

TREBALL FINAL DE GRAU

Anàlisi i demostració de la connexió d'una base de dades amb Unity

Oriol Ferrer Queralt
Tutora: Ester Bernadó Mansilla
Grau en Disseny i Producció de Videojocs

CURS 2020-21



Centre adscrit a la



Abstract

The objective of this project is to analyze and connect a database located on an external server with a prototype video game developed with the Unity game engine. An analysis will be made of all the existing databases to find out their properties, strengths and weaknesses, in order to determine which one to use. The results of the analysis will also be collected in order to specify which database is the most suitable to use according to the video game to be developed.

Resum

L'objectiu d'aquest projecte és analitzar i realitzar la connexió d'una base de dades situada en un servidor extern amb un videojoc en fase prototip desenvolupat amb el motor de jocs Unity. De totes les bases de dades existents es realitzarà una anàlisi per conèixer les seves propietats, punts forts i febles, per poder determinar quina fer servir. També es recolliran els resultats de l'anàlisi per precisar quina base de dades és la més adequada a fer servir segons el videojoc que es vulgui desenvolupar.

Resumen

El objetivo de este proyecto es analizar y realizar la conexión de una base de datos situada en un servidor externo con un videojuego en fase de desarrollo con el motor de juegos Unity. De todas las bases de datos existentes se realizará un análisis para conocer sus propiedades, puntos fuertes y puntos débiles, para poder determinar cual usar. También se recogerán los resultados del análisis para precisar cuál sería la base de datos más adecuada a usar según el videojuego que se quiera desarrollar.

Índex

Índex de figures	V
Índex de taules	VII
1. Introducció	1
2. Marc teòric.....	5
2.1 Què és una base de dades?	5
2.2 Què és un sistema de gestió de bases de dades?	5
2.3 Tipus de models d'informació	6
2.4 Bases de dades relacionals	9
2.5 Dades estructurades i no estructurades	12
2.6 Bases de dades no relacionals.....	14
2.6.1 Bases de dades basades en documents de dades	15
2.6.2 Bases de dades clau-valor	16
2.6.3 Bases de dades orientades en columnes.....	17
2.6.4 Bases de dades basades en gràfics	18
2.7 Motors de bases de dades	19
2.8 Les bases de dades com a eina de l'actualitat	21
2.9 El motor de jocs Unity	23
2.10 Les bases de dades en videojocs	24
2.10.1 Gèneres	24

2.10.2 Elements dels videojocs.....	27
2.10.3 Elements de les bases de dades	29
2.11 Resum	31
3. Anàlisi de referents	33
3.1 Bases de Dades.....	33
3.1.1 MySQL.....	33
3.1.2 Oracle	35
3.1.3 MongoDB.....	36
3.2 Prototip	37
3.3 Resum	43
4. Objectius	45
5. Disseny metodològic i cronograma	47
5.1 Metodologia	47
5.1.1 Context	47
5.1.2 Recerca de les bases de dades	47
5.1.3 Anàlisi sobre les bases de dades i els videojocs	47
5.1.4 Desenvolupament del prototip.....	47
5.2 Cronograma	48
6. Disseny del prototip	51
6.1 Disseny del videojoc	51
6.2 Navegació	52

6.3 Casos d'ús	53
6.4 Diagrama de classes.....	55
6.5 Disseny de la base de dades	61
6.6 Connexió amb la base de dades	63
6.7 Resum	65
7. Resultats	67
7.1 Resultats de la investigació.....	67
7.1.1 Anàlisi de les bases de dades.....	67
7.1.2 Anàlisi dels videojocs	72
7.1.3 Les bases de dades més adequades	76
7.2 Resultats del prototip	81
8. Conclusions.....	91
9. Bibliografia/Referències	95

Índex de figures

Figura 1. Exemple d'un model d'informació	7
Figura 2. Exemple d'un model d'informació jeràrquic	7
Figura 3. Exemple d'un model relacional	8
Figura 4. Exemple d'un model E-R	9
Figura 5. Exemple d'una taula	10
Figura 6. Exemple d'una base de dades relacional	11
Figura 7. Aportació visual de les dades estructurades en comparació amb les no estructurades	12
Figura 8. Exemple d'una base de dades basada en documents.....	16
Figura 9. Exemple d'una base de dades clau – valor.	17
Figura 10. Exemple d'una base de dades orientada en columnes.....	17
Figura 11. Exemple d'una base de dades basada en gràfics.....	18
Figura 12. Classificació de les bases de dades més populars del desembre de 2020	19
Figura 13: Imatge del menú principal de <i>Chess Royale King</i>	38
Figura 14. Imatge del mode de joc Jugador contra IA del videojoc <i>Chess Royale King</i>	39
Figura 15. Imatge del mode de joc Puzzle del videojoc <i>Chess Royale King</i>	40
Figura 16. Imatge del menú principal del videojoc <i>Word Crack</i>	41
Figura 17. Imatge d'una partida del videojoc <i>Word Crack</i>	42

Figura 18. Diagrama de navegació.....	52
Figura 19. Diagrama de casos d'ús segons l'accés	53
Figura 20. Diagrama de casos d'ús segons la jugabilitat	54
Figura 21. Diagrama de casos d'ús segons les dades	55
Figura 22. Diagrama de classes de les dades d'usuari.....	56
Figura 23. Diagrama de classes de la part jugable.....	58
Figura 24. Diagrama de classes de la connexió amb la base de dades	60
Figura 25. Diagrama de classes de la resta de les classes	61
Figura 26. Diagrama de la base de dades.....	62
Figura 27. Splash Screen del prototip	82
Figura 28. Finestra de registre del prototip	83
Figura 29. Finestra de l'inici de sessió del prototip	84
Figura 30. Finestra del menú principal del prototip	85
Figura 31. Finestra de la part jugable del prototip.....	86
Figura 32. Finestra del perfil del prototip	87
Figura 33. Finestra de l'historial del prototip	88

Índex de taules

Taula 1. Taula d'elements d'un videojoc rellevants per una base de dades	30
Taula 2. Diagrama de Gantt de les fases del projecte	49
Taula 3. Taula de dates de les fases del projecte.....	49
Taula 4. Característiques de les bases de dades. Primera part.....	69
Taula 5. Característiques de les bases de dades. Segona part.....	70
Taula 6. Característiques dels gèneres de videojocs. Primera part	73
Taula 7. Característiques dels gèneres de videojocs. Segona part	74
Taula 8. Característiques dels tipus de videojocs. Primera part	75
Taula 9. Característiques dels tipus de videojocs. Segona part.....	76
Taula 10. Relació de les característiques de les bases de dades i els gèneres de videojocs	77
Taula 11. Relació dels gèneres de videojocs amb les bases de dades escollides	78

1. Introducció

Les bases de dades o BBDD són elements que es troben pràcticament en la majoria de videojocs actuals. No és necessari que el videojoc contingui una jugabilitat multijugador, però només que es faci ús d'una taula de puntuacions entre diversos jugadors ja ha d'existir una base de dades. Per tant, tot videojoc que vulgui compartir dades entre diversos dispositius ha d'estar preparat per implementar-ne una.

Segons Berg i Seymour (2012) l'origen de les bases de dades comença amb la necessitat d'emmagatzemar totes aquelles dades i informació que fins el moment es trobaven en llibres i informes governamentals, mèdics, empresarials, etc. El 1960 els ordinadors van guanyar popularitat i van millorar en la relació entre cost i eficiència, pel que empreses privades van optar per fer-ne servir i incrementar la seva capacitat d'emmagatzemar les dades. A partir d'aquest punt es va incrementar l'ús d'ordinadors i l'emmagatzematge digital.

Per altra banda, els primers videojocs no disposaven d'una gran trama, pel que la seva duració era curta i no es guardava el seu progrés. Quan els ordinadors ja van ser accessibles pel consumidor mig, es va obrir la possibilitat de disposar d'un ordinador d'ús privat per a les famílies. Primer de tot, es va fer ús dels disquets i més endavant dels discs durs, on cada vegada millorava la qualitat a l'hora d'emmagatzemar les dades: velocitat i quantitat d'espai d'emmagatzematge. Per part de les consoles de videojocs, la primera va ser l'Atari 2600 amb els cartutxos com a espai d'emmagatzematge. A partir del 2005 les consoles van començar a implementar característiques que feien ús de la connexió a la xarxa i el 2011 ja era comú que s'utilitzessin mecanismes per guardar dades al núvol (Christian, 2019).

L'ús del núvol en els videojocs implica una connexió a un servidor extern on emmagatzemar la informació en una base de dades, per tant, la connexió a la xarxa és imprescindible per poder realitzar la tasca. Les BBDD aporten una expansió en les funcionalitats dels videojocs.

Com a exemple, s'il·lustra la diferència que hi ha entre una màquina recreativa i un videojoc de mòbil, concretament el videojoc *Tetris*. Aquest joc conté una taula de puntuacions que es mostra al final de la partida per generar competitivitat entre els jugadors: "qui pot aconseguir la màxima puntuació". En el primer cas, en la màquina recreativa, els usuaris van passant pel videojoc i hi van deixant la seva puntuació. Per tant les dades es guarden de forma local en la pròpia màquina i només es troben en la mateixa. Per altra banda, el segon cas mencionat del *Tetris* en un videojoc mòbil, qualsevol usuari que hi vulgui jugar se'l pot descarregar, jugar-hi i aportar-hi la seva puntuació. Aquestes puntuacions es poden guardar en una base de dades, permetent que tots els dispositius amb el videojoc les puguin consultar i fer-ne ús.

Les bases de dades també es fan servir en grans conceptes com en Big Data. Com expliquen Yu i Guo (2016), Big Data fa referència a l'anàlisi i extracció d'informació de grans volums de dades. Com per exemple, en xarxes socials es fa servir per recollir les dades dels seus usuaris, analitza-les i poder oferir contingut i anuncis personalitzats per a cada un. Segons Abhinandan i Bandyopadhyay (2016), Big Data fa servir el concepte de les 3Vs, les tres V representen volum, varietat i velocitat, també s'han analitzat altres conceptes que porten fins a 4 o 5 V. El volum mesura la quantitat d'informació disponible per una organització, la qual no ha de ser propietària necessàriament sempre i quan hi tingui accés. La varietat mesura la riquesa en la representació de les dades. I la velocitat indica el temps en que es mesuren totes les dades.

Les necessitats en els camps tecnològics com la quantitat de dades a guardar, la seva disponibilitat i la diversitat de les dades ha generat que sorgeixin moltes bases de dades diferents, cada una amb les seves particularitats. Algunes de les quals estan enfocades a les dades estructurades, com la informació dels usuaris d'un banc o els informes mèdics de pacients. També hi han les dades no estructurades com és la informació d'una imatge, els textos d'una pàgina web. O també dades sense una clara estructura com pot ser una xarxa social en sí. Algunes d'elles estan més enfocades a suportar grans quantitats de dades, mentre que d'altres busquen ser més ràpides i eficients.

Davant d'aquesta quantitat d'opcions de bases de dades en el mercat existent, s'ha de realitzar una recerca per poder conèixer quina base de dades pot satisfer millor les necessitats d'un projecte. Aquesta és una tasca que no només es limita en el camp de videojocs.

Per tant, una primera motivació del projecte és fer la conceptualització de les diferents bases de dades per videojocs, les seves aplicacions i estudiar quina BBDD és més adequada en funció de les necessitats del joc. Cada videojoc és diferent. Existeixen classificacions de jocs segons el seu gènere, com per exemple el proposat per Pavlovic (2020), tipus de jugador objectiu, nombre de jugadors, etc. Però quina seria la base de dades més adequada per cada cas? Per exemple, si el videojoc és d'un sol jugador no seria necessari fer ús d'una base de dades. Però si es vol implementar una taula de puntuacions o realitzar el guardat de la partida, tot i no ser el videojoc multijugador, seria necessari l'ús d'una BBDD. En canvi, si el videojoc contempla fer ús d'un sistema d'usuaris per un entorn multijugador la base de dades és imprescindible.

A partir de l'anàlisi a realitzar de les bases de dades existents escollides es determinarà la més adequada per realitzar una connexió amb el motor de jocs Unity, per demostrar com es duu a terme. Per realitzar la connexió també caldrà desenvolupar un videojoc en fase prototip per demostrar que la connexió funciona i fa ús de les funcionalitats de la base de dades. És per això que, durant l'anàlisi, caldrà conèixer a fons el funcionament d'una base de dades i què pot aportar en un videojoc.

En aquest document s'explica el desenvolupament del projecte, estructurat en els següents capítols. En el capítol 2 s'analitza el marc teòric relacionat amb les bases de dades, els videojocs i el motor de jocs Unity. A continuació, el capítol 3 realitza una anàlisi dels referents actuals, on es recullen dades d'altres videojocs i com fan ús de la seva base de dades. Tot seguit, en el capítol 4, s'expliquen els objectius del projecte. El capítol 5 explica el procediment en que es realitza el projecte, des de l'anàlisi de les bases de dades i els videojocs fins al desenvolupament del prototip. El capítol 6 exposa el disseny del prototip on s'explica cada apartat en profunditat. A continuació, en el capítol 7 s'exposen els resultats obtinguts, així com l'assoliment dels objectius establerts. I finalment, en el capítol 8 es recullen les conclusions.

2. Marc teòric

En aquest capítol s'estableix el marc teòric sobre el que es realitza el projecte. S'exposen les bases de dades, explicant què són, els seus tipus i característiques. A més a més, també s'estableix el marc teòric referent als videojoc i al motor de jocs Unity. Els videojocs en l'actualitat abarquen una gran quantitat de camps, aquest marc teòric es troba enfocat en les bases de dades.

2.1 Què és una base de dades?

Com expliquen Sharma, i altres (2010), les bases de dades són repositoris de dades, dissenyades per recolzar de forma eficient l'emmagatzematge, recuperació i manteniment de les dades. Existeixen diferents tipus de BBDD per ajustar-se a les necessitats de la indústria.

Per altra banda, Ceri, Gottlob, i Tanca (1990) defineixen les bases de dades com a sistemes que gestionen grans quantitats de dades i encarregats d'aportar la tecnologia suficient per recollir i actualitzar dades persistents. Aquestes bases de dades funcionen amb "Queries", que consisteixen en comandes per extreure la informació concreta que es vol de la BBDD. Un element que cal destacar en el funcionament d'una base de dades és el concepte de "Constraint", consisteix en restriccions que especifiquen les condicions per realitzar un bon procediment a l'hora de manipular la base de dades, és a dir, evitar emmagatzemar dades incorrectes.

2.2 Què és un sistema de gestió de bases de dades?

Un sistema de gestió de bases de dades (SGBD) és un o més softwares encarregats de permetre als usuaris gestionar les dades emmagatzemades en una base de dades, concretament, accedir, organitzar, emmagatzemar, administrar, recuperar i mantenir les dades (Sharma, i altres, 2010). MySQL i Oracle són exemples de SGBD.

Un exemple de comanda en el SGBD MySQL és: “SELECT name FROM User WHERE age > 18”. En aquesta comanda es recullen tots els noms de la taula d'usuaris d'aquelles persones que tinguin més de divuit anys.

El concepte ACID fa referència a un seguit de propietats que ha de complir un sistema de gestió de bases de dades per assegurar que les seves comandes són segures. ACID és l'acrònim de “Atomicity, Consistency, Isolation, Durability”. Atomicitat vol dir que cada consulta ha de finalitzar correctament i no deixar cap rastre. Consistència indica que tota operació ha de funcionar sense pertorbar el correcte funcionament de la base de dades. Aïllament assegura que una consulta no pot afectar-ne d'altres o elements amb els que no hi està directament relacionat. I finalment, Durabilitat indica que la operació s'ha de guardar un cop feta, inclús si el servei falla s'ha d'assegurar que la operació s'ha realitzat (Date, 2011).

2.3 Tipus de models d'informació

Sharma, i altres (2010) presenten la següent classificació basada en els models d'informació:

- Models “Network”
- Model jeràrquic
- Model relacional
- Model entitat – relació
- Model objecte - relació

Primer de tot, el model de dades “Network” classifica la informació en sistema de pares - fills connectats entre si formant una xarxa.

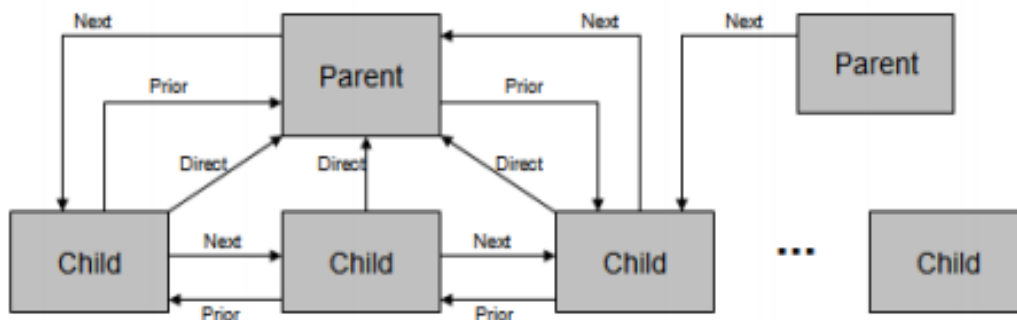


Figura 1. Exemple d'un model d'informació. Font: (Sharma i altres, 2010).

Com es mostra a la Figura 1, el tipus de registre es representa en rectangles. Una col·lecció d'aquests registres s'anomena "CODASYL network". La relació entre registres es mostra en les fletxes, que indiquen el tipus de relació i la direcció.

Tot seguit, es presenta el model d'informació jeràrquic el qual organitza la seva informació utilitzant una estructura d'arbre. L'arrel de l'arbre és el pare seguit dels seus fills en forma de nodes. En el cas de la Figura 2, el pare és el President.

Una col·lecció de camps amb les seves dades relacionades s'anomena "record type". Cada "record type" està forçat a seguir les instruccions de la descripció de les dades indicada en la seva definició.

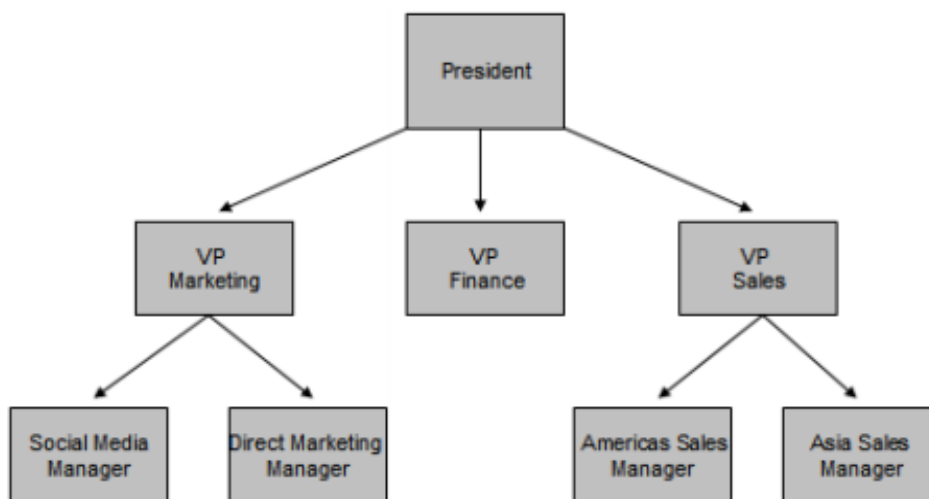


Figura 2. Exemple d'un model d'informació jeràrquic. Font: (Sharma i altres, 2010).

El següent model d'informació és el relacional. Està construït en una base matemàtica sòlida, i és la més utilitzada en bases de dades avui dia. La proposta d'aquest model

és emmagatzemar les dades d'una forma més senzilla (en taules), accedir a la informació en temps real i que les dades siguin independents del punt físic on es guardi.

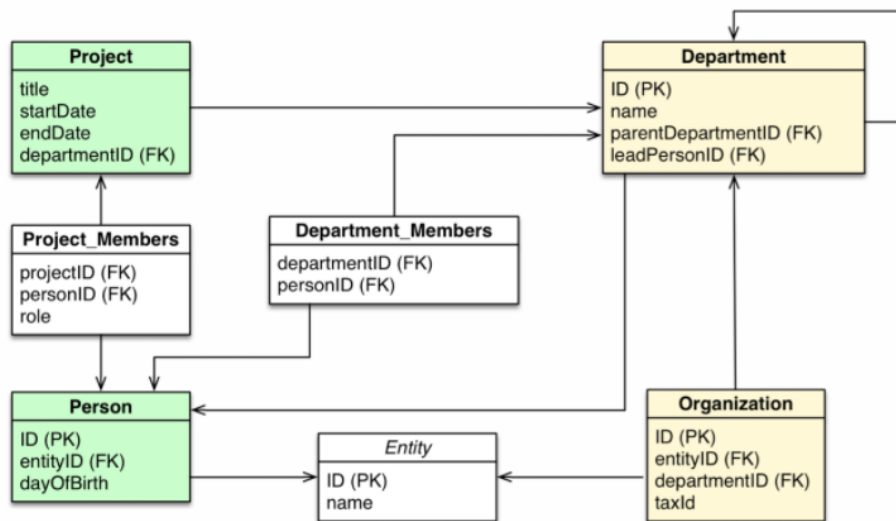


Figura 3. Exemple d'un model relacional. Font: (Hunger, Boyd, i Lyon, 2016).

El model entitat - relació consisteix en pensar en una base de dades com una col·lecció d'instàncies d'entitats, les quals han d'estar basades en les nocions del món real. Les entitats són objectes que tenen una existència independent d'altres entitats dins de la base de dades, les quals tenen atributs que les defineixen. Un o més d'aquests atributs pot ser designat com a clau que es farà servir com a identificador del conjunt d'informació. Les entitats es relacionen entre elles 1-1, 1-n o n-n, on n indica més de una.

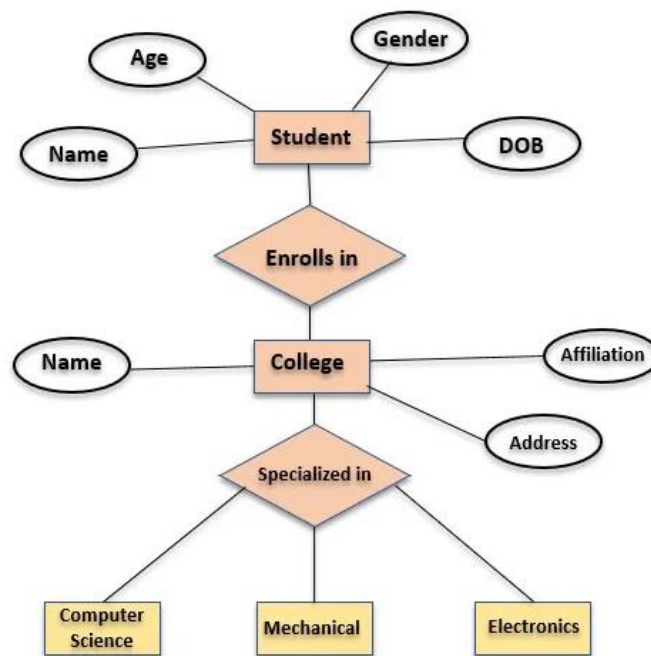


Figura 4. Exemple d'un model E-R. Font: (Gupta, s. d.).

El model entitat - relació pot semblar similar al model relacional. Es diferencien en que el model E-R tracta amb entitats, definint quina és la clau primària de cada entitat i s'encarrega d'especificar el model de forma conceptual. La relació entre entitats la indica però no s'especifica quina és la clau externa o com s'implementa la relació. En canvi, el model relacional és la implementació del model. S'especifica el format dels atributs, les claus externes i la relació exacta entre les taules. La estructura que presenta el model relacional està pensada de tal forma que es pot implementar directament com la estructura de la base de dades.

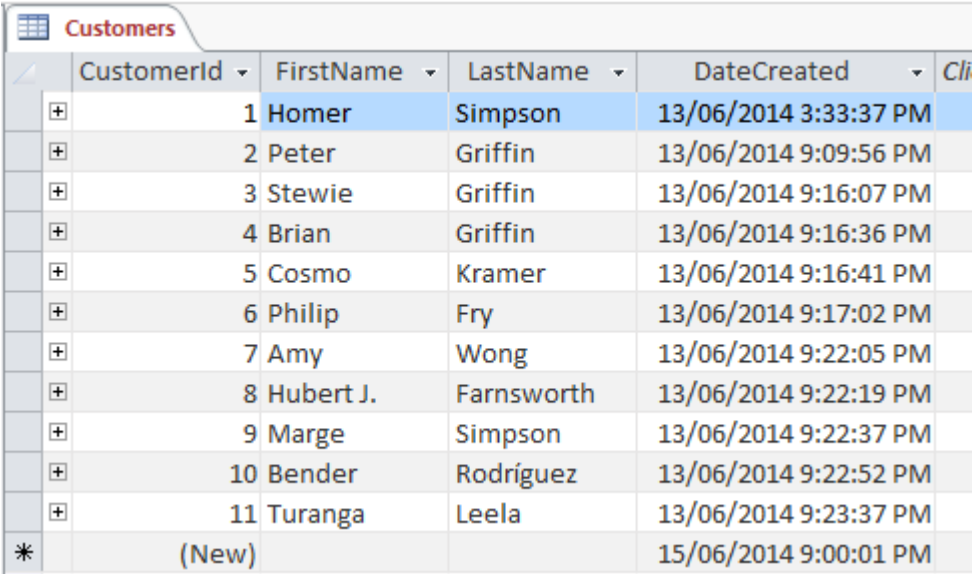
Finalment trobem el model objecte – relació. És molt similar al model relacional, en canvi, tracta cada entitat com si fos un objecte (una instància d'una classe), i ho relaciona per herència. Aquest model aporta una definició que suporta relacions entre dades més complexes, herència i extensions d'objectes.

2.4 Bases de dades relacionals

Les bases de dades relacionals són bases de dades basades en el model relacional. Es caracteritzen per organitzar les dades en diverses taules relacionades entre si. Aquestes taules estan formades per files i columnes. Cada fila indica un registre que està dividit en columnes, on cada una és un tipus d'informació.

Segons Abiteboul, Hull, i Vianu (1995), les bases de dades relacionals són estructures de dades simples. Es representen en taules on a cada columna es presenten dades específiques d'un objecte. A més a més, cada taula té el seu nom i representa una part de la informació del conjunt, en canvi, cada columna representa un atribut segons el seu nom.

Per altra banda, Sharma, i altres (2010) exposen la seva opinió sobre les bases de dades relacionals. Afirma que actualment és el tipus que predomina en el mercat, definida com una BBDD "simple i elegant". Amb la seva base matemàtica sòlida guarda la informació en estructures de dades representades en taules.



	CustomerId	FirstName	LastName	DateCreated	Cl
+	1	Homer	Simpson	13/06/2014 3:33:37 PM	
+	2	Peter	Griffin	13/06/2014 9:09:56 PM	
+	3	Stewie	Griffin	13/06/2014 9:16:07 PM	
+	4	Brian	Griffin	13/06/2014 9:16:36 PM	
+	5	Cosmo	Kramer	13/06/2014 9:16:41 PM	
+	6	Philip	Fry	13/06/2014 9:17:02 PM	
+	7	Amy	Wong	13/06/2014 9:22:05 PM	
+	8	Hubert J.	Farnsworth	13/06/2014 9:22:19 PM	
+	9	Marge	Simpson	13/06/2014 9:22:37 PM	
+	10	Bender	Rodríguez	13/06/2014 9:22:52 PM	
+	11	Turanga	Leela	13/06/2014 9:23:37 PM	
*	(New)			15/06/2014 9:00:01 PM	

Figura 5. Exemple d'una taula. Font: (I., 2020).

En la Figura 5 es pot observar com cada fila fa referència a les dades d'un personatge de ficció i cada columna indica una informació del personatge. La primera columna és una clau única i es fa servir com a identificador.

Ara bé, una base de dades relacional ha de tenir més d'una taula i han d'estar relacionades entre si. Per poder combinar-les es fa ús de la clau única de cada taula.

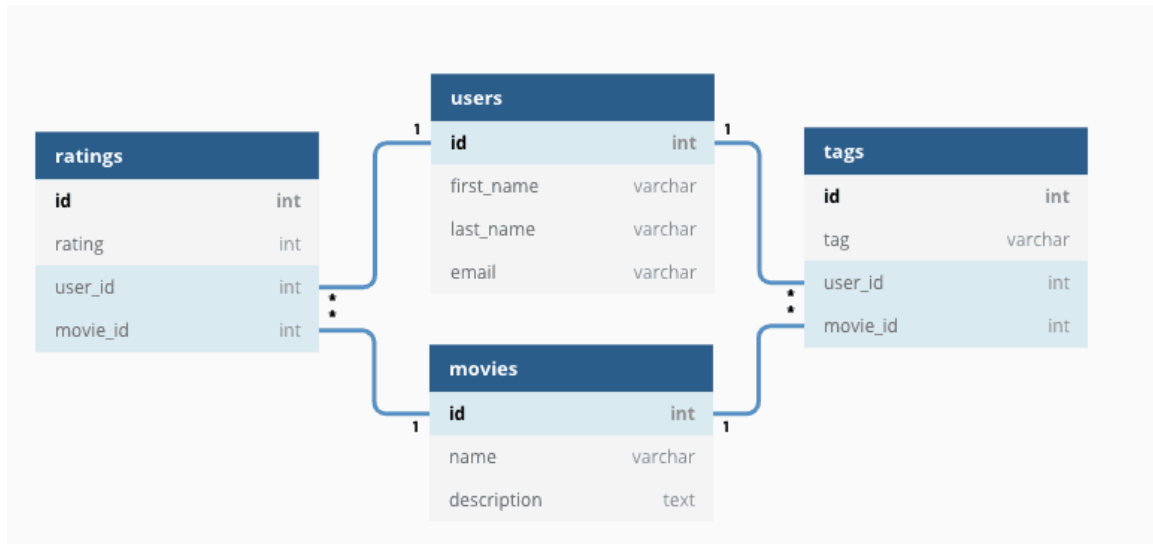


Figura 6. Exemple d'una base de dades relacional. Font: (Omnisci, s. d.).

La Figura 6 mostra l'estructura d'una base de dades relacional. En aquest cas hi ha la taula d'usuaris (users), pel·lícules (movies), puntuacions (ratings) i paraules clau (tags). Aquestes quatre taules estan relacionades entre si per la taula puntuacions, on teòricament, cada usuari puntua pel·lícules. Per tant, la taula puntuacions guarda: la puntuació donada, l'identificador (user_id), de quin usuari ha puntuat i l'identificador de quina pel·lícula ha estat puntuada. D'aquesta forma, segons l'identificador, es pot conèixer tota la informació de l'usuari i/o de la pel·lícula. D'una forma similar a la taula de puntuacions, la taula de paraules clau es relaciona amb la taula usuaris i pel·lícules. En aquesta taula es guarda la relació amb cada una de les taules i la paraula clau de cada una.

S'ha de tenir en compte que les bases de dades poden estar relacionades de diverses formes, aquest concepte es coneix com entitat - relació i existeixen els següents tipus:

- Relació 1 - 1: un element d'una taula es relaciona amb un element d'una altra taula i viceversa.
- Relació 1 - n: un element d'una taula es relaciona amb diversos d'una altra taula. Per exemple: una persona té un càrrec però varies persones poden tenir el mateix càrrec.
- Relació n - n: diversos elements d'una taula poden estar relacionats amb molts d'una altra taula. Per exemple: un agent de policia pot treballar en diversos casos i en un cas hi poden treballar diversos agents.

2.5 Dades estructurades i no estructurades

Les dades poden estar estructurades i no estructurades. Aquesta classificació fa referència a la forma de definir l'organització interna de les dades.

Com explica Smallcombe (2020), les dades estructurades fan referència a les dades que resideixen en un espai fix a l'interior d'un document o registre.

Les dades estructurades són informació altament organitzada i fàcil d'entendre pel sistema. Per treballar amb ella es fa ús de comandes per buscar les dades, afegir-ne de noves o eliminar-ne. Tot això podent relacionar les dades entre sí per obtenir la informació de forma precisa.

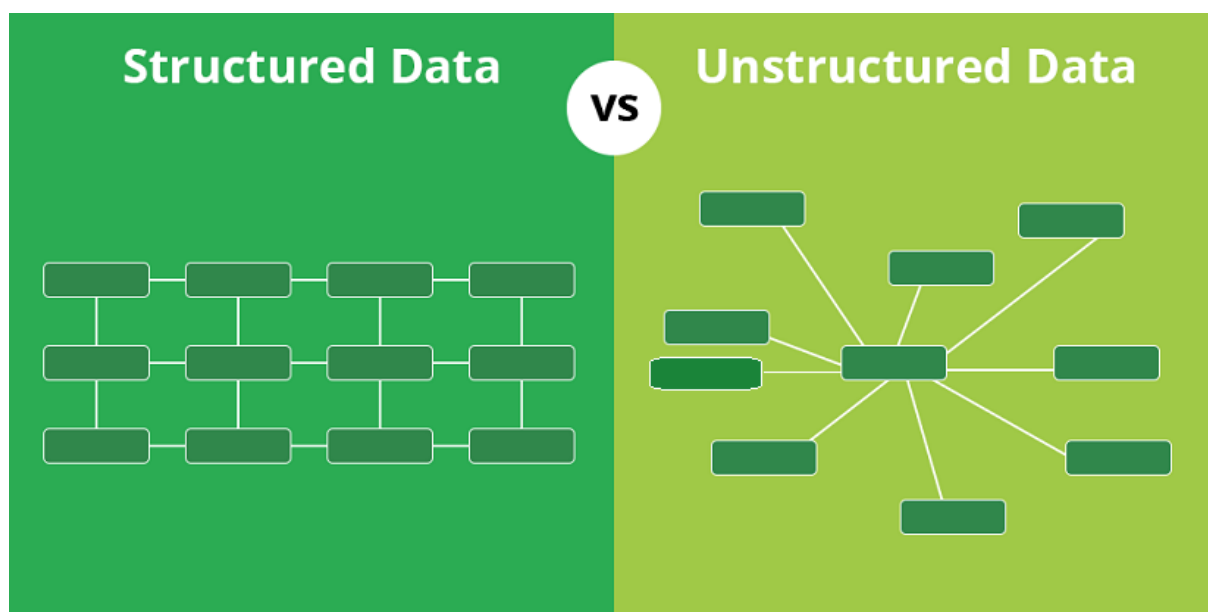


Figura 7. Aportació visual de les dades estructurades en comparació amb les no estructurades. Font: (Gola, 2019).

Taylor (2021) exposa que les dades estructurades usualment resideixen en les bases de dades relacionals. Aquest tipus de dada es caracteritza per emmagatzemar informació com números de telèfon, identificadors de carnets d'identitat, correus electrònics, etc.

El model SQL es basa en dades estructurades, tracta de camps d'informació amb un espai delimitat. Per exemple: si es guarden dades d'un usuari, es pot guardar informació com el nom complet, el DNI, el gènere, l'edat, etc. Els camps que ocuparien aquestes dades s'haurien de definir anteriorment, en el cas del DNI s'hauria d'indicar una llargada de 9 caràcters i també serviria com a identificador de l'usuari, ja que el DNI és un element únic per a cada persona.

Per altra banda, les dades no estructurades es caracteritzen de dades qualitatives, no es poden processar i analitzar utilitzant els mètodes SQL habituals. Les dades que contenen arxius de vídeo, àudio, xarxes socials, imatges, arxius PDF, llibres, etc. Són dades que no estan estructurades ni preparades per ser llegides immediatament. Aquestes dades són complicades de desconstruir perquè no disposen d'un model predefinit, per tant, no es poden relacionar entre elles. Però, existeixen bases de dades NoSQL preparades per poder interpretar aquestes dades. Les dades no estructurades actualment suposen el 80% de les dades.

Taylor (2021) defineix les dades no estructurades com tot el que no es pot identificar com a dades estructurades. És a dir, aquestes dades tenen una estructura interna però no hi han models de dades o esquemes que ho puguin organitzar. Es pot tractar de dades textuais o no, o de dades generades per humans o per màquines. Per tant, tot allò que es podria considerar dades generades per la humanitat són textos, correus electrònics, dades socials, pàgines web, dades mòbils, etc. Per altra banda, el que es pot considerar com a dades generades per màquines són imatges de satèl·lits, dades científiques, dades de sensors, etc.

Segons Smallcombe (2020), les dades no estructurades són totes aquelles que no tenen una estructura fixe. Tot i que puguin tenir una estructura interna, no es considera així a partir d'una base estrablerta.

Smallcombe (2020) indica cinc diferències clau entre les dues estructures de dades. Si les dades són definides o no, la quantitat i qualitat de les dades, on s'emmagatzemen, la facilitat d'anàlisi i el seu format.

La primera consisteix en que les dades estructurades són dades d'un tipus definit dins d'una estructura. Per altra banda, les dades no estructurades s'emmagatzemen en el

seu format original. Les estructurades es guarden en estructures comtaules, amb files i columnes i, per tant, es poden mapejar.

La segona diferència defineix la qualitat i la quantitat de dades. Les dades estructurades són quantitatives, és a dir, són dades formades per números o caràcters que tenen una estructura llegible segons el model establert. Per altra banda, les dades no estructurades són, normalment, dades qualitatives i no es poden analitzar ni processar amb eines i mètodes convencionals. Exemples d'aquestes dades són la transcripció d'una entrevista, una enquesta, una interacció en una xarxa social, com una realitzada a la plataforma Twitter. Per extreure informació d'aquestes dades no estructurades cal utilitzar tècniques analítiques com el data mining.

La tercera diferència clau indica l'emmagatzement de dades. Els magatzems de dades (Data Warehouses) són el punt final del procés d'emmagatzematge de dades. En canvi, els llacs de dades (Data Lakes) són repositoris casi infinits on es guarden les dades en el seu format original, o passat un procés de neteja o filtre. Els dos tipus d'emmagatzematge poden funcionar en el núvol, tot i que les dades no estructurades requereixen una major capacitat que les estructurades.

La quarta diferència és la facilitat d'anàlisi. Les dades estructurades són més fàcils d'anàlitzar, tant per humans com per algoritmes, ja que es disposa d'una estructura predefinida. Per altra banda, les dades no estructurades requereixen un processat previ per considerables analitzables.

La cinquena i última diferència clau es basa en el format de les dades. Les dades estructurades es basen en textos i números, aquest format es defineix des de l'inici. En canvi, les dades no estructurades poden tenir qualsevol format, ja que s'emmagatzemen en el seu format original, es pot tractar d'una pista d'àudio, d'una imatge o un vídeo.

2.6 Bases de dades no relacionals

A part del model SQL també existeix el NoSQL. Aquest model difereix de l'habitual, no segueix el sistema SGBDR (Sistema de Gestió de Bases de Dades Relacionals). En aquest cas, les dades emmagatzemades són no estructurades, no requereixen

estructures fixes com taules. Aquest model funciona en casos que la informació és molt diversa i en grans quantitats, per tant, és més adequat per xarxes socials, desenvolupament web i mòbil i per Big Data, tot i que també es pot fer ús del model SQL. A més a més, el sistema NoSQL a vegades també es pot dir No només SQL, indicant que també suporta llenguatges de consulta SQL.

Sadalage i Fowler (2012) exposen que les bases de dades NoSQL operen sense un esquema definit, permetent que es puguin afegir lliurement camps sense tenir que definir cap canvi en la estructura. Aquesta característica és molt útil a l'hora de tractar amb informació no uniforme i camps personalitzats que una base de dades relacional obligaria a definir prèviament.

Hi han diferents tipus de bases de dades no relacionals definides per Sadalage (2012):

- Bases de dades basades en documents
- Bases de dades clau-valor
- Bases de dades orientades en columnes
- Bases de dades en gràfics

A continuació es descriuen.

2.6.1 Bases de dades basades en documents de dades

Les bases de dades basades en documents guarden les dades en diversos documents, on cada un defineix un valor amb les seves propietats. Els documents de text poden ser en format XML, YAML, JSON i en altres formats binaris com BSON. D'aquesta forma, un conjunt de documents forma un arbre jeràrquic que consisteix en relacionar els valors que formen els documents. Comparant-ho amb el model relacional una taula seria una col·lecció de documents i cada fila un document. Els documents no tindrien una estructura idèntica, ja que es tracta de dades no estructurades, per tant, cada document pot oferir informació diferent.

Per exemple, una col·lecció de documents que tracta de models de cotxes, on cada document indica un model. Tots els documents indiquen les dades bàsiques del model de cotxe que són el número de sèrie (identificador) i el nom del model, però pot ser que un model concret sigui descapotable i, per tant, s'indica en el model en concret.

Si es tractés del model relacional caldria indicar en tots els cotxes si és descapotable o no. D'aquesta forma cada característica extra es pot afegir en el document concret i no cal especificar que la resta no disposa d'aquesta característica.

Per tant, cada base de dades basada en documents disposa de diverses col·leccions de documents i cada col·lecció de diversos documents

Els documents presenten una gran flexibilitat a l'hora de modificar la seva estructura segons els requeriments de l'aplicació en qüestió. Aquesta flexibilitat agilitza el procés de desenvolupament perquè les dades es troben a l'abast immediat dels desenvolupadors.

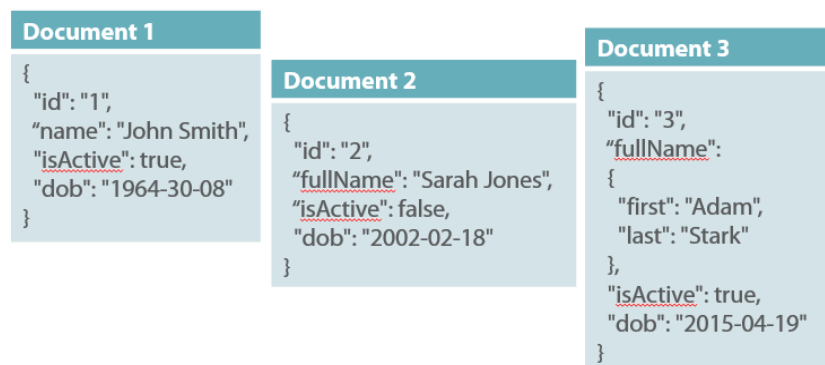


Figura 8. Exemple d'una base de dades basada en documents. Font: (Lobel, 2015).

2.6.2 Bases de dades clau-valor

Les bases de dades clau-valor són el tipus de base de dades més simple: cada valor es guarda relacionat amb una clau. Per poder accedir al valor s'ha de proporcionar la clau. Té un gran símil amb una taula de dues columnes. Es pot obtenir el valor a partir de la clau, canviar el valor o eliminar-lo. La clau no coneix directament el valor que protegeix, d'això s'encarrega l'aplicació. Aquests magatzems de dades basats en clau-valor sempre utilitzen la clau com a accés, degut a que l'únic camp a part del valor és la clau. Generalment tenen un rendiment molt elevat i es pot escalar molt fàcilment, degut a que es fan servir tècniques per optimitzar el procés: ja que l'estructura és senzilla (clau - valor) facilita l'ús de la mateixa. A més a més, s'agrupen els conjunts de clau - valor en nodes segons el tipus de clau, com pot ser per quin caràcter comença la clau. Això permet optimitzar la cerca de la clau perquè es redueix el radi

de busca en només les claus que tenen similitud, per tant, s’ha de buscar entre menys que el total i suposa un increment de la velocitat.

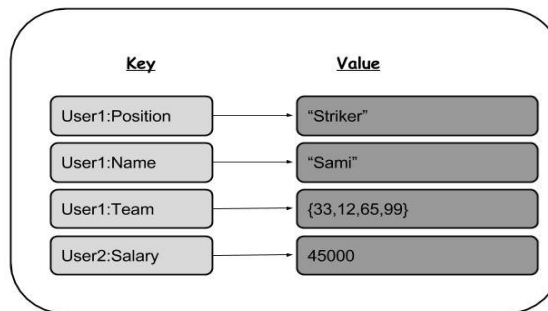


Figura 9. Exemple d’una base de dades clau – valor. Font: (Eshtay, Sleit, & Aldwairi, 2019).

2.6.3 Bases de dades orientades en columnes

Les bases de dades orientades en columnes s’estructuren, com diu el seu nom, en un conjunt de columnes. Guarden les dades en famílies de columnes designades com a files. Cada fila consisteix en un conjunt de columnes associades amb una clau de fila, en aquest cas, la clau és l’identificador i cada columna és un atribut de l’objecte que representa la fila. Aquest format aporta un major rendiment, és a dir, es consumeix menor memòria ja que es poden només analitzar les columnes que es necessiten.

RowKey 1	ColumnKey 1	ColumnKey 2	ColumnKey 3	...
	ColumnValue 1	ColumnValue 2	ColumnValue 3	...
⋮				

RowKey 1	SuperColumn 1		SuperColumn 2		...
	ColumnKey 1	ColumnKey 2	ColumnKey 3
	ColumnValue 1	ColumnValue 2	ColumnValue 3
⋮					

Figura 10. Exemple d’una base de dades orientada en columnes. Font: (Eshtay, Sleit, & Aldwairi, 2019).

2.6.4 Bases de dades basades en gràfics

Les bases de dades basades en gràfics centren la seva estructura en la relació entre diferents elements. Permeten guardar entitats i les relacions entre aquestes. Les entitats es coneixen com a nodes, els quals tenen propietats. El símil més directe és que cada node és la instància d'un objecte en una aplicació. Les relacions es coneixen com a extrems, els quals tenen les seves propietats i el seu significat direccional, depenent de com es relacioni amb un altre node. Agafant com a exemple la Figura 11, cada node és un rectangle i cada fletxa indica com es relacionen els nodes, els extrems serien el significat de cada fletxa, és a dir, quina relació hi ha entre nodes.

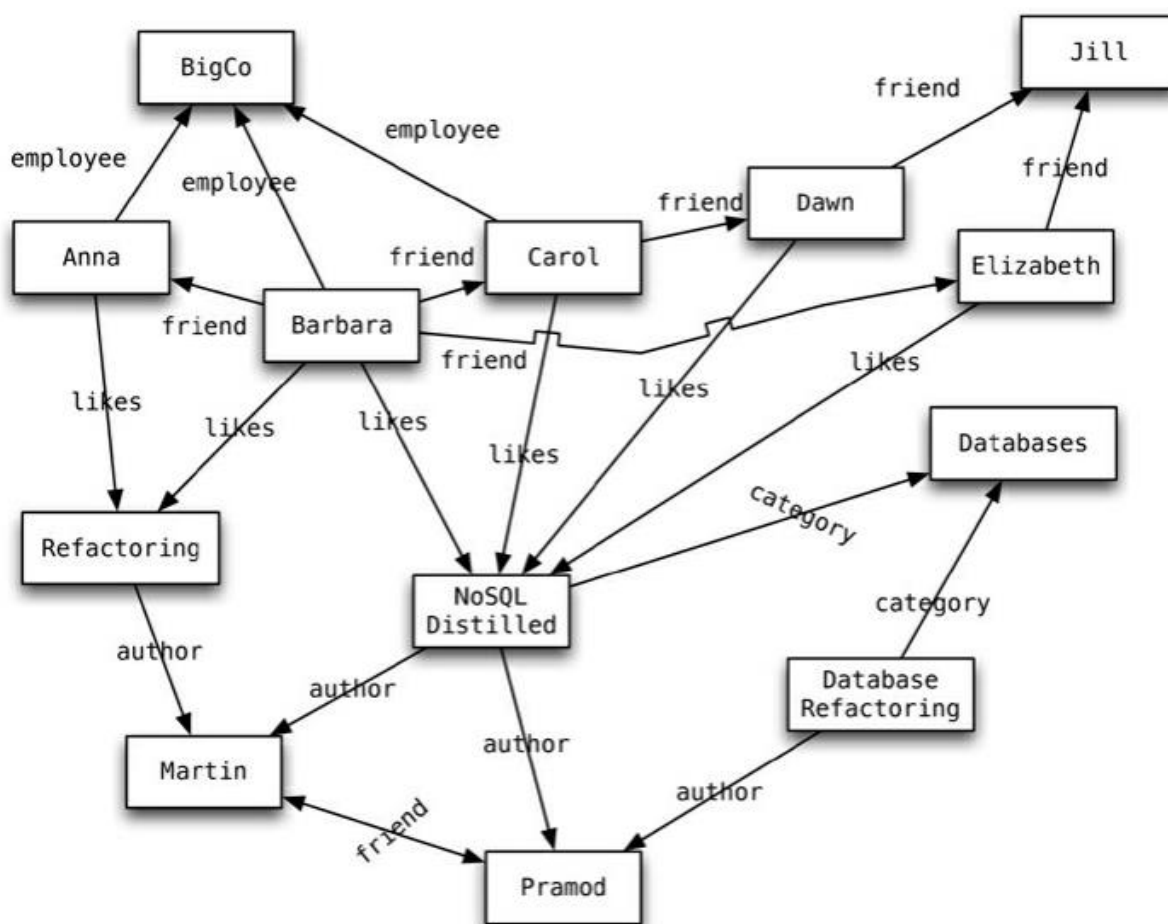


Figura 11. Exemple d'una base de dades basada en gràfics. Font: (Sadalage & Fowler, 2012).

Una forma de realitzar consultes en una base de dades d'aquest tipus seria de la següent forma. Seguint amb l'exemple de la Figura 11: recuperar totes les persones que els hi agrada "NoSQL Distilled" i tenen com a amiga la Barbara. En aquest cas la

resposta seria la Carol i la Elizabeth. A partir dels nodes i les seves relacions es pot buscar la informació necessària.

2.7 Motors de bases de dades

Actualment existeixen moltes bases de dades de diversos tipus. Segons ExploreGroup (2019), MySQL, PostgreSQL, Microsoft SQL Server, SQLite, MongoDB, Redis, MariaDB, Oracle i moltes altres són les bases de dades més populars en l'actualitat. Dins d'aquesta llista ja hi apareixen dues bases de dades que són NoSQL, les quals són MongoDB i Cassandra.

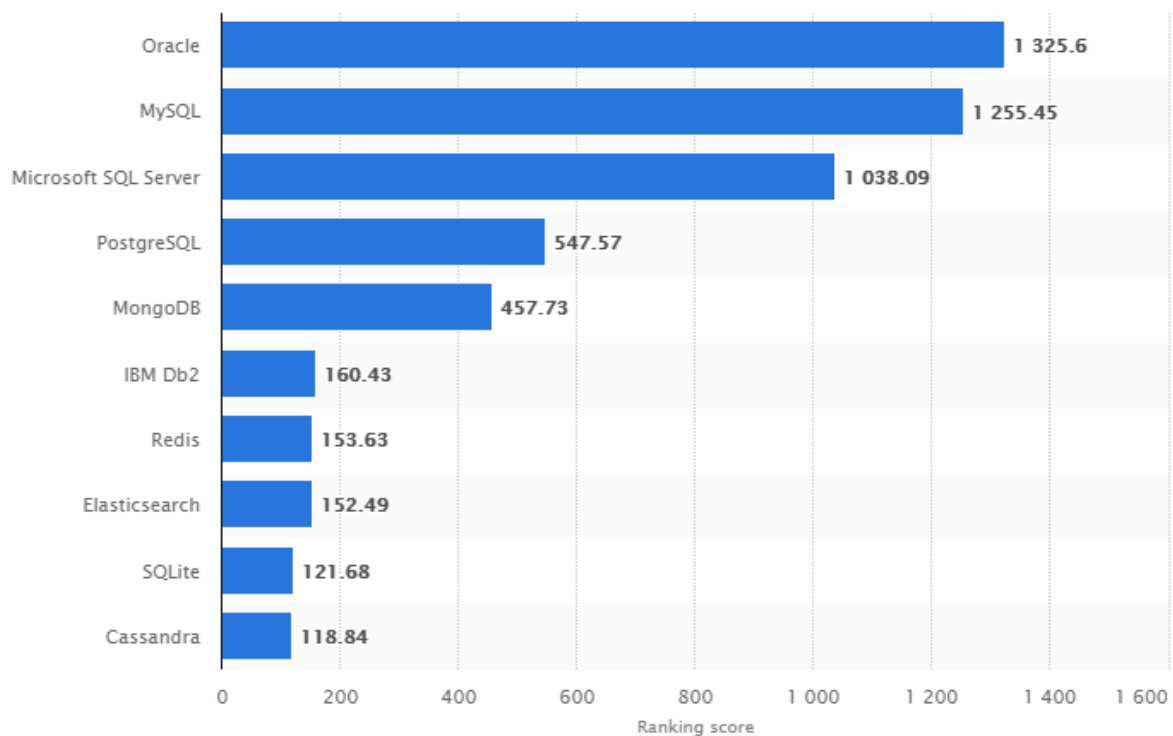


Figura 12. Classificació de les bases de dades més populars del desembre de 2020.
Font: (Statista, 2021).

A continuació s'exposen tres exemples que fan referència a una base de dades relacional, una no relacional i la tercera tracta d'una base de dades de caràcter local, és a dir, no necessita connexió a la xarxa per funcionar.

Si agafem com exemple MySQL, Mack (2014) exposa que es tracta d'una de les bases de dades relacionals més populars mundialment. MySQL disposa dels seus

avantatges. La seva facilitat d'ús és el primer, un procés d'instal·lació senzill gràcies a les eines "third-party" que se s'hi poden afegir. No només per la instal·lació, sinó que també facilita la preparació i implementació de la mateixa. Per altra banda, es tracta de codi parcialment "open-source", parcial degut a que MySQL actualment pertany a Oracle, però el codi segueix sent lliure. Tot seguit, implementar aquesta BBDD pot suposar des d'un cost gratuït fins a deu mil dòlars, depenent del pla que es contracti. Finalment, la popularitat de MySQL ha crescut molt en els últims anys, actualment es manté com una de les més populars. És compatible amb qualsevol sistema operatiu i és pràcticament un estàndard de la indústria.

Per altra banda, MySQL també té els seus punts febles, té problemes d'estabilitat relacionats en la forma en que gestiona certes funcionalitats. A més a més, carrega amb problemes de rendiment quan es tracta d'escalabilitat, disposa d'un límit de volum virtual de dades. Per tant, no es recomana fer servir aquesta base de dades si s'ha de mantenir un gran nivell de tràfic de dades. També cal afegir que, tot i que existeixen complements que faciliten la seva implementació pot arribar a ser massa dependent d'aquests complements.

Com a segon exemple hi ha el motor de base de dades MongoDB. És NoSQL i això suposa un funcionament totalment diferent a l' exemple anterior. A l'article de Altexsoft (2019) s'exposa el següent. MongoDB és gratuït, "open-source" i un sistema de bases de dades no relacional. Inicialment no estava enfocat per treballar amb dades estructurades, tot i això, es pot utilitzar per processar tant dades estructurades com no estructurades.

Primer de tot, disposa d'una naturalesa ràpida i amb operacions fàcils de dur a terme, es pot accedir, emmagatzemar i retirar dades de la base de dades ràpidament sense la necessitat d'una confirmació addicional. També disposa de compatibilitat amb altres sistemes de gestió de bases de dades, tant SQL com NoSQL, és a dir, permet que "third-parties" construeixin els seus propis motors d'emmagatzematge de dades per MongoDB, des d'un punt de vista comercial genera un valor afegit. A més a més, disposa d'escalabilitat horitzontal, això significa que pot incrementar la capacitat connectant diversos dispositius perquè treballin com si fos un sol.

En canvi, MongoDB també té punts negatius. Al tractar-se d'una base de dades enfocada en la velocitat mostra falles en la seguretat. La autenticació d'usuari no és una característica base i per disposar d'una major seguretat és necessari contractar un pla més avançat. També cal destacar que MongoDB disposa d'actualitzacions constants, ara bé, no s'assegura que totes les característiques tindran el mateix funcionament, tota la manipulació de dades ha de tenir en compte aquestes actualitzacions del funcionament de la base de dades, per tant, s'han de realitzar comprovacions periòdiques per assegurar el correcte funcionament.

I com a últim exemple hi ha SQLite. Carrera (2019) exposa la seva simplicitat a l'hora d'operar amb aquest SGBD. Primer de tot, no necessita un servidor extern per gestionar la informació, de la mateixa forma que no necessita un usuari administrador per gestionar els permisos d'accés.

Per altra banda, SQLite consisteix en un conjunt d'arxius que es poden col·locar en qualsevol directori, aquí resideix la seva simplicitat esmentada anteriorment. Per tant, es poden compartir i/o copiar arxius de dades d'un disc dur a un altre o es poden enviar per correu electrònic.

Finalment, el motor disposa d'escriptura directa en l'arxiu manifest¹, això vol dir que es pot emmagatzemar qualsevol tipus de dada en qualsevol espai, no és necessari, com es realitza en altres bases de dades, tenir columnes definides. Una excepció són les columnes de clau primària on només es poden guardar dades numèriques.

2.8 Les bases de dades com a eina de l'actualitat

Les bases de dades són una eina cada vegada més utilitzada en l'actualitat. També ha suposat un canvi l'evolució dels sistemes de gestió de bases de dades. Thakur (2016) exposa que les empreses estan guanyant més eficiència perquè es pot tenir

¹ L'arxiu "manifest" conté metadades d'un grup d'arxius adjunts que formen part d'un conjunt. Poden ser dades com el número de versió, la llicència, el nom, etc.

un registre de tot el que es realitza. A més a més, es poden realitzar cerques més ràpides d'informació i registres de qualsevol persona o producte.

Thakur (2016) llista les següents aplicacions i usos dels SGBD:

- Sistema de gestió de ferrocarrils i aeroports
- Sistema de gestió de llibreries
- Banca
- Universitats i col·legis
- Xarxes socials
- Telecomunicacions
- Finances
- Entorn militar
- Compres en línia
- Gestió de recursos humans
- Manufacturació

No es pot negar que els sistemes de gestió de bases de dades tenen una gran quantitat d'aplicacions.

Per altra banda, en videojocs Blackwell (2019) exposa que les bases de dades són potencialment necessàries en el moment en que un videojoc ha de llegir o generar dades. En aquest cas, ha d'existir una base de dades central on tots els jugadors, des dels seus respectius dispositius, s'hi puguin connectar. D'aquesta forma, els jugadors no s'han de preocupar de gestionar les dades de les seves partides.

L'únic aspecte que els jugadors han de tenir en compte quan es fa servir una base de dades en un videojoc, on hi ha sistema d'usuaris, són les dades de sessió. D'aquesta forma, l'usuari pren part de la responsabilitat de les seves dades, mentre que la base de dades s'encarrega de guardar-les i protegir-les.

Tot i no disposar d'un servidor és útil utilitzar una base de dades. La base de dades externa no és única de videojocs multijugador, també es poden emmagatzemar

mètriques, guardar les opcions de joc, els perfils de jugador, els guardats de partides, etc.

2.9 El motor de jocs Unity

En aquest projecte es farà ús del motor de jocs Unity. Com exposa Dealessandri (2020), aquest motor de videojocs es coneix com l'entorn més adequat per videojocs índies i per mòbil. El setembre de 2019 el 52% dels 1.000 videojocs més destacats de mòbil estaven desenvolupats amb Unity, de la mateixa forma pel 60% dels videojocs AR/VR. Els jugadors de videojocs en Unity es troben en fins a 195 països, que això suposa tots els països del món.

El motor de videojocs Unity suporta fins a 25 plataformes: des de iOS o Playstation 4, fins a Windows Mixed Reality o Android TV. Unity permet desenvolupar en entorns 2D, 3D, AR i VR².

Ara bé, la seva popularitat és principalment gràcies a que és un motor gratuït. La llicència personal i d'un equip molt petit és gratuïta mentre no es superin els 100.000\$ de beneficis. Per altra banda, Unity disposa de llicències per equips més grans que ja no són gratuïtes.

Unity és un bon motor per principiants. Al ser gratuït és una eina a l'abast d'estudiants i de qualsevol desenvolupador. A més a més, el juliol de 2020 es va implementar la eina de scripting visual "Bolt", la qual permet implementar la lògica del videojoc sense tenir que saber com programar. Aquesta eina es basa en un conjunt de nodes que qualsevol desenvolupador, inclosos programadors, poden fer servir per dissenyar la lògica del videojoc o per crear prototips ràpidament.

Per altra banda, Unity disposa de la capacitat per realitzar accions ràpidament, permet realitzar iteracions de forma àgil, és a dir, idees simples es poden desenvolupar i testejar en poc temps. A l'hora de realitzar una pluja d'idees per decidir el concepte

² AR significa realitat augmentada i VR realitat virtual.

d'un nou joc és factible desenvolupar i provar la funcionalitat bàsica de les idees més destacades, sempre i quan suposi una base senzilla.

L'“Asset store” de la que disposa Unity és un altre dels seus punts forts. Al voltant de 1.5 milions de desenvolupadors visiten la botiga cada mes per consultar els 56.000 paquets de contingut disponibles, que representen fins a un milió d'assets individuals. A part de tot el que es pot comprar a la botiga de Unity, el motor també permet desenvolupar eines pròpies que ajudin al desenvolupament. Aquestes eines poden ajudar a automatitzar tasques que, realitzades una gran quantitat de vegades, acaben sent feixugues.

Per acabar, també és necessari conèixer els punts febles del motor. Unity no és adequat per grans projectes, en aquest sentit és poc flexible i no permet desenvolupar a gran escala i mantenir, al mateix temps, un nivell d'optimització adequat. A més a més, el motor no està tant preparat per artistes, és a dir, en l'apartat artístic no hi han tantes funcionalitats disponibles.

2.10 Les bases de dades en videojocs

Actualment, en el món dels videojocs existeixen molts tipus de gèneres diferents, on cada videojoc disposa de les seves característiques, els tipus de jugadors als que estan enfocats, les plataformes en la que estan disponibles, el cost, la duració, etc. Hi ha una gran quantitat de variables que influeixen en el tipus de base de dades, en funció de les dades que es volen emmagatzemar i de la consulta i anàlisi que se'n vol extreure.

2.10.1 Gèneres

Primer de tot cal conèixer quin tipus de videojocs existeixen. Sheldon, Toole, Orkin, i Despain (2009) i Pavlovic (2020) classifiquen els gèneres més rellevants de videojocs en 10 tipus:

- Sandbox
- RTS (Real Time Strategy)
- Shooter

- MOBA (Multiplayer Online Battle Arena)
- RPG (Role-Playing Games)
- Simulació i esports
- Trencaclosques i jocs de festa
- Acció – aventura
- Supervivència i terror
- Plataformeig

A continuació es descriuen.

Sandbox és un gènere que conté decisions lliures del jugador, entorns grans i oberts i una jugabilitat no-lineal. En aquest tipus de videojocs els jugadors tenen diversos objectius i opcions a l'hora d'avançar en el joc. Com a exemples de videojoc Sandbox hi ha *The Sims* i *Minecraft*, ambdós videojocs disposen d'objectius simples, però la principal característica d'aquests és que més enllà dels objectius esmentats hi ha llibertat absoluta del jugador per realitzar el que cregui convenient.

Tot seguit, el gènere RTS, estratègia en temps real, consisteix en la competència de diferents faccions, ja siguin controlats per jugadors o pel propi videojoc, en temps real. En aquest cas, un jugador pot competir contra altres jugadors, contra la IA o contra ambdós a la vegada, on la condició de victòria més comuna és la de ser l'últim viu. Un exemple del gènere és el videojoc *Age of Empires*.

A continuació hi ha el Shooter, que pot ser "First Person Shooter" o "Third Person Shooter", els jocs de dispars es caracteritzen per funcionar en primera, tercera persona o ambdós, és a dir, poder canviar d'un a l'altre. Els FPS tenen el punt de vista des de l'ull humà, només mostrant la informació essencial, per altra banda, els TPS es basen més en mostrar l'entorn, veient el personatge que es controla des de fora. Dins del propi gènere hi han subgèneres concrets com "Shoot-em-up", que consisteix en el jugador contra grans quantitats d'enemics que van apareixent en pantalla, i també hi han els "Bullet-hell", que tracta de que hi han molts projectils en pantalla que el jugador ha d'esquivar i/o eliminar. Com a exemple d'un videojoc FPS hi ha el *Halo* i d'un TPS el *Gears of War*.

El gènere MOBA (Multiplayer Online Battle Arena) consisteix en un tipus de videojoc de tipus multijugador on diversos jugadors per equips competeixen entre ells. Es caracteritza perquè cada jugador fa servir un personatge a escollir i adopta un rol dins de la partida. Els exemples d'aquest gènere són *League of Legends* i *Dota 2*.

Més endavant trobem el gènere RPG (Role-Playing Games). Aquests videojocs tenen com a premissa bàsica simular un univers fantàstic on el jugador crea i controla un personatge que anirà creixent a mesura que avanci en la seva aventura. Existeixen múltiples subgèneres: RPG (Role-playing games), ARPG (Action RPG), CRPG (Computer RPG), MMORPG (Massively Multiplayer Online RPG), TRPG (Tactical RPG), Roguelike i Roguelite, tots aquests subgèneres tenen les seves particularitats dins dels jocs de rol. Un exemple d'aquest gènere és la franquícia *Final Fantasy*.

Com a sisè gènere hi han els videojocs de simulació i esport, aquests es caracteritzen per simular una realitat o un esport. Aquests videojocs no poden innovar en simulació en si ja que l'objectiu és assemblar-se més a la realitat, i aquesta és finita. Per tant, aquest tipus de jocs es centren més en innovar en com funcionen les mecàniques i l'apartat gràfic. El gènere de simulació engloba videojocs de construcció dins del món i videojocs de realitat virtual. Com a exemples hi han *Forza Motorsport*, *NBA2K* i *Fifa*.

Tot seguit, el gènere de trencaclosques i jocs de festa, s'ajunten els dos gèneres perquè fan èmfasi a les mecàniques. Es juga en un tauler enfocat en una temàtica o un joc de taula tradicional amb unes normes establertes. Aquests videojocs poden ser individuals o multijugadors. Alguns exemples d'aquests gèneres són *The Talos Principle*, *Portal 2* i *Mario Party*.

Seguint amb el gènere d'acció - aventura, els videojocs d'aquests gèneres estan enfocats en la història i el combat. Segons quin aspecte destaquí més vol dir que està més enfocat en l'aventura o en l'acció. Dos exemples d'aquests gèneres són *Sekiro: Shadows Die Twice* i *Star Wars Jedi: Fallen Order*.

Els gèneres de supervivència i terror tenen molts aspectes en comú, fins al punt que existeix el subgènere survival-horror. Els jocs de supervivència es centren més en la gestió de recursos i creació d'objectes, per altra banda, els jocs de terror es centren en la història i la ambientació. Com a exemples estan *The Long Dark* i *Don't Starve*.

Finalment, el gènere de plataformeig normalment es caracteritzava per tenir una vista lateral i controls simples, actualment hi ha gran quantitat de jocs 3D de plataformes. Aquest gènere és molt popular per desenvolupar en estudis petits. Com a exemples es presenten *Cuphead* i *Crash bandicoot*.

2.10.2 Elements dels videojocs

En un videojoc hi ha diversos elements a tenir en compte. En aquest apartat s'analitzen els elements dels videojocs que poden necessitar una base de dades com a recolzament. Segons explica Butcher (s.d.) la indústria s'ha vist implicada en el moviment dels videojocs de caràcter social, des de videojocs més enfocats a una tipologia de jugadors més extrems, com són els jocs MMO (Massively Multiplayer Online) fins a videojocs més populars com poden ser els jocs de Facebook.

Per exemple, segons Butcher (s.d.) el videojoc de Facebook *Happy Farm* contempla 228 milions d'usuaris actius i 23 milions diaris, majoritàriament a Àsia. Per altra banda, el videojoc *World of Warcraft* disposa de 11 milions de subscripcions mensuals. En aquest cas, no està centrat en només un apartat del món dels videojocs, aquests dos exemples són de videojocs totalment diferents, però són casos exitosos. Per tant, quins són els elements d'un videojoc que s'han de tenir en compte per implementar una base de dades?

Primer de tot, els comptes del videojocs. Són les dades que cada usuari té fora del videojoc, són dades de sessió com el correu electrònic i la contrasenya, es pot donar el cas que es demanin més dades com la edat o una identificació com el número de telèfon o el número del document identificador per evitar comptes repetits. Aquestes dades són finites ja que hi ha un màxim d'elements que pot tenir, per tant, són les dades més reduïdes i menys utilitzades. Un cop s'identifica correctament l'usuari passa a utilitzar el perfil i no el compte.

El perfil d'usuari és un personatge creat en un videojoc, d'aquesta forma un usuari pot tenir diversos personatges i, a la vegada, diversos perfils d'usuari. El que es considera normal avui dia es poder tenir més de cinc perfils d'usuari en un videojoc que faci servir aquesta característica. Es pot donar el cas que dins de la informació guardada en el perfil de l'usuari també s'inclouen dades com els objectes de l'inventari del

personatge o les habilitats que té assignades, però si aquestes dades són molt extenses es poden separar del perfil.

El següent element a tenir en compte és la llista d'amistats, per mantenir un contacte dins del videojocs hi ha l'opció de sol·licitar amiat, similar al funcionament de Facebook o intercanviar telèfons en la realitat. En aquest cas, és comú limitar el número d'amics a tenir en el videojocs per poder preveure l'escala de les dades.

Tot seguit, cal esmentar la missatgeria. Dins dels videojocs amb aspectes socials, és habitual que els jugadors es puguin comunicar entre ells, la comunicació no necessàriament ha de ser síncrona, és a dir, hi pot haver una missatgeria a temps real, però també es poden deixar missatges a mode de correu electrònic. Per limitar aquestes dades és habitual limitar el número de missatges disponibles o establir una caducitat pels mateixos, com pot ser de 30 dies. En videojocs on la comunicació ha de ser ràpida i fluida ja que el ritme de joc és elevat, és habitual que hi hagi comunicació per veu.

Un dels elements més importants són les dades de pagament dels usuaris. Els desenvolupadors han de decidir si volen guardar aquestes dades o no, aquestes dades van des de targetes de crèdit fins a altres dades sensibles. En aquest cas, la base de dades ha de complir totalment amb les característiques ACID per assegurar la seguretat de les dades.

A continuació estan les sessions, aquestes dades són unes de les freqüentades i les més volàtils. Inclús si hi ha hagut un error en el joc aquestes dades s'han de poder actualitzar.

Per altra banda, l'inventari de cada perfil d'usuari és una dada molt important. Descriu com el jugador interactua amb l'univers del videojoc. Aquestes dades són les pertinences virtuals de cada jugador i mostra com juga cadascun. El servidor encarregat de guardar aquestes dades ha de ser ràpid i gran, ja que aquestes dades poden créixer en gran mesura i canviar amb molta freqüència. Cada x temps el servidor actualitzarà aquestes dades a la base de dades, en el temps en que no s'actualitza en el servidor, si aquestes dades canvien es guardaran en cache (memòria

temporal). Aquestes dades són senzilles de gestionar, degut a que cada usuari només pot accedir a un personatge a la vegada i, per tant, a un inventari a la vegada.

Tot seguit, hi han les taules de puntuacions. Aquestes taules canvien i es consulten constantment. Els usuaris volen saber en quina posició de la taula es troben per intentar millorar el seu resultat si no els hi agraden.

Finalment, es presenta el sistema de trofeus. És un element important perquè mostra públicament el que s'ha aconseguit. Es considera el segon element més important pel qual s'accedeix al joc.

Aquest conjunt d'elements són els que s'han de tenir en compte pel videojoc i escollir quins d'ells implementar.

2.10.3 Elements de les bases de dades

Es proposen dues característiques relacionades amb les bases de dades. Primer de tot, durant les transaccions que duu a terme la base de dades a petició del servidor o el videojoc es poden perdre dades degut a un mal funcionament, en aquest aspectes s'ha de minimitzar o anular aquestes possibles pèrdues. També s'ha d'assegurar la seva integritat al llarg del temps, és a dir, un cop estan les dades emmagatzemades cal assegurar que es mantenen segures. El segon aspecte és la velocitat en que es realitzen les consultes, algunes de les característiques que es presenten a continuació es poden realitzar a temps real i cal assegurar que no s'alenteixi el procés degut a un lent funcionament de la base de dades.

Butcher (s.d.) presenta els següents elements a tenir en compte en una base de dades per un videojoc.

	Integritat	Velocitat
Compte de joc	x	
Perfil d'usuari	x	x
Llista d'amistats	x	
Missatgeria	x	
Dades de pagament	x	
Inventari	x	x

Taula de puntuació		x
Sistema de trofeus	x	
Dades de sessió		x

Taula 1. Taula d'elements d'un videojoc rellevants per una base de dades. Font: Elaboració pròpia, a partir de les dades proporcionades per (Butcher, s.d.).

La Taula 1 mostra quins aspectes cal tenir en compte d'un videojoc relacionats amb una base de dades i com s'han de gestionar. En aquest cas, cal remarcar que la integritat de les dades relacionades amb la taula de puntuacions és opcional perquè en cas de perdre aquestes dades es podrien tornar a generar a partir de la resta de dades. Per exemple, si cada usuari guarda la quantitat de partides guanyades i jugades, en cas de perdre les dades relacionades amb la taula de puntuació, es podria tornar a generar a partir de les dades que té cada usuari.

Segons Butcher (s.d.) el criteri per classificar cada característica d'un videojoc segons la prioritats que ha de tenir per la base de dades. Primer de tot, s'ha d'evitar la pèrdua de dades pel compliment legal quan es tracta de les dades de pagament, l'inventari i el compte de joc. El compte de pagament és de les tres característiques la dada més sensible i, per tant, ha de estar protegida. L'inventari són les pertinences virtuals que ha acumulat l'usuari i són de gran importància pel jugador. Per últim, el compte de joc pot guardar dades relacionades amb l'usuari, més enllà de les dades com el correu electrònic i la contrasenya, que també són importants, es pot guardar el nom real o el número de telèfon de la persona.

També s'ha d'evitar la pèrdua de dades de el sistema de trofeus i la missatgeria però en aquest cas degut per mantenir la reputació professional dels desenvolupadors implicats. Com a menys important però necessari també s'han d'evitar la pèrdua de dades de la llista d'amistats i el perfil d'usuari per assegurar mantenir el progrés de l'usuari, dins del perfil també es pot donar el cas que s'inclogui l'inventari, però aquesta decisió depèn del disseny.

Tot seguit, la velocitat és prioritària per les dades de sessió, pel perfil d'usuari, per les dades de l'inventari i per la taula de puntuacions. Aquests tres elements requereixen accés a temps real i, per tant, cal que es duguin a terme sense alentir la sessió de joc del jugador. La taula de puntuacions és una característica volàtil, ja que pot canviar

en qualsevol moment, un sol usuari pot canviar una gran part de la taula de puntuacions.

2.11 Resum

El desenvolupament d'un videojoc és molt extens i consumeix una gran quantitat de recursos, per tant, és necessari conèixer prèviament totes les característiques que pot oferir una base de dades. Una base de dades en sí és un element que emmagatzema dades, però a vista del desenvolupador és molt més extens. Cal escollir el model de base de dades que més s'ajusti al videojoc que es vol desenvolupar com el SGBD que millor pugui satisfer les necessitats del joc a desenvolupar.

El model relacional ofereix més consistència en la seva estructura, tots els elements s'han de definir prèviament i tot el contingut que es vagi emmagatzemant a la base de dades mai podrà anar més enllà dels aspectes preestablerts. En canvi, una base de dades que segueix un model no relacional, és a dir, que les seves dades no són estructurades, perd les limitacions que presenta el model relacional i pot arribar a emmagatzemar dades que no segueixen una estructura clara.

Ara bé, els videojocs també tenen una gran quantitat de funcionalitats que s'han de tenir en compte. Com més complex vulgui ser un videojoc més sistemes i funcionalitats necessita implementar. Un sistema d'usuaris permet diferenciar els usuaris i mantenir la individualitat, per altra banda, necessita una base de dades al darrere que recolzi el seu correcte funcionament.

El desenvolupament d'un videojoc és un procés molt complex on molts camps s'hi veuen implicats, en aquest marc teòric s'ha investigat tota la informació necessària per poder conèixer les bases de dades i les seves implicacions, els videojocs i el motor de jocs Unity, amb la finalitat de complir els objectius establerts en el Capítol 4.

3. Anàlisi de referents

En aquest capítol s'exposen els referents que s'han analitzat. Aquests consisteixen en videojocs que fan servir bases de dades per recolzar les seves funcionalitats. També s'analitzen dos videojocs que serviran per dur a terme el desenvolupament del videojoc en fase prototip, el qual s'explica en la secció 6.1.

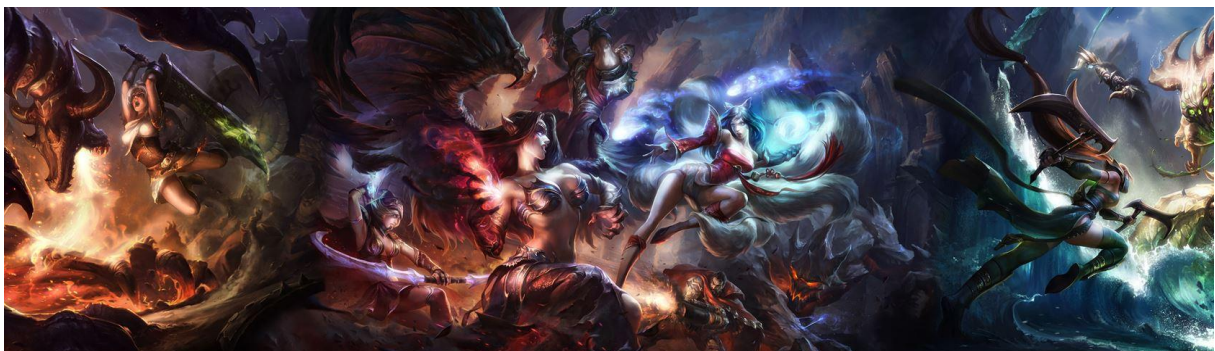
3.1 Bases de Dades

La base de dades que es fa servir en un videojoc acostuma a ser informació que les empreses desenvolupadores de videojocs no comparteixen. Aquestes dades es consideren confidencials i resulta en informació de coneixença difícil. Tot i això, es presenten tres bases de dades de les quals s'ha aconseguit obtenir informació.

3.1.1 MySQL

Turk (2018), Enginyer Sènior de la infraestructura dels comptes dels usuaris a Riot Games, explica el funcionament de la seva base de dades encarregada d'emmagatzemar les dades dels usuaris dins la companyia.

Riot Games ha desenvolupat videojocs com *League of Legends*, *Legends of Runaterra* i *Valorant*, com a més destacats. Fa ús de MySQL com a base de dades, la qual s'encarrega majoritàriament de gestionar les dades dels seus jugadors.



Il·lustració 1. Imatge del videojoc *League of Legends*.

La principal raó per escollir la base de dades és que la companyia a la que pertany Riot Games, que és Tencent, també treballa amb MySQL, per tant ja tenen una gran quantitat d'enginyers amb els coneixements necessaris per treballar-hi.

Riot Games realitza una còpia de seguretat diària de les dades dels seus usuaris. D'aquesta forma es pot garantir que si el sistema falla, només es perden les dades que s'han gestionat en el mateix dia. Una gran empresa com és Riot Games ha de garantir la seguretat de les dades dels seus usuaris, ja que si es produís un error en el sistema de dades es podrien perdre una gran quantitat d'informació.

Tyler Turk aporta una visió del que s'ha après i què es pot millorar. Primer de tot, s'ha de poder realitzar un seguiment de la base de dades en tot moment, s'ha de facilitar el màxim possible aquesta tasca per poder detectar qualsevol anomalia.

S'ha de considerar i practicar diverses metodologies de migració per trobar la que més s'ajusta al projecte. Per conèixer el que es necessita també s'ha de documentar el disseny de la mateixa, d'aquesta forma es pot consultar i analitzar amb més profunditat el sistema de bases de dades que es vol aconseguir. Per altra banda, per aconseguir que el sistema pugui anar evolucionant a mesura que ho fa el projecte s'ha d'assegurar que segueixi una estructura iterativa.

També cal evitar en la mesura possible crear un sistema nou, és més segur utilitzar serveis ja provats i fiables, que poden assegurar el seu correcte funcionament. Un cop ja establert el sistema s'ha de testejar intentant simular al màxim possible casos reals, és a dir, si s'espera una gran quantitat d'usuaris aleshores cal realitzar una prova d'estrès per comprovar la capacitat del servidor.

Un segon cas de l'ús de SQL és el videojoc *Osiris: New Dawn*. En l'actualització de contingut realitzada el 30 de gener de 2021, FenixFireEntertainment (2021) exposa que deixen de fer servir el sistema SQL perquè conduïa a diversos errors.

Com explica Brian a l'informe de la actualització, el seu equip estava tenint problemes amb el sistema de base de dades SQL, concretament a l'hora de transmetre les dades en les seves connexions. Degut a aquest cas, a partir d'aquesta versió del videojoc faran ús del sistema de guardat en el núvol.



Il·lustració 2. Imatge del videojoc *Osiris: New Dawn*.

Ja que el videojoc *Osiris: New Dawn* no és del gènere MMO (Multijugador Massiu Online) es poden permetre no fer ús directe d'una base de dades. Ara bé, el videojoc és multijugador. S'han de seguir guardant les dades dels seus usuaris. Com a resultat, han escollit un servei menys potent però més eficient.

3.1.2 Oracle

La companyia Blizzard Entertainment també fa ús de bases de dades en els seus videojocs. El més conegut és *World of Warcraft*, joc del gènere MMORPG (Massive Multiplayer Online Role-Playing Game). Aquest gènere tracta de videojocs de rol que permeten que grans quantitats de jugadors entrin en un món virtual de forma simultània on creen un personatge a través del qual poden interactuar en el món i entre ells.

Aquest tipus de videojocs estan dissenyats perquè hi juguin grans quantitats de jugadors, cada un d'ells té el usuari amb totes les seves dades. Per tant, un videojoc d'aquest estil necessita un sistema gestió d'usuaris, a part de gestionar totes les dades dins del joc, això de forma massiva.



Il·lustració 3. Imatge del videojoc *World of Warcraft*.

Segons la oferta de feina recuperada de Blizzard (2021), la base de dades que fa servir Blizzard és Oracle, és un sistema de gestió de base de dades relacional de tipus objecte-relació, desenvolupat per Oracle Corporation, preparada per suportar grans quantitats de dades i desenvolupar-se sense alentir el procés. També disposa de seguretat a l'hora d'emmagatzemar les dades. Per altra banda, fer ús d'aquest servei suposa pagar un preu elevat, més quan cal guardar una gran quantitat de dades.

En aquest cas, Blizzard Entertainment ha escollit un tipus de base de dades preparada per suportar una gran quantitat de dades, degut a que, *World of Warcraft* és un MMO (Multiplayer Massive Online), que es caracteritza principalment per disposar d'una gran quantitat de jugadors connectats simultàniament.

3.1.3 MongoDB

Drumgoole (2018) explica en el seu blog el cas de l'estudi SEGA HARDlight, desenvolupador de videojocs per mòbil. L'equip Inicialment, els jocs desenvolupats per aquest equip estaven destinats en un entorn d'un sol jugador sense funcionalitats que requerissin connectivitat a la xarxa, pel que no feien ús d'una base de dades externa. Però, segons les necessitats del mercat i el tipus de videojoc que es volia desenvolupar, va ser necessari fer ús d'una base de dades externa.

Es va realitzar un anàlisi de les bases de dades existents i l'equip HARDlight va arribar a la conclusió de fer ús de MongoDB. La raó principal, com explica Drumgoole (2018), va ser *"It was well known that MongoDB natively had the scalability and flexibility to handle large volumes"*.



Il·lustració 4. Imatge del videojoc *Sonic Forces*. Desenvolupat per SEGA HARDlight. Van determinar que la millor opció seria la base de dades MongoDB. És un model NoSQL i funciona amb MongoDB Atlas, un servei d'allotjament en el núvol que ofereix MongoDB. La base de dades que van escollir va ser MongoDB del tipus basat en documents ja que era el que millor encaixava per un videojoc basat en un compte per usuari. I també, perquè facilitava la tasca de la gestió i manteniment de la base de dades gràcies a l'habilitat de crear un gran volum de tràfic, i a la vegada disposar d'una gran escalabilitat.

Cada companyia ha escollit una base de dades diferent, a partir de les seves necessitats. També cal destacar que la informació referent a la base de dades utilitzada per una companyia de videojocs no es publica normalment, ja que es tracta d'informació confidencial.

3.2 Prototip

En la secció 6.1 s'explica l'elecció del videojoc per a dur a terme el prototip. En aquest cas, s'ha escollit un videojoc d'escacs asíncron, és a dir, per torns. Els dos jugadors

no estan presents al mateix temps durant la partida. És per això, que es realitzarà un anàlisi d'un joc d'escacs per mòbil i d'un videojoc amb que faci ús de partides asíncrones.



Figura 13: Imatge del menú principal de *Chess Royale King*.

El videojoc d'escacs escollit és *Chess Royale King*. Aquest joc de mòbil permet a l'usuari competir contra la IA, la intel·ligència artificial proporcionada pel propi videojoc. També disposa d'un mode per poder jugar contra un altre usuari de forma local en el propi dispositiu i d'un tercer mode anomenat "Puzzle" on es presenta un escenari d'una partida d'escacs i l'usuari l'ha de resoldre.

També disposa d'unes opcions per evitar que el dispositiu entri en mode dormir, això és per evitar que si s'està pensant el següent moviment i el jugador no toca el dispositiu, que no entri en mode repòs per falta d'interacció. A més a més, es pot canviar el color del taulell, com el patró de les caselles, que es mostri la última tirada realitzada, les coordenades del taulell, mostrant

els números i lletres als laterals i si es vol activar el so, entre altres opcions.



Figura 14. Imatge del mode de joc Jugador contra IA del videojoc *Chess Royale King*.

El mode per jugar contra la IA es juga per torns, seguint les normes del propi joc de taula. Al realitzar un moviment es mostra en text la casella objectiu del moviment, tant per la tirada del propi jugador com per la de la IA. Cada moviment es registra i es pot consultar en qualsevol moment de la partida.

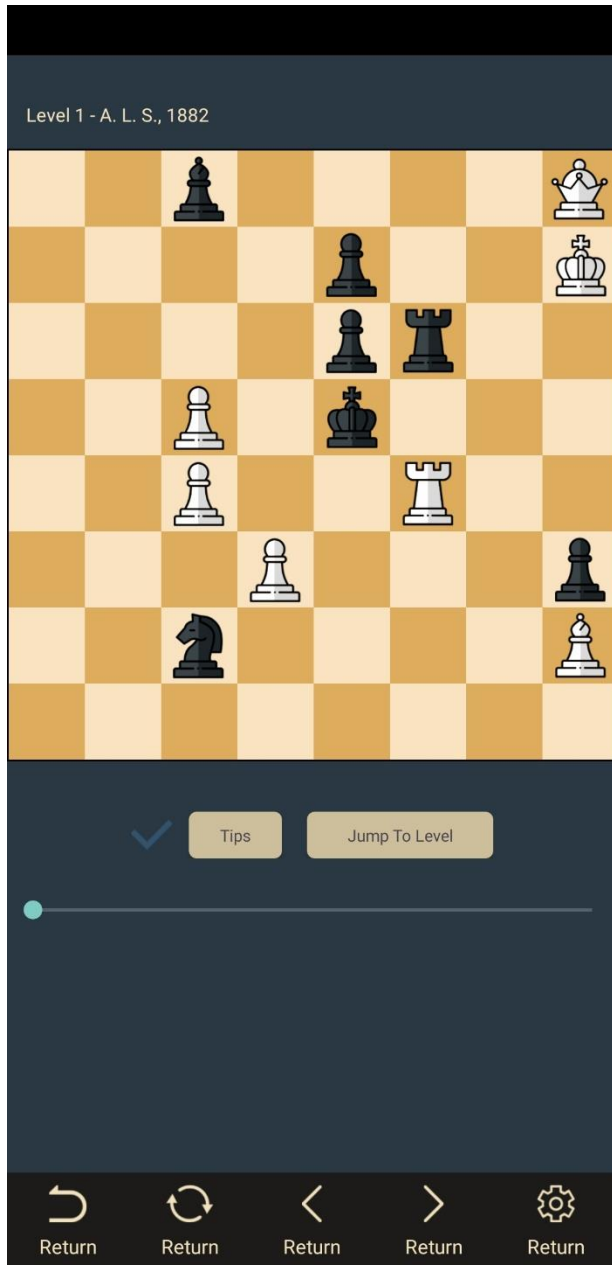
Al seleccionar una peça es marca la mateixa de color groc i també, del mateix color, les caselles disponibles per moure la peça.

Al seleccionar una nova partida o quan es pressiona el botó de "Settings" es mostren les opcions. Es pot seleccionar el color de peces que l'usuari portarà a la pròxima partida. També es pot escollir si es volen veure els moviments realitzats i de quina forma. A més a més, es pot escollir la dificultat de l'oponent, la qual s'escull seleccionat el temps màxim que té la IA per calcular la següent tirada. La opció predeterminada és de dos segons i la major dificultat disponible és de quinze minuts.

Per altra banda, es disposa d'una funcionalitat per gestos. Si es realitza un gest vertical, és a dir, es toca la pantalla verticalment cap a dalt, es gira el tauler per tenir el punt de vista de les peces que no es juguen i viceversa. Si es realitza un gest horitzontal, s'avança o es retrocedeix en els moviments realitzats durant la partida, per tant, si es realitza un gest cap a l'esquerra el tauler es col·locarà en l'estat que estava en el moviment anterior i si es realitza el gest cap a la dreta es tornarà a l'estat actual.

Aquesta funcionalitat no serveix per retrocedir i corregir la tirada, sinó per visualitzar les tirades anteriors.

El mode jugador contra jugador funciona igual que el mode de jugador contra la IA però en aquest cas els dos jugadors interaccionen en el mateix dispositiu, no es tracta d'una funcionalitat online.



Finalment, el mode trencaclosques presenta un escenari d'una partida a mitges on el jugador ha de resoldre la partida. Per resoldre la partida el jugador ha de realitzar les tirades establertes per finalitzar el trencaclosques. Només hi ha un "camí" de tirades correctes. A la part inferior es mostra una barra horitzontal que mostra el progrés, a la imatge es mostra a l'esquerra del tot que significa el principi.

Aquest videojoc no presenta cap funcionalitat que presenti l'ús d'una base de dades, de fet, no es connecta a la xarxa en cap moment. L'anàlisi d'aquest videojoc es realitza per conèixer les funcionalitats que pot presentar un joc basat en els escacs.

Figura 15. Imatge del mode de joc Puzzle del videojoc *Chess Royale King*.

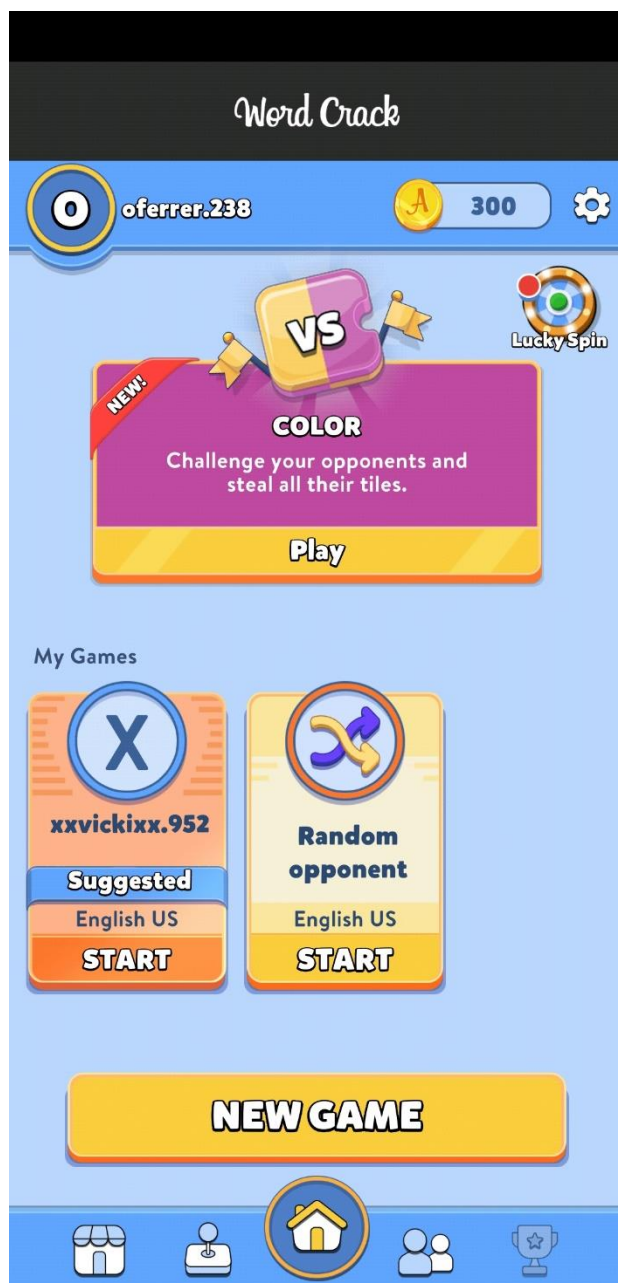


Figura 16. Imatge del menú principal del videojoc *Word Crack*.

El següent videojoc de mòbil per analitzar és *Word Crack*, en castellà *Apalabrados*. Aquest videojoc presenta la funcionalitat de partides per torns, partides asíncrones. Consisteix en partides per torns on els jugadors no han d'estar presents al mateix temps per jugar. L'usuari que ha de jugar realitza la seva tirada i és queda a l'espera. A partir d'aquest punt l'oponent veurà que és el seu torn i procedirà a continuar. D'aquesta forma, la partida avança més lentament. Però la funcionalitat encaixa amb la visió que es busca del videojoc, ja que cada moviment s'ha de pensar i d'aquesta forma s'elimina la pressió de tenir el contrincant present i esperant.

A la pantalla d'inici es mostra les partides actives, com es mostra a la figura no hi ha partides actives i, en aquest cas, es suggereix un contrincant i es disposa del botó per iniciar una nova partida. A l'iniciar-la es pot escollir l'idioma.

A l'hora d'iniciar sessió es pot introduir un correu o iniciar amb Facebook, en aquest cas és necessari tenir una base de dades que s'encarregui de gestionar els usuaris. De la mateixa forma, cada usuari ha de tenir guardades les seves partides i tota la informació relacionada. Quan es crea una nova partida el videojoc ha d'enviar les dades a la base de dades per realitzar la cerca d'un oponent nou i retornar les dades al joc perquè el jugador pugui accedir a la nova partida creada.

Un cop dins d'una partida el jugador disposa d'un llistat de lletres a la part inferior amb la qual ha de crear una paraula vàlida en l'idioma en que s'ha generat la partida. Per tant, cal que l'aplicació disposi d'un algoritme o un llistat de paraules per a realitzar les combinacions pertinents. A més a més, s'ha de tenir un llistat de paraules per a cada idioma disponible.



Figura 17. Imatge d'una partida del videojoc *Word Crack*.

dels dos dispositius realitzés aquesta tasca.

Es pot donar el cas que aquest llistat de paraules es guardi a la base de dades o que la pròpia aplicació guardi en format local les dades respecte l'idioma.

Quan es realitza una tirada el tauler s'actualitza de forma definitiva i l'usuari ja no el pot modificar més, degut a que ja no és el seu torn de joc. En aquest cas, l'oponent ha d'accedir a la partida per seguir amb el joc. Quan es realitza una tirada la base de dades ha d'actualitzar el registre de la partida, ja que l'oponent ha de veure en temps real el progrés de la partida.

Com que es tracta d'un model de videojoc multijugador però per torns. Fer servir una base de dades per dur a terme el guardat de les dades relacionades amb les partides és la solució més adequada.

Si fos el cas que les partides seguessin síncrones, és a dir, els dos jugadors competeixen en temps real, seria necessari un servidor amfitrió o que un

En el cas d'aquest videojoc, l'ús de la base de dades és imprescindible. La BBDD s'encarrega de guardar les dades de les partides i així cada jugador, des del seu dispositiu, pot veure en temps real l'estat de les seves partides actives.

3.3 Resum

En aquest Capítol s'han exposat referents de videojocs segons la base de dades que han fet servir i segons el tipus de joc en el que es basen i les seves mecàniques.

Per part de MySQL RiotGames ha indicat un seguit de punts a tenir en compte a la hora d'utilitzar una base de dades. Destaca la metodologia de migració de bases de dades com un element important a tenir en compte. El traspàs de dades entre bases de dades és un punt d'inflexió que ha de resultar sense problemes per assegurar les dades. De la mateixa forma que no és necessari inventar nous sistemes, actualment n'existeixen una gran quantitat ja funcionals i que poden cobrir els aspectes necessaris.

Seguint amb MySQL també s'analitza el cas de *Osiris: NewDawn*. L'equip encarregat de desenvolupar el videojoc es veu obligat a canviar a un sistema de guardat al núvol.

Per part d'Oracle s'analitza el videojoc *World of Warcraft*, el qual fa ús d'una base de dades que disposa d'una gran escalabilitat. És una funcionalitat adequada pel tipus de videojoc, ja que aquest contempla una gran quantitat de jugadors simultanis.

L'últim referent de les bases de dades és l'equip HARDlight el qual es veu obligat a adaptar-se a les noves exigències del mercat i introduir les bases de dades i les seves funcionalitats als seus videojocs.

Per acabar, s'han analitzat dos referents relacionats amb el joc de taula en que es basa el prototip i amb la mecànica bàsica. *Chess Royale King* és un joc per dispositius mòbils basat en els escacs mentre que *Word Crack* implementa les partides asíncrones.

4. Objectius

L'objectiu principal del projecte és realitzar el disseny i implementació d'una base de dades en el prototip d'un videojoc. Caldrà analitzar les bases de dades existents i escollir la més adequada, dissenyar una base de dades i un videojoc que faci ús de les funcionalitats de la base de dades, i finalment desenvolupar un prototip, la base de dades i realitzar la connexió. La BBDD s'haurà d'establir en un servidor extern per poder garantir l'ús de la mateixa des de qualsevol localització i dispositiu. El prototip es considerarà completat al disposar d'un executable del mateix amb un correcte funcionament.

Com a objectiu secundari es vol analitzar les bases de dades existents segons un conjunt de paràmetres per poder escollir la més adequada, depenent del videojoc que es vulgui desenvolupar. Ara bé, cada gènere de videojocs té les seves particularitats i, per tant, poden necessitar, en més o menys mesura, gestionar les seves dades. Si parlem d'un videojoc multijugador caldrà tenir una secció de gestió d'usuaris, una altra per guardar el progrés, etc. Si tracta d'un videojoc d'un sol jugador guardar les dades dels usuaris no seria necessari, en canvi, guardar el progrés sí que ho seria. D'aquesta forma, caldrà analitzar diverses dades relacionades amb els videojocs i amb les bases de dades per poder creuar-les i obtenir resultats que mostrin quines bases de dades s'adeqüen més a quins videojocs. Per la part dels videojocs s'haurà de determinar quines dades cal tenir en compte a l'hora de realitzar l'anàlisi i per part de les bases de dades quins elements marquen la diferència entre elles.

5. Disseny metodològic i cronograma

Per realitzar aquest projecte es seguirà la següent metodologia i cronograma.

5.1 Metodologia

La metodologia es divideix en diversos apartats que es duran a terme en el transcurs del desenvolupament del projecte.

5.1.1 Context

Primer de tot, aquest projecte es realitza com a treball final del grau de Disseny i Producció de Videojocs. Es busca l'aprenentatge i demostració dels coneixements sobre les bases de dades, Unity i els videojocs, tant teòrics com pràctics.

5.1.2 Recerca de les bases de dades

Es realitzarà una investigació sobre les bases de dades actuals més destacades i les seves propietats, és a dir, quins avantatges i inconvenients tenen i quines necessitats poden satisfer en un videojoc.

5.1.3 Anàlisi sobre les bases de dades i els videojocs

S'analitzaran les dades recollides, ja que, conèixer aquests elements serà necessari per poder determinar quina base de dades utilitzar en el videojoc a desenvolupar. Ja que l'objectiu del projecte és realitzar una demostració d'una connexió d'una base de dades caldrà escollir la més adequada.

L'objectiu secundari consisteix en obtenir les dades suficients de les bases de dades per poder determinar quina seleccionar segons el videojoc que es vulgui desenvolupar. Aquest objectiu es podrà completar durant la fase d'investigació i anàlisi.

5.1.4 Desenvolupament del prototip

S'investigarà com realitzar una connexió de la base de dades escollida amb el motor de jocs Unity. Per connectar-la caldrà contractar un espai web on poder allotjar la base

de dades escollida, per realitzar aquest pas s'haurà de cercar per la xarxa un servei que ho permeti. Caldrà crear un seguit de scripts que facin de pont en la connexió, generalment en el llenguatge de programació PHP o similar, aquests scripts estaran col·locats en el mateix servidor web que la base de dades i seran els encarregats d'interpretar les comandes que es rebin des del videojoc.

Per altra banda, s'haurà de desenvolupar un videojoc en fase de prototip per a realitzar la demostració de la connexió. Aquest prototip serà un joc senzill encarregat de contenir les funcionalitats necessàries per poder fer ús de la base de dades. Per començar, s'haurà de realitzar una fase de disseny del videojoc on es determinarà quin tipus de videojoc serà, quines mecàniques contindrà i en quin punt es farà ús de la base de dades i quins scripts seran necessaris per realitzar la connexió. A continuació, s'haurà de desenvolupar el prototip i preparar la base de dades en el servidor web. Tot seguit, s'hauran de crear els scripts que connectaran amb els situats en el servidor per poder accedir a la base de dades.

Finalment, segons el tipus de videojoc s'hauran de realitzar les proves pertinents, és a dir, si es tracta d'un videojoc multijugador caldrà realitzar proves des de diferents dispositius per demostrar que les dades es recullen de la base de dades i no són locals, en canvi, si es tractés d'un videojoc d'un sol jugador no faria falta diversos dispositius però si diverses localitzacions, per demostrar que la base de dades es troba en un servidor extern.

5.2 Cronograma

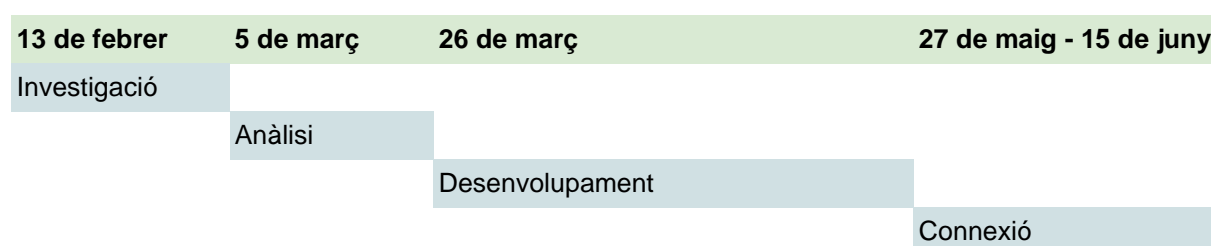
Per part del cronograma es disposen aproximadament de cinc mesos per realitzar el projecte, en aquest temps s'hauran de realitzar les següents fases: investigació de les bases de dades actuals, anàlisi de les bases de dades per determinar quina utilitzar i per completar l'objectiu secundari, disseny i desenvolupament d'un videojoc en fase prototip i connexió de la base de dades.

Aquestes fases es repartiran en el temps del que es disposa i es realitzaran en el següent ordre. Primer de tot, la fase d'investigació de la base de dades tindrà una

durada de vint dies, temps en el que s'haurà de recollir tota la informació necessària per realitzar les següents fases, la fase es considerarà completada quan es tingui prou informació per poder iniciar l'anàlisi. Tot seguit, es procedirà a analitzar tot el que s'hagi obtingut per determinar la base de dades a fer servir, aquesta fase serà també de vint dies aproximadament i tindrà com a objectiu escollir la base de dades a fer servir i completar l'objectiu secundari del projecte.

Un cop acabada la fase d'anàlisi general s'iniciarà el desenvolupament del prototip, es destinaran seixanta dies, dos mesos, a desenvolupar-lo. Primer de tot, s'haurà de dissenyar el videojoc de tal forma que pugui fer ús al complet de la base dades escollida. Un cop es consideri que el prototip és funcional es podrà passar a la següent fase. Es destinaran vint dies per trobar un lloc web on es pugui crear una base de dades i realitzar la connexió amb el prototip desenvolupat.

La durada de temps plantejada fins el moment és de quatre mesos, però se'n disposa de cinc, aquest mes restant està destinat a possibles imprevistos i a la preparació de la defensa del projecte. Al no conèixer amb total certesa quina base de dades es farà servir, ja que depèn de la investigació i anàlisi que es realitzarà, s'ha determinat que cal deixar un marge per problemes que podrien sorgir durant el procés.



Taula 2. Diagrama de Gantt de les fases del projecte. Font: Elaboració pròpia.

Fases	Investigació	Anàlisi	Desenvolupament	Connexió	Imprevistos
Dies	20	20	60	20	30
Data d'inici	13 de febrer	6 de març	27 de març	28 de maig	-
Data de fi	5 de març	26 de març	27 de maig	15 de juny	-

Taula 3. Taula de dates de les fases del projecte. Font: Elaboració pròpia.

Resumint, es disposa de cinc mesos per realitzar el projecte. Els primers vint dies seran destinats a investigar les bases de dades, els següents vint dies serviran per realitzar l'anàlisi de la informació recollida. El desenvolupament del prototip del videojoc es realitzarà en seixanta dies i, finalment, per la connexió amb la base de dades es destinaran vint dies. Es deixa un mes de marge per tal de cobrir possibles imprevistos, com dificultats sobrevingudes, o retards que es puguin produir en el desenvolupament del projecte.

6. Disseny del prototip

En aquest capítol s'exposa tot el contingut respecte al disseny i desenvolupament del prototip amb l'objectiu de completar la demostració sobre la connexió de la base de dades amb el motor de jocs Unity.

Una base de dades és una eina més dins del desenvolupament d'un videojoc, en aquest cas, s'han de determinar diferents aspectes. Quin videojoc es pot adequar més al projecte? Com es poden expressar les funcionalitats per crear una base de dades que pugui satisfer l'objectiu del projecte? Per començar, es determina el videojoc a desenvolupar.

6.1 Disseny del videojoc

S'ha escollit un videojoc basat en el joc de taula escacs. Però, primer de tot, de que tracta? Segons Soltis (2020) els escacs són un dels jocs de taula més antics i populars, jugats per dos oponents en un tauler amb unes peces, les quals tenen un disseny i un color determinat, acostumen a ser blanques i negres, per diferenciar el bàndol. Les blanques juguen primer i el torn de joc va canviant a mesura que cada jugador realitza un moviment. El primer jugador que aconsegueix forçar el rei enemic en un escac i mat guanya. Que consisteix en una posició en la que el rei enemic no pot evitar morir dins del joc.

Ara bé, aquest joc de taula s'ha d'adaptar a un videojoc que faci servir una base de dades. Primer de tot, cal exposar el seu funcionament. La part jugable del prototip consistirà en un usuari enfrontant un altre usuari en una partida asíncrona d'escacs. El primer jugador realitzarà un moviment en el seu torn i donarà per finalitzada la seva part de forma temporal, a partir d'aquí l'altre jugador haurà d'accedir al videojoc per continuar la partida amb el seu moviment i així successivament fins la finalització de la partida.

D'aquesta forma, un usuari podrà tenir diverses partides actives amb un oponent diferent en cada una, ja que es tracta de partides asíncrones, és a dir, els dos oponents no han d'estar presents al mateix temps per jugar. Aquest funcionament encaixa amb

l'essència dels escacs, perquè el joc de taula és un joc lent, que requereix anàlisi del tauler en cada moviment per poder predir el que farà l'oponent i trobar la millor resolució per a cada situació. En aquest cas, potenciar les pauses entre moviments amb partides asíncrones elimina qualsevol pressió a l'hora de pensar com realitzar el següent moviment.

Per entendre el funcionament del videojoc es disposa dels següents diagrames. El primer fa referència a la navegació de finestres en el videojoc.

6.2 Navegació

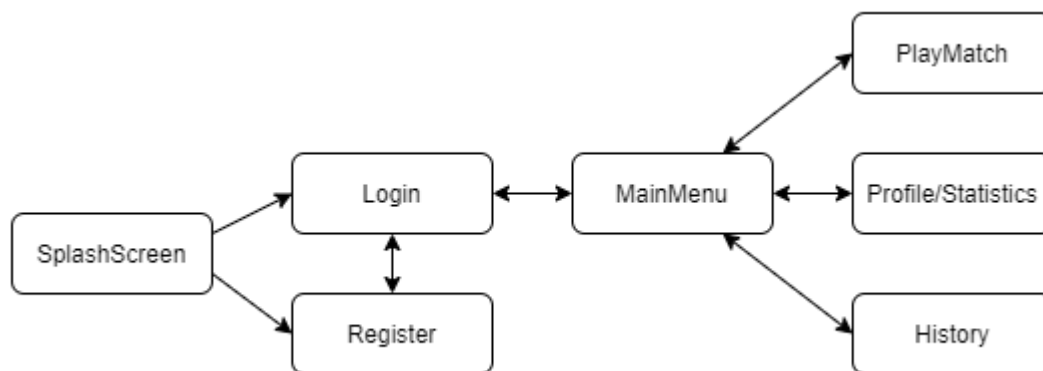


Figura 18. Diagrama de navegació. Font: Elaboració pròpia.

La primera finestra és la presentació (Splash screen), on es mostra el nom del videojoc, una imatge i l'accés al registre i a l'inici de sessió. A continuació, es disposa del registre. En aquesta pantalla s'han d'introduir les dades bàsiques d'usuari, com el sobrenom, el correu electrònic i la contrasenya. Un cop introduïdes les dades correctament es pot dur a terme la creació de l'usuari que es farà servir dins del videojoc.

Per altra banda, hi ha l'inici de sessió, en cas que ja es disposi d'un usuari ja creat s'ha d'accedir des d'aquesta finestra. S'ha d'introduir el correu electrònic i la contrasenya per accedir al menú principal.

En el menú principal, es mostra el llistat de partides. Cada element de la llista indica una partida activa amb un rival, on s'especifica el nom del rival i quin color de peces es controla. Aquest llistat de partides es divideix en dues seccions segons si es el torn

de l'usuari o del rival, d'aquesta forma es pot identificar ràpidament quines partides actives requereixen del moviment de l'usuari per continuar.

En el perfil es troben les dades d'usuari, es poden consultar en qualsevol moment accedint des del menú principal. Aquestes dades mostren la informació utilitzada a l'hora del registre, a més a més, hi ha l'opció d'un canvi de contrasenya. A part de les dades del perfil hi ha una secció d'estadístiques mostrant les partides jugades, guanyades i perdudes.

Una altra finestra és l'historial, es mostren dades de les partides ja acabades, indicant el sobrenom del rival, el número de moviments jugats, qui és el guanyador i quin bàndol es jugava, és a dir, el color de les peces.

Un cop s'escull la partida a jugar s'accedeix al tauler on es mostren les peces col·locades en el torn de joc actual. El jugador només pot moure les peces del seu color i un cop acabat el seu torn no es pot modificar el tauler fins que el rival hagi finalitzat el seu torn.

6.3 Casos d'ús

El diagrama de casos d'ús està dividit segons els apartats generals del videojoc. El primer indica els casos d'ús segons les funcionalitats de l'accés.

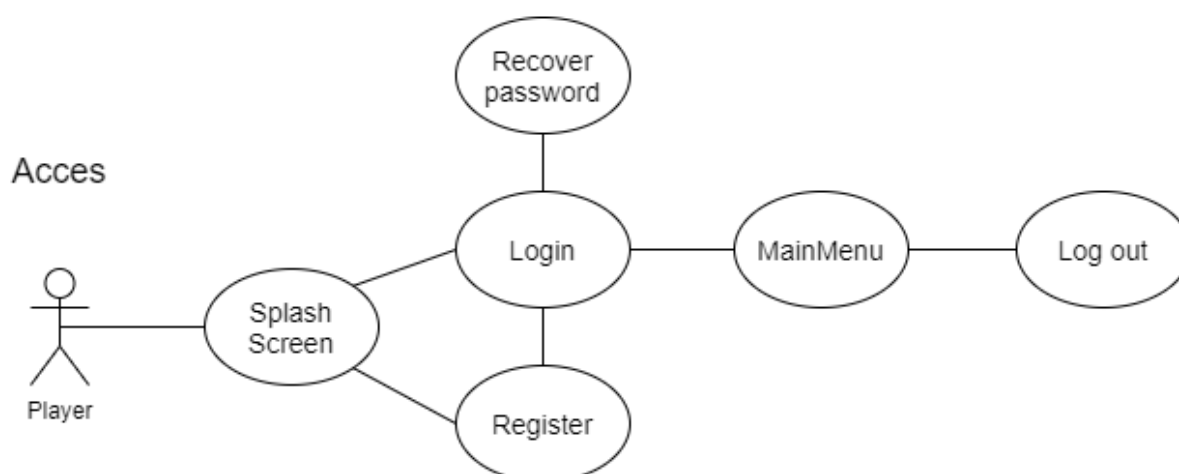


Figura 19. Diagrama de casos d'ús segons l'accés. Font: Elaboració pròpia.

Les funcionalitats que no s'han especificat en l'apartat de la navegació són les següents. Des de la pantalla d'inici de sessió hi ha l'opció de recuperar la contrasenya, en cas que l'usuari se n'oblidi podrà utilitzar aquesta funcionalitat per rebre una nova contrasenya temporal al correu electrònic introduït durant el registre.

Per altra banda, si es dona el cas que es vol iniciar sessió amb un altre usuari, des del menú principal es pot tancar sessió, tornant a la finestra de presentació, a partir d'aquest punt l'usuari és lliure d'iniciar sessió amb un altre usuari o dur a terme un registre.

El segon diagrama de casos d'ús indica la funcionalitat centrada en el joc en sí.

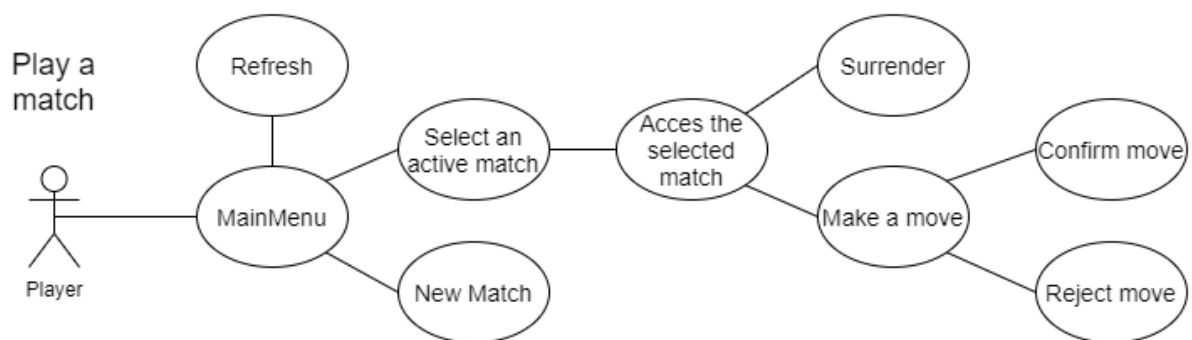


Figura 20. Diagrama de casos d'ús segons la jugabilitat. Font: Elaboració pròpia.

Tornant al menú principal, es disposa de la funcionalitat de refrescar les partides actives, per poder veure en temps real si un rival ha completat el seu moviment, d'aquesta forma s'evita que l'usuari hagi de tancar i obrir sessió per poder veure els canvis en temps real en el llistat de partides. Des del menú principal també hi ha la funcionalitat de generar una nova partida, en aquest cas, es buscarà un rival que no coincideixi amb cap rival d'una partida activa, és a dir, no es pot tenir dues partides actives contra el mateix usuari. L'usuari que activi una partida serà el que jugui les peces blanques.

Un cop dins d'una partida activa seleccionada hi han diverses funcionalitats a tenir en compte. La primera consisteix en la rendició, els escacs són partides llargues i d'anàlisi, per tant, es pot conèixer amb antelació quan una partida no es podrà guanyar, i si el jugador es veu forçat a acabar la partida o simplement a abandonar-la

pot generar frustració, per tant, donar l'opció de rendició i tancar la partida és la millor opció per aquests casos.

Per evitar errors es disposa d'una finestra en forma de "pop-up"³ que es mostra al realitzar un moviment, aquesta finestra serveix per confirmar o denegar el moviment realitzat. En cas que es confirmi el moviment, el tauler queda bloquejat, és a dir, el jugador no podrà modificar el tauler, ja que, el seu torn haurà acabat. Un cop confirmat el moviment s'actualitzarà la base de dades amb la nova disposició de les peces. Per altra banda, si es denega el moviment, el tauler passarà a l'estat anterior a que es realitzés el moviment en qüestió.

El tercer diagrama de casos mostra les funcionalitats centrades en les dades dins l'aplicació.

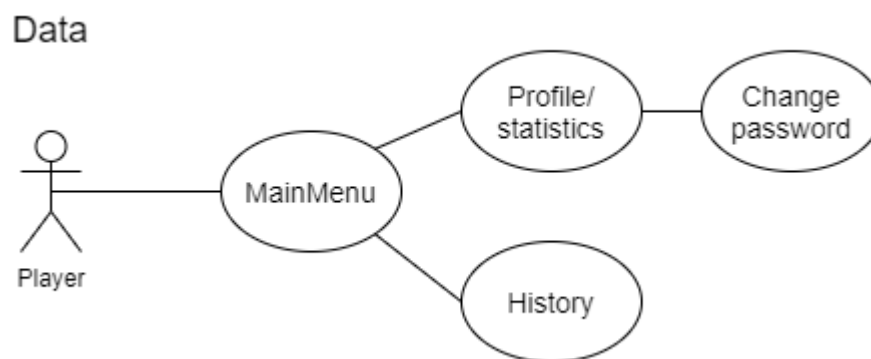


Figura 21. Diagrama de casos d'ús segons les dades. Font: Elaboració pròpia.

En aquest diagrama es mostren les finestres de perfil i historial, a les dues s'hi accedeix des del menú principal. En el perfil hi ha l'opció de canviar la contrasenya, es demana introduir la nova contrasenya i la actual, per poder confirmar la identitat de qui la canvia.

6.4 Diagrama de classes

El diagrama de classes també es troba dividit en diferents parts. Per començar hi ha el diagrama de classes de les dades de l'usuari. Aquest diagrama té com a base la

³ Finestra emergent.

classe estàtica Data, la qual permet mantenir la informació entre escenes. Ja que totes aquelles dades que no siguin guardades de forma local, fent servir player prefb o en classes estàtiques, es perden al canviar d'escena. La classe conté les dades sobre l'usuari i un llistat de l'historial de partides acabades sobre el mateix jugador.

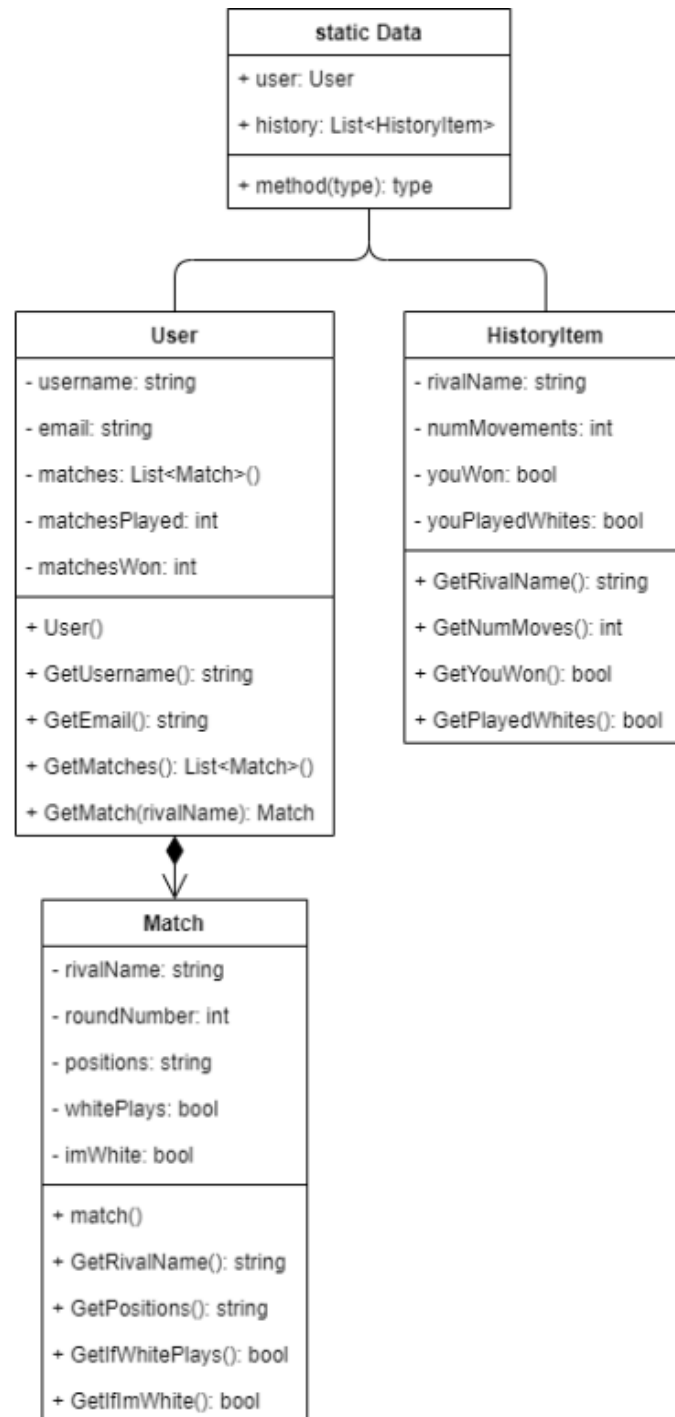


Figura 22. Diagrama de classes de les dades d'usuari. Font: Elaboració pròpia.

Tot seguit, la classe User gestiona les dades directes de l'usuari, que són el sobrenom, el correu electrònic i el llistat de partides actives. Com a mètodes de la classe hi ha el constructor, el get de cada característica anterior i el get segons el nom del rival d'una partida.

A continuació, la classe Match conté les dades sobre una partida. El nom del jugador rival, el número de rondes jugades fins el moment, una cadena string amb les posicions de les peces en el tauler, un booleà que indica si es el torn de les peces blanques i un segon booleà que indica si el jugador controla les peces blanques. Els mètodes de la classe són el constructor i els Get corresponents, encarregats de retornar les dades.

Finalment, la classe HistoryItem guarda un element del llistat de partides acabades de l'historial. Aquestes dades es guarden fora de la classe User ja que són partides que ja s'han acabat i no cal que tinguin una relació directa amb l'usuari, al tractar-se d'un històric. Les seves característiques són el nom del rival, el número de moviments totals fins la resolució de la partida, si el jugador ha guanyat i quin color de peces ha jugat. Els mètodes són els Get de les dades anteriors.

A continuació, la Figura 23 mostra el diagrama de classes basat en la part jugable del prototip. La primera classe és el GameManager, s'encarrega de la gestió general del projecte, guarda una referència de la classe BoardManager i conté el mètode utilitzat pel canvi d'escena. Cal destacarà que el GameManager es troba present en totes les escenes però s'ha inclòs només en el diagrama de classes de la part jugable perquè és on participa més.

Tot seguit, trobem el BoardManager, aquesta classe s'encarrega de gestionar tot allò relacionat amb el tauler. Guarda dades de cada casella, dos llistats de les peces, un per cada color, la seqüència de peces actual i la seqüència establerta per una nova partida. Aquestes seqüències guarden les posicions de cada peça en el tauler. Com a exemple, la següent informació correspon a una peça: A0W4. La primera lletra indica la columna, un tauler és de 8x8, per tant les lletres van des de la A fins la H, el següent caràcter és un número que va des del 0 fins el 7, indicant les files, el tercer caràcter pot ser W o B i indica si el color és blanc o negre en anglès. Finalment, el quart caràcter indica amb un número de quina peça es tracta, 0 el rei, 1 la reina, 2 l'alfil, 3 el cavall,

4 la torre i 5 el peó. Dins de la seqüència la informació de cada peça es troba separada per una barra baixa: A0W4_B0W3.

Aquestes seqüències només indiquen informació de les caselles ocupades, si no es menciona una casella es dona per fet que es troba buida.

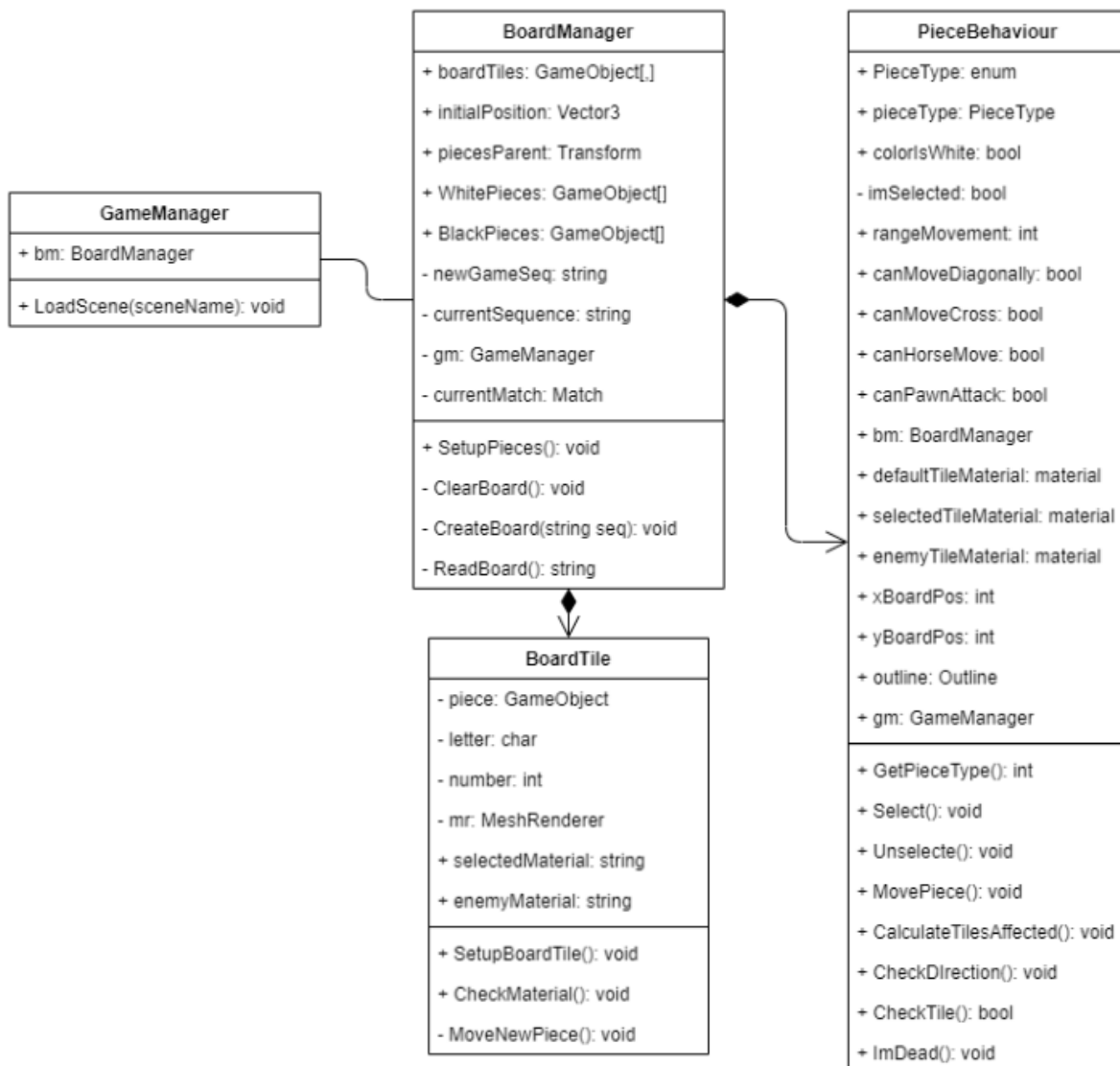


Figura 23. Diagrama de classes de la part jugable. Font: Elaboració pròpia.

Els mètodes de la classe BoardManager són els següents. Primer de tot, SetupPieces, crida els mètodes ClearBoard i CreateBoard en aquest ordre, explicats a continuació. ClearBoard elimina totes les peces del tauler i la classe CreateBoard llegeix la seqüència escollida i s'encarrega de col·locar les peces en el tauler, relacionant cada casella amb la seva peça que l'ocupa. Finalment, ReadBoard observa el tauler i

genera la seqüència segons la posició actual de les peces, d'aquesta forma es podrà guardar la disposició del tauler a la base de dades.

La següent classe és BoardTile, gestiona la informació de cada casella del tauler, que en aquest cas és la peça assignada, si en té, i la posició que li pertoca dins del tauler, que són els dos primers caràcters de la informació d'una peça en la seqüència. El mètode SetupBoardTile és el constructor. CheckMaterial canvia el color de la casella depenent de si la peça seleccionada pel jugador hi pot interaccionar. Si la casella canvia a verda vol dir que la peça que té seleccionada s'hi pot moure, si canvia a vermell vol dir que pot eliminar la peça que es troba en aquesta casella. L'últim mètode MoveNewPiece realitza el moviment de la peça segons el color canviat al mètode anterior, és a dir, si el jugador fa clic sobre una casella verda la peça seleccionada es mourà cap a la casella verda indicada.

La última classe del diagrama és PieceBehaviour, s'encarrega de controlar el comportament de cada peça. El primer atribut és un enumerador que indica el tipus de peça que és. La resta d'atributs assignen el comportament a l'hora de jugar. El primer atribut indica quin color és, en cas que la casella es trobi seleccionada. L'atribut FirstMovement serveix per indicar que el primer moviment d'un peó pot ser doble. També s'indica el rang de moviment de les peces i els últims quatre booleans estableixen quin tipus de moviment realitza. Els mètodes de la classe són els següents: primer de tot GetTypePiece retorna amb un número el tipo de peça que és. A continuació hi han els dos mètodes per seleccionar i deseleccionar una peça en el tauler, quan es selecciona una se li col·loca un contorn groc a la peça i les caselles disponibles al voltant se'ls hi canvia el color.

El tercer diagrama, Figura 24, fa referència a la base de dades. Mostra els scripts que es fan servir des del motor per interaccionar amb la base de dades. El primer i més important és DatabaseManager, conté la connexió amb la base de dades i dins de la jerarquia ocupa el paper de pare.

La resta d'scripts hereten la capacitat de connectar-se amb la base de dades i de fer servir el mètode VerifyInputs per verificar els espais que ha d'omplir l'usuari, per exemple, el correu i la contrasenya durant l'inici de sessió, per desbloquejar el botó corresponent per dur a terme la funcionalitat.

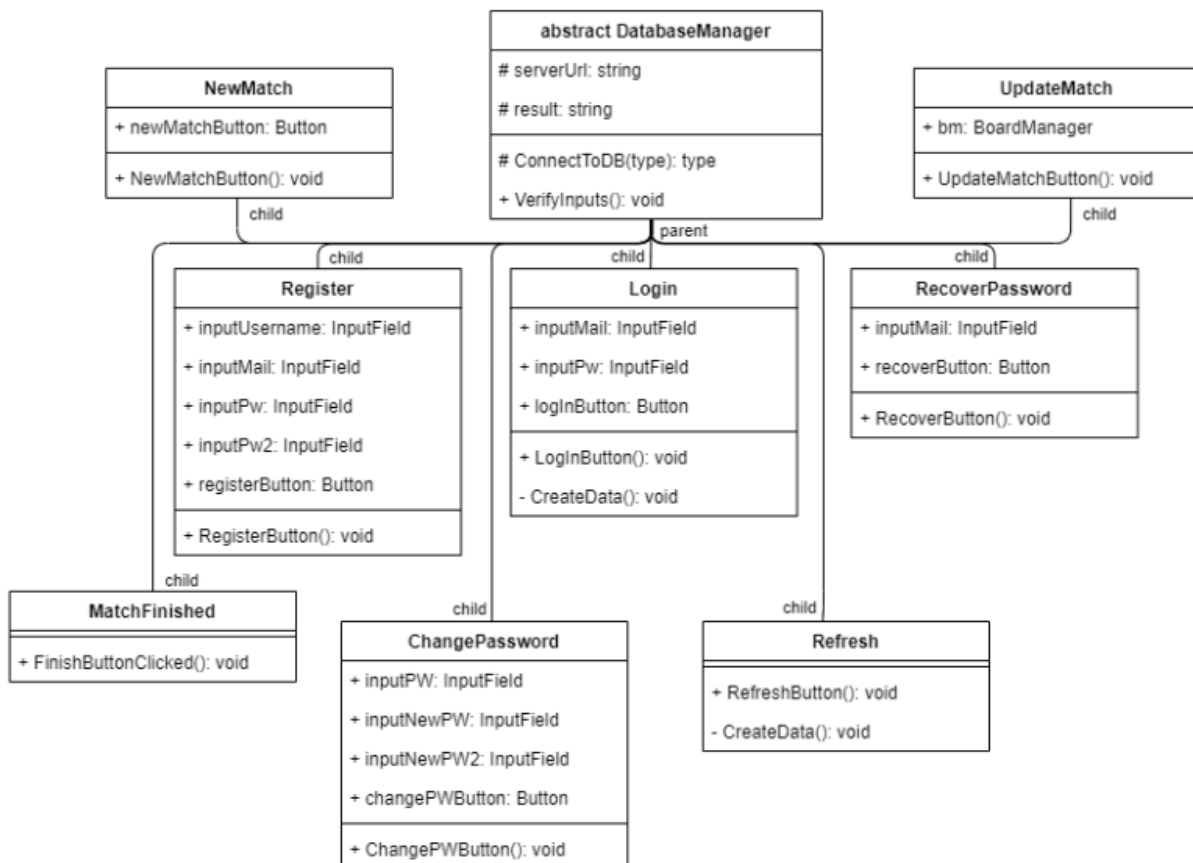


Figura 24. Diagrama de classes de la connexió amb la base de dades. Font: Elaboració pròpia.

Cada un dels scripts fills guarda una referència als camps que s'han d'omplir i al botó per desbloquejar-lo un cop s'hagin omplert tots els camps. L'script Login té el mètode CreateData, el qual s'encarrega de crear una instància de la classe usuari amb les dades rebudes de la base de dades un cop s'ha verificat que les dades de l'usuari que es vol connectar són correctes. Un cop creades les dades es guarden a la classe estàtica Data.

I com a últim diagrama de classes, representat en la Figura 25, es disposa d'una agrupació de les classes que no es troben directament relacionades entre si. Les classes SetupMainMenu, SetupHistory i SetupMyAccount tenen un funcionament similar, la diferència és que operen en diferents escenes. El funcionament consisteix en recollir dades ja guardades i col·locar-les en l'escena.

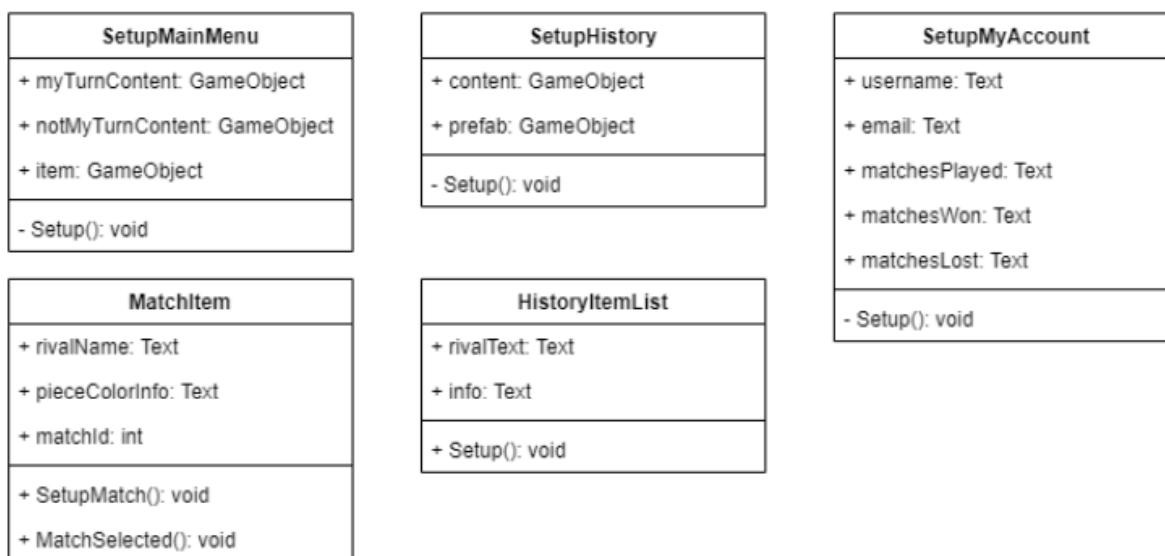


Figura 25. Diagrama de classes de la resta de les classes. Font: Elaboració pròpia.

En el menú principal es col·locaran les partides actives en dues llistes, segons si és el torn del jugador o del rival, és per això que la classe SetupMainMenu té dos atributs de content. També cal guardar el item, que és el prefab de la estructura de panells que es vol col·locar. El mètode Setup s'encarrega de dur a terme el procés.

Per altra banda, MatchItem i HistoryItemList són els scripts que formen part dels objectes individuals de cada llista. Són els prefabs i cada un d'aquests scripts configura l'objecte quan s'afegeix a la llista. Per exemple: quan es realitza el SetupMainMenu es col·loquen a les llistes les partides actives, cada partida activa és un MatchItem, el qual col·loca el seu nom del rival i el color de peça que juga.

6.5 Disseny de la base de dades

En aquest apartat s'exposa el diagrama de la base de dades que es fa servir en el prototip. S'ha escollit el model relacional, explicat en la secció 2.3, per crear el diagrama, ja que és el que mostra un model més preparat per a la implementació a la base de dades.

Aquesta base de dades disposa de quatre taules principals de dades i de dues auxiliars creades per la seva relació.

La taula User indica les dades de l'usuari que són el sobrenom, la contrasenya, dividida en el hash i el salt explicats en següent apartat 6.6, i el correu electrònic. Tant el sobrenom com el correu electrònic seran de caràcter Unique, que consisteix en una Constraint de la base de dades indicant que no es pot repetir. La següent taula és MatchGame, indica les dades de cada partida activa que té l'usuari, en aquesta taula es guarda el nom de l'usuari rival, el número de rondes jugades, la seqüència de posicions de les peces i si és el torn de les blanques. Ja que la relació entre la taula User i la taula MatchGame és n-n és necessari tenir una taula central User_MatchGame que reculli la relació entre taules. En aquesta taula es guarden les id's de cada usuari i partida, i quin color juga cada usuari.

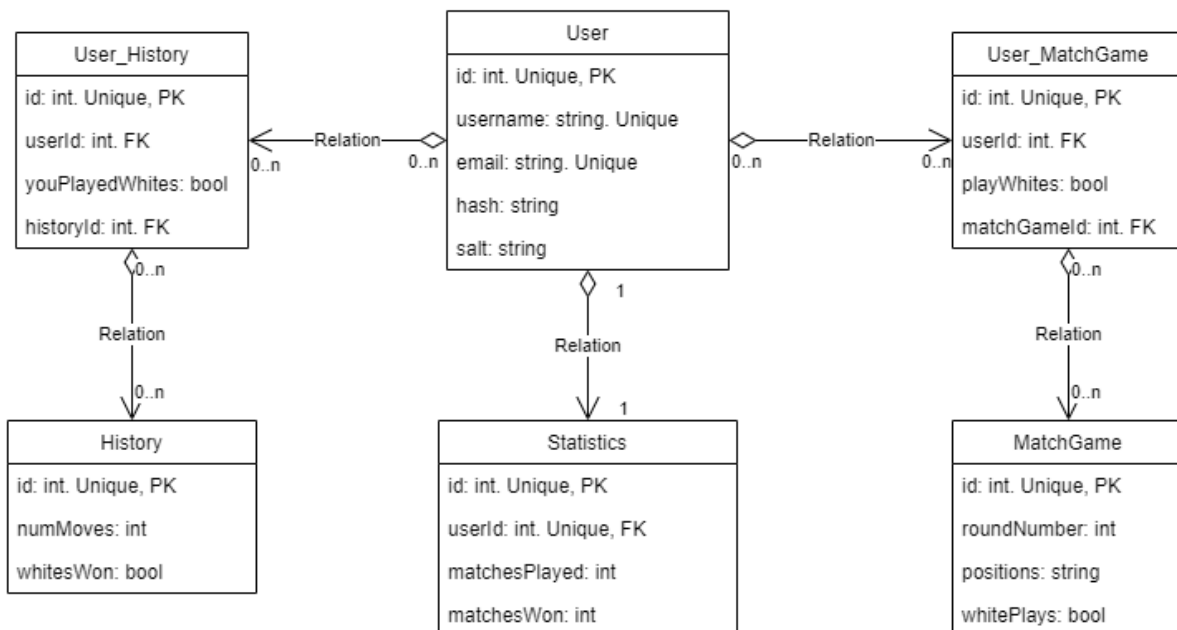


Figura 26. Diagrama de la base de dades. Font: Elaboració pròpia.

Tot seguit, la taula Statistics emmagatzema les dades relacionades amb les estadístiques de l'usuari, les quals són les partides jugades i guanyades, les perdudes es poden calcular segons les dades anteriors, per tant, no fa falta guardar-les.

La última taula History guarda l'històric de partides acabades, amb el número de moviments fins la finalització i quin color ha guanyat la partida. En el cas d'aquesta relació, també cal tenir una taula central, que guarda l'identificador dels dos usuari implicats, l'identificador de la partida de l'històric i quin jugador ha jugat quin color de peces.

6.6 Connexió amb la base de dades

L'ús d'una base de dades implica disposar d'un allotjament per la mateixa, aquest es situa en una localització externa al dispositiu que fa ús del videojoc. És per això que s'ha fet ús del servei gratuït *000webhost* per allotjar una base de dades. Aquest lloc web permet utilitzar una base de dades allotjada en els seus servidors i ofereix un espai on col·locar scripts enfocats a fer d'intermediaris entre el videojoc i la BBDD.

Els scripts amb la funcionalitat de pont s'han creat en el llenguatge PHP, llenguatge enfocat principalment en el desenvolupament web.

Segons la base de dades explicada en la secció 6.5 i les funcionalitats del prototip explicades en les seccions 6.2 i 6.3 s'han desenvolupat sis scripts enfocats a complir la comunicació entre ambdues parts.

Aquest conjunt d'scripts tenen en comú un seguit d'aspectes. El primer és començar amb la connexió a la base de dades, pel que es necessita la direcció "localhost", aquesta direcció és la que es fa servir quan la base de dades es troba situada al mateix servidor que els scripts. Tot seguit, el nom de la base de dades a la que es vol accedir, el nom d'usuari i la respectiva contrasenya. Les últimes dades les proporciona el propi lloc web des d'on es gestiona la localització dels scripts i la base de dades. El següent aspecte en comú és tancar la connexió prèviament feta quan s'acaba l'ús de la mateixa, que acostuma a ser al final de l'script. Com a tercer aspecte, en cada script es reben dades des del videojoc, com per exemple, el nom que fa servir l'usuari en l'aplicació, a partir d'aquestes dades es realitzen consultes, que poden ser dels quatre tipus: select, insert, update i delete. La realització de les consultes és el quart aspecte, el qual s'explica a continuació.

Cada script fa referència a una funcionalitat de la base de dades per part de l'aplicació. La primera de totes és el registre.

L'script *registre.php* recull un seguit de dades enviades des del videojoc. Comprova que dades com el correu electrònic o el nom utilitzat en l'aplicació no es repeteixin, i si no és el cas es realitza una consulta insert a la taula User i un segon insert a la taula Statistics. És a dir, es crea l'usuari i les dades de les estadístiques referents al nou

usuari, aquestes estadístiques seran de zero partides jugades ja que es tracta d'un nou usuari. Com a particularitat en el registre s'encripta la contrasenya fent ús del concepte "salt" i "hash". Ambdós són conjunts de caràcters enfocats a la protecció de dades. "Salt" és un string format per caràcters aleatoris i sense cap tipus de sentit entre ells, per generar aleatorietat i diferenciació. Per a cada usuari s'agafen números aleatoris i la data del moment. A partir d'aquesta seqüència aleatòria es recull la contrasenya proporcionada per l'usuari i el "salt", aleshores s'encripta la contrasenya fent ús del "salt" com a clau d'accés. El resultat és el "hash".

El *login.php* s'encarrega de l'inici de sessió i consisteix en enviar les dades d'accés al servidor, que són el correu electrònic i la contrasenya. Es realitzen comprovacions i es retornen les dades completes de l'usuari. Primer de tot, es comprova que el correu electrònic existeix, si és el cas es passa a comprovar si la contrasenya introduïda és igual a la emmagatzemada a la base de dades. En aquest cas, no es desencripta la contrasenya de la base de dades per a fer la comprovació, sinó que s'encripta la proporcionada per l'usuari i es compara amb el "hash" guardat, és a dir, a partir del "salt" de la base de dades es crea el "hash" d'accés i es comprova amb el "hash" de la base de dades, d'aquesta forma, en cap moment es coneix quina contrasenya exacta és la que està emmagatzemada a la base de dades, ja que la contrasenya proporcionada per l'usuari no es guarda.

Un cop s'han realitzat les comprovacions necessàries per procedir amb l'inici de sessió de l'usuari, és a dir, s'han introduït les dades d'accés correctes, es recullen totes les dades necessàries de l'usuari per enviar-les al videojoc. Es recupera el nom de l'usuari, les dades relacionades amb les partides actives, les estadístiques que es mostraran en el perfil i les dades que formen l'històric de les partides acabades. Aquestes dades es transformen en una cadena string que s'envia al videojoc.

El següent script és *newmatch.php*, el qual s'encarrega de generar una nova partida per part del servidor. Primer de tot, es busca un usuari rival que no tingui una partida activa amb l'usuari que envia la sol·licitud. Un cop es selecciona l'usuari rival es duu a terme una consulta insert per introduir la nova partida creada. A continuació es realitza un segon insert amb dos valors, un per a cada jugador implicat relacionant-los

en la taula "User_MatchGame" de la Figura 26. Finalment, s'envien les dades d'aquesta nova partida al videojoc perquè no faci falta actualitzar les dades completes.

L'script *updatematch.php* entra en funcionament quan un usuari confirma un moviment realitzat en una partida activa. S'envien les dades: identificador de la partida, posició de les peces, número de moviments realitzats i si es juguen les blanques. Es realitza la consulta `update` per actualitzar les dades de la partida de la taula "MatchGame".

A continuació hi ha l'script *matchfinished.php*. Aquest script es fa servir quan un usuari acaba una partida. Des del videojoc, s'envien les següents dades: l'identificador de la partida, el número de moviments totals, si han guanyat les peces blanques, el nom de l'usuari que ha jugat les peces blanques i el mateix amb les peces negres. Tot seguit, és realitza un insert per crear un nou registre a la taula "History", ja que es tracta d'una partida acabada. Per tant, es realitza un insert amb doble valor a la taula "User_History" relacionant el nou registre de l'històric amb els dos usuaris implicats en la partida, en aquest registre s'indica quin usuari ha jugat quin color de peça. A continuació, es recullen les dades de les estadístiques dels dos jugadors per actualitzar-les amb la consulta `update` degut a la partida acabada. Finalment, un cop s'han actualitzat i creat tots els registres necessaris, s'esborra amb la consulta `delete`, la partida acabada i els registres relacionals de la taula "User_MatchGame".

L'últim script és *refresh.php*, se'n fa ús quan l'usuari vol refrescar les partides actives des del menú principal. Aquest script realitza el mateix procés de recollida de dades de l'script *login.php*. A partir, del nom d'usuari proporcionat des del videojoc recull les dades relacionades amb les partides actives, les estadístiques i l'històric, es transformen en una cadena string i s'envien de retorn cap al videojoc.

6.7 Resum

En aquest Capítol s'ha explicat el disseny del prototip del videojoc. Tracta d'un videojoc basat en el joc de taula escacs. La mecànica principal són les partides asíncrones, és a dir, per torns, on cada jugador realitza el seu torn per separat. D'aquesta forma cada jugador pot tenir diverses partides actives a la vegada.

També s'ha exposat el disseny de la base de dades formada per sis taules per cobrir les necessitats de que es presenten: dades de l'usuari, de les partides actives, de l'historial i de les estadístiques.

Finalment, el disseny explicat anteriorment comporta un videojoc i una base de dades externa. Aquests dos aspectes s'han de connectar a través d'un conjunt d'scripts explicats en la secció 6.6.

7. Resultats

7.1 Resultats de la investigació

En aquest capítol es mostren els resultats de la investigació realitzada, enfocada a completar l'objectiu secundari proposat. Aquest objectiu consisteix en analitzar les bases de dades existents segons un conjunt de paràmetres a determinar per poder escollir la més adequada, com s'explica al Capítol 4.

S'han escollit un conjunt de bases de dades, un seguit de paràmetres a tenir en compte i s'ha realitzat un anàlisi de les BBDDs. Per altra banda, s'ha realitzat un procés similar amb el camp dels videojocs. S'han escollit un conjunt de gèneres de videojocs i unes funcionalitats que fan ús de les bases de dades, per poder analitzar com funciona cada gènere i poder obtenir dades relacionades entre bases de dades i videojocs.

7.1.1 Anàlisi de les bases de dades

A partir de les bases de dades seleccionades com les més destacades de 2019 segons ExploreGroup (2019) s'ha realitzat la següent investigació i anàlisi.

Per realitzar l'anàlisi s'ha recollit informació de les respectives pàgines oficials de cada base de dades, a més a més, SoftwareAdvice (2021) proporciona una comparació dels sistemes de gestió de bases de dades com l'opinió dels usuaris de les mateixes. També s'ha fet ús de la informació proporcionada per Altexsoft (2019).

Primer de tot, s'han seleccionat els següents paràmetres a tenir en consideració per a cada BBDD. La seva accessibilitat, el preu, la corba d'aprenentatge, el rendiment, si disposa de servei núvol, si és Open Source, si el servidor té una localització externa, si és escalable, la rapidesa i si compleix les característiques ACID.

- Accessibilitat: determina en quin grau qualsevol persona pot fer ús de l'objecte o funcionalitat que es vol definir, en el cas de les bases de dades

és si disposen de característiques o funcionalitats que facilitin el seu ús i l'accés a les mateixes.

- Preu: cada base de dades disposa de plans de pagament on ofereixen noves funcionalitats o les mateixes millorades, depenent del preu i del què s'ofereix.
- Corba d'aprenentatge: consisteix en la dificultat que suposa aprendre el funcionament de la base de dades en consideració, a més a més, es té en compte si es necessita algun coneixement previ.
- Rendiment: consisteix en com s'optimitza l'ús dels recursos per incrementar la eficiència dels procediments i reduir el cost dels mateixos, aconseguint la màxima càrrega de treball disponible per la base de dades.
- Núvol: si la base de dades disposa d'un servei d'emmagatzematge en la xarxa més enllà de la base de dades en sí.
- Open source: si la base de dades és gratuïta i de codi lliure. Tenint en compte l'apartat del preu, s'ha de remarcar que les bases de dades de codi lliure disposen d'una versió inferior gratuïta o limitada, en aquest cas si es vol una versió completa és necessari contractar un dels plans de pagament.
- Extern: tracta de si l'emmagatzematge és extern al videojoc. Pot ser de forma local i que es guardin les dades en arxius junt a la pròpia aplicació o que la base de dades es trobi en un servidor extern, a la xarxa.
- Escalabilitat: indica el grau en que pot créixer una base de dades a partir del hardware. Es pot realitzar de dues formes. Si escala horitzontalment vol dir que s'afegeixen nous hardwares per treballar en la base de dades, en canvi, si escala verticalment, es millora el hardware existent.
- Segur: depenent del tipus de dades és necessari assegurar la seva integritat, sobretot si es tracta de dades sensibles.
- Ràpid: si la base de dades en qüestió assegura una velocitat alta a l'hora de realitzar els procediments.
- ACID: si es compleixen les característiques ACID explicades en la secció 2.2.

La Taula 4 presenta la primera part de la taula de característiques de les bases de dades seleccionades. Una part de les característiques estan marcades amb una creu

o un guió, en aquest cas, la característica pot estar present o no. Si és que sí s'ha col·locat una creu, en cas contrari un guió. Per la resta de característiques, s'ha col·locat un número de l'un al cinc, és a dir, de menor a major, segons la característica que estigui definida. Per exemple, en el cas del preu com major sigui el número més reduït i competitiu serà el preu.

Base de dades	Corba			Open			
	Accessibilitat	Preu	d'aprenentatge	Rendiment	Núvol	source	Extern
Oracle	4	3	4	4	x	x	x
MySQL	4	1	5	5	x	x	x
Microsoft SQL Server	3	2	5	4	x	x	x
PostgreSQL	4	5	4	2	x	x	x
MongoDB	5	2	3	5	x	x	x
IBM Db2	3	3	1	4	x	x	x
Elasticsearch	4	5	3	3	x	x	x
SQLite	5	4	5	5	-	x	-
Cassandra	4	4	2	4	x	x	x

Taula 4. Característiques de les bases de dades. Primera part. Font: Elaboració pròpia.

De les bases de dades escollides la que presenta una menor accessibilitat és IBM db2, ja que, les aportacions en aquest aspecte són molt reduïdes. La resta de bases de dades tenen un accés més ben definit, com pot ser en el cas d'Oracle que disposa d'un programa d'accessibilitat per facilitar l'accés a les persones amb dificultats.

Per altra banda, en el cas del preu hi han moltes diferències. Les bases de dades disposen de plans de pagament on ofereixen més característiques com més elevat és el preu, en canvi, com que totes les BBDD són Open Source totes disposen d'un versió inferior i/o limitada gratuïta. Algunes de les característiques que s'ofereixen amb els plans són capacitat d'emmagatzematge per guardar les dades recollides a la base de dades, quantitat de registres que es poden guardar, quantitat d'aplicacions que es poden tenir enllaçades a les bases de dades, disposar de suport tècnic, etc.

En quant a la corba d'aprenentatge es tenen en compte dos aspectes. El primer fa referència a la dificultat que suposa aprendre el funcionament i els coneixements necessaris sobre la base de dades en qüestió. El segon aspecte ve influenciat segons

la quantitat de documentació disponible de cada BBDD, com més n'hi hagi més fàcil serà aprendre. En el cas de les bases de dades Oracle, MySQL, Microsoft SQL Server i PostgreSQL es disposa d'una gran quantitat de documentació, facilitant en gran mesura el procés d'aprenentatge, a més a més, la corba d'aprenentatge de SQLite es considera reduïda perquè no presenta una dificultat elevada. Per la resta de bases de dades, la corba és més elevada perquè la dificultat ho és més que la resta i no hi ha tanta documentació.

En el cas del rendiment la majoria de bases de dades compleixen aquest apartat, en el cas de PostgreSQL. Fehrle (2018) indica que el funcionament pot millorar però que no és eficient en alguns punts. En canvi, la resta de bases de dades presenten un rendiment estable.

El següent element a tenir en consideració és el servei en el núvol. Ja que avui dia es tracta d'un element recurrent i bàsic totes les bases de dades amb un funcionament extern disposen d'aquest servei. No és el cas de SQLite ja que tracta d'una base de dades de caràcter local, les dades es guarden dins de la pròpia aplicació o en algun fitxer en el mateix dispositiu des d'on s'ha executat el programa o videojoc. S'ha de destacar que SQLite pot tenir servei d'emmagatzematge al núvol però es tracta d'una característica externa, en aquest cas la BBDD no està preparada per un gran tràfic de dades i no s'ha tingut en compte.

Base de dades	Escalabilitat	Segur	Ràpid	ACID
Oracle	5	4	3	x
MySQL	3	5	3	-
Microsoft SQL Server	4	5	4	x
PostgreSQL	4	4	2	x
MongoDB	5	1	5	x
IBM Db2	4	4	3	x
Elasticsearch	4	4	5	-
SQLite	-	2	5	x
Cassandra	5	3	4	-

Taula 5. Característiques de les bases de dades. Segona part. Font: Elaboració pròpia.

Tot seguit, hi ha l'escalabilitat. La majoria de bases de dades són escalables i estan preparades per afegir nou hardware al ja existent. En canvi, en el cas de SQLite no existeix el concepte d'escalabilitat, ja que és una base de dades que no competeix en termes de client - servidor. És per això, que no hi ha un hardware específic per la base de dades de SQLite i, per tant, no hi ha cap hardware a ampliar. Com expliquen Shiff i Rowe (2018) hi ha molta diferència en com escalen les bases de dades SQL i NoSQL. Les bases de dades SQL tendeixen a escalar de forma vertical, és a dir, incrementar la capacitat del servidor, per altra banda les BBDD NoSQL escalen horitzontalment. D'aquesta forma les NoSQL tenen més potencial en escalabilitat.

En el procés d'emmagatzemar i consultar informació en una base de dades es passa per un procés d'enciptació, que consisteix en transformar amb una clau d'enciptació un conjunt d'informació per fer-la incomprensible, i poder recuperar-la al seu estat original només si es disposa de la clau. Totes les bases de dades llistades disposen d'aquesta funcionalitat però en el cas de MongoDB prioritza la velocitat de la transmissió de dades sacrificant en part la seguretat de les mateixes. En canvi, MySQL prioritza la seguretat de les dades, a canvi de perdre velocitat. A continuació hi ha SQLite, el qual disposa de diversos nivells de seguretat, però al tractar-se de dades que es troben dins del mateix dispositiu en forma de fitxer resulta menys segur evitar que s'hi pugui accedir.

Per part de la rapidesa, totes les bases de dades excepte PostgreSQL disposen d'una correcte velocitat de processament de les dades. PostgreSQL està dissenyat per bases de dades de gran volum, és per això que la velocitat de consultes de inserts i updates són lentes en bases de dades de volum reduït i mig, a més a més, els problemes especificats anteriorment amb el rendiment afecten a la velocitat de la BBDD.

Per últim, les característiques ACID no les compleixen les bases de dades MySQL, Elasticsearch i Cassandra. White (2020) indica que MySQL no ho compleix perquè no suporta les característiques consistència, aïllament i durabilitat, en canvi, sí que compleix atomicitat. Per altra banda, com exposa Vanderbush (2017), la base de dades Elasticsearch treballa emmagatzemant les dades en múltiples documents, i això no comporta complir les característiques ACID. Per últim, DataStax (s.d.) mostra que

la base de dades Cassandra no segueix el model relacional, ja que es tracta d'una BBDD NoSQL i, per tant, no compleix la característica de Consistència, la resta, en canvi, sí que les compleix. Ara bé, com s'exposa a la documentació de MongoDB, tot i ser una base de dades NoSQL sí que compleix els requeriments perquè les seves transaccions es considerin ACID, degut a que funciona amb emmagatzematge en documents individuals.

7.1.2 Anàlisi dels videojocs

Per dur a terme l'anàlisi dels videojocs i les seves necessitats en emmagatzematge de dades, s'han escollit dotze gèneres de videojocs segons la classificació de gèneres llistada en la secció 2.10.1. Aquesta llista forma una recopilació dels tipus de videojocs més rellevants segons Pavlovic (2020). Per disposar d'un anàlisi més complet s'han separat els gèneres simulació i esports, i supervivència i terror, considerats junts en la llista proposada per Pavlovic.

Els elements a analitzar dels gèneres s'han escollit a partir dels elements llistats en el la secció 2.10.2. Aquests elements poden fer ús d'una base de dades per ser emmagatzemats. Cada element de la taula, segons el gènere en qüestió, s'ha determinat de forma generalitzada, és a dir, es pot donar el cas que un videojoc es situï en contra de l'element seleccionat, però que funcioni a mode d'excepció, en aquest cas no es tindrà en compte per determinar el resultat. A més a més, el contingut d'aquest apartat s'ha realitzat de forma objectiva, però a partir de l'experiència pròpia i el coneixement en aquest camp.

Primer de tot, el compte de joc fa referència al sistema de registre i inici de sessió d'un usuari, són dades d'identificació i no tenen una repercussió directe dins del videojoc. S'han determinat els videojoc indicats en la Taula 6, ja que, són els més propensos a incloure contingut multijugador i, per tant, tenir la necessitat d'una identificació prèvia abans d'accedir al videojoc.

Tot seguit, el perfil d'usuari es pot definir com diverses partides guardades dins d'un videojoc, les quals poden ser jugades per diferents persones i aquesta és la forma de diferenciar el seu progrés individual. En els gèneres RPG i Plataformeig és comú disposar d'aquesta característica, ja que es tracta d'un tipus de videojoc on les partides

són individuals i d'una duració elevada. Per altra banda, els gèneres RTS i MOBA, tot i disposar de la característica compte de joc, no es fa ús del perfil d'usuari, ja que el perfil ja va enllaçat al propi compte de joc i no se'n pot tenir més de un. La tercera variant es troba dins del gènere Shooter, no es tracta d'un perfil complet, però en aquests videojocs es poden personalitzar diverses variants d'equipament, simulant perfils de jugador, però no és un element que canvia segons l'usuari que juga, ja que un usuari pot canviar d'equipament.

Gèneres	Compte de joc			Dades de	
	joc	Perfil d'usuari	Llista d'amistats	Missatgeria	pagament
Sandbox				x	
RTS	x		x		
Shooter	x	x	x		
MOBA	x		x	x	x
RPG		x			
Simulació					
Esports	x		x		
Puzzle					
Festa/familiars				x	
Supervivència				x	
Terror					
Plataformeig		x			

Taula 6. Característiques dels gèneres de videojocs. Primera part. Font: Elaboració pròpia.

El següent element és la llista d'amistats, aquesta característica depèn majoritàriament de si el videojoc és multijugador, en cas que ho sigui i es doni la possibilitat de jugar amb amics, la llista és totalment necessària, i ve directament relacionada amb el compte de joc.

A continuació, la característica de missatgeria indica un sistema de comunicació entre jugadors dins del videojoc, es pot trobar com un xat dins d'una partida, com a xat directe entre dos usuaris o com un sistema similar al correu electrònic.

Per finalitzar la primera part de la Taula 6 hi ha les dades de pagament. Només s'indica el MOBA en aquest cas, ja que són els més propensos a que el videojoc sigui d'accés gratuït amb compres integrades dins del videojoc.

A partir d'aquest punt es fa referència a la Taula 7, la qual fa referència a la segona part de les característiques dels gèneres de videojocs.

Per començar, la característica inventari fa referència a tots aquells elements dins del videojoc que el jugador va obtenint i no són exclusius d'una sessió de joc, sinó que s'han de guardar ja que en futures sessions de joc es poden fer servir. Els gèneres escollits són els que poden disposar d'aquesta característica, els que ho fan de forma assegurada són RPG i Supervivència.

Gèneres	Inventari	Taula de puntuació	Sistema de trofeus	Dades de sessió
Sandbox	x		x	x
RTS			x	x
Shooter		x	x	x
MOBA				x
RPG	x		x	x
Simulació	x		x	x
Esports			x	x
Puzzle		x	x	x
Festa/familiars		x	x	x
Supervivència	x		x	x
Terror	x		x	x
Plataformeig		x	x	x

Taula 7. Característiques dels gèneres de videojocs. Segona part. Font: Elaboració pròpia.

A continuació, la taula de puntuacions és un element que indica en forma de classificació la puntuació obtinguda per diferents jugadors, però perquè hi hagi aquesta taula cal que existeixi algun tipus de repte que serveixi per generar una puntuació. Gèneres com Plataformeig, Puzzle i Festa/familiars són els més comuns per presentar reptes acompanyats de puntuacions. El gènere Shooter fa ús d'una taula

de puntuacions al final de la partida mostrant quin jugador ha tingut un major rendiment de forma ordenada.

Tot seguit, el sistema de trofeus recull fites aconseguides pel jugador. En aquest cas, el gènere MOBA és l'únic que no inclou aquesta característica. La resta, en canvi, són més propensos a incloure un sistema de trofeus.

I, finalment, les dades de sessió són una característica comuna en qualsevol gènere de videojocs. Recullen les dades de sessió de cada jugador i les guarden en la base de dades. Com que hi han molt tipus de dades que es poden recollir en aquest apartat no es descarta cap gènere.

Finalment, fora dels gèneres, s'agrupen aquestes característiques llistades prèviament segons si el videojoc és d'un sol jugador o multijugador. D'aquesta forma els aspectes a analitzar no depenen únicament del gènere i, per tant, es pot tenir en compte una segona variable. Els videojocs poden ser molt diferents i que un gènere implementi una característica que segons la llista no ha de ser implementada, per tant la següent taula aporta un punt de vista diferent.

Tipus de videojoc	Compte de joc	Perfil d'usuari	Llista d'amistats	Missatgeria	Dades de pagament
D'un sol jugador		x			
Multijugador	x	x	x	x	x

Taula 8. Característiques dels tipus de videojocs. Primera part. Font: Elaboració pròpia.

En la Taula 8 el perfil d'usuari és la única característica que pot aparèixer en ambdós tipus de videojoc. En cap dels dos és segur que hi pot aparèixer, però és possible. Només que es presentin diferents guardats simultanis ja hi és la característica.

La resta de característiques són pròpies de videojocs de tipus multijugador, ja que totes són necessàries en un entorn on hi han diversos jugadors.

En el cas de la Taula 9, l'inventari és un element possible en els dos tipus, és a dir, ja sigui el videojoc individual o multijugador no vol dir que no hi pugui haver un inventari.

Tot seguit, la taula de puntuacions pot semblar més adequada per un videojoc multijugador, ja que significa la implicació directa de més d'un usuari. Però també pot existir en videojocs d'un sol jugador ja que es poden presentar reptes individuals i generar una puntuació per ser compartida posteriorment, ja sigui en forma de classificació global, només d'amics, segons la zona geogràfica, etc.

Tipus de videojoc	Sistema de			
	Inventari	Taula de puntuació	trofeus	Dades de sessió
D'un sol jugador	x	x	x	x
Multijugador	x	x	x	x

Taula 9. Característiques dels tipus de videojocs. Segona part. Font: Elaboració pròpia.

A l'igual que l'inventari, el sistema de trofeus pot estar o no present en qualsevol tipus de videojoc, en aquest cas depèn d'altres aspectes, però no es pot descartar a primeres.

Finalment, les dades de sessió no depenen de cap aspecte en concret, en qualsevol situació es poden aconseguir i donar ús. Poden tractar-se de dades referents al temps de connexió, quins són els punts més difícils del videojocs, el tipus de jugabilitat que escull el jugador, etc.

7.1.3 Les bases de dades més adequades

Un cop analitzades les bases de dades més destacades i la llista de gèneres seleccionada es creuen aquestes dades per obtenir quines bases de dades són les que satisfan millor les necessitats de cada gènere. El creuament de dades no es realitza pels tipus de videojocs, d'un sol jugador i multijugador, ja que es tracta d'una divisió massa global.

Primer de tot, cal concretar una relació directa entre les característiques analitzades de les bases de dades i dels gèneres dels videojocs, mostrat a la Taula 10.

Per part del compte de joc i les dades de pagament es prioritza la seguretat de les dades. Tot el que fa referència al pagament necessita ser assegurat com a màxima

prioritat, en canvi, per part de les dades de compte de joc també són necessàries que es disposi de seguretat elevada, ja que són les dades d'inici de sessió on, entre elles, hi ha la contrasenya.

Gènere	Base de dades
Compte de joc	Seguretat
Perfil d'usuari	Escalabilitat
Llista d'amistats	Escalabilitat
Missatgeria	Rapidesa
Dades de pagament	Seguretat
Inventari	Escalabilitat
Taula de puntuacions	Escalabilitat
Sistema de trofeus	Escalabilitat
Dades de sessió	Escalabilitat

Taula 10. Relació de les característiques de les bases de dades i els gèneres de videojocs. Font: Elaboració pròpia.

Per altra banda, la missatgeria és l'única característica que s'enfoca en la rapidesa, ja que es tracta d'una dada que s'ha de transmetre en temps real, existint la possibilitat que hi pugui haver una conversació o diàleg. No deixa de ser important la seguretat en aquest cas però no es tracta de dades sensibles com és el cas del punt anterior.

La resta de característiques s'enfoquen en l'escalabilitat de la base de dades, ja que es tracta de dades que creixen a mesura que avança el temps, no només per a cada usuari sinó en funció de la quantitat d'usuaris. Cal remarcar que tant la seguretat com la rapidesa també són elements importants però el que està relacionat més directament és la capacitat de creixement de la base de dades.

Finalment queden la resta de característiques de la base de dades. Ja que no estan directament relacionades amb les especificades dels gèneres de videojocs no se'ls hi ha assignat una relació. Aquestes, en canvi, es tenen en compte a l'hora de decidir entre dues bases de dades que poden plantejar una decisió ajustada. Les característiques de les bases de dades que tenen més impacte fora de la relació de la Taula 10 són la accessibilitat, el preu, la corba d'aprenentatge i el rendiment.

A continuació es mostra el resultat de l'elecció de quines bases de dades s'ajusten més a les necessitats de cada gènere, a partir de la Taula 10. En la Taula 11 es mostren tres eleccions de bases de dades ordenades de més adequada a menys per a cada gènere seleccionat.

Gèneres	Primera opció	Segona opció	Tercera opció
Sandbox	Elasticsearch	MongoDB	MySQL
RTS	Microsoft SQL Server	Oracle	Elasticsearch
Shooter	Oracle	PostgreSQL	Microsoft SQL Server
MOBA	Microsoft SQL Server	Elasticsearch	Oracle
RPG	Oracle	MongoDB	Cassandra
Simulació	Oracle	MongoDB	Cassandra
Esports	Elasticsearch	Microsoft SQL Server	Oracle
Puzzle	Oracle	MongoDB	Cassandra
Festa/familiars	MongoDB	Elasticsearch	Microsoft SQL Server
Supervivència	MongoDB	Elasticsearch	Microsoft SQL Server
Terror	Oracle	MongoDB	Cassandra
Plataformeig	Oracle	MongoDB	Cassandra

Taula 11. Relació dels gèneres de videojocs amb les bases de dades escollides.
Font: Elaboració pròpia.

Per començar, el gènere de videojocs Sandbox presenta les característiques Missatgeria, Inventari, Sistema de trofeus i Dades de sessió. Aquestes característiques estan enfocades en la rapidesa i l'escalabilitat d'una base de dades, com a elements principals. En aquest cas, la base de dades Elasticsearch és la més adequada, aconsegueix complir els requisits que demanen les característiques del gènere, a més de comptar amb un preu competitiu. La següent base de dades pel gènere Sandbox és MongoDB, tant la rapidesa com l'escalabilitat són molt elevades, però no és la primera elecció degut a la falta de seguretat que presenta. Per últim hi ha MySQL, en quant als requeriments es semblant a altres bases de dades, però ocupa aquest espai gràcies a la resta de característiques que presenta, com la corba d'aprenentatge i el rendiment del que disposa.

El següent gènere és el RTS, com es mostra en les Taules 6 i 7, les característiques del gènere són el compte de joc, la llista d'amistats, el sistema de trofeus i les dades de sessió. Ara bé, segons la Taula 10, les característiques mencionades es basen en

l'escalabilitat i la seguretat. Per tant, a partir de les Taules 6 i 7 es considera la base de dades més adequada Microsoft SQL Server, ja que predomina la seguretat i l'escalabilitat. A més a més, ve acompanyat d'una corba d'aprenentatge molt reduïda, degut principalment a la gran quantitat de documentació disponible, i d'un bon rendiment. Tot seguit, les següents bases de dades seleccionades són Oracle i Elasticsearch, ja que ambdues BBDDs compleixen els requeriments del gènere, s'ha posicionat per sobre oracle degut al seu rendiment i la corba d'aprenentatge, tot i que el preu sigui superior per part d'Oracle.

Tot seguit, pel gènere de videojocs Shooter s'ha seleccionat com a primera opció la base de dades Oracle. Seguint les referències a les taules corresponents, el gènere seleccionat disposa de compte de joc, perfil d'usuari, llista d'amistats, taula de puntuacions, sistema de trofeus i dades de sessió. Les prioritats són l'escalabilitat i la seguretat. Ara bé, semblant al cas anterior, en aquest l'escalabilitat té molta més presència, per tant, Oracle suposa una millor opció ja que escala millor. Com a segona opció s'ha determinat que la base de dades PostgreSQL és la següent opció degut, principalment, a la resta de característiques que presenta. Comparat amb la tercera opció, Microsoft SQL Server, el preu que presenta la segona opció mostra una gran competitivitat.

El quart gènere llistat és MOBA, aquest gènere és l'únic que presenta, com una de les característiques, les dades de pagament, la qual demana una gran seguretat ja que són dades sensibles per l'usuari. Per les característiques que presenta el gènere es busca seguretat, rapidesa i escalabilitat en les bases de dades. És per això que s'ha seleccionat Microsoft SQL Server com a primera opció, ja que presenta la major seguretat entre les bases de dades llistades, tot seguit Elasticsearch, degut a l'equilibri general com a base de dades tenint en compte que disposa d'una elevada seguretat i finalment Oracle perquè disposa de gran escalabilitat i bon rendiment.

A continuació, els gèneres RPG, Simulació, Puzzle, Terror i Plataformeig s'agrupen degut als seus requeriments sobre les característiques de les bases de dades. Tot i que, les seves pròpies característiques són diferents totes elles es basen en l'escalabilitat, per tant, aquesta es col·loca com a major prioritat. L'escalabilitat en aquest cas es veu més acompanyada per bases de dades NoSQL, ja que aquestes

tenen una major facilitat en aquest camp, no obstant això, les dues bases de dades no relacionals llistades presenten clars punts febles. És per això que la primera opció per aquest conjunt de gèneres de videojocs és la base de dades Oracle, entre les bases de dades relacionals és la que mostra una major capacitat a l'hora de tractar l'escalabilitat. Ara bé, també mostra gran capacitat en altres aspectes, com són la corba d'aprenentatge, l'accessibilitat i el rendiment. Les següents dues bases de dades són NoSQL. La primera és MongoDB, principalment perquè mostra una gran capacitat per escalar i una enorme rapidesa, al igual que la seva accessibilitat i la disposició d'un preu totalment competitiu. En canvi, presenta una clara falla, que és la seguretat de la mateixa. Tot i no ser un requeriment pel gènere en qüestió, la seguretat és un element a tenir en compte. Finalment, la última base de dades seleccionada és Cassandra, també és no relacional, per tant, la seva capacitat d'escalar és excepcional, ara bé, encara que presenta un preu més reduït i una seguretat més elevada, en la resta d'aspectes es queda enrere respecte MongoDB.

El següent gènere a exposar és Esports, aquest gènere de videojocs contempla com a característiques el compte de joc, la llista d'amistats, el sistema de trofeus i les dades de sessió. Per tant, pel gènere d'Esports es busca que la base de dades disposi de seguretat, rapidesa i escalabilitat, ja que es busca que tingui de tot i en un quantitat similar, ja que només hi han dues característiques que busquen escalabilitat, s'han seleccionat les bases de dades més equilibrades en tots els aspectes. La primera és Elasticsearch, és una base de dades que presenta un gran equilibri tenint en consideració tots els seus aspectes, sobresurt més en el preu i la rapidesa, i menys en la corba d'aprenentatge i el rendiment. Les següents seleccionades són Microsoft SQL Server i Oracle, també disposen d'un bon equilibri en les seves característiques, però al tractar-se de bases de dades més conegudes disposen de major documentació disponible i, per tant, d'una corba d'aprenentatge més reduïda, per altra banda, el seu preu és més elevat.

Finalment, s'agrupen els gèneres Festa/familiars i Supervivència, ja que presenten unes característiques similars, buscant principalment rapidesa i escalabilitat en les bases de dades. La primera base de dades seleccionada és MongoDB, degut a que compleix de forma més precisa els requeriments, l'escalabilitat i rapidesa són les més elevades. Tot seguit, Elasticsearch i Microsoft SQL Server són les següents bases de

dades que poden satisfer millor els gèneres en qüestió. Ja s'ha parlat de l'equilibri que mostren les dues bases de dades.

Per concloure destacar que la majoria de les bases de dades mostren competències d'alt nivell, totes elles competeixen per ser les seleccionades per la immensa quantitat d'empreses de videojocs, de pàgines web, d'aplicacions, etc. Ara bé, cada una ha buscat especialitzar-se i oferir diversitat buscant principalment diferenciar-se de la competència. També cal destacar que aquest recull de les bases de dades és ínfim comparat amb la gran quantitat de bases de dades i de serveis similars disponibles al mercat.

Hi ha hagut un conjunt de bases de dades que no s'han seleccionat en cap cas degut a que les seves característiques són inferiors respecte la resta o no adequades. Com pot ser el cas de SQLite, que al tractar-se d'una base de dades de caràcter local presenta una gran limitació per a ser escollida, eliminant a primeres tots els gèneres de videojocs que puguin contenir característiques que necessitin d'un servidor extern. Ara bé, les bases de dades que no s'han seleccionat, disposen també de gran capacitat per satisfer les necessitats de videojocs i aplicacions, ja que han estat seleccionades com a més destacades durant el 2020 segons Statista (2021).

7.2 Resultats del prototip

El prototip s'ha desenvolupat en el motor de jocs Unity basat en un videojoc multijugador asíncron d'escacs, tal i com s'explica en la secció 6.1.

El prototip està desenvolupat per fer-ne ús en un dispositiu mòbil, ja que és el tipus de plataforma on encaixa més, degut a la seva naturalesa de sessions de joc curtes. Es disposa de set finestres, les seves funcionalitats bàsiques s'expliquen en les seccions 6.2 i 6.3. Tot i això, a continuació s'aporten imatges de les finestres del prototip junt amb una descripció de cada una.

Ja que el prototip s'ha desenvolupat per un entorn mòbil la orientació del mateix és vertical, facilitant l'ús amb una sola mà. Per altra banda, la selecció de colors és temporal, pensada per funcionar en un videojoc en fase prototip.

La primera finestra del prototip o "Splash Screen" mostrada en la Figura 27 mostra una imatge del tauler del prototip. A continuació es troben dos botons per accedir al registre o a l'inici de sessió del videojoc. Aquesta finestra és la que es visualitza a l'obrir el videojoc.

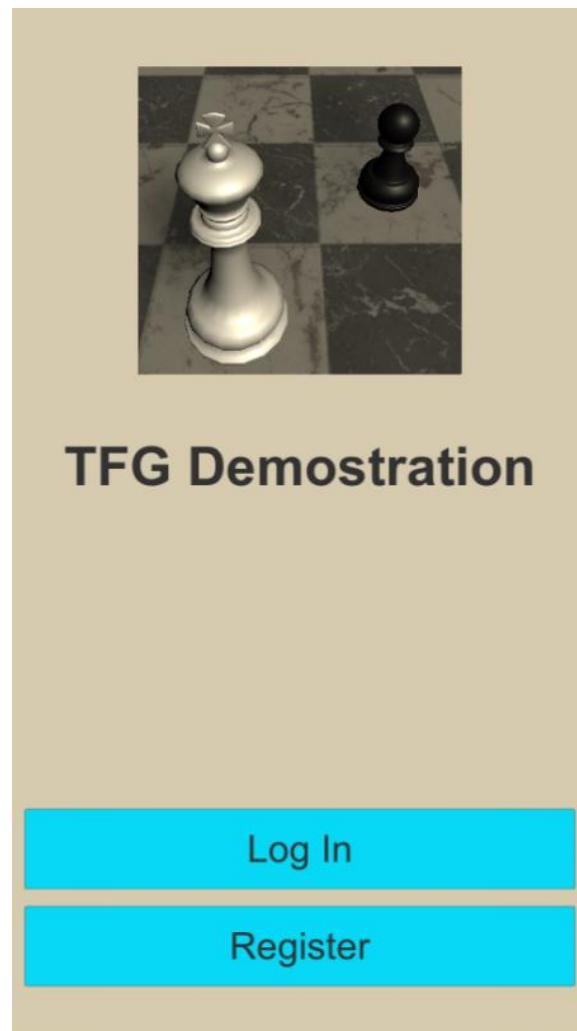


Figura 27. Splash Screen del prototip. Font: Elaboració pròpia.

La Figura 28 mostra la finestra de registre del prototip. L'usuari que faci ús de l'aplicació ha d'introduir un sobrenom, el correu electrònic i la contrasenya. També cal repetir la contrasenya per evitar errors d'escriptura. Quan tots els camps es troben omplerts correctament el botó de registre es desbloqueja, fins el moment mostra un color blau més apagat que significa que el botó es troba desactivat.

Els camps que s'han d'omplir tenen unes condicions concretes perquè es considerin correctes i es desbloquegi el botó de registre. El correu electrònic ha de tenir una llargada mínima de vuit caràcters, tot i que "gmail.com" ja n'ocupa nou. La contrasenya ha de tenir una llargada de cinc caràcters o més i les dues contrasenyes introduïdes han de coincidir. I finalment, el tamany del sobrenom ha de ser d'un mínim de quatre caràcters.

Actualment no es troba implementat un sistema que informi a l'usuari sobre aquestes condicions, però més endavant es proposen un seguit de funcionalitats que es desenvoluparien en fases més avançades del videojoc, on una d'elles és un sistema de missatges per informar errors o interaccions satisfactòries.

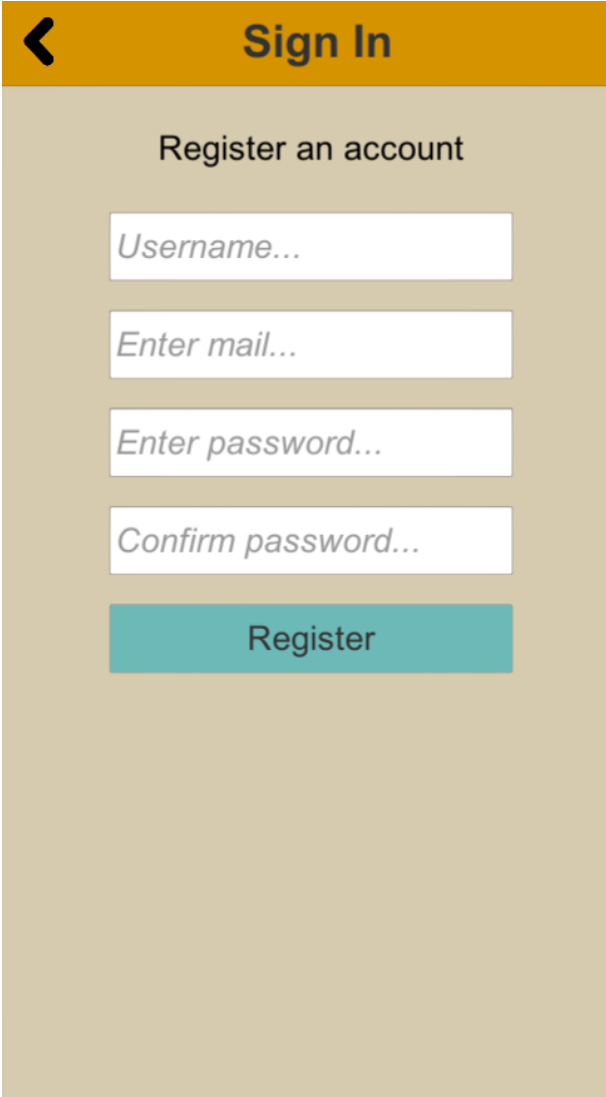
Aquesta imatge mostra un prototip d'una finestra de registre d'una aplicació mòbil. La pantalla té un fons de color taupe i una barra superior de color taronja amb un símbol de retrocedre a l'esquerra i el títol "Sign In" a la dreta. Just a sota de la barra superior, hi ha el text "Register an account". A continuació, hi ha quatre camps d'entrada de text blancs amb text de marcador de posició gris: "Username...", "Enter mail...", "Enter password..." i "Confirm password...". A la part inferior dels camps, hi ha un botó rectangular de color verd clar amb el text "Register" en negre.

Figura 28. Finestra de registre del prototip. Font: Elaboració pròpia.

La Figura 29 mostra la finestra d'inici de sessió. De la mateixa forma que en el registre s'han d'omplir els camps per poder interaccionar amb el botó, també en el cas de l'inici de sessió. S'apliquen les mateixes condicions pel correu electrònic i la contrasenya, es necessita un mínim de vuit i cinc caràcters, respectivament. Un cop desbloquejat el botó, es realitzen les consultes de dades i la descàrrega d'informació de l'usuari de la base de dades, i s'accedeix a la finestra del menú principal.

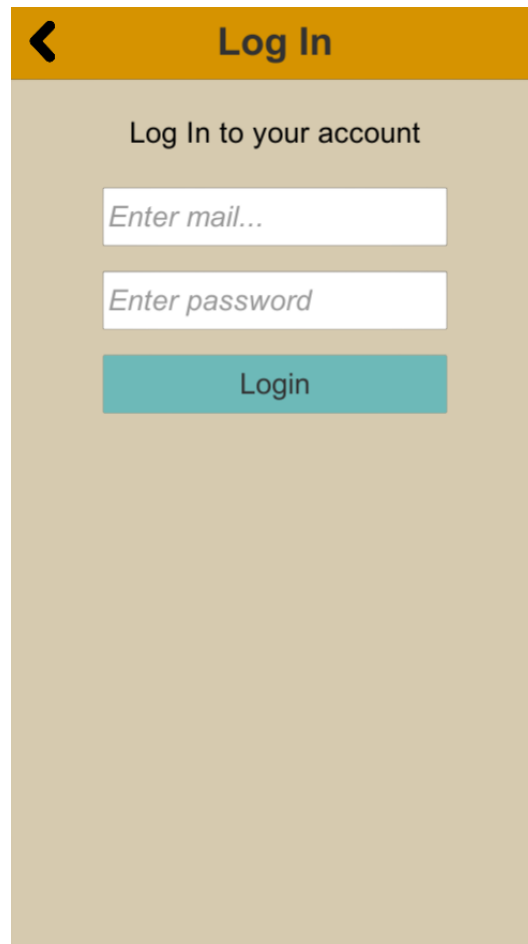


Figura 29. Finestra de l'inici de sessió del prototip. Font: Elaboració pròpia.

La Figura 30 ensenya la finestra del menú principal del prototip. Aquesta finestra és el punt central de l'aplicació on l'usuari un cop dins pot navegar a les diferents finestres disponibles. També es mostren dos llistats, un per les partides on el jugador ha de realitzar el moviment i l'altre per les partides on ha de moure l'oponent. Cada objecte de la llista és una partida i mostra la informació de la mateixa, que és: el nom de l'oponent i quin color de peces es juga. Amb el botó "Play" es pot accedir al tauler.

A la part inferior de la finestra es troba el botó “New Match” per crear una nova partida. La nova partida apareixerà a la llista de partides on el mateix jugador ha de moure, ja que qui crea la partida juga les peces blanques, les quals juguen primeres.

A la part superior esquerra es troba el botó per tancar sessió i tornar a la Splash Screen. A la part superior dreta es troben els botons que, d'esquerra a dreta, serveixen per accedir al perfil, a l'historial de partides i per refrescar les partides actives.

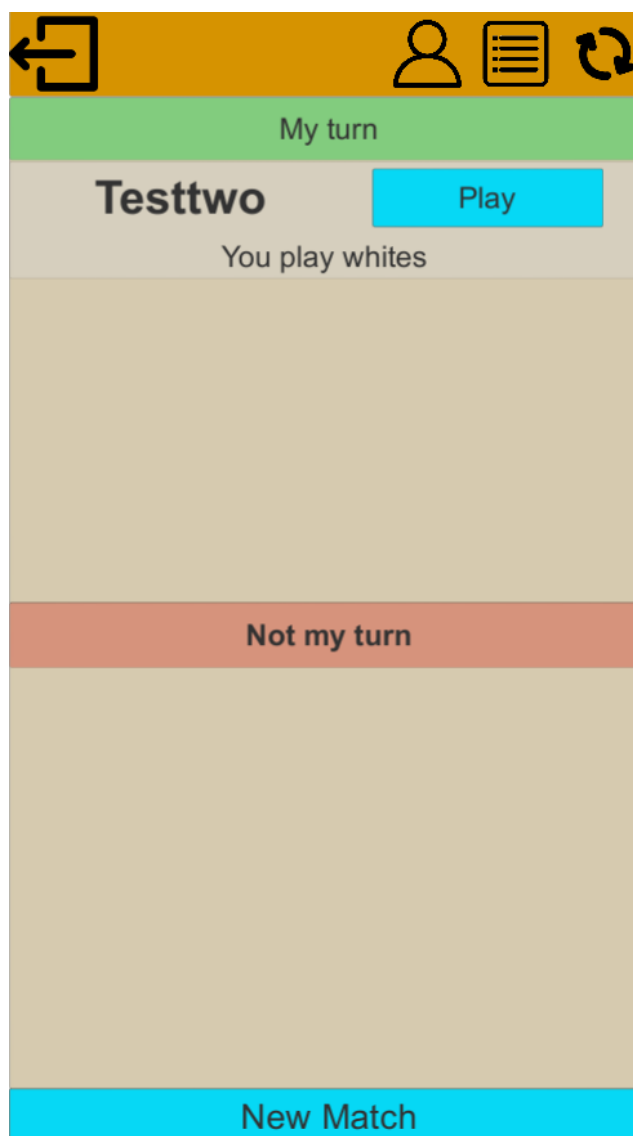


Figura 30. Finestra del menú principal del prototip. Font: Elaboració pròpia.

La Figura 31 mostra la finestra de la part jugable del prototip, on es troba el tauler d'escacs.

A la part superior es troba el nom de l'oponent. Si és el torn del jugador es podrà interaccionar amb les peces del color que es juguen. En aquest cas, en fer clic sobre una peça del bàndol del jugador s'hi afegirà un contorn groc, per remarcar la peça seleccionada. Les caselles on la peça es pugui moure es marcaran de color verd i si un enemic es troba en una casella disponible de moviment es marcarà de color vermell, indicant que si es decideix moure la peça en una casella vermella s'eliminarà una peça enemiga.

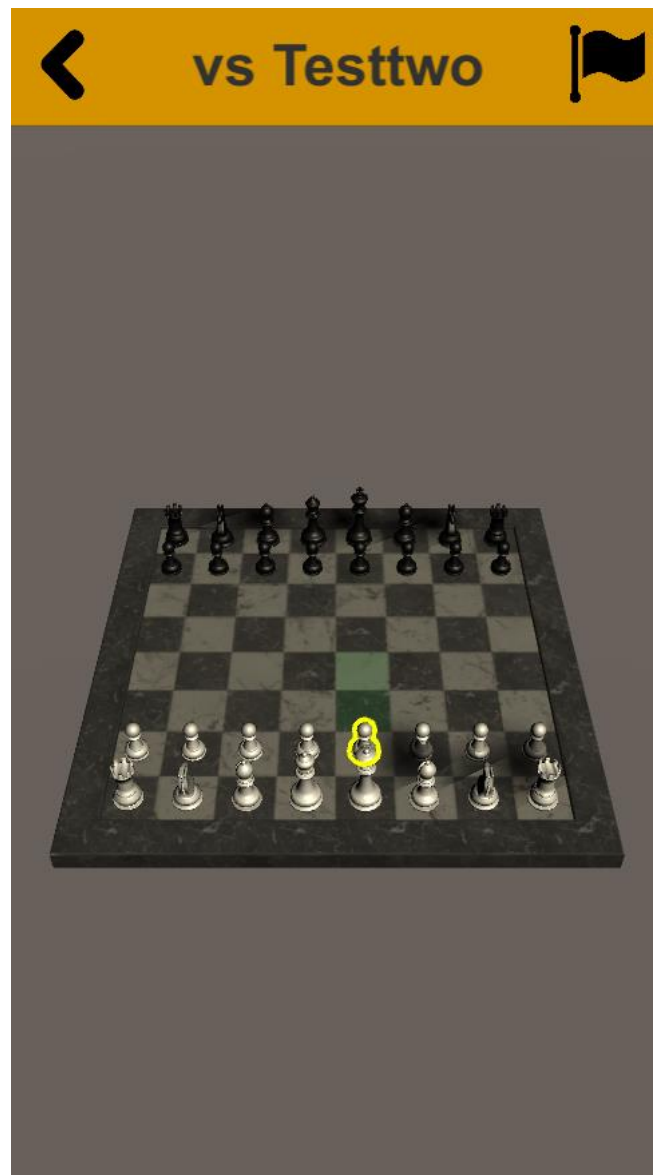


Figura 31. Finestra de la part jugable del prototip. Font: Elaboració pròpia.

Al completar un moviment apareixerà una finestra emergent demanant una confirmació del moviment. Si s'accepta es donarà el torn com a finalitzat.

A la part superior dreta es troba el botó de rendició. Si es pressiona i s'accepta la confirmació de la finestra emergent, es donarà la partida per finalitzada i es tornarà al menú principal on es refrescant automàticament les dades.

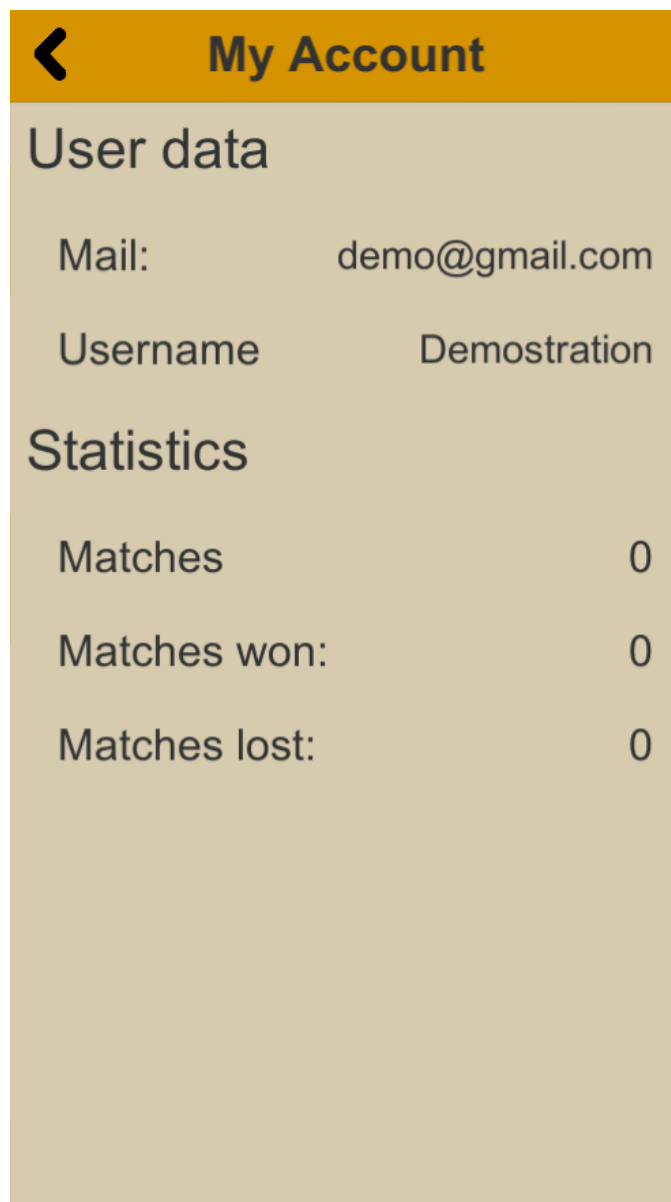


Figura 32. Finestra del perfil del prototip. Font: Elaboració pròpia.

La Figura 32 mostra la finestra del perfil on es troben les dades de l'usuari i les estadístiques de les partides jugades, guanyades i perdudes fins el moment. L'exemple de la Figura 32 mostra una partida recent creada pel que les estadístiques indiquen que s'han jugat zero partides.

La Figura 33 ensenya la última finestra desenvolupada del prototip. Aquesta llista un historial de les partides acabades. Cada objecte de la llista és una partida on es mostren les dades de cada una. Es mostra el nom de l'oponent, el color de peces jugades, el número de moviments que s'han realitzat fins la finalització i si s'ha guanyat o perdut. En cas que s'hagi guanyat la partida l'objecte tindrà un color verd de fons, en cas contrari vermell.

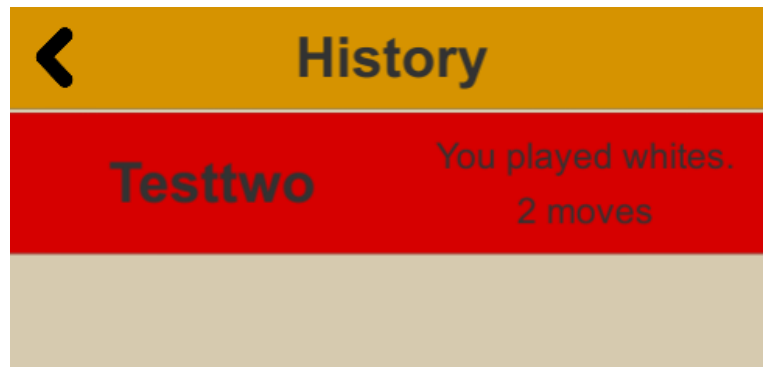


Figura 33. Finestra de l'historial del prototip. Font: Elaboració pròpia.

A continuació es llisten un seguit de funcionalitats que s'han descartat. S'ha realitzat aquesta decisió per assegurar la finalització del projecte dins del temps establert. Ara bé, aquestes funcionalitats descartades són, dues d'elles menors i la tercera no afecta a l'objectiu principal pel que s'ha desenvolupat el prototip.

Les dues primeres funcionalitats descartades són canviar la contrasenya i recuperar la contrasenya, són funcionalitats menors que tenen un lleu impacte en la base de dades en cas d'haver estat desenvolupades. Per altra banda, la tercera funcionalitat consisteix en la finalització d'una partida d'escacs de forma tradicional, és a dir, amb l'escac i mat. No obstant això, haver descartat aquesta funcionalitat no ha suposat una reducció de la qualitat del prototip, ja que, es pot arribar al mateix resultat fent ús de la rendició. D'aquesta forma es pot finalitzar igualment una partida i les funcionalitats posteriors de les estadístiques i l'historial de partides no es veuen afectades.

A continuació, es proposen un seguit de funcionalitats que s'afegirien al videojoc a desenvolupar en futures versions. Primer de tot, afegir les funcionalitats descartades anteriorment perquè completen el funcionament mínim de la gestió d'usuaris. Tot seguit, un sistema de feedback per l'usuari on es mostren missatges segons el que

realitza l'usuari i com resulta la seva acció, per exemple, quan l'usuari introdueix les dades per iniciar sessió es pot donar els cas que falli de dues formes: pot resultar que el correu electrònic no existeix o que la contrasenya introduïda és incorrecta, en aquests dos casos mostrar un missatge aclarint a l'usuari perquè no ha iniciat sessió correctament. Aquest sistema de missatges es trobaria al llarg del videojoc per mostrar un feedback en una situació en la que el videojoc no pot realitzar l'acció perquè hi ha hagut algun fallo o no ha resultat com es voldria. En casos com iniciar sessió correctament no caldria mostrar un missatge, degut a que, la pròpia acció comporta avançar fins la finestra del menú principal, i això ja suposa un feedback per l'usuari.

La següent funcionalitat ha desenvolupar en una versió futura del prototip consisteix en una funcionalitat que estava pensada per ser afegida en la fase de prototipat, en canvi, no ha resultat d'aquesta forma ja que suposa una gran càrrega de feina pel desenvolupament. Aquesta funcionalitat consisteix en una extensió de l'actual historial. L'historial mostra les partides acabades, amb el nom del rival, el número de moviments fins la finalització i en color verd si s'ha guanyat o vermell si s'ha perdut. La extensió consisteix en mostrar una repetició en viu de tots els moviments que s'han realitzat durant el transcurs de la partida. És a dir, al seleccionar una partida a l'historial es podria veure en directe tots els moviments de la partida. Això suposaria adaptar el sistema de joc actual perquè es poguessin moure les peces de forma automàtica segons una llista de moviments. A més a més, serviria per expandir la base de dades amb una nova taula on es guardarien els moviments que es realitzen.

Les següents funcionalitats es basen en l'espai de joc, el gameplay. Aquestes millores es realitzarien amb l'objectiu de millorar l'experiència de joc i la informació que es proporciona durant el temps en que es juga específicament. La primera consisteix en una millora de la càmera, permetent una rotació lliure amb el punt central el tauler, també es podria apropar i allunyar la càmera per poder veure millor el tauler des de diferents posicions. El segon consistiria en veure als extrems laterals de la pantalla una llista amb les peces que ja han estat eliminades de la partida, per oferir una millor visió de la situació actual de la partida.

L'última aportació de funcionalitats consisteix en dues relacionades amb el temps. La primera consisteix en mostrar a l'historial de partides acabades el temps d'inici i de fi,

és a dir, quan va començar la partida i quan es va acabar. D'aquesta forma, els usuaris podrien veure amb precisió les dates de cada partida. En aquest moment, les partides es mostren de forma cronològica, ja que, els registres a la base de dades s'afegeixen en ordre segons es finalitzen, i com que es recullen sense alterar el seu ordre ja es mostren ordenats per temps. Però si s'afegeixen les dates d'inici i de fi, els usuaris podran veure exactament quan van començar i acabar. La segona funcionalitat relacionada amb el temps consisteix en col·locar un temps màxim a cada jugador per poder realitzar la següent jugada, per exemple, quan el primer jugador finalitza la seva tirada el segon jugador té fins a 24h per realitzar la seva tirada on aquestes hores han de ser decidides. En cas que no es finalitzi la tirada abans del temps establert el jugador implicat perdrà la partida. Aquesta funcionalitat ajuda a que les partides no s'allarguin en excés i també elimina totes les partides de jugadors inactius.

8. Conclusions

En aquest projecte s'ha investigat el funcionament de les bases de dades, els seus tipus, les BBDD actuals més destacades i el funcionament individual de cada una. Per altra banda, també s'han investigat els videojocs, principalment els seus gèneres en relació a les seves necessitats d'emmagatzematge de dades. Es considera que els objectius proposats en el projecte han estat assolits satisfactòriament. El primer objectiu consisteix en desenvolupar la demostració d'un videojoc en fase prototip amb el motor de jocs Unity fent ús d'una base de dades situada en un servidor extern. El segon objectiu consisteix en l'anàlisi de la relació de les bases de dades i els videojocs i com a resultat obtenir quines bases de dades poden satisfer millor les necessitats de cada gènere de videojocs.

La recollida d'informació de les bases de dades s'ha realitzat sense cap inconvenient ja que les bases de dades disposen de gran quantitat de documentació i informació, comparat amb la recollida de dades dels videojocs. Per part dels videojocs, i en especial conèixer les bases de dades que implementa cada videojoc, ha estat una tasca més complexa, donat que moltes empreses de desenvolupament de videojocs no publiquen aquest tipus d'informació. La informació relacionada amb el desenvolupament d'un videojoc és confidencial, en la majoria de casos.

Ara bé, aquest projecte aporta coneixement al lector sobre la finalitat i la implementació de les bases de dades en el món dels videojocs. Com s'ha vist, les bases de dades són eines de gran importància i permeten implementar funcionalitats basades en la recollida i guardat de dades. Especialment, amb el desenvolupament de jocs MMO i la gran disponibilitat de guardar dades al núvol, fan que les bases de dades siguin un element essencial en molts videojocs. També s'ha repassat la tendència creixent d'emmagatzemar dades no estructurades, com poden ser textos o dades de grafs, i com s'han desenvolupat nous dissenys de bases de dades per facilitar la gestió del seu emmagatzematge i consulta. Un altre aspecte rellevant és el de Big Data, o grans volums de dades de manera molt resumida. El fenomen Big Data ha entrat també en el món dels videojocs per guardar la informació relativa al joc dels jugadors, que s'usa en moltes ocasions per prendre decisions de disseny o de negoci.

El prototip s'ha desenvolupat en el temps establert. S'ha investigat tot el necessari de les bases de dades per concloure en l'ús d'una base de dades MySQL per implementar en la demostració. La base de dades emprada en el projecte s'ha allotjat en un servidor web gratuït. El prototip s'ha basat en el joc de taula escacs. La seva essència de partides per torns ha permès implementar la mecànica de les partides asíncrones on cada jugador resol el seu torn per separat. Implementar aquesta mecànica ha permès que la base de dades no es centri en oferir velocitat, ja que els dos usuaris implicats no juguen al mateix temps.

Per oferir una base de dades més amplia i diversa s'ha implementat un historial de partides on es poden visualitzar les partides acabades i es mostra informació de les mateixes.

La base de dades creada pel videojoc s'ha allotjat en un servidor extern, per dur a terme la connexió entre la base de dades i el videojoc s'ha desenvolupat una sèrie d'scripts en el llenguatge de programació PHP. Aquest conjunt d'scripts fan d'intermediaris entre els dos punts.

El desenvolupament i la connexió a la base de dades han aportat experiència i nous coneixements a disposició del lector. Aquests coneixements poden arribar a ser necessaris per una persona amb un perfil de desenvolupador de videojocs.

De cara a futures versions del videojoc s'han plantejat un seguit de funcionalitats llistades en la secció 7.2. Aquestes funcionalitats no només amplien el videojoc sinó també la base de dades.

Un altre aspecte pel futur seria implementar una base de dades no estructurada, com per exemple MongoDB. Una base de dades NoSQL suposa una gran diferència respecte la implementada actualment, MySQL. Aquest canvi suposaria un gran repte, ja que les bases de dades disposen d'un funcionalment totalment diferent al d'una base de dades encarregat de guardar dades estructurades. També caldria realitzar la cerca d'un lloc web que permetés allotjar una base de dades NoSQL. El desenvolupament del mateix prototip però amb una base de dades no estructurada podria permetre analitzar les diferències que pertoquen. És a dir, s'hauria d'analitzar si cal desenvolupar el prototip de diferent forma.

Un últim aspecte que resultaria interessant investigar és l'aparició de noves bases de dades. A l'igual que hi han llenguatges de programació que estan apareixen recentment en el món de la programació seria interessant identificar bases de dades, models de bases de dades o llenguatges amb un comportament similar. Les bases de dades més conegudes i utilitzades porten ja un gran recorregut però es pot donar el cas que noves bases de dades aportin una nova visió.

9. Bibliografia/Referències

- Abhinandan, B., i Bandyopadhyay, S. (2016). Big Data - A Review on Analysing 3Vs. *Journal of Scientific and Engineering Research*, 1-4.
- Abiteboul, S., Hull, R., i Vianu, V. (1995). *Foundations of Databases: The Logical Level*. Reading: Addison Wesley.
- AltexSoft. (2019). Comparing Database Management Systems: MySQL, PostgreSQL, MSSQL Server, MongoDB, Elasticsearch and others. *Altexsoft*. Recollit de <https://www.altexsoft.com/blog/business/comparing-database-management-systems-mysql-postgresql-mssql-server-mongodb-elasticsearch-and-others/>
- Berg, K., Seymour, T., i Goel, R. (2012). History Of Databases. *International Journal of Management & Information Systems*, 29-36.
- Blackwell, C. (2019). Managed Databases for Awesome Games. *AWS Game Tech Blog*. Recollit de <https://aws.amazon.com/es/blogs/gametech/managed-databases-for-awesome-games/>
- Blizzard. (2021). Software Engineer Database Server World of Warcraft. *Blizzard Entertainment*. Consultat el 23 / 3 / 2021, a <https://careers.blizzard.com/es/es/job/R004825/Software-Engineer-Database-Server-World-of-Warcraft>
- Butcher, J. (s.d.). Databases in Online (Social) Gaming. *Severalnines*. Recollit de <https://severalnines.com/resources/database-management-tutorials/databases-online-social-gaming>
- Carrera, F. J. (2019). Beginners Guide to SQLite. *Datacamp*. Recollit de <https://www.datacamp.com/community/tutorials/beginners-guide-to-sqlite>
- Ceri, S., Gottlob, G., i Tanca, L. (1990). *Logic Programming and Databases*. Heidelberg: Springer.

- Christian, M. (2019). Saved, But Not Forgotten. The evolution of saving in video games, from the password to the cloud, and nearly every obscure memory card format in-between. *Tedium*. Recollit de <https://tedium.co/2019/02/21/video-game-save-state-history/>
- Comeau, A. (2015). *MySQL Explained: Your Step-by-step Guide to Database Design*. Scotts Valley: CreateSpace.
- DataStax. (s.d.). How are Cassandra transactions different from RDBMS transactions? *Data Stax*. Recollit de <https://docs.datastax.com/en/cassandra-oss/2.2/cassandra/dml/dmlTransactionsDiffer.html>
- Date, C. J. (2011). *SQL and Relational Theory: How to Write Accurate SQL Code* (2a ed.). Sebastopol: O'Reilly Media.
- Dealessandri, M. (2020). What is the best game engine: is Unity right for you? *Gameindustry*. Recollit de <https://www.gamesindustry.biz/articles/2020-01-16-what-is-the-best-game-engine-is-unity-the-right-game-engine-for-you>
- Draken, N., i Turner, B. (2021). Best cloud databases of 2021. *Techradar*. Recollit de <https://www.techradar.com/best/best-cloud-databases>
- Drumgoole, J. (2018). SEGA HARDlight Migrates to MongoDB Atlas to Simplify Ops and Improve Experience for Millions of Mobile Gamers. *MongoDB Blog*. Recollit de <https://www.mongodb.com/blog/post/sega-hardlight-migrates-to-mongodb-atlas-simplify-ops-improve-experience-mobile-gamers>
- Eshtay, M., Sleit, A., i Aldwairi, M. (2019). Implementing bi-temporal properties into various NoSQL database categories. *International Journal of Computing Science and Mathematics*, 45-52.
- ExploreGroup. (2019). The Most Popular Databases 2019. *Explore group*. Recollit de <https://www.explore-group.com/blog/the-most-popular-databases-2019/bp46/>
- Fehrle, B. (2018). PostgreSQL Running Slow? Tips & Tricks to Get to the Source. *Severalnines*. Recollit de <https://severalnines.com/database-blog/postgresql-running-slow-tips-tricks-get-source>

- FenixFireEntertainment. (2021). Travel to Aziel. *Experimental Build 0.4.221*. Steam.
- Frawley, W., Piatetsky-Shapiro, G., & Matheus, C. (1992). *Knowledge Discovery in Databases: An Overview*. AI Magazine. Recollit de <https://ojs.aaai.org//index.php/aimagazine/article/view/1011>
- Getz, A. (2019). Graph NoSQL Database. *Bi-insider*. Recollit de <https://bi-insider.com/posts/graph-nosql-database/>
- Gola, M. (2019). Difference between structured data & unstructured data. *Curvearro*. Recollit de <https://www.curvearro.com/blog/difference-between-structured-data-unstructured-data/>
- Gupta, P. (s. d.). Entity-Relationship Model. *Educba*. Recollit de <https://www.educba.com/entity-relationship-model/>
- Hunger, M., Boyd, R., & Lyon, W. (2016). RDBMS & Graphs: Relational vs. Graph Data Modeling. *neo4j*. Recollit de <https://neo4j.com/blog/rdbms-vs-graph-data-modeling/>
- I. (2016). What is a Table? *Database.Guide*. Recollit de <https://database.guide/what-is-a-table/>
- Lai, E. (2009). No to SQL? Anti-database movement gains steam. *Computerworld*. Recollit de <https://www.computerworld.com/article/2526317/no-to-sql--anti-database-movement-gains-steam.html>
- Lawtomed. (2019). Structured Data vs. Unstructured Data: what are they and why care? *Lawtomed*. Recollit de <https://lawtomed.com/structured-data-vs-unstructured-data-what-are-they-and-why-care/>
- Lobel, L. (2015). Relational Databases vs. NoSQL Document Databases. *Lenni's Technology Blog*. Recollit de <https://www.computerworld.com/article/2526317/no-to-sql--anti-database-movement-gains-steam.html>

- mack, J. (2014). Five advantages & Disadvantages Of MySQL. *Datarealm*. Consultat el 31 / 3 / 2021, a <https://www.datarealm.com/blog/five-advantages-disadvantages-of-mysql/>
- MongoDB. (s. d.). ACID Transactions in MongoDB. *MongoDB*. Recollit de <https://www.mongodb.com/basics/transactions>
- Omnisci. (s. d.). Relational Database. *Omnisci*. Recollit de <https://www.omnisci.com/technical-glossary/relational-database>
- Pavlovic, D. (2020). Video Game Genres: Everything You Need to Know. *HP*. Recollit de <https://www.hp.com/us-en/shop/tech-takes/video-game-genres>
- Sadalage, P., i Fowler, M. (2012). *NoSQL Distilled: A Brief Guide to the Emerging World of Polyglot Persistence*. Crawfordsville: RR Donnelley.
- Sam, S. (2018). Types of databases. *Tutorialspoint*. Recollit de <https://www.tutorialspoint.com/Types-of-databases>
- Sharma, N., Perniu, L., Chong, R. F., Iyer, A., Nandan, C., Mitea, A.-C., . . . Danubianu, M. (2010). *Database Fundamentals*. Markham: IBM Canada.
- Sheldon, L., Toole, A., Orkin, H., i Despain, W. (2009). *Writing for Video Game Genres: From FPS to RPG* (1a ed.). Natick: A K Peters/CRC Press.
- Shiff, L., i Rowe, W. (2018). SQL vs NoSQL Databases: What's The Difference? *Bmc*. Recollit de <https://www.bmc.com/blogs/sql-vs-nosql/>
- Smallcombe, M. (2020). Structured vs Unstructured Data: 5 Key Differences. *Xplenty*. Recollit de <https://www.xplenty.com/blog/structured-vs-unstructured-data-key-differences/>
- SoftwareAdvice. (2021). Database Management Systems. *Software Advice*. Consultat el 14 / 4 / 2021, a <https://www.softwareadvice.com/database-management-systems/p/all/>
- Soltis, A. E. (2020). Chess: game. *Encyclopedia Britannica*. Recollit de <https://www.britannica.com/topic/chess>

- Statista. (2021). *Ranking of the most popular database management systems worldwide, as of December 2020*. Statista. Recollit de <https://www.statista.com/statistics/809750/worldwide-popularity-ranking-database-management-systems/>
- Taylor, C. (2021). Structured vs. Unstructured Data. *Datamation*. Recollit de <https://www.datamation.com/big-data/structured-vs-unstructured-data/>
- Thakur, S. (2016). Application and Uses of Database Management System (DBMS). *Whatisdbms*. Recollit de <https://whatisdbms.com/application-and-uses-of-database-management-system-dbms/>
- Turk, T. (2018). Globalizing Player Accounts. *Technology Riot Games*. Recollit de <https://technology.riotgames.com/news/globalizing-player-accounts>
- Vanderbush, A. (2017). Data Concurrency Issues in Elasticsearch. *Qbox*. Recollit de <https://qbox.io/blog/data-concurrency-issues-in-elasticsearch/>
- White, A. (2020). MySQL vs PostgreSQL: What's the Difference? *Izenda*. Consultat el 14 / 5 / 2021, a <https://www.izenda.com/mysql-vs-postgresql/>
- Yu, S., i Guo, S. (2016). *Big Data Concepts, Theories, and Applications*. Berlín: Springer.