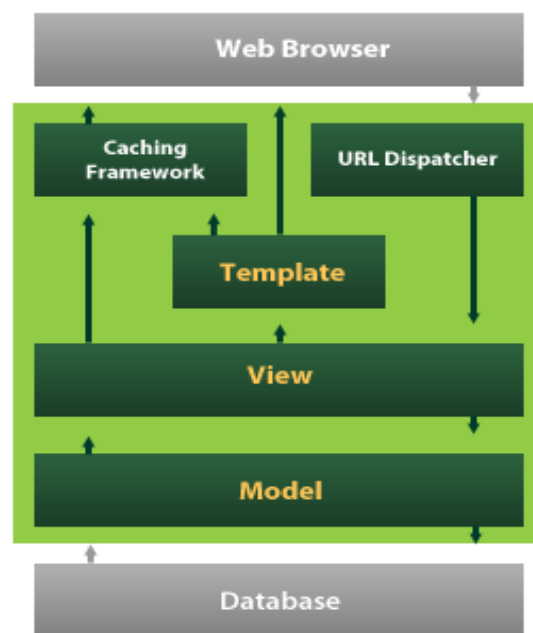
Djangok berak dakar hainbat python kode. Besteak beste klase diagraman ikusiko dugun User eredua eta honen hainbat metodo (erregistroa, logina, logouta,...). Baina atal hau gehiago zentratuko da Djangotik aparteko liburutegiekin. Produktuan bi datu base mota erabiltzen dira, alde batetik MySQL eta bestetik MongoDB. Lehenengoaz Django arduratzen da, baina bigarrena erabiltzeko liburutegiak bilatu behar izan dira lana errazteko. Ala ba, 'Pymongo 2.7.2' driver paketea instalatu da makinan. Honekin, python kodetik Mongo datu baseak modu errazean kontrolatu daitezke: konexioa ezarri, fitxategia aukeratu, fitxategian dokumentuak sortu, ezabatu eta aldatu...

## 3.2- Django

---

Django, python programazio lengoaian idatzitako kode irekiko lan ingurunea da. Lan ingurune honek, eredu-bista-kontrolatzailea (model-view-controller (MVC)) diseinu patroia errespetatzen du, hau da, hiru geruzatako web aplikazioa sortzen laguntzen du. Patroi honek softwarearen arkitekturaren datuak, negozio logika eta erabiltzailearen interfazea bereizten ditu. Betiko MVC patroian bista arduratzen da erabiltzaile interfazeaz, kontrolatzailea negozio logikaz eta azkenik, eredu data basearekin.

Djangon berriz, patroi honen ordena ezberdina da: Modeloa (model) datu basearekin parekatu daiteke, bista (view) negozio logikarekin, eta txantiloia (template) berriz, erabiltzaileak ikusten duenarekin. Ikusi 3.1. Irudia.



**3.1 Irudia** - Djangoren komunikazio arkitektura

Djangoren oinarrizko helburua web orrialde konplexuen sorrera erraztea da eta bere filosofia “Don't repeat yourself” da, hau da, hainbat baliabide eta kode berrerabiltzen dira.

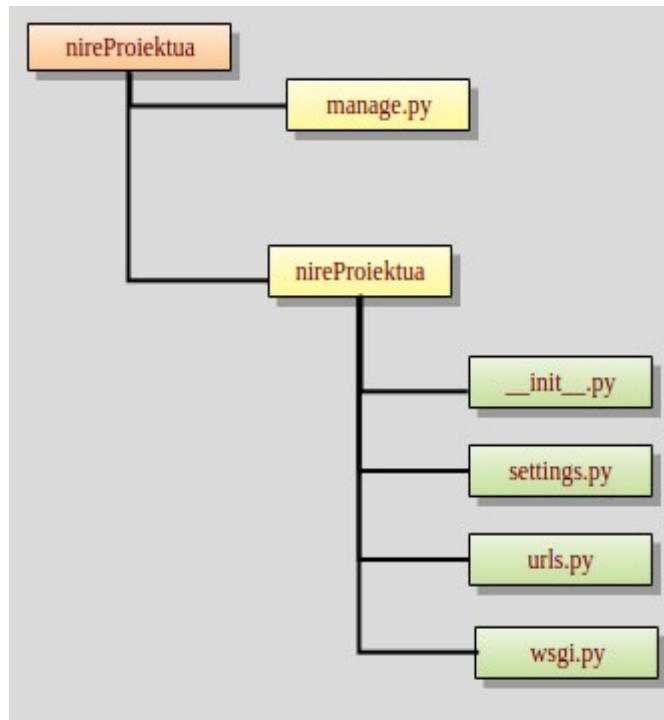
### 3.1.1- Djangoren karpeten egitura

Djangon karpetek eta beren barruko fitxategien izenek garrantzia handia dute, batzuk

gutxienez derrigorrez egon behar dutelako. Hasieran proiektu berri bat sortzean:

```
$ django-admin startproject nireProiektua
```

3.2 irudian ikus dezakegu 'nireProiektua' fitxategiaren edukia.



**3.2 Irudia** - 'nireProiektua' proiektuaren egitura

Karpeta eta egitura horiek derrigorrez mantendu behar dira dauden bezala.

- Manage.py

Fitxategi hau da proiektua martxan jartzen arduraduna. Fitxategi hau exekutatu jarri daiteke webgunea martxan edota aurrerago ikusiko ditugun datu meataritzako script-ak exekutatzeko. Webgunea martxan jartzeko esaterako:

```
$ python manage.py runserver
```

- NireProduktua

Direktorio hau proiektuko direktorio nagusia izango da. Hemen daude kokatuak proiektuaren ezarpenak, URL-ak eta beharrezkoak diren `__init__.py` eta `wsgi.py` fitxategiak (azken hauek ez direnez ukitu, ez ditugu aipatuko).

- Settings.py

Proiektuko ezarpen nagusiak hemen jarri behar dira. Besteak beste: Proiektuan erabiliko diren aplikazioak (eta hauen ezaugarriak), templateak (txantiloiak), datu baseen konfigurazioa, multimedia fitxategien kokapena, ...

- urls.py

Hemen Django ahalbideratzen duen URL bereziak kokatuko dira. Django, nabigazio barran agertzen den URLak eta erabilitako txantiloiak ezberdinak dira. Fitxategi honetan, URL lotura lista bat joango da eta hauen kokapenak garrantzia izango du, goren dagoenak garrantzi gehien izanda. Hasiera batean URL bakarra dauka:

```
urlpatterns = [  
    url(r'^admin/', include(admin.site.urls)),  
]
```

Honek esan nahi du, URLan “localhost/admin” (lokalean bagabiltza) jarrita, domeinuaren ostean “admin” dagoenez “admin.site.urls”-era bidaliko duela. Hau da, espresio erregularrak erabiltzen ditu patroiak bilatzeko. Adibidez urlpatterns horretan gehitzen bada URL pattern bat:

```
urlpatterns = [  
    url(r'^admin/', include(admin.site.urls)),  
    url(r'^tokia/(?P<tokia_id>[0-9]+)/$', myView.tokia),  
]
```

Nabigazio barran “localhost/tokia/4” jarriz gero, myView.tokia exekutatuko du. Aurreko 3.1 irudian ikusi den bezala, view hauek negozio logikari dagozkie. Negozio logikak bere lana egiten duenean behar duen txantiloari pasako dio informazioa.

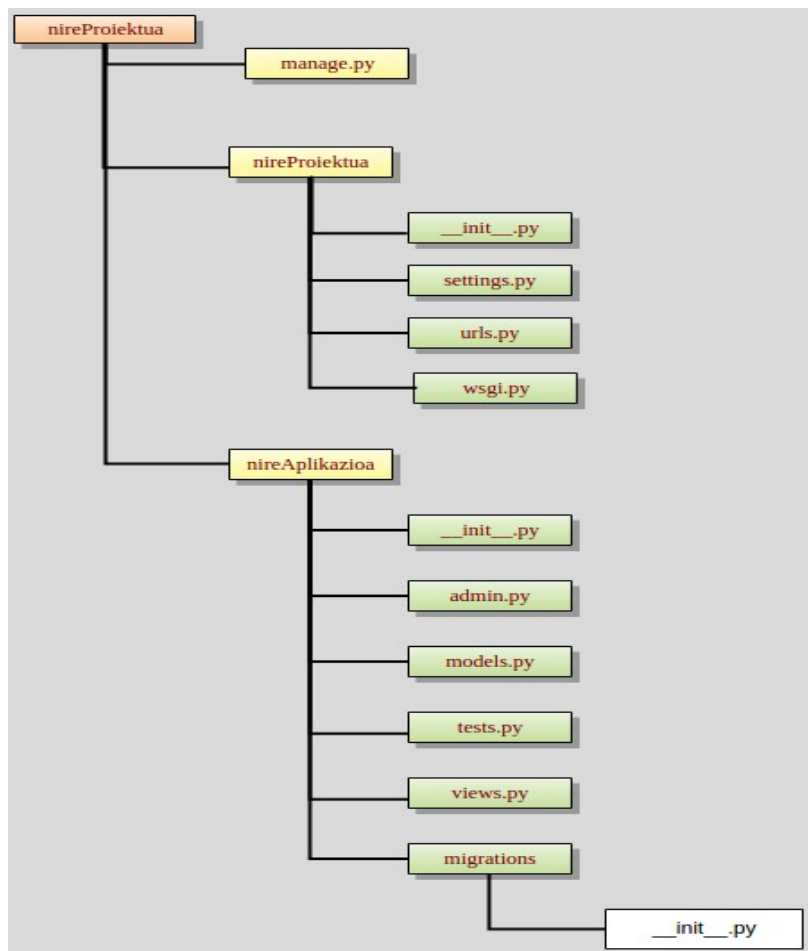
Laburbilduz, nabigazio barran idazten diren URLak urls.py fitxategiko listan espresio erregular bidez bilatzen saiatzen da eta lehenengoa bilatzen duena exekutatuko du.

Behin proiektua sortuta dagoela, aplikazioa sortu daiteke:

```
$ python manage.py startapp nireAplikazioa
```



### 3.3 irudian ikus dezakegu fitxategien egitura berria.

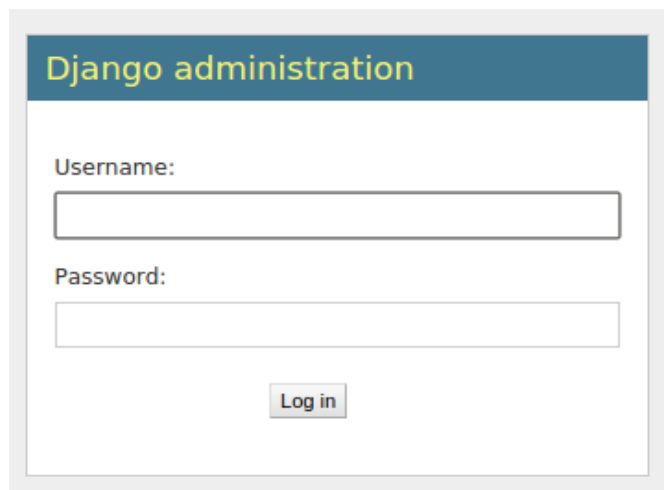


**3.3 Irudia-** Django proiektu baten egitura aplikazio batekin

“nireAplikazioa” izeneko fitxategia sortu du proiektu barruan, eta honek aplikazio batek behar dituen fitxategiak ditu. Aipagarrienak: admin.py, models.py eta views.py dira.

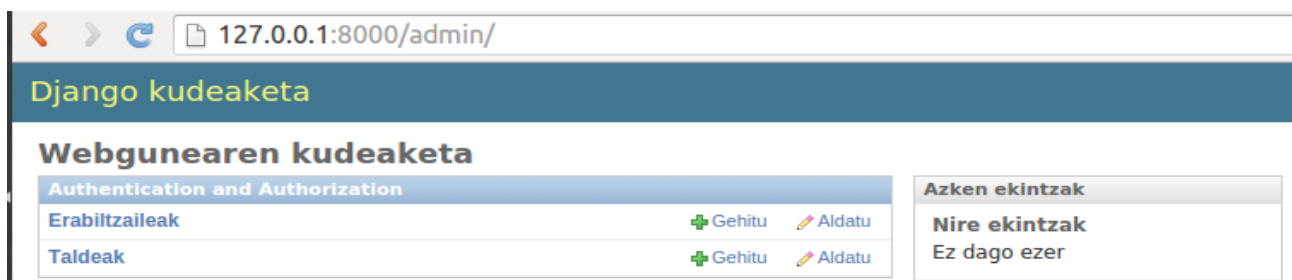
- Admin.py

Hemen sistemako administrazio gunean agertuko diren datu baseko ereduak erregistratu behar dira. Lan inguruneak hasieratik administrazio panel bat ekartzen du eginda eta hemen sartu al izateko super-erabiltzaile bat sortzeko eskatzen du. Erabiltzaile honekin aplikazioaren kontrola eramaten den orrialdeetara sar daiteke (ikusi 3.4 irudia).



**3.4 Irudia** - Django administrazio gunean sartzeko logina

Behin sistemako administrazio gunean sartuta, `admin.py` fitxategian idatzitako ereduak agertuko dira fitxategi berdinean adierazitako moduan (ikus 3.5 irudia). Inplementazioaren atalean ikusiko da nola geratu den emaitza.



**3.5 Irudia** - Django administrazio gunea

- `Models.py`

Hementxe erazagutuko dira datu baseko ereduak. Django berak kontrolatzen ditu *foreignKey* eta abarrak. Adibide simple bat ikusteko, bi klase nola lotzen diren:

```
class Galdera(models.Model):  
    text = models.CharField(max_length=100)  
class Aukera(models.Model):  
    galdera = models.ForeignKey(Galdera, on_delete=models.CASCADE)  
    bozkak = models.IntegerField(default=0)
```

Bi eredu sortu dira, alde batetik 'Galdera' eredu bere atributuetan testua daukana. Eta bestetik 'Aukera' eredu galdera bati lotuta dagoena, bozka kopurua atributuaz aparte. Modu honetan, eroso baten igaro daiteke aukeratik bere galderara eta alderantziz ere. Aplikazioko ereduak nola erazagutu diren ikusteko joan E3 eranskinera.

Behin gure models.py fitxategia beteta daukagula, manage.py erabiliz sortuko dira datu basean taulak (settings.py-n jarri dugun datu basean).

- Views.py

Hemen kokatzen da Djangoren negozio logika, fitxategi honetan joango dira da txantiloiak eta datu basearen bitartekari egiten duten metodoak (pythonez idatziak). Hemengo metodoen izenak jarri behar dira aipatutako urls.py fitxategian.

### **3.3- Datu baseak**

---

Produktuan bi datu base daude, alde batetik datu base zentrala MySQL da eta bestetik, datu meatzaritzarako datuak gorde diren MongoDB.

- MySQL

Datu base mota hau erlazionala da eta tauletan antolatzen da, taulak elkarren artean erreferentzia egin al izanez.

MySQL datu baseaz Django aplikazioa arduratzen da. Proiektuko settings.py fitxategian ezarri da beharrezko konfigurazioa Djangok datu base honekin konexioa egin dezan. Datu basea Apache-ren bidez instalatu eta martxan jarri da. Behin models.py fitxategian datu baseak izan behar dituen ereduak definituak daudela, Djangok MySQL datu basean beharrezkoak zaizkion taulak sortuko ditu. Datu base hau proiektua garatzean ikusezina izango balitz bezala da, abstrakzio maila handitzen laguntzen duelako lan inguruneak.

- MongoDB

Dokumentuetan oinarritutakoa da, datu basea dokumentuetan sailkaturik dago eta hauek BSON formatuan gordetzen dira (JSON formatuan oinarrituak). Ezaugarri

nagusiak eraginkortasuna, datu ereduaren sinpletasuna eta eskalagarritasuna dira.

Adibidez, Mongo-ko fitxategi bat ondorengo dokumentuz osatuta egon daiteke:

```
user_doc = {  
  "username" : "janedoe",  
  "firstname" : "Jane",  
  "surname" : "Doe",  
  "dateofbirth" : datetime(1974, 4, 12),  
  "emails": [{"email": "foouser1@example.com", "primary": True},  
             {"email": "foouser3@example3.com", "primary": False }]  
  "score" : 0  
}
```

Datu base honetan txertatu, ezabatu, bilatu eta eguneratzeko arazorik ez dago, Mongo-ren sintaxia erabiliz noski.

Kasu honetan ez da Djangoren esku utzi datu base hau. Hemengo dokumentuak aldatzen arduraduna gehienbat datu meatzaritzako script-ak dira, aurrerago ikusiko direnak. Django aplikaziotik bi elkarrekintza egingo dira Mongo-ra. Alde batetik erabiltzaileak gustuko duen zerbait adieraztean behar den tokian jarriko da, eta bestetik gomendioen kontsulta soilik.

# 4

---

---

## **Sistemaren funtzionamendua. Betekizunen azterketa egoera erralak erabilia**

Dokumentuko atal honetan sistemaren funtzionamendua nolakoa izan nahi den hobeto ulertzeko, adibide praktiko batzuk ikusiko dira txantilo batzuen bidez aplikazioaren erabilgarritasuna islatzeko. Horrela sistemak edukiko dituen betekizunak hobeto ulertuko dira eta aplikazioaren *front-end* zatia nolakoa izango den zehaztuko da. Aplikazioak erantzuna eman behar izango dien kasu praktiko batzuk hauek izan beharko dira:

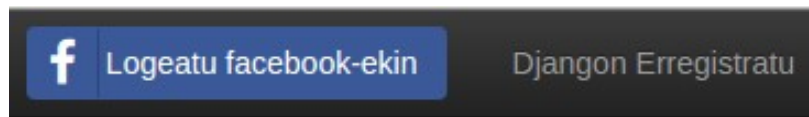
1. Erabiltzaile berriak egin beharreko lehen pausuak sisteman.
2. Familia bat turismo bulegora informazio eske.
3. Bizkaitar bat Donostiara hiria ezagutzen.
4. Herri txiki bateko kanpin arduraduna inguruko tokiak 'saltzen'.
5. Gazte talde bat aste buruan zer egin jakin gabe.
6. Turismo bulegoko administratzailearen ikuspuntutik.

### **4.1- Kasu praktikoa 1: Erabiltzaile berriak egin beharreko lehen pausuak sisteman**

---

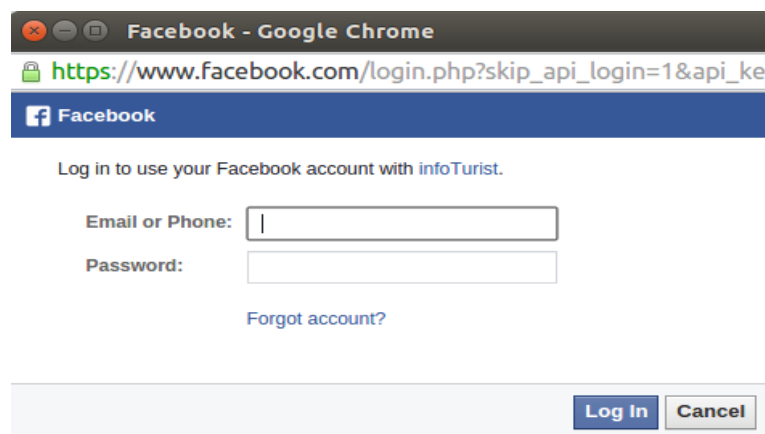
Sisteman lehen aldiz sartzen den pertsona batek hainbat kontsultak egiteko aukera

izango du. Baina gomendagarriena da lehendabizi erregistratzea, iritziak eman al izateko eta sistema gutxika ikasten joateko.



#### 4.1 Irudia- Erregistratzeko bi aukerak

Erregistratzeko bi modu egongo dira (ikusi 4.1 irudia), alde batetik betiko erregistroa: erabiltzaile izena, helbide elektronikoa eta pasahitza sartuz. Bestetik Facebook bidezko erregistroa: Facebook<sup>8</sup>-eko kontuarekin sartzea sisteman (ikusi 4.2 irudia).



#### 4.2 Irudia - Facebook bidezko logina

Lehen aldian erregistratu bezala sartzean, probintzi eta toki batzuk ebaluatzeko eskatuko du sistemak. Gutxienez aukeretako bat ebaluatu beharko da (ikusi 4.3 irudia).

Ez daukagu zure gustoko tokien informaziorik, lagun gaitzazu zure interesekoak direnak identifikatzen  
Aukeratu zuretzat interesekoak direnak

Probintziak:

Bizkaia  
Balorazioa ☆☆☆☆☆



Gipuzkoa  
Balorazioa ☆☆☆☆☆



#### 4.3 Irudia - Lehen aldiz login egitean eskatutako informazioa


8 <https://www.facebook.com/>

Gustuko bezala adierazitakoak geroago ikusteko aukera egongo da profileko 'Nire gogokoak' atalean. Profileraren sartuz gero, iritziak hedatzeko aukera egongo da. Esaterako 'Bizkaia' 5 izarrekin baloratu bada eta gainontzekoak ez badira ebaluatu, profilean baloratu gabeko probintziak eta gustuko probintzietako ebaluatu gabeko herriak agertuko dira ebaluatzeko. Behin probintzia ebaluatzean, balorazioa 3koa baino handiagoa izan bada gustuko bezala hartuko du sistemak eta beraz bertako herriak ere ebaluagarri jarriko dira. Herriaren ebaluazioa positiboa bada, herriko tokiak agertuko dira gero. Hau da, handitik txikirako ebaluazioa eskatuko da, negatiboa den kasuan bertako barrukoak ere ez intereseko bezala kontuan hartuz.




Adibidea: Gustuko 'Bizkaia' probintzia eta 'Ispaster' herria izanda, ebaluatzeko falta diren probintziak eta gustuko herrietako tokiak (ikusi 4.4 irudia).

Gustoko al dituzu? 1-"Bat ere ez", 5-"Askok!"

Gipuzkoa  
Balorazioa ☆☆☆☆☆



---

<p>Ispasterreko Frontoia Balorazioa ☆☆☆☆☆</p> 	<p>Ispasterreko plaza Balorazioa ☆☆☆☆☆</p> 	<p>Ogella hondartza Balorazioa ☆☆☆☆☆</p> 
<p>Ispasterreko Eliza Balorazioa ☆☆☆☆☆</p>	<p>Ispasterreko eliza berria Balorazioa ☆☆☆☆☆</p>	<p>Autobus geltokia Balorazioa ☆☆☆☆☆</p>

#### 4.4 Irudia - Profilean ebaluazio gunea

Aurreko kasuan 'Gipuzkoari' 5 puntu emanda, Gipuzkoako herriak agertuko lirateke ebaluagarri, bestela ez.

Behin denbora bat igarota (12-24 ordu), datu meatzaritzako scriptak exekutatu eta erabiltzaileak nahiko informazio emanda sistemari, gomendioak agertuko litzaizkioke (bai tokiak eta baita beste erabiltzaile batzuk ere) hasiera orrian eta bilaketak egitean emaitzak bere gustuen arabera egongo dira ordenatuak. Bizkaia gustuko badu eta 'Hondartza' bilatzen, lehendabizi Bizkaiko hondartzak agertuko dira emaitzetan.

Tokiak ebaluatzeko modu gehiagorik ere egongo da eta ezagunena, tokietan edo

jaietan iritzia idaztea da. Hau toki edo jaiaren deskripzio orrian egingo da (ikusi 4.5 irudia) eta 1-5 bitarteko puntuazioa eman ondoren, titulua eta iritzia idazteko aukera egongo da.

Jendearen iritziaik:

MIKEL  
★★★★☆

Oso polita

Ederra da baina oraindik itxita dago

Baliogarria - Ez egokia Iritziaren puntuak: 0 (bozkak: 0)

1

..

Balorazioa  
★★★★☆

Laburpena

Askok gustatu zait

Iritzia

Oso polita da baina duela hile betetik itxita dago

Iritzia eman

#### 4.5 Irudia - Jendearen iritziaik eta iritzia idazteko gunea

Jendearen iritziaik ere ebaluagarria izango dira, beste norbaitek idatzitako iritzia baliagarria den edo ez adieraztea ahalbideratuz. Iritzia iraingarria bada berriz, salatzeo aukera ere egongo da, gero administrariak erabakitzeo ezabatu edo ez.

Erregistratuek ibilbideak sortzeo aukera izango dute, hau sakonago ikusiko da hurrengo puntuan.

## 4.2- Kasu praktikoa 2: Familia bat turismo bulegora informazio eske

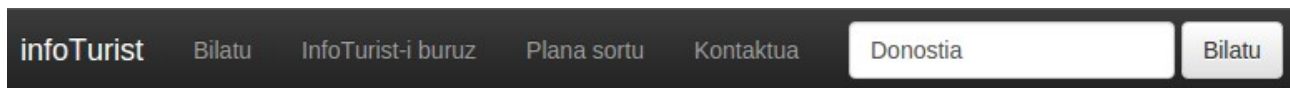
---

Demagun familia bat Donostiara datorrela eta turismo bulegora informazio eske doala. Hemen, informazio bulegoko langileak bere memoria erabili beharrean gomendioak egiteko, web aplikazioa erabili dezake. Familiako inor aurretik ibilita ez badago, informazio bulegokoak Donostiako tokirik baloratuenak gomendatu ditzake eta nahi



izanez gero, zer egiteko asmoa daukaten jakinda etiketekin tokiak filtratu.

Adibidez hondartzak bilatu nahi izanez gero lehendabizi helburu herriaren informazioan sartu beharko litzateke erabiltzailea (kasu honetan informazio bulegoko arduraduna), nabigazio barran (esaterako) herriaren izena idatziz eta bilatu botoian zapalduz (ikusi 4.6 irudia).



#### 4.6 Irudia - Nabigazio barraren zati bat, bilaketa kutxan 'Donostia'

Bilatu botoia zapaltzean, bilaketa kutxan idatzitakoa aurkitzen saiatuko da sistema, herrien izenetan, tokien izenean eta jaien izenean ere. Lehendabizi erabiltzaileari gustuko zaizkionak bueltatuko zaizkio baldin eta emaitza asko posible badira. Bilaketa kutxa horretan hitz bat baino gehiago ere idazteko aukera egongo da, adibidez 'Donostia hondartza' eta bi hitzak bere aldetik bilatuko lituzke sistemak. Kasu praktiko honetan, herri bat eta bi toki erakusten dira (ikusi 4.7 irudia).

Herriak:



Donostia-SS - Gipuzkoa

Deskribapena:

Donostia Gipuzkoako kostaldean kokatuta dagoen hiriburua da.

Tokiak:



Donostia Kursaal -Donostia-SS

Donostiako Kursaal Kongresu Jauregia, Rafael Moneoren obra arkitektoniko ikusgarri bat da. Kantauri itsaso parean kokatuta, hiriko kultura eta kongresu aktibitateen erdigunea da eta urtero 200 ekitaldi eta 500.000 ikusle baino gehiago biltzen ditu. Beira zeharrargitsuzko bi kubok osatzen duten Kursaalak punta-puntako teknologiarekin hornitutako espazio balio-aniztunak ditu: 1.806 pertsonentzako edukiera daukan Auditorioa, bilera aretoak (15etik 575 pertsonako edukierarekin) eta erakusketa eremuak. Era guztietako kongresu, konbentzio, jarduera, azoka, kontzertu eta ikuskizunak antolatzeko toki berezi eta aproposa da. Gainera, 15 urtetako esperientzia dauka dagoneko. Eskaintza bereizgarri eta bitxia sukaldaritza zerbitzua da: taldeko burua Andoni Luis Aduriz da eta arrakasta ziurtatuta dago "Michelin bi izar" dituen sukaldari honen esperientzia eta sormenaren bermearekin.

Etiketak

[Hondartza](#) , [Jatetxea](#) , [Autobusa](#)



Bermatutako tokiak



Fnac donostia -Donostia-SS

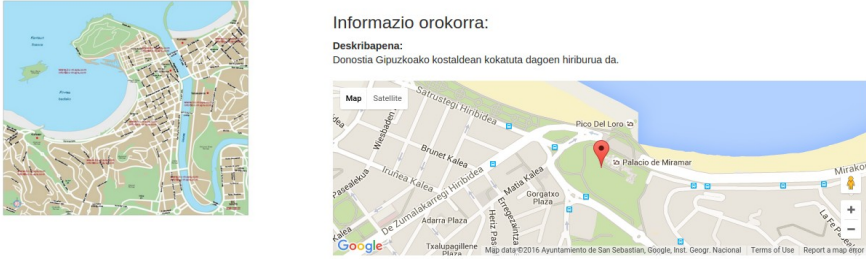
FNAC-ek Donostia aukeratu zuen bere Euskal Erkidegoko lehen denda irekitzeko 2005ean. San Martin Merkataritza Zentroan kokatuta, denda gune

#### 4.7 Irudia -'Donostia' hitzaren bilaketa

Donostia herria aukeratuta, honen informazio orrira eramango du non Donostiako toki guztiak egongo diren orrietan sailkatuta (lehenengo postuetan herriko tokirik baloratuenak izanik). Pantaila honetan beste bilaketa kutxa bat egongo da, kasu honetan filtro gehiagorekin. Bilaketa kutxa berri honetan idatziz gero, herriko toki eta jaiak soilik bueltatuko ditu (Donostiakoak kasuan) eta zerbitzuak aukeratuta, toki horiek gutxienez aukeratutako zerbitzu bat izan beharko dute (ikusi 4.8 irudia).

Herria: Donostia-SS - Gipuzkoa

Informazio orokorra:  
Deskribapena:  
Donostia Gipuzkoako kostaldean kokatuta dagoen hiriburua da.





Bilaketa herrian: Donostia-SS

Hondartza,  Jatetxea,  Gaua pasatu,  Denda,  Unibertsitatea,  Taberna,  Taxiak,  Trena,  Autobusa.

Bilaketa herrian...

Herriko tokirik baloratuenak:

	<b>Kontxa</b> -Donostia-SS Tralari tralara kursala donostiakoa3 Etiketak <input type="checkbox"/> Hondartza, <input type="checkbox"/> Taberna, <input type="checkbox"/> Autobusa	<input type="button" value="EDITATU"/>
	<b>Donostia Kursaal</b> -Donostia-SS Donostiako Kursaal Kongresu Jauregia, Rafael Moneoren obra arkitektoniko ikusgarri bat da.	<input type="button" value="EDITATU"/>

#### 4.8 Irudia - Donostia herriaren orrialdea

Bilaketa kutxa hutsik utziz gero, zerbitzuak dituzten tokiak bueltatuko dira, herriko jaiekin batera. Bestalde bilaketa kutxan zerbait idatziz gero emaitza askoz ere murriztuagoa agertuko da. Bilaketa herri baten deskribapenetik egiten bada, emaitzen azpian beste bilaketa bat egiteko aukera ematen duen botoia agertuko da (ikusi 4.9 irudia).

Ez da bilatu jairik

Aurkitutako tokiak:



Puntuazioa: Puntuaziorik gabe (0)

### Ondarreta hondartza - Donostia-SS

Ondarreta hondartza donostiako antiguon dago kokatuta  
Etiketak  
📍 Hondartza , 📅 Gaua pasatu

Beste bilaketa bat egin: Donostia-SS

## 4.9 Irudia - Donostia herrian bilaketa emaitza

Funtzionalitate honekin beraz, turismo bulegoko arduradunak bisitariei gomendioak emateko gai izango lirateke.

Gainera, informazio bulegoko arduradunak 'ibilbide publiko' bat sortzeko aukera izango dute (erregistratu arruntek ere) toki hauekin gero familiarekin partekatzeko.

ibilbidea ondo gorde da, orain tokiak sartzen hasi zaitezke

Jarri izena eta deskribapena ibilbideari. Gero tokiak sartzen hasteko prest egongo zara

Izena\*

XX Familiarentzako Donostia ibilbidea

Deskribapena\*

Donostiako kosta ezagutzeko ibilbidea

Pribatua

Sortu



Nire ibilbideak:

- **Ispastar** 🗺️ Ezabatu ibilbidea
- **Donostia** 🗺️ Ezabatu ibilbidea
- **XX Familiarentzako Donostia ibilbidea** Ezabatu ibilbidea







## 4.10 Irudia - Ibilbide bat sortu berri

Hasieran ibilbideak hutsik egongo dira eta tokiak sartu beharko dira ordenan. Sistemak gomendatutako tokiak agertuko zaizkio ibilbide listaren behealdean modu eroso batean tokiak sartzeko, nahi izanez gero (ikusi 4.11 irudia).

Nire ibilbideak:

- **Ispastar**  Ezabatu ibilbidea
- **Donostia**  Ezabatu ibilbidea
- **XX Familiarentzako Donostia ibilbidea** Ezabatu ibilbidea

Gomendatutako tokiak


<p><b>Ispasterreko Frontoia</b></p> <p>Puntuaziorik gabe Donostia</p> <p>Ibilbidean gorde</p> 	<p><b>Ogella hondartza</b></p> <p>Puntuaziorik gabe XX Familiarentzako Donostia ibilbidea</p> <p>Ibilbidean gorde</p> 	<p><b>Kontxa hondartza</b></p> <p>Puntuaziorik gabe Ispastar</p> <p>Ibilbidean gorde</p> 
<p><b>Zurriola hondartza</b></p> <p>Puntuaziorik gabe Ispastar</p> <p>Ibilbidean gorde</p> 	<p><b>Kontxa</b></p> <p>★★★★★ Ispastar</p> <p>Ibilbidean gorde</p> 	<p><b>Ispasterreko plaza</b></p> <p>Puntuaziorik gabe Ispastar</p> <p>Ibilbidean gorde</p> 

#### 4.11 Irudia - Ibilbideen azpian gomendioak ibilbideetan sartzeko

Kasu honetan Donostiako kosta ezagutzeko ibilbidea denez, toki hauek bilatu eta sartu beharko dira. Horretarako, tokien deskribapenean banan banan sartu eta ibilbidera gehituko dira ordenan.



Herria: Donostia-SS

Tokiko etiketak:  
 Jatetxea ,  Taberna

Gehitu tokia zure ibilbideetara

XX Familiarentzako Donostia ibilbidea




Norek sartu: mikel

#### 4.12 Irudia - Haizearen orrazia Ibilbidean sartzeko botoiarekin

Ibilbidean toki guztiak sartu ostean, ibilbidean zeintzuk toki dauden ikusteko aukera egongo da (ikus 4.13 irudia).

Ibilbidea: XX Familiarentzako Donostia ibilbidea

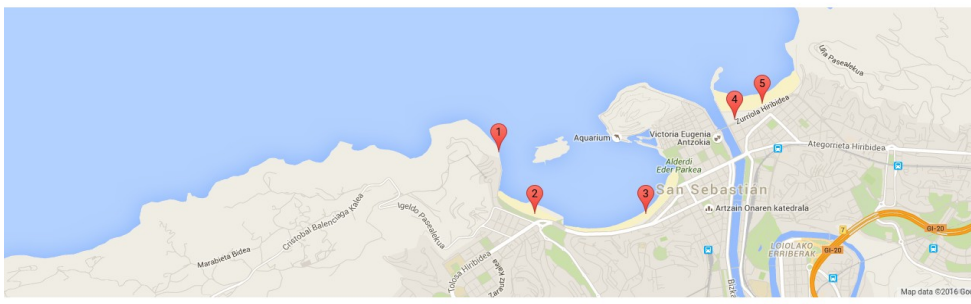
Ibilbideko tokiak:

1		<p><b>Ondarreta hondartza</b> - Donostia-SS</p> <p><b>Deskribapena:</b> Ondarreta hondartza donostiako antiguon dago kokatuta</p> <p><a href="#">Ibilbidetik hemendik hasi</a></p>	Ibilbidetik ezabatu
2		<p><b>Kontxa hondartza</b> - Donostia-SS</p> <p><b>Deskribapena:</b> Donostia erdiko hondartza</p> <p><a href="#">Ibilbidetik hemendik hasi</a></p>	Ibilbidetik ezabatu
3		<p><b>Zurriola hondartza</b> - Donostia-SS</p> <p><b>Deskribapena:</b> Donostian gros aldean dagoen hondartza da hau. Ekialderen kokatuta</p>	Ibilbidetik ezabatu



### 4.13 Irudia - Ibilbide baten edukia

Gainera erabiltzailearentzat ibilbideak hartzen duen azaleraren eta ze puntu igarotzen dituen hobeto ikusteko, ibilbidearen detaileko orrialdean mapa bat egongo da eta mapa horretan ibilbideko puntuak agertuko dira markatuak etiketa bezala ibilbideko posizioa adieraziz (ikusi 4.14 irudia).

Ibilbidea: XX Familiarentzako Donostia ibilbidea



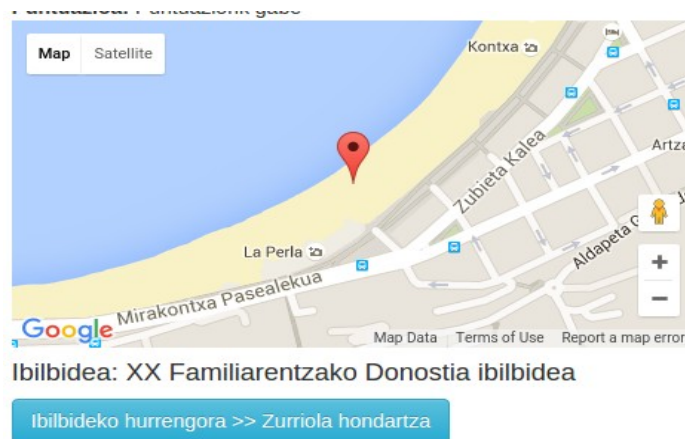
Ibilbideko tokiak:

1		<p><b>Haizearen orrazia</b> - Donostia-SS</p> <p><b>Deskribapena:</b> Eduardo Chillida eskultoreak (Donostia, 1924-2002) eta Luis Peña Ganchequi arkitektoak (Oñati, 1926 - Donostia, 2009) egina, Haizearen Orrazia edertasun hutsaren erakusgarri ezin hobea da, artearen eta paisaiaren arteko elkarriketa sakona. Itsasoaren indarra eta burdinazko egituren sendotasuna batuta, artelan garaikide naturala ez ezik elkarretzeko eta atsedean hartzeko gune ere bada obra hori donostiar nahiz turistentzat.</p> <p><a href="#">Ibilbidetik hemendik hasi</a></p>	Ibilbidetik ezabatu
2		<p><b>Ondarreta hondartza</b> - Donostia-SS</p> <p><b>Deskribapena:</b></p>	Ibilbidetik ezabatu

### 4.14 Irudia - Ibilbidearen mapa

Informazio bulegoko arduradunak ibilbidea publiko bezala sortuta, edonork ikusi eta erabili dezake, baina sortzaileak bakarrik izango du tokiak gehitzeko eta ezabatzeko

baimena. Ibilbidea gainera, ez da zertan hasieratik hasi beharrik 'Ibilbidea hemendik hasi' botoia zapalduz ibilbideko puntu horretatik hasiko da. Ibilbidea hastean, tokien deskribapenak agertuko dira (aurretik ikusitakoak) baina kasu honetan maparen azpian ibilbidearen izena eta hurrengo tokira joateko botoia agertuko dira (ikusi 4.15 irudia).



**4.15 Irudia** - Ibilbideko toki bat eta hurrengora joateko botoia

Ibilbidearen bukaerara iristean hurrengo tokira joateko agertzen den botoiaren tokian mezu bat agertuko da ibilbidearen bukaerara iritsi dela adierazten.

Laburbilduz, turismo bulegoko arduradunek modu ezberdinetan lagundu al izango die turistei informazioa emanez. Alde batetik kontsulta ezberdinak eginez eta bestetik ibilbideak sortuz eta publiko jarritz besteak beste.

### **4.3- Kasu praktikoa 3: Bizkaitar bat Donostiara hiria ezagutzen**

---

Demagun bizkaitar batek webgunean Bizkaiko tokiak ebaluatu dituela eta bere garaian Gipuzkoa ez gustuko bezala ebaluatu zuela, ondorioz, ezin izango ditu Gipuzkoako gomendioak ikusi? Ez da horrela. Nahiz eta ez gustuko bezala markatu izan baliteke gomendioren bat sortzea probintzia horretakoa.

Adibidearekin jarraituz, Bizkaiko herriak gustuko eta Gipuzkoa ez intereseko bezala markatu duen pertsona batek Donostiarako plan bat eskatzen badu, seguruenik gomendatutako tokiak bueltatuko dizkio eta bestela, Donostiako tokirik

interesgarrienak. Gomendio hauek erabiltzaileak gustuko dituen tokietatik ateratakoak izango dira. Pertsona batek 'Hondartza' etiketa daukaten tokiak gustuko baditu, beste probintzia bateko tokietako etiketa hori dutenak gustuko izateko probabilitatea altua da.

Beraz, Bizkaitar honek Donostian nora joan ez badaki, plan bat sortzeko esan al dio sistemari, egun batzuetarako (jaiak bilatzeko) edota eskaera egiten den egunerako (ikusi 4.16 irudia).

#### Plan pertsonalizatua sortu:

**Probintzia:**

**Herria:**

Egun batzuetarako plana    Gaurkorako plana




Noiztik:    Noizra:

Hondartza,  Jatetxea,  Gaua pasatu,  Denda,  Unibertsitatea,  Taberna,  Taxiak,  Trena,  
 Autobusa,

#### 4.16 Irudia - Plana sortu interfazea

Sistemak filtro horiek betetzen dituen toki eta jaiak bueltatzea espero da, nahiz eta Gipuzkoa probintzia ez gustuko bezala markatuta eduki. Bi emaitza posible egongo dira, alde batetik gomendatutako tokiak bueltatzea, eta bestetik, nahiko toki gomendatzeko ez daukalako herriko tokirik baloratuenak bueltatzea (ikusi 4.17 irudia).

Informazio gutxiegi daukagu zutaz. Herriko TOP tokiak bueltatu ditugu

 Puntuazioa: ★★★★★☆ (4,0)	<h3>Talasoterapia Zelai -Zumaia</h3> <p>Zumaiaiko talaso bakarra seuruna Etiketak 📍 Hondartza , 🗺️ Jatetxea , 🏠 Gagua pasatu</p> <p>✓ Bermatutako tokiak</p>
 Puntuazioa: ★★★★★☆ (4,0)	<h3>Luisen etxea -Zumaia</h3> <p>Luis bizi den tokiak Etiketak 🏠 Gagua pasatu</p>
 Puntuazioa: ★★★★★★ (5,0)	<h3>San telmo ermita -Zumaia</h3> <p>Zumaian dagoen san telmoren ermita Etiketak 🏠 Gagua pasatu</p>

#### 4.17 Irudia - Plana sortuk gomendio 1 eta herriko TOP-ak bueltatu

Ondorioz, adibideko Bizkaitarrak ez luke arazorik izango Donostiako gustuko izango dituen tokiak ezagutzeko. Gainera toki interesgarri baten deskribapenean sartuta, tokiarekin lotutako beste intereseko tokiak agertuko dira, eta hauen barruan beste batzuk... (ikusi 4.18 irudia).

Iritzia

Iritzia...

Iritzia eman

Interesekoa izan daiteke zuretzat...

Fnac donostia id:214



Talasoterapia Zelai id:216



Tolosako frontoia id:218



#### 4.18 Irudia - Toki bateko deskripzioaren oina



## 4.4- Kasu praktikoa 4: Herri txiki bateko kanpin arduraduna inguruko tokiak 'saltzen'

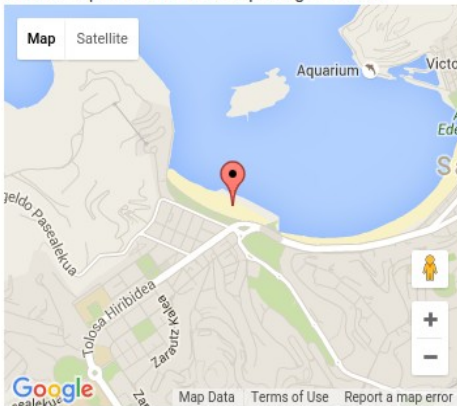
Demagun herri txiki bateko kanpin arduradunak bertaraten diren turistentzako ibilbideak eta toki interesgarriak antolatu nahi dituela. Ibilbideak 2. puntuan adierazi bezala sortu eta antolatu ditzake gero beste pertsonekin partekatzeko. Tokiak sartzeko ordea, erabiltzaile\_izena->tokia sartu interfazera joan beharko litzateke. Sisteman sartu nahi duen herria agertzen ez bada, kontaktu bidez bere probintziako administratzaileei edo web administratzaileari mezua bidaltzeko aukera izango du, herria gehitu dezaten.

Behin tokia sartzeko interfazean egonda, era erosoan mapan tokia non dagoen aukeratu al izango du klik bat eginda eta horrela, latitudea eta longitudea adierazita geratuko dira. Gero tokiaren gainontzeko informazioa sartu beharko da, ze herritan dagoen, izena, deskribapena (euskaraz gutxienez), irudia eta bertako zerbitzua(k).

### Toki berri bat sartu

Non dago toki berria?

Klikatu mapan toki berria kokapena gordetzeko



Latitude: 43.31643554010213  
Longitude: -2.000855505466461

Ze herritan\*

Donostia-SS

Izena\*

Ondarrata hondartza

Deskribapena\*

Ondarreta hondartza Donostiako mendebalderen dagoen hondartza da...

Latitudea

43.31643554010213

Longitudea

-2.000855505466461

Irudia\*

Choose File No file chosen

Eskeinitako zerbitzuak\*

- Hondartza
- Jatetxea
- Gaua pasatu
- Denda
- Unibertsitatea
- Taberna
- Taxiak
- Trena
- Autobusa

submit

### 4.19 eta 4.20 Irudiak - Tokia sartu interfazeak

Tokia sartuta, publiko egongo da eta zonako (probintziako edo herriko) administrariak bermatu edo editatu al izango du. Sartu duen erabiltzaileak ere berak sartutako tokiak editatzeko aukera izango du.

Beraz herri txikiko kanpin arduradun horrek bere inguruko toki interesgarriak sartzeko aukera izango du eta turistei informazio hori adierazteko tresna ona izango da webgunea.

## 4.5- Kasu praktikoa 5: Gazte talde bat aste buruan zer egin jakin gabe

---

Udan sarritan gertatzen den egoera da hau, gazte talde bat, aste buruan edo egun batzuetan zer egin jakin gabe. Toki berriak ezagutzeko aurreko puntuetan dauden moduez aparte, jaiak bilatzeko aukera ere egongo da gertutasunaren arabera. Horretarako, orrialde nagusietan agertzen den bilaketa gunean egin beharko da (ikusi 4.21 irudia). Bilaketa gertutasunarekiko egiteko nabigatzaileari kokapena partekatzeko esan beharko zaio (orrialdean sartzen lehen aldiz agertuko da mezua).

Distantzia kilometrotan jarri, zer bilatu nahi den (jaiak, herriak edota tokiak) eta nahi izanez gero data tarte bat (jaiak bilatzeko egokia).

Bilatu dataka

Non:   Km Noiztik:  Noizra:

Zer nahi duzu bilatu?  
Hurbiltasunagatik bada, baimendu kokapena

Jaiak  Herriak  Tokiak

### 4.21 Irudia - Bilaketa dataka eta hurbiltasunaz

Inguruko toki eta jai berriak bilatzeko modu aparta izango da hau, eta aipatutako gazte taldeak hurrengo egunetan zer egin daitekeen jakin al izango du.

Bestalde, probintzian bilatu nahi badituzte hurrengo jaiak, hasiera orrian kontsultatu al izango dira hauek.

## 4.6- Kasu praktikoa 6: Turismo bulegoko administrariaren atazak

---

Erabiltzaile hauek edozein erregistratuk baino gehiago egin al izango dute sisteman. Hasteko, herri edo probintziaren bateko administraria izango da eta beren konpetentzia diren herrietako jaiak sartzeko aukera izango du (herriekin egiten den antzera). Jai hauek urteroko jaiak izan daitezke edo jai puntualak eta hau, jai sortzerakoan adierazi beharko da (*checkbox* batean). Urtero errepikatzen diren jaiak adibidez: Herriko jaiak, danborrada eguna, San Juan sua jaiak, ... eta jai puntuala izan daiteke: Ibilaldia 2016, Kontzertua herriko plazan, ...

Jaiak sartzeaz gain, salatutako iritziak administratzeaz arduratu beharko du bulegoko arduradunak. Erabiltzaileek salatzen dituzten iritziak beren zonako tokietakoak badira, administrazio panelean (ikusi 4.22 irudia) agertuko zaizkie salaketa horiek: nork egin duen, zein iritziri eta iritziaren edukia. Baldin administratzaileari iritzi egokia iruditzen bazaio, dagoen bezala utzi, edo ez egokia dela iruditzen bazaio ezabatzeko aukera izango du.

Zure zonako iritzi salatuak:

Tokien iritzi salaketak:

---  
Norek salatu: 1->mikel      Iritzia: Ondo dago - =)      Ze tokiri: Fnac donostia      Mezua ezabatu      Mezua egokia da

Jaien iritzi salaketak:

---  
Norek salatu: 5->user3      Iritzia: Oso eskasa - oso eskasa. T#@#-#k!      Ze jairi: Tanborrada eguna      Mezua ezabatu      Mezua egokia da

### 4.22 Irudia - Administrazio panela - iritziaren administrazioa

Administrariak egin beharreko beste eginkizun bat, tokietako informazioa bermatzea izango da. Beraiek administratzen duten herri edo probintzietako tokiak soilik bermatzeko boterea izango dute. Horretarako administrazio panelean 'zona administratu' atalera joan beharko dira eta bertan agertuko zaizkie bermatu gabeko tokiak beraiek administratzen duten tokietan (ikusi 4.23 irudia).

Administratzen dituzun probintziak:

- Bizkaia
- Gipuzkoa

Bermatzeke dauden tokiak zure zonan:



Puntuazioa: Puntuaziorik gabe (0)

### Ispasterreko Frontoia -Ispaster

Ispasterren dagoen frontoi irekia

Etiketak

Taberna



Puntuazioa: Puntuaziorik gabe (0)

### Ispasterreko plaza -Ispaster

Ispaster erdian dagoen plaza

Etiketak

Jatetxea , Autobusa



### Onella hondartza -Ispaster

#### 4.23 Irudia - Administrazio panela - tokiak bermatu

Agertzen diren tokiren bateko informazioa okerra bada, editatzeko aukera izango dute eta horrela informazioa bermatuta geratuko da. Bermatzeko beste aukera bat tokiaren deskribapenean egonda agertuko den “Bermatu. Informazioa egokia da” botoian klik egitea da (ikusi 4.24 irudia).

### Tokia: Ispasterreko Frontoia -Ispaster

Bermatu. Informazioa egokia da.



Informazio orokorra:

Deskribadena:

**4.24 Irudia** - Tokiko deskribapena administratzaile ikuspuntutik. 'Bermatu' botoiarekin

Erabiltzaile mota honek egin al izango duen azken gauza, mezuak erantzutea izango da. Webgunean kontaktatzeko aukera egongo da eta hemen, aukeratu egiten da nori bidali nahi zaion mezua (probintziaren bateko administratzaileei edo web administratzaileari). Baldin norbaitek beren probintziako administratzaileari mezua bidali badio, administrazioa panelean 'Mezuak ikusi' atalean agertuko zaizkie (ikusi 4.25 irudia). Webgunetik bertatik ez da aukerarik egongo erantzuteko baina helbide elektronikoa ofiziala izango dutela espero denez (azken finean turismo bulegoko langileak dira), bertatik erantzun al izango dute.

Zure zonako kontaktuak:

Erantzun gabeko mezuak: 2 Norek bidalita: Galdera	Bere emaila: Tralari@hotmail.com	Mezua: nire galdera daaa...	Markatu erantzunda bezala
Norek bidalita: Haitz	Bere emaila: haitz@hotmail.com	Mezua: Zergatik ez da lgerdo herria agertzen??	Markatu erantzunda bezala

#### 4.25 Irudia - Administrazio panela - mezuak ikusi

Behin mezua erantzun dutela 'Markatu erantzunda bezala' botoiak klikatuta listatik kenduko da mezua hori.

Hau da administrariak egin al izango duten ezberdintasuna gainontzeko erabiltzaileekin konparatuz. Noski, hauek ere tokiak sartzeko aukera izango dute informazioa bermatuta edukita hasieratik.

# 5

---

---

## Analisi eta Diseinua

Sisteman hainbat gauza egin daitezke baina erabiltzaile denak ez daukate gauza guztiak egiteko ahalmena. Lau rol ezberdin bereizten dira sistemaren erabiltzaileetan: Ez erregistratua, erregistratua, zona administraria (herrikoa edo probintziakoa) eta sistemako administraria.

### **5.1- Erabilpen kasuak**

---

Jarraian, rol bakoitzak egin ditzakeen erabilpen kasuak deskribatuko dira beren sekuentzi diagramekin. Lehen 3 rolen erabilpen kasuak ikusi nahi izanez gero, joan 5.2 irudira, laugarren rola sistemaren kontrol guztia daukalako ez da sartu grafikoan.

#### **5.1.1- Lehen rola: erabiltzaile ez erregistratua**

Sisteman lehen aldian sartzen den erabiltzaileak, ez erregistratu bezala egingo du eta honen funtzionalitateak murriztuak egongo dira sisteman. Kotsultak egiteko gai izango da era eta toki ezberdinetan: nabigazio barran, gertutasunaren arabera, herri/probintzietan edota zerbitzuen bidezko filtroak erabiliz. Agertzen zaizkion emaitzen (jai, toki, herri edota probintzi) deskribapenean ere sartu al izango da eta jendearen iritziak irakurtzeko aukera izango du.

Hasiera orrian jendeak ikusitako azken tokiak agertuko zaizkio herririk baloratuenekin batera eta hurrengo jaiekin.

Bi modu ezberdinez erregistratzeko aukera izango ditu, alde batetik bere Facebook-eko kontua erabiliz eta bestetik, sisteman izena emanda. Azken hau aukeratzen badu, bere helbide elektronikora bidalitako estekan sartu beharko da kontua aktibatuta gera dadin.

Administrari mota ezberdinekin kontaktuan jartzeko aukera izango du, bai probintzietako administrariekin eta baita web administrariarekin ere. Horretarako noski, bere zenbait datu eman beharko ditu mezuaz aparte (izena eta helbide elektronikoa).

Sistemako informazioa zein hizkuntzatan atzitu nahi duen aukeratzeko baimena izango du, sistemako hizkuntzen artean aukeratuz. Edozein momentutan aldatu al izango dute hizkuntza hau.

### **5.1.2- Bigarren rola: erabiltzaile erregistratua**

Bigarren rol honetan egoteko, ez erregistratua sisteman identifikatu behar da (login egin) eta sistemaren funtzionalitateak hedatu egiten zaizkio. Behin sistemak erabiltzailea zein den dakiela, bere informazioa gorde eta geroago analizatuko ditu.

Jendeari aurkeztuko zaizkien toki interesgarriak sartzeko eta editatzeko aukera izango du, erabiltzaile mota hau parte-hartzailea izan daiteke sistemarekin beraz. Gainera, sistemako toki eta jaietan iritziak idatzi al izango ditu gainontzeko erabiltzaileek irakur ditzaten. Idatzitako iritzi hau gainontzeko erabiltzaileek baloratu al izango dute egokia edo ez egokia bezala markatuz, eta iritzia iraingarria den kasuetan salatuz. Iritzia idaztean balorazioa egiten da baina badauka beste modurik balorazioak egiteko. Profilean sartu ezkerro, gustuko eta ez gustuko etiketak adierazi al izango ditu eta probintzi, herri eta jaiei 1-5 bitarteko puntuazioa eman.

Gainontzeko erabiltzaileen profila ikusteko aukera izango du eta gainontzekoek bere zer informazio ikusiko duten erabakitzekoa ere: helbide elektronikoa, sartutako tokiak, ibilbide publikoak, ... erakutsi edo ez, aldagarria izango da.

Herri batean zer toki edo jai ikusi ez dakiten erabiltzaile erregistratuek plan bat

sortzeko eska diezaiokete sistemari. Honek, erabiltzailearen preferentziak kontuan hartuz berarentzat interesgarriak diren tokiak eta jaiak bueltatuko ditu.

Ibilbideak sortu, editatu eta ezabatu al izango ditu. Sortuta dituen ibilbideetan tokiak kateatuko dira sartzen doazen heinean eta ibilbide hauek partekatu al izango ditu gainontzeko erabiltzaile erregistratuekin bere profilean ikusgai jarriz. Ibilbide hauetatik tokiak gehitu eta sartu ditzakeen bakarra izango da sortu duen erabiltzailea. Beste erabiltzaileen ibilbideak ikusi al izango ditu publikoak direnean soilik.

### **5.1.3- Hirugarren rola: erabiltzaile administraria**

Hirugarren rol hau informazio bulegoko langilearen rola da. Gutxienez herri edo probintzia baten administraria izango da sisteman eta hauek, administrari bat baino gehiago izan dezakete. Bigarren rola zenbait funtzionalitate hedatzen zaizkio honi.

Administratzen duten gunean soilik (probintzia bat bada, bertako herri guztiak hartzen ditu), tokiak sartu al izango ditu berme batekin edota sartuta dauden tokiei bermea eman, beharrezkoa den kasuetan informazioa editatuz. Bermea daukaten tokietan agertzen den informazioa beraz, turismo bulegoko langileren batek ziurtatuta dago. Gainera, administratzen duten herrietan jaiak sortzeko aukera izango du. Jai hauek puntualak edo urtero errepikatzen direnak izango dira.

Erabiltzaileek idatzitako mezuren bat beren administrazio gunekoa bada, irakurtzeko aukera izango du eta beren helbide elektronikotik erantzun ostean 'erantzunda' bezala markatu. Horrela gainontzeko administrariei ez zaie mezu bera agertuko eta behin bakarrik erantzungo zaio erabiltzaileari.

Azkenik, beren administrazio zonan salatu diren iritziak gestionatu beharko ditu, irakurri eta egokiak edo ez egokiak diren esanez. Egokiak diren kasuetan, iritziak dauden bezala geratuko dira eta bestela ezabatzeko aukera izango du.

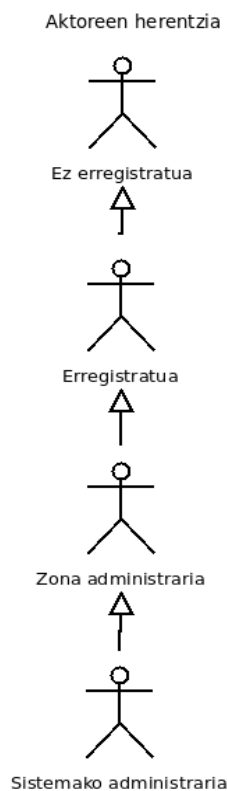
### **5.1.4- Laugarren rola: web administraria**

Azken rol hau da sistemako funtzionalitate guztiak erabili ditzaken bakarra. Sistemaren

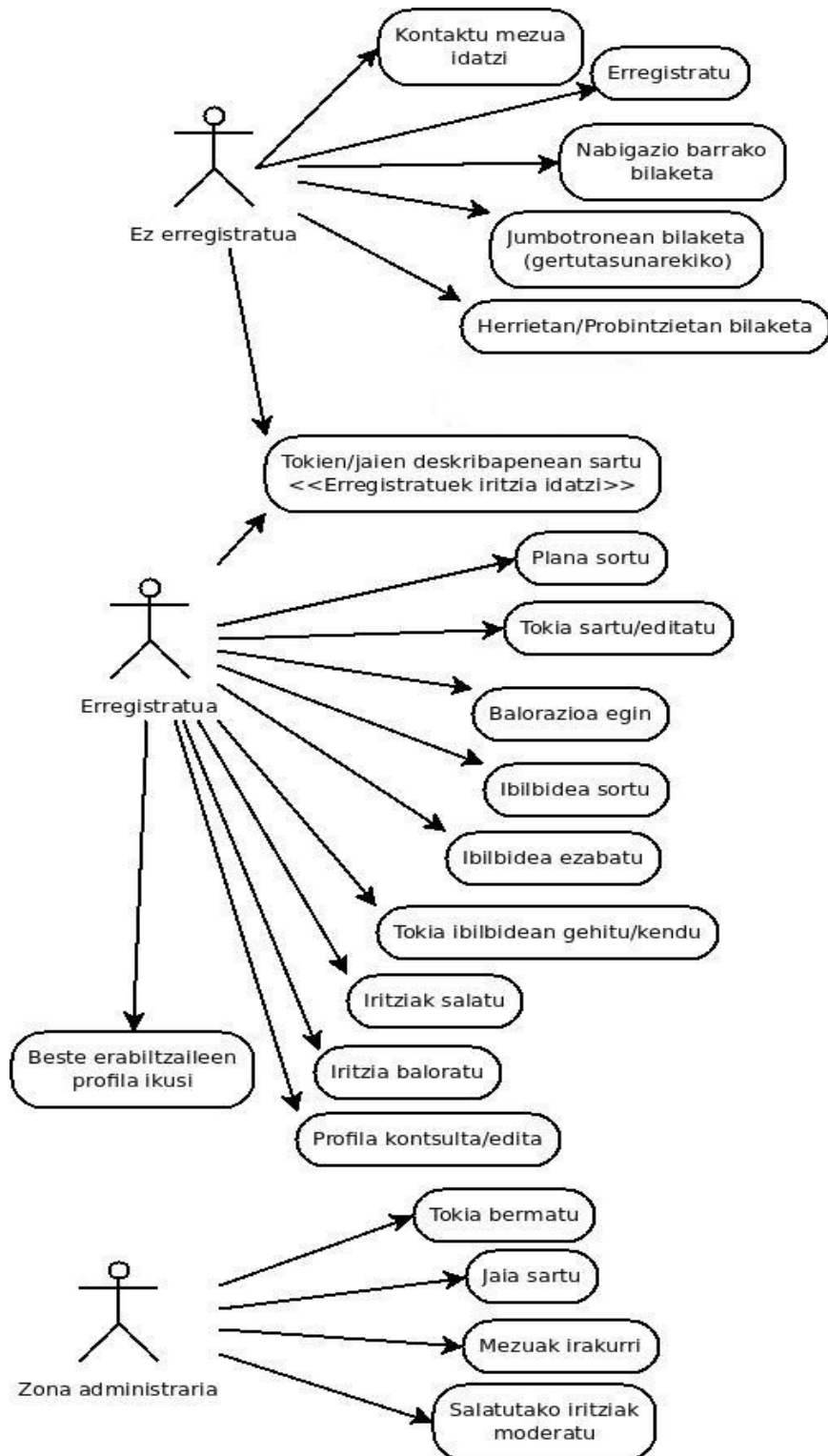


kontrol guztia izango du, probintziak, herriak, zerbitzuak administratzeko aukera izango du administrazio paneletik. Sisteman dagoen informazio guztia ikusi eta editatzeko aukera izango du. Administrazio panelera sartzeko aukera izango duen rol bakarra da hau.

Rolen herentzia hierarkia islatzen duen irudia (5.1 irudia). Oinarrizko erabiltzailea ez erregistratua da, erregistratzen den momentuan erabili ditzakeen funtzionalitateak hedatu egiten dira. Are gehiago guneren bateko administraria bada. Sistema administrariak aurrekoek egin dezaketena eta administrazio panelean sartzeko aukera daukate.



**5.1 Irudia** - Rolen herentzia-hierarkia

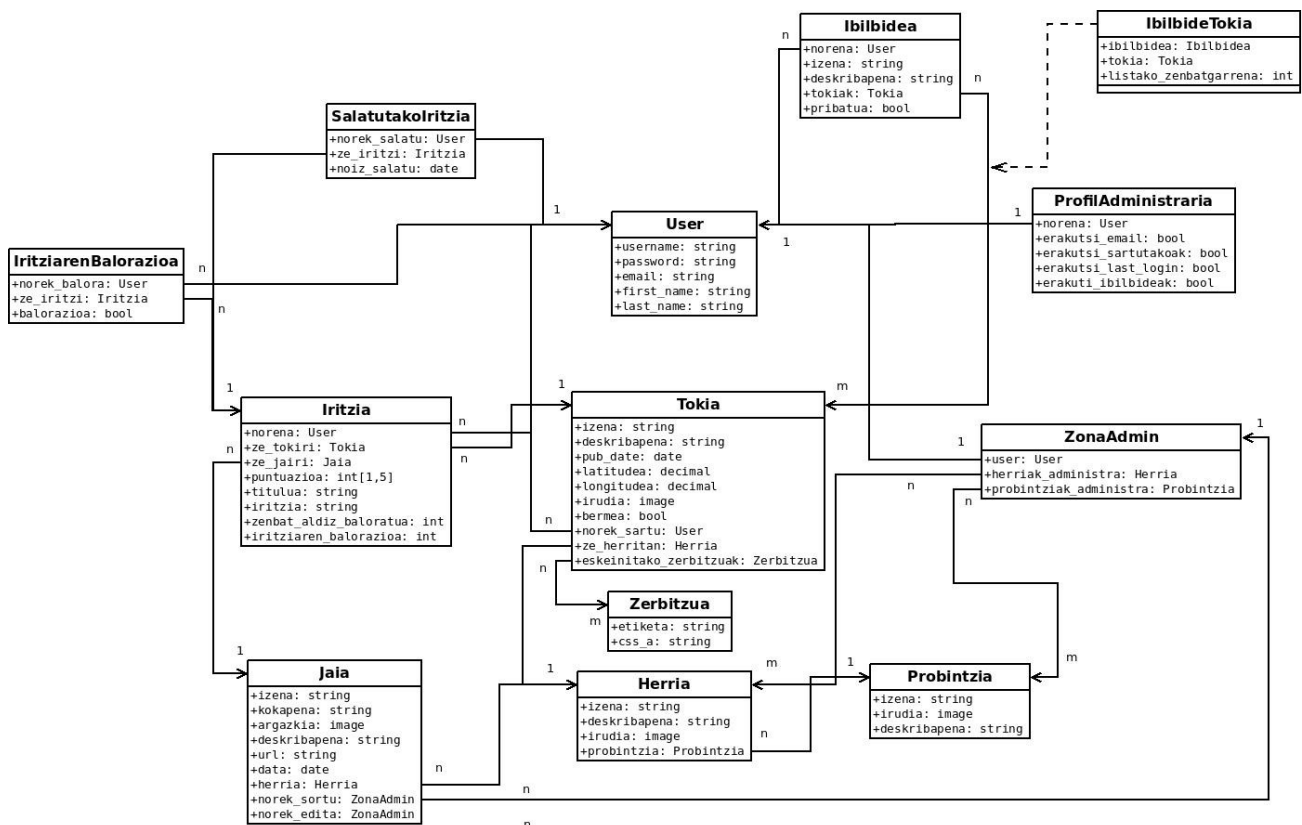


**5.2 Irudia** - Erabiltzaile ez erregistratua ,erregistratuaren eta administrarien erabilpena kasuak

Sistemako administrariaren erabilpen kasuak ez dira grafikoki adieraziko, aplikazioko erabiltzaile ia guztien rolak aurreko hirurak direlako. Sistemako administrariak soilik dauka aukera administrazio panelean sartzeko, eta hemendik, aplikazioan agertzen diren toki, jai, zerbitzu, iritzi, erabiltzaile, herri, probintzia, ... guztien kontrola dauka. Erabiltzaile mota hau izango da probintziak, herriak eta zerbitzuak sartzeko arduraduna.

## 5.2- Klase diagrama

Gure sistemaren betekizunak betetzeko ondorengo klase diagramak islatzen duen datu basea eraiki da. Kontuan izan Django sortzen diren datu baseko ereduak objektu gisa erabili daitezkeela guztiek hartzen dutelako beren barnean 'objects' objektua eta honek dituen funtzio guztiak al dira erabili beraz (get, filter...).



5.2 Irudia - Datu baseko ereduak eta beren loturak

Djangoren estentsio baten bidez klase eta lotura hauek automatikoki sortzeko aukera

dago (ikus E1 eranskina), baina sortzen duen diagrama zabalegia da eta Djangok bere barnealdean erabiltzen dituen ereduak agertzen dira, horregatik 5.2 irudian sortu behar izan diren ereduak adierazten dira.

5.2 irudian datu basean iraunkorrak izango diren eta beren artean lotuta duten ereduak soilik jarri dira baina, eredu laguntzaile pare bat ere badaude: gomendioa eredu eta kontaktua eredu. Gomendioa eredu 6.2.2 atalean ikusiko diren gomendioak pantailaratzen laguntzeko balio izango duen eredu da eta kontaktuarena berriz, egondako kontaktu guztiak biltzen dituena.

Eredu bakoitza laburrean esplikatuz:

- *User* klasea da erabiltzailea islatzen duena eta honek, *ProfilAdministraria* izeneko klase bat edukiko du, azken hau erabiltzailearen profilean erakutsiko dena finkatzeko da. Bi klase hauek bereizita daude, *User* klasea *Djangok* eginda ekartzen baitu eta hobe da <1,1> bezalako lotura egitea ekartzen duena aldatzea baino.
- *ZonaAdmin* klasea berriz, erabiltzaile kopuru murriztu batek besterik ez du izango. Hauek izango dira turismo bulegoko langileak. Herri batzuk edo probintzi batzuetako administrariak izango dira.
- *Tokia* klasea izango da berriz sistemaren muina. Toki bat *User* batek sortu beharko du eta lotura bat egin hainbat zerbitzurekin eta herri batekin.
- *Herria* klaseak berriz, *Probintzia* bat izan beharko du intuitiboa den bezala, eta bere barnean hainbat toki eta jai izango ditu.
- *Jai* klaseak berriz, *ZonaAdmin* batek sortuko du eta horregatik egon behar du bati lotua. Jai hauek herri batean egongo dira eta baliteke beste zona administrari batzuek etorkizunean editatzea.
- *Iritzia* klasea bai *Tokia* klaserako eta baita *Jai* klaserako erabiliko da, negozio logikan kontrolatuz. *User* batek hainbat iritzi idatz ditzake toki edo jai batean (negozio logikan iritzi hauek toki bakoitzerako hilean behin izatera murrizten dira).

- *Iritziaren Balorazioa*, *User* batek egiten du *Iritzi* batean egokia den edo ez adierazteko. *Iritzi* bat baloratzen den bakoitzean sortzen da.
- *Salatutako Iritzia* erabiltzaile batek *iritzi* bat salatzean sortzen da. Gero zona administrariak kontrola hartu dezan.
- *Ibilbidea* klasea erabiltzaileek sortzen dituzten ibilbideak gordetzeko dago. Hainbat toki sartu daitezke ibilbidean eta erabiltzaile batek hainbat ibilbide izan ditzake. Nahiz eta ibilbide bat erabiltzaile bakarrarena izan, publikoa den kasuan gainontzekoek ere ikusi al izango dute.
- *Ibilbide Tokia* ereduak *Ibilbidea*-ren laguntzailea da. *Ibilbide* batek *ManyToMany* motako lotura  $(n,m)$  dauka *Tokia* ereduarekin eta toki hauek ibilbidean sartzen joatean ordena ez da errespetatzen. Horregatik sortu da eredu laguntzaile hau, ibilbidean sartzen joan diren tokien ordena gordetzeko.

## 5.3- Sekuentzi diagramak

---

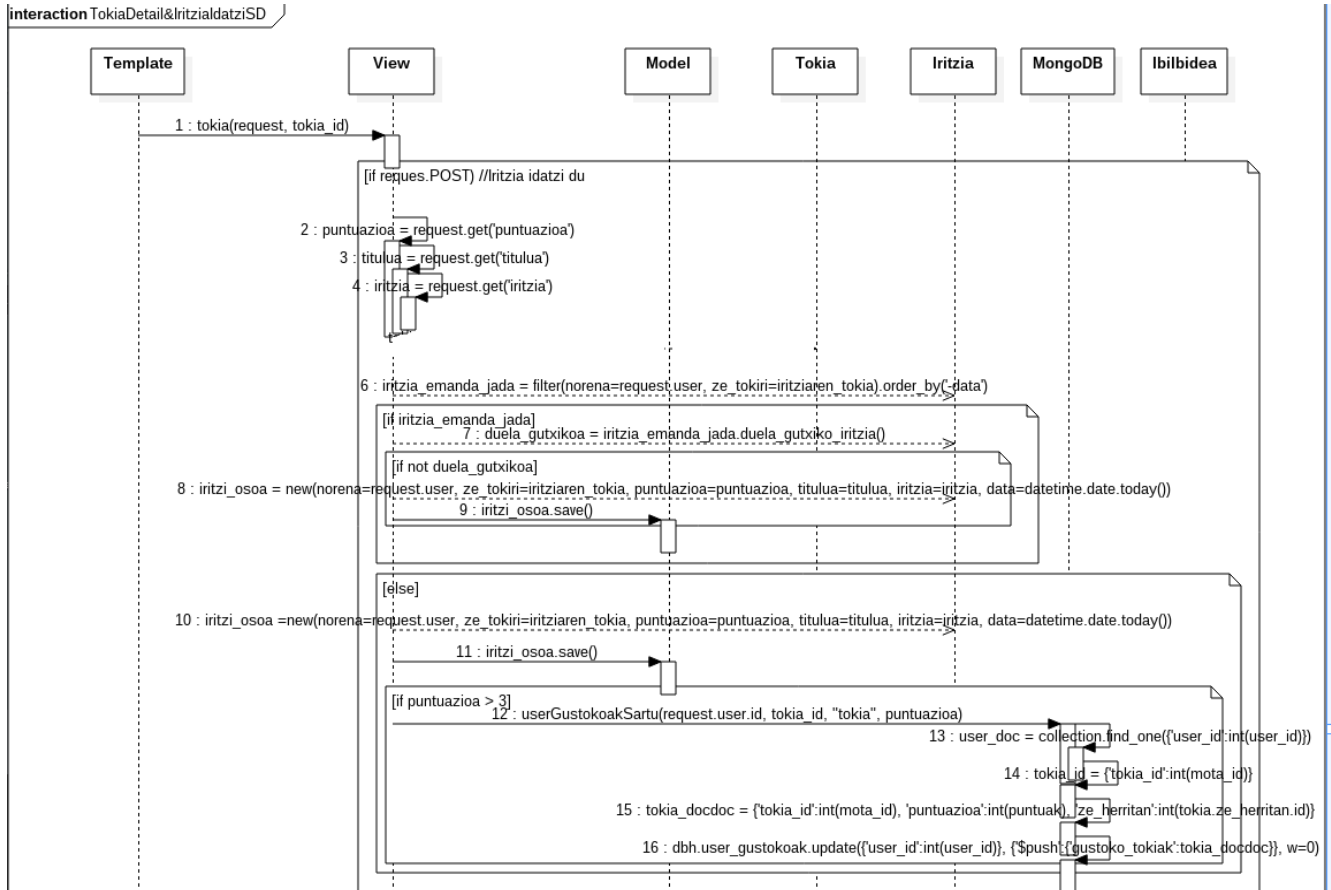
Sistemaren erabilpen kasuak hobeto ulertzeko, sekuentzi diagrama ikusiko ditugu eta haietako garrantzitsuenak azalduko dira atal honetan. Kontuan izan behar da, Django-ko '*objects*' atributua gehitzen diela eredu guztiei eta honek hainbat funtzio ematen dizkiela (erabilienak *get()*, *filter()* eta *all()*). Sekuentzi diagrametan '*objects*' ez da azalduko ulergarritasun hobea lortzeko, baina egon badago. Horrela, *tokia.get()* agertzen den tokian, *tokia.objects.get()* egiten da errealitatean.

Kasu batzuetan agertzen diren deiak ezin dira burutuak izan, adibidez tokia ikusi sekuentzi diagraman ezin dira ibilbideak jaso erabiltzailea login eginda ez badago. Hau ez da sekuentzi diagrametan adierazten, baina python kodean *try/catch* bidez egiteko aukera dagoenez, login eginda ez badago *catch* exekutatu litzateke eta hemen '*continue*' bat egon daiteke exekuzioa errore gabe exekutatzen jarraitzeko.

### 5.3.1- Tokia ikusi / iritzia idatzi sekuentzi diagrama

Sekuentzi diagrama hau bi iruditan banatuko da. Toki baten deskribapenean sartzean

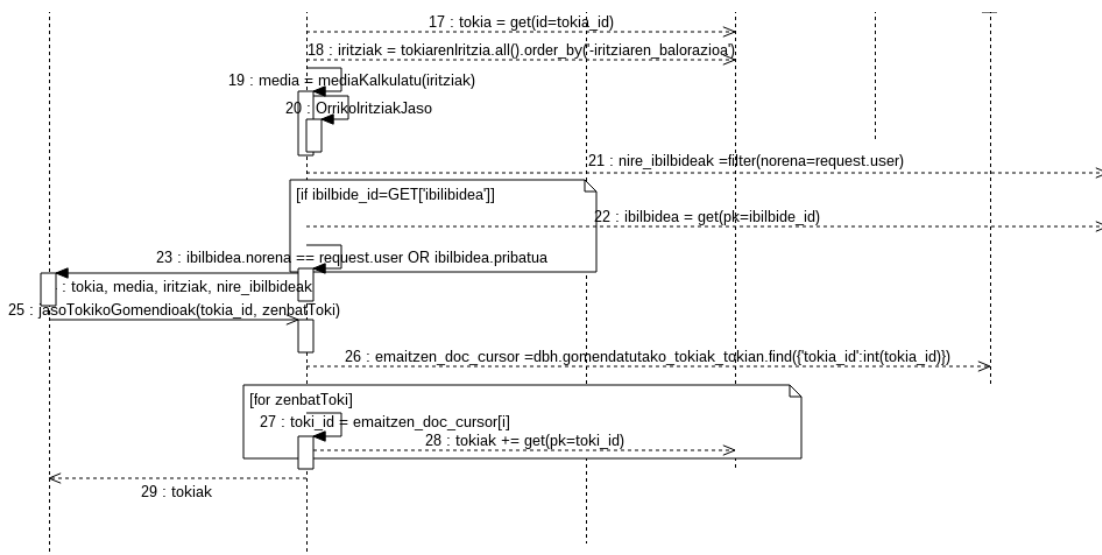
eta iritzia idaztean sekuentzi diagrama bera exekutatzen da. Eskaerarekin batera tokiaren identifikadorea bidali behar dio txantiloiak negozio logikari (1 pausua sekuentzi diagraman).



5.3 Irudia - Tokia detail / iritzia idatzi SD 1

Negozio logikan eskaeran ea POST bidez ezer bidali den ikusten da eta horrela bada, erabiltzaileak iritzi bat idatzi duela adierazten du eta honen datuak eskuratzen dira (2,3 eta 4). Toki horretan erabiltzaileak aurretik iritzirik idatzi badu ikusten da eta horrela bada, ea aspaldiko iritzia den (aspaldiko iritzia izango da baldin duela 30 egun baino aurrerago idatzia bada). Duela askoko iritzia bada, iritzi berri bat sortuko da eta gorde egingo da.

Aurretik iritzirik idatzita eduki ez badu, iritzia sortu eta gordetzeaz gain (10 eta 11), emandako puntuazioa 3koa baino handiagoa izan bada MongoDB-ko erabiltzailearen dokumentuan dagoen 'gustoko\_tokiak' listan gordeko dira tokiaren beharrezko datuak.



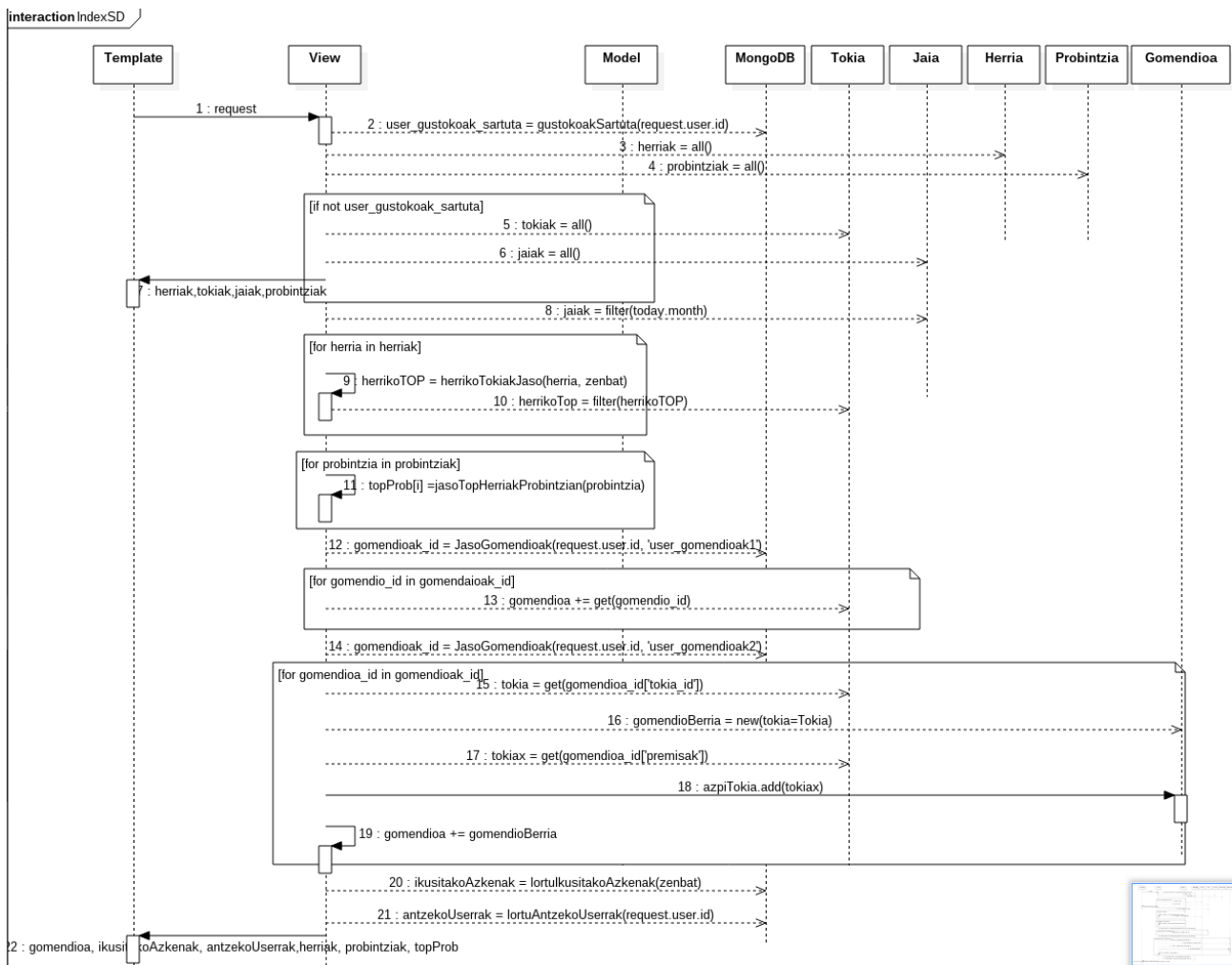
**5.4 Irudia** - Tokia detail / iritzia idatzi SD 2

Iritziaren atala atzeaz utziz, tokia eta honen iritziak eskuratzen dira. Iritziekin tokiak daukan puntuazioa lortzen da eskatutako orrialdeko iritziak hartuz. Tokia ibilbideetan sartu al izateko, erabiltzailearen tokiak eskuratzen dira eta gero, kontsultatu nahi den tokia ea ibilbideren baten parte den ikusten da GET metodoaren bidez. Horrela bada, ibilbidea jasotzen da eta norberarena bada edo publikoa bada datuak bueltatzen dira txantiloia geruzara.

Txantiloia kargatzen denean, asinkronoki tokian gomendatu beharreko tokiak eskatzen zaizkio zerbitzariari (25). Honek, MongoDBn kontsultatzen ditu uneko tokian zeintzuk toki gomendatu behar diren (hauen identifikatzaileak) eta ezarpen fitxategian adierazita dagoen 'zenbatToki' aldagaia bezain beste toki gomendatzen dira (zenbaki bat da, 3 normalean).

### 5.3.2- Hasiera orriko kontsultaren sekuentzi diagrama

Hasiera orrian hainbat informazio erakusten da, batez ere erabiltzaile erregistratuentzat. Nahiz eta erabilpen kasu espezifiko bat ez izan interesgarria da orrialdea kontsultatzen denean zer gertatzen den ikustea. Erregistratuek informazio ezberdina ikus dezakete beren ezaugarrien arabera.



### 5.5 Irudia - Hasiera orriaren kontsulta

Hasiera orriaren eskaera egiten denean, sistemak erregistratua den kasuan ea gustokoak sartuta dituen ikusten du (login egiten den lehen aldiari datu hauek eskatzen dira). Ez badauka gustokorik sartuta, datu baseko toki, probintzi, herri eta jaiak bueltatuko zaizkio baten batzuk baloratu ditzan (7).

Bestela, hileko jaiak lortzen dira, gero herri bakoitzeko tokirik hoberenak edo baloratuenak. Probintzietako herririk arrakastatsuenak ere lortzen dira, bai erregistratu eta baita ez erregistratuentzat ere.

Erregistratuen kasuan, gomendio mota ezberdinak lortzen saiatzen da sistema. Lehendabizi, erabiltzaileari berari pertsonalizatutako gomendioak (ikus 5.6 irudia) gustoko dituen tokietatik abiatuta aurrerago ikusiko dugun bezala datu meatzaritzako scriptetan. Gero beste gomendio batzuk jasotzen saiatzen da, kasu honetan gustoko duen toki batzuetatik induzitutako gomendioak ('x,y eta z tokiak gustoko baditu,



orduan w tokia ere gustuko izango du' erakoak. Ikusi 5.7 irudia). Azken hauek egiteko erabiltzen da 5.2 atalean adierazitako 'Gomendioa' eredua. Honetan, gomendioa betetzen duten premisak eta ondorioa gordetzen dira eta horrela txantiloiak era errazean pantailaratu ditzake datu guzti hauek.

Erabiltzaile guztientzako ere, ikusitako azken tokiak jasotzen dira eta azkenik erregistratuentzat soilik, antzeko erabiltzaileak eskuratzen dira beraiek sartutako toki edota ibilbideak kontsultatu ditzaten.

Hasiera orriaren bi adibide ikusiko ditugu erabiltzaile erregistratu ezberdinekin:

Herrietako TOP-ak

Jendeak ikusitako azken tokiak:

Laster datozen jaiak

Bizkaia **Gipuzkoa** Araba

1-Donostia-SS  
-Frac donostia  
-Udaletxea  
-Zuriola hondartza  
2-Zumaia  
-San telmo ermita  
-Talasoterapia Zelai  
-Luisen etxea  
3-Tolosa  
-Tolosako frontoia  
-Oria Ibaia  
-Eroski Tolosa

**Kontxa hondartza**  
Donostia erdiko hondartza...

**Ondarreta hondartza**  
Ondarreta hondartza donostiako antiguan dago kokatuta...

**Kursala**  
Tralari tralera kursala donostiakoa...

Bizkaia Gipuzkoa Araba

-Dantzari eguna (May 31, 2016) -  
Ispaster

Gomendatutako tokiak:

**Ispasterreko Frontoia**  
Ispasterren dagoen frontoi irekia...

**Autobus geltokia**  
Autobus Geltokia  
- Autobus geltokia  
- Buztintegiaren alorrean dago  
- Herriko Plazan dago  
- Buztintegiaren alorrean dago  
- Herriko Plazan dago  
- Buztintegiaren alorrean dago  
- Buztintegiaren alorrean dago  
- Buztintegiaren alorrean dago

**Fnac donostia**  
FNAC-ek Donostia aukeratu zuen bere Euskal Erkidegoko lehen denda irekitzeko 2005ean. San Martin Merkataritza Zentroan kokatuta, denda...

Intereseko erabiltzaileak

user1

### 5.6 Irudia - Hasiera orri 1, gomendatzeko toki eta erabiltzaileekin

Herrietako TOP-ak

Jendeak ikusitako azken tokiak:

Laster datozen jaiak

Bizkaia Gipuzkoa Araba

1-Ispaster  
-Autobus geltokia  
-Ispasterreko Frontoia  
-Ispasterreko plaza

**Kontxa hondartza**  
Donostia erdiko hondartza...

**Ondarreta hondartza**  
Ondarreta hondartza donostiako antiguan dago kokatuta...

**Kursala**  
Tralari tralera kursala donostiakoa...

Bizkaia Gipuzkoa Araba

-Dantzari eguna (May 31, 2016) -  
Ispaster

Gustoko dituzulako...

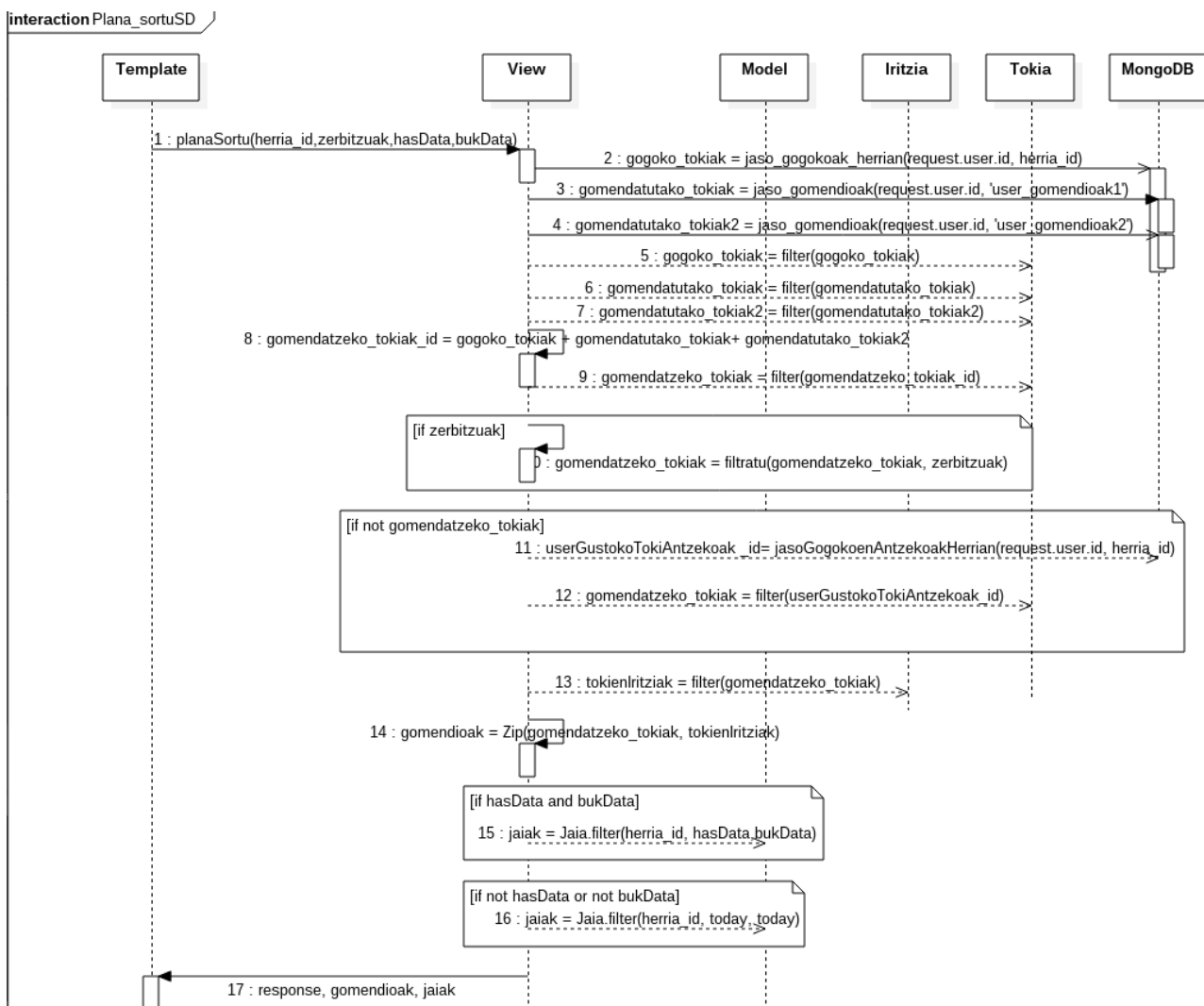
Ispasterreko Eliza, Autobus geltokia, gustuko dituzulako... seguru gustuko duzula:  
**Ogella hondartza**

Ispasterreko Eliza, Autobus geltokia, gustuko dituzulako... seguru gustuko duzula:  
**Ispasterreko Frontoia**

### 5.7 Irudia - Hasiera orria 2, 'x,y,z gustuko dituzulako...' gomendioekin

### 5.3.3- Plana sortu sekuentzi diagrama

Erabilpen kasu honek emaitza onak eman ditzan sistemak nahikoa informazio eduki behar du erabiltzailearengandik, hauetan oinarrituko baita aukeratutako herriko tokiak bueltatzean. Hau gertatzen da (ikusi 5.8 irudia) sistemak plana sortzeko eskatzean herri batean:



5.8 Irudia - Plana sortu SD

Erabiltzaileak herria aukeratu behar du eta nahi izanez gero zerbitzu batzuk eta data batzuk (jaiak bilatzeko data horien artean).





Negozio logikako mailan, aukeratutako herriko gogoko tokiak eta gomendatzeko tokiak jasotzen ditu (baliteke hauek hutsik egotea). Baldin erabiltzaileak zerbitzuak

aukeratu baditu, gomendatzeko toki horiek filtratu egiten dira gutxienez aukeratutako zerbitzu bat izan dezaten (gomendioak dituen emaitza ikusteko, ikusi 5.9 irudia).

Aurrekoa egin ostean gomendatzeko tokirik ez badauka, orduan herriko tokirik baloratuenak eta erabiltzaileak gustuko dituen antzeko tokiak lortzen saiatzen da herrian (ikusi 5.10 irudia). Gero gomendatzeko toki horien iritziak jaso eta gomendioekin objektu batean batzen ditu (tokia eta iritziak lotuta geratzeko).

Azkenik, datak aukeratu baditu, data horien arteko herriko jaiak lortzen saiatzen da eta bestela eskaera egin den eguneko jaiak.



#### Gomendatutako tokiak

 <p>Puntuazioa: Puntuaziorik gabe (0)</p>	<h3>Ogella hondartza</h3> <p>-Ispaster</p> <p>Ispasterren dagoen hondartza dotorea</p> <p>Etiketak</p> <p><a href="#">Hondartza</a></p>
 <p>Puntuazioa: Puntuaziorik gabe (0)</p>	<h3>Ispasterreko Frontoia</h3> <p>-Ispaster</p> <p>Ispasterren dagoen frontoi irekia</p> <p>Etiketak</p> <p><a href="#">Taberna</a></p>
 <p>Puntuazioa: Puntuaziorik gabe (0)</p>	<h3>Ispasterreko plaza</h3> <p>-Ispaster</p> <p>Ispaster erdian dagoen plaza</p> <p>Etiketak</p> <p><a href="#">Jatetxea</a> , <a href="#">Autobusa</a></p>
 <p>Puntuazioa: Puntuaziorik gabe (0)</p>	<h3>Ispasterreko eliza berria</h3> <p>-Ispaster</p> <p>Eliza berritua</p> <p>Etiketak</p> <p><a href="#">Gaua pasatu</a></p>

#### Gustoko dituzun tokiak

 <p>Puntuazioa: ★★★★★ (5,0)</p>	<h3>Ispasterreko Eliza</h3> <p>-Ispaster</p> <p>Plazan dagoen eliza</p> <p>Etiketak</p> <p><a href="#">Gaua pasatu</a></p>
	<h3>Autobus geltokia</h3> <p>-Ispaster</p> <p><b>5.9 Irudia</b> - Gomendiodun plana</p>

Informazio gutxiegi daukagu zutaz. Herriko TOP tokiak bueltatu ditugu

 <p>Puntuazioa: ★★★★★☆ (4,0)</p>	<h3>Tolosako frontoia -Tolosa</h3> <p>Tolosa herrian dagoen frontoia Etiketak Taberna</p> <p>✓ Bermatutako tokia</p>
 <p>Puntuazioa: Puntuaziorik gabe (0)</p>	<h3>Eroski Tolosa -Tolosa</h3> <p>Tolosan dagoen eroski bat Etiketak Jatetxea , Denda</p> <p>✓ Bermatutako tokia</p>

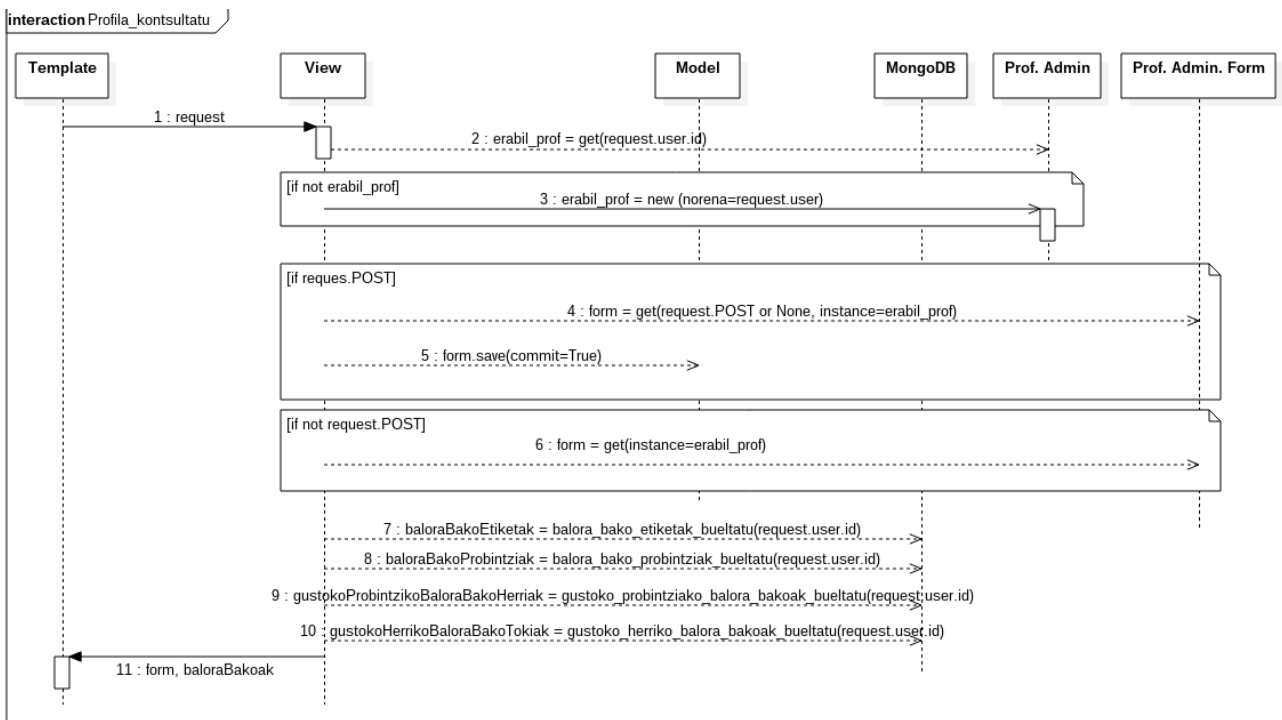
Ez dugu aurkitu bilatzen ibili zarenik, baina agian gustuko izango duzu...

 <p>Puntuazioa: Puntuaziorik gabe (0)</p>	<h3>Oria Ibaia -Tolosa</h3> <p>Tolosa erditik igarotzen den ibaia Etiketak Hondartza</p> <p>✓ Bermatutako tokia</p>
--	---

**5.10 Irudia** - Plana gomendio gabe. Herriko tokirik onenak eta antzeko gustukoa

### 5.3.4- Norbere profila kontsultatu sekuentzi diagrama

Erabiltzaileak bere profila kontsultatu dezake eta hemen, beste erabiltzaileek berengandik ikusiko duten informazioa aldatu dezake eta bere iritzia eman gustuko duen probintzietako tokietan eta herrietan.



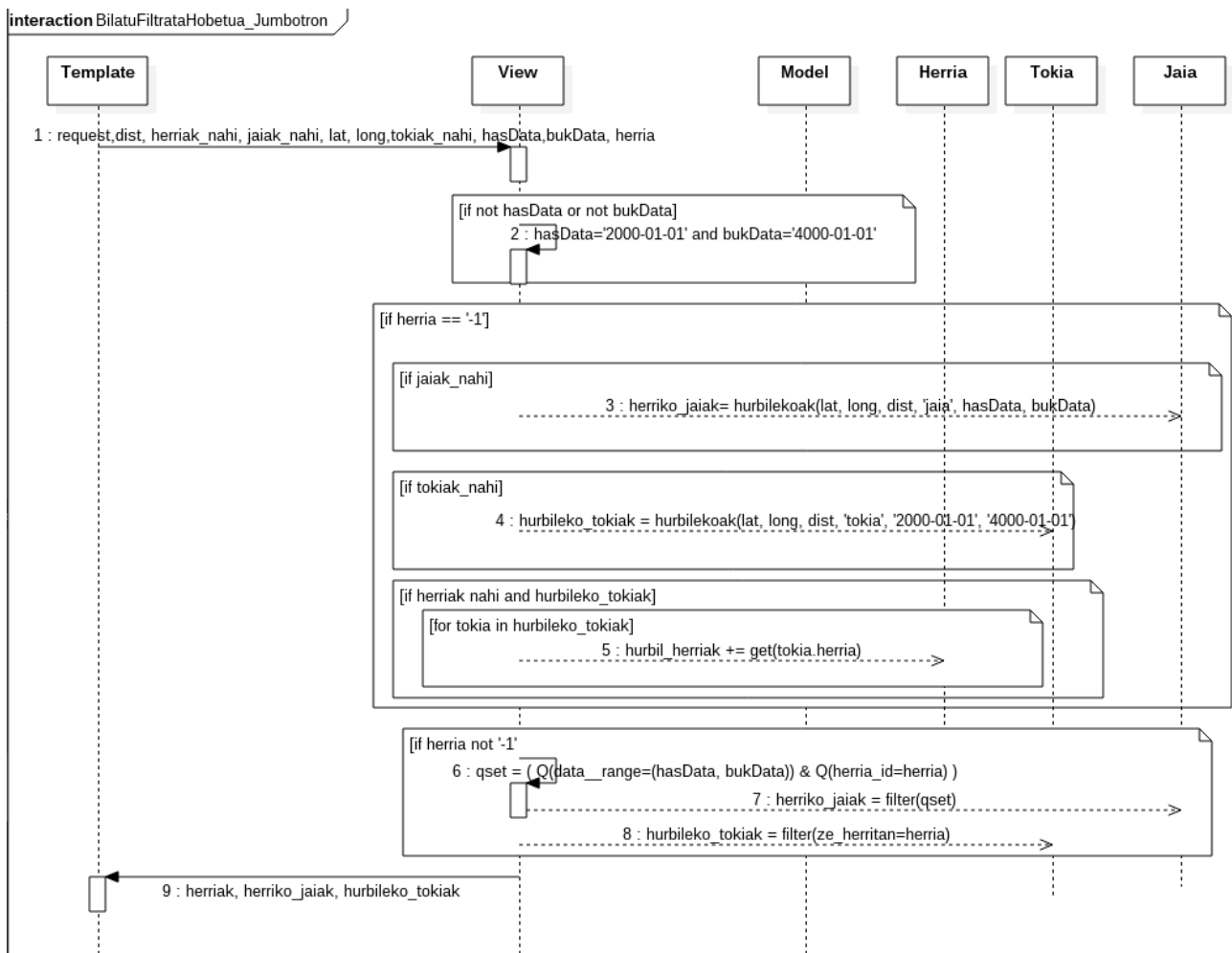
**5.11 Irudia** - Profila kontsulta SD

Lehendabizi ea erabiltzaileak profil administraririk duen kontsultatzen da. Lehen aldi ez du edukiko eta bat sortuko da, gainontzekoetan ordea existituko da. Jarraian, eskaeran ea POST metodoa dagoen ikusten da eta ala bada, esan nahi du erabiltzaileak bere profilean eguneraketaren bat egin duela eta eguneratu egiten da informazioa egokia bada eta bestela, bere profil administrariaren formularioa eskuratzen da.

Bukatzeko erabiltzaileak bere iritzia eman dezan, baloratzeke dituen etiketak eta probintziak jasotzen dira. Probintzi horietan baloratu gabeko herriak, eta gustoko herrietan baloratu gabeko tokiak. Horrela, bere gustuak gero eta zehatzagoak izatea lortzen da. Ez dauka sentsurik gustuko ez duen probintzia bateko herrien balorazioa eskatzeak.

### 5.3.5- Bilaketa data, herri edota gertutasunarekiko sekuentzi diagrama

Webguneko orri askotan agertzen den bilaketa kutxan herri eta hurbilpenarengandik bilaketa egiteko aukera dago, gainera data tarte bat ere jarri daiteke jaiak bilatzeko.



### 5.12 Irudia - Bilaketa data, toki edota hurbilpenarekiko

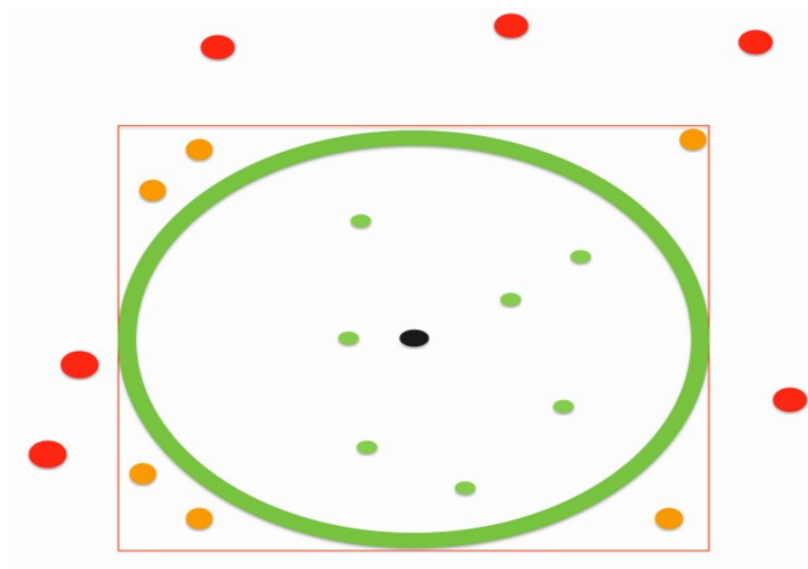
Erabiltzaileak aukera dauka herri batean bilaketa egiteko edota bere uneko kokapenarengandik hurbil dauden tokiak, herriak eta jaiak bilatzeko. Datak ere ezarri ditzake, beren tartean jaiak bilatzeko. Hurbilekoak bilatzeko nabigatzaileari kokapena partekatze baimena eman behar zaio (behin soilik).

Bilaketa egiteko eskatzean, datu guztiak GET bidez pasatzen zaizkio negozio logikari. Honek, hasiera data eta bukaera data dituen ikusten du eta ez baditu, balio tarte handia ematen dio (jai guztiak hartzeko haina).

Gertutasunagatik bilaketa egin nahi badu (herria == '-1' 5.12 irudian), eskatutakoak bilatzen saiatzen da (herriak, jaiak edota tokiak). Hurbilekoak funtzioak, ez ditu hurbilekoak zirkunferentzian hartzen baizik eta karratua. Hau da, puntu bat emanik

(latitude eta longitude bidez), latitudea  $\pm$  distantzia eta longitudea  $\pm$  distantzia karratuan daudenak bueltatzen ditu. Beraz, errore margin bat dauka funtzio horrek (ikusi 5.13 irudia), benetan puntu beltzetik (norberaren kokapena) distantzia batetara daudenak zirkunferentzia bat osatzen dute baina programak karratu bezala hartzen du eta marrazkian puntu laranja eta gorriak ere ontzat ematen ditu.

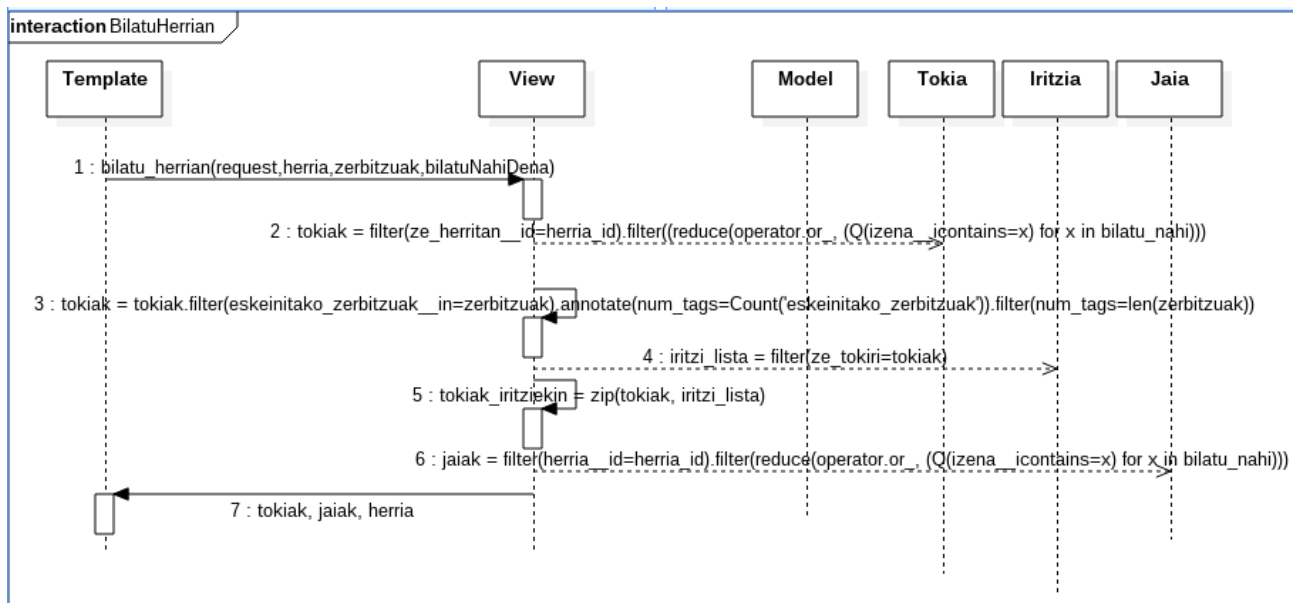
Bilaketa herri batean izan bada ordea, herri horretako jaiak eta tokiak eskuratzen ditu.



**5.13 Irudia** - Gertutasunarekiko bilaketa

### 5.3.6- Bilaketa herrian sekuentzi diagrama

Herri baten informazio orrian, herriko hainbat toki agertzen dira baina aukera dago bilaketa aurreratu bat egiteko ere (Ikusi 5.15 irudia).



**5.14 Irudia** – Bilatu herrian SD

Bilaketa egitean tokiak filtratzen ditu bilaketa kutxan idatzitako hitzak tokiaren izenean egotea eta tokia bera bilaketa egin den herrian egotea. Gero lortutako toki horietan gutxienez aukeratutako zerbitzu bat egoteko iragazten da berriro.



## Herria: Donostia-SS - Gipuzkoa



### General information:

#### Description:

Donostia Gipuzkoako kostaldean kokatuta dagoen hiriburua da.



### Bilaketa herrian: Donostia-SS

Hondartza,  Jatetxea,  Gaua pasatu,  Denda,  Unibertsitatea,  Taberna,  Taxiak,  Trena,  Autobusa,

Bilaketa herrian...

### Herriko tokirik baoratuenak:



Puntuazioa: ★★★★★ (5,0)

### Fnac donostia -Donostia-SS

FNAC-ek Donostia aukeratu zuen bere Euskal Erkidegoko lehen denda irekitzeko 2005ean. San Martin Merkataritza Zentroan kokatuta, denda gune handien eta aldi berean merkataritza tradizionalaren abantailak ditu. bisitariak produktu mota bakoitzerako eskaintza zabala izango du, merkatuko gutxieneko prezioan, eta langileen aholkularitza eta adetasuna. FNAC-ak argazkigintza, irudia, soinua, informatika, GPS, PDA, telefonia, musika, zinema eta liburu kontuetan azkenekoa du. Hiriko eragile kultural bat gehiago da, eta emanaldiak zuzenean, erakusketak, zinema proiektzioak.... antolatzen ditu. Telf: 902 100 632

Etiketak

Denda



Bermatutako tokia



### Udaletxea -Donostia-SS

**5.15 Irudia** – Donostia herriaren informazio orria. Herrian bilaketa egiteko kutzarekin

# 6

---

---

## Implementazioa

Atal honetan webgunea nola garatu den azalduko da bi ataletan bereizita, lehendabizi web aplikazio soilak (ikasketa gabe) nola dagoen egituratua eta bestetik datu meatzaritza egiteko gidoiak (scriptak) nola implementatu diren ikusiko da.

Produktua GIT karpeta batean implementatu da bertsio kontrola eramateko. Implementaziorako erabili den *PyCharm* testu editoreak gainera GIT-en erabilera asko errazten du, bertsio kontrol hau interfaze grafiko bidez kontrolatu daitekeelako.

### **6.1- Django aplikazioaren implementazioa**

---

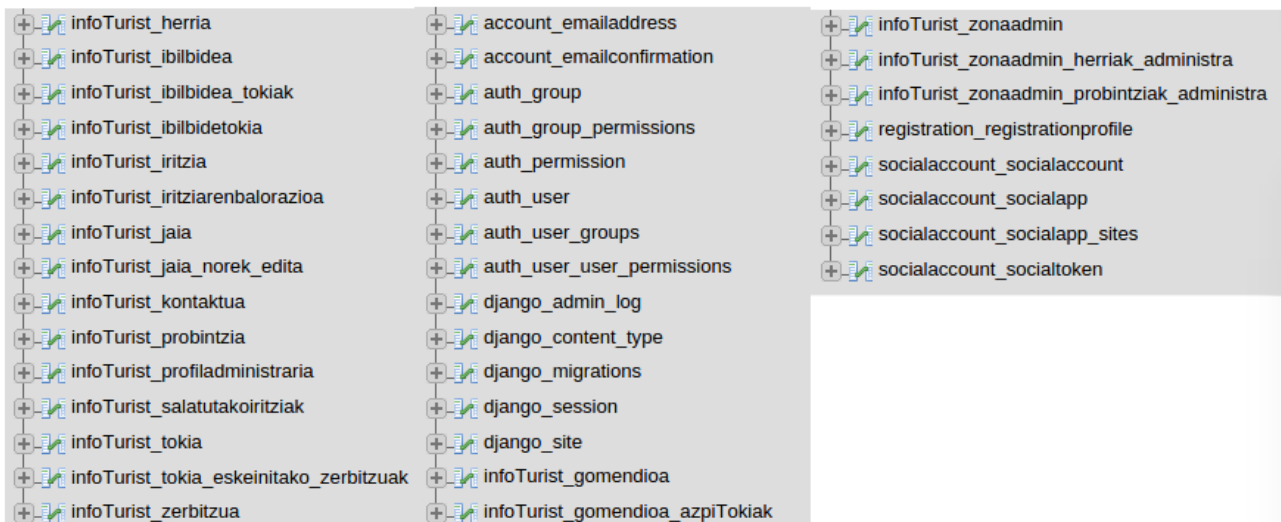
#### **6.1.1- Datu basea**

Sisteman datuen iraunkortasunaz arduratzeko MySQL datu basea erabiltzea erabaki da eta lan inguruneari (Djangori) eman zaio honen kontrola. Horretarako, teknologietako atalean aipatu bezala, proiektuko `settings.py` fitxategian adierazi behar izan da non dagoen datu basearen izena (aurretik sortu behar da), honen kokapena eta zein den bertako erabiltzaile eta pasahitzak (IP, portua, username eta password). MySQL datu basea *xampp* (apacheren distribuzio bat) bidez abiarazten da eta honekin konektatuko da proiektua.

Hasieran sortzen dituen taulak eta emaitzaren taulen arteko ezberdintasuna ikus daiteke 6.1 eta 6.2 irudietan.



### 6.1 Irudia - Django aplikazio hutseko taulak MySQL-en



### 6.2 Irudia - Aplikazioaren datu baseko taulak MySQL-n

Baina taula guzti hauek ez ditu programatzaileak sortzen, baizik eta Django lan inguruneak berak, eta datu baseko ereduak eta beren erlazioak adierazteko, teknologiak atalean aipatutako *models.py* fitxategian egin behar da. 'User' ereduak djangok berak eginda daukana erabili da eta horretarako “*from django.contrib.auth.models import User*” jarri behar da fitxategi hasieran. Honek Djangok emandako autentifikazio modeloetako User ereduak erabiltzea ahalbidertzen du. Teknologiak atalean ikusi da ereduak idazteko adibide bat, dokumentuko 3. atalean.

Behin eredu guztiak idatzita daudela lan inguruneari esan behar zaio taulak

sortzeko (models.py fitxategian eguneraketaren bat egiten den bakoitzean egin behar da hau) bi agindu idatziz terminalean proiektuan kokatuta gaudela:

```
$ python manage.py makemigrations
```

```
$ python manage.py migrate
```

```
mikel@mikel-pc:~/MEGA/Ingenieritza_informatikoa/GAP/produktua$ python manage.py migrate
Operations to perform:
  Synchronize unmigrated apps: staticfiles, messages
  Apply all migrations: admin, contenttypes, auth, sessions
Synchronizing apps without migrations:
  Creating tables...
    Running deferred SQL...
  Installing custom SQL...
Running migrations:
  Rendering model states... DONE
  Applying contenttypes.0001_initial... OK
  Applying auth.0001_initial... OK
  Applying admin.0001_initial... OK
  Applying contenttypes.0002_remove_content_type_name... OK
  Applying auth.0002_alter_permission_name_max_length... OK
  Applying auth.0003_alter_user_email_max_length... OK
  Applying auth.0004_alter_user_username_opts... OK
  Applying auth.0005_alter_user_last_login_null... OK
  Applying auth.0006_require_contenttypes_0002... OK
  Applying sessions.0001_initial... OK
mikel@mikel-pc:~/MEGA/Ingenieritza_informatikoa/GAP/produktua$
```

### Irudia 6.3 - Django datu baseko taulak sortzeko prozesua

Django aplikazioa konektatzen da MySQL-rekin eta beharrezkoak dituen taulak sortu edo aldatzen ditu.

## 6.1.2- Django administrazio panela

Sortu berri diren ereduaren administrazio errazerako, aplikazioaren administrazio gunean gehitu behar dira eta horretarako, aurretik aipatu bezala *admin.py* fitxategian erregistratu behar dira agertzea nahi ditugun ereduak, hauen atributuak eta azken hauek nola agertzea nahi diren. Horretarako agertzea nahi diren ereduak erregistratu behar dira modu errazean:

```
admin.site.register(Tokia)
```

Baina horrela, 'Tokia' ereduaren informazio guztia agertzen da Djangok defektuzko formatuan adieraziz: Testua bada input kutxa bat, "ManyToMany" lotura bada

'multiselect' kutxa bat... (ikusi 6.4 irudia).

### Change tokia

<b>Izena:</b>	<input type="text" value="Haizearen orrazia"/>
<b>Deskribapena:</b>	<input type="text" value="Eduardo Chillida eskultoreak (Donostia, 1924)"/>
<b>Descripcion:</b>	<input type="text" value="Realizado por el escultor Eduardo Chillida (D)"/>
<b>Description:</b>	<input type="text" value="Created by the sculptor Eduardo Chillida (Sar)"/>
<b>Date published:</b>	Date: <input type="text" value=""/> Today   Time: <input type="text" value="00:00:00"/> Now   <small>Note: You are 2 hours ahead of server time.</small>
<b>Latitudea:</b>	<input type="text" value="43.321493645820920"/>
<b>Longituda:</b>	<input type="text" value="-2.005104795098304"/>
<b>Irudia:</b>	Currently: <a href="#">tokienIrudiak/orrazia.jpg</a> Change: <input type="button" value="Choose File"/> No file chosen
<input checked="" type="checkbox"/> <b>Bermatua</b>	
<b>Norek sartu:</b>	<input type="text" value="mikel"/>
<b>Ze herritan:</b>	<input type="text" value="Donostia-SS"/>
<b>Eskeinitako zerbitzuak:</b>	<input type="text" value="Hondartza"/> <input type="text" value="Jatetxea"/> <input type="text" value="Gaua pasatu"/> <input type="text" value="Denda"/>

Hold down "Control", or "Command" on a Mac, to select multiple items.

#### 6.4 Irudia - Administrazio guneko Tokia ereduaren adibidea, defektuzko moduan

Hau aldatu daiteke eta erabiltzailearentzat (nahiz eta sistema administraria izan) errazagoa izateko ('*user experience*' hobea) admin.py fitxategian ereduko zein datu eta nola agertzea nahi den adieraziko da. Horretarako klase berri bat sortzen da 'admin.ModelAdmin'-en herentzia bezala eta klase honetan jarri behar dira ereduko zeintzuk atributu eta nola agertzea nahi den. Honela geratuko da Tokia ereduarentzat klase berri hau:

```
class TokiaAdmin(admin.ModelAdmin):  
    # Hau bekoa, Tokien lista ikusteko  
    list_display = ('izena', 'deskribapena', 'ze_herritan', 'pub_date',  
                  'norek_sartu', 'latitudea', 'longituda', 'bermatua',  
                  'was_published_recently') #Informazio hau ere erakusteko  
    list_filter = ['pub_date'] #Datekiko filtroa gehitzeko  
    search_fields = ['izena', 'deskribapena'] #Izenarekiko bilaketa  
        ahalbideratzeko.  
    form = TokiaForm #Toki bat sartzeko orduan dagoen formularioa
```

Erregistratzeko orduan bi klaseak lotuak joan behar dira:

```
admin.site.register(Tokia, TokiaAdmin)
```

Aurreko adibidean jarritako 'TokiaForm' hori, Djangoen beste ezaugarrietako bat da. Formularioak forms.py izeneko fitxategi batean idatz daitezke gero edozein txantilo edo negozio logikatik erabili al izateko. Formulario bakoitza "forms.ModelForm" klasearen herentzia izango da eta klase honetan jarriko dira zeintzuk atributu eskatuko diren eta ze erataraz erakutsiko diren (lan inguruneak eskaintzen dituen widget-ak erabiliz). Adibide bezala ikusteko, hau da tokien formularioa:

```
class TokiaForm(forms.ModelForm):
    class Meta:
        model = Tokia
        fields = ['ze_herritan', 'izena', 'deskribapena',
'descripcion','description', 'latitudea', 'longitudea', 'irudia',
'eskeinitako_zerbitzuak']
        widgets = {
            'deskribapena': Textarea(attrs={'cols': 80, 'rows': 15,
'placeholder': 'Euskaraz...'}),
            'descripcion': Textarea(attrs={'cols': 80, 'rows': 15,
'placeholder': 'En castellano...'}),
            'description': Textarea(attrs={'cols': 80, 'rows': 15,
'placeholder': 'In English...'}),
            'eskeinitako_zerbitzuak': forms.CheckboxSelectMultiple,
        }
    def clean_latitudea(self):
        latitudea = self.cleaned_data.get('latitudea')
        #email_base, provider = email.split("@") #Talalala, myweb.com
        if not -90 <= float(latitudea) <= 90: #Latitudea bere limiteetatik kanpo
sartu bada
            raise forms.ValidationError("Mesedez, sartu latitude egoki bat. [-
90, 90]")
        return latitudea
    def clean_longitudea(self):
        longitudea = self.cleaned_data.get('longitudea')
        if not -180 <= float(longitudea) <= 180: #Latitudea bere limiteetatik
kanpo sartu bada
            raise forms.ValidationError("Mesedez, sartu longitude egoki bat. [-
180, 180]")
        return longitudea
```

Formularioan bertan ere idatz daitezke metodo batzuk zeintzuk formularioko inputak balidatzen dituzten. Aurreko adibidean ikusten da nola latitudea eta longitudea balidatzen diren, eta zeintzuk *widget* erabiltzen diren erabiltzaileentzat erosoagoa izan

dadin datuak sartzea.

Description: Created by the sculptor Eduardo Chillida (San Sebastián, 1924-2002) and the architect Luis Peña GancheGUI (Oñati, 1926-San Sebastián 2009), the Peine del Viento (Wind Comb) is the perfect example of beauty in its purest state, a profound dialogue between art and the landscape. The fusion between the fierceness of the sea and the strength of the iron structures has elevated the natural character of this contemporary art work to a place of meeting and leisure for the locals and tourists.

Latitudea:

Longituedea:

**Irudia:** Currently: tokienIrudiaK/orrazia.jpg  
Change:  No file chosen

**Eskeinitako zerbitzuak:**

- Hondartza
- Jatetxea
- Gaua pasatu
- Denda
- Unibertsitatea
- Taberna
- Taxiak
- Trena
- Autobusa

### 6.5 Irudia - Tokia ereduaren formularioa filtroekin

Horrela nahi ditugun ereduaren kontrola izan dezakegu administrazio paneletik.

**Django administration**

**Site administration**

Accounts	
Email addresses	<a href="#">+ Add</a> <a href="#">Change</a>
Email confirmations	<a href="#">+ Add</a> <a href="#">Change</a>
Authentication and Authorization	
Groups	<a href="#">+ Add</a> <a href="#">Change</a>
Users	<a href="#">+ Add</a> <a href="#">Change</a>
Infoturist	
Celebrations	<a href="#">+ Add</a> <a href="#">Change</a>
Cities	<a href="#">+ Add</a> <a href="#">Change</a>
Iritziaren Balorazioak	<a href="#">+ Add</a> <a href="#">Change</a>
Kontaktuas	<a href="#">+ Add</a> <a href="#">Change</a>
Places	<a href="#">+ Add</a> <a href="#">Change</a>
Probintziak	<a href="#">+ Add</a> <a href="#">Change</a>
Profil administrarias	<a href="#">+ Add</a> <a href="#">Change</a>
Reviews	<a href="#">+ Add</a> <a href="#">Change</a>
Salatutako iritziak	<a href="#">+ Add</a> <a href="#">Change</a>
Services	<a href="#">+ Add</a> <a href="#">Change</a>
Tours	<a href="#">+ Add</a> <a href="#">Change</a>
ZonaAdminak	<a href="#">+ Add</a> <a href="#">Change</a>
Registration	
Registration profiles	<a href="#">+ Add</a> <a href="#">Change</a>
Sites	
Sites	<a href="#">+ Add</a> <a href="#">Change</a>
Social Accounts	
Social accounts	<a href="#">+ Add</a> <a href="#">Change</a>
Social application tokens	<a href="#">+ Add</a> <a href="#">Change</a>
Social applications	<a href="#">+ Add</a> <a href="#">Change</a>

**Recent Actions**

**My Actions**

- [Donostia Kursaal](#)  
Tokia
- [Dantzari eguna](#)  
Jaiak
- [Kurtso bukaera](#)  
Jaiak
- [Donostiatic ibilbidea](#)  
Ibilbidea
- [Ispastar](#)  
Ibilbidea
- [Ispasterretik ibilbidea](#)  
Ibilbidea
- [Ispasterretik ibilbidea](#)  
Ibilbidea
- [Ibilbidea object](#)  
Ibilbidea
- [Tolosa](#)  
City
- [user1](#)  
Zona admin

### 6.6 Irudia - Administrazio guneko ereduak

## 6.1.3- Txantiloiak sortu

Erabiltzaileak sistemaren txantiloiak soilik ikusiko ditu eta hauek itxurosoak izatea komeni da beraz. Egitura eta itxura erakargarria izateko *Bootstrap*<sup>9</sup>-en 3.3.6 bertsioa erabili da, eta honen laguntzarekin sortu da “responsive web” (pantaila tamainetara moldatzen dena), CSS estilo nagusia eta zenbait JavaScript kode.

Egitura nagusiaren estiloa erazagututa, are gehiago egin da eta formularioen estiloa aldatzea ere erabaki da eta horretarako, *Crispy-Forms*<sup>10</sup> erabili da (Django-ren estentsio bat). Honek, instalazio bat eskatzen du eta Ubuntu makinetan 'pip' erabilia erraz batean instalatu daiteke:

```
$ sudo pip install --upgrade django-crispy-forms
```

Behin Crispy Forms instalatuta, settings.py fitxategian installed\_apps-en gehitu behar da.

```
INSTALLED_APPS = (  
    ...  
    'crispy_forms',  
)
```

Azkenik, formularioak dotoreago ikusteko txantiloietan **{{form}}** jarri beharrean **{{form|crispy}}** jarriko da eta ezberdintasunak ikusteko ikusi 6.7 eta 6.8 irudiak.

### Erregistro orria:

Username\*  
  
This field is required.  
Required. 30 characters or fewer. Letters, digits and @/./+/-/\_ only.

First name

Last name

Email address  
  
Mesedez, sartu email bat

Password\*  
  
This field is required.

Password berriz\*  
  
This field is required.

### Erregistro orria:

This field is required.

Username:  Required. 30 characters or fewer. Letters, digits and @/./+/-/\_ only.

First name:

Last name:

Mesedez, sartu email bat

Email address:

This field is required.

Password:

This field is required.

Password berriz:

### 6.7 eta 6.8 Irudiak – Erregistro formularioa Crispy-rekin eta gabe

9 <http://getbootstrap.com/>

10 <http://django-crispy-forms.readthedocs.io/>



Estiloa hobetzeko erabili den beste tresna erabilgarri bat, *Awesome fonts*<sup>11</sup> izan da. Honek estilo orrien bidez irudiak erabiltzeko aukera ematen du eta irudi hauei dinamismoa emateko aukera ere badago: biratzen, handiago, txikiago, ...

Djangon txantiloietan negozio logikak pasatutako testuinguru aldagaiak jartzeko '{{' eta ''}}' ikurren artean egingo da aurreko adibidean ikusi den bezala. Esaterako, Tokia objektu bat badago {{ Tokia.izena }} idatz daiteke eta tokia objektuaren izena agertuko da pantailan. Era berean, txantiloietan baldintza eta iterazioak egin daitezke eta hau oso erabilgarria da toki multzo bat dagoenerako adibidez. Hauen formatua aldagaiekin konparatuz ezberdina da '{%' eta '%}' ikurren artean joan behar dira eta gainera, bukaera etiketak ere jarri behar dira. Hobeto ulertzeko adibide batzuk:

```
{% for tokia in tokiLista %}
    {% if tokia.bermatuta %}
        <p>{{ tokia.izena }} berrmatuta dago</p><br />
    {% else %}
        <p>{{ tokia.izena }} berrmatu gabe dago</p><br />
    {% endif %}
{% endfor %}
```

Aurreko adibidea posible izan dadin negozio logikak (dagokion view-ak) 'tokiLista' izeneko testuinguru aldagaia pasatu behar dio txantiloari.

## 6.1.4- Erregistratzea

Bi modu nagusi daude sisteman erregistratzeko, alde batetik sisteman bertan modu tradizionalan, eta bestetik Facebook kontua erabiliz.

- Erregistro tradizionala

Egin den erregistroa aipatzekoa da *Django-registration-redux*<sup>12</sup> aplikazioa erabiltzen delako. Aplikazio hau, besteak beste, erregistroaz arduratuko da eta konfigurazio gutxi batzuk eginda funtzionatuko du. Lehendabizi instalazioa behar du, Ubuntu sistemetan:

```
$sudo pip install django-registration-redux
```

11 <http://fontawesome.io/>

12 <https://django-registration-redux.readthedocs.io>

Gero, ezarpen fitxategian aplikazioen listan gehitu beharko da 'Crispy' formularioekin egiten den bezala, baina kasu honetan gauza gehiago zehaztu beharko dira. Hasteko, aplikazio honek balidazio korreo elektronikoko bat bidatzen du erregistratzailearen helbide elektronikora eta horretarako ezarpenen fitxategian zein helbide elektronikotatik bidaliko den jarri behar da. Gainera, aktibazio kode horrek iraungitze data edukiko du eta fitxategi berdinean ezarri beharko da zenbat egunetakoa den. Azkenik eta garrantzitsuenak, produktuko aplikazioan `urls.py` fitxategian ahalbideratuko da erregistroko path edo bidea:

```
...(r'^accounts/', include('registration.backends.default.urls')),...
```

Horrela URL-ak ondokoak izango dira,

-Erregistratzeko: <IP>:<portua>/accounts/register

-Login egiteko: <IP>:<portua>/accounts/login

-Pasahitza aldatzeko: <IP><portua>/password/change

-(...)

Era honetara erregistro eta loginean ez dela arazorik egongo ziurtatzen da.

Erregistratzerako orduan defektuz erabiltzaile izena begiratzen du Django ea erregistratuta dagoen jada, baina helbide elektronikorekin ez du hori egiten eta beraz, helbide elektronikorekin berdinarekin hainbat erabiltzaile sor daitezke sisteman hasiera batean. Hau konpontzeko bi modu ezberdin daude (egongo dira agian gehiago ere), alde batetik erregistratzeko klasearen hedapena egitea eta helbide elektronikoa balidatzea, edota bestetik Django-ren tripetan kuxkuxeatu eta "User" eredu aldatzea. Bigarren aukera erabaki da egitea, normalean aplikazio guztietan helbide elektronikoa unibokoa izatea nahi izango delako. Horretarako, Django-ren fitxategietara murgildu beharra dago makinan, Ubuntu makinetan `"/usr/local/lib/pythonX/dist-packages/django/"` direktorioan. Hemen barruan egonda, `"contrib/auth/"` direktorioko `"models.py"` fitxategia editatu behar da. Hemen daude erregistro eta login egiteko beharrezkoak diren eredu guztiak, haien artean "User". Fitxategian "AbstractUser" eredu editatu behar da eta helbide elektronikorekin atributuetan `"unique=True"` ezarri behar da, nahi izanez gero laguntza testuarekin erabiltzaileari pantailan agertzeko.

- Facebook bidezko logina

Erregistratu beharrik ez izateko, erabiltzaileak bere Facebook-eko kontua erabil dezake sisteman sartzeko, horretarako Facebook-i baimena eman behar dio bere informazioa sistemarekin partekatzeko.

Lehendabizi *django-allauth*<sup>13</sup> izeneko paketea instalatu da. Pakete honek hainbat sare sozialen bidezko logina ahalbideratzen du eta (aplikazioan Facebook bakarrik erabiliko da).

```
$ pip install django-allauth
```

Behin instalazioa burututa, *settings.py* fitxategian ezarpen batzuk gehitu behar dira: “*authentication backends*”-en gehitu, “*installed apps*”-en gehitu eta garrantzitsuen, “*Social account providers*” erazagutu. Azken honetan, erabili nahi den sare soziala erazagutu behar da eta bakoitzari eskatu nahi zaion informazioa jarri. Aplikazio honetan Facebook bakarrik erabiliko da eta honakoa da erazagupena:

```
SOCIALACCOUNT_PROVIDERS = \
{
    'facebook':
        {
            'METHOD': 'oauth2',
            'SCOPE': ['email', 'public_profile', 'user_friends', 'user_hometown'],
            'AUTH_PARAMS': {'auth_type': 'reauthenticate'},
            'FIELDS': [
                'id',
                'email',
                'name',
                'first_name',
                'last_name',
                'verified',
                'locale',
                'timezone',
                'link',
                'gender',
                'friends',
                'hometown',
                'updated_time'],
            'EXCHANGE_TOKEN': True,
            'LOCALE_FUNC': lambda request: 'en-US',
            'VERIFIED_EMAIL': False,
            'VERSION': 'v2.5'}
}
```

Horrela Facebook-etik hainbat informazio jasoko da, “*FIELDS*” atalean

---

13 <https://readthedocs.org/projects/django-allauth/>

adierazitakoak. Ezarpen hauek idatzi ostean datu baseko taulak eguneratu behar dira, eta administrazio panelean atal berri bat gehituko da (ikusi 6.9 irudia).

Social Accounts	
Social accounts	<a href="#">+ Add</a> <a href="#">✎ Change</a>
Social application tokens	<a href="#">+ Add</a> <a href="#">✎ Change</a>
Social applications	<a href="#">+ Add</a> <a href="#">✎ Change</a>

### 6.9 Irudia - Administrazio panelean sare sozialen atala

Gero Facebook developers-en kontu bat ireki behar da eta login egiteko erabiliko duen aplikazioa erazagutu. Django administrazio paneleko 'Social applications' eta Facebook developersen sortu berri den aplikazioa lotu egin behar dira, horretarako azken honek emandako pasahitzak sartuko dira lehenengoan.

Aipatutakoak egindakoan, login egin nahi den tokiak hiperesteka bat jartzea besterik ez da falta ondorengo helbidearekin:

```
<a href="{% provider_login_url 'facebook' method='js_sdk' %}">Facebook login</a>
```

### 6.1.5- Google maps

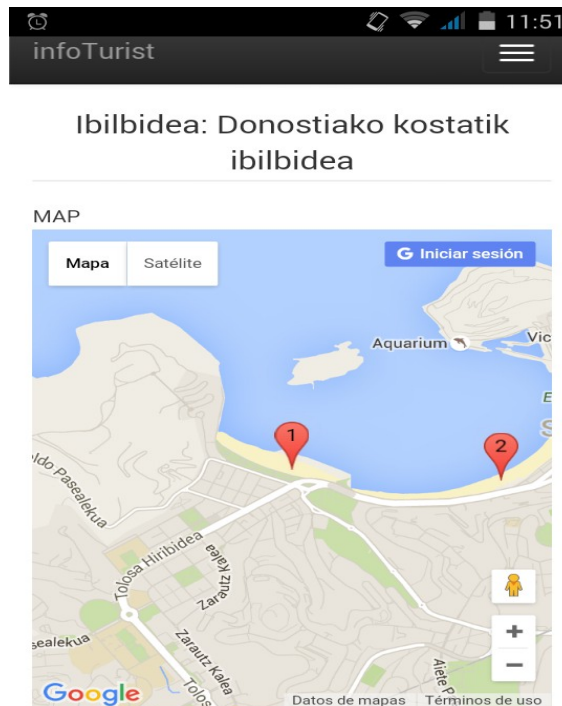
Aplikazioaren muina toki interesgarriak dira eta beraz, hauen kokapena agertzea ere ezinbestekoa da. Horretarako "Google"-k eskaintzen dituen mapen API-ak erabili dira. Hauek erabiltzeko lehendabizi beren garatzaileen webgunean izena eman behar da eta kode bat ematen dute aplikazioarekin erabil dadin. Garatzaileen orrialdean bertan jarri behar da kode horrek zein webgunetik egingo dituen eskariak, gainontzeko eskariak atzera botatzeko.

API-a erabiltzeko txantiloietan JavaScript kodea idatzi behar izan da, eta tokiaren beharrezko atributuak pasatu zaizkio (latitudea eta longitudea gutxienez). Tokiaren txantiloia kargatzen denean AJAX bidez (asinkronoki) Google-ri mezua bidaltzen zaio maparen informazioa bidal dezan, nabigatzailean ikus daiteke asinkronoki egindako deia (ikusi 6.10 irudia).

Name	Status	Type	Initiator	Size	Time	Timeline – Start Time	200.00 ms
QuotaService.RecordEvent?1shttp%...	200	script	js?key=AlzaSyCvmJTJ7pU8UUPI9s0...	112 B	108 ms		

### 6.10 Irudia - Tokiaren txantiloian Google-ri dei asinkronoa

Ibilbideen kasuan ere, hauek igarotzen dituzten tokiak era erosoagoan ikusteko Google-ren mapak erabili dira. Mapa hauetan ibilbideko puntu guztiak agertzen direlarik izen bezala ibilbideko posizioaren zenbakia jarrita. Adibide bezala ikusi mugikor (pantaila txiki) batean nola ikusiko litzatekeen ibilbidearen mapa 6.11 irudian.



Ibilbideko tokiak:

### 6.11 Irudia - Ibilbide baten mapa adibidea pantaila txikian

## 6.1.6- Ajax deiak

Aplikazioaren azkartasunerako txantiloia batzuetan lehendabizi derrigorrezko informazioa kargatzen da eta gero, erabiltzailea konturatu gabe, informazio hedagarria kargatzen da. Hau zerbitzarira dei asinkronoak eginda lortzen da, negozio logikak testuinguruan beharrezko aldagaiak pasatzen dizkio txantiloari eta azken hau behin kargatzen denean, zerbitzariari mezua bidaltzen dio informazio gehiago eskatzen (honetaz erabiltzailea ez da ohartzen, google maps-ekin gertatzen den bezala). Zerbitzariak informazio hedagarria bueltatzen dionean txantiloari honek pantailaratu

egiten du. Batzuetan teknika hau erabiltzen da erabiltzailearen erosotasunarengatik eta orrialdea behin eta berriz kargatzen ez egoteko. Erabilgarritasunean irabazten du aplikazioak.

Teknika hau aplikazioko hainbat tokitan erabiltzen da (10 tokitan zehatzak izanda): iritziaren balorazioak egitean, profilean etiketak baloratzean, lehen aldian login egitean eskatzen den balorazioetan, toki bateko gomendioak eskatzean, iritzia salatzean, iritzia ezabatzean, salaketa ezabatzean, toki bat ibilbidean gehitzean, toki bat ibilbidetik kentzean eta ibilbidea ezabatzean.

Adibide bezala toki bat kargatzean eskatzen diren gomendatutako tokiak nola kargatzen diren ikusiko da. Mapekin gertatzen den bezala, nabigatzaileko sareko mugimenduetan ikus daiteke deia (ikusi 6.12 irudia).

Name	Status	Type	Initiator	Size	Time	Timeline - Start Time	1.00 s	1.50 s
<input type="checkbox"/> tokikoGomendioakLortu	200	xhr	jquery-1.11.3.min.js:5	557 B	14 ms			

### 6.12 Irudia - Toki batean gomendioak eskatzen asinkronoki

Ikusten da deia egiten eta informazioa jasotzen 14ms iraun duela, gutxi da makina berean daudelako bai eskalea baita zerbitzaria ere (aurreko adibideko maparen iraupena: 108ms).

Demagun zerbitzaria lanean dagoela eta ezin diola eskariari erantzun denbora batez, ba denbora tarte horretan tokiaren deskribapena iritziekin agertuko litzateke eta gomendioen atalean 'pentsatzen' egongo balitz bezala agertuko litzateke (ikusi 6.13 irudia). Kargatzen dagoen irudi dinamikoa aurreko punturen batean (6.1.3) azaldutako "Awesome font" bidez egin da.

Interesekoa izan daiteke zuretzat...



### 6.13 Irudia - Gomendioak kargatzen, zerbitzariaren zain

Baina ajax dei hauek arazo bat daukate, zeren Djangok formulario eta informazio

trukean “*csrf*” izeneko aldagai bat bidali behar da eta dei asinkrono hauetan ezin da bidali. Formularioetan txantiloian bertan jarri behar da “{% csrf\_token %}” aldagaia. Hau segurtasun aldagai bat da (Cross Site Request Forgery protection) eta ajax deietan aldagai honi ez ikusi egiteko eskatu zaio negozio logikan (arazoa konpontzeko beste modu batzuk ere badaude) deia jasoko duen metodoaren aurrean “@*csrf\_exempt*” idatziz.

Adibide gisa:

```
@csrf_exempt
def tokikoGomendioakBuelatu(request):
    ...
```

### 6.1.7- Hizkuntza ezberdinak sartu

Aplikazioak informazio turistikoa eskaini behar badu, hizkuntza batean baino gehiagotan egotea da sentsuzkoena. Hau egiteko Django hainbat aukera ematen ditu eta haietako baten bat erabili da. Web aplikazioan hasiera batean euskaraz idatzi da eta hau izango da besterik ezeko hizkuntza. Hainbat tokitan dago testua euskaraz, URL-tik hasi eta tokien informaziora arte, etiketatik pasatuz. Badago URL-ak hizkuntzetara egokitzeko aukera, esaterako nabigazio barran “*.../tokia/12*” jarri beharrean “*.../place/12*” edo “*.../sitio/12*” agertzeko. Baina proiektu honetan webguneko informaziora bakarrik mugatu da itzulpen hau, zehatzago izateko, toki eta jaien deskribapenera eta interfazeko mezuetera (botoiak, testuak, ... txantiloietako testua asken finean). Itzulpen gabe geratu dira etiketen izenak, toki eta jaien izenak. Baina etiketek irudi intuitiboa daukate eta edonork ulertzekoak dira, eta tokien izenak bilaketak egiteko hobeto da hizkuntza bakarrean egotea.

Txantiloietako testuak hizkuntza ezberdinetan egon daitezkeela adierazteko “{% **trans .. %** %}” ikurrean joan behar dute. Adibide gisa:

```
<option value="-1">---{% trans 'Lehendabizi aukeratu probintzia bat' %}---</option>
```

Txantiloietan hizkuntza ahalbideratzeko hasieran {% load i18n %} etiketa jarri behar da. Behin txantilo guztiak eginda daudela, zein hizkuntza nahi diren adierazi behar da terminaletik. Aplikazioaren karpeta kokatuta ondorengo exekutatu:

```
$ django-admin makemessages -l en
```

Aplikazio karpeta 'lokale' izeneko direktorioa sortuko da eta hemen, hizkuntza bakoitzeko beste direktorio bat. Aurreko adibidean ingelesezko karpeta sortuko du dagokion fitxategiarekin eta beste hizkuntzaren bat nahi izanez gero adibideko 'en' aldatu behar da, erdararako adibidez 'es' jarrita.

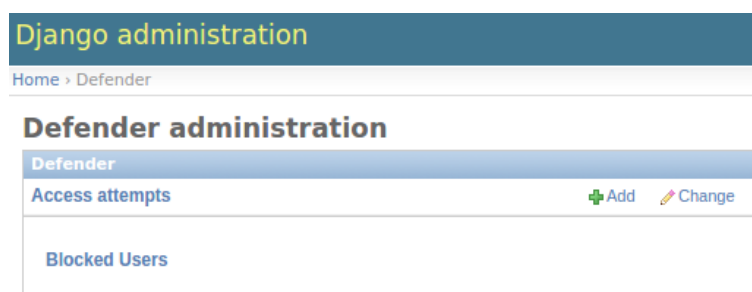
Hizkuntza bakoitzeko direktorio barruan sortutako fitxategian, guk hizkuntza anitzetan joango den testua gertuko da eta hauentzako itzulpena idatzi beharko da fitxategian bertan eskuz. Behin itzulpenak eginda, mezuak konpilatu behar dira terminaletik:

```
$ django-admin compilemessages
```

Hizkuntzaren kontrola testuinguru aldagai bezala eraman da.

## 6.1.8- Segurtasuna administrazio panelean

Administrazio panelean sartzeko super-erabiltzailea izan behar da. Baina erabiltzaile-izen eta pasahitza besterik ez da jakin behar eta Djangok defektuz ekartzen duen moduan, indarrean atera daitezke izen eta pasahitza nahi beste saiakera egin daitezke eta. Hau kontrolpean jartzeko *Django defender*<sup>14</sup> izeneko aplikazioa instalatu da. Honek, administrazio paneleko saiakera kopuru baten ostean, IP helbidea blokeatzen du denbora baterako eta saiakera guztiak log batean gordetzen ditu. Hau erabiltzeko *Redis*<sup>15</sup> instalatu behar izan da, memoriako datu baseen motor bat da. Instalatu eta martxan egon beharko da aparteko prozesu batean (redis-server). Administrazio panelean gune bat gehituko da, ikusi 6.14 irudia.



6.14 Irudia - Defender administrazio panelean

14 <https://github.com/kencochrane/django-defender>

15 <http://redis.io/>



Defender-en administrazio panelean ikus daitezke zeintzuk izan diren login saiakerak: ze IP-tatik burutu den, noiz, zein erabiltzaile izen sartu duen eta sartzea lortu badu edo ez. Erabiltzaile blokeatuak ere ikus daitezke eta hauek defektuz denbora baterako bakarrik geratzen dira blokeatuak baina denbora hau aldatu daiteke.

## 6.1.9- Produktuaren egitura

Django proiektu batean aplikazio bat baino gehiago egon daitezke eta hauek ordenatuta mantentzea komeni da. Kasu honetan aplikazio bakarra sortu da baina fitxategiak ordenatu egin dira. Teknologia atalean aipatu bezala, URL guztiak fitxategi batean erazagutu behar dira proiektuko direktorioan. Baina aukera dago urls.py fitxategi horretako helbideak aplikazioko urls.py fitxategi bat inportatzea. Horrela proiektuko URL kopurua txikitu egiten da eta aplikazio bakoitzaren direktorioan beste urls.py bat sortu da aplikazioaren helbide guztiekin.

Adibidez proiektuko urls.py fitxategiko helbideak:

```
url(r'^infoTurist/', include('infoTurist.urls', namespace="infoTurist")),
```

Horrela nabigazio barran `<ip>:<portua>/infoTurist/edozer` idazten denean infoTurist.urls fitxategian dauden URL-ak begiratuko ditu, zein negozio logikako metodori deitu behar dion jakiteko.

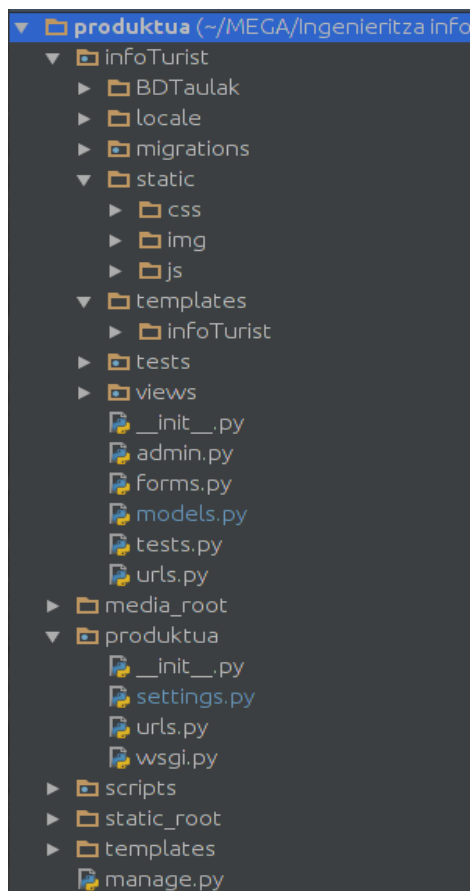
Infoturist.urls fitxategiko adibidea:

```
url(r'^kontaktua/$', viewsOrokorra.kontaktua, name='kontaktatu'),
```

```
url(r'^edozer/$', viewsOrokorra.edozer, name='edozer'),
```

Adibideko kasuan “viewsOrokorra” fitxategiko “Edozer” metodoari egingo zaio dei.

Proiektua horrela egituratu da gero aplikazio gehiago sartzeko aukera zabalik uzteko eta fitxategiak ordenatuta mantentzeko. Proiektuaren amaierako egitura ikusteko ikusi 6.15 irudia.



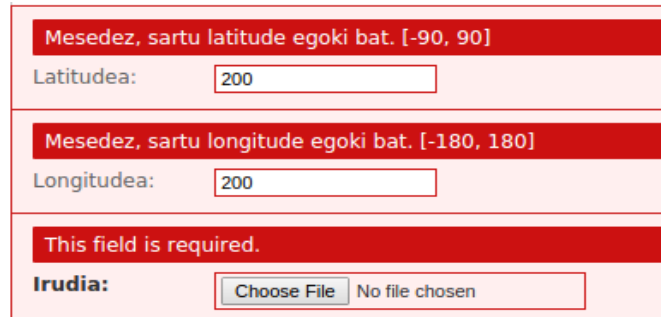
**6.15 Irudia** - Proiektuaren egitura

### 6.1.10- Probak

Hasiera batean proba unitarioak egitea pentsatu bazen ere, ezin izan dira proba unitario hauek egin aplikazioaren ezaugarriengatik (txantiloiak bistatik jasotako informazioa ezin zelako konparatu besteak beste). Horregatik eskuzko probak egin dira garapena aurrera joan ahala. Proba hauek egoera kritikoak izan daitezkeenetan egin dira batez ere, adibidez, erabiltzaileak datuak sartu behar dituen kasuetan.

Toki bat sartzeko orduan hainbat informazio sartu behar du erabiltzaileak eta informazio hau forms.py fitxategian dagoen formulario baten bidez eskatzen da (Ikusi E.2 eranskina). Formularioko balidazioak ondo funtzionatzen dutela ikus daiteke 6.16 irudian, latitude eta longitudeak tartetik sartzean errore mezua azaltzen du eta irudia derrigorrezkoa dela adierazten dio erabiltzaileari (ereduan beharrezko gisa dagoelako erazagutua). Testuzko informazioan erabiltzaileak sartzan duenarekin arduratu beharrik ez dago, Django bera arduratzen delako hau tratatzeaz. Adibidez

erabiltzailerren batek webgunea 'kutxatseko' asmos deskribapen edo iritziren batean '<script>alert(1)</script>' idatziko balu, aplikazioak ez luke agindu bezala hartuko (XSS) eta testu soil gisa agertuko litzateke.



Mesedez, sartu latitude egoki bat. [-90, 90]  
Latitudea: 200

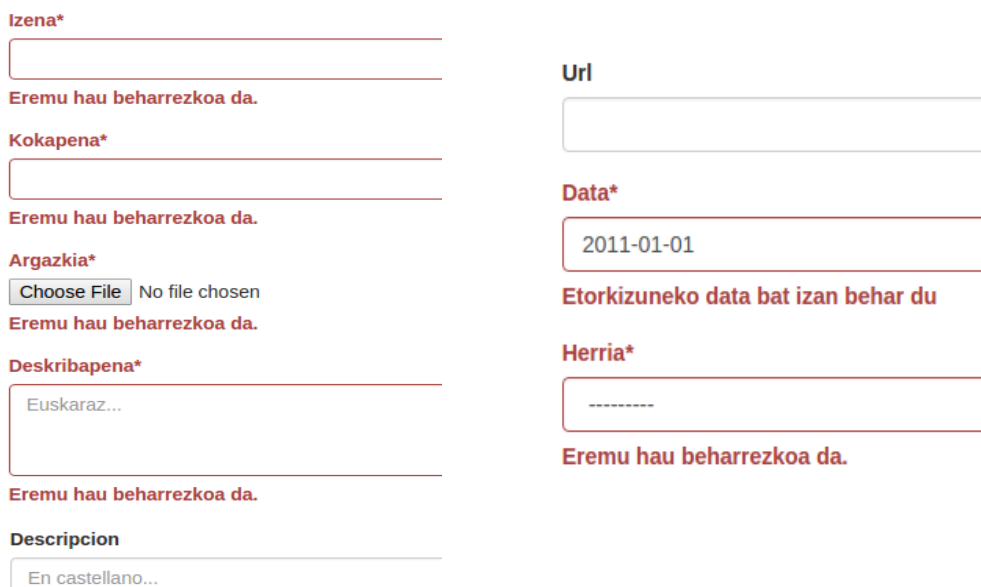
Mesedez, sartu longitude egoki bat. [-180, 180]  
Longitudea: 200

This field is required.  
Irudia: Choose File No file chosen

**6.16 Irudia** - Tokiaren formularioko errore mezuak

Jaien formularioarekin ere proba berdinak egin dira, testu kutxak hutsik utzita, baliozkoa ez den informazioa sartuta... Ikusi 6.17 eta 6.18 irudiak.

## Jai berri bat sartu



Izena\*  
Eremu hau beharrezkoa da.

Kokapena\*  
Eremu hau beharrezkoa da.

Argazkia\*  
Choose File No file chosen  
Eremu hau beharrezkoa da.

Deskribapena\*  
Euskaraz...  
Eremu hau beharrezkoa da.

Descripcion  
En castellano...  
Eremu hau beharrezkoa da.

Url  
Eremu hau beharrezkoa da.

Data\*  
2011-01-01  
Etorkizuneko data bat izan behar du

Herria\*  
-----  
Eremu hau beharrezkoa da.

**6.17 eta 6.18 irudiak** - Jai formularioaren errore mezuak

Ibilbideei dagokien probak egin ostean erabaki zen *IbilbideTokia* eredu sortzea. Probetan ikusten zen tokiak gehitzen zirela ibilbidera baina ez zuela sartze-ordena mantentzen eta beraz, ibilbidea izan beharrean tokien multzoa zen. Eredu laguntzaile berri horrek, ibilbide bat eta tokiak lotzen ditu sartze-ordena mantenduz.

Baina aplikazioan probarik gehien izan dituztenak datu meatzaritzako gidoiak izan dira, hauen emaitza, esperotakoa izan behar delako. Horregatik sortzen dituzte gidoi gehienek .html edo .txt formatuko fitxategiak, pertsona batek emaitzak modu

errazean bistaratu al izateko.

## **6.2- Datu meatzaritzaren implementazioa**

---

Datu meatzaritza egiteko, erabiltzaile erregistratuek sortzen dituzten datuak erabiltzen dira. Datu hauek kuantitatiboak izatea komeni denez, erabiltzaileei 1-5 bitartean ebaluatzeko eskatzen zaie. Iritzi hau emateko modu asko ikusi dira aurreko puntuetan: lehen aldiz login egitean, profiletik eta iritzia idaztean. Aukera guztietan adierazi behar da 1-5 bitarteko puntuazioa eta 3 baino handiagoa den kasuetan, gustuko bezala kontsideratu daiteke.

Gomendioak egiteko, hainbat modu daude eta horiek prozesatzeko 6 script edo gidoi idatzi dira eta haietako bakoitzak modu ezberdinean prozesatuko ditu datuak emaitza ezberdinak lortzeko. Eskerrak Basilio Sierrari atal honetan emandako laguntzarengatik.

Gidoien lista:

1. A motako gidoiak - Tokiak gomendatzeko tokietan
2. B motako gidoiak - Erabiltzaileei arauen bidez gomendatzeko
3. C motako gidoiak - Tokiak erabiltzaileei pertsonalizatzeko
4. D motako gidoiak - Herriko TOP-ak lortzeko
5. E motako gidoiak - Antzeko erabiltzaileekin lotzeko
6. F motako gidoiak - Antzeko tokiak gomendatzeko

Aipatutako gidoi lista banan-banan ikusiko da zer egiten duten hobeto ulertzeko. Gidoi gehienek MongoDB datu basean uzten dute beren emaitza baina beste batzuek, datu basean idazteaz aparte, fitxategietan ere idazten dute emaitza sistemako administrariak errazago ikus ditzan hauek.

Erabiltzaileak iritzia ematen duenean, MongoDB-n ere gordetzen da iritziaren zenbait datu gero hauekin lan egiteko. MongoDB-ko `'user_gustokoak'` fitxategian

gordetzen dira erabiltzailearen gustukoak direnak eta ez direnak. Honelako dokumentuez osatuta dago fitxategia:

```
user_doc = {
    'user_id': int(user_id),
    'gustoko_etiketak': [],
    'gustoko_tokiak': [],
    'gustoko_probintziak': [],
    'gustoko_herriak': [],
    'gustoko_jaiak': [],
    'ez_gustoko_etiketak': [],
    'ez_gustoko_tokiak': [],
    'ez_gustoko_probintziak': [],
    'ez_gustoko_herriak': [],
    'ez_gustoko_jaiak': []
}
```

Atalik interesgarriena gustuko tokiena da atal honetan eta honelako egitura dauka:

```
{'tokia_id':int(mota_id), 'puntuazioa':int(puntuak), 'ze_herritan':int(tokia.ze_herritan.id)}
```

Hurrengo gidoiek aipatutako fitxategitik hartuko dute behar duten informazioa.

Gidoi hauek kostu konputazional handia daukate eta horregatik egunean behin exekutatu dira printzipioz. Webgunean sortutako datu multzoa oso handia izango balitz, maiztasun hau handiagotuko litzateke. Beraz, gomendioak egunean behin eguneratuko dira.

### 6.2.1- A motako gidoiak - Tokiak gomendatzeko tokietan

Atal honetan 3 gidoi ezberdin bereizten dira: A1, A2 eta A3. Lehen gidoiaren emaitza erabiltzen du bigarrenak eta berdin hirugarrenak.

- A1 gidoia - Taulak sortu

Gidoi honi parametro bezala sistemako herriaren identifikatzailea pasa behar zaio, esaterako:

```
$ ./manage.py runscript A1-sortuTaulak --script-args=herria_id 1
```

Gidoi honen helburua herri bateko tokiak prozesatzea da eta horretarako toki hori gustuko izan dutenen artean ze beste toki gustuko izan duten erabaki behar da. Horretarako, herriko toki bakoitzerako hau egiten da:

Lehendabizi herria gustuko izan duten erabiltzaileen informazioa (beste gustuko tokiak) jasotzen dira eta hauek, PyMongok ahalbideratzen duen *map\_reduce* funtzioari pasatzen zaizkio kontaketa egiteko.

```
result=dbh.user_gustokoak.map_reduce(map,reduce,"myresults",query={'gustoko_tokiak.tokia_id':int(tokia.id)})
```

non:

```
map = Code("function () {"
  "this.gustoko_tokiak.forEach(function(z) {"
  "  emit(z,1);"
  "});"
  "}")
  reduce = Code("function (key, values) {"
  "var total=0;"
  "for (var i=0; i < values.length; i++) {"
  "  total+=values[i];"
  "}"
  "return total;"
  "}")
```

Hau da, X tokia gustuko izan duten erabiltzaileen beste gustuko tokiak batuko dira, adibidez Y tokia ere gustuko izan dute 3 pertsonak 5 puntuko balorazioarekin, eta Z tokia 2 pertsonak 4 puntuko balorazioarekin. Lortutako emaitzak ordenatu egiten dira puntuazioaren arabera handienetik txikienera.

Adibide bezala, hauek dira gure gustukoak datuak MongoDBn, prozesatu behar ditugunak (lerro bakoitza erabiltzaile ezberdin bat da eta zutabe bakoitza iritzi ezberdin bat):

Toki A: 5 puntu	Toki B: 4 puntu	Toki D: 4 puntu	Toki E: 5 puntu	Toki F: 4 puntu
Toki A: 4 puntu	Toki D: 5 puntu	Toki E: 5 puntu		
Toki A: 5 puntu	Toki E: 5 puntu			
Toki A: 4 puntu	Toki B: 4 puntu	Toki E:5 puntu		
Toki A: 5 puntu	Toki B: 5 puntu	Toki D: 5 puntu	Toki F: 5 puntu	
Toki A: 4 puntu	Toki F: 5 puntu			

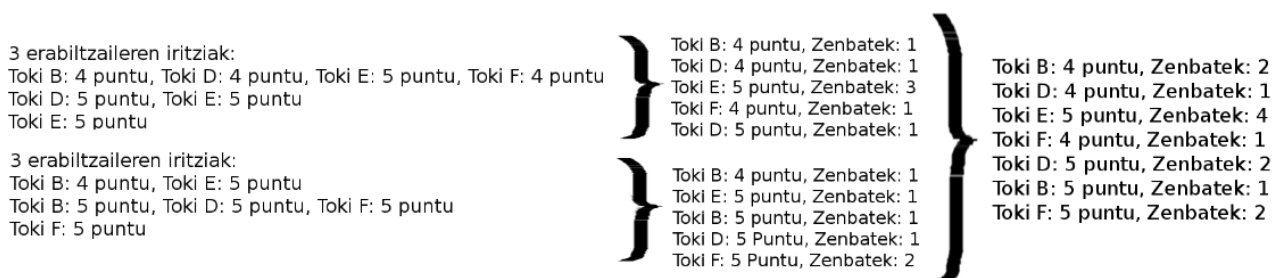
**6.1 Taula** - Gustukoak taula adibidea

Aurreko taulatik (6.1 taula) A tokian gomendatzeko tokiak jaso nahi dira. Lehen gidoi honek egiten duena gainontzeko iritziak zenbatzea da. A tokiarentzako emaitza ikusteko, ikusi 6.2 taula.

Tokia E	Puntuazioa: 5	Zenbatek: 4
Tokia D	Puntuazioa: 5	Zenbatek: 2
Tokia B	Puntuazioa: 5	Zenbatek: 1
Tokia F	Puntuazioa: 5	Zenbatek: 2
Tokia B	Puntuazioa: 4	Zenbatek: 2
Tokia F	Puntuazioa: 4	Zenbatek: 1
Tokia D	Puntuazioa: 4	Zenbatek: 1

**6.2 Taula** - Gustukoaren taulatik ateratako emaitzan A tokiarentzat

Aipatu bezala batuketa hori map\_reduce funtzioaren bidez egiten da. Honek eginkizunak banatu egiten ditu dituen baliabideen artean (prozesagailuak) baturak egiteko, eta azkenean prozesagailu bakoitzak bueltatzen dituen emaitzak batzen ditu (ikusi 6.19 irudia hobeto ulertzeko).



**6.19 Irudia** - Map\_reduce prozesua

Horrela, emaitzatzat toki hori gustuko izan dutenen artean gustuko izan duten beste tokien lista lortzen dugu puntuazio eta balioarekin (zenbat pertsonei gustatu zaien puntuazio horrekin).

Hasiera batean badirudi 5 puntuko balio handiena duena izan behar dela tokiko gomendatua baina ikusiko da hori ez dela beti horrela izango.

Gidoi honek emaitzak *"infoTurist/BDTaulak/herria\_id-herrilzena/Tokia\_tokia.id.txt"* izeneko fitxategian gordetzen ditu honelako egiturarekin:

```
{u'_id': {u'puntuazioa': 5, u'ze_herritan': 1, u'tokia_id': 204}, u'value': 2.0}
{u'_id': {u'puntuazioa': 5, u'ze_herritan': 1, u'tokia_id': 212}, u'value': 2.0}
```

```
{u'_id': {u'puntuazioa': 5, u'ze_herritan': 1, u'tokia_id': 214}, u'value': 1.0}  
{u'_id': {u'puntuazioa': 5, u'ze_herritan': 1, u'tokia_id': 221}, u'value': 1.0}
```

- A2 gidoia – Tauletatik gomendioak atera

Gidoi honek A1 gidoiak emandako emaitzak hartzen ditu sarrera bezala. A1 gidoiak kontaketa egiten du eta honek emaitza horietatik gomendioak atera behar ditu. Kasu honetan ere, exekutatzeko argumentu gisa herriaren identifikazioa eman behar zaio.

Hasiera batean badirudi A1 gidoiak emandako emaitzetatik era tribialean atera daitezkeela gomendioak, baina ez da horrela. Demagun toki batek 5 puntuko bozka gutxi batzuk izan dituela baina beste toki batek 4 puntuko hainbat bozka. Lortutako emaitzarekin 5 puntuko bozka gutxiak jaso dituen toki gomendatuko litzateke 4 puntukoak kontuan hartu gabe eta hori ez da modurik egokiena. Hau ekiditeko, bozken ponderaketa egiten da puntuazioak karratura eramanez.

Herriko toki bakoitzeko, aurreko gidoiak sortutako fitxategitik irakurtzen du eta bertako lerro bakoitzeko (gogoratu puntuazioaren arabera ordenatuta daudela), array batean gordetzen dira.

Sortu berri den array horretako emaitzak ponderatzen dira jarraian, horretarako puntuazioak berbidurara eramaten dira gero bozka kopuruarekin biderkatzeko. Hobeto ulertzeko toki bakoitzaren bozkek batzen dira eta baturarik altuena duten tokiak izango dira gomendatu beharko direnak.

Aurreko adibidearekin jarraituz, Toki B-k lortutako emaitza lehen gidoian: 5 puntuko bozka 1, eta 4 puntuko 2 bozka. Beraz ponderaketa eginda **Tokia = puntuak<sup>2</sup> x bozka kopurua**:

$$Toki B emaitza = 5^2 \times 1 + 4^2 \times 2$$

Horrela, 5 puntuazioko iritziak 4koek baino balio dezente altuagoa izatea lortzen da, 4 puntuko bozkek alde batera utzi gabe.

Demagun sarrerako datuak honakoak direla:

```
{u'_id': {u'puntuazioa': 5, u'ze_herritan': 1, u'tokia_id': 1}, u'value': 3.0}  
{u'_id': {u'puntuazioa': 5, u'ze_herritan': 1, u'tokia_id': 2}, u'value': 1.0}  
{u'_id': {u'puntuazioa': 4, u'ze_herritan': 1, u'tokia_id': 1}, u'value': 1.0}  
{u'_id': {u'puntuazioa': 4, u'ze_herritan': 1, u'tokia_id': 2}, u'value': 7.0}
```



Edo taula batean adierazita:

Tokia 1	5 puntu	3 pertsonak
Tokia 2	5 puntu	1 pertsona
Tokia 1	4 puntu	1 pertsona
Tokia 2	4 puntu	5 pertsonak

**6.3 Taula** - A2 gidoiaren sarrera datuen adibidea

A2 gidoiak datu horietatik ondorengo emaitzak aterako lituzke:

$$Tokia1 = 5^2 \times 3 + 4^2 \times 1$$

$$Tokia2 = 5^2 \times 1 + 4^2 \times 5$$

Hau da,

$$Tokia1 = 91$$

$$Tokia2 = 105$$

Beraz, prozesatzen ari den tokian 'Tokia 2' gomendatu beharko litzateke emaitza altuagoa lortu duelako. Ez da ahaztu behar A1 eta A2 gidoiak herri bakoitzeko toki guztietan exekutatzeko dela eta emaitzak toki bakoitzarentzat ateratzen direla. Hau prozesatzen den tokiaz aparte bi tokirekin egin da baina errealitatean hainbat toki daude eta toki bakoitzerako 3 gomendio hartzen dira.

Lortutako emaitza hauek MongoDB datu basean gordetzen dira 'gomendatutako\_tokiak\_tokian' fitxategian prozesatutako tokiaren identifikatzailearekin eta gomendio listarekin.

- A3 gidioa - emaitzen HTMLa sortu

Emaitzak MongoDB datu basean gordetzen ditu A2 gidoiak eta hauek egokiak diren edo ez ziurtatu nahi izanez gero zaila litzateke. Horregatik, pertsonak gomendioak egokiak diren ziurtatzeko HTML bat sortzen duen gidioa egin da. Honek, A2 gidoiak idatzitako 'gomendatutako\_tokiak\_tokian' fitxategia irakurtzen du eta dokumentu bakoitzeko, 4 eta 5 puntuko bozkak jasotzen ditu eta gero bozketatik ateratako gomendioak idazten ditu HTML fitxategi batean ("scripts/mongonDagona/Tokietan\_gomendatu\_tokia.html" izenarekin, ikusi 6.20 irudia).

---

Dokumentu guztia: {u'tokia\_id': 224, u'gomendatu\_tokian\_bostekoak': [{u'\_id': {u'puntuazioa': 5, u'ze': 16.0}], u'gomendatu\_tokian\_laukoak': [{u'\_id': {u'puntuazioa': 4, u'ze\_herritan': 2, u'tokia\_id': 210}, u'puntuazioa': 4, u'ze\_herritan': 4, u'tokia\_id': 218}, u'value': 1.0]}

Tokia id :224

Gomendatutakoak 5 izar: {u'\_id': {u'puntuazioa': 5, u'ze\_herritan': 1, u'tokia\_id': 214}, u'value': 1.0]}

- 214, zemalek balora olan: 1.0

Gomendatu 4 izarrez:

- 210, zemalek balora olan: 1.0
- 216, zemalek balora olan: 1.0
- 213, zemalek balora olan: 1.0
- 218, zemalek balora olan: 1.0

---

GOMENDIO PONDERATUAK:

- 214 tokiaren bozka ponderatuen batura: 25.0
- 216 tokiaren bozka ponderatuen batura: 16.0
- 218 tokiaren bozka ponderatuen batura: 16.0
- 210 tokiaren bozka ponderatuen batura: 16.0
- 213 tokiaren bozka ponderatuen batura: 16.0

## 6.20 Irudia - Tokiak\_gomendatu\_tokia.html adibide baten zatia

Beraz, gidoi honen helburua datuak sistemako administrariari errazago bistaratzear datza.

### 6.2.2- B motako gidoiak - Erabiltzaileei arauen bidez gomendatu

Mota honetako gidoiaren helburua toki batzuk elkarrekin lotzea da eta gomendio berriak sortzea erabiltzaile bakoitzarentzat. Emaidza bezala, '*erabiltzaile batek x,y eta z tokiak gustuko baditu, orduan w tokia ere gustuko izango du*' bezalakoa izango da. X,y eta z tokiak gustuko izatea premisa izango da eta w tokia gustuko izatea ondorioa.

- B1 gidioa - Bozken fitxategia sortu

Prozesua hasteko bi gauza egingo dira, alde batetik erabiltzaileen gustuko tokiek bozkak fitxategi batean gordeko dira honelako formatuan:

```
erabiltzailea::tokia::puntuak
```

Bozka hauek gustuko tokietatik jasota daude eta beraz puntuak 4 edo 5 balioa izan dezakete. Erabiltzailea eta tokia dauden tokian, hauen identifikatzaileak (zenbakiak) egongo dira.

Beste aldetik, bozka duten toki guztiak gordeko dira beste fitxategi batean honelako formatuan:

Tokiaren\_id::Tokiaren\_izena

B motatako lehenengo gidoiak aurre-prozesaketa hau besterik ez du egiten.

- B2 gidioia - Bozketatik arauak atera

Gidoi honek egiten ditu gomendioak ateratzeko beharrezkoak diren prozesaketa guztiak. Sarrera bezala B1 gidoiak sortutako fitxategiak hartzen ditu kontuan, bozken fitxategia kalkuluak egiteko eta tokien fitxategia emaitzak pertsonentzat modu ulergarriagoan bueltatzeko. Kontuan hartu behar da irakurritako fitxategiko bozka guztiak positiboak direla, hau da, tokia gustukotzat bezala hartu dutela.

Lehendabizi pertsona bakoitzari gustatzen zaizkion tokiak biltzen dira.

Erabiltzaile_id	Gustuko_tokien_id
1	206, 207, 210, 211
2	206, 207, 211
4	207, 210, 211
7	206, 207, 211

**6.4 Taula** - B2 gidioiko gustukoaren taularen adibidea

Aurreko taulatik (6.4 taula) bozka kopuru minimo bat duten tokiak bakarrik hartuko dira, adibidez 3. Beraz, 207 tokia ez da kontuan hartuko.

Aurreko taulatik arau posibleak aterako dira, esaterako 206 gustuko dutenek gustuko izango dute 207. Edo 206 eta 207 gustuko dutenek gustuko izango dute 210.

Gustuko badute... (premisia)	Gustatuko zaie... (ondorioa)
206	207
206, 207	210
206, 207	211
...	...
206	210
206, 210	211

**6.5 Taula** - B2 gidioiko arau posibleen adibidea

Arau posible batzuk izanda, interesekoak bakarrik hartuko dira eta gidoi honetan 1 luzerako premisak alde batera uztea erabaki da.

Azkenik luzera 2 edo gehiagoko premisa duten arauak egiazkoak diren edo ez begiratzen da. Horretarako, premisa bakoitza hartzen da eta erabiltzaileen gustuko tokien taulan begiratzen da ea betetzen den edo ez. Premisa betetzen bada baina ondorioa ez, premisak balioa galtzen du baina ondorioa ere betetzen bada, arauak balioa irabazten du. Azkenean balio horrek ezarritako muga bat gainditzen badu, araua egiazkoa dela esango dugu. Emaitza hau ehunekotan ere adierazi daiteke honela:

$$\frac{\text{Premisa eta ondorioa bete den aldien kopurua} - \text{premisak bete eta ondorioa ez bete aldien kopurua}}{\text{Premisa bete den aldien kopurua}}$$

Aurreko adibideko taulei erreparatuz, baliozko arauak azaldutako prozesuaren ondoren:

- 206, 207 gustuko, beraz 210 ere bai:  $\frac{1-2}{3} = -0.33$  edo -%33.
- 206, 207 gustuko, beraz 211 ere bai:  $\frac{3-0}{3} = 1$  edo %100.

Horrela, ezarritako ehuneko bat gainditzen badu, araua egokitzat hartuko da eta bestela ez. Balio duten arauan fitxategi batean idatziko dira honelako formatuaz:

```
tokiX_id,tokiY_id,..,tokiZ_id::tokiW_id
```

Hau da,

```
premisak::ondorioa
```

- B3 gidoia - Emaitzak datu basean gorde

B motako azken gidoi honek B2-n lortutako emaitzak datu basean gordetzen ditu erabiltzaileen gomendio modura baldin hauek premisa betetzen duten eta ondorioa ez badute jada ebaluatu.

Hau egiten da horretarako:

```
Bozkaren bat eman duen erabiltzaile bakoitzerako:
```

```
B2 gidoiak sortutako lerro bakoitzeko (premisak::ondorioa)
```

```
Baldin Ondorioa ez_gustuko edo gustuko badu:
```

```
Jarraitu hurrengo iteraziora
Baldin premisako tokiak gustuko baditu (premisaren bete da)
Gomendioa gorde
```

Emaitzak MongoDB datu basean gordetzen dira “*user\_gomendioak2*” izeneko fitxategian honelako egiturarekin:

```
user_doc = {
  'user_id': int(user_id),
  'gomendatzeko_tokiak2': []
}
```

eta “*gomendatzeko\_tokiak2*” arraiaren egitura honelakoa da:

```
{'tokia_id':int(gomendioa), 'ze_herritan':int(tokia.ze_herritan.id), 'premisak':str(premisak)}
```

Horrela, login egiten duen erabiltzailearentzat arin batean bilatu daitezke x,y,z beraz w bezalako gomendioak fitxategi hau kontsultatuz, non bilaketa ‘*user\_id*’ aldagaiarekin eta uneko erabiltzailearen identifikazioarekin egingo den.

### 6.2.3- C motako gidoia - Tokiak erabiltzaileei pertsonalizatu

Gidoi honek, aurrekoak motakoak bezala gomendioak sortzen ditu baina modu oso ezberdinean.

Gustuko dituen tokien artean, ikusten da ea toki horietan zeintzuk diren gomendatutako tokiak (A gidoietan sortutako emaitzak), eta toki guztien zerrenda bat egiten da A motako gidoietan sortutako emaitza ponderatuen batura eginez. Kontuan izan behar da ez direla gomendatutako jada ebaluatutako tokiak (gustuko edo ez gustukoak).

Adibide bezala ikus dezagun erabiltzaile baten prozesua.

Erabiltzaile batek gustuko ditu 207, 211 eta 202 tokiak eta hauek dira toki horiei dagozkien emaitzak A gidoitik jasota:

207 tokiarena:

```
[{gomendioa:202, puntuak:25.0}, {gomendioa:200, puntuak: 20.0}]
```

211 tokiarena:

```
[{gomendioa:207, puntuak:30.0}, {gomendioa:202, puntuak: 20.0}]
```

202 tokiarena:

```
[{gomendioa:207, puntuak:25.0}, {gomendioa:190, puntuak: 18.0}]
```

Gomendio bakoitzaren puntuen batura eginez:

202 tokia = 25+20

200 tokia = 20

207 tokia = 30+25

190 tokia = 18

Horiek izango lirateke hasiera batean gomendioak baina 202 eta 207 tokiak jada gustuko ditu, beraz hauek gomendioetatik kendu egingo dira eta berdin egingo da gomendatzeko tokiren bat ez gustukoetan badago.

Beraz adibideko erabiltzaile honentzat gomendioak izango lirateke: 200 tokia 20 punturekin eta 190 tokia 18 punturekin. Puntu gehien duten tokiek probabilitate altuagoa dute erabiltzailearen gustukoak izateko.

Emaitzak MongoDB-n gordeko dira 'user\_gomendioak1' izeneko fitxategian eta B motako gidoietan sortutako egitura izango dute.

Gainera, sistemako administrariak emaitzak errez batean ikus ditzan, emaitzak fitxategi batean ere idatziko dira "*infoTurist/DBTaulak/Users-gomend-tokiak1/username+user\_id.txt*" kokapenean. Honelakoa izango da fitxategi horien edukia:

```
[(217, 100.0), (224, 16.0), (216, 16.0), (218, 16.0)]
```

Hau da, [(tokia\_id, puntuazioa), (tokia\_id, puntuazioa)... ]

## 6.2.4- D motako gidoia - Herrietako TOPak lortu

Gidoi honen helburua, herri bakoitzerako tokien iritzi idatziak hartu eta hauen arabera herriko tokirik honenak zeintzuk diren erabakitzea da. Beraz ez dira kontuan hartuko profiletik izartxo bidez emandako ebaluazioak zati honetarako. Toki batean idatzitako iritziak bozka negatibo asko baditu (10 bozka eta hauetatik 5 baino gehiago 'ez egokia' markatuta), iritzi hori ez da kontuan hartuko tokiaren puntuaziorako. Tokiaren puntuazioa zein den erabakitzeke honakoa izango da formula:

$$X \text{ tokiaren puntuak} = \sum \text{iritzia} \cdot \text{puntuazioa}^2 + \text{iritzia} \cdot \text{balorazioa}$$

Hau da, tokiak duen iritzi bakoitzeko (aipatutako ez egokiak baztertuta) honen puntuazioaren karratua eta jendeak emandako balorazioa gehituz lortzen den batukaria izango da.

Puntuazioen emaitza fitxategi batean gordeko da "infoTurist/BDTaulak/HerrietakoTOPak/+herria.id+--+herria.izena+.txt" izeneko fitxategian honelako formatuarekin (herriko balorazio hoberena lortu duten ordenan):

```
[(tokia_id, puntuazioa), (tokia_id, puntuazioa)]
```

```
adibidez: [(217L, 25L), (216L, 16L), (210L, 16L), (215L, 0)]
```

Algoritmoaren kostua honelakoa da:

$$x \cdot y \cdot z \quad \text{non } x = \text{herri kopurua}, y = \text{toki kopurua} \text{ eta } z = \text{iritzi kopurua}$$

Aipatutako algoritmoa hobeto ulertzeko adibide sinple batekin ikusiko da. Demagun herri bateko tokirik baloratuenak jaso nahi direla eta hauek direla bertako tokiak eta iritziak:

Tokia 1	Iritzia 1: 4 puntu (4 bozka, 2 puntu)
	Iritzia 2: 1 puntu (11 bozka, -2 puntu)
	Iritzia 3: 4 puntu (0 bozka, 0 puntu)
	Iritzia 4: 5 puntu (2 bozka, 2 puntu)
Tokia 2	Iritzia 5: 5 puntu (2 bozka, 0 puntu)
	Iritzia 6: 4 puntu (3 bozka, 1 puntu)
	Iritzia 7: 3 puntu (5 bozka, -2 puntu)
	Iritzia 8: 5 puntu (0 bozka, 0 puntu)

**6.6 Taula** - D gidoia adibidea. Herri bateko tokien iritziak

Bi tokitako herri horretan zein den tokirik baloratuena jakiteko beraz:

$Tokia1 = (4^2+2) + (4^2+0) + (5^2+2)$  hau da, 61 puntu (kontuan izan 2. iritzia baztertu dela 10 bozka baino gehiago izatearren eta bozken puntuazio orokorra negatiboa delako).

$Tokia2 = (5^2+0) + (4^2+1) + (3^2+(-2)) + (5^2+0)$  hau da, 74 puntu

Beraz, adibideko herri honetarako tokirik baloratuena Tokia 2 izango litzateke. Errealitatean hainbat toki gehiago daude eta beraz, TOP hauek ez dira batera murrizten eta puntuazio gehien izan dituzten 3rak hartuko dira.

### 6.2.5- E motako gidoia - Antzeko erabiltzaileak lotu

Gidoi honen helburua erabiltzaileei beren antzeko gustuak dituzten erabiltzaileak aurkeztea da, horrela, bere antzekoa den erabiltzaileak sartutako tokiak eta ibilbideak kontsultatzeko aukera izango du (beste erabiltzailearen profila kontsulta). Baliteke X erabiltzailearentzat Y erabiltzailea interesgarria izatea baina ez alderantziz jarraian ikusiko dugun bezala. Lotura hau norabide bakarrekoa izan daiteke.

Lehendabizi MongoDB-ko 'user\_gustokoak' fitxategian dauden dokumentu guztiak hartzen ditu (dokumentu bat = erabiltzaile bat).



Fitxategiko dokumentu bakoitzerako (informazioa daukan erabiltzaile bakoitzerako), beste erabiltzaile guztiak hartzen dira lehenengoarekin konparatzeko. Antzekoak izango dira X eta Y erabiltzaileak baldin X-ren gustukoak eta Y-ren gustukoak antzekoak badira eta berdin ez gustukoen kasuan. Zehatzago izanda, kontagailu batean gordeko da +1 eginez X-ri gustuko zaionen bat Y-k ere gustuko badu eta berdin egingo da X-k ez duena gustuko Y-k ere ez duenean gustuko. Baina, kontagailu horretan bertan egingo da -1 baldin X-ren gustukoa dena Y-ren ez gustukoetan badago eta alderantziz. X-ren gustuko edo ez gustukoa Y-k ez baldin badu baloratu, orduan +0 egingo da. Azkenean,

$$\text{puntuak} = X\text{ren gustuko eta ez gustukoen kopurua} - \text{kontagailua}$$

X eta Y antzekoak izango dira baldin:

$$\frac{X\text{ren gustuko eta ez gustukoen kopurua}}{2} > \text{puntuak}$$

Eta antzekoak badira, orduan lehenengo erabiltzaileari antzekoa den erabiltzaile hau gomendatuko zaio MongoDB-ko 'user\_antzekoak' izeneko fitxategian gordez. Aurretik erabiltzailearen dokumenturik ez badago datu basean, sortu egingo da eta honelako egitura izango du:

```
user_doc = {  
  'user_id': int(doc['user_id']),  
  'antzeko_userrak': [],  
}
```

Non 'antzeko\_userrak'-en egitura ondorengoa izanik:

```
{'antzeko_user_id':int('user_id'), 'antzekotasun-errorea':int(puntuak)}
```

Emaitza Mongo datu basean gortzeaz gain eta sistemako administrariarentzat datuak ikus-errazago izan daitezen, emaitzak fitxategi batean ere gordetzen dira eta grafo eran .png bat sortzen du zeinetan nodoak erabiltzaileak diren eta loturak argi ikusten diren gezi eran. Emaitzen fitxategi hori "scripts/mongonDagona/User\_antzekoak.html" izeneko dokumentuan gordetzen da eta honelako egitura dauka:

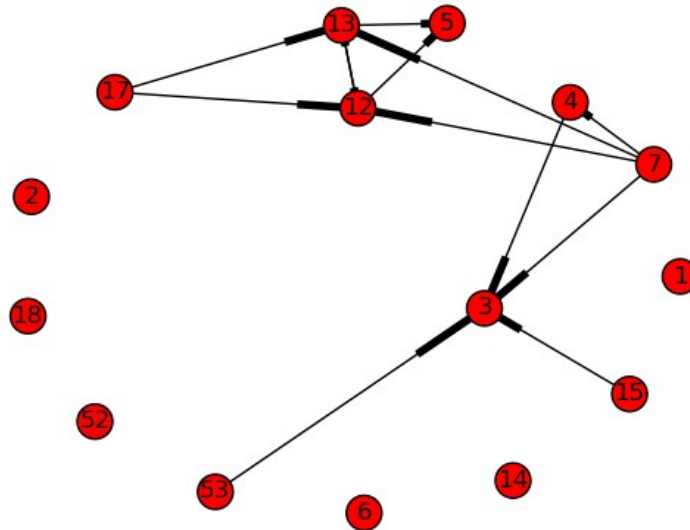
X-ren antzeko userrak:

```
{'antzekotasun-errorea': 0, 'antzeko_user_id': 3}  
{'antzekotasun-errorea': 0, 'antzeko_user_id': 4}  
{'antzekotasun-errorea': 2, 'antzeko_user_id': 12}  
{'antzekotasun-errorea': 0, 'antzeko_user_id': 13}  
-----
```

```

Y-ren antzeko userrak:
-----
Z-ren antzeko userrak:
(...)
W-ren antzeko userrak:
{'antzekotasun-errorea': 1, 'antzeko_user_id': 12}
{'antzekotasun-errorea': 0, 'antzeko_user_id': 13}
    
```

Bestalde, grafoaren itxura honelakoa izan daiteke (ikusi irudia 6.21):



**6.21 Irudia** - Antzeko erabiltzaileen grafoa

6.21 irudian ikus daiteke nola sisteman 15 erabiltzaile dauden eta beraietako batzuk lotuta dauden. Esaterako, 7. erabiltzailearentzat (nodoa) 4, 12, 13 eta 3 erabiltzaileak izango dira gomendatuak baina hauentzat ez da 7-a gomendatuta egongo. Erabiltzailerik gomendatuena 3. erabiltzailea dela ikus daiteke 4 erabiltzailerik gomendatuz.

Jarraian agertzen den adibidean ikusten da zergatik erabiltzaileen lotura hau norabide bakarrekoa den. Demagun X eta Y erabiltzaileak daudela eta hauek direla beren gustuko eta ez gustukoak:

	Gustuko tokien identifikazioak	Ez gustuko tokien identifikazioak
X erabiltzailea	1, 2, 3, 9	4
Y erabiltzailea	1, 2, 5, 6, 7, 9	3, 4, 8

**6.7 Taula** - E gidoia adibidea, X eta Y gustuko eta ez gustukoak

X erabiltzailearentzako Y gomendatu behar da?

Kontagailua = 0

Biek dute gustuko 1, 2 eta 9 tokiak beraz kontagailua +3

Biek dute ez gustuko 4 tokiak beraz kontagailua +1

X erabiltzaileak 3 tokiak gustuko baina Y-k ez, beraz kontagailua -1

Kontagailua = 3 beraz.

X-ren gustuko eta ez gustuko kopurua = 5

Formula aplikatuz (  $puntuak = Xren\ gustuko\ eta\ ez\ gustuko\ kopurua - kontagailua$  )

$$puntuak = 5 - 3 \text{ beraz, } puntuak = 2$$

Betetzen da honakoa  $\frac{Xren\ gustuko\ eta\ ez\ gustuko\ kopurua}{2} > puntuak$  ?

$$\frac{5}{2} > 2 \text{ Bai, beraz gomendatu behar da.}$$

Y erabiltzailearentzako X gomendatu behar da?

Kontagailua = 0

Biek dute gustuko 1, 2 eta 9 tokiak beraz kontagailua +3

Biek dute ez gustuko 4 tokiak beraz kontagailua +1

Y erabiltzaileak 3 ez gustuko baina X-k gustuko, beraz kontagailua -1

Kontagailua = 3 beraz

Y-ren gustuko eta ez gustuko kopurua = 9

Formula aplikatuz (  $puntuak = Xren\ gustuko\ eta\ ez\ gustuko\ kopurua - kontagailua$  )

$$puntuak = 9 - 3 \text{ beraz, } puntuak = 6$$

Betetzen da honakoa  $\frac{Xren\ gustuko\ eta\ ez\ gustuko\ kopurua}{2} > puntuak$  ?

$$\frac{9}{2} > 6 \text{ Ez, beraz ez da gomendatu behar.}$$

Ondorioz, aurreko adibidean ikusi da erabiltzaileen arteko lotura norabide bakarrekoa izan daitekeela. Normalean iritzi gutxien eman duenarentzat egongo dira gomendatzeko erabiltzaileak, azken hauen bidea jarraitu dezaketelako.

Algoritmo honen kostua gutxienez  $n^2$  motakoa da, erabiltzaile bakoitza gainontzeko erabiltzaileekin konparatu behar delako.

## 6.2.6- F motako gidoia - Antzeko tokien gomendioak

Azken gidoi honen helburua erabiltzaile bati berak iritzirik eman ez duen herri bateko tokiak gomendatzea da. Horretarako gustuko dituen tokiak hartzen dira abiapuntutzat eta hauetatik deduzitzen dira gustukoak izan daitezkeen tokiak beste herri batean.

Honakoa da ideia: Toki guztiek etiketak dituztenez, gustukoak diren tokien etiketak zeintzuk diren ikusiko dira eta horrela, etiketei pisu ezberdina emango zaie (gehien agertzen den etiketa garrantzitsuena izanda, beraz pisu gehien). Behin etiketei pisua esleituta, toki bat gustuko izateko eduki beharreko puntuazioa ezartzen da (behe atalasea) eta horretarako, gustuko tokien artean zein tokik lortzen duen punturik gutxien ikusiko da etiketen pisuen batura eginda. Horrela jakingo da toki batek gutxienez izan behar duen puntuazioa etiketen pisua kontuan hartuz. Behin behe atalase hori ezarrita, gainontzeko ebaluatu gabeko tokiek zein puntuazio lortzen duten ikusiko da eta atalase hori gainditzen badute, gomendio bezala gehituko da toki hori.

Idea zein den jakinda, honela egin da:

MongoDB datu basetik '*user\_gustokoak*' fitxategitik erabiltzaile guztien dokumentuak hartzen dira eta erabiltzaile bakoitzerako egiten da prozesaketa. Erabiltzaileak gustuko duen toki bakoitzeko, etiketa bakoitza zenbat aldiz agertzen den kontatzen da (ikusi 6.8 taula).

Gustuko tokiak	Tokien etiketak
Toki 1	'Hondartza', 'Jatetxea'
Toki 2	'Hondartza', 'Gaua pasatu', 'Familian'
Toki 3	'Jatetxea', 'Familian'
Toki 4	'Familian', 'Hondartza'
Etiketen maiztasuna (etiketen puntuazioa)	'Hondartza': 3 aldiz 'Jatetxea': 2 aldiz 'Gaua pasatu': 1 aldiz 'Familian': 3 aldiz

**6.8 Taula** - F gidoiko gustuko tokien adibidea

Behin gustuko tokien etiketen maiztasuna jakinda, toki batek behar dituen etiketa minimoak zeintzuk diren erabakiko da. Horretarako gustuko dituen tokiak hartuko dira proba kasu bezala eta emaitzarik baxuena izango da puntuazio minimoa.

Hau izango da formula:

$$X \text{ tokiaren puntuazioa} = \sum \text{etiketa} \cdot \text{puntuazioa}^2$$

Beste era batera esanda, tokiak dituen etiketa bakoitzak lortutako puntuazioaren karratua. Adibidea jarraituz emaitza hau izango litzateke (ikusi 6.9 taula).

Gustuko tokiak	Tokien puntuazioa
Toki 1	'Hondartza', 'Jatetxea', beraz $3^2 + 2^2 = 13$ puntu
Toki 2	'Hondartza', 'Gaua pasatu', 'Familiar', beraz $3^2 + 1^2 + 3^2 = 19$ puntu
Toki 3	'Jatetxea', 'Familiar', beraz $2^2 + 3^2 = 13$ puntu
Toki 4	'Familiar', 'Hondartza', beraz $3^2 + 3^2 = 18$ puntu
Behe atalasea	<b>13 puntu</b>

**6.9 Taula** - F gidoia adibidea. Behe atalasearen kalkulua

6.9 taulan ikusi den bezala, behe atalasearen kalkulua egin da. Orain, kalkulu berdinak egingo dira ebaluatu gabe dauden toki guztietan eta behe atalasea gainditzen duten tokiak gomendatuko dira. Gustuko tokietan agertzen ez den etiketaren bat agertzen bada prozesatzen ari den tokiren batean, etiketa horren puntuazioa 0 izango da. Emaitzen adibide bezala ikusi 6.10 taula.

Prozesatzeko tokiak	Tokien etiketak eta puntuazioa
Toki 5	'Hondartza', 'Umeentzako', beraz $3^2 + 0^2 = 9$ puntu. <b>Ez du atalasea gainditu</b> beraz ez da gomendatuko.
Toki 6	'Familiar', 'Hondartza', 'Umeentzako', beraz $3^2 + 3^2 = 18$ puntu <b>Atalasea gainditu du</b> beraz gomendatuko zaio.

**6.10 Taula** - F gidoia adibidea. Ebaluatu gabeko tokien prozesaketa

Emaitzak 'user\_gomendioak3' izeneko fitxategian gordeko dira MongoDB datu basean.

## 6.3- Diseinua eta implementazioaren arteko loturak

Atal honetan aipatutako implementazioak eta sekuentzi diagramen lotura egingo da, zein diagrametan zer erabiltzen den ikusteko.

- Tokia ikusi / iritzia idatzi

5.3.1 puntuko sekuentzi diagraman azaldu den bezala, toki baten deskribapena kargatzean, 6.1.6 puntuan ikusitako ajax deien bidez aplikazioko negozio logikari zenbait gomendio eskatzen zaizkio. Sekuentzi diagraman agertzen ez den beste dei bat ere egiten zaio 6.1.5 puntuko Google maps-i, tokiko latitudea eta longitudea pasatuz. Honen emaitza maparentzat gordetako gunean kokatuko da (mapa bera). 6.1.3 puntuko txantiloiei dagokienez, negozio logikak pasatutako datuetan zehar iterazioak egin daitezkeela ikusi da, *'tokia\_detail.html'* txantiloia honetan, *tokia* ereduak da nagusia, baina badaude gehiagorik ere: *ibilbidea* aldagaia (baldin ibilbide baten barnean badago), *nireibilbideak* aldagaia (erabiltzailearen ibilbideak dituenak), *iritziak* aldagaia (tokiaren orriko iritziekin) eta behin kargatzean gomendioak. Tokien deskribapeneko txantiloia ikusteko, joan E.4 eranskinera (E.4.1 atala). Gomendioen egitura honelakoa da tokietan:

```
[{'izena':tokiBat.izena, 'herria':tokiBat.ze_herritan.izena,  
  'idea':tokiBat.id,'irudia':tokiBat.irudia.url}, ...]
```

Json formatuan itzultzen ditu gomendioak eta emaitza javascript bidez kokatzen da dagokion tokian. Gomendioen informazio guztia ez da behar (tokien informazio guztia), gomendio atalean bistaratuko direnekin soilik nahikoa da: izena, herria, identifikazio zenbakia eta irudia. Gomendioak mongoDB datu basetik eskuratzen dira, 6.2.1 ataleko emaitzetatik hain zuzen ere. Ikusi da emaitza horiek ordenaturik gordetzen direla (gomendatuena listako lehenengoa izanda) beraz exekuzio orduko denbora murrizten da.

Tokiaren deskribapenean kontsulta egiteko soilik sartuta (5.3.1 sekuentzi diagraman POST gabe), MySQL datu basean ez da ezer aldatzen, kontsultak bakarrik egiten direlako, baina toki baten informazioa eskatzen den bakoitzean eguneratuko da MongoDB-ko *'ikusitako\_tokiak'* fitxategian. Baina iritzia idazten den momentuan, helburu URLa berdina den arren, POST bidez informazioa bidaltzen denez (iritziaren

datuak), sekuentzi diagramako POST baldintzaren eragiketak exekutatu dira. Iritziaren balorazioa positiboa bada, erabiltzaileak gustuko duela adierazten du eta beraz, MongoDB-ko *'user\_gustokoak'* dokumentuko erabiltzaileari dagokion fitxategian gordeko da tokia. Gero datu hauekin erabiltzailearen gustukoak izan daitezkeen tokiak deduzituko dira eta hauek baliagarriak izango dira 4. atalean ikusitako zenbait erabilpen praktikotarako.

- Hasiera orriko kontsulta

Hasiera orria kargatzean, batez ere erabiltzaile erregistratuen kasuan, MongoDB-ko hainbat informazio eskuratzen da. Erregistratuen kasuan, ea baloraziorik eginda daukan ikusten da (balorazio positiboak) eta hau, 6.2 atalean adierazi den *'user\_gustokoak'* fitxategian begiratzen da ea erabiltzailearen dokumentua existitzen den. Ez bada existitzen txantiloia bat kargatuko da (*userGustokoakJaso.html*) eta bestela hasiera orriko txantiloia (*base.html*).

Hasiera orrian kargatzen dira herrietako tokirik baloratuenak, hurrengo hilabetean egongo diren jaiak, ikusitako azken tokiak eta erregistratuen kasuan gomendio mota bi eta antzeko erabiltzaileak ere (baldin baditu).

Herrietako baloratuenak 6.2.4 ataleko emaitzetatik jasotzen dira (tokirik gehien dituzten herriak lehen postuetan agertuko dira), ikusitako azken tokiak berriz MongoDB-ko *'ikusitako\_tokiak'* fitxategian sartu diren azkenak hartuko ditu (ikusi diren azken tokiak).

Gomendioen kasuan bi tokitatik eskuratzen dira hauek, alde batetik 6.2.3 atalean sortutako gomendio pertsonalizatuak *'user\_gomendioak1'* dokumentutik eta bestetik, 6.2.2 ataleko emaitza izan diren *'gomendatzeko\_tokiak2'* dokumentutik. Antzeko erabiltzaileak 6.2.5 atalean lortzen diren emaitzetatik eskuratzen dira.

- Plana sortu

Plana sortzeko erabilpen kasua da 6.2.6 ataleko gidoiak sortzen dituen emaitzak erabiltzen dituen bakarra. Sistemaren plana sortzeko eskatu duen herriaren baloraziorik egin ez badu erabiltzaileak, orduan *'user\_gomendioak3'* fitxategian gomendiorik dagoen begiratzen da. Erabilpen kasu honek ez du datu basea aldatzen, kontsulta soilak egiten direlako. Agertzen diren tokietan klikatu ezkerreko, 'Tokia ikusi' sekuentzia

diagrama exekutatu da.

Gidoiek beraz, MongoDB-ko eta zerbitzariko zenbait fitxategi sortzen dituzte (beste fitxategi batzuetako informaziotik abiatuta) eta gero Django aplikazioak irakurtzen ditu gidoiek sortutako fitxategiak, azken hauek beharko dituzten fitxategiak idazten dituelarik. Hau da, gidoiek behar duten informazioa Djangok sortzen du eta Djangok behar duen informazioaren zati bat gidoiek sortzen dute eta beren arteko komunikazioa MongoDB eta fitxategien bidez egiten da. MySQL datu basearekin Django soilik komunikatzen da.



# 7

---

## Internetesa migrazioa

Sortutako produktua martxan ikusteko internetetik atzigarri egon behar du eta irismenean kalitate gehigarri gisa dagoen puntu hau ere egin da.

### 7.1- Domeinua eta makina birtuala

---

*Puntueus*<sup>16</sup> fundazioak eskutitz bidez jakinarazi zuen promozio bat zegoela 2016ko martxoan, “.eus” domeinuak 15 euroren trukean baitzeuden. Promozio hau *Izenpe*<sup>17</sup>-k egindakoa zen eta interneterako migrazioa egiteko, zer hobeto '.eus' domeinu bat baino? Horregatik promozioa aprobeztatuz “*infoturist.eus*” domeinua eskuratu zen urte beterako.

Domeinu izena alde batera utziz, proiektua nonbait kokatu beharra zegoen internetetik atzigarri egoteko. Hainbat aukera konparatuz, egokiena *Amazon*<sup>18</sup>-en makina birtual bat kontratatzea zela erabaki zen. Makina birtual hau urte betez ordaindu beharrik ez zegoenez eta produktua eraiki zen (Ubuntu) sistema eragile berdina izan dezakeenez (Django, extentsioak eta MongoDB instalatzeko egokia), aukerarik onena izan zen.

Amazonek hainbat baliabide jartzen ditu eskura baina erabili den bakarra makina birtual baten instantzia bat izan da. Makina hau, proiektua garatu den makina lokalaren sistema eragile berean jarri da (Ubuntu 14.04) eta terminaletik SSH\* (konputagailu-sarearen bidez urruneko gailuei sarbide izateko programa eta sare protokoloa) bidez egiten da konexioa. Konexioa egiteko aipatzekoa da ez dela pasahitz batean oinarritzen baizik eta RSA klabe pribatua eta hau daukan fitxategia izango da

16 <http://www.domeinuak.eus/>

17 <http://www.izenpe.com/>

18 <https://aws.amazon.com/>

konexioa zabaltzeaz arduratuko dena.

Amazon makinan, lokalean egindako instalazio berdinak egin behar izan dira eta gero gure makinako proiektua bertara inportatu. Hau egitean hainbat birkonfigurazio egin behar izan dira eta hauetako batzuk arazoak eman dituzte hurrengo puntuan ikusiko den bezala.

Domeinua eta markina birtuala lotzeko, domeinua kontratatu den hornitzailearen administrazio panelean jarri behar izan da domeinuaren edukia zein IP-tan dagoen kokatua 'A' erregistroa editatuz. Horretarako, 'A' erregistro horretan makina birtualaren IP publikoa jarrita nahikoa da (ikus 7.1 irudia).

<input checked="" type="checkbox"/> Entrada DNS	Tipo	Valor	Modificar
[REDACTED]	[REDACTED]	[REDACTED]	[icon]
<input type="checkbox"/> infoturist.eus	A	52.29.33.245 [icon]	[icon]

**7.1 Irudia** - Domeinua eta makina lotzen dituen erregistroa

Makina birtualean *Apache*<sup>19</sup> 80 (web portua) ez den beste portu bat entzuten jarri behar izan da, 80 portua *Djangok* erabil dezan. Gainera, aldi berean 3 prozesu martxan jarri behar izan dira eta horretarako *Screen*<sup>20</sup> aplikazioa erabili da, zeinek terminal batetik beste terminal batzuk irekitzea ahalbideratzen duen eta terminal hauetako exekuzioa martxan jarraitzen da nahiz eta terminalaren leihoa itxi. Horrela 3 terminal ipini dira martxan, lehenak Django aplikazioa exekutatzeko du, bigarrenak *Redis-Server* eta azkenak aldiro-aldiro exekutatzeko diren gidoiak.

## 7.2- Izandako arazoak

---

Makina lokaletik birtualera pasatzean hainbat konfigurazio aldatu behar izan dira eta hauek arazoak ekarri dituzte. Konfigurazio fitxategian esaterako, datu basearen ezarpenak aldatu behar izan dira. Gainera Google eta Facebook-en zerbitzuek ere funtzionatzeari utzi diote hauen konfigurazio panelean ezarritako identifikatzaileak ez zetozelako bat URLarekin. Horregatik beste aplikazio berri bat bezala konfiguratu behar izan da Google-en mapak erabili al izateko eta Facebook-en logina

<sup>19</sup> <http://www.apache.org/>

<sup>20</sup> <https://help.ubuntu.com/community/Screen>

ahalbideratzeko.

Google-ren kasuan ez da aparteko arazorik egon aplikazio berri bat sortzean. Dagokion kodea ordeztu, URL berria baimendu eta mapak ondo bistaratzen dira interneten dagoen webgunean ere. Atal honetan arazoa mugikorretik sartzean dator. Mugikorretan hasiera batean mapa ez zen kargatzen, javascript kodea exekutatu ere ez zela egiten zirudien eta defektuzko mezua agertzen zen mapa agertu beharko litzatekeen tokian. Hau konpontzeko botoi bat jarri da zeinek Google maps (aplikaziotik edo nabigatzailetik) irekitzen duen eta beraz, tokira arteko nabigazioa ahalbideratuz. Errorea non zegoen ere aurkitu zen: javascript kodean. Sistemak tokien kokapena gordetzeko string luze bat gordetzen du eta batzuetan Djangok zenbaki hamartarrak bereizteko '.' karaktereaz egiten zuen eta beste batzuetan ',' karaktereaz. Hau konpontzeko, txantiloietan Djangok ahalbideratzen duen 'built in tag' erako etiketa erabilia konpondu da, latitudea eta longitudea filtratuz txantiloian bertan.

Facebook bidezko loginean ordea arazo handiagoak egon dira. Honekin ere aplikazio berri bat sortu behar izan da Facebook developers-en eta emandako kodeak behar bezala kokatu administrazioko ezarpenetan. Baina logina egiteko orduan modu ezberdinetan erantzuten zuen nabigatzailean Facebook-en sartuta egonda edo sartu gabe egonda. Arazo hau konpondu ezinik, erabaki da nahiz eta lokalean funtzionalitate honek ondo funtzionatu, interneten Facebook bidezko identifikazioa ez baimentzea. Hortaz, sisteman erregistratzeko aukera bakarra egongo da eta modu hori erregistro arrunta izango da.

Komunikazio segurua mantentzeko HTTPS jarri nahi izan da *Let's Encrypt*<sup>21</sup> erabiliz baina honek Apache hartzen zuen helburu puntu gisa eta ezin izan da konfigurazioa ondo burutu. Atal hau burutzeko Djangoko ikasketa ordu gehiago egon beharko lirateke eta beraz, HTTP soilik ahalbideratu da.

Erabiltzailearen kokapena hartzeko orduan ere, nabigatzaile batzuetan errorea ematen du. Badirudi programan erabiltzen den kokapena hartzeko funtzioa zaharkituta dagoela eta eguneratu beharra dagoela.

---

21 <https://letsencrypt.org/>

# 8

---

---

## Produktuaren hedapen puntuak

Lortu den produktua funtzionala da eta bere betebeharra burutzen du. Baina beti dago zertan hobetu eta atal honetan azalduko dira produktuan egin daitezkeen zenbait hedapen puntu erabiltzaileei erraztasunak emateko eta baita sistema jakintsuagoa egiteko ere. Hedapen puntu asko webgunea martxan jartzen direnean agertzen dira, erabiltzaileek nola erabiltzen duten orrialdea ikusita, baina batzuk aurretik ere jakin daitezke.

Interfazeari dagokionez, hainbat hobekuntza egin daitezke:

- Bilaketa kutxan idatzi bitartean emaitzak agertzen joatea. Webgune askotan egiten den bilaketa modua da hau, erabiltzailea bilaketa kutxan idazten joan ahala, emaitzak agertzen joatea. Hau hedapen puntu gisa uzten da erabiltzaileen konfort-a hobetzeko hainbat gauza egin daitezkeelako baina gauza garrantzitsuagoak egin behar izan direlako. Adibidez, bilaketa kutxan 'Le' idatzita, sistemak 'Lekeitio', 'Lezama', 'Leioa', 'Lemoa', 'Lemoiz'... aukerak erakustea.
- Toki, herri eta jaien informazio hedatua ematea. Produktua dagoen moduan, toki, herri eta jai bakoitzak argazki bakarra izan dezake eta beren informazioan deskribapena soilik sartu daiteke. Hedapen puntu gisa beraz, hauetako bakoitzean argazki bat baino gehiago sartzeari ahalbideratzea eta informazio hedatua ematen uztea da (beharren arabera).

- Tokia sartzeko interfazean hobekuntzak. Herria aukeratzeko herrien lista luzea agertzen da eta erosoagoa izango litzateke lehendabizi probintzia aukeratzea, eta gero probintzia horretako herria, horrela herrien filtro handia egingo litzateke. Interfaze hau horrela utzi da, Django-ren formularioekin eginda dagoelako (txantiloian bertan egon beharrean) eta hau nola egin ez jakiteagatik.
- Zerbitzuen (etiketen) hierarkiak sortu. Produktuan zerbitzu guztiek daukate pisu bera eta agian hobeto litzateke hauen hierarkia sortzea eta horrela bilaketak zehatzagoak egin al izango liriateke. Gainera, tokia sartzeko interfazean zerbitzu guztiak listan agertu beharrean etiketa garrantzitsuenak agertuko liriateke eta aukeratu ezker bere 'ume-etiketak'. Adibidez, 'kirola' etiketa gurasoaren barruan, 'korrika', 'eskiatu', 'surf'... egonda.
- Etiketak, toki eta jaien izenak hizkuntza anitzetan jarri. Etiketek iruditxo bat daukatenez ez zaio garrantzi handiegirik eman atal honi eta hedapen puntu gisa utziko da.
- Mugikorretarako aplikazioa egitea. Nahiz eta webgunea 'ondo' ikusten den pantaila txikietan, funtzionalitate asko ordenagailu bidez egiteko pentsatuak daude eta hauek ez liriateke mugikorretan egon beharko. Hedapen puntu gisa uzten da mugikorretarako aplikazioa egitea, denbora eta jakintza faltarengatik.
- Administrazio gunetik mezuak erantzuteko aukera egotea. Dagoen modura, administrazio guneko langileak mezuak bere helbide elektronikotik erantzun behar ditu. Hedapen puntu gisa uzten da mezuen elkartrukea sisteman bertan egiteko aukera jartzea.
- Hizkuntza gehiago sartzeko aukera. Momentuz sistema 3 hizkuntzatan dago eskuragarri, hedapen puntu gisa uzten da besteren bat(zuk) sartzea.
- Orrialde guztia hizkuntza anitzetan egotea. Esan bezala txantiloiko informazioa eta deskribapenak bakarrik agertzen dira hizkuntza anitzetan baina, esaterako URL-ak beti daude euskaraz. Esaterako, toki baten informazioa ikusteko URL-a ".../tokia/8" da edozein hizkuntza aukeratuta dagoela ere. Ba hizkuntzaren araberrako URL-ak ere egin daitezke Djanganon esaterako ingelesez ".../place/8" agertzea, horrela hizkuntza finkatuta geratuko litzateke erabiltzailea sartzan den lehen momentutik.

### Sistemaren adimenari dagokionez:

- Sistemaren erabiltzaileek sortutako datu kuantitatiboak analizatzen ditu: balorazioa ematean 1-5 bitarteko izartxoak baliatuz edo gustuko/ez gustukoa dela (0,1) adieraziz. Hedapen puntu gisa egongo litzateke erabiltzaileek idatzitako iritziaren testutik informazioa eskuratzea. Iritziak hizkuntza ezberdinetan egon daitezkeenez eginkizun hau zailagoa bihurtzen da.
- Hizkuntzaren prozesamenduarekin jarraituz, erabiltzaileei profilean bere Twitter-eko kontua zein den adierazten utziz, beraiek idatzitakoa prozesatzean datza beste hedapen puntu bat. Python-ek Twitter-eko APIa era errazean erabiltzea ahalbideratzen du, eta erabiltzaileak idatzitakotik agian toki bat gustuko izan duten edo ez jakin al izango da.
- Erabiltzaileen tokien historiari patroi bat sortzea. Tokietan tokiak gomendatzen diren bezala jendearen iritzia kontuan edukiz, erabiltzaileek tokiak ikusteko jarraitu duten patroi bat sortu eta patroi hori jarraitzen hasten diren erabiltzaileei lista horretako tokiak gomendatzean datza hedapen puntu hau.
- Toki berri bat sartzen den momentuan, toki hori zein erabiltzailek gustuko izango duen erabakitzen duen programa bat exekutatzea eta gustuko izango duten erabiltzaileei helbide elektronikoko bidez jakinaraztea.

# 9

---

---

## Jarraipen eta kontrola

Jarraipen eta kontrolean egindakoak ikusiko dira hemen, proiektua plangintzan ezarritakoarekin bat egin den edota zer aldaketa egon diren ikusteko.

### **9.1- Proiektuaren garapena eta desbideraketak**

---

Proiektua aurrera eramateko plangintzan ezarritako datak eta orduak errespetatzen saiatu da, baina desbideraketa batzuk egon dira, besteak beste planifikatu gabeko gertakari bat agertu zelako 2016ko martxoan.

Plangintzan aurreikusi gabeko gertakari bat agertu zen, aukera bat, 2016ko urtarrilean 'gradu-amaierako lanen proiektuak sustatzeko deialdia' jarri zuten martxan DSS2016koek, eta hemen, kulturarekin zer ikusia zuten gradu bukaerako lanak aurkezteko aukera zegoen. Proiektu honek turismoa sustatu dezakeenez, aurkeztea erabaki zen. Honen ondorioetariko bat, garapena alde batera uztea eta aurkezpen dokumentu bat idatzi beharra izan zen (ordu hauek proiektutik kanpo). Eranskinetako E-6 atalean ikus daiteke DSS2016koak aukeratutako proiektuak deialdian parte hartzeko.

Lehenengo egunetan planifikatu bezala, plangintzan jardun zen eta honekin batera, produktuaren analisi eta diseinuarekin hasi zen. Atal hauetan egon den desbideraketa nagusia, analisisian eta diseinuan izan da. Proiektua aurrera joan den bitartean datu

baseko eredu batzuk aldatzea erabaki zelako, baina honek ez du proiektuaren atzerapenik ekarri ordu kopuru maximora iritsi ez zelako analisi eta diseinuan.

Behin planifikazioa eginda, Djangoren ikasketarekin hasi eta hau egin bitartean, eramangarrian lan ingurunea ere instalatu zen probak egin ahal izateko. Django tutorialak jarraitu bitartean, produktua martxan jartzen joan zen. Produktuaren itxura nolakoa izango zen bisualizatzeko, orrialde nagusietako txantiloiak prestatu ziren benetako inplementazioarekin (datu eta programekin) lanean hasi baino lehen.

Material bilaketan arazo handiegirik ez zen egon, atal honetan zuzendariak berak asko lagundu zuen eta beraz, hemen ere denbora aurreztu zen.

Garapenari begira, hasiera batean Django oinarrizko txantiloiak egin ziren, gero datu basean datu batzuk jarriz (datu meatzaritza gabe). Djangoren ikasketa 10 ordutara murriztuz (planifikatutakoa) egin zen garapena. Gero, aplikazioari datu meatzaritzako teknikak gainjarri zitzaizkion. Hauek planifikatutako orduekin bat egin dute, nahiz eta batzuetan datu meatzaritza edo Django lan egiten egon den bereiztea zaila izan. Zeren datu meatzaritzako emaitzak aplikazioan kokatu beharrak zeuden. Hauen bien dedikazioen batura bat etorri da planifikatutakoarekin.

Proben kasuan, ezin izan dira proba unitarioak egin Djangok eskaintzen duen modua eta egindako aplikazioa bat ez datozelako eta horregatik eskuzko probak egin ziren emaitzak norberak egiaztatuz. Horregatik probek planifikatutakoa baino denbora gutxiago eraman zuten.

Azkenik memoriari begira, planifikazioan esan bezala atazaren bat garatzen zenean Django, bere esplikazio txiki bat idazten zen dokumentu batean. Beraz, planifikatutako ordu kopurua betetzea lortu da.

Zuzendariarekin bilerak egin dira aldiro-aldiro. Bilera hauen isla dago E.5 eranskinean. Bilerak helbide elektroniko bidez hitzartzen izan dira sarritan, edo bileran bertan adosten zen hurrengo bilerako data.

Planifikatutakoarekiko desbideraketak 9.1 taulan:

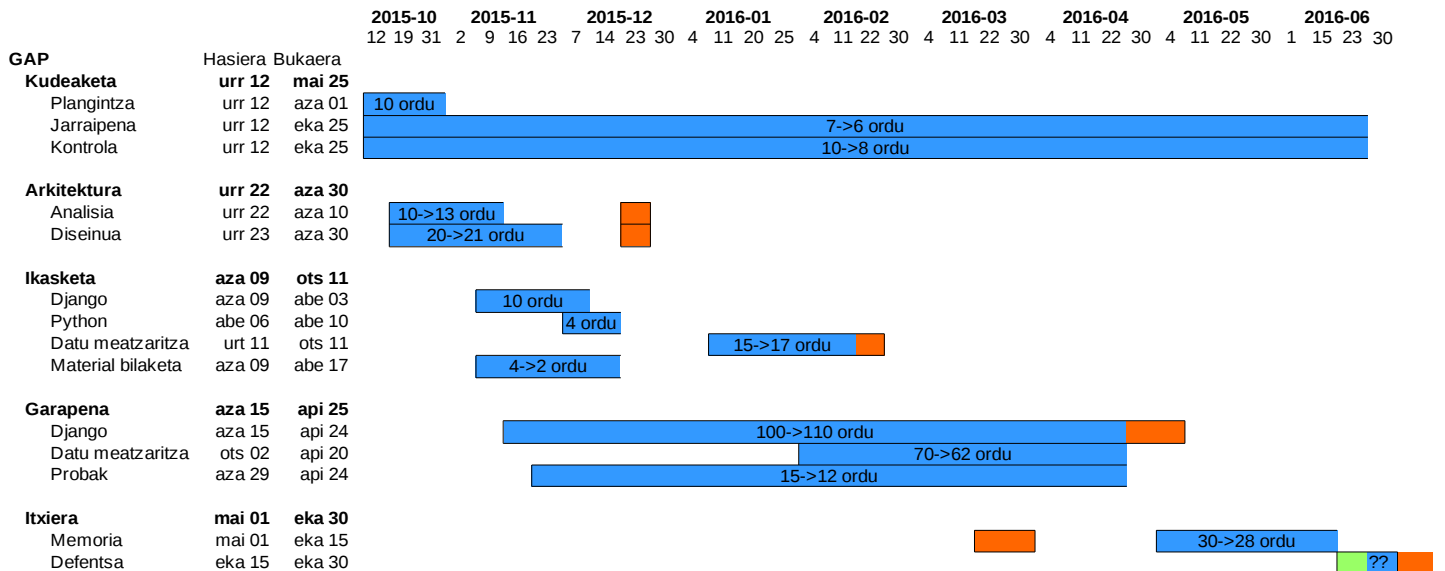


Kodea	Ataza	Estimazioa	Erreala	Desbideratzea	
1	Kudeaketa	27 ordu	24 ordu	-3:00	-%11,1
1.1	Plangintza	10 ordu	10 ordu	0:00	%0
1.2	Jarraipena	7 ordu	6 ordu	-1:00	-%14,2
1.3	Kontrola	10 ordu	8 ordu	-2:00	-%20
	-Kalitate kontrola -Arriskuen kontrola -Bukaera ziurtatu	--5 ordu --2 ordu --3 ordu			
2	Arkitektura	30 ordu	34 ordu	+4:00	+%13,3
2.1	Analisia	10 ordu	13 ordu	+3:00	+%30
2.2	Diseinua	20 ordu	21 ordu	+1:00	+%5
3	Ikasketa	33 ordu	33 ordu	0:00	%0
3.1	Django	10 ordu	10 ordu	0:00	%0
3.2	Python	4 ordu	4 ordu	0:00	%0
3.3	Datu meatzaritza	15 ordu	17 ordu	+2:00	+%13,3
3.4	Material bilaketa 1	4 ordu	2 ordu	-2:00	-%50
	-Djangorekin zerikusia duen materiala -Datu meatzaritzarekin zerikusia duen materiala				
4	Garapena	185 ordu	184 ordu	-1:00	-%0,54
4.1	Django	100 ordu	110 ordu	10:00	+%10
4.2	Datu meatzaritza	70 ordu	62 ordu	-8:00	-%11,4
4.3	Probak	15 ordu	12 ordu	-3:00	-%20
5	Itxiera	35 ordu	????	????	????
5.1	Memoria	30 ordu	28 ordu	-2:00	-%6,6
5.2	Defentsaren prestaketa	5 ordu	egiteke		

**9.1 Taula** - Desbideraketa taula

Beraz, orokorrean desbideraketa txikiak egon dira nahiz eta ehunekotan handiak iruditu, ordu gutxiko aldaketak izan dira eta.

Plangintzako Gantt diagrama eta errealitatean gertatu direnak ikusteko, ikusi 9.1 irudia.



### 9.1 irudia - Plangintzako Gantt eta errealitateko Gantt

9.1 irudiko Gantt diagraman agertzen diren denbora tarte urdinak, plangintzan ezarritakoak dira, laranja kolorekoak berriz, plangintzatik kanpo egindako lan tarteak eta berdea, planifikatutako lan tarteak baina bete ez dena. Aurrerako aipatu bezala, analisi eta diseinuan aldaketak egon ziren behin produktua egiten hasita (batez ere datu basean) eta garapenaren birfinketak luzatu egin ziren maiatzaren hasierara arte. Memorian planifikatutakoa baino askoz arinago lan egin dela ikusten da. Hori DSS2016ko dokumentua egitearen orduak dira, dokumentu honetan aurkezpena, aurrekariak eta teknologiak azaldu ziren edonork ulertzeko moduan (teknizismo gehiegirik gabe). Defentsaren prestaketa atzeratu egin da eta dokumentua entregatu ostean egingo da prestaketa, beraz ez da jarri zenbat ordu emango den defentsa prestatzen.

## 9.2- Kalitatearen jarraipena eta kontrola

Gutxieneko kalitatea bermatua geratzen dela ziurtatzeko, kontrol bat eraman da proiektuaren bizi zikloan. Kalitate adierazlean astero begiratzen izan dira (jarraipen eta kontrola, kode kopurua eta kode ulergarritasuna). Hauek eta plangintza kontuan hartuz gutxieneko kalitateira iritsiko dela ziurtatu behar zen.

### **-Irismenean zehaztutakoa betetzen da:**

Proiektuaren barne mugarriak errespetatu direnez, ez da arazorik egongo kanpo mugarriak betetzean. Produktuaren irismenari begira beteta geratzen da web orria lokalki funtzionala dela, erabiltzaileek beren iritzia eman dezaketela eta datuak analizatzen direla. Irismenaren 3 mailak betetzen dira. Gainera irismenean zehaztutako 4. maila ere egin al izan da produktua Internetera migratuz.

### **-Emangarri nagusiak epean emango dira:**

Aurreko puntuan esan bezala, barne mugarriak bete dira eta beraz, ez da arazorik egongo emangarrien epeak betetzean.

### **-Aplikazioa lokalki funtzionala da:**

Web aplikazioa lokalki funtzionala da eta hirugarrenen softwareak arazorik gabe funtzionatzen dute (Facebook eta Google maps besteak beste).

### **-Planifikatutako estimazioak errespetatzen dira:**

Desbideraketa taulan ikusi den bezala, zenbait desbideraketa egon dira planifikazioan baina honek ez du produktuaren lan karga handitu ezta entrega epeak atzeratu ere. Beraz ez da kontrolik hartu beharrik izan.

### **-Erabiltzaileek web orrian sortzen dituzten datuak prozesatzen dira:**

Datu meatzaritza script batzuen bitartez exekutatzen da aplikazioan. Hauek, datu basean dauden datuak hartu eta ezagumendua ateratzen dute hauetatik, gero aplikazioan pantailaratu dadin. Atal honetan hainbat hedapen puntu dauden arren, gure kalitatea bermatua geratzen da.

### **-Kodearen ulergarritasuna:**

Kodea Pythonez idatzia dagoenez, tabulazioak derrigorrez ondo egon behar dute eta honek kodearen ulergarritasunean asko laguntzen du. Gainera funtzioen izenak auto-deskribatzaileak izatea saiatu da behar izan denean komentarioak idatziz.

Kalitate gehigarri bezala web aplikazioa Internetera migratzea erabaki da, baina hau

egitean funtzionalitate batzuk kendu behar izan zaizkio sortutako zenbait arazo direla medio (ikus gertatutako arazoak 7.2 atalean). Baina oinarrizko funtzionalitateak betetzen direnez aplikazioak ondo funtzionatu dezake interneten ere.

### **9.3- Arriskuen jarraipen eta kontrola**

---

Hainbat arrisku aurreikusi ziren plangintzan eta horien azterketa egingo dugu lehendabizi.

#### **-Plangintza zehatz mehatz ez jarraitzea:**

Ikusi dugun bezala plangintzako orduak zehatz mehatz ez dira bete desbideraketa txikiak egon dira eta. Baina atal honi begira plangintzan idatzi ziren ondorioak mugarren datak ez betetzea izan zen, eta gure kasuan bete dira. Beraz atal honetan ez da arazorik egon. Gainera, plangintzan barne mugarren atzetik denbora margen bat utzi zen eta produktuaren kasuan birfinketa batzuk egiteko margen hau erabili izan da.

#### **-Lan ingurunea-teknologiek pentsa baino ikasketa gehiago exijitzea:**

Ikusi da ikasketan ordu guztiak aprobetxatu direla. Izan ere, ikasketa prozesua luzeagoa ere izan litekeen baina ikasitakoarekin gure produktua aurrera eramatea posible izan da. Egia da, agian Djangoren filosofia %100ean ez dela errespetatu baina ikasitakoa aplikatu da. Hauxe izan da hartu beharreko kontrola, ikasitakoarekin aurrera egitea.

Datu meatzaritzaren kasuan ikasketa ezberdina izan da, material fisikoa irakurriz izan delako. Gainera zenbait gauza ikasteko datu meatzaritzako irakasleari ere galdetu izan zaio.

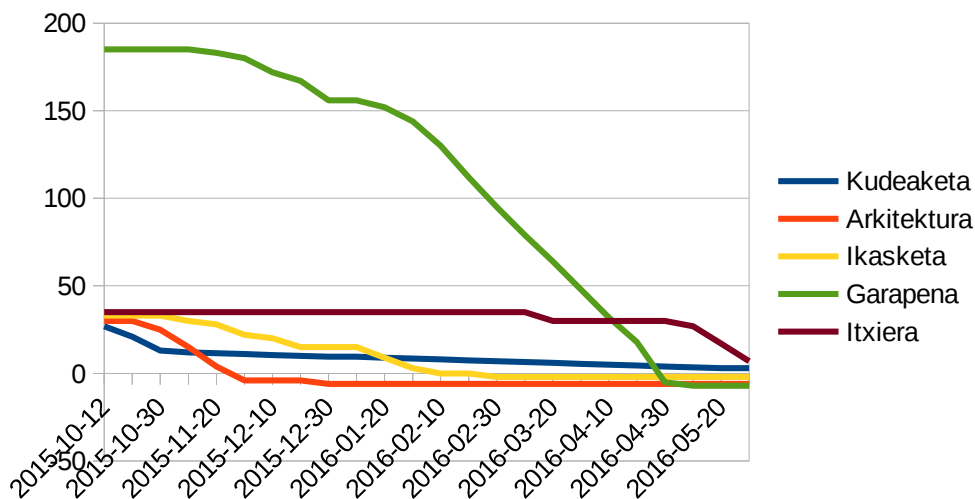
#### **-Egindako lana 'galtzea':**

Arriskuen artean hauxe izan da gehien kontrolatu dena. Lan egun bakoitzean egiten izan dira segurtasun kopiak eta egindako lana dokumentu batean idatzi. Beraz arrisku honetan ez da kontrolik hartu behar izan.

### -Aplikazioaren garapenean arazoak:

Hainbat arazo egon dira aplikazioa garatzean eta erabakiak hartu behar izan dira. Esaterako Facebook-eko logina gure makinatik egitea posible izan da baina Internetera migratzean hainbat arazo izan direnez, funtzionalitate hau kentzea erabaki da. Gainera denbora faltarengatik produktuan hainbat hedapen puntu geratu dira Django teknologiarekin zerikusia dutenak (besteak beste, URL-en itzulpena).

Planifikatutako gabeko gertakari bat esan bezala, DSS2016ko deialdia izan zen eta proiektu hau bertan aurkeztean, hauek ere interesatu bilakatu ziren.



### 9.2 Irudia - Lan orduen eboluzioa

Lan gehien egin diren egunekoak:

- 2016-02-10 eta 2016-02-20 bitartean (18,5 ordu)
- 2016-04-10 eta 2016-04-30 bitartean (18,5 ordu)

Lan gutxien egin diren egunak:

- 2015-12-30 eta 2016-01-10 bitartean (0 ordu)

Lan gehien eta gutxien egin den hilabeteak:

- Otsaila (53,5 ordu)
- Urtarrila (25 ordu)

# 10

---

---

## Ondorioak

Graduan zehar egin behar izan den proiekturik handiena izan da hau, bai lan orduetan eta baina autonomian ere. Ikasleak bere lan orduak hainbat hiletan banatu behar izan ditu planifikazio batean eta gero, planifikazio hori bete zuzendariaren kontrolpean baina autonomi handiarekin.

Produktuari erreparatuz, nahi izan diren gutxieneko funtzionalitateak baditu eta produktuaren aurkezpenean adierazitako arazoa konpon dezake, nahiz eta hedapen puntuak izan oraindik. Gutxienez urte betez martxan egongo den produktua da, domeinua iraungi arte edo makina birtuala mantentzeak kostuak ekarri arte. Hedapen puntuetan aipatutako hainbat hobekuntza gehitu ahal dira sisteman, haietan aipagarrienak testuaren prozesamendua eta erabiltzailearen esperientzia hobetzen duten detailetxoak izanik. Mugikorretarako aplikazioa egitearen arazoa Djangoren negozio logika eta mugikorretako programazioak nola bat egin izango litzateke.

Proiektua garatu bitartean idatzitako dokumentuak asko lagundu dute memoria idazterako orduan. Proiektuaren hasieratik hainbat hilabete igaro dira eta beraz, hasieran egindakoen esplikazio eta irudirik gabe, ezinezkoa izango litzateke proiektuan egindako bidea islatzea memorian.

Proiektuari erreparatuz ordea, eskertzekoa izan da hasiera-hasieratik martxan jarri izana, bukaeran lan karga gehiegi eduki ez izateko eta atzerapenik ez egoteko. Eskertzekoa da ere tutore baten laguntza, ikasleak mota honetako proiektuetan esperientziarik ez daukalako eta bere bidean nola doan eta zer egin behar duen gidatzen laguntzeko. Aprobetxatuz, eskerrak Jose Ramon Zubizarretari. Faltan bota izan dena gradu bukaerako proiektuei buruzko aurre-informazioa izan da, hau da, ikaslea bera informatu eta mugitu behar da GAP-ari buruzko informazioa eskuratu nahi badu.

## Bibliografia

- Django [<https://www.djangoproject.com/>]
- Django tutorial [<https://docs.djangoproject.com/en/1.8/intro/>]
- Django tutorial [[youtube CodingEntrepreneus](#)]
- Crispy forms [<http://django-crispy-forms.readthedocs.io>]
- Allauth [<https://readthedocs.org/projects/django-allauth/>]
- Defender [<https://github.com/kencochrane/django-defender>]
- Extensions [<https://github.com/django-extensions/django-extensions>]
- Stackoverflow [<http://stackoverflow.com/>]
- PyMongo [<https://api.mongodb.com/python/current/>]
- PyCharm [<https://www.jetbrains.com/pycharm/>]
- MongoDB [<https://www.mongodb.com/>]
- Amazon Web Services [<https://aws.amazon.com/>]
- Redis [<http://redis.io/>]
- Xampp [<https://www.apachefriends.org/index.html>]
- Bootstrap [<http://getbootstrap.com/>]
- Awesome fonts [<http://fontawesome.io/icons/>]
- Facebook Developers [<https://developers.facebook.com/>]
- Google Developers [<https://console.developers.google.com/>]
- Schall, D. (2015), *Social Network-Based Recommender System*, USA, Springer International Publishing.
- Russell, M. (2014), *Mining the Social Web 2nd Edition*, USA, O'Reilly Media, Inc.

O'Higgins, N. (2011), *MongoDB and Python – Patterns and processes*, USA, O'Reilly Media, Inc.

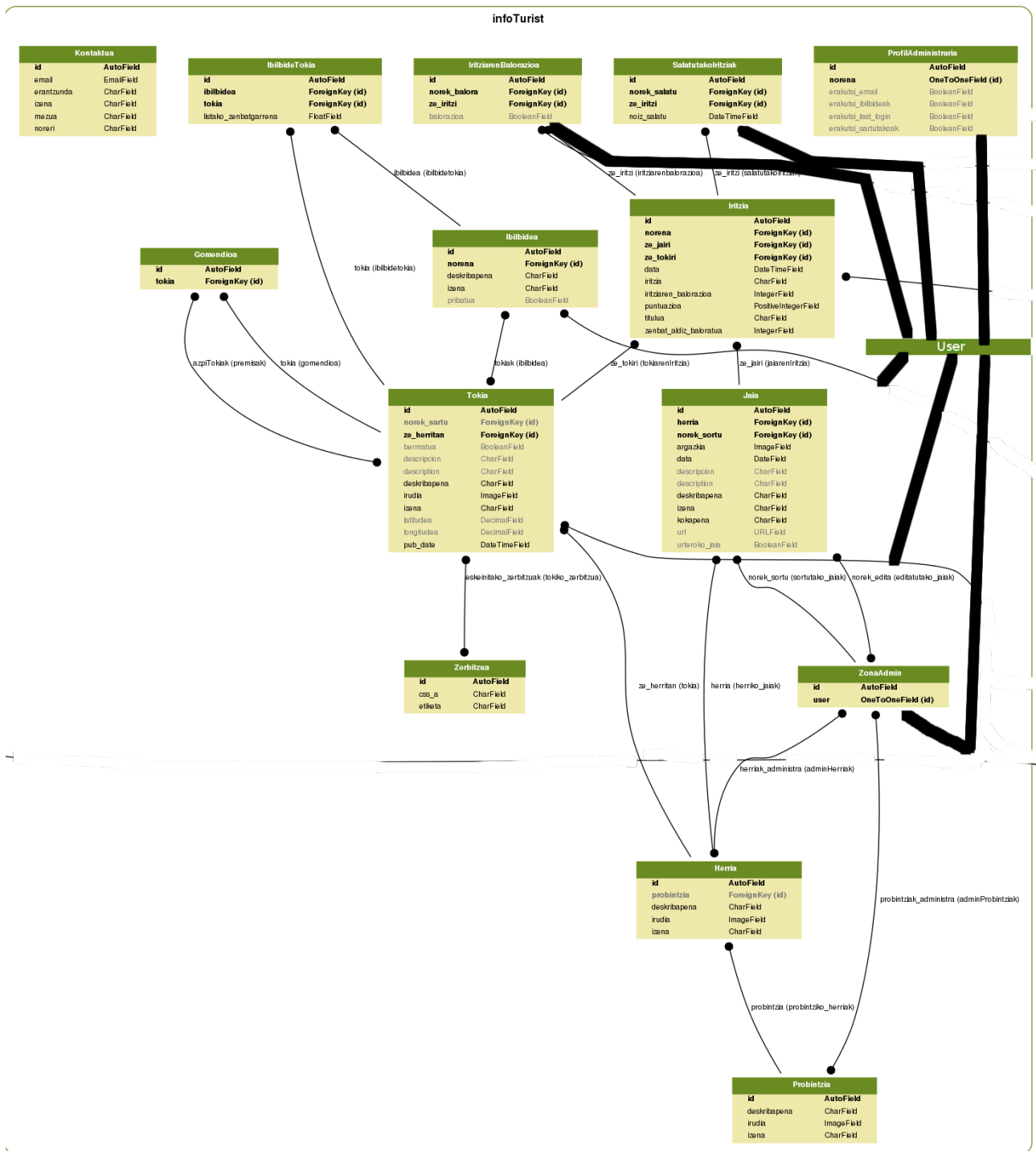


## Eranskinak

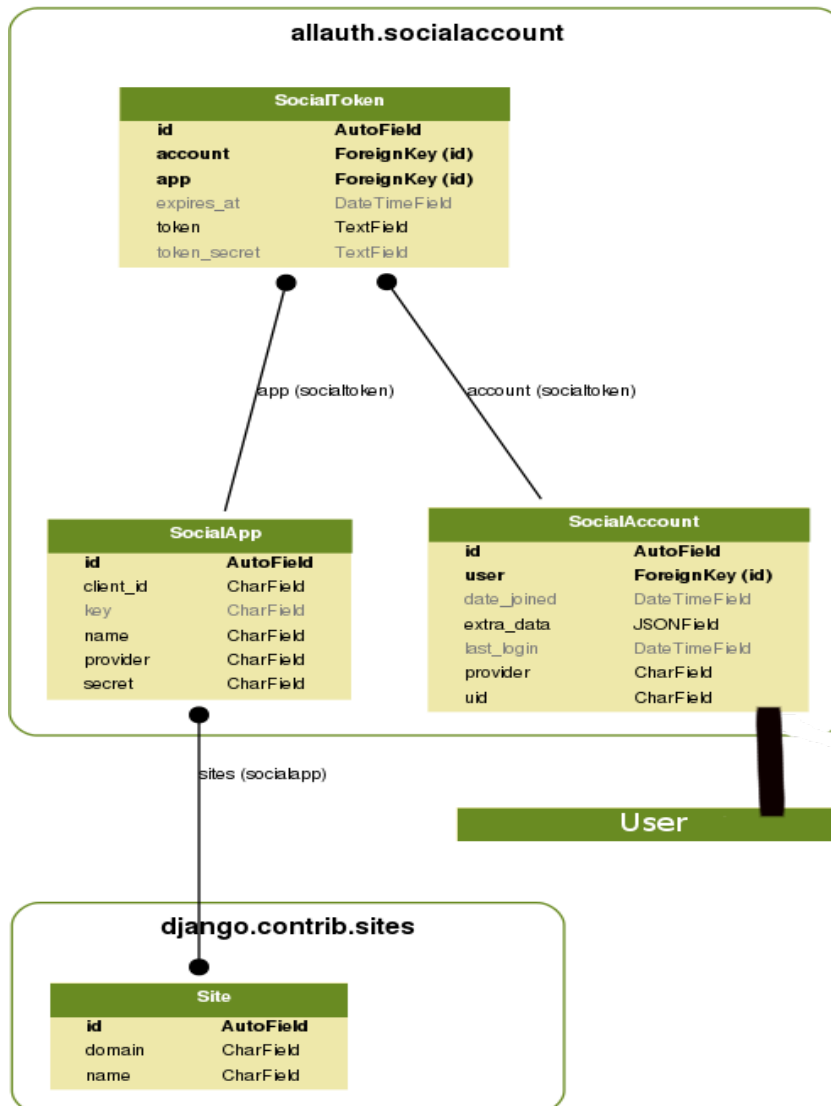
### **E.1- Automatikoki Django-extensionen bidez sortutako ereduaren informazioa**

---

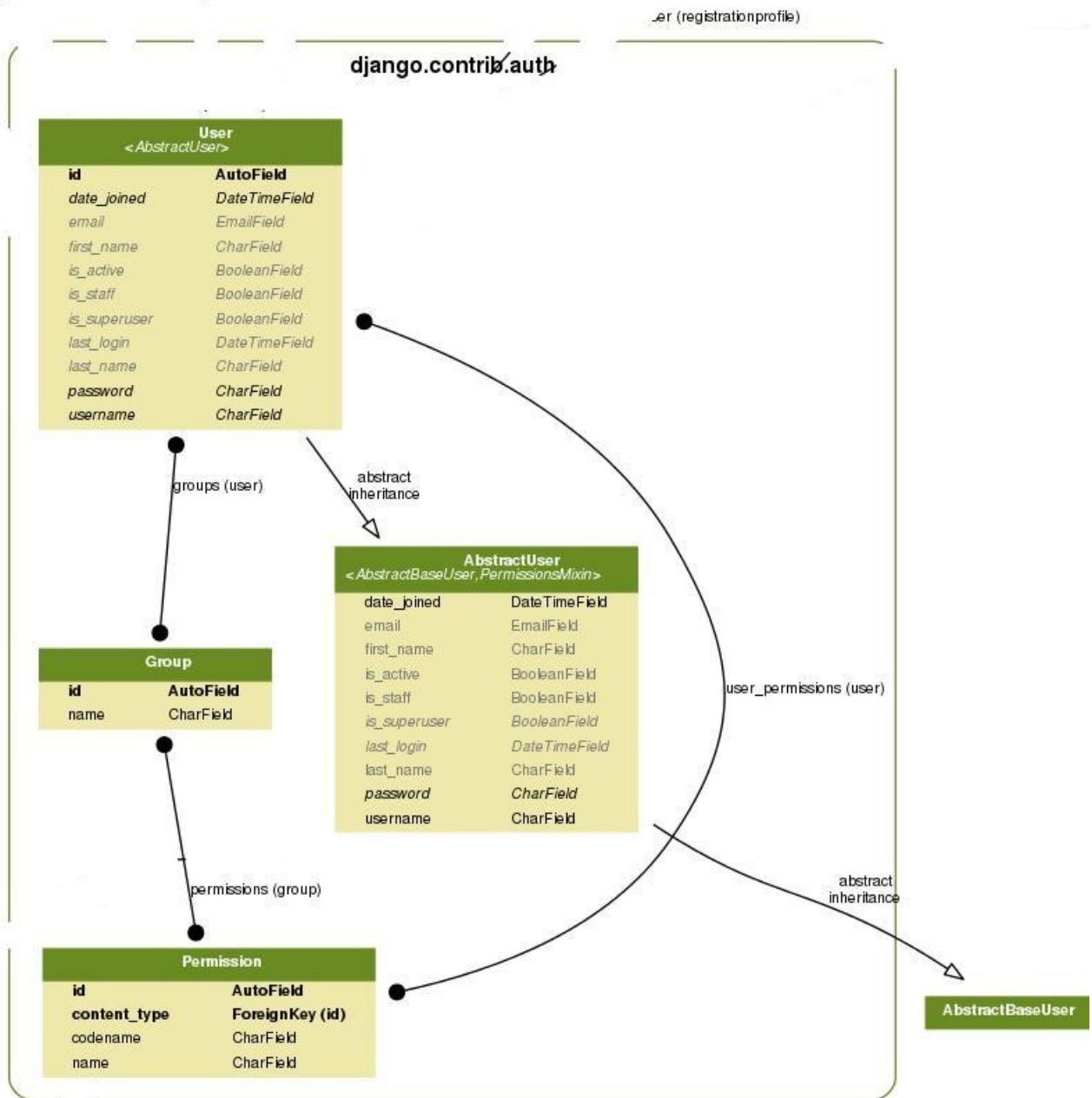
Djangorekin estentsio bat erabili daiteke zeinek aplikazioko eredu guztien diagrama sortzen duen. Baina diagrama hau irudi handi batean bueltatzen du eta horregatik adieraziko dira azpi irudietan. Irudien arteko loturak sinplifikatu egin dira ereduaren izena irudian sartuz eta loturak bertara eramanez.



**Irudia E1 - Aplikazioaren ereduak eta barne loturak**



**Irudia E2** - Aplikazioan sare sozialen bidezko logina egiteko ereduak



**Irudia E3** - Django.contrib-eko User eredua lotura gabe

## E.2- Aplikazioko formulario batzuk

---

Djangoren txantiloietan agertuko diren formularioak forms.py izeneko fitxategian erazagutu daitezke. Honen abantaila nagusia berrerabilgarritasun eta segurtasuna dira baina formulario eroso eta itxurosoak sortu nahi badira zenbait widget erabili behar dira. Aplikazio guztiko formularioak ez dira era honetara sortuak izan.

```
class TokiaForm(forms.ModelForm):
    class Meta:
        model = Tokia
        fields = ['ze_herritan', 'izena', 'deskribapena', 'descripcion','description',
'latitudea', 'longitudea', 'irudia', 'eskeinitako_zerbitzuak']
        widgets = {
            'deskribapena': Textarea(attrs={'cols': 80, 'rows': 15, 'placeholder':
'Euskaraz...'}),
            'descripcion': Textarea(attrs={'cols': 80, 'rows': 15, 'placeholder': 'En
castellano...'}),
            'description': Textarea(attrs={'cols': 80, 'rows': 15, 'placeholder': 'In
English...'}),
            'eskeinitako_zerbitzuak': forms.CheckboxSelectMultiple,
        }
    def clean_latitudea(self):
        latitudea = self.cleaned_data.get('latitudea')
        #email_base, provider = email.split("@") #Talalala, myweb.com
        if not -90 <= float(latitudea) <= 90: #Latitudea bere limiteetatik kanpo sartu
bada
            raise forms.ValidationError("Mesedez, sartu latitude egoki bat. [-90, 90]")
        return latitudea
    def clean_longitudea(self):
        longitudea = self.cleaned_data.get('longitudea')
        if not -180 <= float(longitudea) <= 180: #Latitudea bere limiteetatik kanpo
sartu bada
            raise forms.ValidationError("Mesedez, sartu longitude egoki bat. [-180,
180]")
        return longitudea
    def clean_irudia(self):
        image = self.cleaned_data.get('irudia',False)
        if image:
            if image._size > 3*1024*1024:
                raise forms.ValidationError("Image file too large ( > 3mb )")
            return image
        else:
            raise forms.ValidationError("Ezin izan da irudia irakurri")
class JaiaForm(forms.ModelForm):
    def __init__(self, request, *args, **kwargs):
        super(JaiaForm, self).__init__(*args, **kwargs)
        zonaAdmin = request.user.zonaAdmin
```

```

        if zonaAdmin.probintziak_administra:
            self.fields["herria"].queryset =
Herria.objects.filter(probintzia__in=zonaAdmin.probintziak_administra.all())
        elif zonaAdmin.herriak_administra:
            self.fields["herria"].queryset =
Herria.objects.filter(adminHerriak__user=request.user)
    class Meta:
        model = Jaia
        fields = ['izena', 'kokapena', 'argazkia', 'deskribapena', 'descripcion','description',
'url', 'data', 'herria', 'urteroko_jaia']
        widgets = {
            'data': TextInput(attrs={'placeholder': '2016-02-28'}),
            'deskribapena': Textarea(attrs={'cols': 80, 'rows': 15, 'placeholder':
'Euskaraz...'}),
            'descripcion': Textarea(attrs={'cols': 80, 'rows': 15, 'placeholder': 'En
castellano...'}),
            'description': Textarea(attrs={'cols': 80, 'rows': 15, 'placeholder': 'In
English...'}),
        }
    def clean_data(self):
data = self.cleaned_data.get('data',False)
    if not data > datetime.now().date():
        raise forms.ValidationError("Etorkizuneko data bat izan behar du")
    else:
        return data

GIPUZKOA = 'Gipuzkoa'
BIZKAIA = 'Bizkaia'
ARABA = 'Araba'
ADMIN = 'Web-Admin'
norentzako_aukerak = (
    (GIPUZKOA, 'Gipuzkoa'),
    (BIZKAIA, 'Bizkaia'),
    (ARABA, 'Araba'),
    (ADMIN, 'Web-Admin')
)
class ContactForm(forms.Form):
    full_name = forms.CharField(required=False)
    email = forms.EmailField()
    message = forms.CharField()
    #norentzako = forms.MultipleChoiceField(required=False,
widget=forms.CheckboxSelectMultiple, choices=norentzako_aukerak)
    norentzako = forms.ChoiceField(required=False, choices=norentzako_aukerak)
    def clean_email(self):
        email = self.cleaned_data.get('email')
        email_base, provider = email.split('@')
        domain, extension = provider.split('.')
        if domain == "yopmail":
            raise forms.ValidationError("Email hori ezin da sartu")
        return email

```

```
class IbilbideaSortuForm(forms.ModelForm):
    class Meta:
        model = Ibilbidea
        fields = ['izena', 'deskribapena', 'pribatua']
        widgets = {
            'deskribapena': Textarea(attrs={'cols': 40, 'rows': 5}),
        }
class ProfilAdminForm(forms.ModelForm):
    class Meta:
        model = ProfilAdministraria
        fields = ['erakutsi_email', 'erakutsi_sartutakoak', 'erakutsi_last_login',
'erakutsi_ibilbideak']
```

### E.3- Ereduen erazagupena

---

Aplikazioko models.py fitxategian erazagutu behar dira datu baseko ereduak beren loturekin. Eredu horiek zenbat meta-informazio izan ditzakete, besteak beste, pluraleko izena eta renderizatzean agertuko den testua. Jarraian adierazten da fitxategi horretako edukia.

```
class ZonaAdmin(models.Model):
    user = models.OneToOneField(User, null=True, related_name='zonaAdmin')
    herriak_administra = models.ManyToManyField('Herria',
related_name='adminHerriak', blank=True)
    probintziak_administra = models.ManyToManyField('Probintzia',
related_name='adminProbintziak', blank=True)
    def __str__(self): # __unicode__ on Python 2
        return self.user.username
    def herriak_list(self):
        return ', '.join([a.izena for a in self.herriak_administra.all()])
    def probintziak_list(self):
        return ', '.join([a.izena for a in self.probintziak_administra.all()])
    class Meta:
        verbose_name_plural = "ZonaAdminak"

class Probintzia(models.Model):
    izena = models.CharField(max_length=20)
    irudia = models.ImageField(upload_to='probintzienIrudiak', default="")
#media_root barruan daon karpeta
    deskribapena = models.CharField(max_length=1000, default="")
    def __str__(self): # __unicode__ on Python 2
        return self.izena
    class Meta:
        verbose_name = _('Probintzia')
        verbose_name_plural = _("Probintziak")

class Herria(models.Model):
    izena = models.CharField(max_length=30)
```

```

    deskribapena = models.CharField(max_length=1000)
    irudia = models.ImageField(upload_to='herrienIrudiak', default="")
#media_root barruan daon karpeta
    probintzia = models.ForeignKey(Probintzia,
related_name='probintziko_herriak', blank=True, null=True)
    def __str__(self): # __unicode__ on Python 2
        return self.izena
    class Meta:
        verbose_name = _('Herria')
        verbose_name_plural = _("Herriak")
        ordering = ['izena']

class Tokia(models.Model):
    izena = models.CharField(max_length=200)
    deskribapena = models.CharField(max_length=3000)
    descripcion = models.CharField(max_length=3000, null=True, blank=True)
    description = models.CharField(max_length=3000, null=True, blank=True)
    pub_date = models.DateTimeField('date published',
default=datetime.date.today)
    latitudea = models.DecimalField(decimal_places=25, max_digits=28,
blank=True, null=True)
    longitudea = models.DecimalField(decimal_places=25,
max_digits=28,blank=True, null=True)
    irudia = models.ImageField(upload_to='tokienIrudiak', default="")
#media_root barruan daon karpeta
    bermatua = models.BooleanField(default=False) #ZonaAdministrariiek eman
dezakete bermea tokiei.
    norek_sartu = models.ForeignKey(User, null=True, blank=True,
related_name='sartuta')
    ze_herrian = models.ForeignKey(Herria, null=True)
    eskeinitako_zerbitzuak = models.ManyToManyField('Zerbitzua',
related_name='tokiko_zerbitzua')
    # Tokia klasearentzat agertuko den izena
    def __str__(self): # __unicode__ on Python 2
        return self.izena
    # Tokia-rentzat funtzio gehigarriak
    def was_published_recently(self):
        return self.pub_date >= timezone.now() - datetime.timedelta(days=1)
    class Meta:
        verbose_name_plural = _("Tokiak")

class Iritzia(models.Model):
    norena = models.ForeignKey(User, null=True)
    ze_tokiri = models.ForeignKey(Tokia, null=True,
related_name='tokiarenIritzia')
    ze_jairi = models.ForeignKey('Jaia', null=True,
related_name='jaiarenIritzia')
    puntuazioa = models.PositiveIntegerField(validators=[MinValueValidator(1),
MaxValueValidator(5)])
    titulua = models.CharField(max_length=30)
    iritzia = models.CharField(max_length=500)
    data = models.DateTimeField(default=datetime.date.today)
    zenbat_albiz_baloratua = models.IntegerField(null=True, default=0)
    iritziaren_balorazioa = models.IntegerField(null=True, default=0)
    class Meta:
        verbose_name_plural = _("Iritziak")
    def __str__(self): # __unicode__ on Python 2
        return str(self.id)+" - "+self.titulua+" - "+self.iritzia

```



```
def duela_gutxiko_iritzia(self): #Toki bakoitzari hilean behin bakarrik
    idatzi al zaio
    return self.data >= timezone.now() - datetime.timedelta(days=30)

class IritziarenBalorazioa(models.Model):
    norek_balora = models.ForeignKey(User, null=True)
    ze_iritzi = models.ForeignKey(Iritzia, null=True)
    balorazioa = models.BooleanField()
    class Meta:
        verbose_name_plural = "Iritziaren Balorazioak"

class SalatutakoIritziak(models.Model):
    norek_salatu = models.ForeignKey(User, null=True)
    ze_iritzi = models.ForeignKey(Iritzia, null=True)
    noiz_salatu = models.DateTimeField('Salaketa data')
    class Meta:
        verbose_name_plural = "Salatutako iritziak"

class Zerbitzua(models.Model):
    etiketa = models.CharField(max_length=20)
    css_a = models.CharField(max_length=300, default="")
    #irudia = models.ImageField(upload_to='zerbitzuak') #Irudiak daukoz:
    irudia.name, irudia.path, irudia.url
    def __str__(self): # __unicode__ on Python 2
        return self.etiketa
    class Meta:
        verbose_name_plural = _("Zerbitzuak")

class Jaia(models.Model):
    izena = models.CharField(max_length=100)
    kokapena = models.CharField(max_length=50, null=True)
    argazkia = models.ImageField(upload_to='jaiak')
    deskribapena = models.CharField(max_length=2000)
    descripcion = models.CharField(max_length=3000, null=True, blank=True)
    description = models.CharField(max_length=3000, null=True, blank=True)
    url = models.URLField(max_length=200, null=True, blank=True)
    data = models.DateField()
    herria = models.ForeignKey(Herria, related_name='herriko_jaiak')
    norek_sortu = models.ForeignKey(ZonaAdmin, related_name='sortutako_jaiak')
    norek_edita = models.ManyToManyField(ZonaAdmin,
related_name='editatutako_jaiak', blank=True)
    urteroko_jaia = models.BooleanField(default=True)
    class Meta:
        verbose_name_plural = _("Jaiak")
    def __str__(self): # __unicode__ on Python 2
        return self.izena

class ProfilAdministraria(models.Model):
    norena = models.OneToOneField(User, null=True, related_name='config')
    erakutsi_email = models.BooleanField(default=True)
    erakutsi_sartutakoak = models.BooleanField(default=True)
    erakutsi_last_login = models.BooleanField(default=True)
    erakutsi_ibilbideak = models.BooleanField(default=True)

class Ibilbidea(models.Model):
    norena = models.ForeignKey(User, null=True,
related_name='gordetako_ibilbideak')
    izena = models.CharField(max_length=100)
```

```
deskribapena = models.CharField(max_length=1000)
tokiak = models.ManyToManyField(Tokia, blank=True)
pribatua = models.BooleanField(default=False)
def tokiak_list(self):
    return ', '.join([a.izena for a in self.tokiak.all()])
class Meta:
    verbose_name_plural = _("Ibilbideak")
def __str__(self): # __unicode__ on Python 2
    return self.izena
class IbilbideTokia(models.Model):
    ibilbidea = models.ForeignKey(Ibilbidea, on_delete=models.CASCADE)
    tokia = models.ForeignKey(Tokia, on_delete=models.CASCADE)
    listako_zenbatgarrena = models.FloatField()
def __str__(self): # __unicode__ on Python 2
    return self.tokia.izena + " pos: " + str(self.listako_zenbatgarrena)
class Gomendioa(models.Model):
    tokia = models.ForeignKey(Tokia)
    azpiTokiak = models.ManyToManyField(Tokia, related_name='premisak')
class Kontaktua(models.Model):
    izena = models.CharField(max_length=50)
    email = models.EmailField()
    mezua = models.CharField(max_length=2000)
    norentzako_aukerak = (
        (GIPUZKOA, 'Gipuzkoa'),
        (BIZKAIA, 'Bizkaia'),
        (ARABA, 'Araba'),
        (ADMIN, 'Web-Admin')
    )
    noreri =
models.CharField(choices=norentzako_aukerak, default=ADMIN, max_length=20)
    erantzunda = models.CharField(max_length=10, default=False)
```

## E.4- Txantiloiak

---

Atal honetan txantiloia batzuen adibide interesgarrienak jarriko dira. Txantiloiak asko eta luzeak direnez, guztiak ez jartzea erabaki da.

### E.4.1- Tokia\_detail.html txantiloia

Txantiloia atal ezberdinetan zatitu da errazago ulertzeko. Atal bakoitzean zer dagoen adierazi ostean bere kodea agertuko da.

Txantiloiko javascript kodea:

```
<script type="text/javascript">
    function initMap() {
        var latitudea = {{ tokia.latitudea|stringformat:'g' }};
        var longitudea = {{ tokia.longitudea|stringformat:'g' }};
        var tokia = '{{ tokia.izena }}';
        var myLatLng = {lat: latitudea, lng: longitudea};
        // Create a map object and specify the DOM element for display.
        var map = new google.maps.Map(document.getElementById('map'), {
            center: myLatLng,
            scrollwheel: false,
            zoom: 16
        });
        // Create a marker and set its position.
        var marker = new google.maps.Marker({
            map: map,
            position: myLatLng,
            title: tokia
        });
    }
</script>
<script async defer
    src='https://maps.googleapis.com/maps/api/js?
key=AlzaSyCvmJTJ7pU8UUPI9s0IAp4IACKsNdTdQfM&callback=initMap'>
</script>
<script type="text/javascript">
    function izarGanean(div)
    {
        if(document.getElementById('puntuazioa').value == "")
        {
            var izarra = div.id.split("r"); //izar1, izar2,... olan izarren zenbakia jakin al dou
            for (i = 1; i <= izarra[1]; i++)
```

```

        {
            var divIzena = "izar".concat(i);
            document.getElementById(divIzena).className = "fa fa-star";
        }
    }
}
function izarKanpoan(div)
{
    if(document.getElementById('puntuazioa').value == "")
    {
        var izarra = div.id.split("r"); //izar1, izar2,... olan izarren zenbakia jakin al dou
        for (i = 1; i <= izarra[1]; i++)
        {
            var divIzena = "izar".concat(i);
            document.getElementById(divIzena).className = "fa fa-star-o";
        }
    }
}
function aukeratuzarra(div)
{
    var izarra = div.id.split("r"); //izar1, izar2,... olan izarren zenbakia jakin al dou
    for (i = 1; i <= 5; i++) //Guztiei hasierako tamaina jarri
    {
        var divIzena = "izar".concat(i);
        document.getElementById(divIzena).className = "fa fa-star-o";
    }
    for (i = 1; i <= izarra[1]; i++)
    {
        var divIzena = "izar".concat(i);
        document.getElementById(divIzena).className = "fa fa-star fa-lg";
    }
    document.getElementById('puntuazioa').value = izarra[1];
}
function bozkatu(iritzia_id, balorazioa) {
    console.log("Bozkatu is working!") // sanity check
    $.ajax({
        url : "/infoTurist/bozkatulirritzia", // the endpoint
        type : "POST", // http method
        data : { iritzia_id : iritzia_id, balorazioa: balorazioa }, // data sent with the
post
        // handle a successful response
        success : function(json) {
            //$('#post-text').val(""); // remove the value from the input
            console.log(json); // log the returned json to the console
            console.log("success"); // another sanity check
            alert(json);
        },
        // handle a non-successful response
        error : function(xhr,errmsg,err) {
            $('#results').html("<div class='alert-box alert radius' data-alert>Oops!
We have encountered an error: "+errmsg+
            " <a href='#' class='close'>&times;</a></div>"); // add the error

```

```

to the dom
        console.log(xhr.status + ": " + xhr.responseText); // provide a bit more
info about the error to the console
    }
    });
};
</script>
<script type="text/javascript"> //Gomendioak jasotzeko ajax bidez.
    tokia_id = {{ tokia.id }}; //Aldagai globalak (var gabe)
    zenbatTokijaso = 3;
    $(document).ready(
        function(){ //Prest ajaxeaz toki honentzako gomendatutako tokiak hartzeko.
            jasoTokikoGomendioak();
        });
    function jasoTokikoGomendioak()
    {
        console.log("Gomendioak jaso is working!") // sanity check
        $.ajax({
with the post
            // handle a successful response
            success : function(json) {
                console.log("success"); // another sanity check
                var HTMLerantzuna = "";
                json = JSON.parse(json);
                //alert(json[0].irudia);
                for(i=0; i < json.length; i++)
                {
                    HTMLerantzuna = HTMLerantzuna+ '<div class="col-md-3"
style="background:white;margin-left:20px;">';
                    var idea = parseInt(json[i].idea);
                    var url_mask = "{% url 'infoTurist:tokia_detail' 12345
%}".replace(/12345/, idea.toString());
                    HTMLerantzuna = HTMLerantzuna+ '<a
href="" + url_mask + ""><h4><b>' + json[i].izena + ' id:' + json[i].idea + '</b></h4></a>';
                    HTMLerantzuna = HTMLerantzuna+ '<img class="img-
responsive" src="" + json[i].irudia + "" alt="" + json[i].izena + ' irudia" style="max-width:350px;
max-height:200px;"></div>';
                }
                document.getElementById('interesekoTokiak').innerHTML =
HTMLerantzuna;
            },
            // handle a non-successful response
            error : function(xhr,errmsg,err) {
                $('#results').html("<div class='alert-box alert radius' data-
alert>Oops! We have encountered an error: "+errmsg+
                " <a href='#' class='close'>&times;</a></div>"); // add the
error to the dom
                console.log(xhr.status + ": " + xhr.responseText); // provide a bit
more info about the error to the console

```

```

    }
    });
};
</script>
<script type="text/javascript">
//IRITZIAK SALATU
function iritziaSalatu(iritzi_id)
{
    iritzia_id = iritzi_id;
    //window.scrollTo(0, 0);
    $('html, body').animate({scrollTop:0}, 'slow');
    document.getElementById("dialog").style.display='inline';
}
function baiSalatu()
{
    document.getElementById("dialog").style.display='none';
    iritziaSalatuDef();
}
function ezSalatu()
{
    document.getElementById("dialog").style.display='none';
}
function iritziaSalatuDef()
{
    console.log("Iritzia salatu is working!") // sanity check
    $.ajax({
        url : "{% url 'infoTurist:iritziaSalatu' %}", // the endpoint
        type : "POST", // http method
        data : { iritzia_id : iritzia_id }, // data sent with the post
        // handle a successful response
        success : function(json) {
            console.log("success"); // another sanity check
            //json = JSON.parse(json);
            alert(json);
        },
        // handle a non-successful response
        error : function(xhr,errmsg,err) {
            alert("Oops! We have encountered an error: "+errmsg+
                " <a href='#' class='close'>&times;</a></div>"); // add the
            // error to the dom
            console.log(xhr.status + ": " + xhr.responseText); // provide a bit
            // more info about the error to the console
        }
    });
};
function ibilbideanGehitu(tokia_id, ibilbideSelect_id)
{
    ibilbidea_id = document.getElementById("tokia"+ibilbideSelect_id).value;
    console.log("Ibilbidea is working!") // sanity check
    $.ajax({
        url : "{% url 'infoTurist:ibilbidean_gehitu' %}", // the endpoint

```

```
type : "POST", // http method
data : { tokia_id : tokia_id, ibilbidea_id: ibilbidea_id }, // data sent with the
post
// handle a successful response
success : function(json) {
    $('#post-text').val(""); // remove the value from the input
    //console.log(json); // log the returned json to the console
    console.log("success"); // another sanity check
    //alert(json);
    if (json == "OK")
    {
        var x = document.getElementById("tokia"+ibilbideSelect_id);
        x.remove(x.selectedIndex);
        alert("{% trans 'Tokia gehitu da ibilbidera' %}");
    }
},
// handle a non-successful response
error : function(xhr,errmsg,err) {
    $('#results').html("<div class='alert-box alert radius' data-alert>Oops!
We have encountered an error: "+errmsg+
" <a href='#' class='close'>&times;</a></div>"); // add the error
to the dom
    console.log(xhr.status + ": " + xhr.responseText); // provide a bit more
info about the error to the console
}
});
};
</script>
```

Tokiaren izena, herria eta bermatzeko botoia:

```
<h1 class="text-center">{{ izenburua }} <small>-<a href="{% url
'infoTurist:herria_detail' tokia.ze_herritan.id %}">{{ tokia.ze_herritan }}</a></small>
    {% if tokia.ze_herritan in request.user.zonaAdmin.herriak_administra.all or
tokia.ze_herritan.probintzia in request.user.zonaAdmin.probintziak_administra.all %}
    {% if tokia.bermatua == False %}
        <a href="{% url 'infoTurist:tokia_bermatu' tokia.id %}" type="button"
class="btn btn-success btn-lg">Bermatu. Informazioa egokia da.</a>
    {% endif %}
    {% endif %}
</h1>
```

Tokiaren informazio orokorra: Deskribapena, irudia, mapa, etiketak eta norek sartu duen.

```
<div class="row">
    <div class="col-md-3 col-sm-offset-2" style="">
```

```

<div class="row">
  {% if tokia.irudia != "" %}
    
    <b>Puntuazioa: </b><!-- { { media } } eduki behar da -->
    {% include 'infoTurist/tokiaHerria/puntuakBistaratu.html' %}
  {% else %}
    tokia eta puntuazioa
  {% endif %}
</div>
<div class="row">
  <a type="button" target="_blank" class="btn btn-info"
href="https://www.google.es/maps/dir/{ { tokia.latitudea|stringformat:'g' } },
{ { tokia.longitudea|stringformat:'g' } }/">Zabandu Mapa</a>
  <div id="map" style="height:250px;">
    MAPIA DAO HAMEN?
  </div>
</div>
{% if ibilbidea %}
  <div class="row">
    <h4>Ibilbidea: { { ibilbidea } } </h4>
    {% if not hurrengoTokia == "" %}
      <a type="button" href="{% url 'infoTurist:tokia_detail'
hurrengoTokia.id %}?ibilbidea={ { ibilbidea.id } }" class="btn btn-info">{% trans
'Ibilbideko hurrengora' %} >> { { hurrengoTokia } } </a>
    {% else %}
      <h3>{% trans 'Ibilbidearen bukaerara iritsi zara' %} </h3>
    {% endif %}
  </div>
{% endif %}
</div>
<div class="col-md-5" style="">
  <h3>{% trans 'Informazio orokorra' %}:
  {% if tokia.bermatua == True %}
    <i class="fa fa-check" aria-hidden="true"></i>
  {% endif %}
</h3>
<b>{% trans 'Deskribapena' %}:</b><br />
  {# tokiaren deskribapena hainbat hizkuntzatan sartu #}
  {% include 'infoTurist/sartu-editatu/tokia-describapena-hizkuntzetan.html' %}
<br /><br />
<b>{% trans 'Herria' %}:</b> { { tokia.ze_herritan } }
<br />
<br /><br />
<b>{% trans 'Tokiko etiketak' %}:</b><br />
  {% for zerbitzua in tokia.eskeinitako_zerbitzuak.all %}
    {% if etiketa.css_a != "" %}
      {% if not forloop.first %}
        ,
      {% endif %}
      <a href="{% url 'infoTurist:bilatuZerbitzuaz' zerbitzua.id %}"><i
class="fa { { zerbitzua.css_a } }"></i> { { zerbitzua.etiketa } } </a>

```



```

        {% endif %}
    {% endfor %}
<br /><br />
{% if nirelbilbideak %}
    <h4>{% trans 'Gehitu tokia zure ibilbideetara' %}</h4>
    {% include 'infoTurist/tokiaHerria/ibilbidenGordeUITokia_detail.html' %}
{% endif %}
<br /><br />
<b>{% trans 'Norek sartu' %}</b> <a href="{% url 'infoTurist:user_detail'
tokia.norek_sartu.id %}">{{ tokia.norek_sartu }} </a>{{ tokia.pub_date }}-an
</div>
</div>

```

Iritziak irakurtzeko gunea:

```

<div class="row">
    <div class="col-md-8 col-sm-offset-2" style="">
        <p class="lead">{% trans 'Jendearen iritziak' %}</p>
        <!-- IRITZI BAKOITZEKO -->
        {% for iritzia in iritziak %}
            <div class="row" style="border:black 1px solid;">
                <div class="col-md-3" style="">
                    {% with media=iritzia.puntuazioa %} <!-- Ze puntuakBistaratuk
'media' behar dau -->
                        <b class="text-uppercase">{{ iritzia.norena }}</b>{% if
request.user.is_staff %} (admin){% endif %} - {{ iritzia.data.date }}<br />{% include
'infoTurist/tokiaHerria/puntuakBistaratu.html' %}
                    {% endwith %}
                    <br />
                    <br />
                    {% if request.user.is_authenticated %}
                        {% if not iritzia.norena == request.user %}
                            <button type="button"
onclick="iritziaSalatu({{ iritzia.id }})" class="btn btn-danger btn-sm">{% trans 'Iritzia
salatu' %}</button>
                        {% endif %}
                    {% endif %}
                </div>
                <div class="col-md-8" style="margin-bottom:10px;">
                    <p class="lead ">{{ iritzia.titulua }}</p><p class="text-
justify">{{ iritzia.iritzia }}</p>
                    <!-- Like/Dislike egiteko -->
                    {% if request.user.is_authenticated %}
                        <kbd><span onclick="bozkatu({{ iritzia.id }}, 'True')"
onmouseover="style.cursor = 'pointer'">{% trans 'Baliogarria' %} <i class="fa fa-thumbs-
o-up fa-lg"></i></span></kbd>
                        -
                        <kbd><span onclick="bozkatu({{ iritzia.id }}, 'False')"
onmouseover="style.cursor = 'pointer'">{% trans 'Ez egokia' %} <i class="fa fa-thumbs-
o-up fa-lg fa-rotate-180 fa-flip-vertical"></i></span></kbd>
                    {% endif %}
                </div>
            </div>
        {% endfor %}
    </div>

```

```

        {% endif %}
        Iritziaren puntuak: {{ iritzia.iritziaren_balorazioa }} ( {% trans
'bozkak' %}: {{ iritzia.zenbat_aldiz_baloratua }})
    </div>
</div>
{% endfor %}
<!-- Iritzien paginatorra -->
<br />
<div class="col-md-10 col-sm-6 col-sm-offset-3">
    {% for num in iritziak.paginator.page_range %}
        {% ifequal num iritziak.number %}
            <span class="current"><button type="button" class="btn btn-default
btn-sm active">{{ num }}</button></span>
        {% else %}
            <a class="btn btn-default btn-sm" href="?page={{ num }}"
role="button"> {{ num }}</a>
        {% endifequal %}
    {% endfor %}
</div>
<!-- /Iritzien paginatorra -->
</div>
</div>

```

### Iritziak idazteko gunea:

```

<div class="row">
    {% if request.user.is_authenticated %}
        <div class="col-md-8 col-sm-offset-2">
            {% if feedback == "" %}
                <h5>{% trans 'Eman zure iritzia' %}
                {{ request.user.username }}</h5>
            {% else %}
                <b>{{ feedback }}</b>
                <script>alert('{{ feedback }}');</script>
            {% endif %}
            <br />
            <form action="." method="POST">{% csrf_token %}
                <div class="form-group" >
                    <label for="titulua">Balorazioa</label><br />
                    <i class="fa fa-star-o" id="izar1" onmouseover="izarGainean(this);"
onmouseout="izarKanpoan(this);" onclick="aukeratulzarra(this);"></i>
                    <i class="fa fa-star-o" id="izar2" onmouseover="izarGainean(this);"
onmouseout="izarKanpoan(this);" onclick="aukeratulzarra(this);"></i>
                    <i class="fa fa-star-o" id="izar3" onmouseover="izarGainean(this);"
onmouseout="izarKanpoan(this);" onclick="aukeratulzarra(this);"></i>
                    <i class="fa fa-star-o" id="izar4" onmouseover="izarGainean(this);"
onmouseout="izarKanpoan(this);" onclick="aukeratulzarra(this);"></i>
                    <i class="fa fa-star-o" id="izar5" onmouseover="izarGainean(this);"
onmouseout="izarKanpoan(this);" onclick="aukeratulzarra(this);"></i>
                </div>
            </form>
        </div>
    </div>

```

```

        <input type="hidden" name="puntuazioa" id="puntuazioa" value="" />
        <div class="form-group">
            <label for="titulua">{% trans 'Laburpena' %}</label>
            <input type="text" class="form-control" id="titulua"
placeholder="{% trans 'Laburpena' %}..." name="titulua" maxlength="30">
        </div>
        <div class="form-group">
            <label for="iritzia">{% trans 'Iritzia' %}</label>
            <textarea class="form-control" name="iritzia" id="iritzia" rows="3"
placeholder="{% trans 'Iritzia' %}..." maxlength="500"></textarea>
        </div>
        <button type="submit" class="btn btn-default">{% trans 'Iritzia eman'
%}</button>
    </form>
</div>
{% else %}
    <div class="col-md-8 col-sm-offset-2">
        <legend>{% trans 'Logeatu iritzia emateko' %}</legend>
        <form class="navbar-form navbar-left" method='POST' action="{% url
'auth_login' %}">{% csrf_token %}
            <div class="form-group">
                <input type="text" class="form-control" name="username"
placeholder="Username" />
            </div>
            <div class="form-group">
                <input type="password" class="form-control" name="password"
placeholder="Password" />
            </div>
            <button type="submit" class="btn btn-default">Login</button>
        </form>
    </div>
{% endif %}
</div>

```

Azkenik, gomendioen gunea hasiera batean ez daukala informaziorik:

```

<div class="row">
    <div class="col-md-8 col-sm-offset-2">
        <legend>{% trans 'Interesekoa izan daiteke zuretzat' %}...</legend>
        <div id="interesekoTokiak">
            <center><i class="fa fa-spinner fa-4x fa-pulse"></i></center>
        </div>
    </div>
</div>

```

## E.5- Zuzendariarekin bilerak

---

Eranskinen atal honetan zuzendariarekin egindako bilera akten laburpenak daude. Bilera bakoitzean bat egiten izan da eta bileran garatutakoa eta adostutakoa islatzen dute. Aipatzekoa da, hemen agertzen diren bilerak baino lehen beste batzuk egin genituela proiektuaren gaia aukeratzeko 2015eko ekainean hasita.

---

### **Bilera akta zenbakia: 1**

**Data:** 2015-10-13

**Garatutako gaiak:** Produktuaren ideian hitz egin da. Proiektuaren plangintzan martxan jarri behar naiz, plangintzan atazak zeintzuk diren finkatu eta bakoitzari denboraren zenbat % emango diodan erabaki. Produktuaren irismena finkatu, kalitate plana egin, kudeaketa plana...

**Hurrengo bilerarako egin beharrekoak:** Hurrengo bilera Azaroaren 2a (astelehena) 10:30tan izango da. Ordurako plangintza eginda auki behar dut.

---

### **Bilera akta zenbakia: 2**

**Data:** 2015-11-02

**Garatutako gaiak:** Plangintza nola dagoen ikusi da. Produktuaren analisi eta diseinuaz hitz egin da eta funtzionalitate batzuk gehitu dira. Analisi eta diseinuan egindakoaz, aplikazioa garatzen hasi behar naizela, txantiloak egin, datu baseko ereduak... Hurrengo bilerarako denbora tarte handiagoa aplikazioak denbora asko eramango duelako.

Driven kontu bat egin eta bertara proiektuko dokumentuak igo. Jose Ramoneaz partekatu karpeta.

**Hurrengo bilerarako egin beharrekoak:** Analisia bukatzeko aste bete daukat eta diseinua be aurrera jarraitu. Hurrengo bilera Abenduaren 9an (asteazkena, asteleheneko ordutegiaz) 10:30tan. Ordurako, txantiloak eginda ekarri behar ditut, diseinua eginda eta datu basea ere.

Behin hori eginda hasi igotzen materiala betiko hiru ataletan[sortu hiru karpeta] banatuta:

-Dokumentazioa

-Planifikazioak eta Estimazioak

-Inplementazioa

"3.- Igo beharrekoa GoogleDrivera -> Gaurko materiala eta erabaki duguna igotzeko aurki: bilera-akta(k), planifikazioak-estimazioak, analisi orokorra[Erabilpen kasuak eta Domeinuaren Diseinua,...]. Saiatu igotzen materiala era txukunenean, horrela zenbat eta gehiago eta landuagoa igo materiala orduan eta bihar

gutxiago gerorako.”

---

**Bilera akta zenbakia: 3**

**Data:** 2015-12-09

**Garatutako gaiak:** Egindako implementazioa komentatu. Funtzionalitate gehiago gehitu, txangoak, jaiak (ekitaldiak), administrazioa, agenda (aste buru pasa), herrien gainetik probintzia jarri.

**Hurrengo bilerarako egin beharrekoak:** Implementazioarekin jarraitu. Hurrengo bilera 2016ko urtarrilaren 11an 12:00-13:00 bitartean.

---

**Bilera akta zenbakia: 4**

**Data:** 2016-01-11

**Garatutako gaiak:** Non gauden eta nola goazen aztertu. Implementazioarekin jarraitu

**Hurrengo bilerarako egin beharrekoak:** Implementazioarekin jarraitu. Hurrengo bilera Urtarrilak 25

---

**Bilera akta zenbakia: 5**

**Data:** 2016-01-25

**Garatutako gaiak:** Donostia 2016koa komenta da. Dokumentu bat egin behar da produktua saltzen. Presentatzeko eskatzen dutena lortu. Aurkezpen dokumentua egin.

Produktuan, egutegi bat sortu, *calendar* estilora. Datu meatzaritzako pasatako dokumentuak irakurri.

**Hurrengo bilerarako egin beharrekoak:** Gaietan aipatutakoa egin. Hurrengo bilera Otsailak 15.

---

**Bilera akta zenbakia: 6**

**Data:** 2016-02-22 (atzeratuta)

**Garatutako gaiak:** Datu meatzaritzan egindako aurreraketak ikusi. Mongon zelan dauden datuak, gidoiak sortutako taulak, map-reduce... Domeinuen testuinguruari buruz, DSS2016rako dokumentua egiteko, ze atal egongo diren.

**Hurrengo bilerarako egin beharrekoak:** Aplikazioa egiten jarraitu eta DSS2016ko dokumentua ere bai. Hurrengo bilera Martxoak 17.

---

**Bilera akta zenbakia: 7**

**Data:** 2016-03-17

**Garatutako gaiak:** DSS2016 dokumentua entrega zuzentzeko. Aste santuen aurreko bilera da hau. Datu meatzaritzak eta aplikazioak nola egiten duten bat.

**Hurrengo bilerarako egin beharrekoak:** Apirilaren 12rako da DSS2016 dokumentua. Entrega behar direnak lortu. Hurrengo bilera apirilak 11.

---

**Bilera akta zenbakia:** 8

**Data:** 2016-04-11

**Garatutako gaiak:** DSS2016 komenta, entrega behar direnak. Aplikazioari bukaera emateko atzan birfintzeak.

**Hurrengo bilerarako egin beharrekoak:** Aplikazioan ibilbideen ordena konpondu, antzeko tokiak gomendatu iritzirik emanda ez daukan herrietan. Hurrengo bilera Apirilak 25ean.

---

**Bilera akta zenbakia:** 9

**Data:** 2016-04-25

**Garatutako gaiak:** Produktua alde batera uzten eta dokumentuaz hasten joan beharra. Produktuko azken birfinketak egin behar.

**Hurrengo bilerarako egin beharrekoak:** Memoriako aurkibidea, aurkezpena, sarrera eta motibazio, aurrekariak eginda. Hurrengo bilera helbide elektronikoz bidez adostuko da aurrerago.

---

**Bilera akta zenbakia:** 10

**Data:** 2016-05-19

**Garatutako gaiak:** DSS2016 onartuta geratu dela. Memoria ikusi zelan dagoen egindakoa.

**Hurrengo bilerarako egin beharrekoak:** Hurrengo bilera ekainak 1.

---

**Bilera akta zenbakia:** 11

**Data:** 2016-06-01

**Garatutako gaiak:** GAPari buruzko informazioaz ehu.eus-en. Matrikulatzeko moduak (idazkaritzatik egin behar), nota bat oraindik jarri gabe dagoelako GAURen...

**Hurrengo bilerarako egin beharrekoak:** Memoria amaitzen. Hurrengo bilera ekainaren 14an.

---

**Bilera akta zenbakia:** 12

**Data:** 2016-06-14

**Garatutako gaiak:** Memoria entrega zuzendariari.

**Hurrengo bilerarako egin beharrekoak:** Zer zuzendu bidaliko dit.

## **E.6- DSS2016ko ebazpena**

---

Hurrengo orrietan DSS2016koek gradu bukaerako lanen ebazpena dago autatutako 9 proiektuekin. Horietako bat proiektu hauxe izanda.

[http://dss2016.eu/images/Diru-laguntza/Ebaluazioa/Graduondoko\\_ikerlana/Gradu\\_lana\\_esleipen\\_ebazpena.pdf](http://dss2016.eu/images/Diru-laguntza/Ebaluazioa/Graduondoko_ikerlana/Gradu_lana_esleipen_ebazpena.pdf)



DONOSTIA / SAN SEBASTIÁN  
DSS2016.EU

EUROPAKO KULTUR HIRIBURUA  
CAPITAL EUROPEA DE LA CULTURA  
CAPITALE EUROPÉENNE DE LA CULTURE  
EUROPEAN CAPITAL OF CULTURE

Partaidetza eta Ebaluazio Arloa  
Dpto. Participación y Evaluación

Easo kalea 43  
Donostia / San Sebastián  
20006

+34 943 483 344

www.dss2016.eu  
info@dss2016.eu

1/2

### Aurrekariak:

- 2016ko urtarrilaren 25ean, DSS2016EUren webgunean eta DSS2016EUren ikergaietakoren bati buruzko Gradu-Amaierako lanen proiektuak sustatzeko asmoz, hautatutako gehienez 10 ikerketa-lanen artean 2.000 eurora arteko EcOC bidaia-poltsa esleitzeko deialdiaren iragarkia argitaratu zen.

- Eskaerak aurkezteko epea 2016ko apirilaren 12an amaitu zen, 17 eskaera aurkeztu direlarik.

- 2016ko apirilaren 21ean, Balorazio Batzordeak jasotako eskaerak aztertu eta ebaluatu ditu, VIII. Oinarrian adierazitako balorazio irizpideak jarraituz.

Azaldutakoaren arabera, eta deialdiaren oinarrietan ezarritako tramiteak eginda, Balorazio Batzordeak DSS2016ren Zuzendaritzari proposatzen dio, hurrengoan onartzea

### Antecedentes:

- Con fecha 25 de enero de 2016, se publicó en la web de DSS2016EU el anuncio de la convocatoria para el otorgamiento de una bolsa de viaje EcOC valorada en 2.000€, entre uno de los 10 (máximo) trabajos seleccionados, en relación a proyectos de trabajos de Fin de Grado sobre alguna de las líneas de investigación de DSS2016EU.

- El plazo de presentación de proyectos finalizó el 12 de abril de 2016, habiéndose presentado un total de 17 solicitudes.

- Con fecha 21 de abril de 2016, el Comité de Valoración ha analizado y evaluado las solicitudes recibidas, con aplicación de los criterios de valoración establecidos en la base VIIIª.

Por lo expuesto, una vez instruidos los trámites establecidos en las bases de la convocatoria, el Comité de Valoración propone a la Dirección de DSS2016EU la adopción de la siguiente

### EBAZPENA

- Ondoko Gradu Amaierako Lanen proiektuak hautatzea:

*Euskal Herriko Unibertsitateko:*

1. Heredero Martínez, Julia. *Encuentros sobrenaturales con Dios. El proceso de conversión de jóvenes evangelistas pentecostales en la CAPV.*
2. Iturralde Agirre, Beñat. *Ondare kultural ukiezina. Ulerkera eta kudeaketari buruzko hausnarketa antropologikoa.*
3. Iturrioz Huarte, Jone. *DSS2016 Europako Kultur Hiriburutza, azterketa ekonomikoa.*
4. Murillo Martínez, Saioa. *Euskal komunitate eta identitatearekiko jarreraren ikerkuntza, Berria egunkariarekin lankidetzan.*
5. Padilla Lopez, Leire. *La música como herramienta resocializadora.*
6. Recuero Linares, Mikel. *Los derechos y garantías de las personas refugiadas en la normativa.*
7. Santiago Martínez, Uxue. *Donostiako Piratak eta Donostiako*

### RESOLUCIÓN

- Seleccionar los siguientes Trabajos de Fin de Grado:

*De la Universidad del País Vasco:*



*komunitatearen arteko harremanera gerturapen eraldatzailea gizarte hezkuntzaren ikuspegitik.*

8. Urkiaga Aburruza, Mikel. *GAP: Turismo bulegoen webgune adimentsua.*

Partaidetza eta Ebaluazio Arloa  
Dpto. Participación y Evaluación

Easo kalea 43  
Donostia / San Sebastián  
20006

+34 943 483 344

www.dss2016.eu  
info@dss2016.eu

2/2

*Deustuko Unibertsitateko:*

*De la Universidad de Deusto:*

9. Icardo Isasa, Irati. *Medición del Impacto Cultural y la participación ciudadana en la Capitalidad Europea de la Cultura Donostia/San Sebastián 2016.*

- 10. lanaren hautapena hutsik uztea, aurkeztutako gainontzeko lanen ez-egokitasunarengatik. Hori horrela dela eta, ordezkotako lanen zerrendarik ere ez da osatu.
- Hautatutako bederatzi lanen egileek, beraien Gradu Amaierako Lanak aurkeztu beharko dituzte Balorazio Batzorde Bateratuaren aurrean. Lan horien balorazioaren ostean, EcOC bidaia-poltsa esleituko da.
- Declarar desierta la elección del décimo trabajo, por falta de idoneidad del resto de trabajos presentados. Del mismo modo, no se ha constituido la lista de trabajos sustitutos.
- Las personas autoras de los proyectos seleccionados deberán presentar sus Trabajos Fin de Grado ante la Comisión de Valoración Unificada. Una vez realizada la valoración, se adjudicará la bolsa de viaje EcOC.

Donostian, 2016eko apirilaren 26an.

En Donostia / San Sebastián, a 26 de abril de 2016.



Pablo Berástegui Lozano  
D. Zuzendari Orokorra