Desenvolupament d'una eina per estudiar i analitzar l'ELO en els videojocs

EloSim

TFG
Especialització en Enginyeria del Software

Autor: Arnau Gesa Pascual
Director: Manuel Rello Saltor
Ponent: Ernest Teniente López
21 de juny de 2021

Resum

Els videojocs cada vegada estan agafant més popularitat i importància. Per fer un bon joc s'han de contemplar molts aspectes, i un d'ells és el sistema d'Elo. Encara que pot tindre altres interessos, aquest s'encarrega de representar l'habilitat del jugador. D'aquesta manera, el videojoc final és més competitiu i entretingut.

L'objectiu d'aquest projecte no és crear la millor fórmula d'Elo possible, sinó permetre trobar la que més d'adequa a les necessitats de cada empresa. EloSim és un simulador que permet a l'usuari introduir la seva fórmula d'Elo i comprovar si és la desitjada o no. Per poder fer les simulacions tan personalitzables com sigui possible, el client pot introduir ell mateix els jugadors i les mecàniques de joc. D'aquesta manera es pot simular de la forma més real possible i obtenir millors resultats.

A partir de les dades extretes, cada empresa pot trobar la seva fórmula d'Elo i fer un millor videojoc. Com a resultat, aquest tindrà més exit i arribarà a més jugadors, generant així més ingressos.

Resumen

Los videojuegos cada vez están teniendo más popularidad e importancia. Para desarrollar un buen juego hay que contemplar muchos aspectos, y uno de ellos es el sistema de Elo. Aunque puede tener otros intereses, este se encarga de representar la habilidad del jugador. De esta manera, el videojuego final es más competitivo y entretenido.

El objetivo de este proyecto no es crear la mejor fórmula de Elo posible, sino permitir encontrar la que más se adecua a las necesidades de cada empresa. EloSim es un simulador que permite al usuario introducir su fórmula de Elo y comprobar si es la deseada o no. Para poder hacer las simulaciones lo más personalizables posible, el cliente puede introducir él mismo los jugadores i las mecánicas de juego. De esta manera se logra simular de la forma más real posible y obtener mejores resultados.

A partir de los datos extraídos, cada empresa puede encontrar su fórmula de Elo y hacer un mejor videojuego. Como resultado, este tendrá más éxito y llegará a más jugadores, generando así más ingresos.

Abstract

Nowadays, video games are becoming more popular and important. To develop a good game, there are many aspects to be considered. One of them is the Elo system. Although it may have other interests, it is in charge of representing the player's skills. In this way, the final video game is more competitive and enjoyable.

This project does not want to create the best Elo formula, but to find the one that suits better the needs of each company. EloSim is a simulator that allows the user to introduce his/her Elo formula and check if it is the one he/she wanted or not. In order to make the simulations as customizable as possible, the customer can insert the players and the game mechanics himself/herself. So it is possible to simulate in a more realistic way and obtain better results.

From the extracted data, each company can find its Elo formula and create a better video game. As a result, it will be more successful and more players will play it. Therefore, it will generate more income.

Índex

Índex de figures

Índex de taules

1 Contextualització

1.1 Introducció

Aquest projecte anomenat *Desenvolupament d'una eina per estudiar i analitzar l'ELO en els videojocs* és un Treball de Fi de Grau (TFG) realitzat en la Facultat d'Informàtica de Barcelona (FIB) de la Universitat Politècnica de Catalunya (UPC). En aquest cas, és un treball realitzat per al Grau en Enginyeria Informàtica, concretament, per a l'especialitat d'Enginyeria del Software. L'objectiu del projecte és crear un simulador de partides per comprovar i analitzar les fórmules de l'Elo.

1.2 Situació actual

Avui en dia, gràcies a l'Internet i a l'evolució i disponibilitat de les noves tecnologies, tothom pot jugar a un videojoc des d'un dispositiu mòbil, ordinador o consola. No és estrany dedicar-hi temps en els moments lliures, ja sigui per entretenir-se amb els amics, per divertir-se o per competir contra altres jugadors; o en els temps morts del dia, com per exemple quan es fa cua en un supermercat.

Des de l'aparició i gran èxit del Pong fins al dia d'avui, els videojocs han evolucionat molt i la indústria ha crescut notablement. Han aparegut diversos gèneres de jocs, entre els quals es poden destacar els d'acció, d'aventura, de lluita o d'esports. També, gràcies a l'aparició d'Internet, s'ha afegit la component online als videojocs, la qual permet jugar amb gent d'arreu del món. D'altra banda, també s'han creat diverses competicions molt importants a escala mundial, com serien el cas de la Call of Duty World League (CWL), el campionat del League of Legends [1] o el torneig de lluita de fama mundial EVO [2].

Altrament, les plataformes que permeten pujar vídeos a la web també s'han vist beneficiades pels videojocs. Un bon exemple d'això és You Tube. Aquest ha arribat a tenir més de 100 mil milions d'hores en vídeos d'aquest aspecte i més de mil milions d'hores en transmissions en viu de la mateixa temàtica [3]. Un altre exemple és la plataforma de Twitch, la qual s'especialitza a fer transmissions en viu. En aquest cas, ha aconseguit que jocs com el League of legends o el Fortnite hagin tingut més de 35 o 20 mil milions de visites en el darrer any respectivament. [4].

Com a resultat, la indústria dels videojocs segueix augmentant cada cop més. Avui en dia, generen una gran quantitat d'ingressos, la qual s'espera que s'incrementi encara més en els anys vinents. Com es pot veure en la gràfica de la figura 1, s'observa que en 2019 es van generar més de 150 mil milions de dòlars i es preveu que en 2020, a causa de la Covid-19, se'n generin més de 179 mil milions [5].

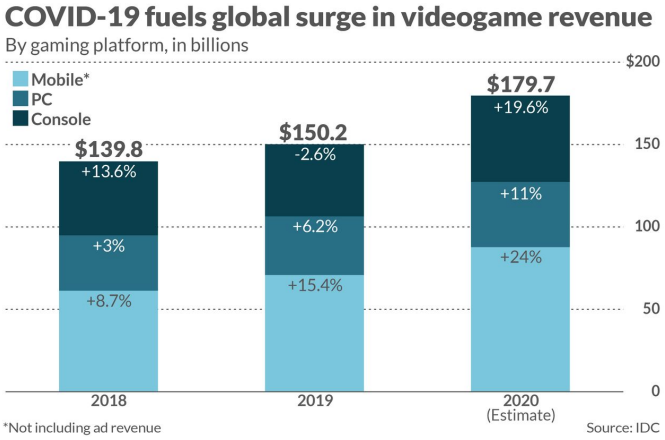

Figura 1: Ingressos dels videojocs. Font: [5]

1.3 Termes i conceptes bàsics

En aquest apartat s'expliquen els termes i conceptes relacionats amb el projecte que poden donar lloc a la confusió o desconeixement del significat pel lector.

- **Elo:** Es coneix com a Elo el sistema de puntuació creat pel físic estatunidenc d'origen hongarès Arpad Elo, que mesura els nivells d'habilitat relatius dels jugadors en els esports. En són uns bons exemples els escacs o els videojocs competitius com el League of Legends o el Counter Strike. [6]
- **Matchmaking:** S'anomena matchmaking a l'algorisme utilitzat en la majoria dels videojocs multijugador que tracta d'emparellar els jugadors o equips d'un nivell o habilitats similars per enfrontar-se entre ells. Això s'aconsegueix mitjançant l'anàlisi de les estadístiques dels diferents usuaris. [7].
- **Videojoc competitiu:** S'entén com a videojoc competitiu un videojoc on els jugadors competeixen entre ells, sigui en equip o en individual, per ser uns millors que els altres. Aquests poden ser online o offline, és a dir, que es poden jugar a través d'Internet o no. Uns bons exemples en són els escacs o els jocs de lluita, com el Street Fighter o Mortal Kombat.
- **Habilitat del jugador:** Es coneix com a habilitat del jugador a la capacitat que té aquest per demostrar la seva destresa en un joc, en altres paraules, que tan capaç és de derrotar als seus contrincants.
- **Emparellar:** Acció d'agrupar a una o més persones per formar un equip. Si s'agrupen dos o més equips es forma una partida.
- **Equip:** Un equip és un conjunt de jugadors aliats dintre del videojoc que tenen el mateix objectiu. Generalment han de col·laborar entre ells per assolir-lo.

1.4 Identificació del problema

La popularització i gran creixement dels videojocs fan que la indústria segueixi agafant més importància: cada cop són més coneguts, hi ha més inversió i generen més ingressos. Per aquests motius, és molt important dissenyar-los bé i per fer-ho, s'han de considerar molts aspectes, com per exemple la temàtica, els escenaris, els personatges, etc. Però també és molt important la jugabilitat, fer que sigui divertit i entretingut. A més, dintre dels jocs competitius, es necessita un bon sistema d'Elo i d'aparellament, per fer encara el videojoc més entretingut i generar més rivalitat, tot i que també pot haver-hi interessos comercials.

L'Elo pot ser molt diferent depenent de la situació en què es vulgui aplicar. Per exemple, si en un videojoc es fa un campionat setmanal, on es vol reduir el nombre de jugadors durant el transcurs d'aquest, es pot generar un algorisme que contempli un Elo molt variat entre els diferents participants al principi de la competició, però que sigui més similar cap al final. Per una altra banda, si es juga un campionat mensual on es vol que hi participin tots els jugadors i es desitja una classificació apropiada a l'Elo de cada un, es pot crear un algorisme que agrupi les partides mitjançant un Elo similar.

Per tant, desenvolupar una bona fórmula de l'Elo equival, normalment, a representar el nivell d'habilitat del jugador i al seu progrés a l'hora de millorar. Tot i que sembla una idea senzilla de realitzar, a la pràctica no ho és. L'Elo ha d'augmentar progressivament respecte a l'habilitat del jugador. Teòricament, quantes més partides es juguin, més millora l'usuari i més ha d'augmentar la seva puntuació. També ha de ser estable i consistent, no pot ser que d'un dia per l'altre hi hagi una gran diferència de punts. A més a més, també s'ha de considerar quina puntuació es necessita per arribar a un rang d'Elo bo, no pot ser que tothom es trobi entre els millors. Per tant, s'ha d'estudiar rigorosament com fer el seu algorisme i saber quines variables la modificaran (guanyar, perdre, morir més o menys, etc.).

Com es pot observar, s'han de fer diverses proves per arribar a la fórmula adequada. El problema és que no hi ha cap programa que permeti simular les diferents versions per saber quina és vàlida i quina no. El que s'ha de fer actualment és plantejar una simulació que s'adeqüi a l'algorisme i fer moltes partides per veure l'evolució. Això pot tenir conseqüències, i és que testejar i comparar els resultats de les diferents fórmules pot portar molt temps, el qual es podria invertir en altres aspectes del desenvolupament del videojoc.

És per aquest motiu que és necessària una eina software per a les empreses desenvolupadores de jocs amb interès comercial, que els permeti simular partides amb l'algorisme d'Elo que se li introdueixi i avaluar els resultats. D'aquesta manera, podran escollir quin s'adapta millor a la jugabilitat i progressió del joc.

1.5 Exemples de sistemes d'Elo en els videojocs

L'Elo es pot calcular de diverses fórmules, des de la més simple que és la d'obtenir punts depenent de si es guanya o es perd, fins a la de contemplar el progrés fet en la partida. A continuació es mostraran alguns exemples de sistemes d'Elo utilitzats en videojocs molt coneguts:

- **Escacs:** Els escacs és un joc d'estratègia que consisteix en partides d'u contra u on l'objectiu és capturar el rei enemic. Aquest utilitza el mètode més senzill per calcular la puntuació d'Elo i és el següent: és mesura la probabilitat de guanyar al rival més la de fer taules (empatar). Després, depenent del resultat i del rang on es jugui la partida, es guanyen o perden més punts d'Elo [6].
- **Rocket League:** El Rocket League és un videojoc de futbol, amb la diferència que en canvi de jugar amb futbolistes, es juga amb cotxes. En aquest cas, l'Elo s'anomena MMR (MatchMaking Rank) i funciona similar al dels Escacs. Si es guanya al rival, s'obté MMR i si es perd, es resta. La quantitat pot variar depenent del rang en el qual es trobi el participant i del nivell de l'equip rival. El progrés en la partida, com ara marcar gols, fer parades o assistències, no contribueix a obtenir més punts d'Elo [8].
- **League of Legends:** El League of Legends o LOL és un joc d'estratègia per equips on l'objectiu és enderrocar una estructura anomenada Nexus que es troba en la base del rival. Aquest videojoc funciona per rangs i per pujar-ne d'un a un altre, s'ha d'acumular una certa quantitat de LP (League Points). Si el jugador guanya la partida, n'adquirirà més o menys depenent del nivell del rival. En canvi, si la perd, se li restaran. La diferència amb els videojocs anteriors és que un cop s'ha adquirit els LP suficients per pujar de rang, és necessari jugar unes fases d'ascensió. Si es passen, s'assolirà la categoria superior i s'obtindran els LP corresponents. En canvi, si es perd, els LP seran restablerts al del rang que pertoca [9].
- **Counter Strike: Global Offensive:** El Counter Strike: Global Offensive o CSGO és un videojoc en primera persona que es juga per equips. Les partides duren diverses rondes on hi han diferents objectius per complir per guanyar-les. El principal és eliminar a tots els rivals. Els altres depenen del rol de cada equip. Es poden diferenciar dos grups: terroristes i antiterroristes. Els primers tenen una bomba, la qual han de plantar i fer-la explotar per guanyar. Els segons han d'evitar que l'explosiu sigui activat, i en cas de que passi, l'han de desactivar. En aquest joc el sistema d'Elo funciona de manera diferent: cada ronda que es juga és com si fos una partida d'escacs. Els punts d'Elo es veuen repartits al final de cada una d'elles. La diferència està en el fet que aquest joc si contempla el nombre de morts que ha tingut un jugador, el número de baixes, si ha complert l'objectiu, etc. per repartir els punts d'Elo. També té molt en compte qui ha estat el MVP de la ronda, enduent-se una gran part dels punts [10] [11].

1.6 Motivació

Els videojocs és un aspecte que m'ha apassionat tota la vida. Des de ben petit fins al dia d'avui que els porto jugant i gaudint. De fet, el quadrimestre passat vaig cursar l'assignatura de videojocs per tal de veure i entendre com funcionaven.

Durant les primeres setmanes del mes de desembre vaig estar pensant quina temàtica escollir per fer el TFG. Van sorgir algunes idees, però cap d'elles em van acabar de convèncer, així que vaig optar per demanar ajuda. Vaig preguntar a diversos professors que havia tingut durant la carrera per veure si em podien donar alguna idea.

Tots ells em van respondre amablement amb propostes molt interessants, però va haver-hi una que em va cridar molt l'atenció. Aquesta és la del meu TFG, i que consisteix a fer un simulador d'Elo per estudiar aquest mateix i veure com evolucionava.

Van haver-hi dos motius pels quals vaig escollir aquesta idea: el primer és que era una temàtica sobre els videojocs, que m'agraden molt. El segon és que és un aspecte que mai m'havia plantejat com funcionava i que malauradament no s'havia estudiat en l'assignatura. Per tant, vaig escollir aquest TFG per augmentar el meu coneixement respecte als videojocs i sobre com estan fets.

2 Competència en el mercat

2.1 Solucions existents

El sistema d'Elo d'un videojoc és molt important, ja que ha de complir amb molts objectius, com el de representar el nivell d'habilitat del jugador. A més, ha de ser utilitzat per aconseguir fer les partides entretingudes. Llavors, es pot arribar a la conclusió de què ha d'estar ben desenvolupat perquè un joc triomfi. És per aquest motiu que s'ha realitzat una recerca de programes que simulin i permetin configurar el sistema d'Elo d'un videojoc.

Malauradament, no s'ha trobat cap programa o aplicació web que permeti introduir diverses fórmules d'Elo i simular partides per poder analitzar el seu resultat. Només s'han trobat repositoris de Github que simulen el seu càlcul, on utilitzen una fórmula predeterminada, on a vegades es poden canviar algunes variables. També s'ha trobat una pàgina web que simula el càlcul d'Elo, però fa el mateix que els repositoris. Els millors resultats obtinguts han estat els següents:

- **Repositori de RemiFabre [12]:** En aquest repositori anomenat "Elo" es troba un programa fet en python que simula l'evolució de l'Elo en diversos jugadors amb puntuacions diferents.

  Aquest programa té dues modalitats, en la primera és que els jugadors només s'enfronten entre si una vegada rere l'altre, mentre que en la segona modalitat juguen contra jugadors creats només per aquella partida i amb un "winrate" fixat. L'algorisme de l'Elo està predeterminat i com a molt es pot canviar el valor de certes variables.

  El resultat de les diverses partides es mostren en una gràfica en el mateix programa, on es pot veure com augmenta o decreix l'Elo de cada jugador.

- **Repositori de cardsorg [13]:** Aquest repositori de Github anomenat "Elo" inclou un software que simula l'enfrontament entre dos jugadors i diu el respectiu Elo de cadascú.

  Té tres funcionalitats diferents:

  - Enfrontar a dues persones amb l'Elo i el resultat que es vulguin assignar. També permet canviar el valor de la variable K, que és el màxim de punts que pot guanyar o perdre un jugador en una partida.
  - Mostrar la diferència entre els dos jugadors. Després simular un seguit de partides, calcula la diferència d'Elo entre els participants
  - Mostrar la diferència estimada i real d'Elo dels jugadors. Funciona igual que la segona funcionalitat, però, a més, també mostra la diferència estimada.

- **Repositori d'iain [14]:** En aquest tercer repositori anomenat "Elo" es troba un altre programa que permet simular i modificar alguna variable del sistema d'Elo estandard.

  En aquest cas, es poden crear jugadors amb la puntuació desitjada i enfrontar-los entre si, decidint el resultat. També permet diferents configuracions, com la modificació de la variable K, on es pot definir depenent del nombre de partides fetes. Addicionalment, permet modificar a partir de quants punts es deixa de ser un principiant, o amb quina puntuació es comença.

El software et mostra el número de partides jugades de cada jugador, el seu nivell d'Elo i saber si és professional o principiant.

- **Elo Calculator [15]:** Aquesta calculadora online de la pàgina web "Omnicalculator" permet calcular l'Elo d'un jugador després de fer diferents jocs i també modificar la variable K.

  Hi ha dos opcions a escollir: una única partida o múltiples d'elles. La calculadora deixa assignar els punts desitjats al jugador principal i a cada un dels seus rivals, permetent també escollir el resultat de cada enfrontament.

  Al final, mostra una taula amb quants punts s'han guanyat i perdut en cada joc i quina és la puntuació final.

2.2 Justificació de la meva proposta

Un cop vistos tots els productes que es troben actualment a Internet, es pot concloure que tots ells tenen aspectes positius: permeten modificar la variable K, mostren un gràfic de com va evolucionant l'Elo després de diverses partides, permeten escollir el resultat de cada joc o ensenyen quants punts es guanyen o es perden. Tot i això, tots ells estan lligats a la mateixa fórmula, i és aquesta la principal diferència amb el projecte desenvolupat.

La solució proposada, anomenada EloSim, també permet modificar el valor de diverses variables, simular partides i exportar els resultats per fer una gràfica. Però el més important és que permetrà canviar l'algorisme per calcular l'Elo en la seva totalitat. Es podrà utilitzar la fórmula que es desitgi, ja sigui la predeterminada o una creada per l'usuari que dissenyi el sistema. D'aquesta forma es podrà comprar entre els diferents algorismes i escollir el més adequat pel videojoc. A més, per tal de simular millors les partides, també permetrà canviar el sistema de joc i crear jugadors amb les característiques desitjades. Aquestes poden fer referència a dades reals d'usuaris, a models generats a través d'una mineria de dades, etc. Tot això fa el simulador més dinàmic i més adaptable, ajustant-se a la voluntat de l'empresa.

A continuació, per tal d'aclarir les diferències entre els diferents programes, en la taula 1 es pot observar una graella comparativa entre les distintes funcionalitats dels diversos programes trobats i el meu.

Funcionalitats	RemiFabre	cardsorg	iain	EloCalculator	El meu projecte
Fórmula d'ELO determinada	✓	✓	✓	✓	✓
Canviar variables algorisme d'ELO	✓	✓	✓	✓	✓
Simulació amb gran nombre de partides	✓				✓
Personalitzar puntuació de jugadors		✓	✓	✓	
Decidir resultat d'un enfrontament		✓	✓	✓	
Mostrar resultat en gràfiques	✓				
Personalitzar algorisme d'ELO					✓
Personalitzar algorisme de partida					✓
Personalitzar creació de jugadors					✓
Exportar dades a Excel					✓

Taula 1: Taula comparativa de les diferents funcionalitats dels programes. Font: Elaboració pròpia

3 Abast

L'abast d'un projecte defineix els seus objectius, els obstacles i riscos que poden sorgir i quines són les parts interessades en el software. Per tant, tots aquests aspectes s'han d'identificar i definir per assolir un bon producte.

3.1 Objectius

L'objectiu principal d'aquest projecte és poder crear un producte software que permeti simular el càlcul de l'Elo en els videojocs. Així, es poden obtenir dades per veure la seva evolució, per determinar si és un algorisme vàlid i per comprovar si l'aparellament és el correcte. Per aconseguir que el programa pugui complir amb aquestes funcionalitats, cal assolir un seguit de sub-objectius:

- **Personalització de l'algorisme del sistema d'Elo:** El programa ha de permetre modificar la fórmula sencera o poder canviar algunes de les seves variables. Així es podrà introduir qualsevol algorisme i es podrà veure el seu funcionament.
- **Simulació de partides:** El software ha de permetre simular partides per així poder aplicar la fórmula de l'Elo i comprovar que l'aparellament de jugadors és el correcte. S'ha de contemplar que poden haver-hi variables que afectin el resultat de la partida. Per exemple un jugador pot tenir una arma millor que l'altre, una armadura més resistent, un cotxe més ràpid, etc. També ha de tenir en compte la diferència d'habilitat entre els contrincants, el resultat esperat vs. el real i la quantitat de punts d'Elo guanyats o perduts per cada partida.
- **Comparar resultats entre diferents variables i algorismes:** Per poder determinar si la fórmula de l'Elo és la correcta, cal comparar els resultats entre els diferents algorismes. És per aquest motiu que es guardaran les dades i es permetrà exportar-les a un fitxer amb format .csv, per poder analitzar-les i comparar-les.

A la figura 2 es pot observar un esquema dels diversos objectius i subojectius del programa.

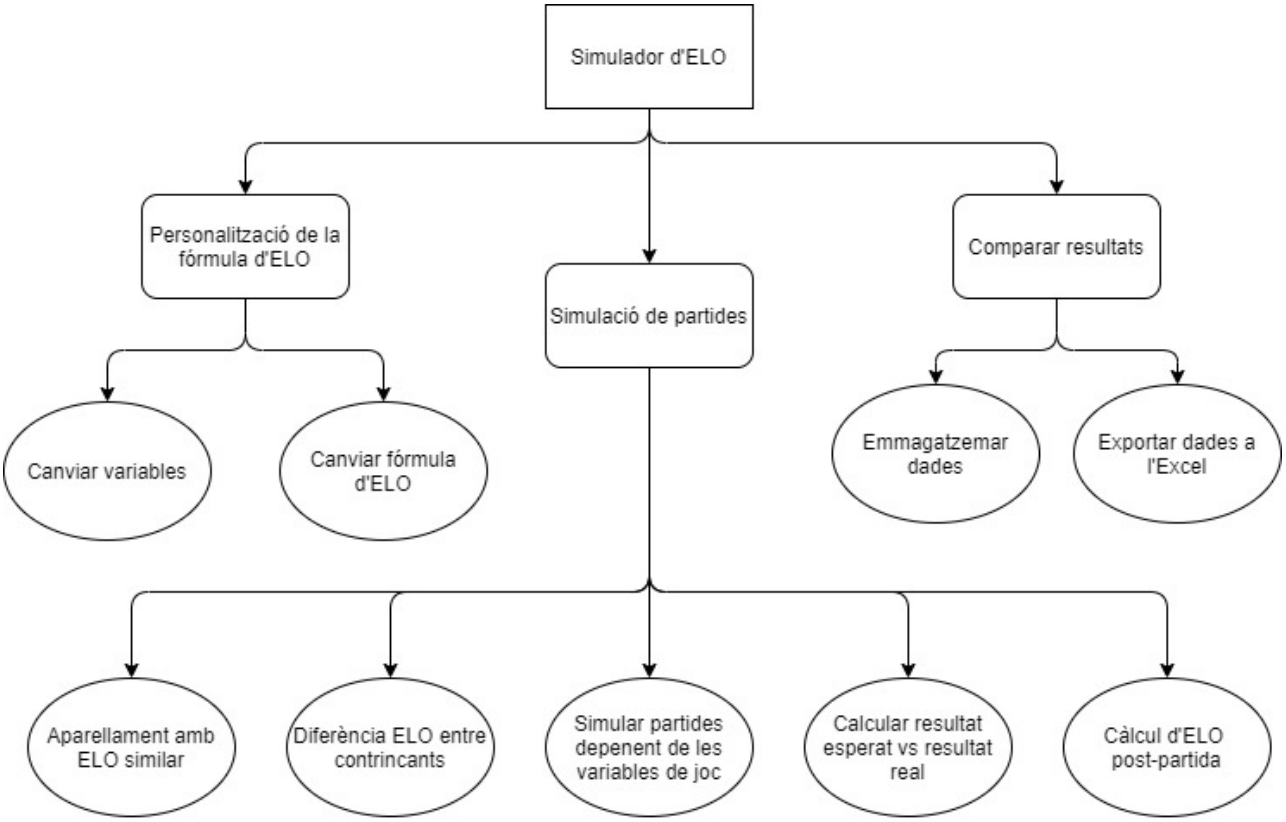

Figura 2: Esquema dels objectius i subobjectius del sistema. Font: Elaboració pròpia

3.2 Obstacles i riscos

Durant tot el transcurs del projecte poden aparèixer diversos obstacles i riscos que alenteixin el desenvolupament del producte. Per aquest motiu, cal identificar-los i saber-ne d'ells.

- **Data d'entrega fixa:** El TFG té una data límit fixada i, per tant, que s'ha de complir. Però els possibles problemes que es poden trobar durant el desenvolupament del projecte poden generar endarreriments i, per tant, empitjorar la qualitat del programa o eliminar alguna funció no essencial per falta de temps.
- **Desconeixement en simulació:** Un aspecte fonamental del programa és la simulació de partides. Però a causa del seu desconeixement en l'àmbit de la informàtica, és probable que es generin endarreriments i possibles errors a l'hora de programar.
- **Desconeixement en les tecnologies emprades:** A causa del desconeixement sobre el llenguatge de programació Lua o sobre el poc ús fet del software de QT, és probable que s'alenteixi el desenvolupament del projecte, ja que s'haurà d'aprendre a utilitzar-los correctament.

- **Bugs i errors:** Durant tot el transcurs de programar i testejar el programa, poden aparèixer diferents errors i bugs que facin fer perdre molt temps, i per tant, endarrereixin el desenvolupament.

3.3 *Stakeholders*

Els *Stakeholders* són una part fonamental pel desenvolupament del projecte, ja que són totes aquelles persones o organitzacions afectades pel producte a crear i que tenen interès en aquest, ja sigui de forma directa o indirecta. A continuació es llistaran les parts interessades del software a desenvolupar.

- **Empreses de videojocs:** Les empreses desenvolupadores de videojocs es veuran principalment afectades pel projecte. El seu objectiu principal és generar ingressos econòmics. Però per fer-ho, necessita desenvolupar un bon joc. Un aspecte important per aconseguir-ho és un bon sistema d'Elo, que el poden arribar a obtenir gràcies al simulador. Per tant, quan més èxit tingui el joc, més conegut serà, més jugadors captarà i més beneficis obtindrà l'empresa.
- **Desenvolupadors de videojocs:** Els desenvolupadors de videojocs són qui utilitzaran el software de forma directe. Seran els encarregats d'estudiar i plantejar les diferents fórmules del sistema d'Elo i simular-les en l'eina a desenvolupar. D'aquesta manera podran decidir quina d'elles s'adapta més al videojoc i farà que hi hagi una millor experiència.
- **Jugadors:** Els jugadors de videojocs són les principals parts interessades i són les més importants. Es veuran implicats de forma indirecta, ja que són els que jugaran i seran puntuats pel sistema d'Elo resultant. També són qui hauran de trobar el joc entretingut i divertit, atès que seran els encarregats de fer que aquest tingui èxit.
- **Propietari del TFG:** Jo mateix soc un dels principals *steakeholders* del projecte, ja que seré l'únic desenvolupador. El meu objectiu serà fer-lo de la millor forma possible i eficient, testejar-lo i documentar tot el procés.

4 Metodologia i rigor

La metodologia amb la qual es desenvolupa un projecte és un aspecte molt important a tenir en compte, ja que pot canvia per complet com evoluciona. Hi han dues grans metodologies: la cascada o *waterfall* i l'*agile*.

La primera és la més tradicional. Aquesta té en compte els *stakeholders* i clients al principi del projecte, i després es desenvolupa un pla seqüencial per satisfer els seus requisits. S'anomena *waterfall* perquè cada fase del pla proposat passa a la següent un cop s'ha acabat l'anterior [16].

La metodologia cascada és utilitzada quan els requeriments estan molt clars i fixes, la definició del producte és estable, la tecnologia no evoluciona constantment i el projecte és curt [17].

Per una altra part, l'*agile* consisteix en una contínua iteració de desenvolupament i testeig del software on s'interacciona de forma freqüent amb els clients i *stakeholders* per saber el seu *feedback* [18].

Aquesta és utilitzada per projectes llargs on es vol el *feedback* del client constantment, quan es vol treballar amb tecnologies en evolució continua, també quan els requeriments no estan clars i poden canviar freqüentment, igual que la definició del producte. Per regla general, se la considera la contrapart de la metodologia *waterfall*.

Dintre de la metodologia *agile* es poden trobar diferents processos, com és el cas de Scrumm o Kanban. La primera es caracteritza per dividir el temps de desenvolupament disponible en diferents iteracions, anomenades sprints. La seva duració sol ser fixa, de dos o tres setmanes. Al principi de cada iteració es fa una planificació d'aquesta, on s'escull la feina a fer. Durant cada dia d'aquest període, es fa una reunió diària de 15 minuts on s'explica quina feina ha fet cada membre de l'equip, quins problemes ha trobat i que es planteja fer. Al final del sprint, es fa una revisió i presentació de la feina feta, i una retrospectiva de la iteració per veure que es pot millorar de cara a la següent [19]. A la figura 3 es pot veure un esquema simplificat del procés Scrumm.

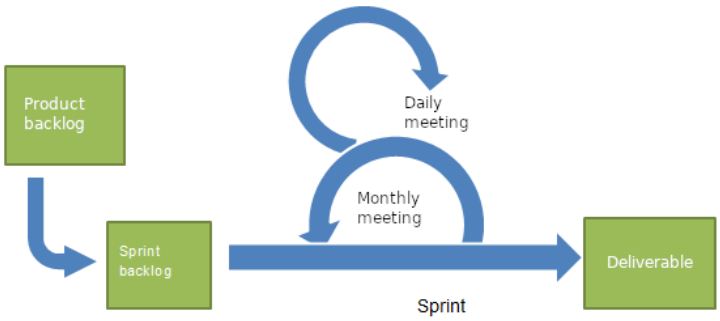

Figura 3: Fases del procés Scrumm. Font: [19]

Per una altra part, el procés Kanban consisteix a gestionar un projecte de manera general i de forma visual. Cada tasca es crea físicament, normalment en format de nota on s'especifica informació sobre la feina a realitzar, i s'enganxa en un taulell. Aquest està dividit en diferents columnes on es pot veure la feina que està per fer, la que s'esta fent, la que s'esta testejant i la que s'ha acabat [20]. A la figura 4 es pot veure un esquema de com és un taulell del procés Kanban.

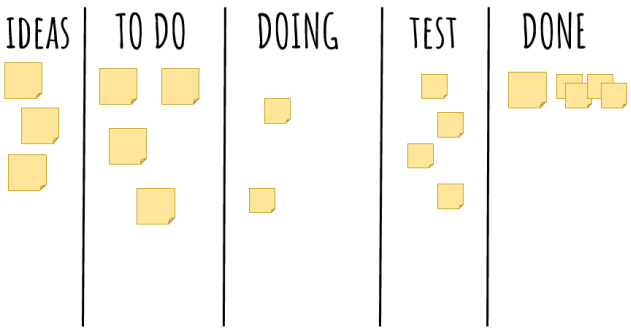

Figura 4: Taula del procés Kanban. Font: [21]

Un cop explicades les dues metodologies, i els processos de l'*agile*, cal explicar quina segueix aquest projecte. Per una banda, reuneix moltes característiques de la *waterfall*: és un període curt, ja que el TFG dura entre tres i quatre mesos, els requeriments són clars i fixes i la idea del producte és estable. Per una altra banda, una metodologia *agile* on es testegin les parts implementades de forma contínua i es segueixi el concepte del Scrumm, pot ajudar a desenvolupar el projecte de forma exitosa. Addicionalment, mostrar les tasques en el format Kanban pot beneficiar la planificació del treball a realitzar. Per tant, aquest projecte segueix una metodologia Kanban *agile*. Però per poder facilitar el seu desenvolupament, s'ha decidit agafar matisos del Scrumm i del *waterfall*.

4.1 Control de versions

Durant el desenvolupament del projecte s'ha utilitzat Git, un software de control de versions, per poder treballar amb diferents variants del programa. També s'ha utilitzat Github, un sistema de control de versions, per pujar tot el codi del treball i tota la documentació necessària. D'aquesta manera, el director ha pogut veure el progrés i analitzar com s'ha anat implementant la solució. Addicionalment, s'ha pogut consultar i accedir a qualsevol versió anterior.

4.2 Control de seguiment

Durant el transcurs de tot el TFG, s'ha fet servir Google Meet, un servei per fer videotrucades, per mantenir al director informat i poder-li comentar qualsevol dubte o error. A principi de curs es va acordar de fer, sempre que fos possible i necessari, una reunió per setmana. D'aquesta manera es podria discutir el progrés del projecte i observar si s'anava pel bon camí.

4.3 Control de tasques

El Trello s'ha utilitzat per tenir estructurades les diverses tasques a realitzar durant tot el projecte. Cada gran funcionalitat ha estat descomposta per petites tasques per tal de facilitar la implementació. A més, s'han fet quatre columnes per organitzar bé l'estat del projecte: TO DO, DOING, READY, DONE. D'aquesta manera, es segueix la metodologia Kanban.

5 Especificació de requisits

Per tal d'assolir el bon funcionament del sistema i complir tots els objectius mencionats prèviament, aquest projecte ha d'assolir un seguit de requisits funcionals i no funcionals. A continuació es mostraran tots dos tipus de requisits.

5.1 Requisits funcionals

El simulador que es desenvolupa en aquest treball disposa d'un seguit de requisits funcionals que permeten a l'usuari fer-ne ús del sistema. Seguidament, s'enumeraran cada un d'ells:

- **Entrada de fitxers Lua:** El programa ha de permetre carregar un fitxer de script en Lua que contingui diverses característiques i funcions per poder personalitzar la simulació. En són un exemple l'algorisme del sistema d'Elo o les característiques dels jugadors.
- **Configuració de la fórmula del sistema d'Elo:** El sistema ha de poder entendre i executar perfectament la fórmula d'Elo que hagi introduït l'usuari mitjançant el script de Lua.
- **Configuració de la fórmula per jugar la partida:** El sistema ha de poder llegir i executar la fórmula de joc de les partides per tindre en compte totes les propietats que vol l'usuari.
- **Creació dels jugadors per fer les simulacions:** El sistema ha de crear el nombre de jugadors necessaris per poder executar la simulació. Cada un d'ells ha de tenir les propietats introduïdes per l'usuari.
- **Configuració de les característiques dels jugador:** El client ha de ser capaç de modificar les característiques o propietats que han de tenir els jugadors mitjançant el fitxer Lua.
- **Creació d'equips:** El sistema ha de permetre crear equips tenint en compte les dades introduïdes per l'usuari. Aquestes són la diferència d'Elo entre jugadors i el nombre de membres per equip. Cada un tindrà una diferència màxima inferior o igual a l'especificada i un nombre de components idèntics a l'indicat.
- **Creació de les partides:** El sistema ha de permetre crear partides tenint en compte la diferència màxima d'Elo que pot haver-hi entre els dos equips. Aquesta dada haurà estat introduïda per l'usuari prèviament.
- **Simulació d'una partida:** El sistema ha de simular una partida entre els equips i determinar un resultat. Per fer-ho, tindrà en compte el sistema implementat per l'usuari en el fitxer Lua.
- **Càlcul de l'Elo:** El software ha de permetre poder executar l'algorisme de l'Elo i modificar la puntuació de cada un dels jugadors de la forma adequada.
- **Configuració de la simulació:** El sistema ha de permetre la configuració d'una sèrie de variables per fer la simulació. Concretament ha de configurar: el nombre de jugadors, la quantitat de components d'un equip, el número d'equips per partida, el nombre de partides a simular, la diferència de puntuació entre els jugadors d'un equip i un enfrontament i la destinació de les estadístiques.

- **Emmagatzematge de dades:** El software ha de poder emmagatzemar les dades extretes de les simulacions per utilitzar-les i actualitzar-les més tard.
- **Sortida de fitxers Excel:** L'eina a desenvolupar ha de poder permetre exportar les dades obtingudes a un fitxer Excel per així analitzar-les i comparar-les.

5.2 Requisits no funcionals

Els requisits no funcionals representen característiques o restriccions del software que es desenvolupa. Com tot projecte, EloSim també n'ha de complir diversos d'ells. Aquests s'han extret i consultat de la plantilla Volere [22].

Tipus de requisit	**10a. Aparença**
Descripció	El producte ha de ser atractiu pels usuaris que l'utilitzin. S'han d'escollir els colors i els elements adequats per fer la interfície gràfica agradable.
Justificació	Un bon disseny facilitarà l'ús del programa i el farà més intuïtiu. A més, si és agradable, farà més amena l'experiència de l'usuari.

Tipus de requisit	**11a. Facilitat d'ús**
Descripció	El programa ha de ser intuïtiu i fàcil d'usar. Ha de poder ser utilitzat per a qualsevol tipus d'usuari.
Justificació	Un sistema intuïtiu fa estalviar temps a l'usuari, pel simple fet que no ha d'aprendre com funciona. Addicionalment, qualsevol ha de ser capaç de familiaritzar-se amb el software, ja que pot ser utilitzat per gentde diverses edats.

Tipus de requisit	**11a. Aprenentatge**
Descripció	El sistema ha de ser fàcil d'aprendre a utilitzar per qualsevol usuari.
Justificació	Qualsevol usuari ha de ser capaç d'executar el programam i treure-li profit el més ràpid possible.

Tipus de requisit	**12c. Precisió o exactitud**
Descripció	El programa ha de treballar amb la precisió desitjada per l'usuari a l'hora de fer les simulacions.
Justificació	Per poder aprofitar i contemplar bé els resultats, el programa ha de fer les simulacions amb la precisió indicada pel client. D'aquesta manera es podrà estudiar l'Elo des de diferents casos.

5.3 Casos d'ús

Un cop s'han analitzat els requisits funcionals i no funcionals, cal identificar i explicar els diferents casos d'ús del programa. Com es pot observar en la figura 5, s'han trobat un total de vuit. A continuació s'explicarà cada un d'ells.

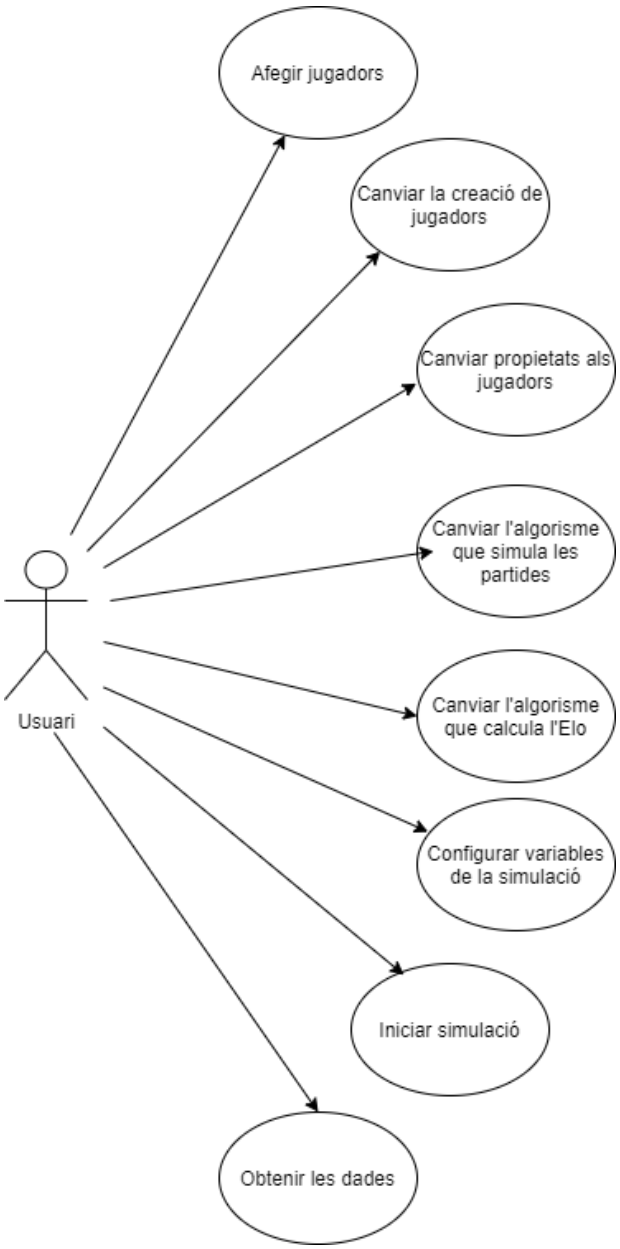

Figura 5: Casos d'ús del sistema. Font: Elaboració pròpia

1. Afegir jugadors

Actor principal	Usuari
Precondició	L'usuari ha accedit al fitxer Lua amb un editor.
Disparador	L'usuari vol afegir manualment els jugadors a la simulació.
Escenari principal d'èxit	1. L'usuari accedeix a la taula "playerProperties" 2. L'usuari afegeix manualment els diferents jugadors en la taula. 3. L'usuari afegeix les propietats de cada jugador. 4. L'usuari desa el fitxer Lua.
Extensions	2a. L'usuari no afegeix els jugadors a la taula. 2a1. El programa crea els jugadors de forma aleatòria. 3a. L'usuari no afegeix en primer lloc la propietat ID. 3a1. El programa dona error a l'hora de crear els jugadors. 3a2. No s'executa la simulació. 3b. L'usuari no afegeix les propietats especificades en la taula "properties". 3b1. El programa dona error a l'hora de crear els jugadors. 3b2. No s'executa la simulació 4a. L'usuari no desa el fitxer Lua. 4a1. El programa crea els jugadors com estava definit prèviament.

2. Canviar la creació de jugadors

Actor principal	Usuari
Precondició	L'usuari ha accedit al fitxer Lua amb un editor.
Disparador	L'usuari vol canviar com la simulació crea els jugadors
Escenari principal d'èxit	1. L'usuari accedeix a la funció "getPlayerProperties" 2. L'usuari afegeix manualment l'algorisme de creació de jugadors. 3. L'usuari desa el fitxer Lua.
Extensions	2a. L'usuari no afegeix un algorisme vàlid. 2a1. Hi ha un error al crear els jugadors. 2a2. No s'executa la simulació. 2b. L'algorisme no contempla les propietats especificades en la taula "properties". 2b1. El programa dona error a l'hora de crear els jugadors. 2b2. No s'executa la simulació 3a. L'usuari no desa el fitxer Lua. 3a1. El programa crea els jugadors com estava definit prèviament.

3. Canviar propietats als jugadors

Actor principal	Usuari
Precondició	L'usuari ha accedit al fitxer Lua amb un editor.
Disparador	L'usuari vol canviar les propietats que tenen els jugadors.
Escenari principal d'èxit	1. L'usuari accedeix a la taula "properties" 2. L'usuari afegeix manualment les diferents propietats en la taula. 3. L'usuari desa el fitxer Lua.
Extensions	2a. L'usuari no afegeix cap propietat a la taula. 2a1. Hi ha un error al crear els jugadors. 2a2. No s'executa la simulació. 2b. L'usuari no afegeix les propietats ID i Elo 2b1. Hi ha un error al crear els jugadors. 2b2. No s'executa la simulació. 3a. L'usuari no desa el fitxer Lua. 3a1. El programa crea els jugadors com estava definit prèviament.

4. Canviar l'algorisme que simula les partides

Actor principal	Usuari
Precondició	L'usuari ha accedit al fitxer Lua amb un editor
Disparador	L'usuari vol canviar l'algorisme que simula les partides
Escenari principal d'èxit	1. L'usuari accedeix a la funció "playMatch" 2. L'usuari afegeix manualment l'algorisme per determinar el resultat de la partida. 3. L'usuari retorna la classificació 4. L'usuari desa el fitxer Lua.
Extensions	2a. L'usuari introdueix un algorisme que no funciona correctament 2a1. El programa dona error al simular la partida 2ab. La simulació no s'executa 3a. La funció no retorna la classificació de la manera que ho espera el programa. 3a1. El programa classifica els equips d'una manera diferent 3a2. El resultat de la simulació es veu afectat greument. 3b. La funció retorna el resultat amb un format diferent a l'esperat. 3b1. El programa dona error a l'hora de simular la partida. 3b2. No s'executa la simulació 4a. L'usuari no desa el fitxer Lua. 4a1. El programa simula com estava definit prèviament.

5. Canviar l'algorisme que calcula l'Elo

Actor principal	Usuari
Precondició	L'usuari ha accedit al fitxer Lua amb un editor
Disparador	L'usuari vol canviar l'algorisme que calcula l'Elo resultant
Escenari principal d'èxit	1. L'usuari accedeix a la funció çalculateEloScore" 2. L'usuari afegeix manualment l'algorisme per determinar l'Elo resultant. 3. La funció retorna la quantitat d'Elo a canviar 4. L'usuari desa el fitxer Lua.
Extensions	2a. L'usuari introdueix un algorisme que no funciona correctament 2a1. El programa dona error al calcular l'Elo 2ab. La simulació no s'executa 3a. La funció no retorna la quantitat d'Elo a canviar de la manera que ho espera el programa. 3a1. El programa modifica l'Elo dels jugadors de forma errònia 3a2. El resultat de la simulació es veu afectat greument. 3b. La funció retorna el resultat amb un format diferent de l'esperat. 3b1. El programa dona error a l'hora de modificar l'Elo dels jugadors. 3b2. No s'executa la simulació 4a. L'usuari no desa el fitxer Lua. 4a1. El programa calcula la puntuació com estava definit prèviament.

6. Configurar variables de la simulació

Actor principal	Usuari
Precondició	El programa s'inicia correctament.
Disparador	L'usuari vol configurar les variables per iniciar la simulació.
Escenari principal d'èxit	1. L'usuari executa el programa. 2. L'usuari afegeix el número de jugadors amb el que es faran les simulacions. 3. L'usuari afegeix el número de jugadors per equip 4. L'usuari afegeix el número d'equips per partida. 5. L'usuari afegeix la diferència d'Elo entre els jugadors d'un equip i partida. 6. L'usuari afegeix el número de partides a jugar per simulació. 7. L'usuari introdueix la destinació del fitxer .csv.
Extensions	2a - 3a - 4a - 5a - 6a. L'usuari introdueix un número negatiu. El sistema avisa del número negatiu. El sistema permet tornar a introduir el número. 7a. L'usuari introdueix un directori erroni 7a1. El fitxer .csv no es perd.

7. Iniciar simulació

Actor principal	Usuari
Precondició	Les dades s'han introduït correctament.
Disparador	L'usuari vol iniciar la simulació.
Escenari principal d'èxit	1. L'usuari clica el botó de començar la simulació. 2. El programa executa la simulació
Extensions	2a. La simulació s'executa de forma errònia 2a1. La simulació informa de l'error. 2a2. No s'executa la simulació.

8. Obtenir les dades

Actor principal	Usuari
Precondició	La simulació s'executa de forma correcta.
Disparador	L'usuari obtenir les dades extretes de la simulació via un fitxer .csv.
Escenari principal d'èxit	1. El sistema crea el fitxer en el directori indicat. 2. L'usuari obra el fitxer.
Extensions	1a. El directori indicat no existeix. 2a1. El fitxer .csv es perd.

5.4 Diagrama de classes d'especificació

El diagrama de classes d'especificació descriu l'estructura d'un sistema on es mostren les seves classes amb els seus atributs i funcions, i també com es relacionen els objectes entre ells [23]. L'esquema d'aquest projecte és el que es pot observar a la figura 6.

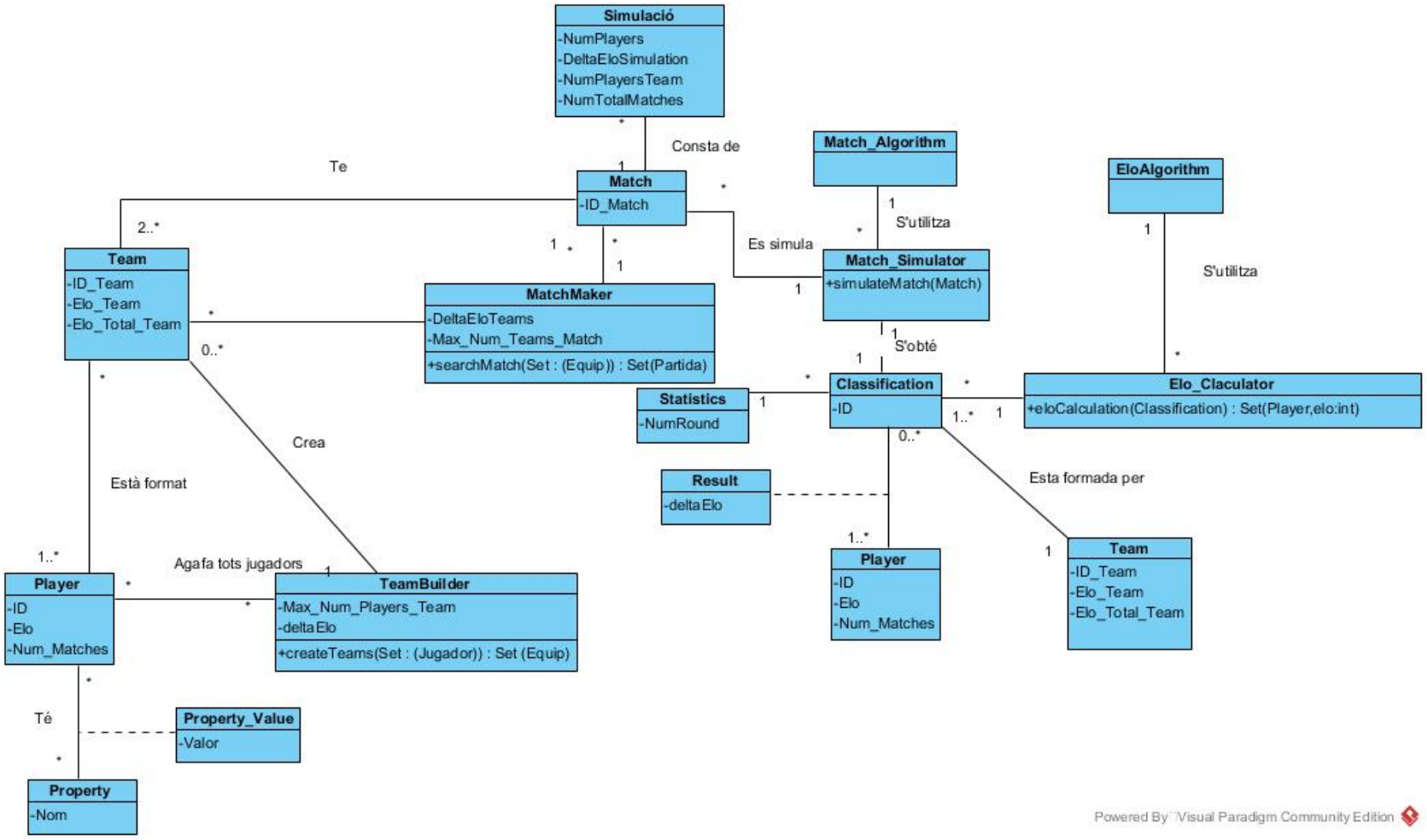

Figura 6: Diagrama de classes d'especificació del sistema. Font: Elaboració pròpia

5.4.1 Restriccions textuals

En aquest apartat s'explicaran les diverses restriccions textuals que s'han de complir perquè el diagrama de classes funcioni bé i tingui coherència.

1. PK: Player(ID), Team(ID), Property(Name), Match(ID), Classification(ID).
2. En un equip no poden haver-hi dos jugadors iguals.
3. Una partida no pot tindre dos equips idèntics.
4. Un jugador no pot jugar més d'una partida alhora.
5. La diferència d'Elo entre els jugadors d'un equip no pot ser més gran que l'atribut DeltaEloSimulation de la classe Simulation.
6. La diferència d'Elo entre els equips d'una partida no pot ser més gran que l'atribut DeltaEloSimulation de la classe Simulation.
7. L'Elo d'un equip i d'un jugador no pot ser més petit que 0.
8. Els atributs de la classe Simulation no poden ser negatius.
9. Els atributs Elo_Team i Elo_Total_Team es calculen a partir de l'Elo dels jugadors del mateix equip.

5.4.2 Descripció de les classes

Simulation

Una simulació és el conjunt de partides a simular amb els diversos jugadors. Consta del nombre de participants que hi haurà en total i de la diferència d'Elo màxima utilitzada en els diferents algorismes. També té el número de jugadors màxim per equip i en nombre total de partides a fer.

Player

Un jugador representa cada un dels participants que forma un equip i que participa en les partides. Aquest es caracteritza per: una ID, un nivell d'Elo i el número de partides que ha jugat.

Property

Una propietat és una característica que té cadascun dels jugadors de la simulació. S'identifica pel seu nom.

Property_Value

Valor de la propietat associada al jugador.

Team

Un equip és un conjunt de jugadors. Està compost per la seva ID, per l'Elo mitjà de l'equip i per l'Elo total d'aquest. Les dos últimes característiques depenen dels jugadors que el formen.

TeamBuilder

Algorisme per crear els diversos equips de la simulació. Té un màxim de jugadors per equip i la diferència d'Elo màxima entre els diferents participants d'aquest.

Match

Una partida és un enfrontament entre dos o més equips per veure quin és millor.

MatchMaker

Algorisme per emparellar els equips i crear una partida. Té un màxim d'equips per partida i la diferència d'Elo màxima entre els diferents participants d'aquest.

Match_Simulator

Simulació d'una partida, és a dir, de l'enfrontament entre dos o més equips.

Match_Algorithm

Algorisme que determina el resultat d'una partida.

Classification

Una classificació és el resultat ordenat de la partida. Informa quin equip ha quedat en primer lloc, quin en segon, etc.

Elo_Calculator

Càlcul de l'Elo de cada jugador després de determinar la classificació de la partida.

EloAlgorithm

Algorisme aplicat per calcular la quantitat d'Elo a modificar per a cada jugador.

Result

Quantitat d'Elo a modificar del jugador.

Statistics

Estadístiques de la simulació on es guarda el progrés de cada jugador.

6 Disseny

En aquest apartat s'explicarà l'arquitectura que s'ha utilitzat per fer el software. També s'indicarà els patrons de disseny utilitzats. Finalment, es mostrarà la interfície gràfica.

6.1 Arquitectura del sistema

Aquest projecte segueix una arquitectura de tres capes, que són la de presentació, la de domini i la de dades. Cada una d'elles es desenvolupa de manera independent. Així, es pot canviar o afegir noves funcionalitats en qualsevol de les tres capes sense que les altres dos es vegin afectades. Per fer-se una idea de com és l'arquitectura de 3 capes, a la figura 7 es pot veure un esquema. A continuació s'explicarà cada una d'elles:

- **Capa de presentació:** La capa de presentació, també coneguda com a interfície gràfica, s'encarrega d'interaccionar amb l'usuari. Ha de mostrar la informació i capturar les dades que introdueixi el client per passar-les a la capa de domini. En aquest projecte, és la part que està desenvolupada amb QT i on l'usuari introdueix els paràmetres per iniciar la simulació.
- **Capa de domini:** La capa de domini conté totes les funcionalitats per processar tota la informació. Rep les dades de la capa de presentació, les processa i retorna el resultat. També es comunica amb la capa de dades per obtenir o emmagatzemar informació. En aquest projecte, és la part que gestiona tota la simulació, és a dir, la creació d'equips i partides, les simulacions d'aquests i el càlcul de l'Elo resultant.
- **Capa de dades:** La capa de dades s'encarrega d'emmagatzemar la informació. És on es gestionaran les dades: es crearan, es modificaran o s'eliminaran. En aquest projecte, aquesta part és la que gestiona les estadístiques i els diferents jugadors de la simulació.

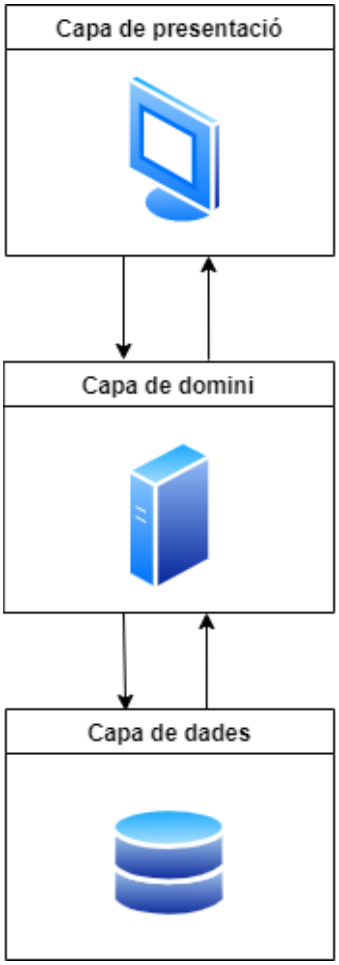

Figura 7: Arquitectura de 3 capes. Font: Elaboració pròpia

6.2 Diagrama de classes de disseny

El diagrama de classes de disseny representa les diverses classes de codi del programa. A la figura 8 es pot veure el d'EloSim. Si es compara amb el diagrama de classes d'especificació de la figura 6, aquest té petites variacions.

La primera és que la classe associativa Property_Value ha desaparegut i s'ha afegit dintre del propi jugador. La segona gran diferència són els objectes *Exporter* i ExcelExporter utilitzats per exportar les estadístiques del programa. La primera classe és abstracta i conté el path de la destinació final dels resultats. Tots els possibles mètodes d'exportació hereten d'ella. En canvi, ExcelExporter és una subclasse d'*Exporter* i s'encarrega d'exportar la informació a un fitxer amb format .csv.

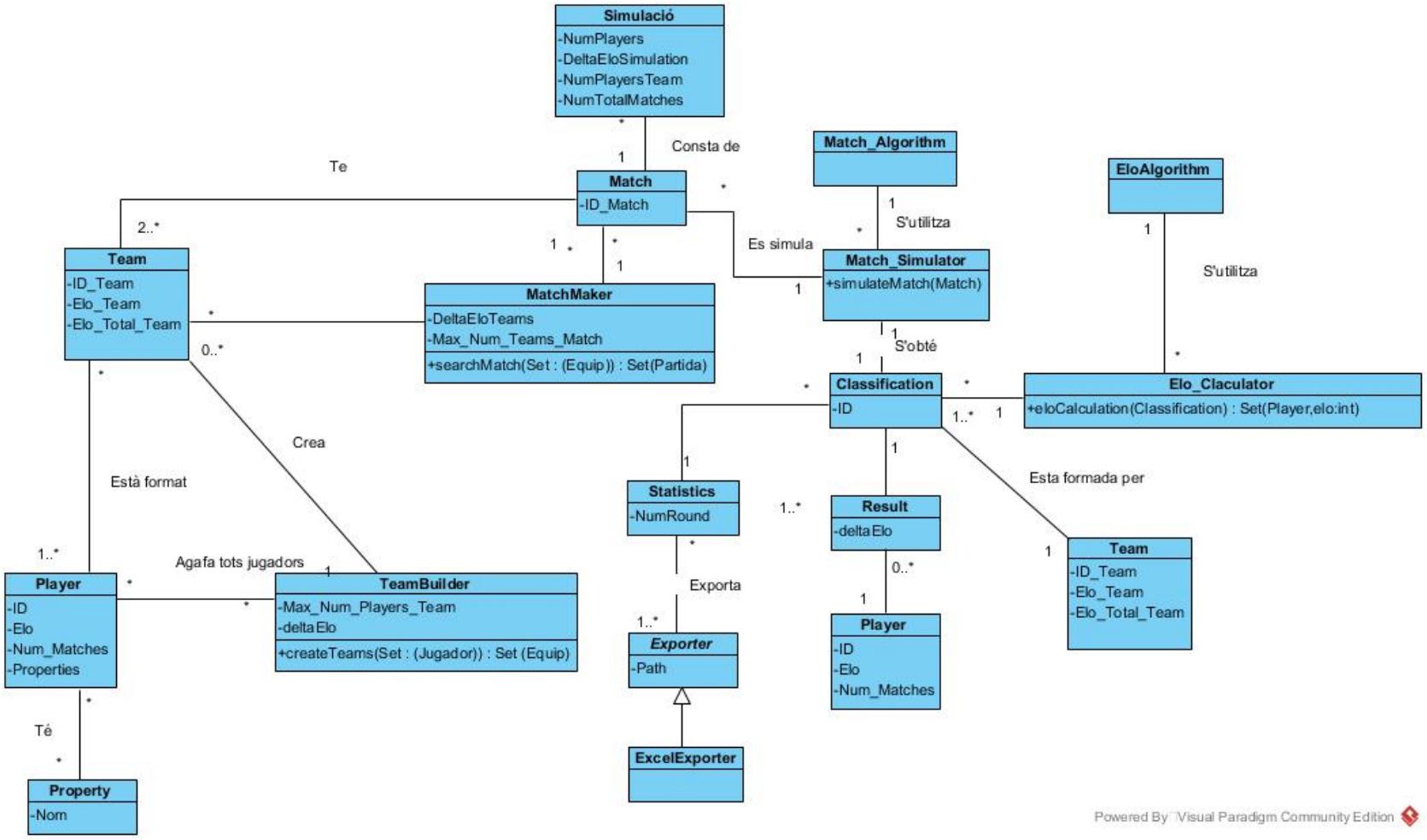

Figura 8: Diagrama de classes de disseny del sistema. Font: Elaboració pròpia

6.3 Patrons de disseny

Els patrons de disseny són tècniques per resoldre problemes comuns que normalment es troben a l'hora de programar. Durant el desenvolupament del software se n'han utilitzat alguns d'ells. A continuació s'explicaran quins s'han fet servir i on.

Patró adaptador

El patró adaptador serveix per adaptar la interfície d'una classe en un altre. S'utilitza quan es vol fer servir una classe existent que és incompatible amb la resta de codi.

Aquesta solució ha estat implementada en diverses ocasions per cridar les funcions del fitxer Lua. Per poder utilitzar els algorismes dels fitxers, se'ls ha de passar els diferents paràmetres necessaris. Però com no es pot utilitzar cap estructura del software implementat, les dades a passar s'han de convertir o adaptar en format de taules, una espècie d'arrays en Lua.

Patró mediador

El patró mediador redueix les comunicacions entre els objectes del sistema. En canvi de comunicar-se de forma directa, ho fan a través d'una classe mediadora. D'aquesta manera, es redueix la dependència de codi. S'utilitza quan hi ha un programa amb moltes classes i la majoria d'elles són dependents de les altres.

Aquesta solució s'ha utilitzat per gestionar la simulació sencera. La mateixa classe "Simulació"s'encarrega de comunicar les diferents dades rebudes a través dels altres objectes. És la que gestiona les funcions i fa de mediadora entre les diferents classes que componen la simulació.

6.4 Disseny de la interfície gràfica

La interfície gràfica és la part del programa que veu l'usuari i és amb la qual interacciona. En el cas d'EloSim, s'ha optat per dissenyar-ne una de la forma més senzilla i simple possible, però sempre sent intuïtiva i fàcil d'entendre. No obstant, s'ha decidit dedicar-li el mínim de temps possible perquè s'ha prioritzat desenvolupar els algorismes del sistema de la millor manera possible.

A la figura 9 es pot observar com es veu el software. Hi han sis graelles per omplir, una per cada un dels paràmetres que s'han d'introduir per executar la simulació. Cada una d'elles té referenciat a sobre quina dada representa. El botó que hi ha al costat dret de l'última graella, la tercera a la dreta començant per dalt, permet escollir el directori on desaran les estadístiques.

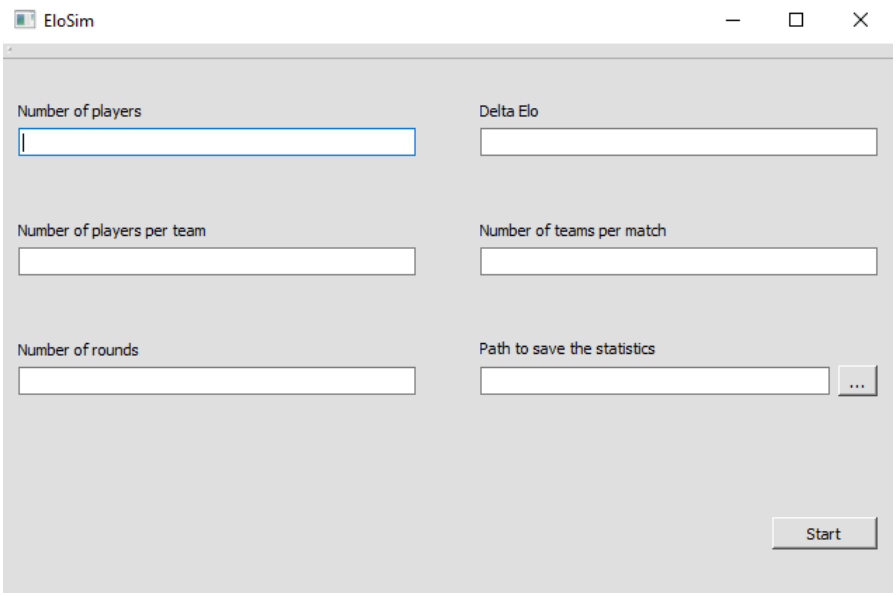

Figura 9: Disseny de la interfície gràfica del projecte. Font: Elaboració pròpia

7 Implementació

7.1 Tecnologies utilitzades

Per dur a terme el projecte d'EloSim s'han escollit un seguit d'eines i tecnologies que facilitaran el seu desenvolupament. A continuació s'explicaran cada una d'elles i el motiu de la seva elecció.

7.1.1 Llenguatges de programació

- **C++:** El projecte s'ha desenvolupat en C++ per tres motius. El primer de tots és que és el llenguatge que s'ha utilitzat més freqüentment i per tant, és el que es té més fresc, ja que es va utilitzar per a l'assignatura de Videojocs el quadrimestre anterior. El segon motiu és la gran facilitat que hi ha a Internet per trobar informació sobre com funcionen certs aspectes o mètodes del llenguatge, o per consultar dubtes o erros que puguin sorgir durant el desenvolupament. L'últim motiu és degut al fet de que és molt eficient. S'ha utilitzat la versió C++14.
- **Lua:** Lua és un llenguatge de programació potent, eficient i lleuger. El seu intèrpret està fet en llenguatge C. Per tant, pot ser integrat dins d'altres softwares o pot ser compilat en qualsevol d'ells, sempre que estiguin basats en C [24]. S'ha escollit Lua per dos motius. El primer és perquè és fàcil d'aprendre i d'utilitzar. El segon motiu és perquè hi ha la suficient documentació i resolució de problemes a Internet.
- **Latex:** Latex és un software lliure i és un sistema de composició de textos, normalment utilitzat per escriure documents i articles científics [25]. S'ha escollit per fer la memòria perquè és un llenguatge còmode per escriure una documentació llarga i clara, ja que permet estructurar el format de forma més senzilla.

7.1.2 Eines

- **QT:** QT és una eina que s'ha fet servir per desenvolupar la interfície gràfica del programa. Ha estat un software que s'ha empleat en alguna assignatura de la carrera i, per tant, ja es té cert coneixement de com funciona. És per aquest motiu que s'ha escollit.
- **Visual Studio 2019:** Visual Studio és una IDE compatible amb diversos llenguatges de programació com C++ o C#. S'ha escollit aquesta eina perquè ja s'ha utilitzat prèviament en la carrera.
- **Overleaf:** Overleaf és l'editor de text que s'ha utilitzat per escriure en Latex. S'ha escollit perquè ja s'ha treballat amb aquesta eina.
- **Git:** Git és un software de control de versions per manejar des de petits fins a grans projectes [26]. Aquesta eina ha estat útil per emmagatzemar tot el codi d'EloSim i poder anar fent diferents versions del propi treball.

- **Github [27]:** Github és un servei de repositoris que utilitza les versions Git. Ha estat emprat per guardar i compartir el codi i la documentació feta durant tot el treball. Té l'avantatge de què es poden accedir a versions anteriors del projecte i també gestionar els errors i bugs trobats.
- **Google Meet [28]:** Google Meet és un servei desenvolupat per l'equip de Google que serveix per fer videotrucades. Aquesta eina ha estat utilitzada per contactar amb el director i/o ponent. D'entre totes les eines que hi ha per fer videotrucades, s'ha escollit aquesta perquè és amb la que s'esta més familiaritzada. La UPC la proveeix i la recomana utilitzar, a més de ser la que s'ha fet servir durant les classes online de la carrera.
- **Trello [29]:** Trello és una eina per gestionar els projectes i les seves tasques en estil de tauler, com al Kanban. Ha estat utilitzada per apuntar les diverses funcions que s'habien de realitzar i així saber quines s'estaven fent, quines s'havien de fer i quines s'havien realitzat del tot. D'entre totes les eines que hi ha per fer un control de tasques, s'ha escollit aquesta perquè ja s'ha utilitzat prèviament en la carrera i és senzilla d'utilitzar.

7.2 Implementació dels algorismes

Durant el desenvolupament del projecte i de les diverses tasques, s'han hagut de desenvolupar diversos algorismes per assolir totes les funcionalitats. En algunes situacions, les implementacions es podrien haver fet de diverses maneres. A continuació es detallarà cada un dels algorismes i s'explicarà per què s'ha escollit.

- **Creació de jugadors:** Els jugadors són molt necessaris per dur a terme la simulació. Per poder-los generar hi han dues maneres: o són introduïts per l'usuari o són generats pel propi sistema. En el primer cas, han de ser afegits pel client mitjançant el fitxer de Lua. Això permet la possibilitat de crear els participants de la simulació amb dades extretes de diferents jugadors reals o de models extrets a partir d'una mineria de dades.

  En el segon cas, el sistema ha de generar, de forma automàtica, els diversos jugadors amb les propietats que els caracteritza. Per fer-ho, s'ha optat per utilitzar una distribució gaussiana [30]. D'aquesta manera, cada un d'ells tindrà valors més realistes i similars als demés.

  Per si es desconeix com funciona una distribució de Gauss, aquesta serveix per generar nombres aleatoris a partir d'una Normal, al voltant d'una mitjana i distribuïts amb la mateixa variabilitat. A la figura 10 es pot veure una campana de Gauss. La μ representa la mitjana i la σ l'arrel quadrada de la variància.

  Per exemple, si es vol crear un jugador que tingui una propietat anomenada "Armadura" i es vol donar-li un valor mitjançant la distribució gaussiana, es pot escollir una mitjana amb valor 10 i una variància amb valor 9. Llavors, hi ha un 68,2% de probabilitats que el número resultant estigui entre el 7 i el 13. Hi ha un 27,2% de que surti una xifra entre el 4 i el 7 o el 13 i el 16. Hi ha un 4,2% de que resulti un nombre entre l'1 i el 4 o el 16 i el 19. Finalment, hi ha un 0,2% de que surti un valor més petit que 1 o més gran que 19.

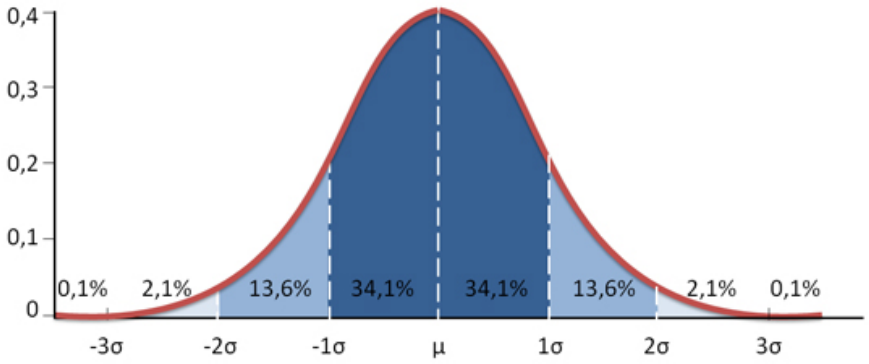

Figura 10: Campana de Gauss. Font: [31]

Per una altra banda, el valor que se li assigna a l'Elo és aleatori. D'aquesta manera, entre els diferents jugadors hi ha una gran diversitat de valors d'Elo. Això permet que les simulacions i els equips siguin més variats i així és contempli una millor evolució de l'Elo.

Tanmateix, tota aquesta generació de valors es podria haver fet de manera diferent. Per exemple, es podria haver assignat nombres aleatoris a cada una de les propietats dels jugadors. Això hagués provocat que hi hagués molta diferència entre ells i hagués fet més irregulars les partides. D'altra manera, els diferents participants podrien haver començat amb 0 d'Elo. Aquesta opció es va provar, però es va descartar perquè sempre es generaven els mateixos equips i no es contemplava bé el progrés de cada jugador.

- **Creació d'equips i partides:** La creació d'equips funciona ajuntant els jugadors d'un Elo similar. Per fer-ho, cada un comprova si amb el seu Elo entra en un dels equips que estan en formació. En cas negatiu, en crea un de nou. En canvi, si hi entra, es mira si el grup està ple, i si ho està, passa a ser un equip format. Per tant, pot passar que hi hagin jugadors que no formin part d'un de complet, provocant el fet de que no juguin. El mateix sistema s'aplica a la creació de partides. S'ha escollit aquest mètode per fer les simulacions de la forma més igualada possible, tenint tots els rivals un nivell d'Elo similar.

  Una solució alternativa seria agrupar els jugadors que no tenen un equip complet de forma aleatòria, creant així grups molt desequilibrats. Aquesta opció s'ha descartat perquè llavors les simulacions podrien ser desigualades i no es podria estudiar bé l'evolució de l'Elo.

  Una altra possible solució podria ser la que s'utilitza en els videojocs de veritat. Aquesta consisteix en esperar a que hi hagi una certa quantitat de jugadors i agrupar-los de la millor manera possible. A diferència del que succeeix en el projecte, en el cas real es desconeix la quantitat de participants que hi haurà. Per tant, s'ha de treballar sobre els usuaris que van arribant.

- **Simulació de partides:** Simular una partida es pot fer de diverses maneres, des de decidir el resultat de forma aleatòria fins a establir-lo a partir de les propietats dels jugadors. Per tal que la simulació s'adapti més al que vol l'usuari, s'ha optat per permetre escriure la fórmula al fitxer Lua. D'aquesta manera, el resultat s'ajusta a la idea del client i no és forçat pel sistema.

Actualment, l'algorisme introduït per determinar la victòria té en compte les propietats dels jugadors. Per decidir el resultat, es compara la força d'un equip amb l'armadura del rival, i es determina el dany fet. L'equip que n'hagi fet més, guanya. D'aquesta manera, la simulació és més realista i podria arribar a simular les mecàniques d'un videojoc de veritat.

- **Càlcul de l'Elo:** Després del resultat de cada partida s'ha de calcular l'Elo que guanya o perd cada jugador. El càlcul pot ser molt variat, ja que pot dependre de diversos factors. Per aquest motiu, se li ha donat l'opció a l'usuari de que introdueixi la fórmula en el fitxer de Lua. D'aquesta manera, l'Elo resultant seguirà la metodologia del client i no una imposada pel sistema.

  A dia d'avui, al fitxer de Lua hi ha un algorisme molt similar al dels escacs. Es calcula la possibilitat de guanyar de cada equip i es multiplica pel factor K, el màxim de punts que un jugador pot obtenir o perdre en una partida. Per saber calcular la possibilitat de victòria d'un equip, s'ha optat per implementar la fórmula de l'Elo de l'Overwatch [32], un videojoc competitiu.

7.3 Estructura del sistema

EloSim consta de diversos algorismes distribuïts entre el propi programa i el fitxer Lua. Per poder dur a terme la simulació de forma correcta, s'ha de seguir un ordre determinat. A continuació es mostrarà un diagrama de l'estructura del programa i els passos que segueix internament.

A la figura 11 es pot veure com estan distribuïts els algorismes. En el fitxer Lua es troben la creació de jugadors, el càlcul de l'Elo i la simulació d'una partida. EloSim s'encarrega de llegir-lo i també de crear els equips i les partides. Finalment, exporta les estadístiques a un format .csv.

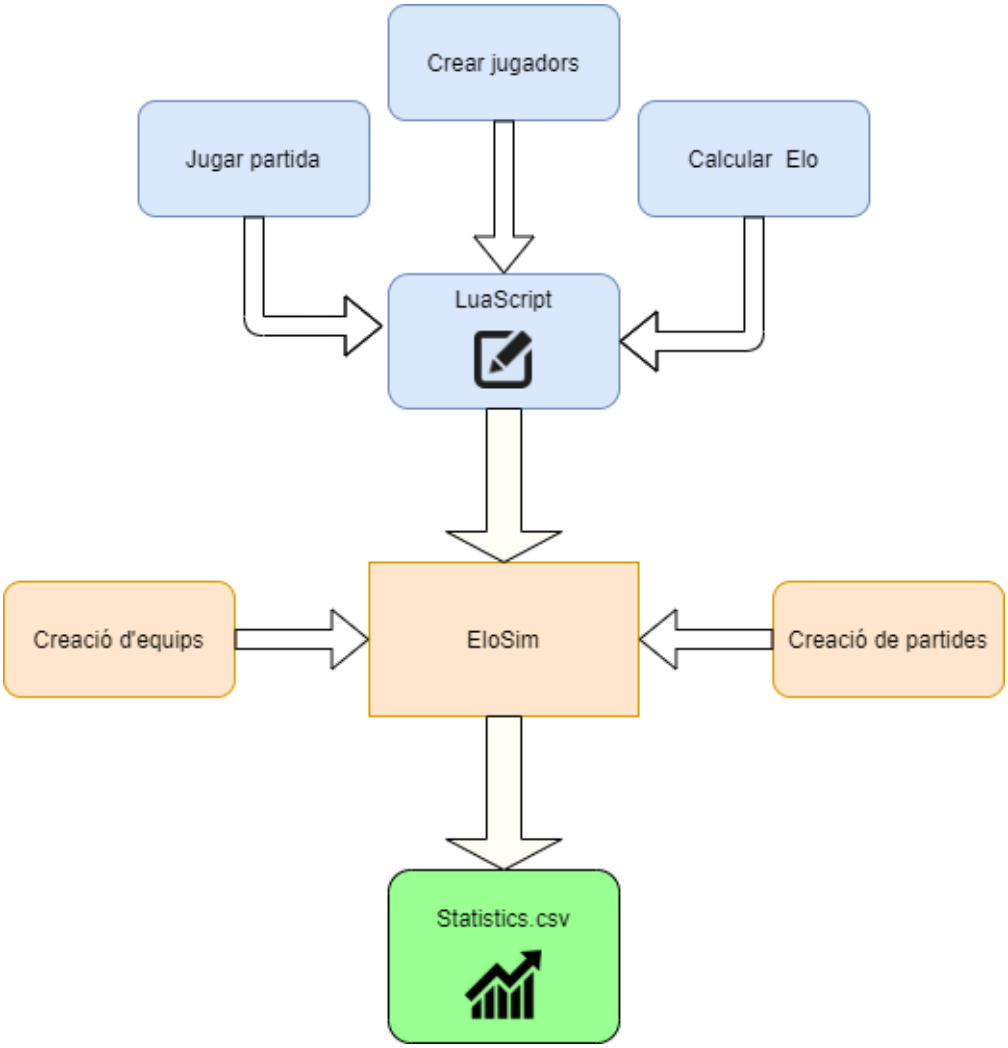

Figura 11: Estructura del sistema. Font: Elaboració pròpia

A la figura 12 es pot veure una fracció del codi de la classe Simulation. Concretament, es veu els passos que segueix el programa per fer una simulació. Primer de tot crea els jugadors. Després, per cada una de les rondes que s'han de jugar, es creen els equips. Tot seguit es formen les diverses partides. A continuació, es simula cada una d'elles, i amb el resultat obtingut, es calcula l'Elo resultant de cada jugador. Després se'ls hi aplica. Per acabar, s'actualitzen les estadístiques. Un com acabada la simulació, aquestes són exportades.

```
vector<shared_ptr<Player>> p1 = m_playersDB->getPlayers();
for (int j = 0; j < m_numTotalMatches; j++) {
    vector<shared_ptr<Team>> teams = m_teamBuilder->createTeams(p1);
    vector<shared_ptr<Match>> matches = m_matchMaker->searchMatch(teams);
    for (int i = 0; i < matches.size(); i++) {
        shared_ptr<Classification> classification = m_matchSimulator->simulateMatch(matches[i]);
        map<int, EloScore> deltaEloTeam = m_eloCalculator->calculateElo(classification);
        m_result->changeEloPlayers(deltaEloTeam);
    }
    m_statistics->updateStatistics();
}
lua_close(m_L);
return m_statistics->getStatistics();
```

Figura 12: Codi de la classe Simulation. Font: Elaboració pròpia

7.4 Hipòtesis simplificades

Per poder dur a terme aquest treball i desenvolupar els diferents algorismes, s'han hagut de simplificar situacions que són més complexes a la vida real.

- **Creació de jugadors:** Tots els jugadors són creats al principi de la simulació i tenen les mateixes propietats. Això pot variar de la realitat, on una persona pot registrar-se i començar a jugar a un joc mesos més tard del seu llançament.
- **Marchmaking:** La creació d'equips també s'ha vist simplificada. Des d'un bon principi es té coneixement de tots els jugadors que hi ha a la simulació. Addicionalment, també pot donar-se el cas on hi hagi participants que no juguin una partida perquè el seu equip no està complet. En la vida real, la cosa és ben diferent. Mai se sap quina quantitat d'usuaris buscaran partida en el mateix moment. Per tant l'emparellament s'ha d'adaptar i procurar fer-lo de la manera més ajustada possible. A més, un jugador no pot quedar-se sense formar part d'un equip, ja que sinó pot estar descontent i l'empresa es pot veure perjudicada. La mateixa situació s'aplica a les partides.
- **Simulació de partides:** En el món real, els jugadors d'una partida poden marxar d'aquesta, ja sigui per problemes de connexió a Internet o perquè els hi és inviable seguir jugant. Aquesta acció normalment es veu sancionada. Addicionalment, afecta el resultat de l'enfrontament, ja que un equip té un membre menys. Per tal de simplificar-ho en la simulació, una partida sempre es jugarà amb el màxim de jugadors possibles, i cap d'ells podrà abandonar-la.

8 Mètodes d'avauluació

Durant la implantació de les diverses funcionalitats que ha de tindre un sistema, sempre s'han d'avaluar i testejar cadascuna d'elles per veure si el seu funcionament és el correcte. Per fer-ho, normalment s'utilitzen tests unitaris, és a dir, fer proves per funcionalitats específiques per veure si el resultat és l'esperat o no.

Per una altra banda, les funcionalitats d'aquest projecte són independents les unes de les altres, és a dir, el funcionament incorrecte d'un algorisme no implica que el dels altres funcioni malament. Per exemple, es pot donar el cas on l'algorisme de partida retorni una classificació errònia. En comptes de posar en primer lloc a l'equip A, pot ficar a l'equip B. Tot seguit, un cop s'hagi de calcular el delta d'Elo resultant de la partida, l'algorisme que s'encarregui de fer-ho realitzarà els càlculs correctament amb les dades que li han passat. L'única diferència serà el resultat, que no serà l'esperat a causa de l'error de la classificació.

Per validar el funcionament correcte de les diverses funcionalitats de l'EloSim s'han realitzat dos mètodes. El primer es va utilitzar al principi de la fase de desenvolupament, i consistia a calcular els diversos resultats a mà. En canvi, el segon és el que s'ha utilitzat més de cara al final treball i consisteix en fer ús dels tests unitaris.

En el primer cas, normalment sempre s'executava el programa amb els mateixos paràmetres (mateix número de jugadors, de deltaElo, de número de partides, etc). A més, s'utilitzaven nombres petits per fer més senzills els càlculs. En cas de que coincidissin, es donava la implementació de la funcionalitat per passada.

Seguint amb el mateix mètode i sabent que certes funcionalitats funcionaven correctament, es va agafar el resultat del fitxer .csv com a referència. Si la sortida de les dades era igual a la de les anteriors proves, significava que la implementació era la correcta.

Òbviament, aquest és un mètode poc pràctic i fiable. Arran de la seva poca eficiència, es va decidir d'utilitzar el segon. Aquest consisteix a fer ús dels tests unitaris per validar tant les funcionalitats antigues com les noves. Per fer-ho, s'ha decidit utilitzar la llibreria Google Test.

De totes les funcionalitats que té el programa, s'han testejat les quatre principals. S'han fet diversos tests, on es contemplen diferents casos. La primera ha estat la creació d'equips. S'han fet proves canviant el número de jugadors disponibles o el nombre màxim d'integrants. Per exemple, s'ha testejat que passa si hi han 0 jugadors per emparellar o si els equips són d'un nombre imparell de participants.

La segona funcionalitat ha estat la creació de partides. S'han fet tests on es comprova el correcte funcionament si es varia el nombre d'equips per partida o la quantitat de grups disponibles. Per exemple, s'ha provat a fer partides de tres equips o amb un Elo molt variat.

La tercera ha estat la simulació dels emparellaments. S'han provat diverses situacions on es canviaven la quantitat d'equips per partida, el seu número d'integrants i les seves propietats. Per exemple, s'han simulat partides de tres equips o amb jugadors amb propietats que tenen valors negatius.

L'última funcionalitat ha estat el càlcul de l'Elo. S'han testejat diversos escenaris canviant la diferència d'Elo i el resultat. Per exemple, s'ha comprovat l'empat o el càlcul resultant de dos jugadors de rangs molt diferents.

8.1 Proves d'Elo

El sistema d'Elo és molt important en un videojoc competitiu. La diferència d'entre un de bo i un de dolent pot afectar molt als jugadors. Per demostrar-ho, s'ha fet la mateixa simulació amb una bona fórmula i amb una de dolenta i s'han agafat dos exemples interessants.

A la figura 13 es pot veure l'evolució de l'Elo d'un jugador amb un bon sistema de puntuació. En aquest cas, és una línia creixent. D'aquí es pot deduir el jugador té bones habilitats, però ha començat en un Elo molt baix que no li correspon. A mesura que juga i guanya partides, el seu rang augmenta fins a arribar a una puntuació de 1200. Això indica que el sistema és bo, ja que el seu progrés augmenta conforme a les seves habilitats.

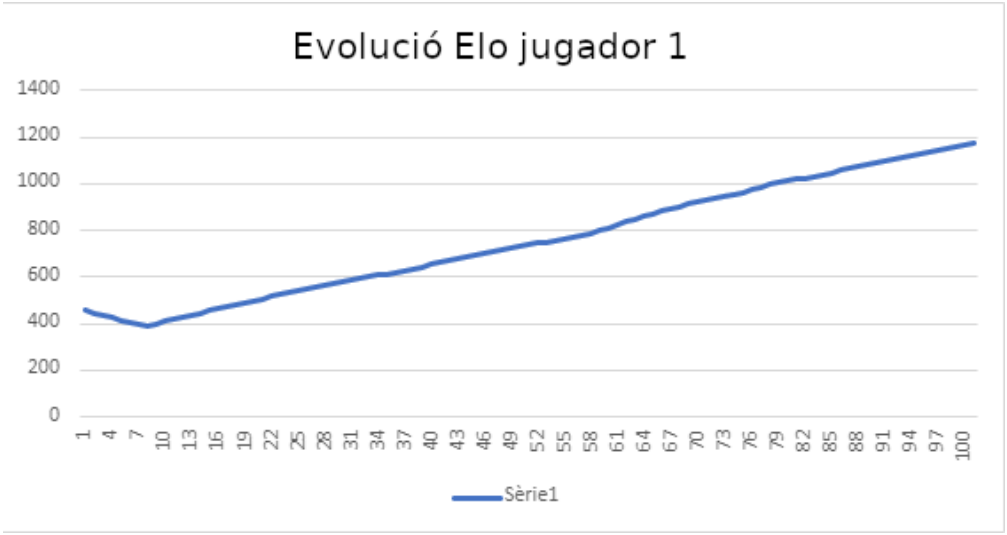

Figura 13: Evolució del jugador 1 amb un bon sistema d'Elo. Elaboració pròpia

En canvi, a la figura 14 es pot observar el resultat d'aplicar una mala fórmula d'Elo. La mateixa simulació s'ha fet sobre el mateix jugador, el qual tenia bones habilitats. Però com es pot observar, el seu rang és estable. Encara que guanyi moltes més partides de les que perd, la seva puntuació no augmenta. Per tant, en aquest cas, la fórmula de l'Elo no associa el progrés del jugador amb les seves habilitats.

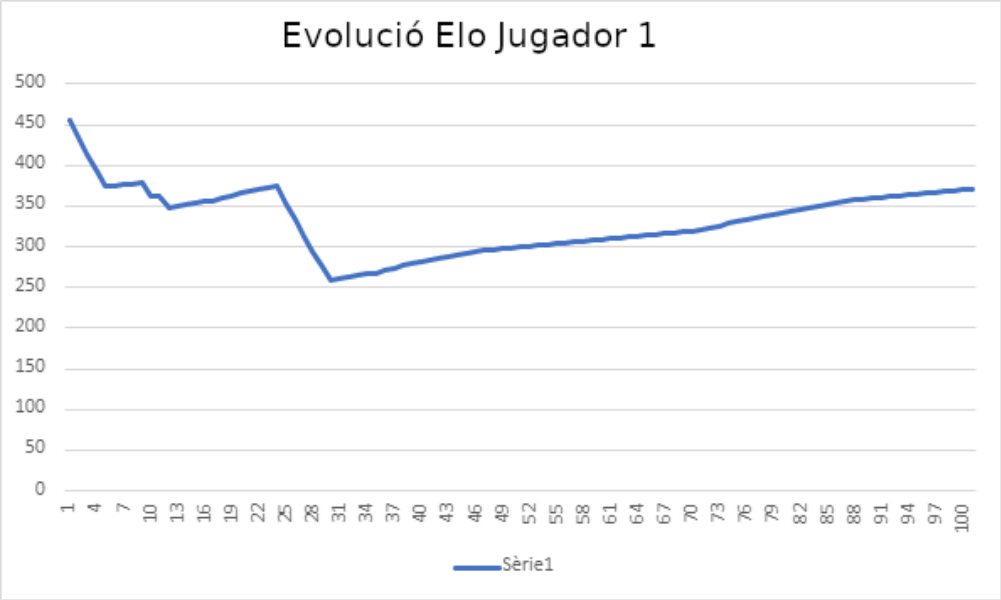

Figura 14: Evolució del jugador 1 amb un dolent sistema d'Elo. Elaboració pròpia

9 Planificació temporal definitiva

Planificar bé un projecte és molt important, ja que et permet gestionar el temps per desenvolupar tot el producte i preveure tots els errors que puguin sorgir. Concretament, aquest TFG té una duració aproximada de quatre mesos i mig, començant a mitjans de febrer, per ser exactes el dia 15, i finalitzant l'última setmana de juny. La duració del treball ha estat d'unes 450 hores, que s'han distribuit entre les 18 setmanes lectives (excloent Setmana Santa). Això equival a 25 hores per setmana, que fa una mitjana de 5 hores per cada dia lectiu.

9.1 Descripció de tasques

Aquest projecte es pot dividir en tres grans parts, on cada una d'elles consta de diverses tasques: gestió del projecte, desenvolupament i documentació i seguiment.

9.1.1 Gestió del projecte

- **GdP1 - Contextualització i abast:** Es farà un document on es definirà la idea del treball, on s'identificaran els *stakeholders*, s'analitzaran els programes similars, s'estudiaran els requisits i inconvenients que poden haver-hi i es descriurà la metodologia utilitzada. Tindrà una duració de 25 hores i serà necessari un ordinador amb Internet i l'Overleaf.
- **GdP2 - Planificació temporal:** S'escriurà un document on s'identificaran, es descriuran i s'organitzaran les diferents tasques que componen el projecte. També inclourà un diagrama de Gantt i la gestió dels diferents riscos que poden sorgir. Tindrà una duració de 10 hores i es necessitarà un ordinador amb connexió a Internet, l'Overleaf i el GanttProject. La tasca dependrà de GdP1.
- **GdP3 - Pressupost i sostenibilitat:** Es redactarà un document on es farà un pla econòmic del projecte i es farà un informe de sostenibilitat. Tindrà una durada de 15 hores i serà necessari un ordinador amb Internet, l'Overleaf i l'Excel. La tasca dependrà de GdP2.
- **GdP4 - Integració del document final:** Document on s'unificaran les tres tasques anteriorment explicades amb les correccions necessàries en cada una d'elles. S'estima una duració de 20h i serà necessari un ordinador amb connexió a Internet i l'Overleaf. La tasca dependrà de GdP3.

9.1.2 Desenvolupament

Dins del desenvolupament, les tasques es poden dividir en tres grans grups, depenent de quin objectiu principal formen part: personalització de l'algorisme del sistema d'Elo, simulació de partides o comparar resultats entre diferents variables i algorismes.

9.1.2.1 Personalització de l'algorisme del sistema d'Elo

- **PA1 - Preparació pel desenvolupament:** S'instal·laran tots els programes i IDEs necessaris per a poder desenvolupar el projecte. També es crearà el repositori de Github i es dissenyarà l'estructura del programa. Es necessitarà un ordinador amb connexió a Internet, l'IDE anomenat Visual Studio (VS), Github i el Visual Paradigm. Durarà 10 hores.
- **PA2 - Creació de l'estructura del programa:** Es crearà tota l'estructura del programa de la forma més senzilla, és a dir aplicant números aleatoris on sigui necessari i sense cap algorisme. També s'exportaran les dades de la forma més senzilla possible. Tindrà una durada 50 hores i serà necessari un ordinador amb connexió a Internet, Github i el Visual Studio. La tasca dependrà de PA1.
- **PA3 - Disseny d'algorisme d'Elo clàssic:** Es dissenyarà un algorisme d'Elo clàssic en un script de Lua per poder realitzar posteriorment l'execució del programa. Tindrà una durada 5 hores i serà necessari un ordinador amb Internet, Github, el Visual Studio i Lua. La tasca dependrà de PA2.
- **PA4 - Introducció i lectura de Fitxers Lua:** Es permetrà carregar un fitxer de script en Lua que contingui l'algorisme de l'Elo i s'haurà de poder executar. Tindrà una durada 15 hores i es necessitarà un ordinador amb Internet, el Github, el Visual Studio i Lua. La tasca dependrà de PA3.
- **PA5 - Interpretació de l'algorisme d'Elo:** S'haurà d'interpretar l'algorisme d'Elo introduït mitjançant el script en Lua per així saber quines variables s'utilitzen i quines s'han de tenir en compte. Tindrà una durada 35 hores i serà necessari un ordinador amb connexió a Internet, Github i el Visual Studio. La tasca dependrà de PA4.

9.1.2.2 Simulació de partides

- **SP1 - Creació de jugadors:** S'hauran de crear els diversos jugadors que participaran en la simulació. Per cada un s'hauran d'afegir les diverses habilitats que poden afectar al resultat de la partida i al càlcul de l'Elo. Tindrà una durada de 30 hores i requerirà un ordinador amb Internet, Github i el Visual Studio. La tasca dependrà de PA5.
- **SP2 - Simulació de partida:** Es simularà una partida entre dos equips. Es tindran en compte tant l'Elo com les habilitats de cada un per decidir el resultat. Tindrà una durada 25 hores i serà necessari un ordinador amb Internet, Github i el Visual Studio. La tasca dependrà de SP1.
- **SP3 - Càlcul de l'Elo resultant:** Es calcularà i modificarà l'Elo de cada un dels jugadors depenent del resultat de la partida i de les estadístiques de cada un d'ells. Per fer-ho, s'utilitzarà l'algorisme introduït en el script de Lua. Tindrà una durada de 15 hores i es necessitarà un ordinador amb Internet, Github i el Visual Studio . La tasca dependrà de SP2.
- **SP4 - Disseny de l'algorisme de creació d'equips:** Es dissenyarà l'algorisme per aparellar els diferents jugadors en equips d'un Elo similar. Tindrà una durada de 15 hores i serà necessari un ordinador amb Internet, Github i el Visual Studio. La tasca dependrà de SP1.

- **SP5 - Disseny de l'algorisme de preparació de partides:** Es dissenyarà un algorisme que seleccionarà dos equips amb un nivell d'Elo similar per a que simulin una partida. S'empraran 5 hores per fer la tasca i requerirà un ordinador amb Internet, Github i el Visual Studio. Dependrà de SP4.
- **SP6 - Creació d'una simulació:** S'haurà de crear una simulació que permeti simular les diverses partides a realitzar. Es podran modificar elements com el número de jugadors, el número de partides que s'han de fer o la diferència màxima d'Elo entre un equip i un altre. Tindrà una durada de 10 hores i es requerirà d'un ordinador amb connexió a Internet, de Github i del Visual Studio. La tasca dependrà de SP5.

9.1.2.3 Comparar resultats entre diferents variables i algorismes

- **CR1 - Creació d'estructura de dades:** S'haurà de crear l'estructura per emmagatzemar les diferents dades de les partides i dels jugadors, per poder-les analitzar i utilitzar després. Tindrà una durada de 10 hores i serà necessari un ordinador amb connexió a Internet, Github i el Visual Studio. La tasca dependrà de SP6.
- **CR2 - Disseny de la interfície gràfica:** S'haurà de dissenyar una interfície gràfica simple i senzilla per facilitar a l'usuari l'ús del programa. Tindrà una durada de 40 hores i requerirà d'un ordinador amb connexió a Internet, de Github i del Visual Studio. La tasca dependrà de CR1.
- **CR3 - Exportació de dades a l'Excel:** S'exportaran les dades emmagatzemades en l'estructura de dades en un fitxer amb format Excel, per així poder-les analitzar amb millor precisió. Les dades s'hauran de mostrar d'una forma entenedora i còmode per l'usuari. Tindrà una durada de 5 hores i es requerirà d'un ordinador amb Internet, de Github i del Visual Studio. La tasca dependrà de CR2.

9.1.3 Documentació i seguiment

- **DS1 - Documentar la memòria:** Es redactarà la memòria de tot el desenvolupament del projecte. La tasca durarà 75 hores i serà necessari un ordinador amb Internet i l'Overleaf.
- **DS2 - Reunió setmanal:** Es realitzarà una reunió setmanal amb el director del projecte, Manel Rello, per parlar de l'evolució d'aquest i dels problemes o dubtes que puguin sorgir. Posteriorment s'afegirà a la memòria els detalls de cada reunió. En total, s'estima una durada de 25 hores i es requerirà un ordinador amb Internet, l'Overleaf i el Google Meet.
- **DS3 - Preparació de la presentació final:** Es prepararà l'exposició per a la presentació final del projecte que es farà davant un tribunal. Tindrà una durada de 10 hores i requerirà d'un ordinador amb connexió a Internet i d'un programa per fer diapositives. La tasca dependrà de DS1.

9.2 Recursos humans

S'entén com a recursos humans a totes aquelles persones que han d'ocupar un càrrec en el projecte per poder-lo dur a terme. Per fer-ho, cada una tindrà un rol i treballarà més o menys depenent de les tasques descrites prèviament en el diagrama de Gantt. A continuació es nomenaran cada un d'ells i s'explicarà la seva funció:

- **Cap de projecte (CP):** El cap de projecte s'encarrega de gestionar el projecte. Gestiona els equips, planifica les tasques, calcula els pressupostos i els recursos necessaris i gestiona els riscos. També s'encarrega d'escriure la documentació.
- **Analista programador (AP):** L'analista programador té la funció d'identificar els requisits del sistema, de garantir la qualitat del programa i de dissenyar l'estructura del programa.
- **Arquitecte del Software (AS):** L'arquitecte del Software és la persona que escull quina tecnologia s'utilitzarà i realitza processos d'avaluació contínuament per veure que es compleixen els requisits.
- **Dissenyador d'UI (DUI):** El dissenyador d'interfícies gràfiques és qui s'encarrega de dissenyar l'apartat visual del programa per a que sigui simple d'utilitzar i clar.
- **Programador (P):** El programador s'encarrega de programar el programa per a que funcioni.
- **Tester (T):** El tester té el paper de dissenyar diferents jocs de prova per comprovar el funcionament correcte del programa i detectar els errors.

9.3 Canvis respecte la planificació inicial

Durant l'assignatura de GEP es fa una planificació completa del TFG per tal d'assolir tots els objectius. Aquesta pot no ser la versió definitiva, ja que pot arribar a ser modificada degut diversos errors o inconvenients que no han estat previstos. En el cas d'EloSim, també s'han hagut de fer canvis, però no pels motius mencionats. La gran majoria de tasques s'han seguit tal com es van organitzar al principi, però hi han hagut unes petites variacions. Bàsicament, s'ha afegit una tasca que consisteix a implementar totes les funcionalitats del sistema, però de forma molt bàsica, és a dir, sense llegir fitxers d'entrada i aplicant nombres aleatoris quan es requereixi d'un algorisme. En altres paraules, consisteix a fer tota l'estructura del programa. Aquesta decisió s'ha pres perquè d'aquesta manera, si hi ha algun inconvenient durant el desenvolupament, es tingui un producte funcional el dia de l'entrega. Igualment, aquesta tasca no ha afectat de forma negativa a les altres. Tot i haver-ne endarrerit la gran majoria, també ha escurçat la durada de cada una, ja que l'estructura ja està feta. D'aquesta manera, el projecte ha seguit amb les mateixes dates de finalització que en la planificació inicial.

10 Estimacions i Gantt

10.1 Estimacions

ID	Tasca	Durada (hores)	Dependències	RR Materials	RR Humans
GdP	**Gestió de projecte**	**70**			
GdP1	Contextualització i abast	25		PC, Overleaf	CP
GdP2	Planificació temporal	10	GdP1	PC, Overleaf, Gantt-Project	CP
GdP3	Pressupost i sostenibilitat	15	GdP2	PC, Overleaf, Excel	CP
GdP4	Integració del document final	20	GdP3	PC, Overleaf	CP
D	**Desenvolupament**	**270**			
PA1	Preparació pel desenvolupament	10		PC, Github, VS, Visual Paradigm	AP, AS, P, T
PA2	Creació de l'estructura del programa	50	PA1	PC, Github, VS	AP, AS, P, T
PA3	Disseny d'algorisme d'Elo clàssic	5	PA2	PC, Github, VS, Lua	AP, AS, P, T
PA4	Introducció i lectura de Fitxers Lua	15	PA3	PC, Github, VS, Lua	AP, AS, P, T
PA5	Interpretació de l'algorisme d'Elo	35	PA4	PC, Github, VS	AP, AS, P, T
SP1	Creació de jugadors	30	PA5	PC, Github, VS	AP, AS, P, T
SP2	Simulació de partida	25	SP1	PC, Github, VS	AP, AS, P, T
SP3	Càlcul de l'Elo resultant	15	SP2	PC, Github, VS	AP, AS, P, T
SP4	Disseny de l'algorisme de creació d'equips	15	SP1	PC, Github, VS	AP, AS, P, T
SP5	Disseny de l'algorisme de preparació de partides	5	SP4	PC, Github, VS	AP, AS, P, T
SP6	Creació d'una simulació	10	SP5	PC, Github, VS	AP, AS, P, T
CR1	Creació de l'estructura de dades	10	SP6	PC, Guthub, VS	AP, AS, P, T
CR2	Disseny de la interfície gràfica	40	CR1	PC, Github, VS, Qt	AP, AS, DUI, P, T
CR3	Exportació de dades a l'Excel	5	CR2	PC, Github, VS	AP, AS, P, T
DS	**Documentació i seguiment**	**110**			
DS1	Documentació de la memòria	75		PC, Overleaf	CP
DS2	Reunió setmanal	25		PC, Overleaf, Google Meet	CP
DS3	Preparació de la presentació final	10	DS1	PC, Power Point	CP
Total		**450**			

Taula 2: Taula d'estimacions de les diverses tasques a realitzar. Font: Elaboració pròpia

A la taula 2 es pot veure una estimació de cada una de les tasques descrites anteriorment, amb la seva duració, les seves dependències i els recursos necessaris per cada una d'elles. Com es pot observar, la suma total d'hores recau en 450, que son les que s'havien especificat prèviament.

10.2 Diagrama de Gantt

A la figura 15 es pot observar la part textual del diagrama de Gantt del nostre projecte. En aquest es mostren les diverses tasques amb la seva durada, que és en dies, amb les dependències, la probabilitat de risc i els principals recursos humans necessaris.

A la figura 16 es pot contemplar la part gràfica del diagrama de Gantt. Cada color representa un conjunt de les tasques anomenades anteriorment, sent el taronja per les de gestió del projecte, el verd per les de documentació i comunicació i la resta per les de desenvolupament. Dintre d'aquest últim, el color vermell representa les tasques de la personalització de l'algorisme del sistema d'Elo, el blau les de la simulació de partides i el groc les de comparar els resultats entre les diferents variables i algorismes. Si un color és més fosc que la resta, vol dir que aquella tasca té un risc alt.

Nom	Durada	Inici	Finalització	Dependències	Risc	Rols
Gestió del projecte	20	23/2/21	22/3/21			
GdP1	6	23/2/21	2/3/21		Baix	CP
GdP2	4	3/3/21	8/3/21	GdP1	Baix	CP
GdP3	5	9/3/21	15/3/21	GdP2	Baix	CP
GdP4	5	16/3/21	22/3/21	GdP3	Baix	CP
Desenvolupament	85	15/2/21	11/6/21			
Personalització de l'algorisme del sistema d'Elo	52	15/2/21	27/4/21			
PA1	26	15/2/21	22/3/21		Baix	AP, AS, P, T
PA2	15	23/3/21	12/4/21	PA1	Alt	AP, AS, P, T
PA3	1	13/4/21	13/4/21	PA2	Alt	AP, AS, P, T
PA4	3	14/4/21	16/4/21	PA3	Alt	AP, AS, P, T
PA5	7	19/4/21	27/4/21	PA4	Alt	AP, AS, P, T
Simulació de partides	22	28/4/21	27/5/21			
SP1	6	28/4/21	5/5/21	PA5	Baix	AP, AS, P, T
SP2	5	6/5/21	12/5/21	SP1	Alt	AP, AS, P, T
SP3	3	13/5/21	17/5/21	SP2	Baix	AP, AS, P, T
SP4	3	20/5/21	24/5/21	SP1	Mitjà	AP, AS, P, T
SP5	1	25/5/21	25/5/21	SP4	Baix	AP, AS, P, T
SP6	2	26/5/21	27/5/21	SP5	Alt	AP, AS, P, T
Comparar resultats	11	28/5/21	11/6/21			
CR1	2	28/5/21	31/5/21	SP6	Baix	AP, AS, P, T
CR2	8	1/6/21	10/6/21	CR1	Alt	AP, AS, DUI, P, T
CR3	1	11/6/21	11/6/21	CR2	Baix	AP, AS, P, T
Documentació i seguiment	95	15/2/21	25/6/21			
DS1	93	15/2/21	23/6/21		Baix	CP
DS2	93	15/2/21	23/6/21		Baix	CP
DS3	2	24/6/21	25/6/21	DS1	Baix	Cap de projecte

Figura 15: Diagrama de Gantt part textual. Font: Elaboració pròpia

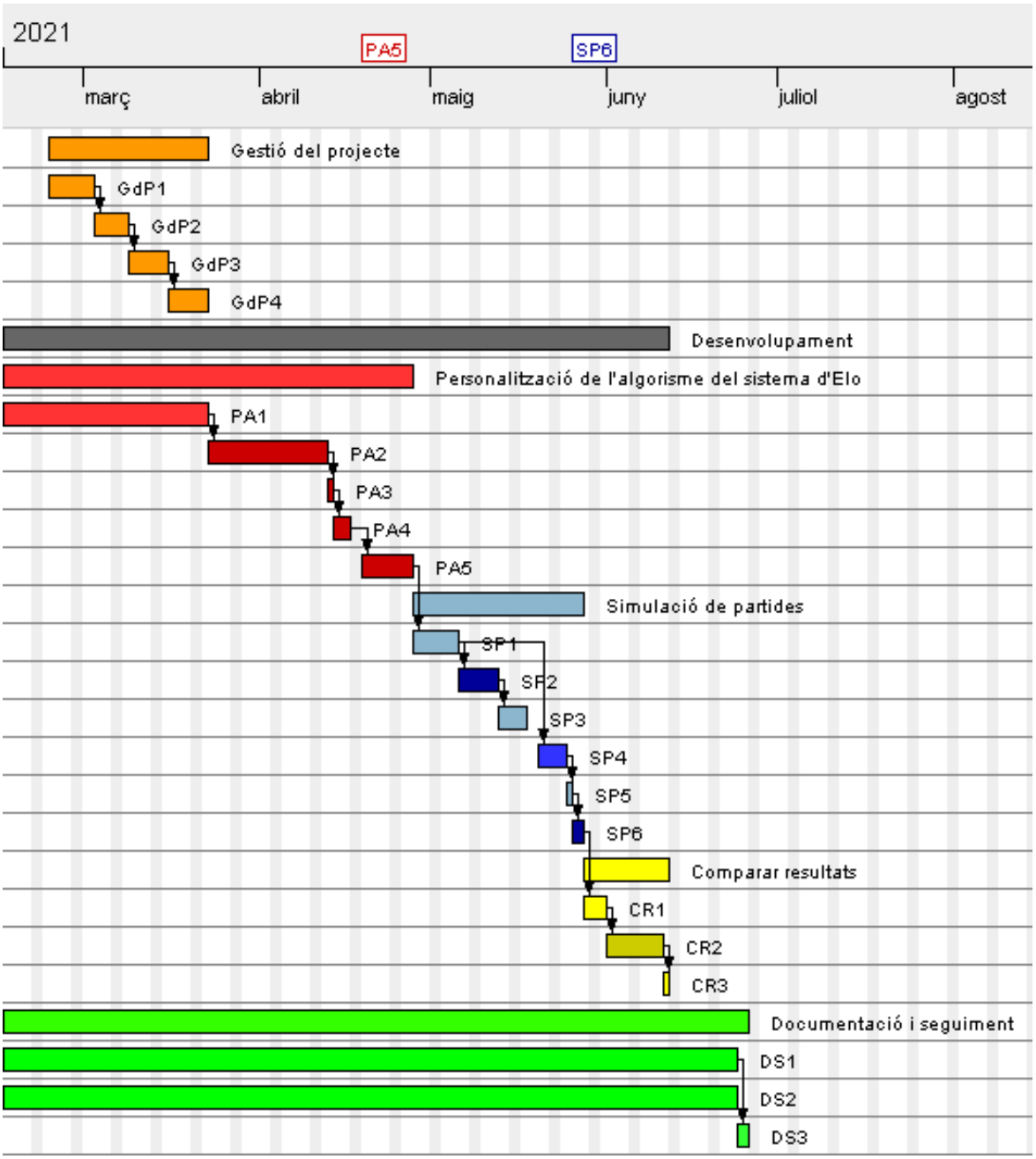

Figura 16: Diagrama de Gantt del projecte part visual. Font: Elaboració pròpia

11 Gestió del risc

Durant el desenvolupament del projecte poden sortir un seguit de riscos que el poden afectar negativament, arribant a endarrerir les tasques programades o, fins i tot, haver-ne d'eliminar alguna per la falta de temps. És per aquest motiu que cal identificar-los bé i preparar plans alternatius per arribar a solucionar aquests problemes, per poder acabar el projecte a temps i amb totes les funcionalitats. A la taula 3 es pot observar els diversos riscos considerats, junt amb la probabilitat de que passin i amb l'impacte que poden tenir sobr el treball.

Risc	Probabilitat	Impacte
Data d'entrega fixa	Baixa	Alt
Desconeixement en simulació	Alt	Baix
Desconeixement en les tecnologies emprades	Alt	Baix
Bugs i errors	Alt	Mig

Taula 3: Taula dels possibles riscos amb les seves probabilitats de que passin. Font: Elaboració pròpia

Ara que ja es coneixen els riscos i la seva probabilitat de que apareguin, s'explicaran els plans alternatius per solucionar-los.

- **Data d'entrega fixa:** Com s'ha mostrat anteriorment, la probabilitat de que passi aquest risc és baixa, ja que tota la planificació s'ha fet pensant en la data màxima. Tot i així, pot arribar a tindre conseqüències molt greus, perquè en els pitjors dels casos es pot arribar a eliminar una funcionalitat de les proposades per arribar a l'entrega. Per evitar-ho, el que es farà serà fer una bona planificació tenint en compte tots els problemes i donant un marge de temps per poder-los solucionar. No farà falta cap recurs addicional.
- **Desconeixement en simulació:** Com es pot observar en la taula 3 prèviament vista, és molt probable que aparegui algun problema degut al desconeixement que hi ha sobre la simulació. Un pla alternatiu per solucionar-ho és afegir més hores de les necessàries a les respectives tasques per així no generar endarreriments. També es preguntarà al professor responsable de l'assignatura sobre algun dubte en concret. Aquest risc només tindrà un impacte baix, ja que només pot arribar a generar un endarreriment. No farà falta cap recurs addicional.
- **Desconeixement en les tecnologies emprades:** Com es pot veure en la taula 3, la probabilitat de que surti un risc relacionat amb el desconeixement de les tecnologies emprades és molt alta. Aquest risc té un impacte mitjà ja que hi ha vàries tecnologies de les quals no hi ha gaire coneixement i, per tant, poden generar diversos retards. Per aquest motiu, s'estudiarà prèviament el seu funcionament i s'afegiran més hores de les necessàries a les tasques on s'utilitzin. No serà necessari cap recurs addicional.
- **Bugs i errors:** En el desenvolupament de tot programa informàtic, és molt habitual que apareguin bugs i errors a l'hora de fer codi i testejar. Per aquest motiu, la probabilitat de que passi en el projecte és molt alta i per solucionar-ho es faran tests unitaris. En cas de que alguna funcionalitat testejada doni un error més tard, se li dedicarà temps per trobar-lo i corregir-lo. L'impacte que pot arribar a tenir és el d'alentir el desenvolupament del projecte i no farà falta cap recurs addicional per corregir-lo.

12 Gestió econòmica

12.1 Identificació i estimació de costos

Un aspecte molt important a tindre en compte a l'hora de desenvolupar un projecte és el pressupost. Per aquest motiu, s'han de valorar tots els costos possibles: recursos humans, recursos materials, recursos generals, contingència i imprevistos.

12.2 Modificació del pressupost respecte a la planificació inicial

A GEP es fa una planificació inicial del projecte, on també es té en compte el pressupost. També s'ha mencionat prèviament que s'ha afegit una nova tasca respecte a la idea original. Aquesta ha comportat un canvi en el pressupost del projecte molt lleu, ja que les noves hores destinades a aquesta feina han estat escurçades d'altres. D'aquesta manera, cada rol ha acabat fent gairebé les mateixes hores que tenia assignades en un inici.

A continuació es calcularà cada un dels costos actualitzats i finalment es veurà el cost final del projecte.

12.2.1 Recursos humans

Com s'ha explicat prèviament, dintre dels recursos humans hi ha diferents rols. Aquests tenen funcions diferents i treballaran més o menys depenent de les tasques descrites prèviament en el diagrama de Gantt. Per tant, per poder saber el cost total del projecte, cal saber primer el seu sou. Aquest s'ha calculat fent la mitjana dels sous que es mostren en la pàgina de Tecnoempleo.com [33] i que es poden veure a la taula 4. Aquesta conté els diversos rols, els seus salaris brut per hora i els mateixos tenint en compte la seguretat social (SS).

Rol	Sou/hora (brut)	Sou/hora + SS(x1,3) (brut)
Cap de projecte	23,54€	30,61€
Analista programador	16,74€	21,77€
Arquitecte del Software	20,77€	27€
Dissenyador d'UI	15,57€	20,24€
Programador	15,51€	20,17€
Tester	15,40€	20,02€

Taula 4: Taula dels sous per hora dels diferents rols del projecte. Font: Elaboració pròpia

Un cop coneguts els preus, ja es pot calcular el de cada tasca. Per fer-ho, es calcularà les hores necessàries de cada rol per cada una d'elles, mostrades anteriorment en el diagrama de Gannt. El cost total de les tasques es pot veure en la taula 5, on s'indica el nom de la feina a fer, i per cada una, les hores per cada respectiu rol, les hores totals de la tasca i el seu cost.

Tasca	Hores totals	CP	AP	AS	DUI	P	T	Cost
GdP	**70**							
GdP1	25	25						765,25€
GdP2	10	10						306,10€
GdP3	15	15						459,15€
GdP4	20	20						612,20€
D	**270**							
PA1	10		5	3,5		1	0,5	233.53€
PA2	50		5	5		35	5	1049,90€
PA3	5		0,5	0,5		3,5	0,5	104,99€
PA4	15		1	1		10,5	2,5	310,61€
PA5	35		3	3		24,5	4,5	735,57€
SP1	30		3	3		22,5	1,5	630,17€
SP2	25		2	2		18	3	419,96
SP3	15		1,5	1,5		10,5	1,5	314,97€
SP4	15		1,5	1,5		10,5	1,5	314,97€
SP5	5		0,5	0,5		3,5	0,5	104,99€
SP6	10		1	0,5		7	1,5	206,49€
CR1	10		0,5	1,5		7	1	212,60€
CR2	40		2	2	18	14	4	824,32€
CR3	5		0,5	0,5		3,5	0,5	104,99€
DS	**110**							
DS1	75	75						2295,75€
DS2	25	25						765,25€
DS3	10	10						306,10€
TOTAL HORES	**450**	**180**	**27**	**26**	**18**	**171**	**28**	
TOTAL COST		**5.509,80€**	**587,79€**	**702€**	**364,32€**	**3.449,07€**	**560,56€**	**11.173,54€**

Taula 5: Taula dels costos de les tasques del projecte. Font: Elaboració pròpia

12.2.2 Recursos materials

Un altre aspecte a tenir en compte a l'hora de calcular el pressupost d'un projecte són els recursos materials. En aquest cas, s'utilitzarà un HP Laptop 15s-fq2093ns [34] durant tot el desenvolupament i un ratolí inalàmbric HP 220 [35]. Respecte els softwares que s'utilitzaran, hi han alguns que son de franc, com és el cas del Lua o el Github, i d'altres de pagament, com el Visual Studio o Qt.

A la taula 6 es pot observar els recursos materials de hardware amb els seus costos, els seus anys de vida útil i les seves amortitzacions. Aquests càlculs estan fets sobre 4 anys de vida útil i amb una utilització de 5 hores diàries. La fórmula utilitzada pel càlcul de l'amortització és la següent:

$$\frac{cost\,del\,hardware(euros)}{vida\,útil(anys) * 220\,dies\,laborables/any * hores\,dedicades\,al\,dia} * durada\,del\,projecte(hores)$$

Hardware	Cost	Vida útil	Amortització
HP Laptop 15s-fq2093ns	650€	4 anys	66,48€
Ratolí inalàmbric HP 220	20€	4 anys	2,05€
Total	**670€**		**68,53€**

Taula 6: Taula dels costos dels recursos hardware i la seva amortització. Font: Elaboració pròpia

A la taula 7 es pot observar els recursos materials de software i els seus costos. Aquesta inclou la llicència de Windows [36], la del Visual Studio [37] i la de QT [38]. Aquesta última s'ha de pagar per mesos durant tot un any.

Software	Cost/mes	Mesos	Cost
Windows	145€		145€
Visual Studio	45€	5	225€
QT	42€	12	504€
Total			**874€**

Taula 7: Taula dels costos dels recursos hardware i la seva amortització. Font: Elaboració pròpia

Finalment, a la taula 8 es pot observar el cost total dels recursos materials.

Recurs material	Cost
Hardware	670€
Software	874€
Total	**1554€**

Taula 8: Taula dels costos dels recursos hardware i la seva amortització. Font: Elaboració pròpia

12.2.3 Recursos generals

A part dels recursos humans i materials, per poder calcular el pressupost sencer del projecte també s'han de tindre en compte altres elements com l'Internet, l'electricitat, el desplaçament, l'espai de treball, etc. En aquest cas, el treball és desenvolupat per una sola persona des de casa, per tant, només es contemplaran els dos primers costos. Per l'electricitat, es tindran en

compte dos llums LEDs que consumeixen 14 W i l'ordinador que consum 150 W. A la taula 9 es pot veure el cost dels diversos recursos generals.

Recursos	Cost	Cost Total
Electricitat	0,1458€/kWh	11,68€
Internet	48,40€/mes	193,60€
Total		**205,28€**

Taula 9: Taula dels costos dels recursos generals. Font: Elaboració pròpia

12.2.4 Contingència

Un altre aspecte a tenir present és la contingència, és a dir, és un cost que se li suma al pressupost per cobrir aquells imprevists que no s'han anticipat. Es calcula amb un percentatge del cost dels recursos calculats. En el sector informàtic, aquest percentatge varia entre el 10% i el 20%, per tant, s'escollirà el punt mig, és a dir, el 15%. A la taula 10 es pot observar la taula de contingència pels diferents recursos calculats anteriorment.

Tipus de recursos	Cost	% de Contingència	Cost contingència
Recursos humans	11.173,54€	15%	1.676,04€
Recursos materials	1544€		231,60€
Recursos generals	205,28€		30,72€
Total			**1.938,36€**

Taula 10: Taula de contingència dels diversos recursos. Font:; Elaboració pròpia.

12.2.5 Cost d'imprevistos

Finalment, cal calcular el cost de tots aquells imprevistos que s'han identificat prèviament en l'apartat de riscos. A la taula 11 es pot veure el cost de cada un d'ells junt amb la seva probabilitat de que passin.

Risc	Probabilitat	Temps	Cost
Data entrega fixa	10%	10 hores	30,61€
Desconeixement en simulació	66%	10 hores	133,12€
Desconeixement en tecnologies emprades	66%	15 hores	199,68€
Bugs i errors	66%	20 hores	264,26€
Total			**627,67€**

Taula 11: Taula del cost d'imprevistos. Font: Elaboració pròpia

12.2.6 Pressupost final

Un cop ja s'han calculat tots els càlculs necessaris dels diferents costos, ja es pot determinar el pressupost final que costarà el projecte. Aquest es pot veure en la taula 12, on si s'arrodoneix el preu a l'alça, queda amb un total de 15.489€, gairebé 15.500€.

	Cost
Recursos humans	11.173,54€
Recursos materials	1544€
Recursos generals	205,28€
Contingència	1.938,36€
Imprevistos	627,67€
Total	**15.488,85€**

Taula 12: Taula de pressupost final. Font: Elaboració pròpia

12.3 Control de gestió

Planificar i gestionar el pressupost d'un projecte és molt important, però també s'ha de gestionar els diners i calcular les desviacions econòmiques per cadascuna de les tasques o costos calculats prèviament. Per fer-ho, per cada una de les tasques fetes, s'anotarà les hores reals que han estat necessàries i s'empraran les següents fórmules.

- **Desviació d'hores per tasca**

$$(Hores\,estimades - Hores\,reals)$$

- **Desviació de costos per tasca**

$$(Hores\,estimades - Hores\,reals) * Cost\,real$$

- **Desviació d'hores dels recursos humans per cada tasca**

$$(Hores\,estimades - Hores\,reals)$$

- **Desviació dels costos de recursos humans per cada tasca**

$$(Cost\,estimat - Cost\,real) * Hores\,reals$$

- **Desviació total de recursos humans**

$$Costos\,recursos\,humans\,estimats - Costos\,recursos\,humans\,reals$$

- **Desviació total de recursos materials**

$$Costos\,recursos\,materials\,estimats - Costos\,recursos\,materials\,reals$$

- **Desviació total de recursos generals**

$$Costos\ recursos\ generals\ estimats - Costos\ recursos\ generals\ reals$$

- **Desviació total de contingències**

$$Cost\ contingncia\ estimat - Cost\ contingncia\ real$$

- **Desviació total d'imprevistos**

$$Costos\ imprevistos\ estimats - Costos\ imprevistos\ reals$$

- **Desviació total d'hores**

$$Hores\ estimades - Hores\ reals$$

- **Desviació total de pressupost final**

$$Pressupost\ final\ estimat - Pressupost\ final\ real$$

13 Informe de sostenibilitat

Un factor molt important a l'hora de desenvolupar un projecte és la seva sostenibilitat. Pensar en l'impacte ambiental, econòmic i social que pot tindre és un requisit indispensable. Tot i això, no sempre es tenen en compte, ja que els interessos comercials solen tenir prioritat.

Pel que respecta a EloSim, durant el seu desenvolupament s'ha procurat tindre una bona sostenibilitat, intentant respectar els tres camps de la millor manera possible. També cal considerar que el projecte està pensat per a que en un futur sigui multithreading. Això agilitzarà el càlcul i farà el software més eficient. Tot seguit, es detallarà cada un d'ells.

13.1 Dimensió ambiental

Sobre l'aspecte ambiental, el projecte utilitza el mínim de recursos possibles. El seu desenvolupament ha estat fet des de casa, per tant, només s'ha requerit d'electricitat per fer servir l'ordinador. Com a conseqüència, és molt complicat reduir el seu impacte ambiental. Tot i això, un cop s'hagi fet multithreading, l'electricitat requerida per utilitzar-lo disminuirà.

Tanmateix, el software desenvolupat té un impacte positiu en la petjada ecològica. Les empreses que l'utilitzin estalviaran temps per disenyar el seu algorisme d'Elo ideal, i per tant, el podran dedicar a desenvolupar els altres aspectes del videojoc. Com a resultat, aquest es finalitzarà abans i, per tant, hi haurà una reducció de recursos utilitzats.

13.2 Dimensió econòmica

Per fer el pressupost del projecte s'han contemplat els diferents recursos materials i humans necessaris per dur-lo a terme. També s'ha tingut en compte els diferents imprevistos que poden sorgir durant el seu desenvolupament.

Per tal de reduir el seu cost, es pot contractar a treballadors especialitzats en els camps o llenguatges que s'utilitzen. Això reduiria el temps per realitzar les diverses tasques, i per consegüent, un estalvi econònim dels recursos humans i materials.

Per una altra banda, un cop realitzat el multithreading, ell programa s'executarà més ràpidament i seran necessaris menys mitjans. Com a resultat, hi haurà un estalvi econòmic.

13.3 Dimensió social

Realitzar aquest treball ha servit per augmentar el conneixement i experiència sobre els llenguatges i eines utilitzats. També ha servit per veure com funcionen els projectes petits i com de similar o diferent pot ser la planificació de la realitat. A més, s'ha hagut de prendre decissions importants a l'hora de desenvolupar certes funcionalitats.

Per una altra part, el software produit serà utilitzat per les empreses desenvolupadores de videojocs. Gràcies a la seva personalització i adaptabilitat, quan els treballadors hagin de disenyar un bon algorisme d'Elo podran fer servir aquest producte per veure si s'adequa o no a les seves necessitats. D'aquesta manera, podran testejar les diferents fórmules d'Elo sense haver de perdre temps en programar simulacions diferents per a cada una d'elles.

14 Lleis i regulacions

A l'hora de fer un programa informàtic és molt important complir les lleis i satisfer les regulacions. L'incompliment d'una d'elles pot provocar greus conseqüències que poden ser penalitzades de forma molt severa.

Si revisem el funcionament del programa, es pot arribar a la conclusió de què no vulnera cap llei ni incompleix cap legislació. Totes les dades utilitzades són creades de forma aleatòria o introduïdes pel propi usuari, i únicament són utilitzades per fer les simulacions. En cap moment són emmagatzemades en cap base de dades ni s'utilitza cap nom real, degut a que els jugadors funcionen amb IDs, no amb noms. Per una altra banda, el projecte tampoc té connexió a Internet. Com a resultat, no hi ha cap llei o regulació a vulnerar amb aquest aspecte. Finalment, l'ús del programa tampoc està restringit a cap edat. Si no és utilitzat amb l'objectiu d'estudiar una fórmula d'Elo particular, l'únic resultat que trobarà l'usuari serà un Excel amb la simulació executada, el qual no generarà cap risc ni inconvenient pel consumidor.

Pels motius que s'acaben d'explicar, es pot concloure que el meu TFG no vulnera cap llei ni regulació ni tampoc està lligat a cap d'elles.

15 Conclusions

Un cop finalitzat el projecte, cal analitzar si s'han complert les funcionalitats principals i les competències transversals que es van escollir en un principi. També cal detectar quines millores se li poden fer al producte per incrementar el seu rendiment i millorar l'experiència de l'usuari.

15.1 Resultats

15.1.1 Assoliment dels objectius

En l'assignatura de GEP, es van plantejar una sèrie de tres grans d'objectius que havia de complir el projecte. Aquests eren:

- La personalització de l'algorisme del sistema d'Elo
- La simulació de partides
- Comparar els resultats entre diferents variables i algorismes

Finalment, s'ha pogut implementar totes aquestes funcionalitats. Es pot personalitzar l'algorisme d'Elo i les selves variables, es pot configurar la simulació de les partides tenint en conte les propietats del jugadors, i es poden extreure les dades a un fitxer .csv. Per tant, s'han assolit tots els objectius.

15.1.2 Assoliment de les competències tècniques

A l'hora de fer la preinscripció del TFG, es van haver d'escollir una sèrie de competències tècniques que havia de complir el projecte. A continuació s'analitzaran una per una per veure si s'han assolit o no.

CES1.1: Desenvolupar, mantenir i avaluar sistemes i serveis software complexos i/o crítics [En profunditat]

Les parts més complexes del sistema han estat la d'implementar el matchmaking i els algorismes al fitxer Lua. En el primer cas, aparellar els jugadors de la manera més adequada respectant el seu nivell d'Elo va ser un repte. Però afortunadament, s'ha pogut implementar de la manera més equilibrada possible.

En el segon cas, la inexpertesa en el llenguatge Lua i la complexitat de passar i accedir als paràmetres de les funcions va generar grans problemes. Tot i això, s'ha aconseguit programar els algorismes després d'aprendre el llenguatge i consultar fòrums d'Internet.

CES1.2: Donar solució a problemes d'integració en funció de les estratègies, dels estàndards i de les tecnologies disponibles [Bastant]

S'ha aconseguit integrar llenguatges i tecnologies externs al Visual Studio, com és el Lua i el QT. El programa interacciona de la manera adequada amb el fitxer Lua i la interfície gràfica.

CES1.3: Identificar, avaluar i gestionar els riscos potencials associats a la construcció de software que es poguessin presentar [En profunditat]

Aquest software ha de ser capaç d'interaccionar amb algorismes creats per l'usuari. Per tant, el programa s'ha desenvolupat contemplant els diferents riscos que podien aparèixer durant el seu desenvolupament i que poden passar en el seu futur amb la interacció del client.

CES1.7: Controlar la qualitat i dissenyar proves en la producció de software [Una mica]

El sistema ha de funcionar de forma correcta cada cop que s'afegeix una nova funcionalitat. Per assegurar-ho, s'han dissenyat diversos unit tests. Aquests s'encarreguen de comprovar que la creació d'equips i de partides funcionen sense cap error. També ho comproven per a les diverses simulacions i el càlcul de l'Elo resultant.

CES2.1: Definir i gestionar els requisits d'un sistema software [En profunditat]

A l'assignatura de GEP ja es van definir i gestionar els diferents requisits del projecte. Des de llavors, a mesura que el software s'ha anat desenvolupant, s'han anat revisant si es complien o no. A més, durant les diverses reunions fetes amb el director, s'han anat repassant els requisits, i en cas de que no fossin correctes, modificant-los.

15.2 Futur

El projecte ha assolit tots els diferents objectius sobre el seu funcionament, però això no significa que no es pugui millorar. Deguda a la curta duració del TFG, hi han algunes idees que s'han hagut de descartar. Si s'hagués disposat de més temps, es podrien haver fet les següents millores:

- **Multithreading:** Actualment el software només aprofita un thread o nucli de l'ordinador. Una manera d'incrementar el rendiment del programa seria paral·lelitzar el seu funcionament, aprofitant els diversos nuclis dels quals disposa el PC.
- **Implementar el matchmaking al Lua:** Actualment, el matchmaking és dels pocs algorismes implementats en C++. Com a conseqüència, l'usuari no el pot modificar. Si l'emparellament es definís al fitxer Lua, el client podria fer els canvis que necessités per adaptar la simulació a les seves necessitats.

- **Sortida de dades al fitxer .csv:** Un cop executada la simulació, el client obté un fitxer .csv amb les estadístiques d'aquesta. Les dades que es mostren són els diversos jugadors amb les seves puntuacions en cada ronda. Una manera de millorar-ho seria permetre al client decidir quina informació vol que aparegui en el fitxer resultant.

15.3 Conclusions finals

El TFG és un treball molt important on l'alumne mostra les habilitats i capacitats apreses durant la carrera. Aquest objectiu no només ha estat assolit, sinó que també s'han après nous aspectes i s'han millorat d'altres.

Per una banda, aquest projecte ha servit per aplicar molts coneixements apresos en les diverses assignatures de la carrera. Alguns d'ells han estat saber programar amb el llenguatge C++, gestionar projectes o identificar i especificar requisits.

Per una altra banda, també ha servit per aprendre una gran varietat de nous aspectes. En primer lloc, ha permès millorar el coneixement i nivell de C++. En segon lloc, s'ha après un nou llenguatge de programació, el Lua. També ha permès veure com evoluciona un projecte i quines accions prendre per tal de dur-lo a terme de la millor manera possible. Finalment, ha permès millorar les habilitats de programar per fer un codi més eficient i estructurant.

El treball de final de grau també ha servit per evolucionar com a persona: saber prendre decisions importants, ser més responsable i més organitzat, i estar més preparat pel món laboral. A més, s'ha pogut aprendre més sobre el món dels videojocs, un dels motius pel qual s'havia escollit aquest projecte.

En conclusió, gràcies al TFG s'han pogut aplicar coneixements apresos durant a la carrera, aprendre nous aspectes, millorar-ne d'altres i evolucionar com a persona.

Referències

[1] *Los 10 Eventos de eSports Más Esperados en 2021*. [en línia]. Influencer Marketing Hub. 2021. URL: https://influencermarketinghub.com/es/principales-10-eventos-de-esports/ (cons. 23-05-2021).

[2] *Evolution Championship Series*. [en línia]. Wikipedia. 2021. URL: https://en.wikipedia.org/wiki/Evolution_Championship_Series (cons. 05-06-2021).

[3] *¡Ni Fortnite ni GTA V! Minecraft y Roblox, los videojuegos más vistos en YouTube en 2020*. [en línia]. Grupo Milenio. 2020. URL: https://www.milenio.com/tecnologia/youtube-gaming-minecraft-roblox-vistos-2020 (cons. 23-05-2021).

[4] *Los 20 mejores juegos para empezar a streamear en Twitch y consejos para encontrar el juego que te conseguirá un montón de seguidores*. [en línia]. Business Insider. 2021. URL: https://www.businessinsider.es/20-mejores-juegos-empezar-streamear-exito-twitch-798681 (cons. 23-05-2021).

[5] *Videogames are a bigger industry than movies and North American sports combined, thanks to the pandemic*. [en línia]. MarketWatch. 2020. URL: https://www.marketwatch.com/story/videogames-are-a-bigger-industry-than-sports-and-movies-combined-thanks-to-the-pandemic-11608654990 (cons. 26-02-2021).

[6] *Elo rating system*. [en línia]. Wikipedia. 2021. URL: https://en.wikipedia.org/wiki/Elo_rating_system (cons. 26-02-2021).

[7] *Definición de Matchmaking*. [en línia]. GamerDic, Diccionario online de términos sobre videojuegos y cultura gamer. 2020. URL: http://www.gamerdic.es/termino/matchmaking (cons. 27-02-2021).

[8] Jamie Hore. *Rocket League Ranks: ranking system explained, MMR, rewards, and more*. [en línia]. The Loadout. 2021. URL: https://www.theloadout.com/rocket-league/ranks (cons. 23-05-2021).

[9] Jonathan Deesing. *League Of Legends Ranking System Explained*. [en línia]. Red Bull. 2020. URL: https://www.redbull.com/us-en/the-league-ranked-system-explained#:~:text=The%20nine%20tiers%2C%20or%20ranks,Master%2C%20GrandMaster%2C%20and%20Challenger. (cons. 23-05-2021).

[10] *CSGO Elo: The definite Guide how CSGO Elo Rankings work*. [en línia]. CSGO Ranks. 2021. URL: https://csgo-ranks.com/elo/ (cons. 23-05-2021).

[11] *Clasificación y sistema de ELO en CS:GO: Guía detallada*. [en línia]. Revistadedalo.com. 2020. URL: http://revistadedalo.com/clasificacion-elo-csgo-guia-detallada/ (cons. 23-05-2021).

[12] RemiFabre. *Elo*. [en línia]. Github. 2020. URL: https://github.com/RemiFabre/Elo (cons. 27-02-2021).

[13] cardsorg. *Elo*. [en línia]. Github. 2018. URL: https://github.com/cardsorg/Elo (cons. 27-02-2021).

[14] iain. *Elo*. [en línia]. Github. 2015. URL: https://github.com/iain/Elo (cons. 27-02-2021).

[15] Wojciech Sas. *Elo Calculator*. [en línia]. Omnicalculator. 2020. URL: https://www.omnicalculator.com/sports/elo (cons. 27-02-2021).

[16] *The Ultimate Guide... Waterfall Model*. [en línia]. Project Manager. 2021. URL: https://www.projectmanager.com/waterfall-methodology (cons. 28-02-2021).

[17] *SDLC - Waterfall Model*. [en línia]. Tutorials Point. URL: https://www.tutorialspoint.com/sdlc/sdlc_waterfall_model.htm (cons. 28-02-2021).

[18] *Agile Vs Waterfall: Know the Difference Between Methodologies*. [en línia]. Guru99. URL: https://www.guru99.com/waterfall-vs-agile.html (cons. 28-02-2021).

[19] *Scrum (desarrollo de software)*. [en línia]. Wikipedia. 2021. URL: https://es.wikipedia.org/wiki/Scrum_(desarrollo_de_software) (cons. 13-06-2021).

[20] *La metodología ágil Kanban: en qué consiste y para qué sirve*. [en línia]. EALDE Business School. 2020. URL: https://www.ealde.es/metodologia-agil-kanban/ (cons. 13-06-2021).

[21] *El primer tablero Kanban del equipo • Jeronimo Palacios & Associates*. [en línia]. Jeronimo Palacios & Associates. 2020. URL: https://jeronimopalacios.com/agile/el-primer-tablero-kanban/ (cons. 13-06-2021).

[22] James & Suzanne Robertson. *Plantilla de Especificación de Requisitos*. [en línia]. Volere. 2006. URL: https://www.volere.org/wp-content/uploads/2018/12/template_es.pdf (cons. 08-06-2021).

[23] *Diagrama de clases*. [en línia]. Wikipedia. 2021. URL: https://es.wikipedia.org/wiki/Diagrama_de_clases (cons. 11-06-2021).

[24] *Lua - About*. [en línia]. Lua. 2020. URL: http://www.lua.org/about.html (cons. 28-02-2021).

[25] *Latex*. [en línia]. Wikipedia. 2021. URL: https://es.wikipedia.org/wiki/LaTeX (cons. 22-05-2021).

[26] *Git*. [en línia]. Git. URL: https://git-scm.com/ (cons. 28-05-2021).

[27] *Github*. [en línia]. Github. URL: https://github.com/ (cons. 28-02-2021).

[28] *Google Meet*. [en línia]. Google. URL: https://meet.google.com/ (cons. 03-03-2021).

[29] *Trello*. [en línia]. Trello. URL: https://trello.com (cons. 28-02-2021).

[30] *Distribución normal*. [en línia]. Wikipedia. 2021. URL: https://es.wikipedia.org/wiki/Distribuci%C3%B3n_normal#Estandarizaci%C3%B3n_de_variables_aleatorias_normales (cons. 28-05-2021).

[31] *Campana de Gauss*. [en línia]. Máster MBA Málaga. 2016. URL: https://www.master-malaga.com/master-mba/campana-de-gauss/ (cons. 28-05-2021).

[32] Lance McDiffett. *The Math Behind Your Competitive Overwatch Match*. [en línia]. Towards Data Science. 2020. URL: https://towardsdatascience.com/the-math-behind-your-competitive-overwatch-match-a5184fc5a50f#:~:text=Very%20broadly%2C%20this%20is%20how,player%20with%20a%20lower%20SR (cons. 07-06-2021).

[33] *Informe Empleo Informática - Marzo 2021*. [en línia]. Tecnoempleo.com. 2021. URL: https://www.tecnoempleo.com/informe-empleo-informatica.php (cons. 12-03-2021).

[34] *Portátil - HP Laptop 15s-fq2023ns*. [en línia]. MediaMarkt. URL: https://www.mediamarkt.es/es/product/_port%C3%A1til-hp-laptop-15s-fq2023ns-15-6-fhd-intel%C2%AE-core%E2%84%A2-i7-1165g7-8gb-512gb-ssd-freedos-plata-1501323.html?utm_source=google&utm_medium=cpc&utm_campaign=MM_ES_SEARCH_GOOGLE_CATEGORIES_PLA_PLA-SMART_INFORMATICA_ALL_ALL&gclsrc=aw.ds&ds_rl=1280902&gclid=Cj0KCQiAv6yCBhCLARIsABqJTjYtER5eRifKONq1I3QUJGQqBKmUF7XDuEekP9o2alsEAo65HP%20tRDWUaAugWEALw_wcB (cons. 13-03-2021).

[35] *HP 220 Ratón inalámbrico*. [en línia]. HP. URL: https://www.pccomponentes.com/hp-200-raton-inalambrico-1000-dpi?gclid=CjwKCAiA4rGCBhAQEiwAelVti4ipabfJailk5V-02t3Om42P_9Yjes_Pey8tgRX263Vp5Q3fZlPFSBoCjv8QAvD_BwE&gclsrc=aw.ds (cons. 13-03-2021).

[36] *Windows 10 Home*. [en línia]. Microsoft. URL: https://www.microsoft.com/es-es/p/windows-10-home/D76QX4BZNWK4?icid=Cat-Windows-mosaic_linknav-1-WindowsHome-es_es&activetab=pivot%3aoverviewtab (cons. 15-03-2021).

[37] *Comprar Visual Studio - Business*. [en línia]. Microsoft. URL: https://visualstudio.microsoft.com/es/vs/pricing/ (cons. 15-03-2021).

[38] *Qt for Small Business*. [en línia]. QT. URL: https://www.qt.io/qt-for-small-business (cons. 15-03-2021).

[39] *Waterfall Model*. [en línia]. Castellan Systems. 2021. URL: https://castellansystems.com/kb/Waterfall.cshtml (cons. 28-02-2021).

[40] *Patrón de diseño*. [en línia]. Wikipedia. 2021. URL: https://es.wikipedia.org/wiki/Patr%C3%B3n_de_dise%C3%B1o (cons. 15-06-2021).