



UNIVERSITAT POLITÈCNICA DE CATALUNYA

Escola Politècnica Superior d'Enginyeria de Manresa

---

# Crear una aplicació per al joc de rol *“La llave del Horizonte”*

8 de juliol de 2022

---

treball de fi de grau que presenta

**MARTA ESPEJO FERNANDEZ**

en compliment dels requisits per assolir el

**GRAU D'ENGINYERIA EN SISTEMES TIC**

Direcció: Marta Isabel Tarres Puertas

Aquesta obra està subjecta a una llicència Attribution-NonCommercial-ShareAlike 3.0 Spain de Creative Commons. Per veure'n una còpia, visiteu <https://creativecommons.org/licenses/by-nc-sa/3.0/es/deed.ca> o envieu una carta a Creative Commons, 171 Second Street, Suite 300, San Francisco, California 94105, USA.

## Agraïments

Agraïments a Ivan Sanchez per ajudar-me amb la programació de l'aplicació i a Gonzalo Rodríguez per deixar-me utilitzar el seu joc com a base per aquest projecte a més a més de fer tots els dibuixos utilitzats a l'aplicació.



# Resum

Al llarg d'aquest treball s'intenta trobar una solució a un problema que tenen un grup d'amics a l'hora de jugar a un joc de rol que ha inventat un d'ells. La qüestió ve que totes les dades i dibuixos del joc estan fets a paper i llapis. Això fa que sigui difícil d'accedir a tota la informació de forma ràpida i còmoda en qualsevol lloc.

Per a solucionar això s'ha proposat crear una aplicació mòbil on guardar totes les dades. A més a més, aquesta aplicació també hauria d'ajudar amb totes les dades que es creen al llarg d'una partida. Qui té que recursos, quants diners té el grup, quines són les estadístiques de tal persona, etc. Per fer això i que tothom estigui al dia dels últims canvis és necessari connectar als jugadors perquè es puguin passar aquestes dades. Aquest problema s'ha acabat solucionant creant una xarxa wifi local i privada amb un servidor que s'encarrega de gestionar la base de dades que guarda totes aquestes dades.

# Abstract

In this project, we're going to try to come up with a solution to a problem that a group of friends has. The thing is, one of their friends invented a role-playing game, but all the information about it is written, or drawn, with a pencil and on paper. This can make the game play a bit difficult due to the difficulty in finding the information easily. Or the fact that every time someone wants to play they have to grab the paper and go wherever they want to play.

It was suggested to create a mobile application where we can record the information as a solution to the problem. This app may also be useful to help recompile all the information created during a game. Information like who has which resources, how much money the group has and so on. For this to happen and to be useful, this information must be shared with the group. In order to be able to share information with the group, it was decided to do local private wireless red. Said red will have a server responsible for managing the database which has all the information about the game.



# Índex

<b>Resum</b>	<b>i</b>
<b>Abstract</b>	<b>i</b>
<b>I. Memòria</b>	<b>1</b>
<b>1. Introducció</b>	<b>3</b>
<b>2. Antecedents</b>	<b>5</b>
<b>3. Manual Tècnic</b>	<b>7</b>
3.1. Tecnologies utilitzades . . . . .	7
3.2. Tractament de les dades . . . . .	8
3.3. Servidor . . . . .	11
3.4. Aplicació . . . . .	14
<b>4. Manual d'usuari</b>	<b>17</b>
4.1. Master . . . . .	17
4.2. Jugador . . . . .	22
4.2.1. Crear un personatge . . . . .	24
4.3. Lider . . . . .	25
4.4. Errors comuns . . . . .	26
<b>5. Conclusions</b>	<b>29</b>
<b>Bibliografia</b>	<b>31</b>
<b>II. Apèndixs</b>	<b>33</b>
<b>A. Normes del Joc per <i>Gonzalo Rodríguez</i></b>	<b>35</b>
A.1. Creación de personajes . . . . .	35
A.2. Sistema de clases . . . . .	36
A.3. Atributos . . . . .	36
A.3.1. Mejora de los atributos . . . . .	37
A.4. Funcionamiento de la magia . . . . .	37
A.5. Combate . . . . .	38
A.5.1. Encuentros . . . . .	38
A.5.2. Ataque . . . . .	38
A.5.3. Aciertos y evasión . . . . .	38
A.5.4. Resistencias y debilidades . . . . .	38

A.6. Inicio de la partida . . . . .	39
<b>B. Codí i taules de dades</b>	<b>41</b>
B.1. Codí de la aplicació . . . . .	41
B.2. Codi del servidor . . . . .	41
B.3. Codi de la Base de Dades . . . . .	46
B.4. Taules de dades . . . . .	53



**Part I.**

**Memòria**



# 1. Introducció

Aquest treball sorgeix del desig d'un grup d'amics de poder jugar a un joc de rol, inventat per un d'ells, sense haver d'estar pendent de portar de dalt a baix tots els papers amb la informació del joc. A més a més de fer facilitar l'accés a ceta informació durant la partida i intentar trobar una millor manera d'ensenyar fotos de llocs i coses al grup. Fins al moment això últim s'havia fet posant un full amb el dibuix fet a llapis en mig del grup de gent perquè tothom ho pogués veure.

A aquest joc se li ha acabat posant el nom *"La llave del horizonte"* i és un joc de rol. En preguntar al seu creador com el resumiria i quins serien els seus objectius aquest el va definir amb la següent frase. *"Juego de ambientación dieselpunk y fantasía donde el objetivo es resolver los antiguos misterios que rodean el continente y encontrar la llave del horizonte."* frase de Gonzalo Rodríguez. Es recomanaria jugar a aquest joc amb un grup d'entre 4 i 8 persones en total, de forma que hi hauria 1 Màster i entre 3 i 7 jugadors.

Un cop analitzada la situació en la qual es va plantejar aquest treball anem a mirar quins són els objectius que es volen aconseguir. Primer de tot s'ha pensat a fer una aplicació mòbil per a facilitar la dinàmica del joc. Aquesta aplicació ha de permetre les funcionalitats que s'especifiquen a continuació.

- Portar un recompte dels diners que tenen els jugadors durant tot el transcurs de la partida.
- Mostrar a cada jugador quins elements (armes, objectes, llibres, etc.) tenen en la seva possessió a cada moment.
- Mostrar a cada jugador les seves estadístiques actuals en tot moment.
- Poder mostrar les dades del joc (regions, bestiar, armes, etc.) d'una forma més amigable.
- Poder mostrar imatges sobre allò que està succeint al joc segons el criteri del Màster.
- Poder escollir un líder dintre del grup que s'encarregui de diferents feines dintre durant el transcurs de la partida. Com ara el manteniment dels diners, cartografiar masmorres, etc.

Amb aquesta aplicació mòbil se solucionarien molts dels problemes esmentats al principi, però encara quedarien alguns. Com ara fer-ho perquè tots els jugadors del grup estiguin connectats entre ell per tal de poder passar-se informació sobre l'estat de la partida. O com fer-ho per a gestionar la gran quantitat de dades i imatges que té el joc.

Amb l'objectiu de solucionar els problemes mencionats al paràgraf anterior s'ha pensat utilitzar una ESP32 com a punt d'accés d'una xarxa wifi privada. D'aquesta manera el servidor estaria allotjat en l'ESP32 i aquest també tindria accés a una base de dades a on es guardarien totes les dades del joc. L'aplicació i l'ESP32 es comunicarien mitjançant una API i amb el protocol HTTP.

La memòria d'aquest treball està dividida en dos grans blocs. Aquests vindrien a ser el *Manual tècnic* on s'explica com està fet aquest treball, quines tecnologies s'han utilitzat i perquè s'ha decidit utilitzar aquestes i no d'altres. I el *Manual d'usuari* on es detalla el funcionament del treball des del punt de vista de l'usuari final, és a dir, on es detalla com l'usuari final hauria d'emprar l'aplicació per a fer un bon ús d'aquesta.

## 2. Antecedents

Aquest projecte parteix amb el joc de rol "*La Llave del Horizonte*" com a base. Abans de començar aquest projecte el desenvolupament d'aquest joc estava bastant avançat. Així i tot, les dades i informació del joc estaven, majoritàriament, fetes a paper i llapis, no existia cap document on es recopilassin les normes del joc, i tampoc hi havia cap document o similar on s'aclarissin les dades del joc. Com ara les espècies dels personatges, la llista d'armes i objectes, una llista d'enemics, animals i bèsties amb les quals et pots trobar al llarg del joc, etc.

Respecte a les tecnologies utilitzades cal mencionar que l'autora d'aquest treball ja tenia una mínima experiència amb algunes d'elles utilitzades, però d'altres venen de noves. Més concretament l'autora d'aquest projecte ja tenia experiència amb bases de dades i havia tocat algun ESP32 abans de fer aquest treball. Per contra l'autora mai havia fet una aplicació per a mòbil, tocat l'*Android Studio* o programat amb Java. És per això que per aquesta part del treball l'autora ha pogut comptar amb l'ajuda d'un amic que ja havia tocat una mica tota aquesta part.



## 3. Manual Tècnic

Aquest treball ha estat dividit en 2 grans blocs per al seu desenvolupament. Aquests serien el bloc del servidor i la base de dades, i el bloc de l'aplicació. En el primer bloc es tracten els aspectes de com fer la comunicació entre els usuaris, el tractament de totes les dades del joc i com comunicar-les de forma simple i clara. En el segon bloc es tracten més els aspectes de com presentar totes les dades i funcionalitats que ofereix el servidor a l'usuari el més clarament i senzill possible. Per tal que l'usuari pugui jugar al joc amb el mínim de dificultats i el màxim de comoditats possibles.

El primer bloc ha estat fet íntegrament per l'autora d'aquest treball. Amb la col·laboració del creador del joc per a poder saber i aclarir totes les dades del joc. Per altra banda, per al desenvolupament del segon bloc l'autora ha pogut comptar amb l'ajuda d'un amic. El qual compta amb un major coneixement i experiència en les tecnologies utilitzades per al desenvolupament d'aquest bloc.

Cap al final del treball s'espera que el primer bloc acabi resultant en un aparell amb les funcionalitats mencionades. I una aplicació mòbil amb les funcionalitats mencionades en el segon bloc. S'esperaria que perquè un grup de gent, d'entre 4 i 8 persones, puguin jugar. Aquests necessitarien 1 aparell, el resultant del primer bloc, per al grup. I una aplicació mòbil, el resultat del segon bloc, per persona.

En relació amb la memòria, el treball a estat dividit en 4 seccions. Una per explicar les tecnologies usades en els dos blocs. Dues seccions per al primer bloc, una per l'explicació de la base de dades i una altra per a parlar sobre el servidor i temes de connexió. I una última una per al segon bloc, per a contar sobre l'aplicació des d'un punt de vista més tècnic que en el *Manual d'usuari* al capítol 4.

### 3.1. Tecnologies utilitzades

En aquesta secció anem a parlar de quines són les tecnologies que s'han utilitzat al llarg d'aquest treball. El motiu pel qual s'han escollit aquestes i no unes altres. I quines són les seves funcionalitats dintre del treball.

Per solucionar el problema de comunicació, al principi de tot, es va pensar a fer una xarxa local i privada. A on els usuaris es puguin connectar per a poder jugar. En aquesta xarxa hauria d'haver-hi un servidor i una base de dades que s'encarreguin de guardar i gestionar tant les dades estàtiques del joc com les dades dinàmiques de les partides. Per fer tot això es va plantejar utilitzar una *ESP32* o una *raspberrry*. Després de pensar-ho una mica, assegurar-se que totes les opcions tenen les funcionalitats necessàries per a fer allò que es desitja. I comparar preus entre les dues es va acabar optant per a fer servir una *ESP32*. El raonament usat per fer aquesta elecció va ser que si aquest joc mai sortís a la venda de cara al públic la diferència de preu entre l'*ESP32* i la *raspberrry* farien que el preu final pugues molt en cas d'escollir la segona opció. I que totes dues opcions acaben fent totes les funcions que es necessiten per al treball. El fet que l'autora ja tenia una mica d'experiència a l'hora de programar un *ESP32* i

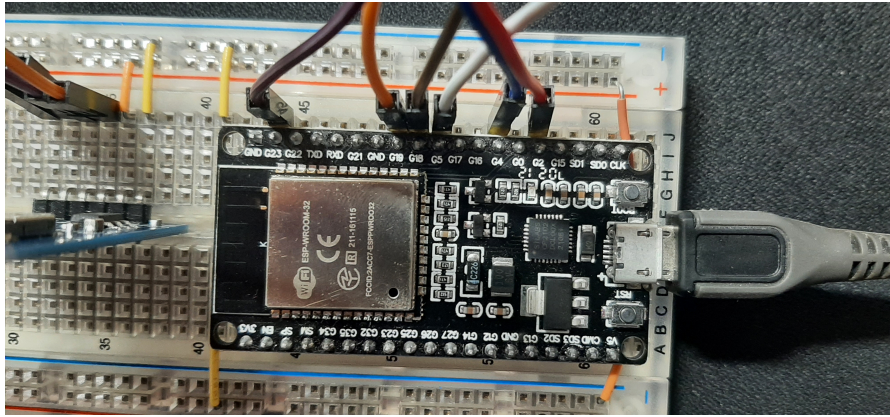


Figura 3.1.: L'ESP32 usada per aquest treball

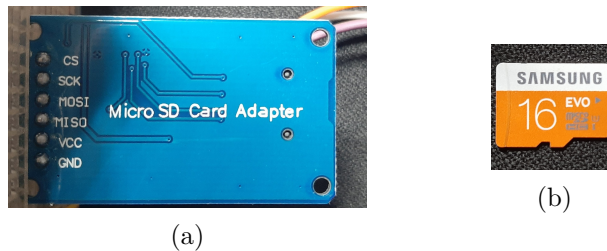


Figura 3.2.: La micro SD i lector corresponent utilitzats per aquest treball

pràcticament cap experiència amb la *raspberrypi* també van influir en la decisió. A la figura 3.1 podem observar l'ESP32 utilitzada durant el desenvolupament d'aquest treball.

Ara que ja tenim l'ESP32 cal pensar en com guardar les dades del joc i a on. Pel fet que s'espera tenir una gran quantitat de dades per aquest joc s'ha acabat decidint de guardar la base de dades i possiblement les imatges relacionades en una micro SD. I utilitzar un lector de la micro SD per a poder llegir les dades des de l'ESP32. La micro SD usada és la que podem veure a la figura 3.2b. Aquesta microSD és una que l'autora ja tenia a casa i és de 16GB. A la figura 3.2a podem observar la part de darrere del lector de targetes emprat durant el desenvolupament d'aquest treball.

Per a programar l'aplicació es va pensar una mica quines tecnologies web utilitza. Si bé fer-ho amb *HTML* o *Android Studio*, però al final es va acabar decidint d'usar *Android Studio* sense gaire discussió. Això vol dir que aquesta aplicació no estarà disponible per tots aquells usuaris que no tinguin un dispositiu amb el sistema operatiu d'Android. Per a ser més concret aquesta aplicació ha estat programada per a funcionar amb Android 8 i superiors. Així i tot, al final només a estat provada en dispositius amb Android 11 i 12.

## 3.2. Tractament de les dades

En aquesta secció es parla sobre com es tracten les dades del joc i s'explica una mica sobre aquestes dades. Per començar, cal mencionar que hi ha dos tipus de dades que es guarden a la base de dades. Estan les dades sobre el joc en general, és a dir, quins tipus d'armes, bestiar, sobre la màgia, les diferents llengües que hi ha, quins objectes inicials li correspon a cada classe,



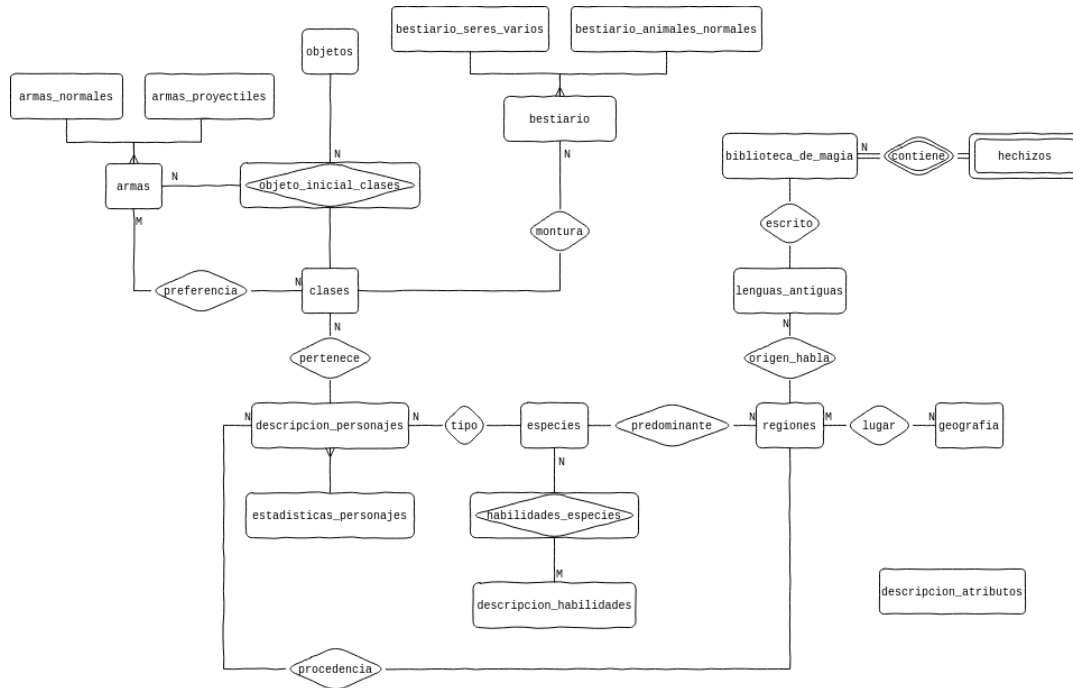


Figura 3.3.: Primer model E/R de la base de dades

les regions i la geografia d'aquesta, els personatges que s'han creat, etc. Aquestes dades serien dades estàtiques que es mantenen d'una partida a una altra. L'altre tipus de dades guardades serien les dades relacionades amb la partida en si. Aquestes dades consistirien en els usuaris que estan connectats/utilitzant l'app, el rol d'aquests usuaris, el personatge de cadascun d'ells, els recursos que tenen, la llengua secundària de qui la tingui, els objectes de grup, etc. Aquestes dades són dinàmiques i van canviant al llarg de la partida. També tenen la peculiaritat de que aquestes s'esborren o reinicien cada cop que s'inicia l'ESP32.

La base de dades està programada amb *SQLite3* i està dividida en 2 fitxers. El fitxer `main_info.sql` que conté totes les taules amb la informació estàtica del joc. I el fitxer `main_partida.sql` per a totes les taules que contindran la informació dinàmica de les partides. Tal com està la BD en l'actualitat, no es pot guardar la informació d'una partida per a reprendre-la més tard. Tot i que no seria gaire difícil modificar-la per a poder acomodar aquesta funcionalitat. El motiu pel qual no s'ha fet és que fer-ho és una cosa que, tot i no ser complexa, porta temps. I es va pensar que aquest temps estava millor aprofitat en intentar fer que l'aplicació tingues les funcionalitats bàsiques abans de posar-hi funcionalitats extres.

Al començament del desenvolupament d'aquesta base de dades es van concentrar els esforços a recollir totes les dades estàtiques del joc. Aquestes van ser demanades a l'autor del joc que les va anar passant a diferents documents de text. L'autora d'aquest treball va haver d'anar llegint aquests documents i organitzant les dades en diferents taules. Això va continuar així fins que es van aconseguir recopilar totes les dades del joc que hi havia en aquell moment. Arribats a aquest punt l'autora va procedir a fer un primer model E/R de la base de dades. A la figura 3.3 es pot veure com va quedar aquest primer model.

A mesura que el temps anava passant, l'aplicació s'anava avançant, es feien reunions amb l'autor del joc i s'anava provant, jugant, el joc en si. Les dades estàtiques del joc van anar



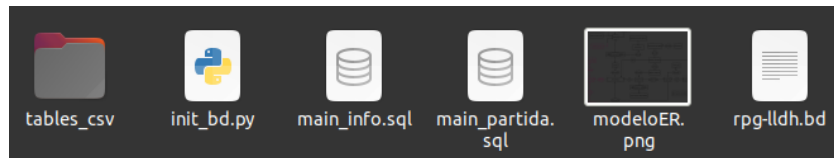


Figura 3.5.: Fitxers guardats dintre de la micro SD

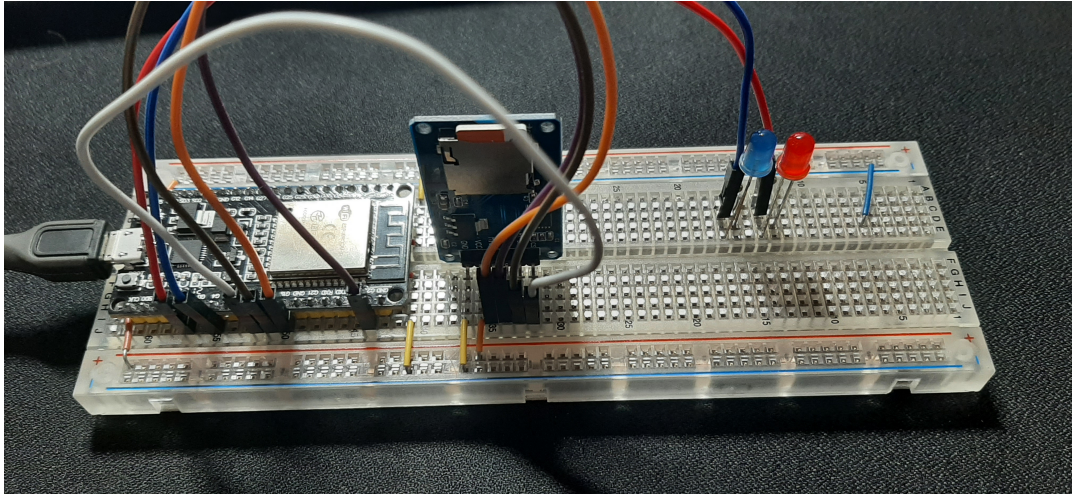


Figura 3.6.: Prototip de l'aparell on està el servidor

I no pas demanar les imatges al servidor cada cop que aquesta fos necessària. Identificar les diferents imatges mitjançant el número identificador que ja mencionat.

### 3.3. Servidor

En aquesta secció es parla sobre el servidor, les seves funcionalitats. També es parla sobre com s'ha fet per a connectar a tot el grup per a poder passar-se dades entre si. Però primer de tot caldria mencionar com ha quedat el prototip d'allò que seria el servidor. A la figura 3.6 es pot veure una foto del resultat final. I a la taula 3.1 es pot veure com van les connexions als ports de l'ESP32. Cadascun dels 2 LEDs està connectat a una resistència de  $1K\Omega$ .

Al començament de tot es va alimentar el lector de targetes a  $3,3V$ . Però en fer proves es va veure que hi havia un error i no s'aconseguia llegir res de la targeta. Després d'investigar sobre aquest error i provar si a canviar la connexió del lector amb l'ESP32 un parell de cops. Es va acabar determinant que el motiu pel qual no es podia llegir la targeta era a causa de l'alimentació i no pas per la connexió. En descobrir això i començar a alimentar el lector a  $5V$ , en comptes dels  $3,3V$  d'abans, es va solucionar el problema. I amb això solucionat ja es va poder llegir i guardar dades a la targeta des de l'ESP32. Cal mencionar que l'ESP32 ha estat alimentat des del port *USB* del ordinador durant tot el desenvolupament i les proves que s'han fet.

Amb les connexions del ESP32 ja explicades anem a parlar sobre quin llenguatge s'ha utilitzat per a programar-lo i les llibreries usades. Al principi de tot es va intentar programar-lo amb *micropython*. Es va poder instal·lar bé i fer algunes proves amb l'ESP32. El problema va arribar quan es va intentar comunicar l'ESP32 amb la base de dades. Aquí l'autora es va

<i>ESP32</i>	Lector <i>micro SD</i>	<i>LEDs</i>
GPIO 5	CS	
GPIO 18	SCK	
GPIO 19	MISO	
GPIO 23	MOSI	
V5	VCC	
GND	GND	
GND		vermell <sup>-</sup> , blau <sup>-</sup>
GPIO 2		vermell <sup>+</sup>
GPIO 4		blau <sup>+</sup>

Taula 3.1.: Conexions ports ESP32

trobar que era necessària alguna llibreria per poder fer comunicar-se amb la BD i extreure dades d'aquesta. Després de buscar llibreries per aconseguir això. D'instal·lar i desinstal·lar el programari necessari perquè l'*ESP32* funcioni amb *micropython* uns quants cops. Al final es va descobrir una llibreria que semblava que anava a funcionar. Però es van acabar descobrint certes dificultats a l'hora d'instal·lar-la a causa del que tot apuntaven a ser qüestions de falta d'espai.

Enfront d'aquestes dificultats i sense veure com poder solucionar-ho es va acabar decidint a provar de programar l'*ESP32* en un altre llenguatge. Es va estar pensant en quin llenguatge seria millor de després del fracàs en intentar amb *micropython*. Les opcions estaven entre programar-ho en *C* amb el plugin d'*Eclipse* o amb *Arduino IDE*. Al principi es va provar de fer en *C*, però al final es va acabar decantant per programar-ho amb *Arduino IDE*. Això és a causa de la senzillesa del codi comparat amb el codi de *C*. I de la facilitat de trobar documentació i ajuda en fòrums[22b] per a programar l'*ESP32* en *Arduino IDE* en comparació amb l'altre llenguatge. En qualsevol dels dos casos es va haver d'instalar Espressif a l'*ESP32* [22c].

En aquest cas si que va ser possible trobar una llibreria per a la comunicació entre l'*ESP32* i la base de dades. Aquesta llibreria va ser trobada al *GitHub* publicada per l'usuari “*siara-cc*” amb el nom de projecte “*esp32-idf-sqlite3*” [sia19]. A part d'aquesta llibreria també s'han utilitzat les següents llibreries.

- `stdio.h` i `string.h`
- `SPI.h`, `FS.h` i `SD.h` per a la comunicació amb la targeta *micro SD*.
- `sqlite3.h`[sia19][Bod22] per a la comunicació amb la BD.
- `ArduinoJson.h`[Ben] per a la lectura i creació de JSON per enviar dades pel protocol HTTP.
- `WiFi.h` i `WebServer.h` per a crear una xarxa wifi amb l'*ESP32* com a punt d'accés a on està allotjat un servidor wifi.

Anem a parlar sobre la qüestió de com aconseguir a tot el grup d'usuaris de manera puguin passar-se dades del joc entre ells. S'ha de dir que a aquesta qüestió se li van donar moles voltes a l'hora de plantejar una solució. Es va mirar de si era millor fer-ho amb Bluetooth, wifi o una barreja de les dues. Si la xarxa wifi hauria d'estar connectada a internet o no, etc. La

qüestió de si utilitzar wifi o no es va solucionar ràpidament. Però un cop ja es va decidir fer ús d'una xarxa wifi, quedava la qüestió de si connectar-se a una ja existent, o crear una amb l'*ESP32*. Per culpa de les dificultats per a connectar-se a una portava, es va acabar decidint que l'*ESP32* faria de punt d'accés. Connectar-se a una xarxa ja existent portava les dificultats de com dir-li a l'*ESP32* a quina xarxa s'havia de connectar sense tenir una ja prefixada al codi. Això es volia fer així per a poder jugar al joc independentment d'on o amb qui s'estigui. I d'aquesta manera no es pot assegurar que la mateixa xarxa wifi estarà disponible. Amb aquestes dificultats superades ara quedava la qüestió de si obrir la xarxa a Internet o no. Al final es va decidir que la xarxa en si ja tenia més que suficient gestionant totes les dades del joc. I obrir-la a internet només faria que aquesta hagués de gestionar més requests i fer-la així més lenta.

Un cop ja està clar amb quin llenguatge i que es vol fer només queda posar-se a programar. El codi de l'*ESP32* ha passat per diferents fases o versions, 4 per ser més exactes. La primera de totes es va intentar fer amb una llibreria que permetia manejar els GETs i POSTs de forma asíncrona. Però no acabava d'anar bé en intentar agafar dades de la base de dades. És per això que es va acabar fent de forma síncrona en les 3 futures versions. En la següent versió es va fer una handler per a agafar dades de cada una de les taules estàtiques. Però no es va diferenciar si eren entre GETs, POSTs i altres mètodes. A la següent versió es va agafar la versió anterior i es va millorar. Les urls que acceptava el servidor es van endreçar una mica i es van diferenciar els handler amb les accions GET, POST, PUT i DELETE.

Per acabar arribem a la versió final del servidor a l'*ESP32*. Aquesta versió torna entrarà una versió en el sentit de què totes els URLs que te les tracta com si fossin GETs. Això és causat pel fet que no ha estat possible esbrinar com enviar peticions HTTP que no siguin GET des de l'aplicació mòbil.

A part d'això els URL estan totes millor ordenades i clares per entendre millor que és el que fan. Per exemple, totes les URLs comencen de la següent forma `http://192.168.4.1/rpg`. Després d'això va el mètode GET, POST, PUT o DELETE en funció del que es vulgui fer. I després ja va més dirigit al tipus d'acció o informació que es vol de la BD.

Els codis que pot arribar a retornar el servidor són el 404 i un JSON indicant que aquest URL no és gestionada pel servidor. Un 204 i el text "204 OK" per indicar que tot ha anat bé, però aquest URL no retorna res. Un 200 i un JSON amb les dades demanades de la base de dades. Un 500 i un JSON on s'indica que hi ha hagut un problema amb la base de dades i quin problema. I per últim, un 400 i un JSON indicant que l'URL té un argument incorrecte.

També estaria bé mencionar el fet que el codi d'aquesta versió està separat en tres moduls. El modul principal, pel nom `webserverAP_2.1_`, on hi han les funcions `setup` i `loop` a més a més de altres funcions comunes per a tots el moduls. Els altres dos moduls van pel nom `info_uri` i `partida_uri` i s'encarreguen de les funcions de les peticions url de les dades estàtiques del joc i les dinàmiques de la partida respectivament.

En tema de concurrència i com gestiona les peticions entrants aquest servidor. Cal dir que aquest va agafant les peticions segons li van arribant i crea una cua amb el protocol FIFO. O almenys aquest és el comportament observat, ja que al codi no s'ha especificat quin protocol seguir. Sí que és veritat que com que no s'ha especificat res diferent l'*ESP32* només gestiona una petició a la vegada, és a dir, no és capaç de gestionar més d'una petició de forma simultània. Això pot acabar provocant un embut quan hi ha moltes persones connectades a la xarxa. I si la majoria de les peticions ja són lentes de per si això pot provocar que el temps de resposta a una petició de l'aplicació sigui encara més lent. També cal mencionar que el màxim d'usuaris que pot gestionar aquesta xarxa és de 8.

Ja per acabar amb el servidor cal parlar sobre com s'ha decidit tractar la seguretat de les dades. Val a dir que totes les comunicacions s'estan fent en una xarxa local i privada sense cap accés a Internet. És per això que no se li ha donat una gran importància a intentar transmetre les dades pel protocol HTTPS en comptes del protocol HTTP. I en qüestió de les dades guardades a la targeta micro SD ja s'ha mencionat amb anterioritat que si algú desitges tocar-les. Aquesta persona haurà de fer-ho sota la seva responsabilitat. També estaria bé dir que el que es pot veure a la figura 3.6 és un prototip. I que en la versió final d'aquest prototip es pot fer perquè no hi hagi un accés fàcil a la targeta per tal d'evitar que els usuaris toquin les dades del seu interior.

### 3.4. Aplicació

En aquesta secció anem a parlar sobre l'aplicació en si. Com s'ha fet i alguns altres aspectes tècnics d'aquesta. Per a saber més sobre l'aplicació, però des del punt de visat de l'usuari ves al capítol 4 a on podràs trobar el *Manual d'usuari*.

Com ja s'ha comentat amb anterioritat al llarg d'aquest document escrit. El propòsit d'aquesta aplicació és facilitar el joc proveït una plataforma a on es mostrin totes les dades d'aquest de forma senzilla i fàcil d'entendre. També ajuda a diferenciar entre els rols de cada usuari a dintre del joc i proporcionant-li a cadascú la informació que li correspongui segons aquest rol. És a dir, el màster pot veure tots els elements del joc. Però pel contrari, un jugador només pot veure quins són els seus recursos, l'estat dels recursos del grup, la informació del seu personatge i una foto escollida pel màster.

Per començar cal mencionar que l'aplicació ha estat programada amb Java. Per ser més concrets amb la versió *openJDK 11.0.15*. I s'ha programat en l'*Android Studio*, a la versió *Android Studio Chipmunk | 2021.2.1*. L'aplicació ha estat programada per a ser compatible amb dispositius que tinguin un sistema operatiu d'Android 8 o superior. S'ha escollit aquesta versió perquè el màxim de gent possible pugui fer ús d'aquesta aplicació. A condició que tinguin un dispositiu amb un sistema operatiu Android.

Una de les primeres coses que es van haver de decidir amb relació a l'aplicació va ser la paleta de colors que utilitzaríem per a aquesta. Per fer aquesta decisió va ser possible comptar amb l'ajuda d'una pàgina web que crea paletes de colors per a Android Studio o altres [22d]. En crear la paleta de colors d'aquesta manera ens podem assegurar la barreja de colors feta servir no faria mal als ulls a l'hora d'usar l'aplicació. Cal dir que es van fer un parell o tres de proves amb diferents paletes de colors per mirar com es veien. Al final es va acabar decidint per fer ús una paleta de colors verds. Perquè es va pensar que quedaria bé amb el tema del joc i donar com una sensació d'exploració de boscs. En resum que es va pensar que aquesta paleta era la que quedava millor dintre de les que es van provar.

L'aplicació compta amb un total de 7 pantalles diferents. A la figura 3.6 es pot observar com van les transició entre elles. A continuació hi ha una llista on s'explic una mica sobre cada una d'aquestes seves pantalles.

- La pantall `MainActivity` té el layout a `activity_main.xml`. Aquesta pantalla és una pantalla de càrrega que s'encarrega de comunicar-li al servidor que hi ha un usuari nou al joc. També s'encarrega de demanar dades estàtiques al servidor. Això es fa aquí perquè són peticions lentes i tot i que d'aquesta forma s'alenteixi l'inici del joc. Un cop ja es comenci a jugar la cosa duria d'anar una mica més ràpida, ja que aquestes dades ja les tindrem.

- La pantall `EleccionRol` té el layout a `activity_eleccion_rol.xml`. Aquesta pantalla serveix perquè l'usuari pugui escollir quin rol farà a la partida i d'aquesta manera dirigir-lo a una pantalla o altra. En cas d'escollir el rol de Màster primer es farà una consulta al servidor per veure si ja hi ha màster o no. En cas que no hi hagi es passarà a la següent pantalla del màster sense cap problema. En el cas contrari, simplement es passarà a la següent pantalla de jugador. Seria om si l'usuari hagués decidit el rol de jugador des d'un inici.
- La pantall `MasterEscojeLider` té el layout a `activity_master_escoger_lider.xml`. Aquesta pantalla seria la primera que li apareixeria a un usuari amb el rol de màster. Aquesta s'encarrega de mirar quins jugadors hi ha connectats al servidor i presentar-se'ls a l'usuari perquè aquest pugui escollir un líder. Val a dir que no va ser possible fer que l'aplicació comproves quins usuaris hi han connectats de forma periòdica. Davant d'aquest problema es va decidir crear un botó que fes la petició aquesta. Així l'usuari pot recarregar la llista de gent connectada al servidor quan vulgui.
- La pantall `PaginaControlMaster` té el layout a `activity_pagina_control_master.xml`. Aquesta pantalla és l'encarregada de gestionar totes les necessitats de l'usuari amb el rol de màster.
- La pantall `EscojerPersonaje` té el layout a `activity_escojer_personaje.xml`. Aquesta pantalla serà la primera que li apareixerà a tots aquells usuaris amb el rol de jugador. Aquesta serveix perquè els jugadors puguin decidir quin personatge volen ser durant la resta de la partida i de comunicar aquesta decisió al servidor.
- La pantall `CrearPersonajeNuevo` té el layout a `activity_personaje_nuevo_by_ivan.xml`. Aquesta pantalla només li apareixerà a aquells usuaris que hagin decidit crear un personatge nou. I és l'encarregada de mostrar tots els camps necessaris per a crear-ne un i assegurar-se que estan ben omplerts abans de deixar que l'usuari guardi les dades d'aquest personatge.
- La pantall `PaginaPrincipal` té el layout a `activity_pagina_principal.xml`. Aquesta pantalla és l'encarregada de gestionar totes les necessitats de l'usuari amb el rol de jugador.

A mesura que es va anant desenvolupant l'aplicació es van anant fent testos per a assegurar-se que allò que s'ha programat realment funcionava. I que funcionava tal com s'esperava. Aquests testos es van anar fent amb dos telèfons mòbils i un emulador d'*Android Estudio* amb un sistema operatiu d'Android 11. Ja cap al final del treball es va poder fer un test del sistema amb 2 o 3 mòbils i un emulador d'*Android Estudio*. Tots els aparells usats per aquest test tenien un sistema operatiu d'Android 11, amb l'excepció d'un aparell que tenia el sistema operatiu d'Android 12. Amb aquests testos també es va poder mirar com quedava l'aplicació per a pantalles de diferents mides.

Durant el desenvolupament de l'aplicació han anat sortint molts errors i problemes. I tot i que ha sigut possible solucionar la majoria hi ha hagut alguns de persistents. Un d'aquests problemes seria la velocitat de la connexió entre l'aplicació i el servidor. Els encarregats de programar l'aplicació han pogut observar que totes aquelles connexions que requereixen dades de la base de dades són especialment lentes. Això és per culpa del fet que la base de dades està en una memòria externa a la de l'*ESP32* i, per tant, la lectura d'aquestes dades és lenta.

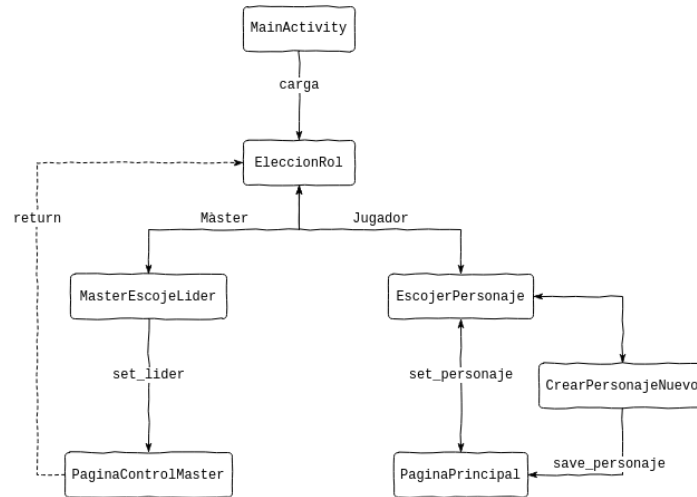


Figura 3.7.: Transició de pantalles

Una possible solució que s'ha pensat seria modificar el servidor i canviar l'*ESP32* per un altre processador amb més potencia, com seria una *raspberrry*.

Un altre problema persistent es trobaria a la pantalla **PaginaControlMaster**. Aquí el problema hi seria al panell per a modificar les estadístiques dels jugadors. La cosa seria que en obrir-lo i clicar a sobre d'alguna de les estadístiques d'aquestes per a modificar-les no s'obre el teclat. Cosa que hauria de passar automàticament i l'autora i el seu amic no acaben d'entendre per què passa això. De forma que ha estat impossible solucionar aquest problema.

Un últim problema que no s'ha pogut solucionar a temps és a la pàgina principal dels jugadors. Al *Manual d'usuari* es menciona que en clicar a sobre d'alguns dels 4 recursos que té el jugador, els que estan a la part de sota de la pantalla. S'ha d'obrir un panell amb la descripció d'aquest recurs. La qüestió és que en clicar no s'obre cap panell. Aquest és un problema al qual encara se li està buscant solució.



## 4. Manual d'usuari

Per poder jugar al joc "*La Llave del Horizonte*" amb aquesta aplicació cal tenir l'*ESP32* encesa i funcional. En encendre l'*ESP32* s'encendra un led blau que indica que el servidor s'està posant a punt per a poder jugar. Un cop el led blau s'apagui cal connectar-se a la xarxa wifi *rpg-wifiAP* amb contrasenya *01234567* per a poder jugar. És important mencionar que aquesta xarxa no està connectada a Internet i, per tant, l'aparell on estigui l'aplicació pot demanar confirmació de què realment vols connectar-te a aquesta xarxa.

En obrir l'aplicació la primera pantalla amb la qual es trobarà l'usuari és la pantalla de càrrega mostrada a la figura 4.1. Es romandrà en aquesta pantalla fins que l'aplicació detecti que l'usuari està connectat a la xarxa wifi mencionada al paràgraf anterior i connecti amb el servidor. Aquesta acció pot trigar una mica, cal tenir paciència.

Un cop l'aplicació detecti que l'usuari està connectat a la xarxa wifi indicada passarà a la següent pantalla de forma automàtica. Aquesta pantalla serveix perquè l'usuari esculli quin rol vol tenir duran la partida. Els dos rols possibles són *Màster* i *Jugador* com podem observar a la figura 4.2. Una cosa important a saber sobre el rol de *Màster* és que només pot haver-hi un per partida. Això vol dir que si l'aplicació detecta que ja hi ha alguna persona del grup que ja ha escollit aquest rol no li estarà permès a ningú més escollir aquest rol. I per qui intenti escollir el rol de *Màster* després d'haver-se establert 1 *Màster* serà tractat com si hagués escollit el rol de jugador. Un altre cop, aquesta acció pot trigar una mica, cal tenir paciència.

En cas d'haver escollit el rol de *Màster* ves-hi a la secció 4.1 per a saber quins són els següents passos amb els quals et trobaràs. Al contrari, si has escollit, o t'ha tocat, el rol de jugador ves a la secció 4.2 per a saber quins seran els següents passos. Sigui quin sigui el rol que has escollit es recomana que li donis una ullada a la secció 4.4, per saber més sobre alguns detalls de l'aplicació i com actua enfront de certes situacions. També seria interessant llegir les normes del joc que es poden aconseguir a l'annex A per poder entendre millor com va el joc i el perquè d'algunes coses de l'aplicació.

### 4.1. Master

Felicitats ja ets el *Màster* d'aquesta partida. Com a màster ets l'encarregat d'explicar la història i portar la partida, tingues en compte que això és una gran responsabilitat i pot arribar a ser difícil i frustrant en certs moments. A la vegada pot jugar amb el coneixement que ets l'únic que té tots els coneixements del terreny i esdeveniments importants de la partida.

Però, abans de poder començar amb la partida com a *Màster* que ets has d'escollir qui serà el *Líder* d'aquest grup. Per a poder fer això la primera pantalla que t'apareixerà després d'entrar amb aquest rol serà la mostrada a la figura 4.3. En aquesta figura podem observar que hi ha tres usuaris connectats, el màxim d'usuaris connectats seria de 8. Podem veure com un usuari és el *Màster*, l'altre usuari va pel nom de Marea i hi ha un tercer usuari *NULL*. Quan veiem un usuari amb el nom *NULL* vol dir que aquest usuari està connectat a la xarxa, però encara no ha escollit un Personatge. En aquesta figura també podem veure com en aquest instant

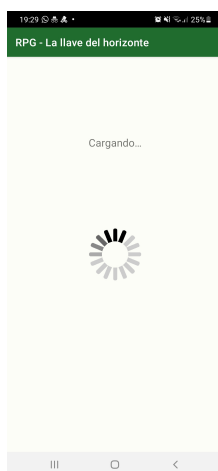


Figura 4.1.: Primera pàgina de l'aplicació

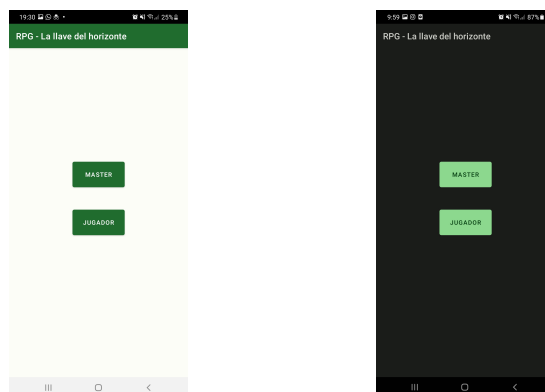


Figura 4.2.: Pagina per escollir el teu rol

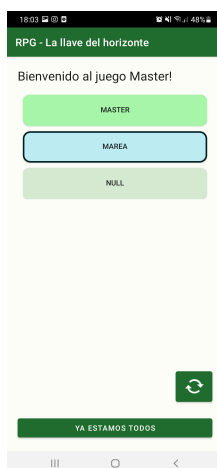
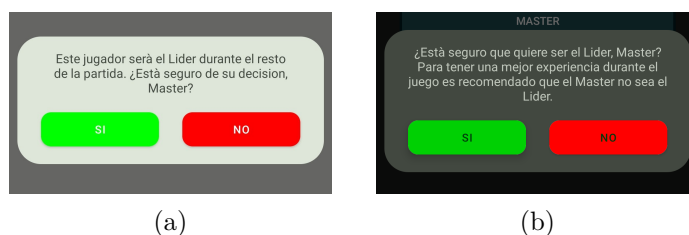
Figura 4.3.: Elecció del *Líder* del grup

Figura 4.4.: Panell de confirmació del líder

l'usuari pel nom Marea és el que està seleccionat per a ser *Líder*. Un cop s'hagi escollit el *Líder* desitjat cal indicar-ho clicant el botó de sota del tot, on posa "YA ESTAMOS TODOS". Una cosa a tenir en compte és que l'aplicació no actualitza la llista d'usuaris connectats de forma automàtica. Per tant, quan el *Màster* vulgui actualitzar aquesta llista haurà de clicar al botó quadrat de la part inferior esquerra de la pantalla. S'ha de tenir en compte que aquesta opció pot ser lenta i, en conseqüència, es recomanaria esperar que tothom hagi escollit el seu personatge abans de clicar aquest botó i després escollir el *Líder* desitjat. Per a saber més sobre les funcions d'aquell qui sigui escollit com a *Líder* mira la secció 4.3.

Un cop escollit el líder et sortirà un panell de confirmació com el que es pot veure a la figura 4.4. Aquest panell de confirmació pot variar en funció de qui s'hagi escollit com a líder del grup. Si el que ha sortit escollit com a *Líder* és el *Màster* llavors et sortia el panell de confirmació de la figura 4.4b. En cas que sigui qualsevol altra persona sortirà el panell de control mostrat a la figura 4.4a. Això passa pel fet que a causa de certes funcions del *Líder* es recomana que aquest sigui un jugador normal i no pas el *Màster*. Així i tot, l'usuari denominat com a *Màster* es pot escollir a si mateix per a cobrir la funció de *Líder*. En cas d'acceptar el panell de confirmació ja es passaria a la pàgina principal de l'usuari amb aquest rol. Igual que en casos anteriors aquesta acció pot trigar una mica, cal tenir paciència.

La pàgina de control, també anomenada pàgina principal del màster, és la que es mostra a la figura 4.5. Aquesta pantalla es podria dividir en tres parts, la de dalt de tot de supervisió de la partida, la part del mig de llistat i assignació de recursos i la part de baix de control de personatges, és a dir, jugadors. La part de supervisió de la partida consisteix en un visualitzador de diners, dos botons verds petits i una imatge en blanc (groc) petita. Aquesta part com el

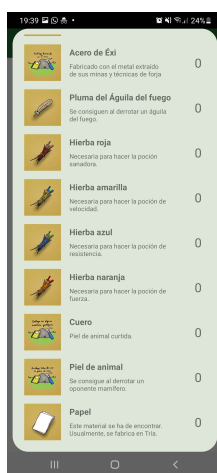
Figura 4.5.: Pàgina principal del *Master*

Figura 4.6.: Llista d'objectes del grup

seu nom indica serveix per a poder veure que com van alguns elements de la partida en tot moment. El boto verd de mes a dalt i petit serveix per a anar actualitzant el visualitzador de diners per així saber com van els diners del grup i poder actuar en conseqüència, si així o desitja el *Màster*. L'altra boto verd d'una mica més avall amb la icona d'un paper i una lupa obre una llista com la mostrada a la figura 4.6 on es mostren tots els objectes del grup i quina quantitat hi ha de cada cosa. Aquests objectes són uns que s'adquireixen com a grup, poden ser utilitzats per qualsevol integrant d'aquest i no ocupen espai per guardar altres objectes i armes a l'emmagatzematge de cada jugador individual. I per últim la imatge en blanc serveix per a veure en tot moment quina imatge estan veient els jugadors, és a dir, si el màster decideix mostrar la imatge d'una regió perquè els usuaris sàpiguen a on han arribat el màster podrà veure-ho reflectit en aquesta part de la seva pantalla. Aquesta imatge te'l valor afegit del fet que si l'usuari clica a sobre li apareixerà un panell amb la descripció d'allò que s'estigui mostrant.

La part designada al llistat i assignació de recursos consisteix en 5 botons verds grans. Un per cada un dels 5 grups de recursos que hi ha al joc. Aquests recursos són diferents entre si

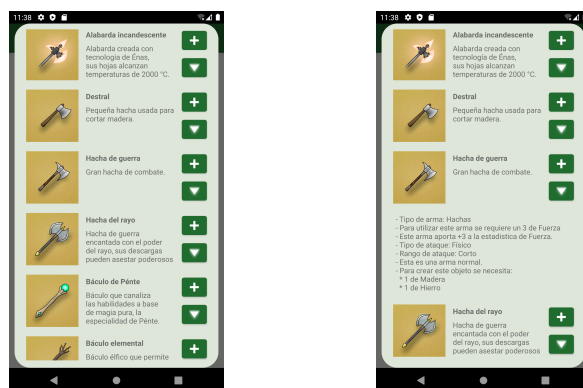


Figura 4.7.: Panell d'informació de recursos (les armes en aquest cas)

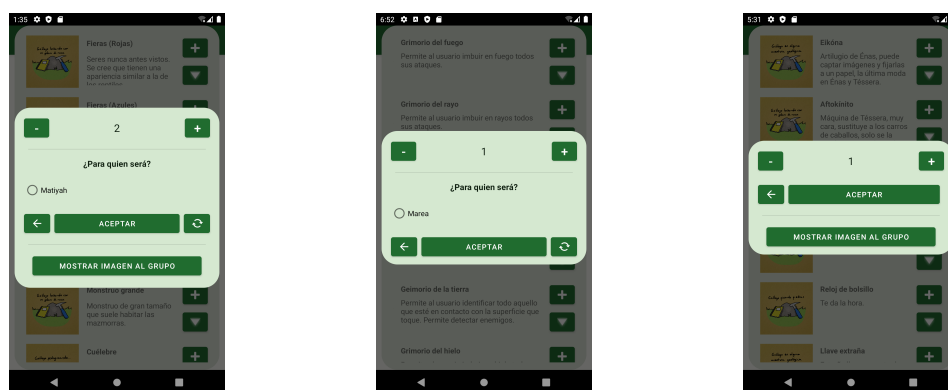


Figura 4.8.: Pantalla d'assignació de recursos

i es poden utilitzar de formes diferents al llarg del joc. En clicar cadascun d'aquests botons s'obrirà un llistat similar al de la figura 4.7 amb tots els recursos d'aquell grup. Com es pot veure així de primeres cada recurs de la llista conté: una imatge un nom i una petita descripció acompanyada per dos botons verds petits a l'esquerra. El boto amb una fletxa indicant avall té la funcionalitat de desplegar una descripció més àmplia de cadascun d'aquests recursos en ser clicat. El botó del mes és una mica més complex, en el cas que en clicar-ho no sembli que hagi succeït res és que aquest recurs en particular no es pot assignar a ningú i, per tant, la seva foto ja es mostrarà a la pantalla de tots els jugadors. Però això només acostuma a passar en el cas de les Regions i algunes Besties. Per contra, si en clicar el botó surt una altra pantalla més petita similar a la de la figura 4.8 és que aquest recurs pot ser assignat a algun jugador a més a més de ser visualitzat per tothom. Hi ha alguns objectes que no poden ser mostrats degut al fet que no tenen imatge com és el cas dels recursos màgics. Hi ha alguns objectes als quals no se'ls permet ser assignats a cap jugador en concret, això significa que aquest és un objecte el qual es pot guardar al dipòsit del grup (el botó del paper i la lupa).

En últim lloc, la part referida com a zona de control de personatges consisteix en dos botons grans amb dues imatges a la part de sota de la pantalla. El botó de l'espasa serveix perquè el màster pugui tenir un recull de quins recursos té cada personatge i poder assignar més o menys d'un recús o treure-li aquest recurs. Segui a petició del jugador o per culpa de com està anant la història de la partida. Cal mencionar que cada jugador pot tenir fins a un màxim de 4 recursos de forma simultània. A la figura 4.9 es pot veure el funcionament d'aquest botó. Per

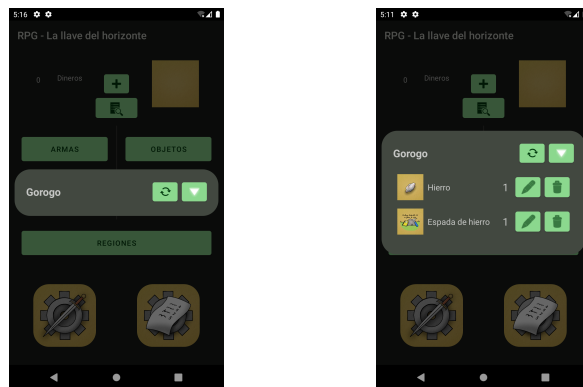


Figura 4.9.: Com modificar els recursos dels jugadors?

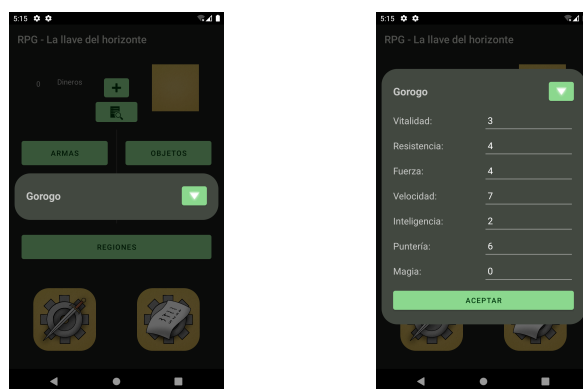


Figura 4.10.: Com modificar les estadístiques dels jugadors?

una altra banda, el boto amb els papers escrits, aporta la funcionalitat de poder visualitzar i modificar les estadístiques dels personatges. El funcionament d'aquest botó s'hi pot veure a la figura 4.9.

## 4.2. Jugador

Benvingut nou jugador! Si en comptes de decidir ser el *Màster* de la partida has escollit ser un jugador el primer de tot que has de fer és escollir amb quin personatge vols jugar durant aquesta partida. Pots escollir un dels personatges que ja et venen fets, un personatge ja haguessis creat per alguna partida anterior o crear un nou personatge des de zero. Per això la primera pantalla que et sortirà si ets un jugador serà la mostrada a la figura 4.11, en aquesta pantalla pots prendre la decisió de quin personatge vols ser o de crear un de nou. En cas que vulguis crear un personatge nou hauries de clicar al botó de dalt a la dreta, l'únic que no té un fons groc. Per saber més sobre els passos a seguir per crear el personatge mira la secció 4.2.1. En cas d'escollir un personatge ja creat clica a sobre de la imatge o el nom del personatge. En qualsevol dels dos casos, cal saber que aquesta l'aplicació pot trigar una mica a reaccionar a aquesta acció, però això no vol dir que l'aplicació no funciona ni que s'ha penjat. Com ja s'ha dit anteriorment cal tenir paciència.

Un cop ja s'hagi escollit el personatge la següent pantalla que veurà el jugador serà la seva pantalla principal on podrà tenir un recull de la informació del seu personatge i dels recursos

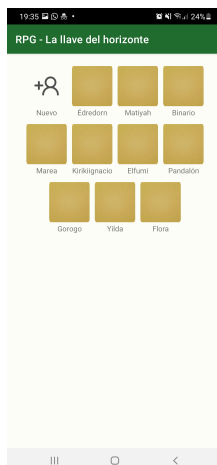


Figura 4.11.: Pantalla per escollir el teu personatge

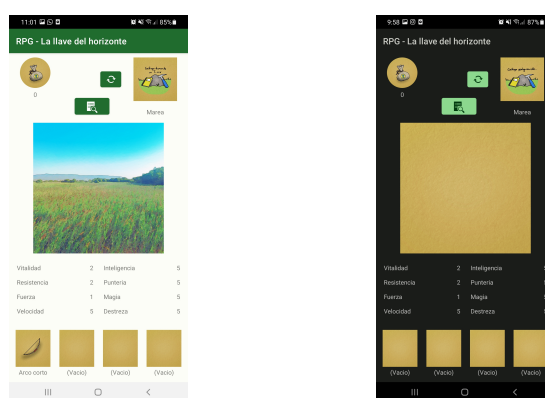


Figura 4.12.: Pàgina principal dels jugadors

que té ell o ella com a jugador i els recursos que té el grup. A la figura 4.12 es pot observar uns exemples de com seria aquesta pantalla. Segons la mida de la pantalla del mòbil l'usuari es pot trobar que no li apareixen les seves estadístiques en aquesta pantalla, en aquest cas, no cal espantar-se, l'usuari podrà veure les seves estadístiques amb la resta de la informació del seu personatge en clicar a la icona d'aquest. A la figura 4.13 es poden veure uns exemples del panell informatiu de les dades del personatge de l'usuari.

Igual que en la pantalla de control del *Màster* la pantalla principal del jugador es pot dividir en tres parts. La primera començant per dalt seria on està la informació de grup, com ara els objectes i diners que té el grup i informació sobre el personatge de l'usuari. La segona part seria la gran imatge del mig que serveix per a visualitzar les imatges que el *Màster* decideixi mostrar als jugadors. I per últim, la part de baix que s'utilitzaria per a poder visualitzar quins recursos té el jugador en cada moment, a més a més, de quines són les estadístiques del seu personatge.

La primera part de totes té un visualitzador de diners indicat per una imatge d'una bossa moneder. Els diners, com qualsevol altra cosa, no s'actualitzen de manera automàtica. Per tant, si un usuari vol actualitzar aquest visualitzador, només cal que cliqui a sobre de la imatge de la bossa. Al costat d'aquest visualitzador podem veure dos botons verds, un serveix per

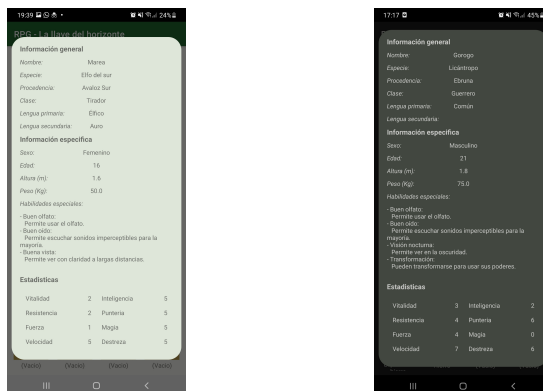


Figura 4.13.: Panell amb tota la informació del personatge

actualitzar totes les dades del jugador, és a dir: diners, la imatge gran, els recursos (tant individuals com de grup) i les estadístiques. Aquesta acció és lenta, per tal cosa, es recomana no abusar d'ella i tenir paciència. L'altre boto verd, el que té un paper amb una lupa, obre una llista d'objectes i la quantitat de cadascun d'ells que té el grup de jugadors. Per acabar, a la dreta del tot hi ha la imatge del personatge i el seu nom. En clicar la imatge del personatge s'obrirà el panell d'informació d'aquest que s'ha comentat en el paràgraf anterior. Com ja s'ha mencionat anteriorment a la figura 4.13 podem observar uns exemples d'aquest panell informatiu.

La segona part de la pantalla, la part del mig, no té molt de misteri. Simplement, és un lloc on poder visualitzar les imatges que el *Màster* anirà mostrant al llarg de la partida per a fer més fàcil la comprensió i seguiment del que està passant. Una cosa que cal mencionar és que en clicar a sobre de la imatge aquesta s'actualitzarà, si el *Màster* l'ha canviat.

Per acabar, la tercera i última part consisteix en dues columnes on es mostren les estadístiques del personatge i quatre quadrats grossos on es mostraran una imatge i el nom dels 4 recursos que té el jugador en cada moment. En clicar a sobre de la imatge d'aquests recursos s'hauria d'obrir un panell amb tota la informació d'aquest recurs en concret.

Segons les normes del joc especificades a l'annex 4.2.1 un jugador pot tenir una o dues llengües en funció de la seva estadística d'intel·ligència. Davant d'aquesta norma quan l'aplicació detecti que un jugador compleix les condicions per aprendre una segona llengua. A aquest li apareixerà un panell com el mostrat a la figura 4.14. Aquest panell pot ser la primera cosa que vegis en entrar com a jugador, si ja compleixes les condicions de base. O quan aconseguis complir les condicions durant el transcurs de la partida.

#### 4.2.1. Crear un personatge

En cas de voler crear un personatge nou aniràs a parar a la pantalla que es mostra a la figura 4.15. En aquesta pantalla es demanen totes les dades necessàries per a crear un nou personatge. Cal emplenar tots els camps disponibles amb l'excepció dels dos que estan amagats, aquests són opcionals. Aquests dos camps es poden descobrir i amagar clicant a sobre de la frase "*Características adicionales*", serveixen per a descriure la forma física i personalitat del personatge. Cal tenir en compte que s'han de seguir les normes especificades a l'annex A.1. Un cop tots els camps estiguin omplerts amb la informació corresponent s'ha de clicar al botó de sota del tot on posa "*GUARDAR PERSONAJE*" per a guardar el personatge. Si està tot bé



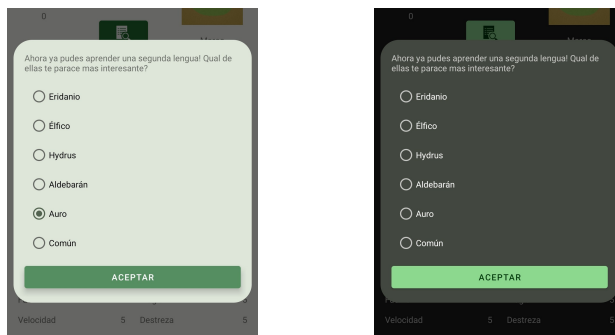


Figura 4.14.: Panell per escollir una segona llengua

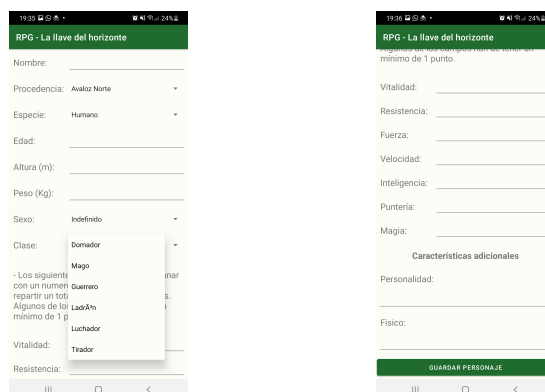


Figura 4.15.: Pantalla per a crear personatges

el personatge es guardarà i podrà ser utilitat en partides futures. En clicar aquest botó cal tenir paciència, ja que com en altres llocs de l'aplicació aquesta acció pot trigar una mica.

### 4.3. Líder

Si has estat escollit com a *Líder* del grup aquí s'expliquen les responsabilitats d'aquest i les visibles diferències que et trobaràs a la pantalla principal. Com a *Líder* seràs l'encarregat de portar el compte dels diners i actualitzar el visualitzador per tal que la resta de companys del grup puguin saber com va de diners el grup. Per a poder fer això la pantalla del líder té un botó amb el signe més a la part de dalt de la pantalla, tal com es pot veure a la figura 4.16. El botó apareixerà en posicions lleugerament diferents segons si ets el *Màster* o un jugador. En clicar aquest botó apareixerà un panell com el de figura 4.17 on es poden ingressar o retirar diners en funció d'allò que estigui passant a la partida.

Una altra responsabilitat del *Líder* és fer de cartògraf allà on ho requereixi el joc i així o indiqui el *Màster*. Aquesta funcionalitat no està implementada dintre de l'aplicació de manera que el *Líder* necessitarà paper i llapis per a poder fer aquesta tasca. Això o una molt bona memòria per a poder saber per on ha passat el grup i que és el que se sap sobre el terreny on estan. Aquesta responsabilitat serà necessària principalment quan el grup arribi a una masmorra, cova o similar. És necessari mencionar que perquè el *Líder* pugui dur a terme aquesta responsabilitat el grup ha de tenir l'objecte de llibreta amb ploma.

Figura 4.16.: Diferències a la pantalla del *Líder*

Figura 4.17.: Pantalla de gestió del diner

## 4.4. Errors comuns

Tenint en compte que aquesta aplicació encara s'està acabant de desenvolupar no es pot garantir que durant l'ús d'aquesta no sorgirà cap error. És per això que en aquest apartat s'expliquen els errors més comuns i com reaccionar enfront d'ells.

El primer error comú amb el qual et trobaràs és que en passar d'una pantalla a una altra la pantalla es quedarà en blanc, o negre, durant un cert període de temps. I, fins i tot, poden sortir missatges d'error com els de la figura 4.18. En cas que passi això només cal esperar i si surt el missatge d'error clicar-li a l'opció d'esperar. El missatge d'error aquest pot arribar a sortir en alguns altres instants, però això no canvia la manera de tractar amb ells. Aquest missatge surt perquè la comunicació amb el servidor és molt lenta i és per això que al telèfon li sembla que l'aplicació no respon.

Un altre error seria el que passa quan s'ha de posar una imatge a la qual l'aplicació no en té accés, sigui perquè encara no existeix o per algun altre motiu. En aquest cas l'aplicació mateixa té unes fotos com les de la figura 4.19 amb les que substituirà la imatge que hauria

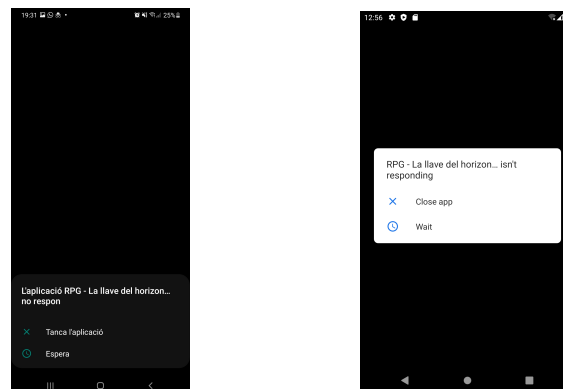


Figura 4.18.: Panell d'error comú



Figura 4.19.: Panell d'error comú

d'anar allà fins al moment en què la imatge correcta estigui disponible. L'elecció de quina d'aquestes imatges d'erro sortirà es fa de forma aleatòria.

Per acabar, aquest error seria més aviat de la part del servidor que no pas de l'aplicació en si. L'aparell on està l'*ESP32* també té dos LEDs, un blau i un vermell. En cas que el LED vermell s'il·luminés o comences a fer pampallugues significarà que ha succeït un error greu al servidor i cal reiniciar-lo per tal de poder jugar amb l'aplicació.



## 5. Conclusions

Ja per acabar s'ha de dir que no s'ha aconseguit que el producte final d'aquest treball estigui a la seva versió final, o versió amb la qual podria ser comercialitzat. En altres paraules, aquesta versió seria una versió beta del producte. L'autora del treball pensa així pel fet que l'aplicació és molt lenta i encara li falten la meitat de les imatges. Això últim és degut al fet que l'amic que ha creat el joc i s'està encarregant de fer els dibuixos no ha tingut temps a fer més.

Com ja s'ha comentat al treball un dels punts que quedaria pendent seria substituir l'*ESP32* per una *raspberry* amb l'objectiu de tenir un servidor amb més potencia. Per tant, es reduiria la velocitat de comunicació i amb això s'aconseguiria augmentar la velocitat de l'aplicació. Fent així una experiència de joc més agradable. Unes altres coses que també estarien pendents de fer serien: poder guardar partida, poder esborrar personatges ja existents (a l'hora d'escollir personatges) i potser afegir uns daus dintre de l'aplicació. Una altra idea seria poder obrir la xarxa wifi ja existent a internet.

També estaria bé mencionar que ha estat possible aconseguir la gran majoria dels objectius de forma satisfactòria. S'ha aconseguit que l'aplicació tingui totes les funcionalitats que es van plantejar al principi de tot. L'únic que queda és que com ja s'ha mencionat molts cops el sistema és molt lent i això fa que la jugabilitat del joc no sigui la millor. També cal mencionar que no ha estat possible provar l'aplicació en una situació de partida real. Però sí que ha estat possible provar-la amb un número de gent similar al que hi podria haver en una partida.

En general l'autora del treball ha quedat satisfeta amb com ha quedat. Fer un treball d'aquesta magnitud a estat una experiència nova i interessant per a l'autora. I el fet de poder solucionar un problema tan proper en l'entorn de l'autora ha estat molt motivador. També s'ha de dir que l'autora ha pogut aprendre molt durant tot el desenvolupament d'aquest treball. I poder veure i provar el producte d'aquest de tan a prop fa que la motivació per a continuar el desenvolupament d'aquest projecte sigui molt gran. En general ha estat una molt bona experiència i amb resultats satisfactoris. Encara que no sigui tot el satisfactori que s'esperava a l'inici del treball.



## Bibliografia

- [22a] *Diagrams.net. Flowchart maker & online diagram software.* 29 de març de 2022. URL: <https://app.diagrams.net/>.
- [22b] *ESP32. techtutorialsx.* 25 de març de 2022. URL: <https://techtutorialsx.com/category/esp32/>.
- [22c] *Espressif.com. Get Started - ESP32 — ESP-IDF Programming Guide latest documentation.* 8 de març de 2022. URL: <https://docs.espressif.com/projects/esp-idf/en/latest/esp32/get-started/index.html>.
- [22d] *Github.io. Material theme builder.* 12 de març de 2022. URL: <https://material-foundation.github.io/material-theme-builder/#/custom>.
- [Ben] BenoitBlanchon. *Assistant - ArduinoJson.* Ang. URL: <https://arduinojson.org/v6/assistant/#/step1> (cons. 5-4-2022).
- [Bin22] Marea99 Binario20. *RPG La llave del horizonte.* 2022. URL: <https://github.com/Binario20/RPGLlavedelhorizonte.git>.
- [Bod22] Jan Bodnar. *SQLite C. Zetcode.com.* 1 d'abr. de 2022. URL: <https://zetcode.com/db/sqlitec/>.
- [sia19] siara-cc. *Sqlite library for esp-idf (esp32) framework.* Ang. 2019. URL: <https://www.gnu.org/software/emacs> (cons. 15-3-2022).





**Part II.**  
**Apèndixs**



# A. Normes del Joc per *Gonzalo Rodríguez*

## A.1. Creación de personajes

La creación de personajes se divide en dos partes, la estética y la jugable, la primera definirá el aspecto, historia y personalidad del personaje, mientras que la segunda decidirá sus capacidades a la hora de jugar.

Sería conveniente empezar por la elección de la especie, en total hay 11 y cada una tiene características especiales que sirven de forma pasiva en las partidas, una de las importantes es la lengua, cada especie tiene una lengua, además deberá escogerse un idioma extra si la inteligencia es superior a 4, las especies son:

- *Humano*: Humanos corrientes, habitan en cualquier región.
- *Humano alado*: Humanos nacidos en Iglem, por algún extraño motivo les crecen alas.
- *Duende*: Habitantes de los bosques y zonas rurales de todo el mundo.
- *Hombre bestia*: Extraños seres, mitad hombre mitad bestia procedentes de Phenomera.
- *Vampiro*: Poderosa raza de inmortales que se alimenta de sangre.
- *Dhampiro*: Híbrido entre humano y vampiro.
- *Licántropo*: Poderosa raza de humanos capaces de transformarse en un ser similar a un lobo. Es una transformación voluntaria y solo tiene habilidades especiales al estar en forma de lobo.
- *Enano*: Humanos de pequeña estatura y complexión fuerte, viven en las montañas.
- *Demonio*: Seres inteligentes de aspectos grotescos y variados, a veces pueden adoptar formas humanas.
- *Elfo del norte*: Elfos de la tribu del norte, de rasgos finos y pálidos, viven en la zona norte del continente de Aveloz.
- *Elfo del sur*: Elfos de la tribu del sur, de rasgos más parecidos a los humanos, viven en la región sur de Aveloz.

En función de la especie y sus habilidades, es más fácil crear una personalidad e historia para el personaje, decidir su género y ponerle nombre para darle más profundidad.

A continuación, se pueden decidir la clase y los atributos del personaje, la clase es el conjunto de objetos con los que empezaras la partida (ver sección A.2) y los atributos decidirán como debe ser jugado y que cosas podrá o no hacer el personaje, para ello se reparten 25 puntos entre 7 atributos con un máximo de 10 en cualquiera de ellos y un mínimo de 1 a excepción de Puntería y Magia que pueden tener 0 (ver sección A.3).

## A.2. Sistema de clases

Son unos conjuntos de objetos para el inicio de la partida orientados a especializarse a las estadísticas del personaje creado o a los gustos del jugador, son un total de 6:

- *Guerrero*: Fuertes y robustos, cargan con grandes armas y escudos para atacar con fuerza y protegerse de sus adversarios.
- *Tirador*: especializado en ataques a distancia, usa arcos y dagas.
- *Luchador*: Confía en su fuerza y pocas veces utiliza armas, a puño desnudo derrota a sus adversarios.
- *Mago*: Usuario de báculos, varitas y pociones, porta armas ligeras como dagas y cuchillos.
- *Domador*: Suelen portar armas de caballería como sables o lanzas para poder blandirlas a lomos de sus bestias.
- *Ladrón*: Prefiere armas pequeñas fáciles de ocultar como dagas y cuchillos.

## A.3. Atributos

Jugadores: Tienen 7 atributos principales y 1 subatributo.

- Los atributos principales son:
  - *Vitalidad*: Es la cantidad de Puntos de vida del jugador.
  - *Resistencia*: Dividida entre dos y redondeada abajo se resta al daño de los ataques enemigos.
  - *Fuerza*: Es la cantidad de Vida que resta un ataque al enemigo, se le suma el daño del arma con la que se ataca, exceptuando armas de proyectiles y mágicas.
  - *Velocidad*: Es el atributo que decide quien ataca primero, el que posea el valor más alto será el primero en atacar, también influye en la posibilidad de acertar los golpes.
  - *Inteligencia*: Indica la capacidad de usar ciertos artefactos.
  - *Puntería*: Es el atributo necesario para utilizar armas de proyectiles y para acertar en cualquier ataque.
  - *Magia*: Similar a Fuerza, pero aplicable a los ataques de armas mágicas y necesario para el uso de hechizos.
- El subatributo:
  - *Destreza*: Es la suma de Velocidad y Puntería dividida entre 2. (Siempre redondeando hacia arriba)

Enemigos y bestias: Presentan 3 atributos principales y dos subatributos. Los atributos principales son Daño, Vida y Velocidad, los subatributos son la Evasión y la Puntería.

- Daño: Es la cantidad de Puntos de vida que resta el ataque del enemigo.

- Vida: Es la cantidad de puntos que hay que restar para derrotar al enemigo.
- Velocidad: Es un atributo fijo, sirve para determinar quién ataca en primer lugar.
- Evasión: Es el parámetro que decide si el enemigo recibe o no el golpe, se decide en función del tipo de enemigo.
  - Enemigos pequeños → Dado de 6 caras.
  - Enemigos medianos → Dado de 4 caras.
  - Enemigos grandes → Dado de 2 caras.
- Puntería: Es el parámetro que decide si el enemigo acierta o no el golpe, se decide en función del tipo de enemigo.
  - Enemigos pequeños → Dado de 8 caras.
  - Enemigos medianos → Dado de 6 caras.
  - Enemigos grandes → Dado de 4 caras.

Así, los enemigos más corpulentos y pesados son más lentos que los menudos y ligeros, acostumbrando a ser los enemigos grandes, fáciles de golpear y difícil que te golpeen, y los enemigos más pequeños más difíciles de golpear y más fácil que te golpeen.

### A.3.1. Mejora de los atributos

En la progresión del juego, los jugadores pueden ser premiados con puntos de experiencia que les servirían para aumentar en uno cualquiera de sus atributos subiendo de nivel, para subir el primer nivel serian necesarios tres puntos de experiencia y por cada nuevo nivel los puntos necesarios aumentarían en uno, por ejemplo, de nivel 1 a 2 se requieren los 3 iniciales mientras que de 2 a 3 se requerirán 4.

Para que el aumento de nivel sea equilibrado, el Máster deberá decidir, en función de como se desarrolle su historia y de cuantos jugadores participen, como serán otorgados los puntos de experiencia y en qué cantidad.

## A.4. Funcionamiento de la magia

La magia es uno de los atributos más complejos en cuanto a su uso, por lo que necesita una explicación más detallada.

A diferencia de la Fuerza que no requiere de armas para ser usada, la Magia solo puede usarse a través de armas y objetos, la magia usada a partir de armas funciona como el atributo de Fuerza, la Magia el daño del arma se suma al atributo de Magia y estos se convierten en daño. A través de objetos la magia se requiere a cierto nivel para utilizar hechizos y encantamientos, pueden ser utilizados dentro y fuera del combate y presentaran efectos variados, los hechizos y encantamientos están escritos en distintas lenguas que se deben conocer para utilizarlos.

## A.5. Combate

### A.5.1. Encuentros

El jugador podrá entablar combate con todo ser que posea características para ello, algunos son neutrales y contra ellos el jugador siempre tendrá la iniciativa en el primer turno, una vez iniciado el encuentro rigen las prioridades de Velocidad. Por otro lado, los enemigos atacarán al jugador siempre, a veces por sorpresa, tomando la iniciativa, luego también los turnos se regirán por la regla de Velocidad. Por último, los seres pasivos solo podrán ser atacados si el jugador los supera en Velocidad y estos nunca atacarán al jugador.

### A.5.2. Ataque

El ataque se divide en tres tipos:

- *Físico*: Utiliza el atributo de fuerza, se tira un dado igual al valor de Fuerza y al resultado se le suma el daño del arma física si se dispone de ella.
- *Mágico*: Igual que el anterior, pero utilizando el atributo de Magia al que se le suma el daño del arma mágica.
- *Proyectiles*: Las armas a distancia o arrojadas únicamente causan el daño del arma.

El daño de los ataques a los jugadores se verá mitigado por su atributo de Resistencia y por armas como el escudo, que suman a este atributo, si se recibe un ataque debe restarse a este la mitad, redondeada hacia abajo, de la suma del atributo de Resistencia y el poder del escudo si se dispone de él.

### A.5.3. Aciertos y evasión

Para determinar si un ataque impacta o no en su objetivo se hace un encuentro de dados entre los contendientes, uno atacará mientras el otro defenderá. El jugador utilizara tanto para ataque como defensa un dado con el valor del subatributo de Destreza, para los enemigos y las bestias utilizaran los dados, según tamaño, de Evasión para defensa y de Puntería para atacar. Si el atacante supera en la tirada a la del defensor, el ataque tendrá éxito, de lo contrario este fallará.

### A.5.4. Resistencias y debilidades

Las bestias y enemigos poseen características que les hacen vulnerables o resistentes a ciertos tipos de ataques. Cuando son resistentes, el daño recibido por el ataque se divide a la mitad, mientras que si son vulnerables el daño se duplica (los jugadores deberán averiguar por sí mismos estas características).

- Monstruos de mazmorra:
  - Resistente: Agua y armas normales.
  - Vulnerable: Electricidad y magia pura.
- Jabalí dorado:

- Resistente: Armas normales.
- Águila del fuego:
  - Resistente: Fuego.
- Escornau:
  - Resistente: Magia pura.
- Fieras:
  - Resistente: Cualquier ataque no mágico.
- Scorpiocaris:
  - Resistente: Agua.
  - Vulnerable: Electricidad.
- Cuelebre:
  - Resistente: Magia pura y elemental.
  - Vulnerable: Armas de acero de Éxi.

Con “armas normales” se indican aquellas armas físicas que los jugadores posean desde un inicio o que no posean una capacidad sobresaliente, como espadas, dagas, hachas, arcos. . . Son básicamente aquellas armas que no requieren de una descripción detallada en el apartado de Armas.

## **A.6. Inicio de la partida**

... ..





## B. Codí i taules de dades

### B.1. Codí de la aplicació

El codi de l'aplicació es pot trobar al *GitHub* sota el nom del treball *RPLlavedelhorizonte* públicat per l'usuari *Binario20* [Bin22]. El codi es troba dintre de la carpeta *RPG - La llave del horizonte* d'aquest repositori. L'últim comit probat i en que s'assegura que no hi ha cap error de codi seria a el de la branca *Entrega-TFG*

### B.2. Codi del servidor

```
#include <stdio.h>
#include <string.h>
#include <sqlite3.h>
#include <SPI.h>
#include <FS.h>
#include "SD.h"
#include "ArduinoJson.h"

#include <WiFi.h>
#include <WebServer.h>

IPAddress ip(192,168,4,1);
WebServer server(ip,80);

const char *ssid = "rpg-wifiAP";
const char *password = "01234567";

const byte led_blue = 4;
const byte led_red = 2;

//Init JSON
String output = "";
String input = "";
int fila = 0;

// Base de Dades
sqlite3 *bd;
char *zErrMsg = 0;
const char* data = "Callback function called";
```

```
static int callback(void *data, int argc, char **argv, char **azColName){
    int i;
    //Serial.printf("%s: \n", (const char*)data);
    //Serial.println((String)fila);
    if (fila != 0)
        output += ",";
    output = output + "\"" + (String)fila + "\":{";
    for (i = 0; i<argc; i++){
        //Serial.printf("%s = %s\n", azColName[i], argv[i] ? argv[i] : "NULL");
        if (i != 0)
            output += ",";
        String value = argv[i] ? argv[i] : "NULL";
        output = output + "\"" + azColName[i] + "\":\"" + value + "\"";
    }
    output += "}";
    fila +=1;
    //Serial.println(input);
    //Serial.printf("\n");
    return 0;
}

int openDb(const char *filename, sqlite3 **db) {
    int rc = sqlite3_open(filename, db);
    if (rc) {
        Serial.printf("Can't open database: %s\n", sqlite3_errmsg(*db));
        return rc;
    } else {
        Serial.printf("Opened database successfully\n");
    }
    return rc;
}

int db_exec(sqlite3 *db, const char *sql) {
    Serial.println(sql);
    //long start = micros();
    fila = 0;
    output = "{";
    int rc = sqlite3_exec(db, sql, callback, (void*)data, &zErrMsg);
    output += "}";
    if (rc != SQLITE_OK) {
        Serial.printf("SQL error: %s\n", zErrMsg);
        sqlite3_free(zErrMsg);
    } else {
        //Serial.printf("Operation done successfully\n");
    }
    //Serial.print(F("Time taken:"));
    //Serial.println(micros()-start);
}
```

```

    return rc;
}

boolean getOutputBD(const char *comand) {
    int rc;
    rc = db_exec(bd, comand);
    if (rc != SQLITE_OK) {
        //Serial.println(zErrMsg);
        String err = "{\"error-bd\":\":";
        char error[100];
        err.toCharArray(error, 14);
        strncat(error, zErrMsg, 98);
        char buff[3];
        String err2 = "\"}";
        err2.toCharArray(buff, 3);
        strncat(error, buff, 100);
        output = String(error);
        return false;
    }
    //Serial.println(output);
    return true;
}

void setup(){
    Serial.begin(115200);
    SPI.begin();
    SD.begin();
    pinMode(led_blue, OUTPUT);
    pinMode(led_red, OUTPUT);
    digitalWrite(led_blue, HIGH);
    Serial.println();

    // AP setup
    WiFi.softAP(ssid, password);

    Serial.print("IP address: ");
    Serial.println(WiFi.softAPIP()); //192.168.4.1

    //Init BD
    sqlite3_initialize();
    if (openDb("/sd/rpg-lldh.bd", &bd)) {
        digitalWrite(led_red, HIGH);
        return;
    }
    getOutputBD("pragma foregin_keys = ON");
    getOutputBD("delete from conectados");
    getOutputBD("delete from cosas_adquiridas");
}

```

```
getOutputBD("delete from estadisticas_conectados");
getOutputBD("update objetos_grupo set cantidad=0");

Serial.println();
digitalWrite(led_blue, LOW);

//Server setup
server.onNotFound(handleNotFound);
server.on("/rpg", getClientIP);

//Aceso BD (info)
server.on("/rpg/get/personaje",           getPersonaje);
server.on("/rpg/post/personaje",         newPersonaje);
server.on("/rpg/delete/personaje",       delPersonaje);
server.on("/rpg/get/nombre_personaje",    getPersonajeNom);
server.on("/rpg/get/personaje/nuevo",     getPersonajeData);
server.on("/rpg/get/personaje/estadisticas", getPersonajeEst);

server.on("/rpg/get/lenguas_antiguas", getLenguas);

server.on("/rpg/get/objetos",             getObjetos);
server.on("/rpg/get/objetos/material_id", getMaterialesId);
server.on("/rpg/get/objetos/armas",       getArma);
server.on("/rpg/get/objetos/armas/cuerpo", getArmaC);
server.on("/rpg/get/objetos/armas/proyectil", getArmaP);

server.on("/rpg/get/bestiario", getBestiario);

server.on("/rpg/get/magia", getMagia);

server.on("/rpg/get/regiones/no_foraneas", getRegiones);

//Aceso BD (partida)
server.on("/rpg/post/conecta", setIP);

server.on("/rpg/get/conectados", getConectados);

//server.on("/rpg/put/set_ip"             updIP);
server.on("/rpg/put/set_master-on",      setMasterOn);
server.on("/rpg/put/set_master-off",     setMasterOff);
server.on("/rpg/get/hay_master",         getMaster);
server.on("/rpg/put/set_personaje",      setPersonaje);
server.on("/rpg/put/set_lider-on",       setLiderOn);
server.on("/rpg/put/set_lider-off",     setLiderOff);
server.on("/rpg/get/soy_lider",          getLider);
server.on("/rpg/post/segunda_lengua",    set2daLengua);
//server.on("/rpg/get/segunda_lengua",    get2daLengua);
```

```

server.on("/rpg/put/conectados/estadisticas", setConEst);
server.on("/rpg/get/conectados/estadisticas", getConEst);

server.on("/rpg/get/dineros", getDineros);
server.on("/rpg/put/dineros", setDineros);

server.on("/rpg/get/cosas_adquiridas", getCosas);
server.on("/rpg/post/cosa_adquirida", newCosa);
server.on("/rpg/put/cosa_adquirida", setCosa);
server.on("/rpg/delete/cosa_adquirida", delCosa);

server.on("/rpg/get/obj_grupo", getObjGrupo);
server.on("/rpg/put/obj_grupo", setObjGrupo);

server.on("/rpg/get/img_on_display", getImg);
server.on("/rpg/put/img_on_display", putImg);

server.on("/rpg/delete/desconecta", delIP);

server.begin();
}

void loop(){
  server.handleClient();
  delay(10);
}

void sendMsg(boolean ok) {
  if (output == "{}") {
    server.send(204, "plain/text", "204 OK");
  } else if (ok) {
    server.send(200, "application/json", output);
  } else {
    server.send(500, "application/json", output);
  }
}

//Handle URI

void handleNotFound() {
  server.send(404, "application/json", "{\"404 NotFound\":\"Aquesta url no existeix\"}");
  Serial.println("{\"404 NotFound\":\""+server.uri()+"\"");
  digitalWrite(led_blue, HIGH);
  delay(750);
  digitalWrite(led_blue, LOW);
}

```

```
void getClientIP() {
    IPAddress ip = server.client().remoteIP();
    output = "{\"ClientIP\": \""+ip.toString()+"\"}";
    Serial.println(output);
    server.send(200, "application/json", output);
}
```

Per a poder mirar tot el codi del servidor s'ha d'anar a la carpeta `\webserverAP_2.1_`.

### B.3. Codi de la Base de Dades

Dades estatiques:

```
PRAGMA foreign_keys = ON;
```

```
create table if not exists genero_especie (
    id          integer not null,
    genero      text,
    primary key (id),
    unique (genero)
);
```

```
create table if not exists especies (
    id          integer not null,
    nombre      text      not null,
    id_genero   int,
    descripcion text,
    primary key (id),
    unique (nombre),
    foreign key (id_genero) references genero_especie (id)
        on update cascade
        on delete set null
);
```

```
create table if not exists habilidades_especies (
    id_especie  int not null,
    id_habilidad int not null,
    primary key (id_especie, id_habilidad),
    foreign key (id_especie) references especies (id)
        on update cascade
        on delete cascade,
    foreign key (id_habilidad) references descripcion_habilidades (id)
        on update cascade
        on delete cascade
);
```

```
create table if not exists descripcion_habilidades (  
    id            integer not null,  
    habilidad     text     not null,  
    descripcion   text,  
    primary key (id)  
);  
  
create table if not exists regiones (  
    id            integer not null,  
    nombre       text     not null,  
    id_especie   int,  
    foranea      boolean not null,  
    imagen_id    int,  
    descripcion  text,  
    primary key (id),  
    unique (nombre, foranea),  
    foreign key (id_especie) references especies (id)  
        on update cascade  
    on delete set null  
);  
  
create table if not exists servicios (  
    id            integer not null,  
    nombre       text     not null,  
    id_region    int      not null,  
    descripcion  text,  
    primary key (id),  
    unique (nombre, id_region),  
    foreign key (id_region) references regiones (id)  
        on update cascade  
    on delete set null  
);  
  
create table if not exists geografia (  
    id            integer not null,  
    nombre       text     not null,  
    id_region    int      not null,  
    imagen_id    int,  
    imagen_id_2  int,  
    descripcion  text,  
    primary key (id),  
    unique (nombre, id_region),  
    foreign key (id_region) references regiones (id)  
        on update cascade  
    on delete set null  
);
```

```
create table if not exists lenguas_antiguas (
    id            integer not null,
    nombre       text     not null,
    id_especie   int,
    descripcion  text,
    primary key (id),
    unique (nombre, id_especie),
    foreign key (id_especie) references genero_especie (id)
        on update cascade
    on delete set null
);

create table if not exists biblioteca_de_magia (
    id            integer not null,
    nombre       text     not null,
    tipo         text,
    id_lengua    int      not null,
    requisito    int      not null,
    imagen_id    int,
    descripcion  text,
    primary key (id),
    unique (nombre, id_lengua),
    foreign key (id_lengua) references lenguas_antiguas (id)
        on update cascade
    on delete set null
);

create table if not exists hechizos (
    id            integer not null,
    hechizo      text     not null,
    id_libro     int,
    descripcion  text,
    primary key (id),
    foreign key (id_libro) references biblioteca_de_magia (id)
        on update cascade
    on delete set null
);

create table if not exists personajes (
    id            integer not null,
    nombre       text     not null,
    vitalidad    int      not null,
    resistencia  int      not null,
    fuerza       int      not null,
    velocidad    int      not null,
    inteligencia int      not null,
    punteria     int      not null,
```



```

    magia            int      not null,
    destreza         int      not null,
    id_procedencia   int      not null,
    id_especie       int      not null,
    edad             int      not null,
    altura_m         float    not null,
    peso_kg          float    not null,
    sexo             text     not null,
    id_clase         int      not null,
    id_lengua        int,
    personalidad     text,
    fisico           text,
    imagen_id        int,
    primary key (id),
    foreign key (id_procedencia) references regiones (id)
        on update cascade
    on delete set null,
    foreign key (id_especie) references especies (id)
        on update cascade
    on delete set null,
    foreign key (id_clase) references clases (id)
        on update cascade
    on delete set null
);

create table if not exists clases (
    id                integer not null,
    clase             text     not null,
    preferencia1     text,
    preferencia2     text,
    preferencia3     text,
    montura          boolean not null,
    descripcion      text,
    primary key (id),
    unique (clase)
);

create table if not exists objeto_inicial_clases (
    id_clase         int not null,
    id_arma          int,
    id_objeto        int,
    primary key (id_clase),
    foreign key (id_clase) references clases (id)
        on update cascade
    on delete cascade,
    foreign key (id_arma) references objetos (id)
        on update cascade

```

```
        on delete set null,
        foreign key (id_objeto) references objetos (id)
            on update cascade
        on delete set null
);

create table if not exists objetos (
    id            integer not null,
    nombre        text    not null,
    tipo          text,
    imagen_id     int,
    descripcion   text,
    primary key (id),
    unique (nombre)
);

create table if not exists genera_objeto (
    objeto_principal  int not null,
    objeto_secundario int,
    objeto_nuevo      int not null,
    primary key (objeto_nuevo),
    foreign key (objeto_nuevo) references objetos (id)
        on update cascade
    on delete cascade,
    foreign key (objeto_principal) references objetos (id)
        on update cascade
    on delete cascade,
    foreign key (objeto_secundario) references objetos (id)
        on update cascade
    on delete set null
);

create table if not exists armas_cuerpo_a_cuerpo (
    id_arma          int not null,
    subtipo          text not null,
    requisito        int not null,
    requisito_campo  text not null,
    operacion        text not null,
    suma1            int not null,
    suma1_campo      text not null,
    suma2            int,
    suma2_campo      text,
    ataque           text,
    rango            text,
    normal           boolean not null,
    primary key (id_arma),
    foreign key (id_arma) references objetos (id)
);
```

```
        on update cascade
    on delete cascade
);

create table if not exists armas_proyectiles (
    id_arma      int not null,
    subtipo      text not null,
    requisito1   int not null,
    campo1       text not null,
    requisito2   int,
    campo2       text,
    dano         int not null,
    ataque       text,
    rango        text,
    normal       boolean not null,
    primary key (id_arma),
    foreign key (id_arma) references objetos (id)
        on update cascade
    on delete cascade
);

create table if not exists bestiario (
    id           integer not null,
    tipo         text not null,
    nombre       text not null,
    clasificacion text not null,
    montura      boolean not null,
    tamano       text not null,
    dano         int not null,
    vida         int not null,
    velocidad    int not null,
    imagen_id    int,
    descripcion  text,
    primary key (id),
    unique (nombre)
);

create table if not exists bestiario_seres_varios (
    id_nombre      int not null,
    clasificacion_add text,
    experiencia_derrota int,
    resistente      text,
    vulnerable      text,
    extras          text,
    primary key (id_nombre),
    foreign key (id_nombre) references bestiario (id)
        on update cascade
);
```

```
        on delete cascade
    );

create table if not exists descripciones_atributos (
    id            integer not null,
    atributo     text     not null,
    tipo         text,
    aplicado     text     not null,
    descripcion  text,
    primary key (id),
    unique (aplicado, atributo)
);
```

Dades dinàmiques de cada partida:

```
create table if not exists conectados (
    id            integer not null,
    ip            text     not null,
    id_personaje text,
    id_lengua     int,
    master        boolean not null,
    lider         boolean not null,
    primary key (id),
    unique(ip),
    foreign key (id_personaje) references personajes (id)
        on update cascade
    on delete set null,
    foreign key (id_lengua) references lenguas_antiguas (id)
        on update cascade
    on delete cascade
);

create table if not exists cosas_adquiridas (
    id_jugador  int  not null,
    id_cosa     int  not null,
    tipo        text not null,
    cantidad    int,
    primary key (id_jugador,id_cosa,tipo),
    foreign key (id_jugador) references conectados (id)
        on update cascade
    on delete cascade
);

create table if not exists objetos_grupo (
    id_objeto int not null,
    cantidad  int,
    primary key (id_objeto),
```

```
foreign key (id_objeto) references objetos (id)
    on update cascade
on delete cascade
);

create table if not exists estadisticas_conectados (
    id_jugador    int not null,
    vitalidad     int not null,
    resistencia   int not null,
    fuerza        int not null,
    velocidad     int not null,
    inteligencia  int not null,
    punteria      int not null,
    magia         int not null,
    destreza     int not null,
    primary key (id_jugador),
    foreign key (id_jugador) references conectados (id)
        on update cascade
    on delete cascade
);
```

El codi de la base de dades es troba a la carpeta \bd

## B.4. Taules de dades

Les taules de dades es poden trobar a la carpeta \bd\tables\_csv