



ERRONKA 3

Data:07/02/2022

Izenak: Beñat Vilarchao, Jon Xuinaga eta Asier Blazquez
Taldea: BJA

AURKIBIDEA:

AURKIBIDEA:	2
1. Sarrera	3
2. Testuingurua	3
3. Helburuak	3
3.1. Helburu Orokorrak	3
3.2. Helburu Zehatzak	3
4. Baliabideak	4
4.1. Baliabide Teknikoak	4
4.2. Giza Baliabideak	5
5. Denboralizazioa eta Ardurak	6
6. Garapen Teknikoa	7
6.1. Multimedia Programazioa Eta Gailu Mugikorrak	7
6.2. Datu atzipena	11
6.3. Interfazeen Garapena	12
6.4. Enpresa Kudeaketa Sistema	16
6.4. Zerbitzu eta prozesuen programazioa	18
7. Etorkizunerako ildoak	20
8. Ondorioak	20
8.1. Ondorio teknikoak	20
8.1.1. Egin ditugun hobekuntzak	20
8.2. Ondorio pertsonalak	20

1. Sarrera

Dokumentu honetan islatzen da erronka honetan egiteko eskatu zaizkigun lanak, helburuak eta nola planteatu eta garatu ditugun.

2. Testuingurua

BJA enpresak eskatu digu bere enpresan garatutako sistema informatikoan gauza batzuk aldatzeko. Alde batetik, mugikorreko joku bat garatu behar genuen. Gainera, mugikorreko aplikazioak funtzionaltasun desberdinak edukiko ditu. Hau da, Odoon erregistrata dauden erabiltzaleak konektatu daitezke login baten bitartez. Aurrekontuak egiteaz gain, jolasten duten partiden datuak PostgreSQL eta MongoDB datu baseetan gordeko dira. Datuak gordetzeko bi metodo erabili dira, jdbc bitartez eta socketen bitartez.

Mugikorreko jokuaz gain, web-gune bat garatzea eskatu digute asp.net bitartez. Web orrialdean datuak irakurtzeko API bat sortu da. Bertan mongo datubaseko datuak jasotzen dira. Api honetan socketeko zerbitzaria konfiguratuta dago eta bezeroa jokuan.



Web orrialdean, enpresako langileek beraien partiden datuak ikusi ditzazkete. Horrez gain, jokoaren iruzkina idatzi eta irakur dezakete. Gainera asebetetze inkesta bat betetzeko aukera daukate. Formulario honetan, jokuari puntuazio bat eman ahalko dute, zenbat denbora jolastu duten aipatzearekin batera..

3. Helburuak

3.1. Helburu Orokorrak

Erronka honetan, Enpresako lankideen arteko txapelketa bat egitea nahi da, horretarako mugikor joku bat garatzea pentsatu da, langileak bere denbora librean jolasteko. Jokalariak irabazten dituen puntuak web orrialde batean erakustea nahi da. Horrez gain, ranking moduko bat egotea da helburua, puntu gehien lortu duten jokalariak jakiteko.

3.2. Helburu Zehatzak

→ MULTIMEDIA PROGRAMAZIOA ETA GAILU MUGIKORRAK

- ◆ Jokoaren diseinua, logika eta itxura definitu.
- ◆ Jokoaren logika definitzen hasi.
- ◆ Jokoaren ranking-a SQLite datu base batean gorde, lokalean.
- ◆ Internet-erako konexioen kudeaketa, Android zerbitzuak erabiliz.
- ◆ Jokoaren partida bakoitzean gordeko diren datuak.

→ ENPRESA-KUDEAKETAKO SISTEMAK

- ◆ Odooko modulu bat sortu beharko da, txapelketako klasifikazioak ikusteko, txapelketako jokalaria enpresako langileak izango dira, beraz Odoon egon beharko dira erabiltzaileak.
- ◆ Mugikorreko jokuan lortutako puntuak PostgreSQL datu basean gordeko dira.
- ◆ Sortutako moduluan langile bakoitzak bere partidetako puntuazioak, departamentuko ranking-a eta adin tarte jakin bateko ranking-ak ikusi ahalko da.

→ ZERBITZU ETA PROZESUEN PROGRAMAZIOA

Prozesu, hariak eta Socket-ak erabiliz, Android aplikazioa eta Odoo-ko zerbitzariaren arteko konexioa egin.

- ◆ Haria kudeatu (sleep, async...).

→ INTERFAZE GARAPENA

- ◆ Visual Studio (ASP.NET) erabilita web orri bat sortu behar da enpresako txapelketa dinamizatzeko.
- ◆ Txapelketan denboraldi ezberdinak egongo dira, hiru hilabetero.
- ◆ Denboraldi bakoitzeko "Wall of Fame" eta "Wall of Shame" aterako dira.
- ◆ Web orriak REST API-a irakurriko du, jokalarien puntuazioa jasotzeko.
- ◆ Web horrialdean iruzkinak eta formulatiok bete ahalko da. Idatzitako informazioa SQL server datu base batean gorde beharko da.
- ◆ Hall of fameko eta hall of shameko erabiltzaileak atera ahalko dira informe baten bitartez.
- ◆ Web orriaren dokumentazio teknikoa egin beharko da. (DocFx).

→ DATU-ATZIPENA

- ◆ Mugikorreko jokoaren inguruko datuak Postgres-etik MongoDBra esportatuko dira.
- ◆ Web orriak kontsumituko duen APIa sortu behar da, MongoDBko datuak artu eta bidali egiten duena.

→ ZEHARKAKOAK

- ◆ Inplikazioa
- ◆ Talde Lana
- ◆ Ahozko komunikazioa

4. Baliabideak

4.1. Baliabide Teknikoak

Erronka hau garatzeko zenbait baliabide erabiltzea behar izan dugu. Baliabide hauek kategoriatan sailkatu daitezke: Software-a, datu baseak, Framework-a eta zerbitzaria:

SOFTWARE-A:

- Odoo: Mugikorreko puntuazioa ikusteko, erabiltzaileak kudeatzeko eta puntuazioak filtratzeko.
- Eclipse: Apia garatzeko erabilitako programa.
- Android Studio: mugikorreko jokia garatzeko erabili dugun aplikazioa.
- DBeaver: Postgres eta sqlite datu baseetan daukagun informazioa hobeto ikusteko eta sql sententziak egiteko erabili dugun aplikazioa.
- Compas: MongoDB datu baseko datuak hobeto ikusteko eta sententziak egiteko erabili den programa.
- Visual Studio 2019: ASP.NET web gunea garatzeko erabili den programa.



DATU BASEAK:

- PostgreSQL: Odoo ERP-ak bere zerbitzuak erabiltzean, gure informazio guztia Postgres datu base batean gordetzeko.
- Microsoft SQL Server: Iruzkinak eta formularioak gordetzeko datu basea.
- SQLite: Konexiorik ez dagoenean lokalean partidak gordetzeko.
- MongoDB: Jolastutako datuak gordetzeko.



FRAMEWORK-A:

- Spring: java aplikazioak garatzeko framework bat da.
- .NET: Windows aplikazioak garatzeko frameworka.



4.2. Giza Baliabideak

Taldeko kideen artean hainbat rol banatu genituen, proiektua hobeto antolatzeko. Hauek dira banatu genituen rolak:

- Koordinatzailea: koordinatzailearen funtzioak hauek dira: taldearen atazak antolatu, Kideak animatu, Taldearen egin beharrekoak ezagutu. Taldearen koordinatzailea Asier izendatu genuen.
- Bozeramailea: bozeramailearen funtzioak hauek dira: taldearen izenean hitz egin eta hartutako erabakiak adierazi besteei. Gure taldearen bozeramailea Jon izendatu genuen.
- Idazkaria: idazkariaren funtzioak hauek dira: Informazio garrantzitsua idatzi, hizkuntza zuzena erabili eta dokumentuak gorde eta partekatu. Taldearen idazkari bezala Beñat izendatu genuen.
- Laguntzailea: Denok izan gara elkarren laguntzaile UwU.



5. Denboralizazioa eta Ardurak

Taldeko kide bakoitzak zer egin behar duen antolatzeko Trello aplikazioa erabili dugu zeregin guztiak zeinek eta noiz egin behar duen kudeatzeko. Erronka honetan bakarrik erabali dugu Trello eta hemen dago Trello-ko linka:

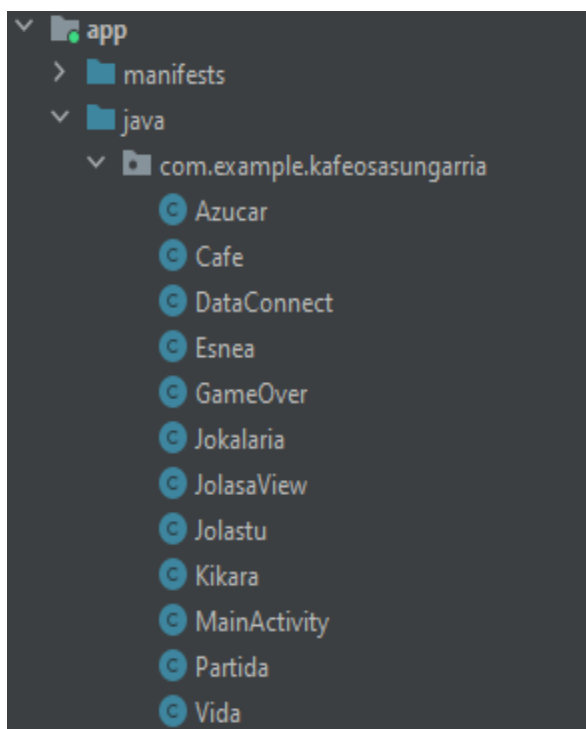


6. Garapen Teknikoa

6.1. Multimedia Programazioa Eta Gailu Mugikorak

Android studio-n Joku bat egin behar izan dugu, jokua helburua “Kafe osasungarri” bat egitea da, horretarako gure jokua mugitu ahal dugun kikara bat daukagu eta helburua “kafe makinatik” jaisten diren esne eta kafe tantak hartu behar izango ditugu eta azukre koxkorak sahiestu.

Horretarako View berri bat sortu dugu (JolasaView) , bertan invalidate “motako” hari batean canvas erabiliko ditugun objektu ezberdinak marrazten ditugu eta handler bat erabiliz kirarari mugikortasuna ematen diogu.



- Azucar: Azukre objektuaren klasea
- Cafe: Kafe objektuaren klasea
- DataConnect: Sqlite-ra eta Socketekin datuak bidaltzeko funtzioak
- Esnea: Esnea objektuaren klasea
- GameOver: jokua amaitzean DataConnect-eko funtzioei egiten die deia.
- Jokalaria: Jokalariaren klasea
- JolasaView: Jolasaren vista, canvas-arenkin etab.
- Jolastu: jolastu klasean JolasaView kargatuko dugu.
- Kikara: kikara objektuaren klasea
- Main Activity: Sqlitetik jokalaria kargatzen dira, eta hasierako menua dago, bertan logeatu zaitezke.
- Partida: Partidaren klasea
- Vida: bizitza objektuaren klasea

Objetu desberdinen jausia

Horretarako Esne klasea definitu izan dugu.

```
public class Esnea {  
  
    Context context;  
    Bitmap EsneaIrudia;  
    int esneaX, esneaY;  
    Random random;  
  
    public Esnea(Context context) {  
        this.context = context;  
        EsneaIrudia = BitmapFactory.decodeResource(context.getResources(),  
            R.drawable.leche);  
    }  
}
```

```

        random = new Random();
        esneaX = random.nextInt(JolasaView.pantallaAncho);
        esneaY =0;
    }

    public Bitmap getEsneaIrudia(){
        return EsneaIrudia;
    }

    int getEsneaIrudiaAncho(){
        return EsneaIrudia.getWidth();
    }

    int getEsneaIrudiaAlto(){
        return EsneaIrudia.getHeight();
    }
}

```

bertan erabiliko dugun argazkia eta hasierako posizioa (y=0 eta x random gure pantallaren rangoaren barruan) definitzen ditugu.

Objetuen marrazketa eta Kolisioa

```

for(int i =0; i<leches.size();i++){

    if(leches.get(i).esneaX > pantallaAncho - leches.get(i).getEsneaIrudiaAncho()){
        leches.get(i).esneaX = pantallaAncho - leches.get(i).getEsneaIrudiaAncho();
    }else if(leches.get(i).esneaX < 0){
        leches.get(i).esneaX = 0;
    }
    // marrazketa
    canvas.drawBitmap(leches.get(i).getEsneaIrudia(), leches.get(i).esneaX,
leches.get(i).esneaY, null);
    leches.get(i).esneaY +=velocidad;

    //kolisioa

    if((leches.get(i).esneaY >= kikara.kikaraY)
        && leches.get(i).esneaY <= kikara.kikaraY + kikara.getKikaraIrudiaAlto()
        && leches.get(i).esneaX <= kikara.kikaraX + kikara.getKikaraIrudiaAncho()
        && leches.get(i).esneaX >= kikara.kikaraX
        && leches.get(i).esneaX <= pantallaAncho){
        puntuacion++;
        leches.remove(i);
    }else if(leches.get(i).esneaY >= (pantallaAlto)) { //para eliminar los que se salen de
la pantalla
        leches.remove(i);
    }
}
}

```


Ausazko zenbaki bat sortu dugu, orduan gure zenbakiaren arabera objektu ezberdinak eroriko dira eta hurrengoa gertatuko da.

Lehendik sortutako objektu mota hori, mota horretako arrayist batean gehituko dugu. Ondoren Arraylistean dugun objektu bakoitzeko bukle bat egingo dugu eta bukle bakoitzean objektu bakoitzaren altuera aldatzen joango gara adibidez bukle bakoitzeko objektuen altuera 15 pixel jeisten joango da eta honela objektuak erortzen joango dira.

Behin erortzen ari diren objektu hauen altuera pantailaren amaierara iristen denean objektu hori arraylistetik ezabatuko dugu (bestela objektu infinituak sortzen arituko ginateke).

Erortzen ari den objektuetako batek kikararen posizioarekin koinziditzen duenean, kolisiotzat joko dugu, honela azukre koxkor bat bada bizitza bat kenduko digu eta esne edo kafe tanta bat bada puntu bat gehituko zaigu gure markagailura.

Bizitzarik gabe geratzen garen momentuan jolasa bukatu egingo da eta lortutako puntuazioa GAME OVER pantaila batean erakutsiko da, pantaila honetan berriz jolasten hasteko aukera izango dugularik.

```
if(life <=0){
    amaitu= true;
    handler= null;
    Jolastu.getPartida().setPuntuacion(puntuacion);
    Intent gameOver = new Intent(getContext(),GameOver.class);
    gameOver.putExtra("partida",Jolastu.getPartida());
    getContext().startActivity(gameOver);
    ((Activity) getContext()).finish();
}

if(!amaitu) {
    cont++;
    handler.postDelayed(haria, 0);
}
```

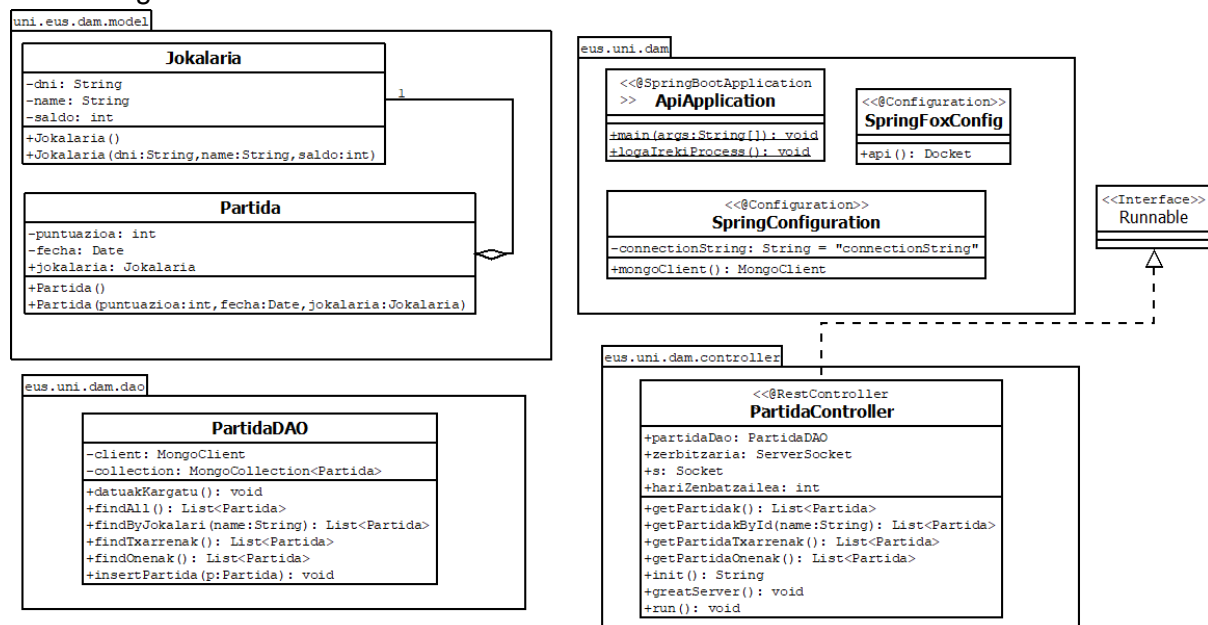


6.2. Datu atzipena

Atal honetan Visual Studioko aplikazioan kontsumitu dugun RestApi-a garatu dugu, horretarako Spring web framework-a erabili dugu. Honen helburua gure jokuan jolasten ditugun partidak MongoDB datu base batean gordetzea zen, eta Spring-en mapeoen bitartez RestApi-a sortu eta Visual aplikazioan partiden buruzko informazioa bistaratzea..

Horretaz gain Swagger funtzionaltasuna ezarri diogu gure proiektuari, bertan guren api-aren dokumentazio osoa daukagu, non bertan api-ak egiten dituen funtzio guztiak eta hauetako bakoitzak duen egitura eta bueltatuko duen informazioa kontsultatu dezakegu.

Klase Diagrama.



Android Studiotik partiden datuak jasotzeko socket baten bitartez egiten dugu, socket honetatik jasotzen dugun informazioa JsonParser libreriarekin atzitzen dugu, eta behin formatu egokian dugunean MongoDB-ra bidaltzen ditugu partida hauek.

Datuak android-etik funtzio honekin bidaltzen ditugu.

```
public String datosJson() throws JSONException {
    String s="[";
    String separador=",";
    int i=0;
    for (Partida p : getPartidas()) {
        JSONObject datos = new JSONObject();
        JSONObject jokalariObj = new JSONObject();
        jokalariObj.put("dni", p.getJokalaria().getDni());
        jokalariObj.put("name", p.getJokalaria().getName());
        jokalariObj.put("surname", p.getJokalaria().getSurname());
        jokalariObj.put("saldo", p.getJokalaria().getSaldo());

        JSONObject partidaObj = new JSONObject();
```

```

partidaObj.put("puntuacion", p.getPuntuacion());
partidaObj.put("fecha", p.printFecha());
partidaObj.put("jokalaria", jokalariaObj);

datos.put("partida", partidaObj);

if(i!=getPartidas().size()-1) {
    s = s + datos.toString() + separador;
}else{
    s = s + datos.toString()+"]";
}
i++;
}
return s;
}

```

Java aplikazioan horrela irakurtzen ditugu jasotako datuak

```

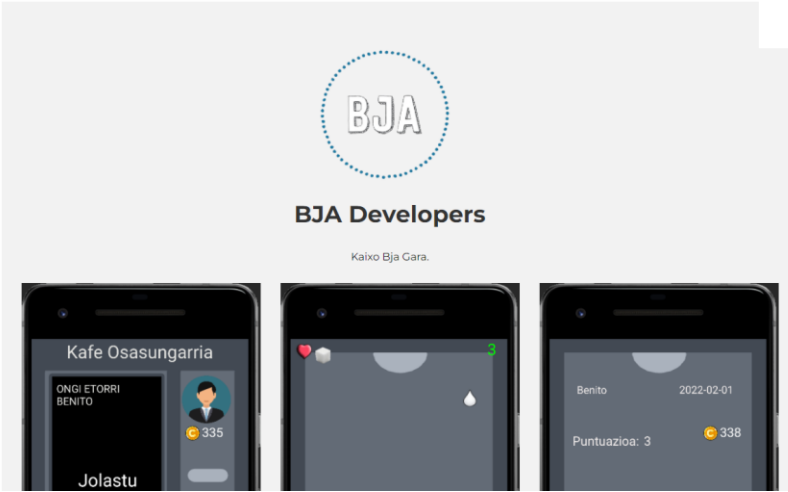
JSONArray jsonArr = new JSONArray(str_mezua);
for (int i = 0; i < jsonArr.length(); i++) {
    JSONObject jsonObj = jsonArr.getJSONObject(i);
    JSONObject partidaJSON = (JSONObject) jsonObj.get("partida");
    JSONObject jokalariaJSON = (JSONObject) partidaJSON.get("jokalaria");
    System.out.println(jsonObj);
    String data = (String) partidaJSON.getString("fecha");
    SimpleDateFormat formato = new SimpleDateFormat("yyyy-MM-dd");
    Date fecha = formato.parse(data);
    fecha = new Date();
    int puntuazioa = partidaJSON.getInt("puntuacion");
    String dni = (String) jokalariaJSON.getString("dni");
    String name = (String) jokalariaJSON.getString("name");
    int saldo = jokalariaJSON.getInt("saldo");
    Jokalaria j = new Jokalaria(dni, name, saldo);
    Partida p = new Partida(puntuazioa, fecha, j);
    partidaDao.insertPartida(p);
}

```

6.3. Interfazeen Garapena

Visual Studio aplikazioa erabiliz Rest-API bat kontsumitzen duen WEB bat egin dugu. Web honetan BJA enpresako langileek KAFE OSASUNGARRIA jolaseko lehiaketan jokatuak partidak erakusten dira. Logeatuta dagoen jokalariaren partidak, jolastu diren partida guztiak, 10 partida onenak, eta 10 partida txarrenak. Azken bi aukera hauetan txosten bat inprimatuko duen aukera daukan botoi bat jarri dugu. Honez gain enpresako kide guztiek iruzkinak egin ahal dituzte haserako orrialdean, eta beste leiho batean formulario bat bete ahalko dute jolasari buruzko puntuazio bat eman ahal izanda, bataz beste zenbat ordu jolastu duten esateaz gain.

HASERAKO ORRIA



JOLASAREN IRUZKINAK

Iruzkinak



01/02/2022 12:22:18

sasdsds

" sds"



07/02/2022 11:32:38

dfsdfsdf

" fsdfsdf"

Iruzkin berria idatzi

Zure iruzkinak lagatzeko konektatu zaitetz

Login

JOKALARI GUZTIEN PARTIDA GUZTIAK

Guztiak

Erakutsi 10 erregistro

Bilatu:

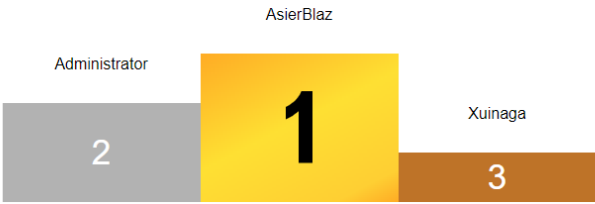
Name	Puntuazioa	Fecha
AsierBlaz	75	2022-02-04
Administrator	69	2022-02-04
Xuinaga	66	2022-02-04
Lander	64	2022-02-04
Xuinaga	63	2022-02-04
Administrator	62	2022-02-04
Administrator	60	2022-02-04
Periko	59	2022-02-04
Lander	57	2022-02-04
Administrator	52	2022-02-04

70 erregistrotik 1 - 10 ikusgai

Aurrekoa1234567Hurrengoa

10 PARTIDA ONENAK

Onenak



Erakutsi 10 erregistro

Bilatu:

Name	Puntuazioa	Fecha
AsierBlaz	75	2022-02-04
Administrator	69	2022-02-04
Xuinaga	66	2022-02-04
Lander	64	2022-02-04
Xuinaga	63	2022-02-04
Administrator	62	2022-02-04
Administrator	60	2022-02-04
Periko	59	2022-02-04
Lander	57	2022-02-04
Lander	52	2022-02-04

10 erregistrotik 1 - 10 ikusgai

Aurrekoa1Hurrengoa

Informea sortu

10 PARTIDA TXARRENAK

Txarrenak

Erakutsi erregistro

Bilatu:

Name	Puntuazioa	Fecha
Benito	1	2022-01-25
Benito	1	2022-01-25
Benito	1	2022-01-25
Xuinaga	0	2022-01-19
Lander	0	2022-02-04
Benito	0	2022-01-25
Benito	0	2022-01-19
Xuinaga	0	2022-01-19
Idoia	0	2022-02-04
Benito	0	2022-01-25

10 erregistrotik 1 - 10 ikusgai

Aurrekoa Hurrengoa

Informea sortu

LOGEATUTAKO LEHIAKIDE BATEN PARTIDAK

AsierBlazren Partidak

Erakutsi erregistro

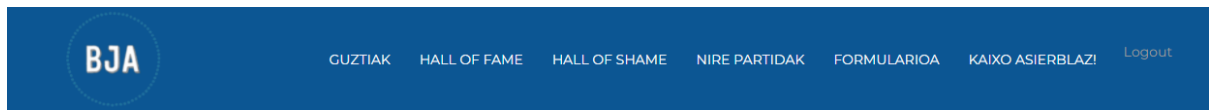
Bilatu:

Name	Puntuazioa	Fecha
AsierBlaz	75	2022-02-04
AsierBlaz	49	2022-02-04
AsierBlaz	12	2022-01-19
AsierBlaz	12	2022-01-25

4 erregistrotik 1 - 4 ikusgai

Aurrekoa Hurrengoa

FORMULARIOA



Zure iritzia

Gure jokuaeren iritzia garrantzitsua da guretzat.

Puntuazioa:



Zenbat ordu jolastu dituzu?

1h - 3h

Zer hobetuko zenuke?

Bidali Formularioa

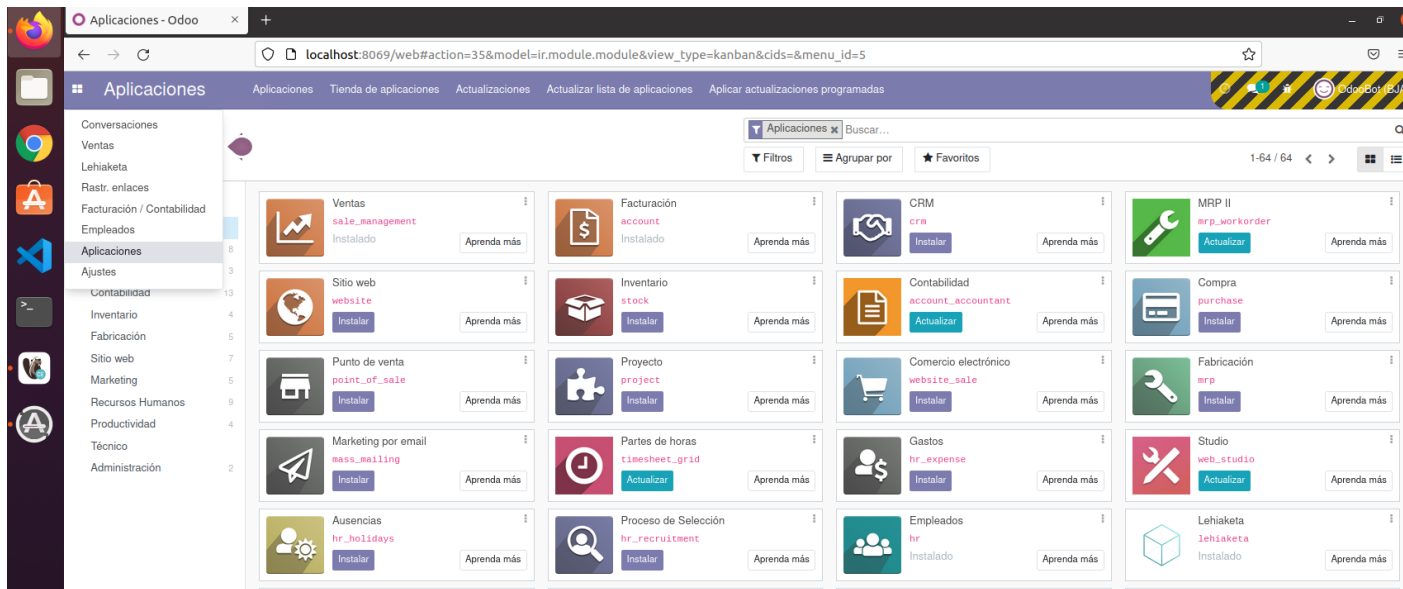
WEB orriari dagokion informazioa **DocFX** herreminta ibilita egin dugu, herreminta honek dokumentazio tekniko bat egiteko aukera ematen digu, gure kodigoan jarri ditugun iruzkinetaz baliatuta.

Dokumentazioa hurrengo **helbidean** aurkitu daiteke: **BJA\WebBja\docfx_site\api**

6.4. Enpresa Kudeaketa Sistema

Gure enpreak Odoo ERP-a erabiltzen du, eta ERP-ak ematen dizkigun aukerez baliatuta **Modulu berri bat sortu dugu** enpresako kideek KAFE OSASUNGARRIA jolasean lortutako puntuekin **lehiaketa** bat sortzeko.

Honetarako Ubuntu duen makina birtual bat montatu dugu. Makina honetan Odoo ERP-a eta PostgreSQL datu base instalatu ditugu.



Lehiaketeko modulura sartzen bagara langile guztiek jolastu dituzten partida guztiak agertzen dira, partida honetan lortutako puntuazioa eta zein egunetan jolastua izan den ikusi ahalko dugu.

Lehiaketa		Buscar...
Saikapena		Filtros Agrupar por Favoritos
<input type="checkbox"/> Izena	Puntuak ▲ Data	
<input type="checkbox"/> AsierBlaz	75 04/02/2022	
<input type="checkbox"/> Administrator	69 04/02/2022	
<input type="checkbox"/> Xuinaga	66 04/02/2022	
<input type="checkbox"/> Benito	66 25/01/2022	
<input type="checkbox"/> Lander	64 04/02/2022	
<input type="checkbox"/> Xuinaga	63 04/02/2022	
<input type="checkbox"/> Administrator	62 04/02/2022	
<input type="checkbox"/> Administrator	60 04/02/2022	
<input type="checkbox"/> Periko	59 04/02/2022	
<input type="checkbox"/> Lander	57 04/02/2022	
<input type="checkbox"/> Periko	52 26/01/2022	
<input type="checkbox"/> Administrator	52 04/02/2022	
<input type="checkbox"/> Lander	52 04/02/2022	
<input type="checkbox"/> Idoia	51 04/02/2022	
<input type="checkbox"/> Periko	51 26/01/2022	
<input type="checkbox"/> AsierBlaz	49 04/02/2022	
<input type="checkbox"/> Idoia	48 04/02/2022	
<input type="checkbox"/> Periko	43 04/02/2022	
<input type="checkbox"/> Periko	39 26/01/2022	
<input type="checkbox"/> Idoia	37 04/02/2022	
<input type="checkbox"/> Benito	36 04/02/2022	
<input type="checkbox"/> Xuinaga	36 04/02/2022	
<input type="checkbox"/> Benito	33 25/01/2022	

Honez gain **Iragazki** batzuk sartu ditugu lehiaketako moduluan, partidak elkartzeko edo **departamentu** edo **adin tarte** jakin bateko **jokalariek jolastutako partidak** errazago eta garbiago ikusi ahal izateko.

<div> <div>Buscar...</div> <div> <div>Filtros</div> <div>Agrupar por</div> <div>Favoritos</div> </div> </div>		
	Adineka filtratu	ak ▲ Data
	18-25 tartekoak	75 04/02/2022
	26-40 tartekoak	69 04/02/2022
	+41	66 04/02/2022
	Departamentuka filtratu	66 25/01/2022
		64 04/02/2022
	Administrazioa	63 04/02/2022
	Diseinua	62 04/02/2022
	Marketing	60 04/02/2022
	GGBB	59 04/02/2022
	Garbiketa	57 04/02/2022
	Añadir filtro personalizado	52 26/01/2022
		52 04/02/2022
		52 04/02/2022

6.4. Zerbitzu eta prozesuen programazioa

Zerbitzaria

```

public void run() {
    System.out.println("Zerbitzaria: HASI da.");
    try{
        System.out.println("Zerbitzaria: 12345 portuan entzuten...");
        Socket socketBezeroarekin = s; // Konexioaren zai geratu
        System.out.println("Zerbitzaria: socketBezeroarekin onartuta eta
sortuta, hari "+ hariZenbatzailea +" konektatuta dago. Mezuaren zai...");
        String str_mezua = "";
        InputStream is = socketBezeroarekin.getInputStream(); //
Bezeroarekin zabaldutako socket-etik irakurtzeko
        OutputStream os = socketBezeroarekin.getOutputStream();
    }
}

```

```
//Bezeroari idazteko zabaldutako fluxua
    DataInputStream dis = new DataInputStream(is); //InputStream-
arekin lan egiteko objektu bat
    DataOutputStream dout = new DataOutputStream(os);
    //OutputStream-arekin lan egiteko objektu bat
    str_mezua = (String) dis.readUTF();
//Mezu bat espero dugu eta irakurri egingo dugu
    JSONArray jsonArr = new JSONArray(str_mezua);
```

Bezeroa

```
public void bidali(){
    Thread t2 = new Thread(new Runnable() {
        @Override
        public void run() {
            try {
                Log.d("json",datosJson());
                Socket socketZerbitzareakin = new Socket("192.168.65.8", 12345);
// Zerbitzariarekin konektatzen saiatuko naiz
                // Socket socketZerbitzareakin = new Socket("127.0.0.1", 12345);
// Zerbitzariarekin konektatzen saiatuko naiz
                OutputStream os = socketZerbitzareakin.getOutputStream(); //
Zerbitzariari idazteko zabaldutako OutputStream
                DataOutputStream dout = new DataOutputStream(os);
// OutputStream-arekin lan egiteko objektu bat

                dout.writeUTF(datosJson()); //
Zerbitzariari bidalitako mezua
                dout.flush();
                dout.close();
                os.close();
                socketZerbitzareakin.close();
            } catch (IOException | JSONException e) {
                e.printStackTrace();
            }

        }
    });
    t2.start();
}
```

7. Etorkizunerako ildoak

Aplikazioa hobetzeko hainbat aldaketa egin daitezke behar den erabileraren arabera.

Lehenik eta behin, ez dira behar hainbeste datu base, Postgre eta Sql server behartuta gaude Odoo eta ASP.NET garatzeko, baina mongo datu basea ez da beharrezkoa.

Horren ondoren, android jokoaren mugimendua hobetu daiteke.

API-an ondo moldatuta dago behar den datuak jasotzeko.

Azkenik, web orrialdean logina diseño ezberdinak eta politagoa jarriko genuke. Gainera iruzkinak atala funtzionala da, baina js erabiliz asko hobetuko genuke.

8. Ondorioak

8.1. Ondorio teknikoak

8.1.1. Egin ditugun hobekuntzak

Erronka egiten joan garen einean, zenbait hobetzeko puntu geneuzkala ohartu gara.

Hasieran egiteko zeuden lan gehienak egiten hasi ginen, baina ikusi genuen lan batzuk egin ahal izateko aurretik beste batzuk egin behar genituela.

Modu honetan egin beharreko lanek zuten lehentasun zerrenda bat egin genuen, eta zuten lehentasunaren arabera lan hauek egiten hasi ginen, gure erabakia izan zen lan guztietan taldeko partaide guztiok parte hartzea, honela denok erronkako atal guztiak ulertu ahal izango genituelarik.

Erabaki hau egokia izan dela ikusi dugu proiektua amaitu dugun momentuan, gure buruak askoz prestatuago ikusten baititugu hurrengo erronkei begira.

8.2. Ondorio pertsonalak

Taldean lan egiten eta proiektua egiteko antolatzen ikasi dugu. Lana banatu eta elkarri lagundu diogu aplikazioaren funtzio eta atal guztiak egiteko. Oso aberasgarria iruditu zaigu talde bezala nola lan egin dugun.