

Euskal Herriko Unibertsitatea

Kudeaketaren eta Informazio Sistema Informatikaren Ingeniaritza

Informazio Sistemen Analisia eta Diseinua

31. taldea

Irakaslea: Mikel de Velasco

TETRIS JOKOA



BILBOKO
INGENIARITZA
ESKOLA
ESCUELA
DE INGENIERÍA
DE BILBAO

Aingeru Bellido, Sergio Martín, Maider Tato eta Maitane Urruela.

Bilbo, 2022ko abenduaren 18a

Aurkibidea

1	GitHub URL-a	2
2	Proiektuaren arkitektura	2
3	Erabilpen kasuen eredua	3
4	Domeinu eredua	5
5	Sekuentzia diagrama	7
6	Implementazioaren xehetasunak	9

1 GitHub URL-a

Hona hemen gure proiektuaren errepositorioaren web helbidea:

https://github.com/Sergiom8m/ISAD22_tetris

2 Proiektuaren arkitektura

Proiektu hau Python programazio lengoaia eta hainbat liburutegi erabiliz garatu da. Liburutegi aipagarrienak hurrengoak izan dira:

- Tkinter: interfaze grafikoeterako.
- Playsound: jokoan soinuak inplementatzeko.
- Pygame: jokoan musika inplementatzeko.

Proiektua kudeatzeko 4 partaide izanda taldean eta guztiok aldi berean lan egin ahal izateko besteen lana zapaldu barik, Git erabili dugu. GitHub plataforman errepositorio bat sortu dugu irakaslearen errepositoriotik abiatuz (fork bat eginez). Bertan, lankide bakoitzak adar bat du bere lan pertsonala egin ahal izateko eta komuna den beste adar bat dago (master), non guztion lana komunean jartzen den. Git erabiltzeari esker bertsioen kontrola egitea asko errazten da, erroreren bat gertatzekotan aurreko bertsioa erraz berreskuratzea posible baita.

Erabiltzaileen datuak, partidak eta horiek ezarritako pertsonalizazio aukerak gordetzeko, SQLite datu base kudeaketa sistema erabili da. Datu base kudeaketa sistema honek, fitxategi batean gordetzen du datu base bakoitza eta horrela ez da behar IP eta ataka bidezko konexiorik.

Kodea antolatzeko, MVC arkitektura patroia erabili dugu, kodea eskalagarriagoa eta ulergarriagoa eginez. Horrela, bistak memorian kargatuta dauden instantziei funtzionalitateak eskatzen dizkiete, hauek haien funtzionalitateak eskainiz edo kanpoko baliabide bat erabiliz, hala nola, datu basea edota fitxategien karga.

3 Erabilpen kasuen eredua

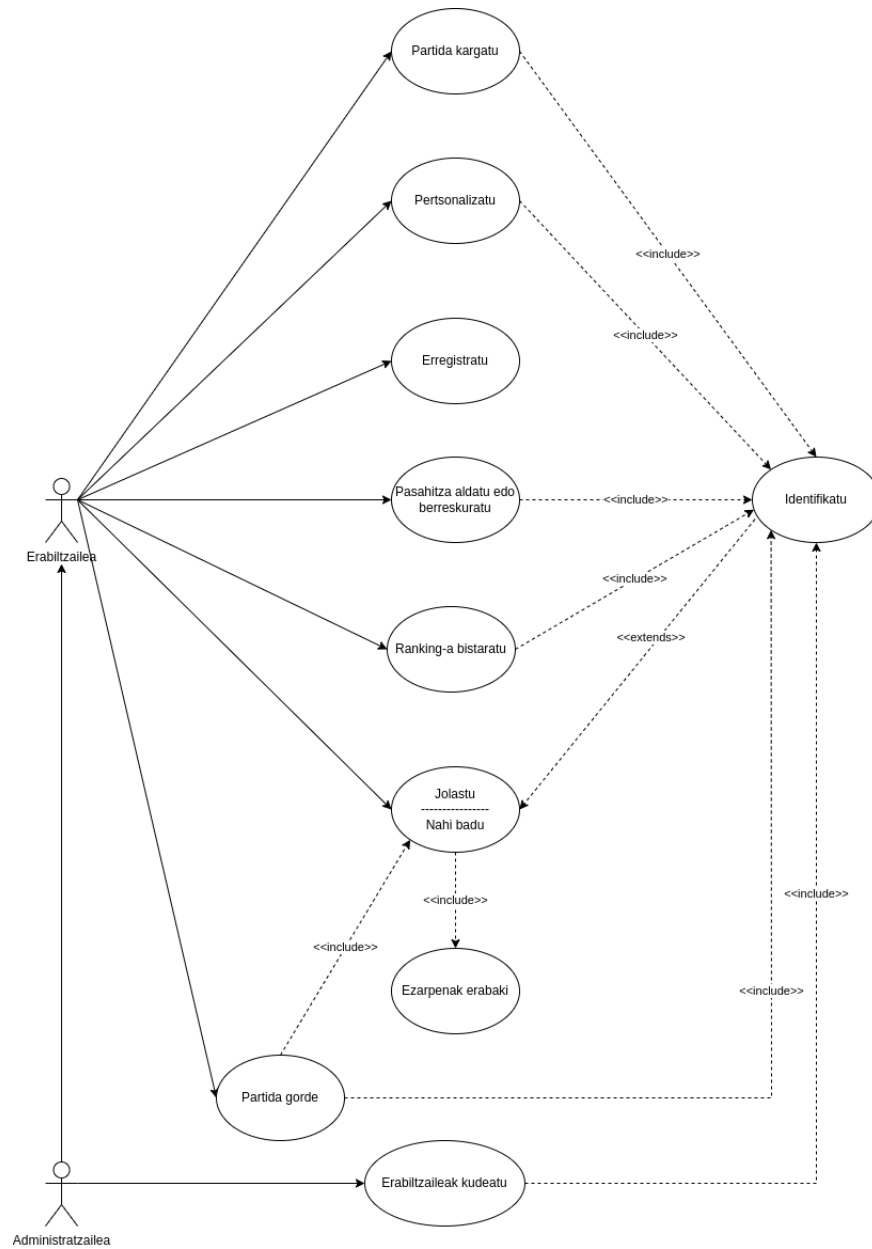
Erabilpen kasuen ereduaren bitartez egindako programak dituen funtzionalitateak erakusten dira. Horrekin batera, funtzionalitateen arteko dependentziak irudikatuta geratzen dira EKE diagraman.

Gure diseinuan bi erabiltzaile mota daude. Alde batetik, erabiltzaile arrunta, identifikatuta egon daitekeena edo ez, eta bestetik, administratzailea. Administratzaileak erabiltzaile arruntak egin ditzakeen akzio guztiak burutu ditzake. Horrez gain, erabiltzaile arruntak kudeatu ahal ditu. Administratzaile baimenak izateko beharrezkoa da administratzaile kontu bat izatea eta saioa hastea (erabiltzaile izena “admin” izanda eta pasahitza “123”).

Erabiltzaileek identifikatu gabe egin dezaketen gauza bakarra partidak jolastea da. Partida bat jolastu aurretik, nahitaezkoa da partida berriaren ezarpenak zehaztea. Horrez gain, erregistratzeko aukera ematen da, geroago saio hasi ahal izateko eta gainontzeko funtzionalitateei sarbidea lortuz.

Saioa hastekotan funtzionalitate dezente gaitzen zaizkio erabiltzaileari. Horien artean pertsonalizazio aukerak daude: profilaren koloreak, partidaren adreiluen koloreak eta partidaren musika aldatzeko aukera. Aukera hauek profilarekin batera gordeko dira eta berriro saioa hastean defektuz agertuko dira. Horrez gain, partidak gordetzeko aukera emango zaio erabiltzaileari erregistratuta badago. Hau da, partida bat erdizka uztea posible izango da eta etorkizunean partida hori kargatzea jolasten jarraitu ahal izateko. Gainera, jokoak automatikoki eraikitzen dituen ranking-ak eta sariak ikusteko aukera emango zaie erregistratutako erabiltzaileei.

Azkenik, erabiltzailearen batek pasahitza aldatu nahi izatekotan hori egiteko aukera ematen zaio eta pasahitza ahaztekotan hori berreskuratzeko prozesua ezarrita dauka programak. Erregistratzerakoan galdera bat eta horri dagokion erantzuna jarri behar ditu erabiltzaileak, beraz, pasahitza ahaztekotan galdera hori formulatuko zaio, eta erantzuna zuzena bada pasahitza aldatzeko aukera emango zaio.



Irudia 1: Egindako diseinuari dagokion EKE

4 Domeinu eredua

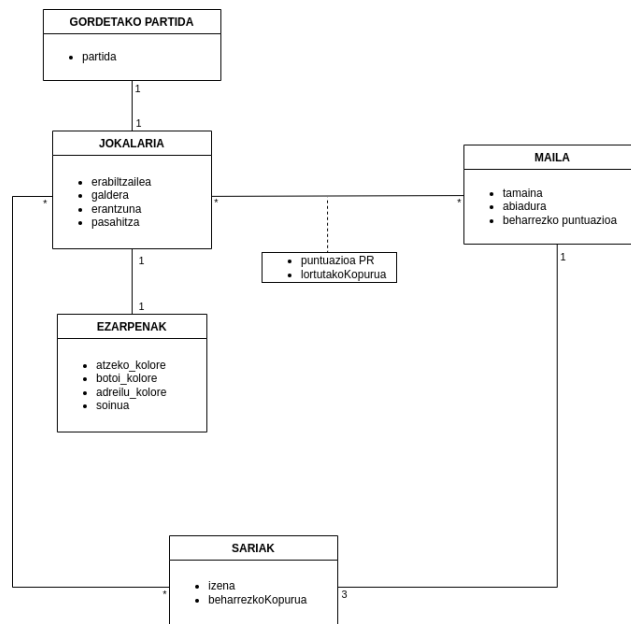
Domeinu ereduaren bidez ikus dezakegu objektu motak eta beraien erlazioak zeintzuk diren sistemaren testuinguruan, edo beste era batera esanda, programaren datu basearen egitura nolakoa den. Domeinu honetan ikusten den moduan, bost entitate ditugu: “Jokalaria”, “Gordetako partida”, “Ezarpenak”, “Maila” eta “Sariak”.

“Jokalaria” entitatea gordeko ditu erabiltzaileak log-in pausuak egiteko beharrezkoak dituen atributuak (erabiltzaile izena, pasahitza, berreskurapen galdera eta erantzuna).

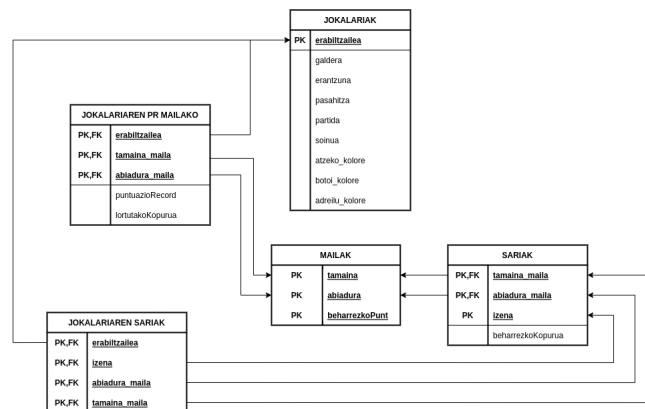
Bestetik, honek bi erlazio ditu 1-1 erlazioak direnak, “Gordetako partida” eta “Ezarpenak” entitateekin. Hauek, azkenengo gordetako partida eta jokalaria esleiturik dituen kolore eta soinu konfigurazioa gordeko dituzte, hurrenez hurren. Atributu hauek tauletara pasatzean “Jokalaria” entitatearen barruan agertuko dira 1-1 erlazioengatik.

Bestetik, “Jokalaria” entitatea beste bi entitateekin erlazionaturik dago *-* erlazioen bidez: “Maila” eta “Sariak”. Lehenak 16 maila ezberdin gordeko ditu 4 tamaina eta 4 abiadura ezberdinen konbinaketa guztiak baititu, hauetan behar den irabazteko puntuazioa ere gordez. Erlazio honen barruan beste bi atributu egongo dira: jokalaria bakoitzak mailako duen puntuazioa eta irabazitako aldi kopurua.

“Sariak” aldiz “Maila” entitatearekin 1-3 erlazioa du, izan ere, 16 maila horietako bakoitzeko 3 sari egongo dira. Azken hauetatik izena eta mailan irabazi beharreko aldi kopurua saria lortzeko gordeko dira.



Irudia 2: Egindako diseinuari dagokion DE



Irudia 3: Egindako diseinuari dagokion DB-ren taulak

5 Sekuentzia diagrama

Sekuentzia diagrama honetan azaltzen da jada erregistratua dagoen erabiltzaile batek identifikatu eta maila bateko ranking-a ikusteko egin behar duen prozesua.

Horretarako, “Hasierako menua” interfazetik hasita “Identifikatu” interfazera joango da botoi bat sakatuz. Bertan, bere ID eta pasahitza sartuko ditu eta “Sartu” sakatuko du. Hau egiterakoan, ID hori duen erabiltzailea bilatuko da jokalaria zerrendan.

Hau egiteko, eta MVC patroia jarraituz, DB-arekin erlazionaturik dituen funtzioak dituen “Jokalaria zerrenda” interfazea deituko dugu ID-a pasatuz eta honek DB-ari kontsultak egiteko interfazeari eskatuko dio ID horren informazio guztia lortzea. Azkenengo hau, “DbConn” da eta honek ResultSet motatako emaitza bat lortuko du kontsulta honen bidez:

```
SELECT *  
FROM JOKALARIAK  
WHERE erabiltzailea = '<ID>'
```

Kontsultak bueltatzen duen erabiltzailearen datuekin jokalaria berri bat sortuko da, hau izanda beste interfazei pasatuko zaien parametroa saio osoan zehar, hau “Jokalarien zerrenda” barruan egongo da. “Jokalarien zerrenda” programa abiarazten den bakoitzean erreseteatzen da, beraz, nahiz eta erregistratuta egon, erabiltzailearen datu horiek DB-an daude eta ez programan bertan.

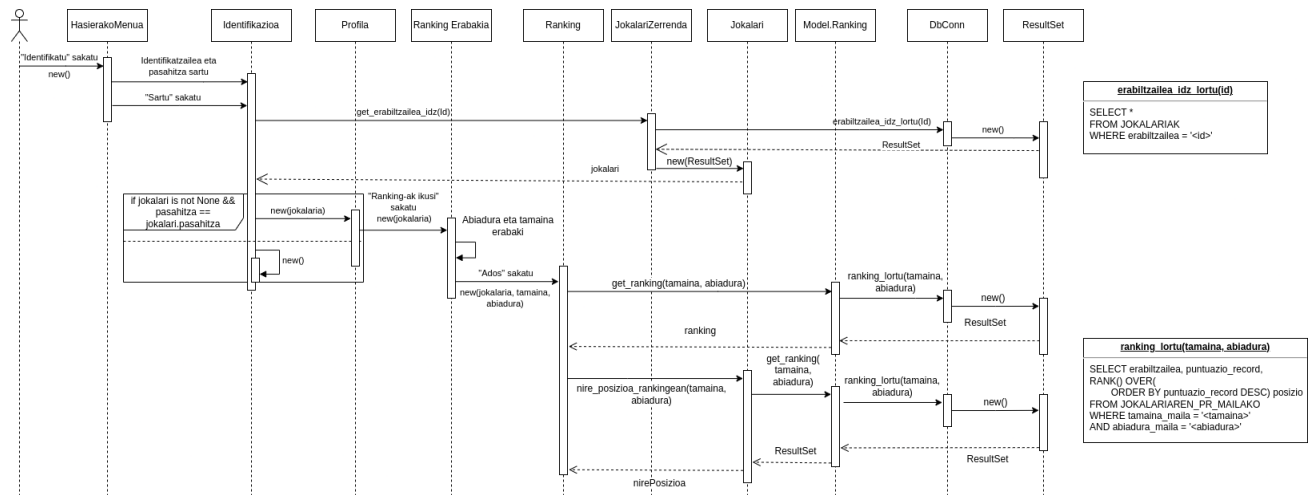
Erabiltzailea DB-an erregistratuta badago eta sartutako pasahitza egokia bada, orduan bere “Profila” agertuko da, aurretik lortu dugun jokalaria informazioarekin pertsonalizazioa burutuz. Aldiz, pasahitza ez egokia bada berriro eskatuko zaio erabiltzaileari pasahitza sartzeko.

Hemendik ranking bat ikusteko, “Ranking-ak ikusi” botoiaren bidez maila aukeratzeko menu batera joango gara (momentu oro erabiltzailea interfazeen artean bidaliz). Behin mailaren abia-

dura eta tamaina erabakita eta “Ados” botoia sakatuta, modeloko ranking-ari tamaina eta abiadura zehatz batzuentzako ranking-ak lortzea eskatuko zaio. Honek “DbConn” -en bitartez egiten du ondorengo kontsulta:

```
SELECT erabiltzailea, puntuazio_record,
RANK() OVER (
ORDER BY puntuazio_record DESC) posizio
FROM JOKALARIAREN_PR_MAILAKO
WHERE tamaina_maila = '<tamaina>'
AND abiadura_maila = '<abiadura>'
```

Azkenik, ranking-a bistaratu nahi duen jokalariairen posizioa ranking-ean zein den kontsultatzen da. Aurreko atalean azaldu den antzeko prozedura jarraitzen da. Ranking-ak jokalariai ranking-ean duen posizioa eskatzen dio, jokalariai tamaina eta abiadura zehatz batzuei dagokien ranking-a eskatzen dio modeloko ranking-ari. Honek “DbConn” -en bitartez gainera kontsulta egiten du DB-an. Ranking-ak dituenean, jokalariai iterazioen bitartez bere posizioa zein den bilatzen du eta bistaratzen da.



Irudia 4: Ranking-ak bistartzeko sekuentzia diagrama

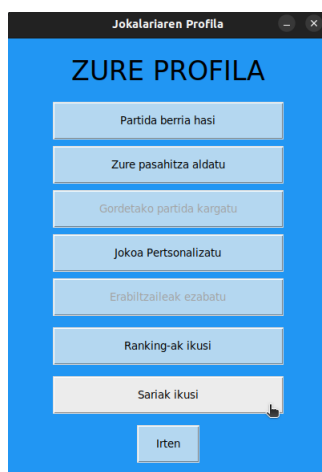
6 Implementazioaren xehetasunak

Ikus daitekenez, garatutako programa ohiko tetris jokoan oinarrituta dago, alabaina, joko tradizionalaren gainean funtzionalitate eta aukera berriak inplementatu dira.

Lehenik eta behin jokalarien erregistroa eta identifikazioa gehitu da, erabiltzaileek euren aurre-rapena eta konfigurazioa gorde ditzaten. Pertsonalizazio ezarpen horien artean partida bakoitzeko adreiluen kolorea, abiadura, tableroaren tamaina, musika eta zenbait gauza gehiago aldatzeko aukera dago.

Jokoaren pertsonalizazioan, erabiltzaile bakoitzak aukeratutako gauzetaz gain, piezen mugimenduetan ere soinuak sartu izan dira. Hauek edozein jokalaria entzun dezake piezak biratzen eta tableroaren amaierara ailegatzen den bakoitzean. Gainera, lerro bat ezabatzen den bakoitzean ere soinu bat entzungo da.

Baita ere, koloreei dagokienez, interfazeen koloreak aldatzean, horiekin batera, botoien kolorea aldatzea erabaki zen, modu honetan koloreen arteko konbinazioak erabiltzailerarentzat erakargarriagoak izan daitezen.



Irudia 5: Koloreen pertsonalizazioa

Jokoa pertsonalizatzeaz gain, baita ere ranking-ak ikusteko aukera sartu izan dugu. Esate baterako, lehen aipatu izan ditugun maila bakoitzean, puntuazio altuena duten lehen hiru jokalariai aterako dira eta behean gure puntuazio eta ranking posizioa agertuko dira. Ranking orokor bat ere egongo da, non posizio metatua gordeko den jokalaria bakoitzeko (maila guztietan izandako partida guztien puntuazioak batuz).

Azkenik, maila bakoitzeko ditugun sariak ikusteko aukera izango dugu jokoan. Hauek oraindik lortu ez baditugu “X” baten bidez adieraziko dugu saria. Lortuta badugu berriz, sariaren arabera ezkutu bat ala beste lortuko dugu, hurrengoko irudian ikusten den moduan.

Sariak						
MAILA	BEHARREZKO P	1 SARIA	2 SARIA	3 SARIA		
10x800	100	2 	4 	6 		
10x400	50	2 X	4 X	6 X	X	
10x200	30	2 X	4 X	6 X	X	
10x100	10	2 	4 X	6 X	X	
20x800	100	2 X	4 X	6 X	X	
20x400	50	2 X	4 X	6 X	X	
20x200	30	2 X	4 X	6 X	X	
20x100	10	2 	4 	6 X	X	

Irudia 6: Lortutako sariak