



# **TXOSTENA OROKORRA**

SwiftTECH Erronka

Eber Quintana eta Ander Ibañez

## AURKIBIDEA

DOKUMENTAZIO OROKORRA .....	3
<b>ENPRESAREN/TALDEA .....</b>	<b>3</b>
Enpresaren identifikazioa:.....	3
Erronkaren atalak:.....	3
<b>INTERFAZE GARAPENA .....</b>	<b>4</b>
Web Orriaren interfazea: .....	4
Ranking datuen kudeaketa: .....	5
<b>MULTIMEDIA PROGRAMAZIOA ETA GAILU MUGIKORRAK .....</b>	<b>7</b>
Jokoaren interfazea: .....	7
Datuen kudeaketa:.....	8
Joko Aukeratuta: .....	9
APPLE JOKOA: .....	10
<b>ENPRESA-KUDEAKETAKO SISTEMAK .....</b>	<b>12</b>
Odoos eta PostgreSQL instalazioak: .....	12
Modulua: .....	12
<b>DATU ATZIPENA .....</b>	<b>14</b>
NoSQL datu basea:.....	14
Web Zerbitzuak: .....	14
FASTAPI PYTHON:.....	15
<b>ZERBITZUAK ETA PROZESUEN PROGRAMAZIOA.....</b>	<b>17</b>
HARIAK:.....	17
<b>ENPRESA ETA EKIMEN SORTZAILEA .....</b>	<b>19</b>
Altxortegi Plana:.....	19
Promozio bideoa: .....	19
Altxortegi Plana:.....	20
<b>PLAY STORE IGOERA .....</b>	<b>21</b>

# DOKUMENTAZIO OROKORRA

## ENPRESAREN/TALDEA

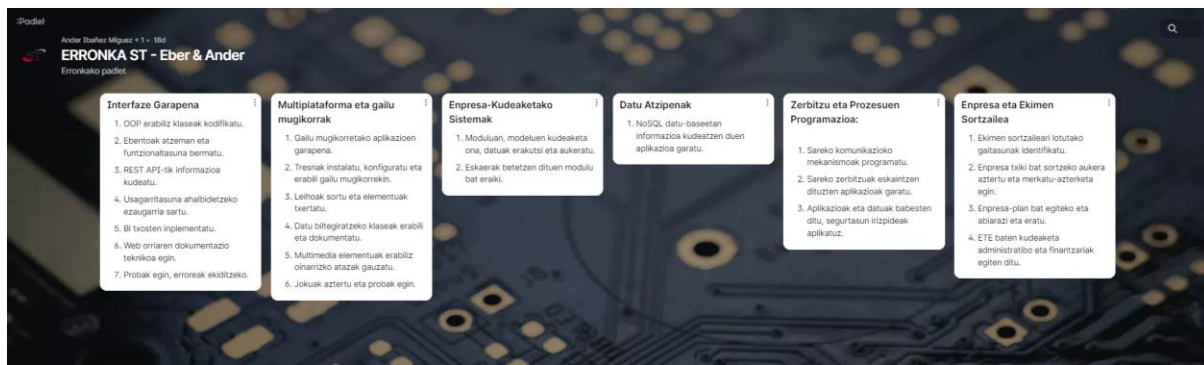
### Enpresaren identifikazioa:

- Enpresaren izena **"SwiftTECH"** da. Izen hau sinplea, originala eta erakargarria da. "Swift" hitzak abiadura eta eraginkortasuna iradokitzen ditu, hori ezin hobea da ordenagailu eta software garatzeko enpresa batentzat. Gainera, "Tech"-ekin konbinatzeak argi adierazten du konpainiaren ikuspegi teknologikoa. Laburbilduz, "SwiftTech"-ek modernoa dirudi eta bizkortasuna islatzen du zerbitzu teknologikoak ematean.
- Eslogana hau da: "Kodea eta soluzioak sortzen, zure mundua berritzen."
- Kolore printzipalak hauek izango ziren: **#9e0b0f** - **#8d99ae** - **#333333**



### Erronkaren atalak:

- Erronkari hasiera emateko taldekideen artean talde kontratua zehaztu eta sinatu dugu.
- Ondoren, zer egin behar dugun irakurri eta gero, parametroak zehaztu ditugu, nondik hasi jakiteko.



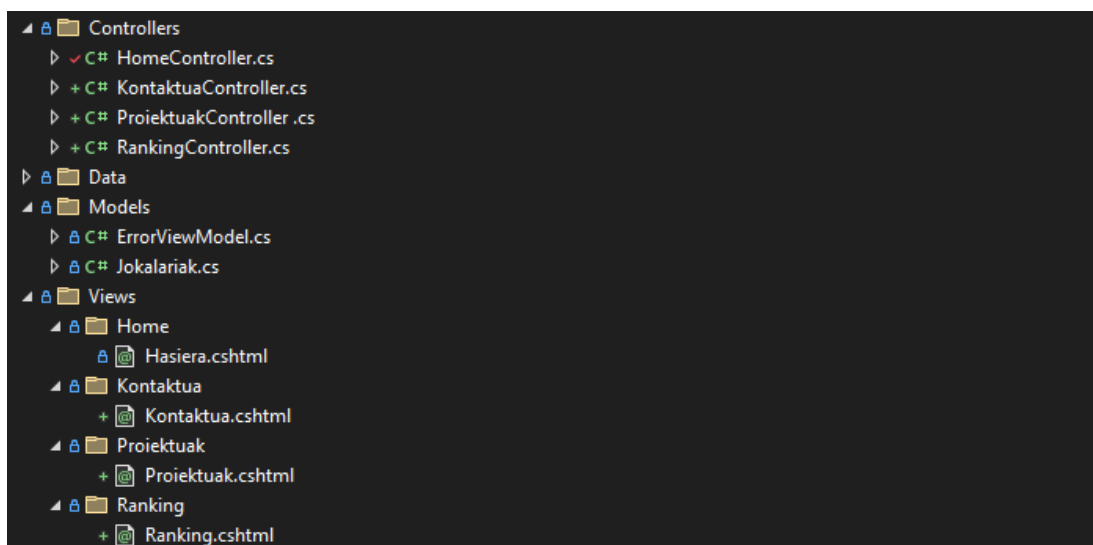
- Bukatzeko, planifikazio bat eramateko dokumentua prestatu dugu.

LAN TALDEAREN BARNE FUNTZIONAMENDUA:									
KONPROMEZUAK			HOBETZEKO EKINTZAK			ARURADUNA			PARTEHARTZAILAK
Egitea egitea etorriko			Dokumentu planifikatu			Idazkaria			Eber & Ander
Lan egitea etorriko						Koordinatzailea			Ander & Eber
Egitea egitea etorriko						Materialak			
ATZEA	ATAZA	ENTREGA	TALDEAN	BIKAKA	BANAKA	AURREKUSITAKO DENOBA	ARURADUNA	ATZARENDEGOKERA	IRAKASLEAREN ONESPENA
08/01/2024 - 09/02/2024	TALDEA: EGUREROKOA		ST				ST	PROZESUAN	
08/01/2024 - 09/02/2024	INTERF: Web Orri bukatu eta erabiltzailearen onarria		ST				ST	PROZESUAN	
08/01/2024 - 09/02/2024	APP: Jokoa bukatu behar den segurtasun eta erabiltzailearen onarria		ST				ST	PROZESUAN	
08/01/2024 - 09/02/2024	ERP: Ondo instalatu eta konfiguratu gure moduluak jara		ST				ST	PROZESUAN	
8/01/2024 - 12/01/2024	TALDEA: Irena, logoa, eslogana...		ST				ST	EGINDA	
8/01/2024 - 12/01/2024	TALDEA: Parametroak zehaztu (eremotan egin beharrekod)		ST				ST	EGINDA	
8/01/2024 - 12/01/2024	TALDEA: Proposamenduak aurkeztu		ST				ST	EGINDA	
8/01/2024 - 12/01/2024	TALDEA: Talde kontratua zehaztu eta sinatu		ST				ST	EGINDA	
8/01/2024 - 12/01/2024	INTERF: MVC proiektua sortu				ANDER		ANDER	EGINDA	
8/01/2024 - 12/01/2024	INTERF: AZ: Disenatzailearen		ST				ANDER	EGINDA	
8/01/2024 - 12/01/2024	ERP: Ondo instalatu				EBER		EBER	EGINDA	
8/01/2024 - 12/01/2024	ERP: Moduluak aurkeztu eta sortu beharrekod zehaztu				EBER		EBER	EGINDA	
8/01/2024 - 12/01/2024	DA: NoSQL:atik informazioa bidali		ST				ST	EGINDA	
8/01/2024 - 12/01/2024	APP: Disenatzailearen				ANDER		ANDER	EGINDA	
8/01/2024 - 12/01/2024	APP: Proiektua sortu		ST				ANDER	EGINDA	

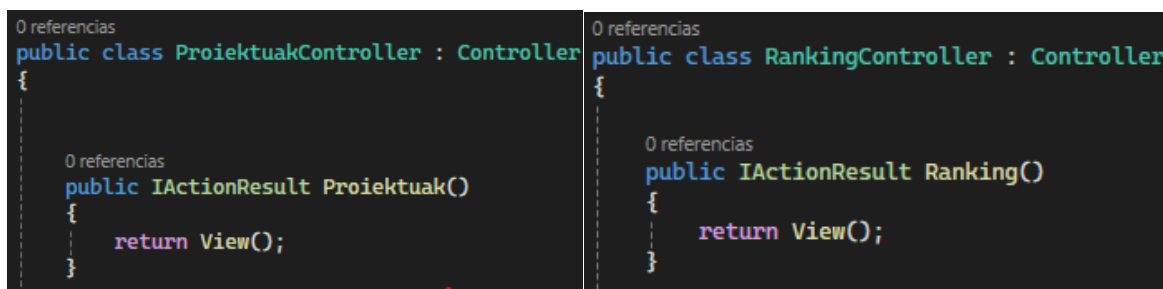
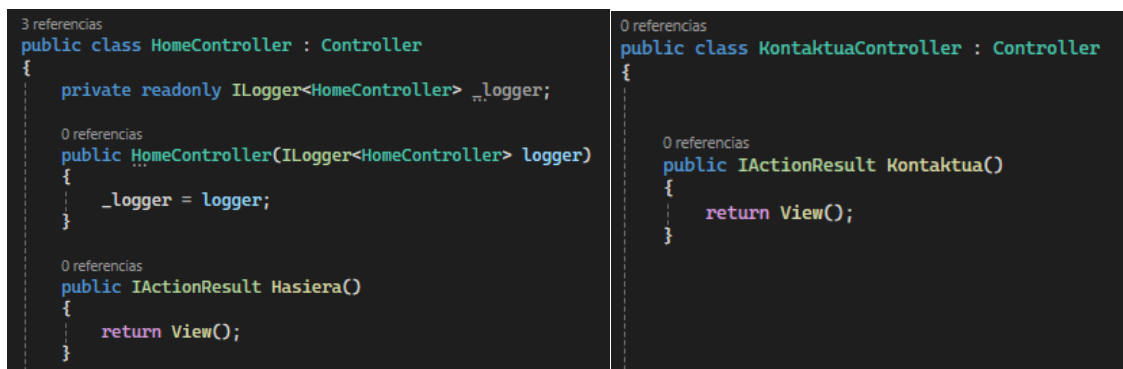
## INTERFAZE GARAPENA

### Web Orriaren interfazea:

- Enpresaren web orriarekin hasteko, lehenengo pausua diseinua pentsatzea izango litzateke. Behin txantiloia edukita, proiektuari hasiera ematea da hurrengo pausua.
- Hauek dira gure web orriak dituen aukerak: Hasiera, Proiektuak, Ranking eta Kontaktua. Bakoitzak bere “Views”, “Controllers” eta “Models”.
- Hau izango zen gure aplikazioaren egitura:



- Kontrolak:



## Ranking datuen kudeaketa:

- Prestatutako jokoaren datuen ranking-a bistaratzeko orria prestatu dugu, horren barruan APIa erabilita datu base barroko datuak JSON moduan jasotzen ditugu eta gero, taula batean gordetzen ditugu, erabiltzaileei pantailaratzeko.
- Horretarako, gure APIak lehenengo konexioa (web orria izanda, beti edukiko du konexioa irekita badago) eta gero, datu base barruko datuak hutsak ez direla konprobatzen ditu.
- Hau da APIko datuak hartzeko erabiltzen dugun “script”-a:

```

1  @model List<st_WebOrria.Models.Jokalariak>
2
3  @{
4      ViewBag.Title = "Ranking";
5  }
6
7  @section Scripts {
8      <!-- JQuery-ren CDN-a gehitzen da -->
9      <script src="https://code.jquery.com/jquery-3.6.4.min.js"></script>
10     <script>
11         $(document).ready(function () {
12             // Datuak eskuratzeko AJAX deia egiten da
13             $.get("http://10.23.28.190:8012/lortu_datuak", function (response) {
14                 var data = response.Jokalariak;
15                 console.log(data);
16
17                 var posizioa = 0;
18
19                 // Jokalariak iteratzeko buklea
20                 data.forEach(function (jokalaria) {
21                     posizioa++;
22
23                     if (posizioa > 10) {
24                         // Buclea bukatu
25                         return false;
26                     }
27
28                     else {
29
30                         if (posizioa > 0 && posizioa < 4) {
31                             // Irudiaren URL-a eraikitzen Url.Content erabiliz
32                             var irudia = '@Url.Content("~/irudiak/Top")' + posizioa + '.png';
33
34                             // Taula errenkadak sortu
35                             var row = "<tr><td><img style='width: 50px;' src='" + irudia + "'></td><td>" + jokalaria.izena +
36                                 "</td><td>" + jokalaria.abizena + "</td><td>" + jokalaria.puntuaketa + "</td></tr>";
37                             $("tbody").append(row);
38                         }
39
40                         // Taula errenkadak sortu
41                         var row = "<tr><td>" + posizioa + "</td><td>" + jokalaria.izena + "</td><td>" + jokalaria.abizena +
42                             "</td><td>" + jokalaria.puntuaketa + "</td></tr>";
43                         $("tbody").append(row);
44                     }
45                 });
46             }
47         });
48     }
49 }
50
51
52

```




- Goiko irudian ikusten den moduan, web orriko “ranking” orria kargatzen denean, APIari hots egiten diogu JSON fitxategia lortzeko. Behin hori lortuta, “Jokalariak” objektuak sortzen ditugu eta orriaren taulako errenkadetan sartzen ditugu.
- Hau izango zen “Jokalariak” objektuak sortzeko klasea:

```

1  namespace st_WebOrria.Models
2  {
3      4 referencias
4      public class Jokalariak
5      {
6          // Jokalarien parametroak
7          0 referencias
8          public string NAN { get; set; }
9          0 referencias
10         public string izena { get; set; }
11         0 referencias
12         public string abizena { get; set; }
13         0 referencias
14         public int denbora { get; set; }
15         0 referencias
16         public int puntuak { get; set; }
17     }
18 }

```

- Ranking orriaren adibide bat:

JOKALARIAK RANKING			
Posizioa	Izena	Abizena	Puntuak
	Cristiano	Ronaldo	971
	Eber	Siuuuuuu	968
	Lander	Camion	858
4	Lander	Chamorro	857
5	Ander	Ibanez	67

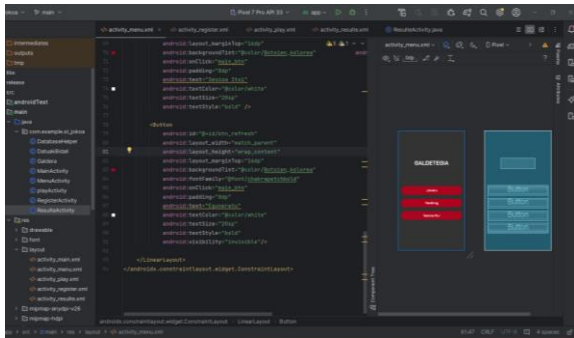
- Ranking-ak ondo egiteko, azkenean probak egiten joan gara, guk nahi dugun moduan azaltzea lortu arte. APItik datuak ondo etortzen direla ikusteko, JSON gorde dugu gure ekipoan, nola etortzen den bistaratzeko, gero “GET” komando konprobatzeko “POSTMAN” eta bukatzeko, edge barruko “INSPECCIONAR ELEMENTO”-rekin datuak jasotzen zituela konprobatu dugu.

# MULTIMEDIA PROGRAMAZIOA ETA GAILU MUGIKORRAK

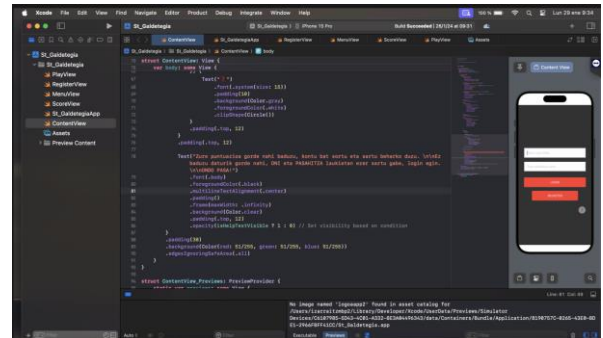
## Jokoaren interfazea:

- Jokoko interfazearekin hasteko, gure aplikazioa nolako izango den aukeratu beharko dugu. Hiru mota ditugu, baina gure kasuan Android Studio Java erabiliko dugu Android mugikorrenzat eta Apple aldetik XCode.

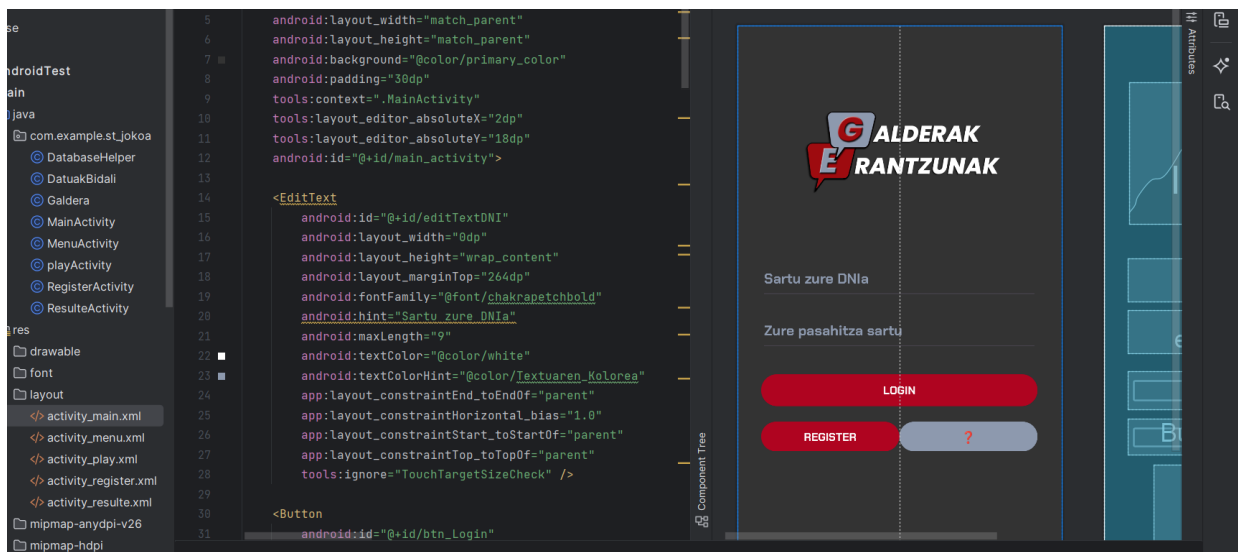
### ANDROID STUDIO



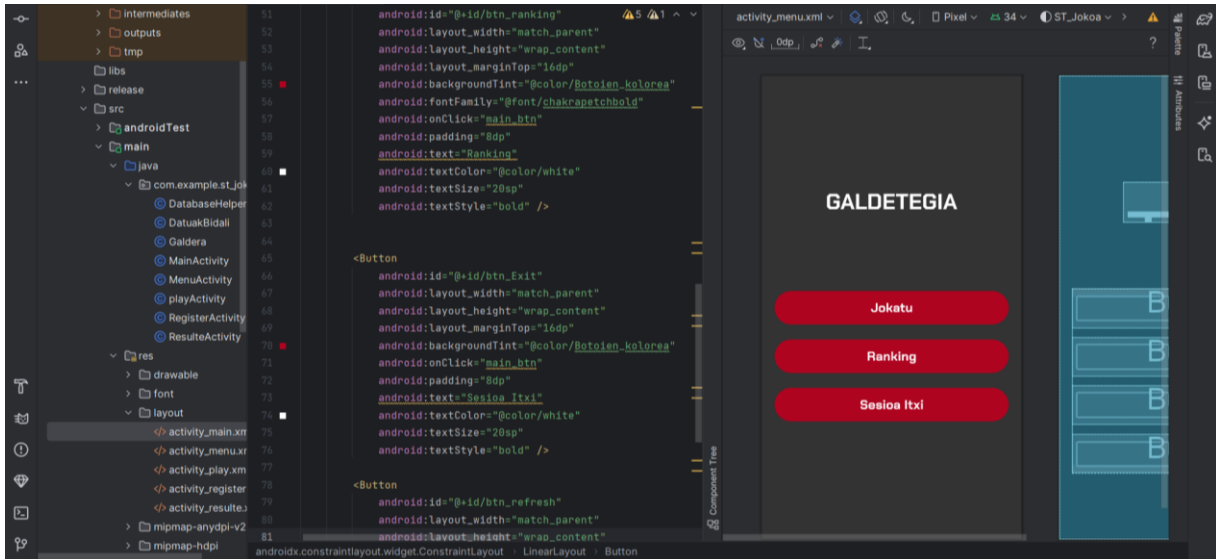
### XCODE SWIFTUI



- Azalpenak Android aplikazioarekin egin dira, baina bi aplikazioak prozesu berdina eman dute.
- Jokoak hainbat leiho ditu, lehenengo leihoa izena ematea edo sesioa hasteko aukera ematen duena, horrekin datu base barruan datuak gordetzerakoan, erregistroak izango ditugu. Noski, ez baduzu nahi izenik eman, beti jolastu dezakezu "OffLine" moduan, ezer sartu gabe.



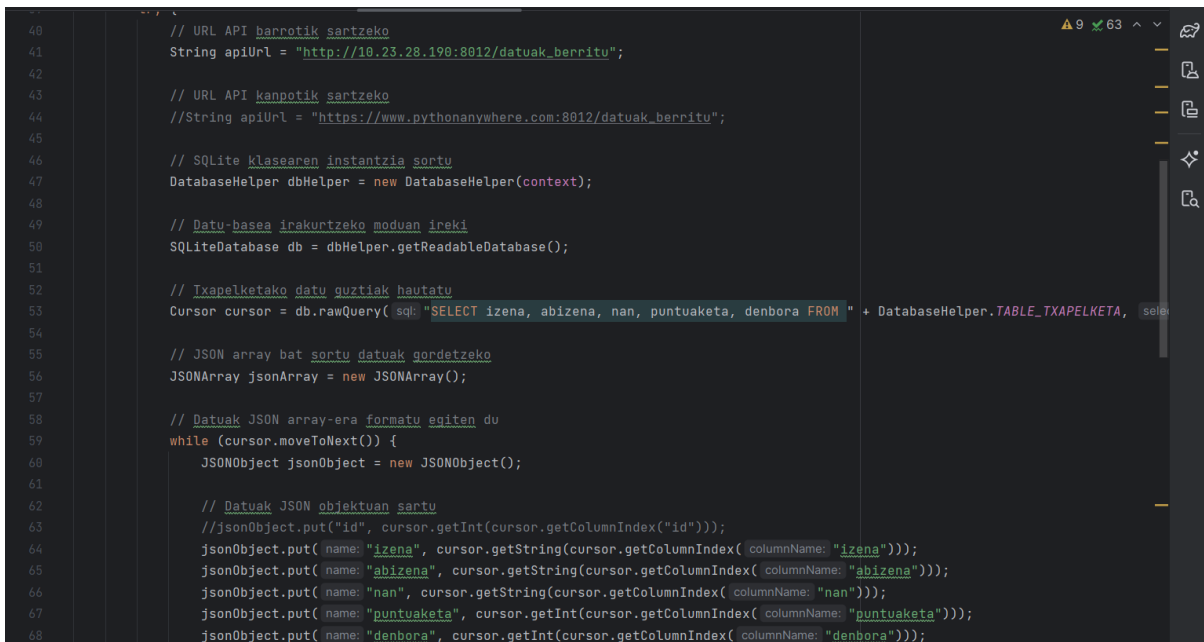
- Gero, behin hortik pasata, menu txiki bat edukiko dugu, horren barruan jolasteko, ranking taula ikusteko eta sesioa ixteko aukerak izango ditugu. Beraien izenak esaten duten moduan, "Jolastu" aukerak jokoari hasiera emateko botoia da, "Ranking" botoiak online ibili diren jokalarien datuak ikusteko aukera ematen du eta bukatzeko, "sesioa itxi" jokolariaren sesioa itxi eta hasiera leihora bueltatuko da.



- Oraingoz, “ranking” taulako datuak ikusteko edo zure partidako datuak gordetzeko, Izarraitz LHII ikastetxeko sare barruan egon beharko zara, APIra ez delako iristen sare kanpotik. Hori bai, datuak lokalki gordetzen dira, nahiz eta ranking taula ez pasa.

### Datuen kudeaketa:

- Lehen aipatu dugun moduan, gure jokoak bi modu ditu datuak gordetzeko, bat da Izarraitzeko sarearen barrutik gure APIari dei bat bota eta datu baseko ranking taulan gordetzea datuak eta bestea, lokalki.
- Datuak JSON moduan bidaltzen dira APIra.



- Lokalki, datuak SQLite barruan gordetzen dira. SQLite barruan hainbat taula ditugu, erabiltzaileen datuak, txapelketa datuak eta galdetegiaren galdera eta erantzunak gordetzeko taulak.
- Hau da gure “datuBaseHelper”, honen barruan SQLite datu baseko taulak etb... Sortzen ditugu:



```

50 private static final String CREATE_TABLE_USERS =
51 "CREATE TABLE " + TABLE_USERS + " (" +
52     COLUMN_ID + " INTEGER PRIMARY KEY AUTOINCREMENT, " +
53     COLUMN_NAME + " TEXT, " +
54     COLUMN_NAME2 + " TEXT, " +
55     COLUMN_NAN + " TEXT, " +
56     COLUMN_PASSWORD + " TEXT);";
57
58 private static final String CREATE_TABLE_GALDERAK =
59 "CREATE TABLE " + TABLE_GALDERAK + " (" +
60     COLUMN_ID2 + " INTEGER PRIMARY KEY AUTOINCREMENT, " +
61     COLUMN_GALDERA + " TEXT, " +
62     COLUMN_ERANTZUN_ZUZENA + " TEXT, " +
63     COLUMN_ERANTZUN_OKERRA_1 + " TEXT, " +
64     COLUMN_ERANTZUN_OKERRA_2 + " TEXT, " +
65     COLUMN_ERANTZUN_OKERRA_3 + " TEXT);";
66
67 private static final String CREATE_TABLE_TXAPELKETA =
68 "CREATE TABLE " + TABLE_TXAPELKETA + " (" +
69     COLUMN_ID3 + " INTEGER PRIMARY KEY AUTOINCREMENT, " +
70     COLUMN_IZENA + " TEXT, " +
71     COLUMN_ABIZENA + " TEXT, " +
72     COLUMN_NAN2 + " TEXT, " +
73     COLUMN_PUNTUAKETA + " INTEGER, " +
74     COLUMN_DENBORA + " INTEGER);";
75

```

id	izena	abizena	nan	pasahitza
Filtro	Filtro	Filtro	Filtro	Filtro
1	Test	Test		
2	Ander	Ibanez	73262242C	***

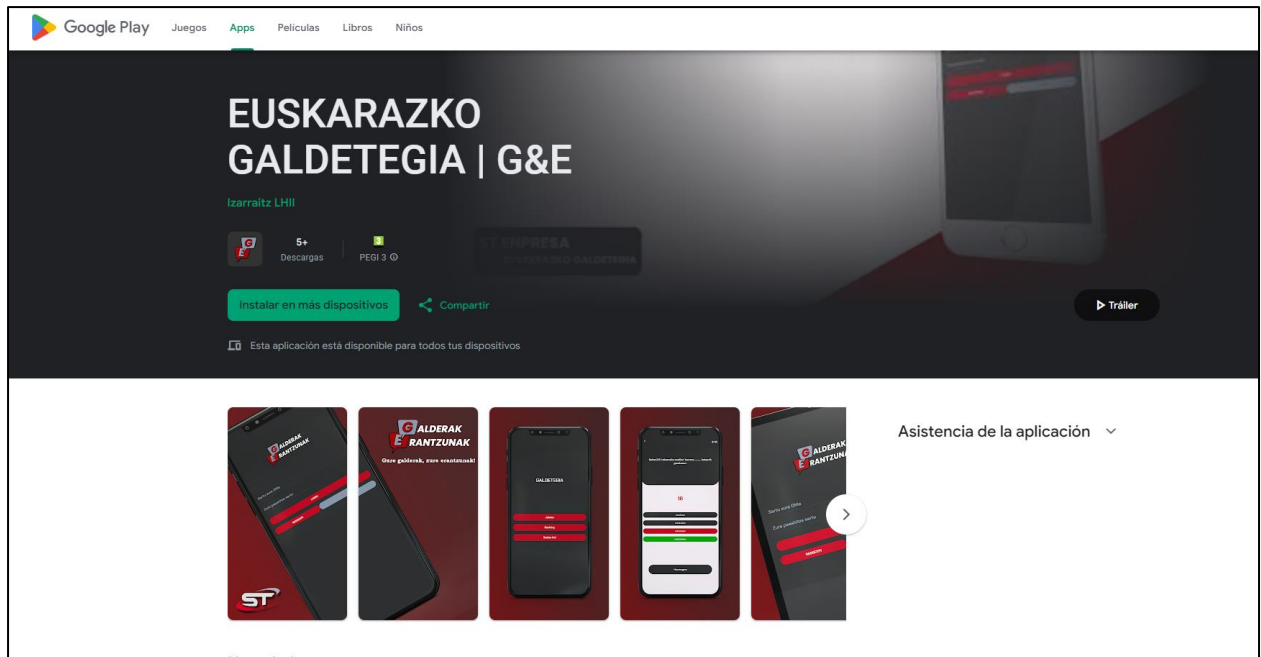
id	izena	abizena	nan	puntuaketa	denbora
Filtro	Filtro	Filtro	Filtro	Filtro	Filtro
1	Ander	Ibanez	73262242C	367	29

id	galdera	erantzunZuzena	erantzunOkerra1	erantzunOkerra2	erantzunOkerra3
Filtro	Filtro	Filtro	Filtro	Filtro	Filtro
1	Ordenagailuko teklatuari begiratzeko ...	R	D	F	U
2	Zein da munduko kirolrik ezagunena?	Futbola	Saskibaloia	Boleibola	Surfa
3	Kolore horia eta kolore urdina nahast...	Berdea	Gorria	Laranja	Morea
4	Munduko zein hiritan aurkitzen da Eiff...	Paris	Erroma	Iruñea	Pisa
5	Haragia jaten duten animaliak dira...	Haragijaleak	Orojaleak	Belarjaleak	Landarejaleak
6	Zein hilabetek izan ditzake 28 edo 29...	Otsaila	Marxoa	Urtarrila	Apirila

- Ranking taula erakustea ez badu lortzen (sare barruan ez egoteagatik), “catch” bati esker mezu bat irakutsiko dugu, zergatia azaltzen.

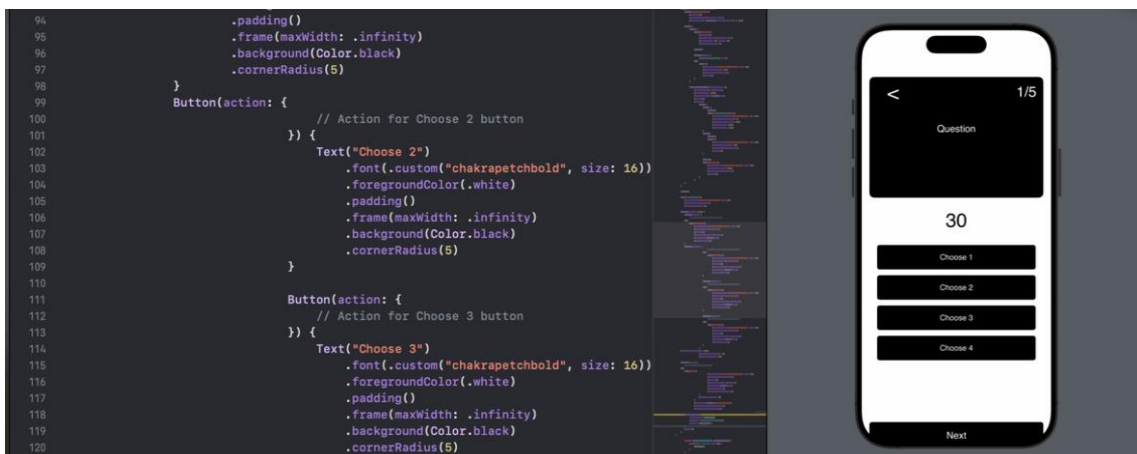
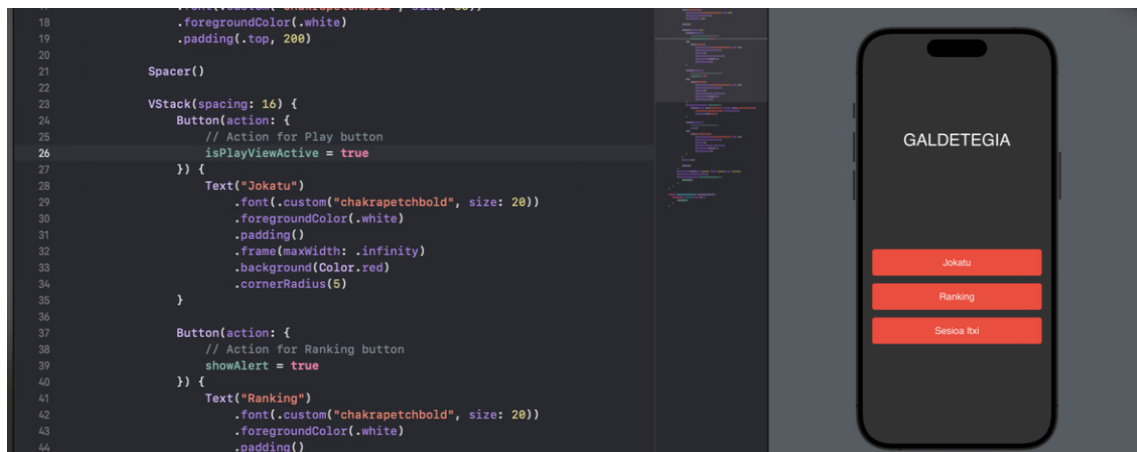
### Joko Aukeratuta:

- Honen aukera egiteko, hainbat pauso eman ditugu, lehenengoa zer motatako puntuaketa edo jokoak egin nahi dugun aukeratu behar genuen. Hori kontutan izanda, bi joko genituen pentsatuta, euskarazko galdetegia edo Flappy Bird antza duena. Azkenean “Euskarazko Galdetegia” pentsatu genuen. Enpresa barruko langileentzat eta puntuazioa emateko errazago izango zelako egitea.
- Gainera, Flappy Bird egoki bat egiteko denbora gehiago beharko genuen benetan ondo egiteko, horregatik galdetegia. Hobeto da joko bat ondo eta “perfekto” egitea, beste bat errore asko edo bukatu gabekoa egitea baino.
- Jokoa hurrengo URLarekin bistaratu dezakegu:  
[https://play.google.com/store/apps/details?id=com.stenpresa.st\\_jokoa](https://play.google.com/store/apps/details?id=com.stenpresa.st_jokoa)

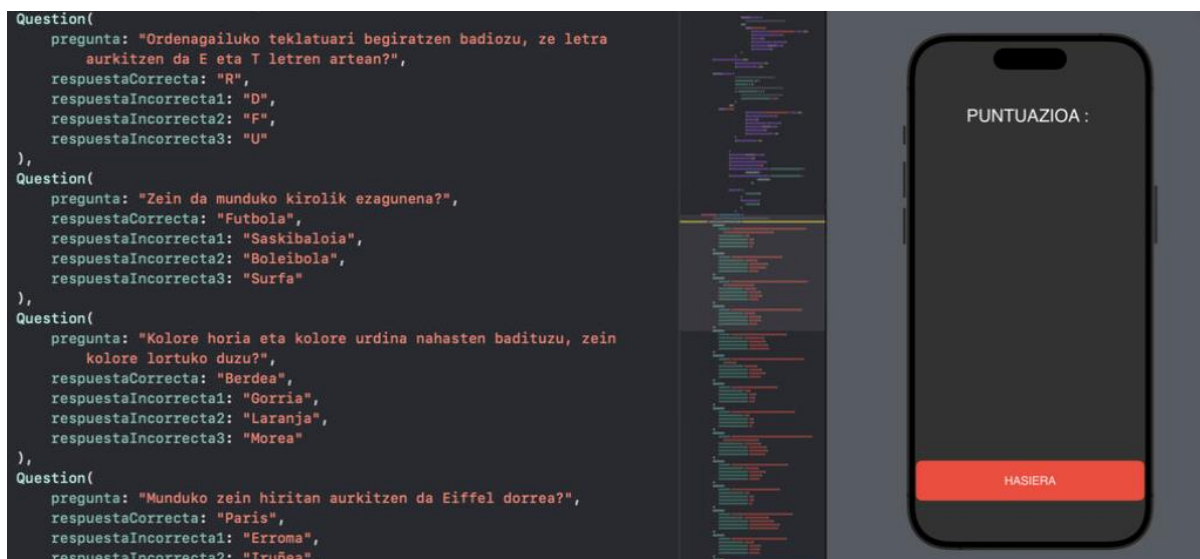


### APPLE JOKOA:

- Hau da gure Apple plataformarentzat egin dugun jokoaren adaptazioa. Azkenean joko berdina da, baina Apple barruko diseinuarekin.















Apple sistemarentzako jokoa lokalean bezala jolasteko pentsatu dugu eta horretara galderak eta erantzunak array baten sartu ditugu.



## ENPRESA-KUDEAKETAKO SISTEMAK

### Odoo eta PostgreSQL instalazioak:

- Lehenengo pausua, bi hauen instalazioa egitea izan da. Hauen instalazioa Docker barruan egin dugu, bi makinetan eta erabilitako sistema Ubuntu da.
- Instalazioa egiterako PostgreSQL portu iriki batekin jarri behar izan dugu, beste ekipotatik sartu ahal izateko.

<input type="checkbox"/>		st-compose	Running (2/2)	0.04%	2 hours ago			
<input type="checkbox"/>		web-1 30d65491e	Running	0.02%	<a href="#">8069:8069</a>			
<input type="checkbox"/>		db-odoo- 2b6f5567bl	Running	0.02%	<a href="#">5434:5432</a>			

### Modulua:

- ERP barruan gure jokoko ranking bat bistaratzea eskatu digute. Horretarako, gure jokoaren "txapelketa" izeneko modulu berri bat sortu dugu. Txapelketa modulua 5 parametro ditu izena, abizena, nan, puntuaketa eta denbora. Gero, hau izango zen gure modulua eskema:
  - Hau da gure modeloa:

```

6 class txapelketa(models.Model):
7     _name = 'txapelketa.txapelketa'
8     _description = 'txapelketa.txapelketa'
9
10    izena = fields.Char()
11    abizena = fields.Char()
12    nan = fields.Text()
13    puntuaketa = fields.Integer()
14    denbora = fields.Integer()

```

- Hau da "ir.model.access.csv":

```

1 id,name,model_id:id,group_id:id,perm_read,perm_write,perm_create,perm_unlink
2 access_txapelketa_txapelketa,txapelketa.txapelketa,model_txapelketa_txapelketa,base.group_user,1,1,1,1

```

- Manifest:

```

3 'name': "txapelketa",
4
5 'summary': ""
6     Txapelketaren modulua"",
7
8 'description': ""
9     enpresaren txapelketaren modulua
10    "",
11
12 'author': "ST enpresa",
13 'website': "http://stenpresa.eus",
14
15 # Categories can be used to filter modules in modules listing
16 # Check https://github.com/odoo/odoo/blob/15.0/odoo/addons/base/data/ir_module_category_data.xml
17 # for the full list
18 'category': 'Uncategorized',
19 'version': '0.1',
20
21 # any module necessary for this one to work correctly
22 'depends': ['base'],
23
24 # always loaded
25 'data': [
26     'security/ir.model.access.csv',
27     'views/txapelketa.xml',
28 ],
29 # only loaded in demonstration mode

```

- Txapelketa bista:

```

2 <odoo>
3 <data>
4 <record model="ir.ui.view" id="txapelketa.list">
5     <field name="name">txapelketa list</field>
6     <field name="model">txapelketa.txapelketa</field>
7     <field name="arch" type="xml">
8         <tree>
9             <field name="izena"/>
10            <field name="abizena"/>
11            <field name="puntuaketa" widget="integer" options="{ 'widget': 'integer', 'align': 'left' }"/>
12        </tree>
13    </field>
14</record>
15
16
17 <record model="ir.actions.act_window" id="jokalaria_list_action">
18     <field name="name">Jokalaria</field>
19     <field name="type">ir.actions.act_window</field>
20     <field name="res_model">txapelketa.txapelketa</field>
21     <field name="view_mode">tree,form</field>
22     <field name="help" type="html">
23         <p class="oe_view_nocontent_create">Sartu lehenengo jokalaria</p>
24     </field>
25</record>
26
27 <!-- Goi mailako menua: no parent -->
28 <menuitem id="main_txapelketa_menu" name="Txapelketa"/>
29
30 <menuitem id="txapelketa_menu" name="Txapelketa" parent="main_txapelketa_menu"/>
31
32 <menuitem id="jokalaria_menu" name="Jokalaria" parent="txapelketa_menu" action="jokalaria_list_action"/>

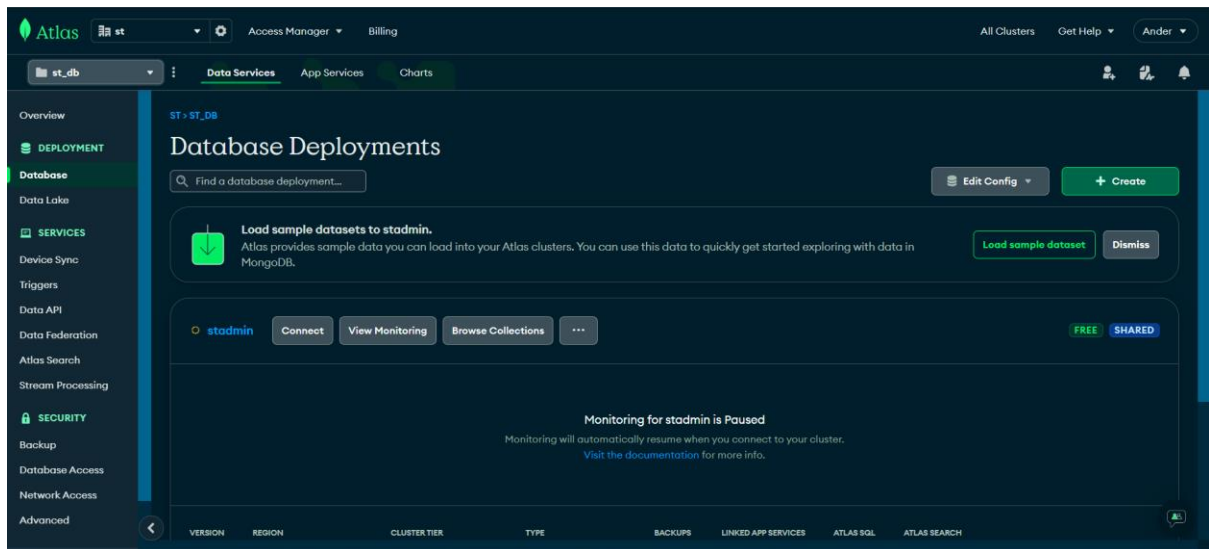
```

Txapelketa		
Jokalaria		
Izena	Abizena	Puntuaketa
<input type="checkbox"/> Cristiano	Ronaldo	971
<input type="checkbox"/> Ander	Ibanez	67
<input type="checkbox"/> Eber	Siuuuuuu	968
<input type="checkbox"/> Lander	Camion	858
<input type="checkbox"/> Lander	Chamorro	857
<input type="checkbox"/> Hh	Ja	0

## DATU ATZIPENA

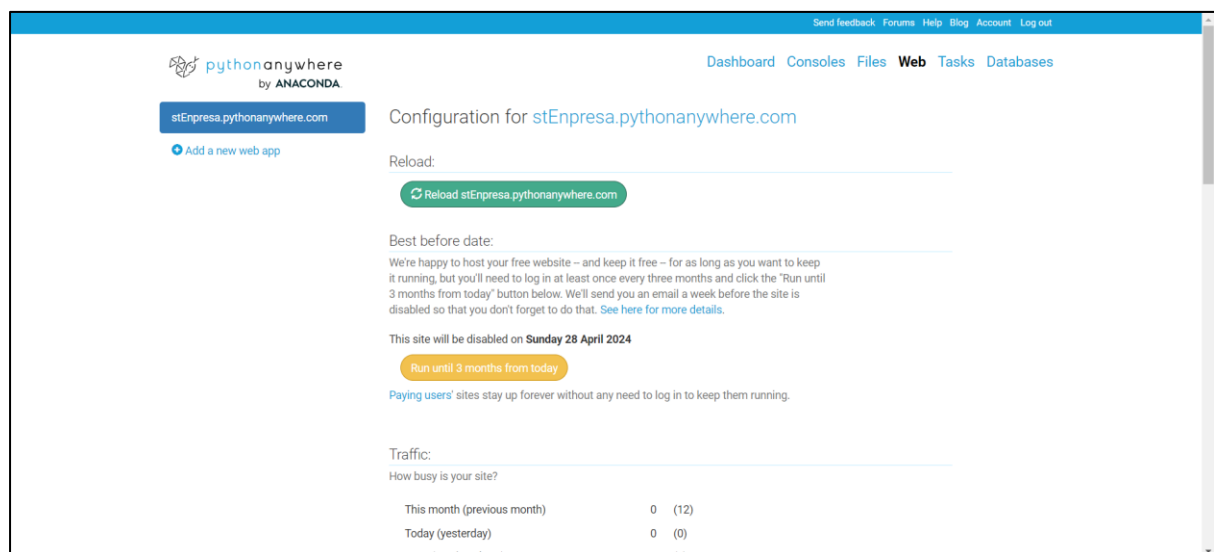
### NoSQL datu basea:

- NoSQL datu basea azkenean ez dugu erabili proiektuan, ez genuen beharrezkoa ikusten. Hori bai, egin eta prestatuta genuen erabiltzeko.
- NoSQL bezala “MongoDB” datu basea aukeratu dugu. MongoDBk web zerbitzuak eskaintzen ditu, horrela gure NoSQL datu basea beti erabiltzeko prest dugu, webean.
- Erabilitako web orria NoSQL kudeatzeko “cloud.mongodb.com” da.



### Web Zerbitzuak:


- Gure API-a web zerbitzu baten barruan jartzea nahi genuen, horrela edozein tokitatik hots egiteko aukera izango genuen. Baina, azkenean asuntu horrekin ez konplikatzearagatik, aurreragoko utzi dugu.
- Erabilitako web zerbitzua “pythonanywhere.com” da, zure API-a github bidez jeisten du eta gero, web horren barrutik nola exekutatu etb... Kudeatzeko aukera ematen du.



- Exekutatzeko orduan errore bat ematen digu, "PostgreSQL"-ra ez delako iristen. Hemen arazoa da, SQLLea ez dugula kanpotik sartzeko prest, orduan ezingo dugu edozein tokitatik berarekin konexioa egin.

```
File "/home/stEnpresa/.local/lib/python3.10/site-packages/sqlalchemy/pool/base.py", line 898, in __connect
self.dbapi_connection = connection = pool._invoke_creator(self)
File "/home/stEnpresa/.local/lib/python3.10/site-packages/sqlalchemy/engine/create.py", line 637, in connect
return dialect.connect(*cargs, **cparams)
File "/home/stEnpresa/.local/lib/python3.10/site-packages/sqlalchemy/engine/default.py", line 616, in connect
return self._pool.connect(*cargs, **cparams)
File "/home/stEnpresa/.local/lib/python3.10/site-packages/sqlalchemy/engine/psycopg2.py", line 122, in connect
conn = _connect(dsn, connection_factory=connection_factory, **kwasync)
sqlalchemy.exc.OperationalError: (psycopg2.OperationalError) connection to server at "10.23.28.192", port 5434 failed: Connection timed out
Is the server running on that host and accepting TCP/IP connections?
(Background on this error at: https://sqlalche.me/e/20/e3q8)
```

- Hala ere, datu base bat jartzeko aukera ematen du, baino, gure kasuan PostgreSQL erabiltzeko ordaindu egin beharko genuen.


pythonanywhere  
by ANACONDA

Dashboard Consoles Files Web Tasks Databases

MySQL

Postgres

### Upgrade required

Postgres is not available on your account. If you want all its shiny postgressy goodness, you'll have to [upgrade](#) to a Custom plan with Postgres enabled :)

Alternatively, if you're a reclusive millionaire, and you'd like to make a massive donation in order to enable free Postgres access for all users, then do get in touch and we'll see what we can do.

## FASTAPI PYTHON:

- Gure APIak hainbat gauz egiten ditu, "GET" egiten badugu adibidez, ranking taulako datuak ekartzen ditu datu basetik (PostgreSQL). Horrela web orri barruan, jokoaren ranking leihoaren barruan etb... Irakusteko aukera dugu.

```
88 @app.get('/lortu_datuak', response_model=List[Jokalaria])
89 async def lortu_datuak(db: Session = Depends(get_db)):
90     try:
91         # SQLAlchemy consulta para obtener los datos ordenados por puntuaketa descendientemente
92         data = db.query(TxapelketaTxapelketa).order_by(TxapelketaTxapelketa.puntuaketa.desc()).all()
93
94         # Convertir los resultados a un formato compatible con la respuesta JSON
95         result_data = [{"id": row.id, "izena": row.izena, "abizena": row.abizena, "nan": row.nan,
96                        "puntuaketa": row.puntuaketa, "denbora": row.denbora} for row in data]
97
98         return JSONResponse(content={"Jokalaria": result_data}, status_code=200)
99
100     except Exception as e:
101         raise HTTPException(status_code=500, detail=str(e))
102
```

- Ondoren, "POST" aukera bat du, post Android jokotik egiten dugu, honekin jokoko datuak hartu eta datu basera pasatzeko aukera dugu, konexioa baldin ba dugu.

```
63 @app.post('/datuak_berritu')
64 async def datuak_transferentzia(ranking_list: List[Ranking], db: Session = Depends(get_db)):
65     try:
66         for ranking in ranking_list:
67             db_entry = db.query(TxapelketaTxapelketa).filter(TxapelketaTxapelketa.nan == ranking.nan).first()
68
69             if db_entry:
70                 db_entry.denbora = ranking.denbora
71                 db_entry.puntuaketa = ranking.puntuaketa
72             else:
73                 db_entry = TxapelketaTxapelketa(**ranking.dict())
74                 db.add(db_entry)
75
76             db.commit()
77             return JSONResponse(content={"Mezua": "Datuak berritu dira."}, status_code=200)
78
79     except HTTPException as http_exc:
80         print(f"HTTPException: {http_exc}")
81         raise http_exc
82
83     except Exception as e:
84         print(f"Exception: {e}")
85         raise HTTPException(status_code=500, detail=str(e))
86
```

- APIak ondo funtzionatzen duela konprobatzeko beste “GET” funtzio bat dugu, “/ping” izenekoa.

```
@app.get("/ping")
def ping():
    return {"message": "¡API ondo dabil, OK!"}
```

```
{"message": "¡API ondo dabil, OK!"}
```

- Hainbat klase ditugu, bat “Jokalariak” izenarekin, honekin JSONen etortzen diren objektuak kontrolatzeko aukera dugu eta gero bestea, “TxapelketaTxapelketa” honekin datu basera sartu diren objektuak nolakoak izan behar duten kontrolatzen dugu.

```
13 class Jokalariak(BaseModel):
14     id: int
15     izena: str
16     abizena: str
17     nan: str
18     puntuaketa: int
19     denbora: int
20
21 class TxapelketaTxapelketa(Base):
22     __tablename__ = 'txapelketa_txapelketa'
23
24     id = Column(Integer, primary_key=True, index=True)
25     izena = Column(String, index=True)
26     abizena = Column(String, index=True)
27     nan = Column(String, unique=True, index=True)
28     puntuaketa = Column(Integer)
29     denbora = Column(Integer)
```



## ZERBITZUAK ETA PROZESUEN PROGRAMAZIOA

### HARIAK:

- Zerbitzuak piztuta dauden jakiteko hari bat sortu dugu non 10 minuturo API-a , Odoo eta PostgreSQL aktiboki dauden konprobaketa egingo duena, API-a eta Odoo url bidez topatuko du.

```
def check_services():
    while True:
        for service in services:
            thread = threading.Thread(target=check_service, args=(service,))
            thread.start()
            thread.join()

        time.sleep(600) # Hurrengo egiaztapena egin baino lehen 10 minutu itxaron
```

```
services = [
    {"name": "API", "url": "http://10.23.28.190:8012/ping", "expected_response": "¡API ondo dabil, OK!"},
    {"name": "Odoo", "url": "http://10.23.28.192:8069", "expected_response": "Odoo"},
]
```

- Eta PostgreSQL piztuta dagoen jakiteko, konexioa egingo genuke datu basera, horrela:

```
if service["name"] == "PostgreSQL":
    # Konexioa PostgreSQLlean saiatu
    conn = psycopg2.connect(
        host="10.23.28.192",
        port="5434",
        user="odoo",
        password="odoo"
    )
```

- Hiru zerbitzutako bat edo gehiagok ez badaude piztuta gure enpresako Gmailetik korreoa bidaliko du jakinarazteko arazo bat dagoela.

Errorea zerbitzua API egiaztatzean Externo Recibidos

st.enpresa.sl@gmail.com  
para mí

Traducir al español

Zerbitzua API egiaztatzean errorea gertatu da: HTTPConnectionPool(host='10.23.28.190', port=8012): Max retries exceeded with url: /ping (Caused by ConnectTimeoutError(<urllib3.connection.HTTPConnection object at 0x0000023302B42410>, 'Connection to 10.23.28.190 timed out. (connect timeout=None)'))

- Beste aldetik joko barruan (Android Studio) beste hari batzuk ditugu, baino garrantzitsu bat API-ra datuak pasatzekoa da.

```
public class DatuakBidali extends AsyncTask<Void, Void, Void> {  
  
    3 usages  
    private final Context context;  
  
    // Eraikitzailea: DatuakBidali klasearen instantzia sortzeko erabiliko den metodoa  
    2 usages  ⚙ anderibanez123  
    public DatuakBidali(Context context) { this.context = context; }  
  
    ⚙ anderibanez123  
    @SuppressWarnings("Range")  
    @Override  
    protected Void doInBackground(Void... voids) {  
  
        try {  
            // URL API barrotik sartzeko  
            String apiUrl = "http://10.23.28.190:8012/datuak_berritu";  

```

## ENPRESA ETA EKIMEN SORTZAILEA

### Altxortegi Plana:

- Lehenengo hilabetea zehar guk 10.000 euro ordaindu ditugu, hasiera gastuak kubritzeko. Ondoren, Urtarrilan zehar gure aplikazioa patrozinatu dugu eta horri esker patrozinatzaileak lortu ditugu (urte bateko lehen kontratua. Hilabetero diru kantitate eta materiala ematen digute.). Enpresaren lehenengo hilabetea ospatzeko lehiaketa bat egin dugu (programa lehiaketa bat).
- Hurrengo hilabetetan normal ibili gara, Ekaina arte.
- Ekainean, beste lehiaketa bat egin dugu (kasu honetan ekipoen mod lehiaketa bat).

	URTARRILA	OTSAILA	MARTXOA	APIRILA	MAIATZA	EKAINA	UZTAILA	ABUZTUA	IRAILA	URRIA	AZAROA	ABENDUA	GUZTIRA
<b>DIRU SARRERAK</b>													
Sozioen Aportazioa	10.000												
Patrozinatzaileak	5.200	5.200	5.200	5.200	5.200	5.200	5.200	5.200	5.200	5.200	5.200	5.200	
Lehiaketa	1.500					1.500						1.500	
Salmentak	12380	4602	4597	3602	6672	4340	4602	4602	8962	3542	10523	15380	
<b>SARRERAK GUZTIRA</b>	<b>27.580</b>	<b>9.802</b>	<b>9.797</b>	<b>8.802</b>	<b>11.872</b>	<b>9.540</b>	<b>9.802</b>	<b>9.802</b>	<b>14.162</b>	<b>8.742</b>	<b>16.723</b>	<b>20.580</b>	
Materiala	12000												
Lehiaketa gastuak	3000					3000						3000	
Alokairua	2000	2000	2000	2000	2000	2000	2000	2000	2000	2000	2000	2000	
Ur/argi horniketak	700	700	700	700	700	700	700	700	700	700	700	700	
Gizarte Segurantzak	100	100	100	100	100	100	100	100	100	100	100	100	
Publizitatea	2500	500	500	500	500	500	500	500	2500	500	2500	2500	
<b>IRTEERAK GUZTIRA</b>	<b>20300</b>	<b>3300</b>	<b>3300</b>	<b>3300</b>	<b>3300</b>	<b>6300</b>	<b>3300</b>	<b>3300</b>	<b>5300</b>	<b>3300</b>	<b>5300</b>	<b>8300</b>	
Hileko saldoa	7.280	6.502	6.497	5.502	8.572	3.240	6.502	6.502	8.862	5.442	10.423	12.280	
<b>SALDOA GUZTIRA</b>	<b>7280</b>	<b>13.782</b>	<b>20.279</b>	<b>25.781</b>	<b>34.353</b>	<b>37.593</b>	<b>44.095</b>	<b>50.597</b>	<b>59.459</b>	<b>64.901</b>	<b>75.324</b>	<b>87.604</b>	

### Promozio bideoa:

- Hau izango zan gure aplikazioa promozionatzeko bideo bat:  
<https://www.youtube.com/watch?v=dHz9xDpXpnY&feature=youtu.be>



### Altzortegi Plana:

- Bankuan 44.095 espero genituen, azkenean espero genuena baino 11.000 euro gehiagorekin bukatu dugu.
- Zor asko ez ditugu, zergaitik dana egunera eramaten dugu.
- 10.000 euro eman ditugu hasiera, beharrezko materiala erosteko.

EGOERA BALANTZEA			
<b>AKTIBOA</b>		<b>ONDARE GARBIA - PASIBOA</b>	
<b>AKTIBO EZ KORRONTEA</b>		<b>ONDARE GARBIA</b>	
<b>IBILGETU UKIEZINA</b>		Kapitala	10.000
Jabetza industriala (patenteak, markak,...)		Erreserbak	
Aplikazio informatikoak (software)		Patrozinatzaileak	36.400
		G-I	
<b>IBILGETU MATERIALA</b>		<b>PASIBO EZ KORRONTEA</b>	
Lurrak eta ondasun materialak			
Eraikuntzak		Epe luzerako maileguak	
Instalazio teknikoak		Epe luzerako hornitzaileak	
Materiala	12.000	Obligazioak eta <del>Bonoak</del>	
Altzairuak			
Informazioa prozesatzeko ekipamenduak			
Garraio elementuak			
Amortizazio metatua			
<b>ONDASUN HIGIEZINETAKO INBERTSIOAK</b>			
Alokatzeko lurrak, makinaria,...			
<b>FINANTZA INBERTSIOAK</b>			
Epe luzerako finantza inbertsioak			
Gordailuak			
Akzioak, <del>bonoak</del> , <del>tiantzak</del>			

<b>AKTIBO KORRONTEA</b>		<b>PASIBO KORRONTEA</b>	
<b>IZAKINAK</b>		Hornitzaileak	9.500
Salgaiak		Ibilgetuagatiko hornitzaileak	
Lehengaiak		Epe laburreko zorrak maileguengatik	
Produktu bukatuak		Ogasun publikoa hartzekoduna	
		Gizarte Segurantz Hartzekoduna	
<b>ZORDUNAK</b>		Epe laburreko maileguak	
Bezeroak		Langileen nominak	
Zordunak			
Epe laburreko finantza inbertsioak			
Epe laburreko kobrantza eskubideak			
<b>ESKUDIRUA</b>			
Bankuak	43.900		
Kutxa			
<b>AKTIBOA GUZTIRA</b>	55.900	<b>OND.GARBIA + PASIBOA</b>	55.900


## PLAY STORE IGOERA

- Lehenengo gauza aplikazioa igotzeko “Build.Gradle.kts” barruan, parametroak zehaztu behar ditugu:

```
defaultConfig {
    applicationId = "com.stenpresa.st_jokoa"
    minSdk = 24
    targetSdk = 33
    versionCode = 12
    versionName = "12.2.1"
```

- Igoera bat egin behar duzun bakoitzean “versionCode” aldatu behar dugu.
- Behin hori igotzeko, “Google Play Console” barruan “Produccion” eta gero, “Crear version”. Horren ondoren, “App Bundle” lekuan, Android Studiotik sortutako fitxategia igoko dugu.

App bundles



Arrastra aquí los app bundles que quieras subir

[Subir](#)
[Añadir de la biblioteca](#)

- Hori igotakoan, aplikazioko izena, irudiak, logoa... Jarri behar ditugu, bestela aplikazioa ezin izango dugu publiko egin.

Nombre de la aplicación \*

EUSKARAZKO GALDETEGIA | G&E

Este es el nombre que tendrá tu aplicación en Google Play 27/30

Descripción breve \*

GURE GALDERAK, ZURE ERANTZUNAK!

Breve descripción de tu aplicación. Los usuarios pueden mostrar la descripción completa. 31/80


Descripción completa \*

Joko hau SwiftTECH-ek sortutako euskarazko galdetegia da. Honen helburua ahalik eta azkarren erantzutea da, eta ondo erantzutea. Baina kontuan izanda lehen helburua ONDO PASATZEA dela!

SORTE ON!

195/4000

Capturas de pantalla de teléfonos \*







- Horren ondoren, Google-ek igoera onartu arte zai egon behar da.