

Grado en ingeniería Informática de Gestión y Sistemas de Información

**Desarrollo de una herramienta digital con comunicación a tiempo
real usando Firebase para la automatización de procesos en
Krosmaster, un juego de mesa**

Memoria

Joel Diaz Clemente

TUTOR: Dr. Enric Sesa i Noguera

CURSO 2020-2021

Abstract

This final degree project aims to create a digital tool that serves as a support to play a board game named Krosmaster, both to reduce the complexity and the number of elements necessary to perform the game, and to maintain control of the different collectibles related to it. On the other hand, this FDP aims to create a space where you can make online exchanges, as well as meetings and a forum to answer questions, receive suggestions or discuss issues related to the game.

Resum

Aquest treball de final de grau té com a objectiu la creació d'una eina digital que serveixi com a suport per a jugar a un joc de taula anomenat Krosmaster, tant per reduir la complexitat i el nombre d'elements necessaris per a realitzar el joc, com per a mantenir un control dels diferents col·leccionables relacionats amb el mateix. D'altra banda, aquest TFG té com a objectiu la creació d'un espai on poder fer intercanvi online, així com quedades i un fòrum per a resoldre dubtes, rebre suggeriments o discutir sobre temes relacionats amb el joc.

Resumen

Este Trabajo de final de grado tiene como objetivo la creación de una herramienta digital que sirva como soporte para jugar a un juego de mesa nombrado Krosmaster, tanto para reducir la complejidad y el número de elementos necesarios para realizar el juego, como para mantener un control de los diferentes coleccionables relacionados con el mismo. Por otra parte, este TFG tiene como objetivo la creación de un espacio donde poder hacer intercambio online, así como quedada y un foro para resolver dudas, recibir sugerencias o discutir sobre temas relacionados con el juego.

Índice

Índice	iii
Índice de figuras.....	v
Índice de tablas	vii
Glosario de términos	viii
1. Introducción	9
1.1 Propósito.....	9
1.2 Historia	9
1.3 Explicacion del juego	13
2. Análisis de antecedentes	20
2.1 Otras páginas web o aplicaciones centradas en dar información sobre el juego (Krosmaster):.....	20
2.2 Otros juegos que tengan herramientas digitales para soportarlos:	22
2.3 Conclusión de los antecedentes	26
3. Marco de desarrollo	27
3.1 Android.....	27
3.2 Entorno de desarrollo: Android Studio	28
3.3 Firebase.....	28
4. Objetivos y alcance	32
4.1 Detalles de objetivos.....	32
4.2 Objetivos del producto.....	33
4.3 Objetivos del cliente	33
4.4 Público potencial	33
5. Metodología	34
5.1 Planificación	35
6. Requisitos funcionales y tecnológicos	36
7. Desarrollo.....	37
7.1 Casos De uso	37
7.2 Estructura del proyecto	50
7.3 Interfaz Grafica.....	53
8. Diseño de base de datos	68

8.1 Cartas.....	68
8.2 Hechizos.....	69
8.3 Partida	70
8.4 Poderes	73
8.5 Recompensas Démoniacas	73
8.6 Usuarios.....	74
8.7 Firebase Storage	74
9. Presupuesto.....	76
Coste de recursos humanos	76
Coste de recursos materiales	77
Coste de licencias, suministros y servicios.....	78
Coste Total	78
10. Conclusiones.....	79
11. Referencias	81

Índice de figuras

Figura 1: Gameplay de Dofus donde se puede ver el formato de casillas estilo ajedrez	10
Figura 2: Fotografía del juego de mesa Krosmaster Arena	11
Figura 3: fotografía promocional del Torneo de Krosmaster de 2015.....	12
Figura 4:Ejemplo de tablero de juego.....	13
Figura 5: Cara crítico del dado	15
Figura 6:: Cara armadura del dado	15
Figura 7: Cara placaje del dado	15
Figura 8: Cara esquivada del dado	15
Figura 9: Cara crítico/esquivada del dado.....	15
Figura 10: Cara dofus del dado	16
Figura 11: Ejemplo de carta de krosmaster	16
Figura 12: Fragmento superior de la figura 11	17
Figura 13: Fragmento medio de la figura 11	17
Figura 14: Fragmento inferior de la figura 11	18
Figura 15: Captura de pantalla de Krosarchive.es [3]	20
Figura 16: Captura de pantalla de Krosfinder.com [4]	22
Figura 17: Fotografía Promocional de venta para el juego de mesa Atmosfear	23
Figura 18: Captura de pantalla de Syrinscape	24
Figura 19: Captura de pantalla de Fight Club 5th Edition.....	25
Figura 20: Captura de pantalla de una partida de Dungeon & Dragons a través de la página web Roll20	26
Figura 21: Figura del funcionamiento de la metodología de desarrollo en cascada	34
Figura 22: Representación del ciclo de vida del TFG	35
Figura 23: Estructura del proyecto	50
Ilustración 24: Estructura de backend	51
Ilustración 25: Estructura Frontend.....	51
Figura 26: Diseño de pantallas	53
Figura 27: Pantalla de Registro / Acceso.....	54
Figura 28: Pantalla Inicial	55
Figura 29: Pantalla de consultar equipos.....	56
Figura 30: Pantalla de figura	57
Figura 31: Pantalla de crear equipo	58
Figura 32: Pantalla de consultar equipo.....	59
Figura 33: Pantalla de seleccionar equipo	60
Figura 34: Pantalla de crear partida.....	61
Figura 35: Pantalla de unirse a partida	62
Figura 36: Pantalla de partida.....	63
Figura 37: Pantalla de crear carta	66
Figura 38: Pantalla de crear hechizo.....	67

Figura 39: Estructura Firebase padre	68
Figura 40: Estructura Firebase Cartas	68
Figura 41: Estructura Firebase Hechizos	69
Figura 42: Estructura Firebase Partida	70
Figura 43: Estructura Firebase Partida – Jugador	71
Figura 44: Estructura Firebase Partida – Jugador - Carta.....	72
Figura 45: Estructura Firebase Poderes.....	73
Figura 46: Estructura Firebase Recompensas Demoníacas.....	73
Figura 47: Estructura Firebase Usuarios	74
Figura 48: Firebase Database	75
Figura 49: Firebase Database / Figuras.....	75
Figura 50: Firebase Database / Figuras / MIAUBUELO.png	75

Índice de tablas

Tabla 3: Gráfico y tabla de utilización de los diferentes sistemas operativos	27
Tabla 1: Tabla de planificación del TFG.....	35
Tabla 2: Requisitos funcionales y Tecnológicos y Soluciones Tecnológicas.....	36
Tabla 4: Precio/Hora en función de la tarea	76
Tabla 5: Precio total de las faenas del proyecto	76
Tabla 6: Coste de recursos materiales	77
Tabla 7: Detalles de coste de la estación de trabajo	78
Tabla 8: Coste de licencias, suministros y servicios.....	78

Glosario de términos

Back-end: Capa de acceso de datos

CSS: Cascading Style Shits, Lenguaje de diseño que será aplicado sobre HTML

Front-end: Interfaz de usuario

Galones de Gloria (GG): Una concurrencia que sirve para ganar o perder una partida en el juego Krosmaster

Gameplay: El hecho de jugar a un juego

HTML: HyperText Markup Language, Lenguaje de programación para páginas web

Java: Lenguaje de programación en el que se ha desarrollado el proyecto

JSON: JavaScript Object Notation, es un formato de texto sencillo para el intercambio de datos.

Kamas: El nombre que recibe la moneda de juego en Krosmaster

TFG: Trabajo de Fin de Grado

MMORPG: Massive Multiplayer Online Role Play Game

OVA: Original Video Animation, producciones audiovisuales animadas destinadas para su consumo en video.

Recompensa Demoníaca: El nombre que reciben los objetos que modifican varios atributos y características de las figuras en el juego Krosmaster

Wiki: El término wiki (palabra que proviene del hawaiano wiki, «rápido») alude al nombre que recibe una comunidad virtual, cuyas páginas son editadas directamente desde el navegador, donde los mismos usuarios crean, modifican, corrigen o eliminan contenidos que, habitualmente, comparten.

1. Introducción

1.1 Propósito

Este proyecto tiene como objetivo ofrecer nuevas herramientas tecnológicas para un juego de mesa ya existente, Krosmaster- Las herramientas que se pretenden ofrecer son:

En primer lugar, una herramienta de seguimiento para la partida que automatiza algunos procesos para así disminuir la posibilidad de que en esta misma ocurran errores humanos, así como hacer la entrada de nuevos jugadores al juego de mesa mucho más asequible en términos de dificultad y disminuir la cantidad de elementos necesarios para la realización de una partida.

En segundo lugar, una herramienta de catálogo donde poder ver todos los coleccionables relacionados con el juego de mesa.

La motivación para realizar este trabajo nace de la dificultad de nuevos jugadores para entender las mecánicas complejas del juego, así como para ayudar a los jugadores a concentrarse únicamente en la estrategia, reduciendo la necesidad de atención de estos mismos en otros elementos que son parte del juego ya que es posible automatizarlos a través de una aplicación.

Este proyecto es el resultado final de la asignatura "Treball de Fi de Grau", cursada por el estudiante Joel Diaz Clemente, y supervisado por el Doctor Enric Sesa i Nogueras. Esta asignatura cuenta con un valor de 20 créditos ECTS y es cursada durante el cuarto curso del Grado en ingeniería Informática de Gestión y Sistemas de Información en la universidad Tecnocampus Mataró-Maresme.

La parte práctica de este trabajo ha sido realizada utilizando el entorno de desarrollo integrado oficial para la plataforma Android, Android Studio. Para la programación en back-end, se ha utilizado el lenguaje JAVA. Para la programación en front-end se ha utilizado HTML con CSS. También se ha usado en gran medida Firebase Realtime Database, una base de datos alojada en la nube.

1.2 Historia

Para hablar de Krosmaster es necesario entender sobre la empresa que lo ha desarrollado:

Ankama es una empresa creada el 2001 en Francia que desarrolló y lanzó al mercado su primer videojuego Dofus el 2004.

Dofus es un MMORPG que implementó un estilo de juego bastante común para la época en juegos como Final Fantasy o Pokemon, siendo este un sistema de combate por turnos donde el jugador debía equipar a sus personajes con diversos objetos y desbloquear nuevas habilidades con el fin de completar una historia pero añadiendo varios toques nuevos, los más importantes siendo un nuevo sistema de desplazamiento bidimensional en combate al estilo ajedrez con celdas y movimientos limitados y el añadido de ser un juego 100% online donde podías jugar con otra gente.



Figura 1: Gameplay de Dofus donde se puede ver el formato de casillas estilo ajedrez

A partir de este punto la empresa no quiso quedarse solo con juegos de ordenador y decidió empezar una serie de cómics relacionados con el mundo de Dofus.

Dofus Arena, tenía un estilo de juego muy parecido a Dofus, pero enfocándose únicamente en el JcJ (jugador contra jugador) y con partidas auto conclusivas, a diferencia de Dofus donde cada combate era un pequeño fragmento de una gran imagen que terminaba formando una historia.

En 2008, también se expandirían a la industria de la producción audiovisual con la creación de la serie animada "Wakfu" la cual tendría los derechos de distribución vendidos a Netflix en 2018 y que cuenta actualmente con 65 capítulos, 3 temporadas y un OVA.

Finalmente, en 2013, Ankama lanzaría su primer juego de mesa, "Krosmaster Arena", el cual fue galardonado como uno de los 10 mejores juegos de mesa en 2013 según Game Informer[2].



Figura 2: Fotografía del juego de mesa Krosmaster Arena

A raíz de su éxito, la empresa seguiría con el desarrollo de "Krosmaster Junior", una versión más sencilla para niños y "Krosmaster Arena Frigost", una expansión del primer juego "Krosmaster Arena" que contaría con un nuevo tablero y un nuevo conjunto de objetos.

En 2015 lanzarían "Krosmaster Quest", una versión del juego enfocada al cooperativo en lugar del competitivo, es decir, los jugadores se tienen que unir para derrotar a un elemento del juego en lugar de tener que luchar entre ellos.

En 2016 realizarían la segunda versión del juego lanzando "Krosmaster Arena 2.0". El cual contaría con un nuevo tablero y nuevos objetos.

En 2018 lanzarían "Boufbowl", un nuevo juego de mesa que mezclaría el estilo de juego de Krosmaster y el Fútbol en un nuevo concepto muy casual.

Finalmente, en 2020, lanzarían "Krosmaster Blast", con un gran número de cambios a las reglas y las mecánicas intentando hacer las partidas más cortas y dinámicas, pero perdiendo parte de la complejidad del juego.

Todos estos juegos de mesa contenían una serie de miniaturas jugables con cartas asociadas que se podían usar para jugar a cualquier juego de la saga, haciendo así que los jugadores quisieran comprar todos los distintos juegos para coleccionar todas las miniaturas jugables.

Paralelamente, también lanzaron distintas colecciones de cajas sorpresas divididas en distintas temporadas donde se podían conseguir otras miniaturas jugables distintas que no venían en los juegos base.

Alrededor de estos juegos de mesa, se empezó a crear una comunidad y Ankama, empezó a promover torneos oficiales, donde se jugaba a los juegos "Krosmaster Arena", "Krosmaster Arena: Frigost" Y "Krosmaster Arena 2.0" a medida que iban saliendo los juegos hasta 2020 donde el torneo que se tenía que hacer para "Krosmaster Blast" fue cancelado debido a la Pandemia mundial que se vivió y no se han realizado más torneos desde ese momento.



Figura 3: fotografía promocional del Torneo de Krosmaster de 2015

En conclusión, la empresa cuenta actualmente con un total de seis juegos de ordenador, diez juegos de móvil, un juego de Xbox, dos series, cuatro series de Comics y diez juegos de mesa.

1.3 Explicación del juego

1.3.1 Introducción

Antes de nada, necesitamos entender el juego de mesa sobre el que está basado el proyecto así que se hará una breve explicación de las mecánicas del juego:

Krosmaster es un juego de mesa táctico parecido al ajedrez que dura de diez a cincuenta minutos.

Tiene distintas modalidades, pero en la que se va a centrar este proyecto es aquella donde dos equipos luchan para dejar al equipo contrario sin puntos y ganar. Estos puntos están en forma de unas fichas llamadas Galones de Gloria (GG a partir de ahora).

1.3.2 Fundamentos del juego

Todo el reglamento del juego ha sido sacado de lastlevel.es [1], donde se recoge el reglamento para las reglas de torneo.

Espacio de juego

El juego se realiza en un tablero. Hay distintos escenarios en los que se puede jugar, con su propia escenografía y casillas especiales. Al montar el tablero hay que colocar la escenografía y las Kamas (está marcado en el tablero donde tiene que ir con iconos). Las Kamas son la moneda del juego y sirven, entre otras cosas para comprar ciertas mejoras.



Figura 4: Ejemplo de tablero de juego

Construcción de equipo

A grandes rasgos, un equipo está compuesto de dos a ocho figuras con un nivel total de equipo que tiene que ser exactamente doce. La repetición de figuras iguales está limitada en función de varios factores

Iniciando el juego

Cada jugador coge seis GG y se coloca uno adicional en medio de los dos. Este GG se llama GG salvaje y se lo quedará el primer jugador en eliminar una figura enemiga.

Se barajan las Recompensas demoníacas y se hacen dos pilas con cada tipo de recompensa demoníaca mirando boca abajo. Estas recompensas Demoníacas son objetos que podremos comprar y equipar en nuestros personajes costando según el tipo:

- Granito: 3 Kamas
- Jade: 6 Kamas
- Oro: 12 Kamas
- Rubí: 18 Kamas

En el primer turno todas las recompensas demoníacas están boca abajo, al segundo turno se desvela una de cada tipo y al tercero otra más. A partir de aquí se irán sustituyendo cuando se compren.

Cada jugador crea su Línea temporal ordenando sus personajes de izquierda a derecha por orden decreciente de iniciativa

Se suma la iniciativa de todas las figuras y el equipo con más iniciativa empieza a colocar sus figuras en el tablero, se escoge un lado y se colocan las figuras en las casillas de salida marcadas en el tablero.

Victoria y Derrota

Para ganar una partida de Krosmaster se necesita o bien dejar al adversario sin GG o bien dejar al adversario sin figuras para jugar.

Al eliminar una figura adversaria, roba al adversario puntos de GG en función del nivel de la figura eliminada.

Dados

Krosmaster se juega con dados especiales. Estos tienen 6 caras y las caras tienen los siguientes símbolos:

- Crítico: Los hechizos de la figura que tenga este dado infligen 1 de daño adicional



Figura 5: Cara crítico del dado

- Armadura: La figura sobre la que se use este dado recibe 1 de daño menos



Figura 6: Cara armadura del dado

- Placaje: La figura sobre la que se use este dado no dejara que otras figuras se muevan de su lado



Figura 7: Cara placaje del dado

- Esquiva: La figura sobre la que se use este dado se podrá mover del lado de una figura que tenga un dado de placaje



Figura 8: Cara esquiva del dado

- Crítico/Esquiva: El jugador que lance este lado podrá escoger entre girar el dado al lado de crítico o esquiva



Figura 9: Cara crítico/esquiva del dado

- Dofus: El jugador que lance este lado podrá escoger entre girar el dado a cualquier lado



Figura 10: Cara dofus del dado

1.3.3 descripción de las cartas

Las cartas en Krosmaster nos muestran toda la información de las figuras a las que están asociadas:



Figura 11: Ejemplo de carta de krosmaster

Nombre

En la parte superior izquierda de la carta, en mayúsculas. En la figura 11, el personaje se llama QILBY.

Clase

Debajo del nombre del Krosmaster. Algunos poderes afectan a clases específicas. En la figura 11, el personaje tiene dos clases: Selatrop y Hermandad del Tofu.

Ilustración

La ilustración está impresa en la parte superior derecha de la carta. Sirve para identificar a que figura está asociada la carta

Características



Figura 12: Fragmento superior de la figura 11

Una figura tiene diferentes características que se imprimen en su carta. Estas características son su Nivel, su Iniciativa, sus Puntos de Movimiento, Puntos de Vida y Puntos de Acción.

- **Nivel:** En la esquina superior derecha dentro de una estrella de 8 puntas amarilla. Sirve al crear el equipo y al eliminar la figura. Ejemplo, este personaje es nivel 6.
- **Iniciativa:** En la esquina superior izquierda, debajo de la clase. Sirve al decidir quien juega primero y en qué orden van las figuras en la línea de tiempo. Ejemplo, este personaje tiene una iniciativa de 6.
- **Puntos de Movimiento (PM):** En verde, debajo de la iniciativa. Representa cuantas casillas se puede mover una figura en su turno. Se recuperan al principio de cada turno. Ejemplo, este personaje tiene 2 PM.
- **Puntos de Vida (PdV):** En rojo, a la derecha de los PM. Representa cuantas heridas puede aguantar una figura antes de ser eliminado. Ejemplo, este personaje tiene 16 PdV.
- **Puntos de acción (PA):** En azul, a la derecha de los PdV. Representa el número de acciones que puede hacer una figura. Se recuperan al principio de cada turno. Ejemplo, este personaje tiene 10 PA.

Hechizos



Figura 13: Fragmento medio de la figura 11

- **Nombre:** El nombre del hechizo. Ejemplo: el primer hechizo se llama muralla
- **Cuadro de texto:** Debajo del nombre del hechizo, describe el hechizo, algún efecto adicional (si tiene) o el área de efecto (si tiene)

- Alcance: A la izquierda del nombre del hechizo, define desde cuantas casillas y hasta cuantas se puede lanzar el hechizo
- Tipo de Lanzamiento: Encima del alcance, define como puede ser el lanzamiento del hechizo. Si tiene una flecha solo se puede lanzar en línea recta. Si tiene una X se puede lanzar a través de obstáculos u otros personajes. Si tiene un puño solo se puede lanzar en contacto
- Coste: El coste en PA de acción de lanzar ese hechizo. Ejemplo: El hechizo muralla cuesta 5 PA, como QILBY tiene 10 PA, podrá lanzar ese hechizo dos veces por turno si no gasta PA en ninguna otra cosa.
- Daño / Curación: la cantidad de heridas que inflige el hechizo al objetivo / la cantidad de heridas que cura el hechizo al objetivo.
- Restricciones de Lanzamiento: Los hechizos tienen distintos niveles de restricciones en función del color del fondo que hay en la parte del nombre:
 - Negro: Sin restricción
 - Azul: Una vez por turno
 - Morado: Una vez por objetivo, pero varias veces por turno si lo usas en varios objetivos
 - Rojo: una vez por partida
- Cada Krosmaster tiene un Hechizo llamado "Puñetazo" sin imprimir en su carta de personaje. Se trata de un ataque cuerpo a cuerpo sobre fondo negro, que cuesta 5 PA y que causa 1 punto de daño Neutral sin ningún efecto adicional

Poderes



Figura 14: Fragmento inferior de la figura 11

Habilidades especiales de cada personaje. Si no tienen descripción son comunes y las descripciones están en el libro de reglas. Si tienen descripción son específicas del personaje en cuestión.

1.3.4 Turno de juego

El turno de juego se divide en 4 fases:

Lanzamiento de Tensión

El jugador activo lanza dos dados. Si ambos dados tienen caras idénticas, ambos jugadores pierden un GG

Inspiración

El jugador puede asignar los dados resultantes del lanzamiento de tensión a una carta para darle uno de los poderes resultantes durante ese turno

Pago

Si el jugador activo no ha colocado ningún dado de inspiración, recibe tres Kamas, si ha colocado uno recibe una Kama.

Turnos de las figuras

El jugador pone en juego cada miembro de su equipo según el orden de su línea temporal. Este turno se divide en 3 fases para cada figura:

- Fase preliminar: Todos los efectos provocados "al comienzo del turno" son activados.
- Fase de activación El personaje puede consumir sus PA y PM.
- Fin de fase de activación: Desaparecen los efectos que duran "este turno" y finaliza el turno de la figura activa.

Al jugar con todos los miembros de su equipo el jugador termina su turno.

2. Análisis de antecedentes

Para el análisis de antecedentes de este proyecto ha habido dos focos:

2.1 Otras páginas web o aplicaciones centradas en dar información sobre el juego (Krosmaster):

Gracias a realizar este análisis podremos discernir que cosas ya existen y cuáles no para poder saber si nuestra aplicación está dando una nueva solución, mejorando una existente o simplemente será una solución más.

1. **Krosarchive.es:** Página web centrada en dar información sobre el juego.

Por un lado, da información sobre las figuras jugables ofreciendo una lista con todas las miniaturas jugables actualmente disponibles, a qué edición y pack pertenecen, qué hacen, qué características tienen, una imagen de la figura en cuestión y una figura de la carta asociada, tanto por delante como por detrás. También en que formato de torneo es posible jugarla, y, en caso de que tenga, una fotografía de las fichas que invoquen por delante y por atrás.

Por otro lado, nos ofrece una herramienta mediante la cual puedes saber si, desde cierta posición es posible usar las habilidades de una figura sobre otra (línea de visión).

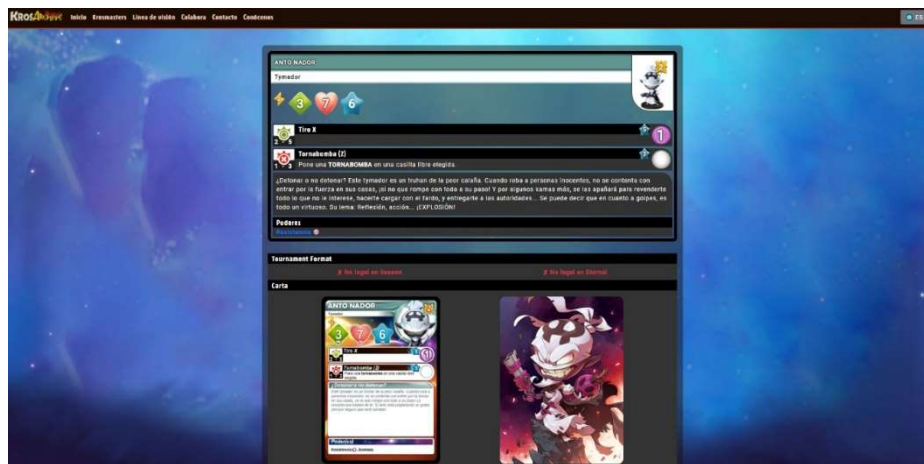


Figura 15: Captura de pantalla de Krosarchive.es [3]

Conclusión: Es una página web poco completa a la que le faltan muchas partes. Todo y eso, al ser la única página en castellano que actualmente existe, mucha gente la usa.

2. **Krosfinder.com:** Esta página web está actualmente cerrada, pero gracias a la página web.archive.org, somos capaces de ver cómo fue en su día. Está centrada en dar información sobre distintos apartados del juego:

Figuras jugables: contiene una lista con todas las miniaturas jugables actualmente disponibles, sus características, a qué edición pertenecen, qué hacen y en qué formato de torneo se pueden jugar.

Ediciones: Muestra una lista con todas las ediciones y los packs existentes y las figuras que forman parte de estos.

Reglas: Muestra una lista con todas las reglas para los torneos, así como las figuras prohibidas y las diferentes figuras que conforman cada nivel.

Fichas: contiene una lista de todas las fichas que se pueden invocar, qué hacen, qué características tienen y quién las puede invocar

Recompensas: muestra una lista de las diferentes recompensas clasificándolas según las diferentes versiones del juego

Poderes: contiene una lista de los diferentes poderes de los personajes y qué personajes los tienen

Clases: muestra una lista de las diferentes clases de los personajes y qué personajes conforman estas

Efectos Adicionales: muestra una lista de los diferentes efectos adicionales que puede tener un hechizo y que hechizos tienen cada efecto adicional

Líneas de visión: nos ofrece una herramienta mediante la cual puedes saber si, desde cierta posición es posible usar las habilidades de una figura sobre otra (línea de visión).

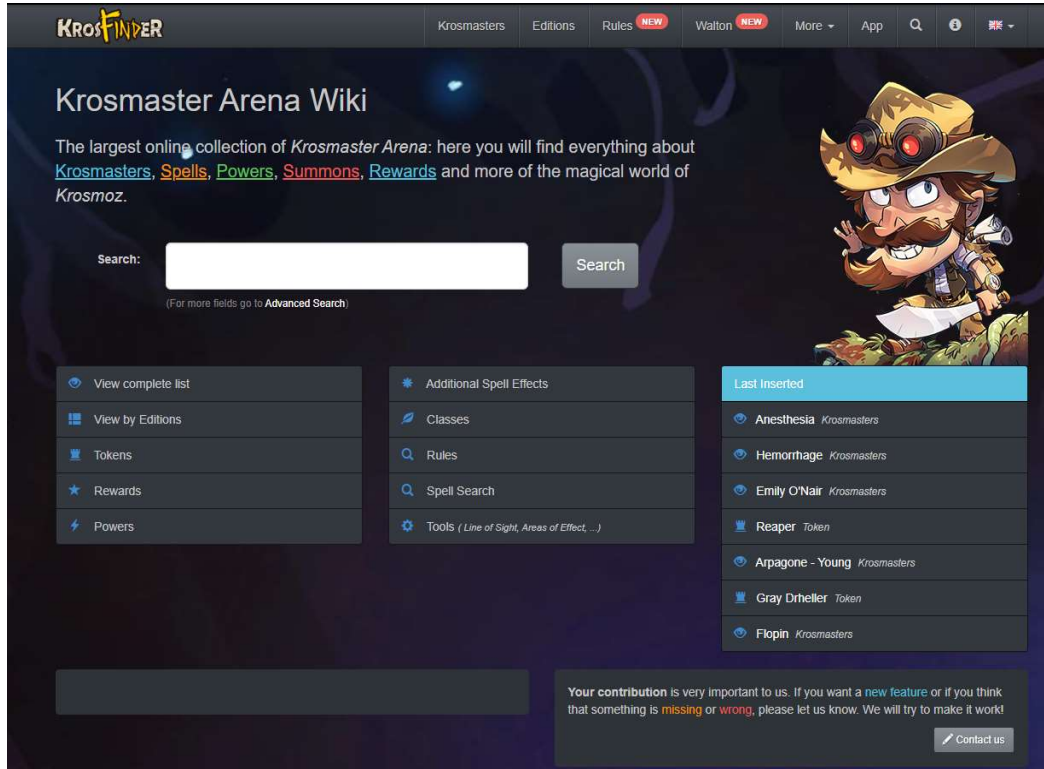


Figura 16: Captura de pantalla de Krosfinder.com [4]

Conclusión: Es una página muy completa pero ya no está vigente y sólo se puede acceder a ella desde una web de terceros donde faltan muchas imágenes y parte de la información es inaccesible.

2.2 Otros juegos que tengan herramientas digitales para soportarlos:

Gracias a realizar este análisis podremos analizar herramientas parecidas en distintos juegos para ver si han funcionado y que cosas se podrían mejorar.

1. **Atmosfear:** Un juego de mesa interactivo, donde mientras se va jugando, se tiene que visualizar un vídeo en la televisión donde un actor hace a los jugadores decir y hacer ciertas cosas, así como tomar ciertas decisiones que acabarán influyendo en la partida.

Conclusión: Más que una ayuda para jugar es una parte del juego y no se puede entender el juego sin el vídeo. Es un buen ejemplo de la implementación de la tecnología en los juegos de mesa como medio que lleva desde 1991 funcionando.



Figura 17: Fotografía Promocional de venta para el juego de mesa Atmosfear

2. **Syrinscape:** Una herramienta para cualquier juego que añade música y efectos de ruido como gritos de personas o rugidos de monstruos para dar ambiente a cualquier partida.

Conclusión: Es una herramienta muy interesante y única que ayuda en la inmersión en el juego de mesa. Es otro buen ejemplo de la implementación de la tecnología en los juegos de mesa como soporte y no como medio.



Figura 18: Captura de pantalla de Syrinscape

3. **Fight Club 5th Edition:** Una herramienta para ayudar al jugador a diseñar su personaje en el juego Dungeons & Dragons y para mantener el seguimiento de los puntos de vida y otras estadísticas de forma digital.

Conclusión: Esta herramienta es muy similar a una parte de lo que se quiere desarrollar, pero es para un juego diferente. Es un ejemplo de que la aplicación que se desarrollará ya se está usando en otros juegos de mesa y funciona.

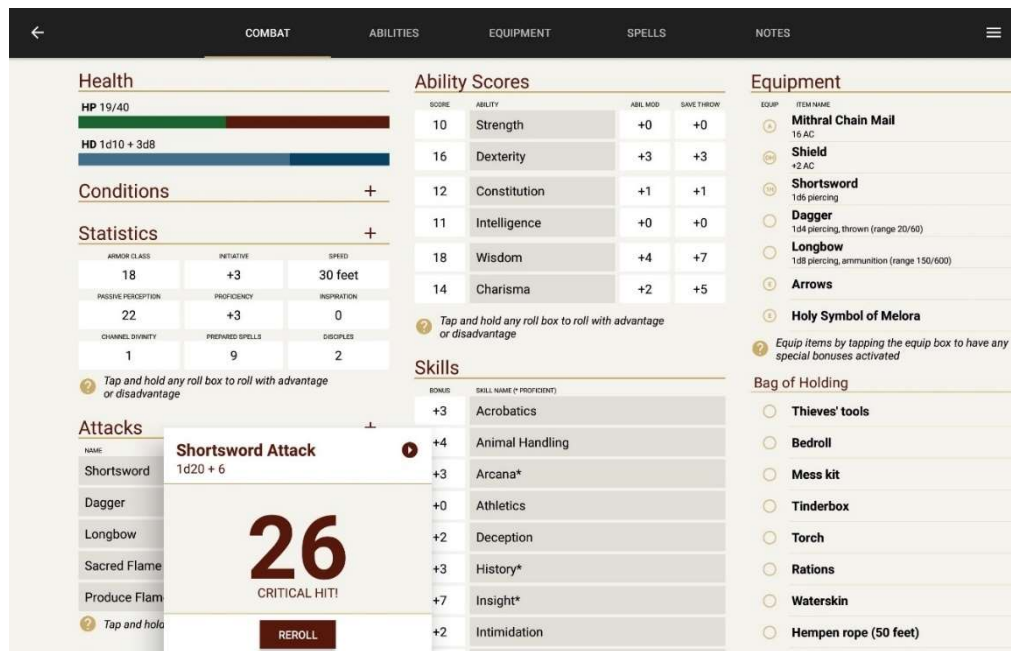


Figura 19: Captura de pantalla de Fight Club 5th Edition

4. **Roll20:** Una página web que sirve para crear y gestionar partidas de diferentes juegos. La forma en que funciona es permitiendo que un jugador cree la partida y sea el administrador de está permitiendo que otros jugadores puedan unir-se a esta y jugar en base a lo que el jugador administrador vaya haciendo. Tiene una gran cantidad de herramientas útiles entre las que se encuentran creación de mapas, creación de personajes que llevaran los jugadores, creación de personajes enemigos para el administrador del juego, tiradas de distintos tipos de dados, gestión de características de personajes, creación de animaciones / efectos, etc.

Conclusión: Una herramienta muy completa y útil similar a lo que se quiere desarrollar. En este caso, sin embargo, la intención de la herramienta es no depender del juego físico y realizar toda la actividad a través del ordenador, mientras que lo que se pretende realizar con la creación de una nueva aplicación es ser un complemento de un juego físico.



Figura 20: Captura de pantalla de una partida de Dungeons & Dragons a través de la página web Roll20

2.3 Conclusión de los antecedentes

Después de realizar el análisis de antecedentes podemos concluir que:

Ya existen algunas páginas que proporciona herramientas para dar información sobre el juego, pero estas están incompletas o ya es imposible acceder a ellas ya que sus creadores han abandonado el dominio web.

Estas páginas en ningún caso intentan ser una herramienta que de soporte a los jugadores durante la partida, sino que son más bien una wiki.

Existen herramientas para dar soporte a los jugadores durante la partida. Especialmente para el juego "Dungeons & Dragons". Estas herramientas resultan útiles a los jugadores y suelen dar soporte a la gestión de los diferentes puntos y características de los personajes dentro del juego de mesa.

No existe ninguna herramienta para dar soporte a los jugadores durante la partida para el juego de mesa "Krosmaster".

3. Marco de desarrollo

3.1 Android

Se desarrollará una aplicación para Android, ya que se quiere llegar al máximo público posible con el menor coste de inversión posible, y, como podemos ver en este gráfico publicado por IDC (International Data Corporation) el 85% de los usuarios de móvil tienen un sistema operativo Android. [6]

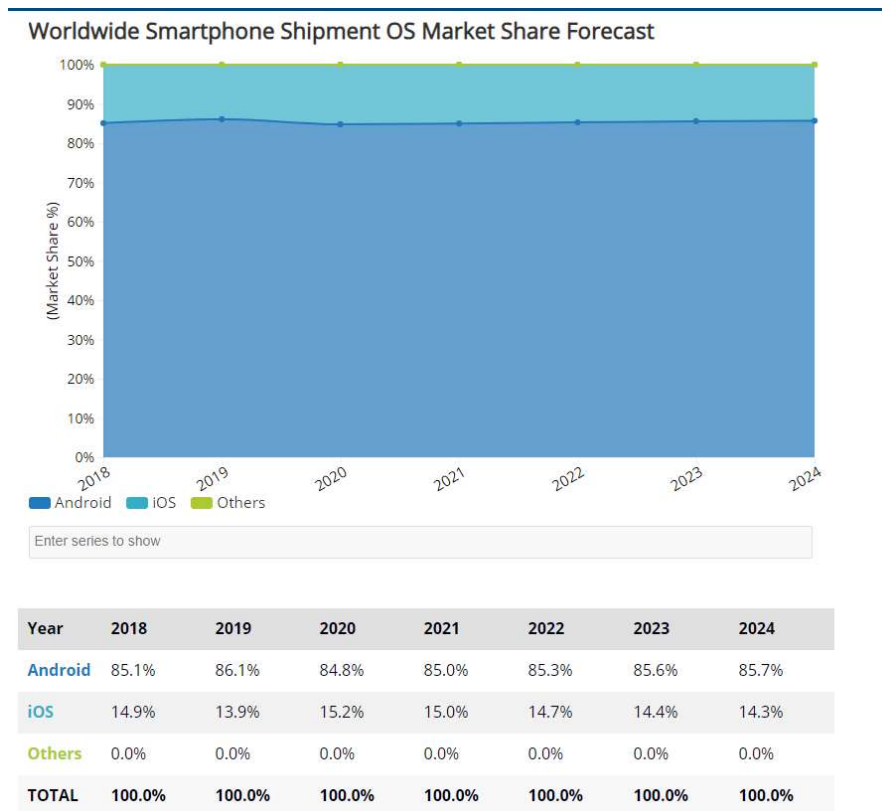


Tabla 1: Gráfico y tabla de utilización de los diferentes sistemas operativos

Pero ¿qué es Android? Android es un sistema operativo móvil basado en núcleo Linux y otros software de código abierto. Fue diseñado para dispositivos móviles con pantalla táctil, como teléfonos inteligentes, tabletas, relojes inteligentes (Wear OS), automóviles con otros sistemas a través de Android Auto, al igual los automóviles con el sistema Android Automotive y televisores Leanback.[7]

3.2 Entorno de desarrollo: Android Studio

Para desarrollar este proyecto haremos uso del IDE oficial de Google junto con JetBrains basado en IntelliJ. [18] para el desarrollo de Android, Android Studio, El cual incluye todo lo necesario para compilar apps para Android [8]. Algunas de las funcionalidades por las que se ha decidido usar este entorno de desarrollo frente a otras opciones son [9]:

- Apply Changes: permite realizar cambios en el código y en los recursos de una app en ejecución sin tener que reiniciarla
- Editor de código inteligente
- Emulador de Android con optimización para todos los dispositivos: Es capaz de emular un gran rango de dispositivos Android con diferentes versiones de instalación
- Integración con herramientas de control de versiones: como GitHub

3.3 Firebase

Firebase es una plataforma de aplicaciones unificada para Android, iOS y desarrollo web móvil.

Actualmente se está expandiendo y está agregando nuevas herramientas para ayudar a realizar un desarrollo más rápido, mejorando la calidad de las aplicaciones [19].

3.3.1 Servicios

Los servicios de los que provee son:

- Firebase Analytics: Es un servicio que permite a los desarrolladores entender como los usuarios están usando la aplicación a partir de un SDK que captura eventos y propiedades por si solo y a la vez permite obtener datos customizados.
- Firebase Cloud Messaging: Es un servicio que permite enviar mensajes y notificaciones para Android, Aplicaciones web e IOS.
- Firebase Auth: Es un servicio que permite autenticar usuarios usando solo código cliente. Soporta proveedores de login como Facebook, Google, Github y Twitter. También incluye un sistema gestor de usuarios.

- Real-time Database: Es un servicio que permite tener una Base de datos a tiempo real sincronizada a través de todos los usuarios y almacenada en la nube. Permite integración con Android, IOS y aplicaciones JavaScript.
- Firebase Storage: Es un servicio que incrementa la seguridad y facilita la transferencia de ficheros sin importar la calidad de la red.
- Firebase Test Lab for Android: Es un servicio que provee de una infraestructura basada en la nube para el testeo de aplicaciones Android.
- Firebase Crash Reporting: Es un servicio que permite consultar los reportes de errores creados en la aplicación de forma detallada.
- Firebase Notifications: Es un servicio que permite enviar notificaciones para cierto objetivo de usuarios.

Para desarrollar la aplicación se usarán tres de estos servicios:

3.3.1 Firebase Authentication

Firebase Authentication proporciona servicios de backend, SDK fáciles de usar y bibliotecas de IU ya elaboradas para autenticar a los usuarios en su aplicación. [20]

Firebase Authentication admite la autenticación mediante contraseñas, números de teléfono, proveedores de identidad federada populares, como Google, Facebook y Twitter, y aprovecha estándares de la industria como OAuth 2.0 y OpenID Connect.

Hay dos formas de permitir que los usuarios accedan a la app a través de Firebase Auth; ya sea con FirebaseUI Auth como una solución completa o mediante el SDK de Firebase Authentication. En el caso de este proyecto se ha usado el SDK de Firebase Authentication.

La autenticación del SDK de Firebase tiene cinco funciones clave:

- Autenticación basada en correo electrónico y contraseña: Autentica a los usuarios con sus direcciones de correo electrónico y contraseñas.
- Integración con proveedores de identidad federada: Autentica usuarios mediante la integración con proveedores de identidad federada como Google, Facebook o Twitter
- Autenticación con número de teléfono: Envía mensajes SMS a los teléfonos de los usuarios para autenticarlos.
- Integración de sistemas de autenticación personalizados: Conecta el sistema de acceso existente de una app al SDK de Firebase Authentication ofreciéndole acceso a Firebase Realtime Database y otros servicios de Firebase.

- Autenticación anónima: Crea cuentas anónimas temporales para permitir el uso de funciones que requieran autenticación sin exigir que los usuarios accedan primero.

En este proyecto, solo se ha usado la primera de las cinco funciones clave de la autenticación del SDK de Firebase " Autenticación basada en correo electrónico y contraseña":

Esta funcionalidad proporciona métodos a fin de crear y administrar usuarios que utilizan sus direcciones de correo electrónico y contraseñas para acceder, así como maneja el envío de correos electrónicos para restablecer la contraseña.

La forma en que funciona es:

1. La aplicación obtiene las credenciales de autenticación del usuario (estas pueden ser el correo y la contraseña o un token OAuth)
2. La aplicación pasa estas credenciales al SDK de Firebase Authenticator
3. Los Servicios backend de Firebase verifican las credenciales y muestran una respuesta al cliente

El motivo por el que se usara el servicio Firebase Authentication es porque es necesario para poder hacer uso de los servicios de Firebase Realtime Databases y Firebase Storage.

3.3.2 Firebase Realtime Database

Firebase Realtime Database es una base de datos alojada en la nube. Los datos se almacenan en formato JSON y se sincronizan en tiempo real con cada cliente conectado. Cuando se compilan aplicaciones multiplataforma con SDK de iOS, Android y JavaScript, todos los clientes de estas comparten una instancia de Realtime Database y reciben actualizaciones automáticamente con los datos más recientes. [10]

Firebase Realtime Database tiene una forma de almacenar datos NoSQL, lo que significa que no utiliza tablas para guardar los datos, sino que utiliza JSON y que no utiliza Query para insertar, actualizar, eliminar o añadir información. Es considerada la parte back-end de un sistema usado para almacenar datos en una base de datos. [14][15]

Sus Funciones clave son:

- Sincronización de datos en tiempo real para todos los dispositivos conectados

- Funcionamiento con conexiones inestables guardando los cambios en el disco al perder la conexión y sincronizando al renovar la conexión
- Acceso desde dispositivos cliente sin necesidad de un servidor de aplicaciones
- Escalamiento en varias bases de datos

La forma en que funciona es permitiendo acceso a la base de datos directamente desde el código del cliente. Estos datos persisten de forma local si se pierde la conexión sincronizando los datos al recuperarla.

Firebase Realtime Database proporciona reglas de seguridad para definir como se deberían estructurar los datos y en qué momento se pueden leer o escribir.

El motivo por el que se usara el servicio Firebase Realtime Database en lugar de una base de datos SQL es porque al ser la principal aplicación del proyecto la creación de un sistema para la automatización de procesos de una partida de Krosmaster en tiempo real, es una forma muy interesante de implementarlo que facilita la comunicación en tiempo real.

3.3.3 Firebase Storage

Firebase Storage es un servicio de almacenamiento de objetos creado para almacenar y entregar contenido generado por usuarios como fotos o videos.

Los SDK de Firebase para Cloud Storage agregan la seguridad de Google a la carga y descarga de archivos de las aplicaciones que usen este servicio sin importar la calidad de la red.

Estos archivos se almacenan en un bucket de Google Cloud Storage haciéndolos accesibles a través de Firebase y Google Cloud. Además, se pueden procesar en el servidor, realizando filtrado de imágenes o transcodificación de video con las API de Google Cloud Storage. También se puede configurar los controles de acceso de manera que los archivos puedan ser públicos o privados.

El motivo por el que se usará el servicio de Firebase Storage es porque es la forma más cómoda y eficiente de acceder a imágenes a través de Firebase Realtime Database, que es la base de datos que se usara para este proyecto.

4. Objetivos y alcance

4.1 Detalles de objetivos

Se creará una aplicación móvil, ya que lo que se pretende es cambiar el formato de juego actual en el que los jugadores deben llevar y controlar:

1. Tablero
2. Orden de turnos (cual jugador va primero y cual después)
3. Orden de miniaturas jugables (qué figura va primera y cuál última, en que turno se está actualmente)
4. Cartas (de una a seis por jugador)
5. miniaturas jugables (de una a seis por jugador)
6. Galones de Gloria (seis por cada jugador más un extra en medio)
7. Modificadores de atributos (Puntos de vida, Puntos de movimiento, Puntos de acción, Puntos de alcance)
8. Dados (tres para jugador mínimo)
9. Moneda del juego (variable, se suelen acumular hasta treinta monedas individuales)
10. Tienda de objetos (tiene ciertas limitaciones de cuántos objetos pueden estar a la venta al mismo tiempo)
11. Objetos comprados (cada personaje puede llevar hasta dos)

A uno donde los jugadores sólo tengan que llevar y controlar:

1. Tablero
2. miniaturas jugables
3. Dispositivo móvil que realizará todo el resto de control

Por lo tanto, lo que se pretende es disminuir la necesidad de material necesario para llevar a cabo la actividad de jugar, así como disminuir la carga de atención que los jugadores necesitan para tener controlado todo lo que está pasando, pudiéndose concentrar así únicamente en la estrategia.

4.2 Objetivos del producto

- Crear una herramienta de seguimiento de la partida para evitar el error humano
- Crear una herramienta que disminuya la cantidad de elementos que es necesario llevar para la realización de una partida
- Crear un catálogo donde ver todas las miniaturas coleccionables

4.3 Objetivos del cliente

- Limitar el error Humano
- Facilitar la gestión de los elementos del juego

4.4 Público potencial

En primer grado:

- Todas aquellas personas que juegan al juego de mesa Krosmaster

En segundo grado:

- Todas aquellas personas que juegan y están interesados en juegos de mesa.
- Todas aquellas personas que consumen alguno de los productos de la compañía Ankama y están interesados en probar nuevos productos relacionados con el universo y con una jugabilidad similar.
- Coleccionistas que hagan colección de las miniaturas sin jugar al juego y quieran un catálogo donde ver todas las miniaturas existentes.

5. Metodología

Se utilizará una metodología de desarrollo en cascada, Una metodología que ordena las etapas del proceso de software de tal forma que, para iniciar una etapa, la etapa anterior debe estar finalizada. Al final de cada etapa, se hará una revisión final conforme el proyecto está listo para avanzar a la siguiente etapa. [5]

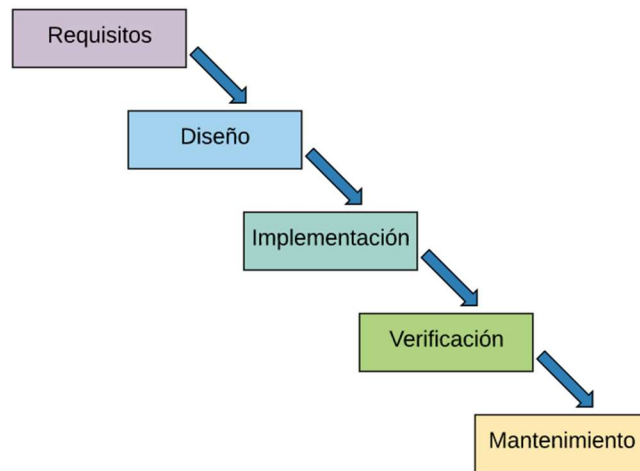


Figura 21: Figura del funcionamiento de la metodología de desarrollo en cascada

5.1 Planificación



Figura 22: Representación del ciclo de vida del TFG

Tarea	Fecha de inicio	Fecha de finalización	Tiempo
Creación del anteproyecto	08/01/2021	22/01/2021	50 horas
Contratar plataformas necesarias (Host, BBDD, Servidor HTTP)	22/01/2021	25/01/2021	5 horas
Diseño	25/01/2021	27/02/2021	100 horas
Programación	01/03/2021	20/04/2021	150 horas
Documentación	21/04/2021	07/05/2021	50 horas
Testing (durante este período se trabajará seis horas al día)	30/04/2021	04/06/2021	150 horas
Despliegue	04/06/2021	07/06/2021	5 horas
Análisis de objetivos	07/06/2021	18/06/2021	40 horas
Total			550 horas

Tabla 2: Tabla de planificación del TFG

6. Requisitos funcionales y tecnológicos

A continuación una lista de los requisitos funcionales y la solución tecnológica a estos.

Requisitos	Solución tecnológica
Conectar a internet	Contratar una línea a internet con un router o a través de la tarifa de nuestro móvil
Intercambiar información entre dispositivos a tiempo real	Contratar servidores a ionos.es y configurarlos para hacer esta función
Almacenar información	Contratar servidores a ionos.es y configurarlos para hacer esta función
Descargar la aplicación	Publicar la aplicación a Play Store
Gestión de figuras coleccionables	Desarrollo de un almacén de figuras coleccionables
Gestión del orden de turnos y de figuras	Desarrollo de funcionalidad que gestione el orden de turnos y figuras
Gestión de cartas (con sus hechizos, poderes, PdV...)	Desarrollo de funcionalidad que gestione las cartas
Gestión de los recursos del juego (GG, monedas, objetos, modificadores de atributos)	Desarrollo de funcionalidad que gestione los recursos del juego

Tabla 3: Requisitos funcionales y Tecnológicos y Soluciones Tecnológicas

7. Desarrollo

7.1 Casos De uso

Registro de usuario

Descripción: El usuario introduce sus datos para registrarse en la aplicación

Actor principal: Usuario de la aplicación

Precondición: La aplicación está instalada en el dispositivo móvil del usuario y el usuario está en la pantalla de registro / Inicio de sesión

Flujo:

1. El usuario inicia el caso de uso haciendo click sobre el botón "Registrarse"
 2. El sistema redirige al usuario a la pantalla de registro
 3. El usuario llena todos los campos necesarios
 4. El usuario hace click sobre el botón "Registrarse"
 - 5.1 El sistema comprueba que la información de todos los campos es correcta
 - 5.2 En caso de que el usuario no haya rellenado todos los campos necesarios el sistema informa al usuario que es necesario rellenar todos los campos para registrarse y redirige al usuario al paso 3 del flujo
 - 6.1. El sistema crea el usuario y lo redirige a la pantalla principal
 - 6.2. En caso de que la información de todos los campos no sea correcta, el sistema informa a el usuario que el campo incorrecto no está rellenado correctamente y lo redirige al paso 3 del flujo
7. Finaliza el caso de uso.

Postcondición: El usuario se ha registrado correctamente en la aplicación

Inicio de Sesión del usuario

Descripción: El usuario introduce los datos necesarios para iniciar sesión

Actor principal: Usuario de la aplicación

Precondición: El usuario ya está registrado en la aplicación y el usuario está en la pantalla de registro / Inicio de sesión

Flujo:

1. El usuario inicia el caso de uso haciendo click sobre el botón "Iniciar sesión"
2. El sistema redirige al usuario a la pantalla de iniciar sesión
3. El usuario llena todos los campos necesarios
4. El usuario hace click sobre el botón "Iniciar sesión"
5. El sistema comprueba que la información de todos los campos es correcta
- 6.1. En caso de que el usuario no haya rellenado todos los campos necesarios el sistema informa a el usuario que es necesario rellenar todos los campos para registrarse y lo devuelve al paso 3 del flujo
- 6.2. En caso de que la información de todos los campos no sea correcta, el sistema informa a el usuario que el Inicio de sesión no se ha podido realizar ya que uno o varios campos son incorrectos y lo devuelve al paso 3 del flujo
- 6.3. El sistema redirige al usuario a la pantalla principal
7. Finaliza el caso de uso.

Postcondición: El usuario inicia sesión correctamente y es redirigido a la pantalla principal

Consultar Figuras

Descripción: El usuario consulta alguna de las figuras

Actor principal: Usuario de la aplicación

Precondición: El usuario ha iniciado sesión y está en la página principal

Flujo:

1. El usuario inicia el caso de uso haciendo click sobre el botón "Lista de figuras"
2. El sistema redirige al usuario a la pantalla de lista de figuras

3. El usuario despliega la temporada de la figura que quiere consultar
4. El usuario selecciona la figura que quiere consultar
5. El sistema redirige al usuario a la pantalla de esa figura
6. Finaliza el caso de uso.

Postcondición: El usuario está en la pantalla de la figura que quería consultar

Crear Equipo

Descripción: El usuario crea un equipo para poder realizar una partida

Actor principal: Usuario de la aplicación

Precondición: El usuario ha iniciado sesión y está en la página principal

Flujo:

1. El usuario inicia el caso de uso haciendo click sobre el botón "Crear equipo"
2. El sistema redirige al usuario a la pantalla de creación de equipo
- 3.1. El usuario hace click en el primer espacio de equipo libre
- 3.2. El usuario hace click en un espacio de equipo ocupado
 - 3.2.1. El sistema muestra la opción de eliminar esa figura del equipo
 - 3.2.2. El usuario hace click en eliminar
 - 3.3.3. El sistema elimina la figura del equipo
4. El sistema muestra una lista con todas las figuras
5. El usuario hace click sobre la figura que le interesa
6. El sistema muestra la opción de consultar y la de añadir
 - 7.1. El usuario hace click en "añadir"
 - 7.2. El usuario hace click en "consultar"
 - 7.2.1 El sistema muestra la información de la figura

8. El sistema comprueba si al añadir esa figura se supera el límite de nivel del equipo y en caso de que así sea informa al usuario

9. El sistema añade la figura (incluso si se supera el límite de nivel)

10.1. El usuario hace click en el botón "Finalizar"

10.2. El usuario hace click en el primer espacio de equipo libre,

10.2.1. El sistema retorna al usuario al paso 4

11. El sistema añade el equipo a los equipos del usuario

12. Finaliza el caso de uso

Postcondición: El equipo es creado correctamente

Consultar Equipos

Descripción: El usuario consulta alguno de los equipos creados que tiene

Actor principal: Usuario de la aplicación

Precondición: El usuario ha iniciado sesión y está en la página principal

Flujo:

1. El usuario inicia el caso de uso haciendo click sobre el botón "Equipos"

2. El sistema redirige al usuario a la pantalla de equipos

3. El usuario hace click sobre el equipo que quiere consultar

4. El sistema despliega la lista con los personajes que conforman ese equipo

5. Finaliza el caso de uso

Postcondición: El usuario ha podido consultar su equipo

Modificar Equipo

Descripción: El usuario modifica alguno de los equipos creados que tiene

Actor principal: Usuario de la aplicación

Precondición: El usuario ha iniciado sesión y está en la página principal

Flujo:

1. El usuario inicia el caso de uso haciendo click sobre el botón "Equipos"
2. El sistema redirige al usuario a la pantalla de equipos
3. El usuario hace click sobre el equipo que quiere consultar
4. El sistema despliega la lista con los personajes que conforman ese equipo
5. El usuario hace click sobre el botón editar equipo
6. El sistema manda al usuario a la pantalla de crear equipo con la información del equipo cargada
7. El sistema manda al usuario al paso 3 del flujo del caso de uso "crear equipo"
8. Finaliza el caso de uso

Postcondición: El usuario ha podido modificar su equipo

Crear Partida

Descripción: El usuario crea una partida

Actor principal: Usuario de la aplicación

Precondición: El usuario ha iniciado sesión y está en la página principal

Flujo:

1. El usuario inicia el caso de uso haciendo click sobre el botón "Crear Partida"
- 2.1. El sistema redirige al usuario a la pantalla de seleccionar equipo
- 2.2. El sistema informa al usuario que no tiene equipos creados y lo redirige a la pantalla de crear equipo
3. El usuario selecciona el equipo con el que quiere jugar
4. El sistema muestra un código al usuario para que el segundo jugador pueda unirse a la partida
5. Finaliza el caso de uso

Postcondición: El usuario 2 se une a la partida

Unirse a partida

Descripción: El usuario se une a una partida de otro usuario

Actor principal: Usuario de la aplicación

Precondición: El otro usuario ha completado el caso de uso crear partida y le ha proporcionado el código al usuario

Flujo:

1. El usuario inicia el caso de uso haciendo click sobre el botón "Unirse a partida"
- 2.1. El sistema redirige al usuario a la pantalla de seleccionar equipo
- 2.2. El sistema informa al usuario que no tiene equipos creados y lo redirige a la pantalla de crear equipo
3. El usuario selecciona el equipo con el que quiere jugar
4. El sistema redirige al usuario a la pantalla de unirse a partida
5. El usuario introduce el código que le ha proporcionado el otro usuario
6. Ambos usuarios son redirigidos a la página de partida
7. Finaliza el caso de uso

Postcondición: se ha creado la partida

Jugar Partida

Descripción: Dos usuarios (jugador 1 y jugador 2) juegan una partida

Actor principal: jugador 1 (J1) y jugador 2 (J2)

Precondición: Ambos usuarios están en la pantalla de partida

Flujo:

1. El sistema inicia el caso de uso al entrar ambos usuarios en la página de partida
2. El sistema calcula que jugador empieza

3. El sistema calcula el orden de la línea temporal de cada usuario
4. El sistema reparte seis GG a cada jugador
5. El sistema inicia el juego para la figura uno del jugador uno
- 6.1. El sistema gira una recompensa demoníaca de cada tipo si es el turno dos o tres
- 6.2. El sistema no hace nada si no es el turno dos o tres
- 7.1. El sistema inicia el caso de uso Lanzamiento de tensión si no es el turno uno
- 7.2. Si es el turno uno el sistema no hace nada
8. El sistema inicia el turno de juego para la figura a la que le toca
9. El sistema activa los efectos provocados "al comienzo del turno"
- 10.1. El usuario inicia el caso de uso Mover una casilla
- 10.2. El usuario inicia el caso de uso Recoger un Kama
- 10.3. El usuario inicia el caso de uso Mover una casilla
- 10.4. El usuario inicia el caso de uso Comprar un GG
- 10.5. El usuario inicia el caso de uso Lanzar un hechizo
- 10.6. El usuario inicia el caso de uso Comprar una recompensa demoníaca
- 10.7. El usuario inicia el caso de uso Utilizar una recompensa demoníaca
- 11.1. El usuario vuelve a escoger cualquiera de las opciones del paso 9
- 11.2. El usuario hace click en "terminar fase de activación"
12. El sistema desactiva los efectos que tienen como duración "este turno"
13. El sistema inicia el turno para la siguiente figura del jugador
- 14.1. Si el jugador tiene una figura jugable, el sistema cambia el turno a la próxima figura jugable y envía al jugador al paso 8 del flujo
- 14.2. Si el jugador no tiene ninguna figura jugable, el sistema cambia el turno al próximo turno y cambia el jugador al próximo jugador y envía al jugador al paso 6 del flujo

Nota. La finalización del caso de uso solo se puede dar desde el subcaso de uso "Lanzar un hechizo"

Postcondición: Ambos jugadores han podido jugar una partida

Lanzamiento de tensión

Descripción: El sistema tira los dados y el usuario decide que hacer con ellos

Actor principal: Sistema

Precondición: Es el principio del turno del jugador y no es el turno uno

Flujo:

1. El sistema inicia el caso de uso
2. El sistema tira dos dados para el jugador uno
3. El sistema comprueba que los dados no son los mismos y en caso de que lo sean, resta un GG a cada jugador
4. El jugador uno decide que opción quedarse de los dados si es que tiene que decidir alguna
5. El jugador decide si quiere quedarse con alguno de los dados o los dos
6. Si el jugador decide quedarse con algún dado, el sistema le hace escoger a que figura asignarlo
7. Si el jugador decide no quedarse con algún dado, el sistema le retira los dados y le da la compensación en Kamas
8. Finaliza el caso de uso

Postcondición: El usuario ha decidido qué hacer con los dados

Mover una casilla

Descripción: El usuario se mueve un número determinado de casillas

Actor principal: Usuario de la aplicación

Precondición: El usuario ha hecho click en "mover una casilla"

Flujo:

1. El usuario inicia el caso de uso decidiendo mover una casilla
2. El sistema le pide al usuario cuantas casillas se quiere mover
3. El usuario selecciona un numero desde uno a los PM que tenga actualmente
4. El sistema resta los PM gastados a la figura
5. Finaliza el caso de uso

Postcondición: El sistema ha restado los PM gastados al usuario

Recoger un Kama

Descripción: El usuario recoge un Kama

Actor principal: Usuario de la aplicación

Precondición: El usuario ha hecho click en "Recoger un Kama"

Flujo:

1. El usuario inicia el caso de uso decidiendo recoger un Kama
2. El sistema comprueba que el usuario tenga como mínimo un PA para recoger un Kama
- 3.1. En caso de que pueda recoger un Kama, el sistema resta un PA a ese personaje y suma un Kama al jugador
- 3.2. En caso de que no pueda recoger un Kama, el sistema informa al jugador que no ha podido recoger un Kama
5. Finaliza el caso de uso

Postcondición: El usuario ha recogido un Kama si era posible

Comprar un GG

Descripción: El usuario compra un GG

Actor principal: Usuario de la aplicación

Precondición: El usuario ha hecho click en "comprar un GG"

Flujo:

1. El usuario inicia el caso de uso decidiendo comprar un GG
2. El sistema comprueba que el usuario sea capaz de comprar un GG
- 3.1. Si el usuario tiene diez Kamas, el sistema incrementa en uno los GG del jugador y decrementa en diez los Kamas del jugador.
 - 3.1.1. El sistema comprueba si el Kama salvaje aún está en juego, de ser así, añade un GG adicional al jugador.
- 3.2. Si el jugador no tiene diez Kamas, el sistema informa al usuario que no tiene suficientes Kamas para comprar un GG
4. Finaliza el caso de uso

Postcondición: El usuario ha comprado un GG si era posible

Lanzar un hechizo

Descripción: El usuario lanza un hechizo

Actor principal: Usuario de la aplicación

Precondición: El usuario ha hecho click en lanzar un hechizo

Flujo:

1. El usuario inicia el caso de uso decidiendo lanzar un hechizo
2. El sistema muestra la lista de hechizos disponibles para esa figura
3. El usuario hace click en uno de los hechizos
4. El sistema muestra los detalles de ese hechizo
5. El usuario hace click en lanzar hechizo
6. El sistema comprueba que el usuario tenga los PA,PM,PdV necesarios para lanzar el hechizo y que dentro de las restricciones de lanzamiento de hechizo según el color, pueda lanzar ese hechizo.
 - 7a. En caso de que no pueda lanzar el hechizo, el sistema informa al usuario que no puede lanzar el hechizo y vuelve al paso 2 del caso de uso

7. El sistema muestra la lista de figuras enemigas
8. El usuario decide a que figura quiere lanzar un hechizo
9. El sistema lanza los dados necesarios acorde a los efectos del hechizo y los poderes de la figura atacante y defensora
10. El sistema calcula los PdV/PA/PM/Alcance restantes de ambas figuras
11. El sistema muestra a ambos jugadores el resultado de los dados y del uso del hechizo
12. El sistema comprueba si la figura aún tiene suficientes PdV para seguir jugando
 - 12.1. Si la figura no tiene suficientes PdV, el sistema marca esa figura como muerta. El sistema resta GG acorde al nivel de esa figura al jugador objetivo mientras que suma GG acorde al nivel de esa figura.
 - 12.1.1 El sistema calcula los GG de ambos jugadores, si uno de los dos tiene 0 GG, finaliza el caso de uso de jugar partida y ese jugador es derrotado.
 - 12.1.2 El sistema comprueba que ambos jugadores tengan al menos una figura viva, si no si uno de los dos no tiene ninguna figura viva, finaliza el caso de uso de jugar partida y ese jugador es derrotado.
 - 12.2. Si la figura tiene suficientes PdV, el sistema no hace nada
13. Finaliza el caso de uso

Postcondición: El usuario ha lanzado el hechizo si era posible

Comprar una Recompensa Demoníaca

Descripción: El usuario compra una recompensa demoníaca

Actor principal: Usuario de la aplicación

Precondición: El usuario ha hecho click en comprar una recompensa demoníaca

Flujo:

1. El usuario inicia el caso de uso decidiendo comprar una recompensa demoníaca
2. El sistema muestra al usuario las recompensas demoníacas visibles actualmente, así como la opción de comprar una aleatoria

3. El usuario hace click en una recompensa demoníaca
4. El sistema muestra los detalles de la recompensa demoníaca y de al usuario la opción de comprarla
5. El usuario hace click en "comprar"
6. El sistema comprueba que el usuario tenga las Kamas suficientes y los PA suficientes para comprar la recompensa demoníaca
- 7.1. El sistema añade la recompensa demoníaca al usuario y disminuye las Kamas del usuario acorde con el precio de la recompensa demoníaca y los PA del usuario en uno. Si la recompensa demoníaca era aleatoria, esta quedará oculta para el otro jugador.
- 7.2. El sistema informa al usuario que no se ha podido realizar la compra
8. Finaliza el caso de uso

Postcondición: El usuario ha comprado la recompensa demoníaca si es posible

Utilizar una Recompensa Demoníaca

Descripción: El usuario utiliza una recompensa demoníaca

Actor principal: Usuario de la aplicación

Precondición: El usuario ha hecho click en utilizar una recompensa demoníaca

Flujo:

1. El usuario inicia el caso de uso decidiendo utilizar una recompensa demoníaca
2. El sistema muestra las recompensas demoníacas que tiene actualmente el usuario
3. El usuario selecciona una de las recompensas demoníacas
4. El sistema comprueba que la figura del turno actual no termine con más recompensas demoníacas que su nivel
- 5.1. El sistema equipa en la miniatura del turno actual la recompensa demoníaca y modifica sus características en función de lo que haga esa recompensa demoníaca. Si la recompensa

demoníaca estaba oculta, ahora queda descubierta para que ambos jugadores puedan consultarla.

5.2. El sistema informa al jugador que esa figura ya no puede tener más recompensas demoníacas equipadas.

6. Finaliza el caso de uso

Postcondición: El usuario ha equipado la recompensa demoníaca si es posible

7.2 Estructura del proyecto

El proyecto está estructurado siguiendo el patrón por defecto de Android Studio. La parte de programación está en la ruta "Nombre del proyecto" (en este caso AutoKrosmaster)/app/src/main

En este punto el proyecto se divide en la parte de backend en la ruta /java/"nombre del paquete" (en este caso com.example.autokrosmaster) y en la parte de frontend en la ruta /res

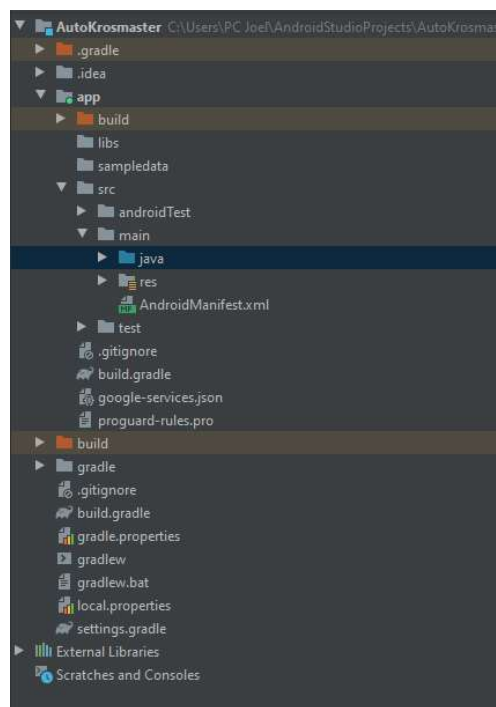


Figura 23: Estructura del proyecto

La parte de backend sobre la que se ha trabajado en este proyecto se divide en:

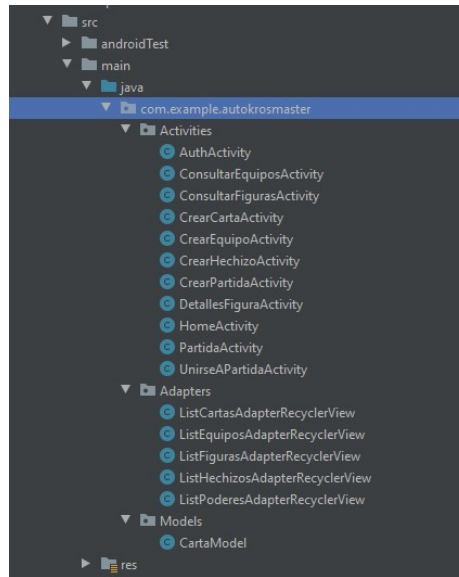


Ilustración 24: Estructura de backend

- Activities: Contiene código en java de todas las actividades que implementan la lógica de la aplicación
- Adapters: Contiene código en java todos los adaptadores para las RecyclerViews y la lógica para estas mismas
- Models: Contiene código en java para las clases propias que se han usado para el proyecto

La parte de frontend sobre la que se ha trabajado en este proyecto se divide en:

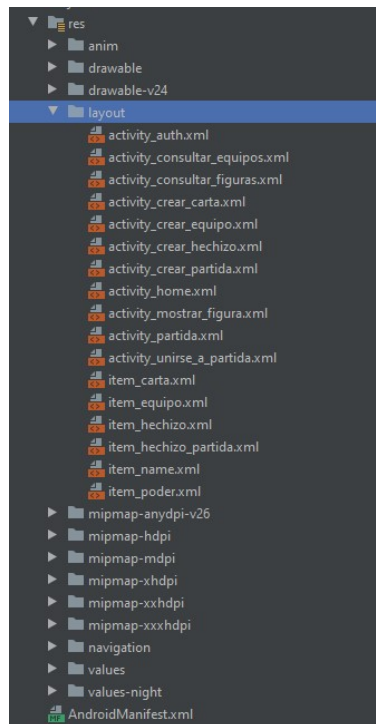


Ilustración 25: Estructura Frontend

- anim: contiene archivos .xml para el código para las animaciones de la aplicación
- drawable: contiene todas las imágenes en formato .png necesarias para la aplicación, así como la configuración para bordes y fondos que se usan en esta
- layout: contiene archivos .xml para el diseño de las distintas pantallas así como los objetos de los RecyclerViews

Cada activity de backend esta asociada con su layout de la siguiente forma:

"Nombre"Activity.java = activity_nombre.xml

Las mayúsculas dentro del nombre se convierten en _minúscula excepto si están en la primera letra que solo se convierten en minúscula.

7.3 Interfaz Grafica

7.3.1 Diseño de las pantallas de la aplicación

Diseño de las pantallas de la aplicación y como interaccionan entre ellas con el formato final en las pantallas en lugar de esbozo para una mejor comprensión.

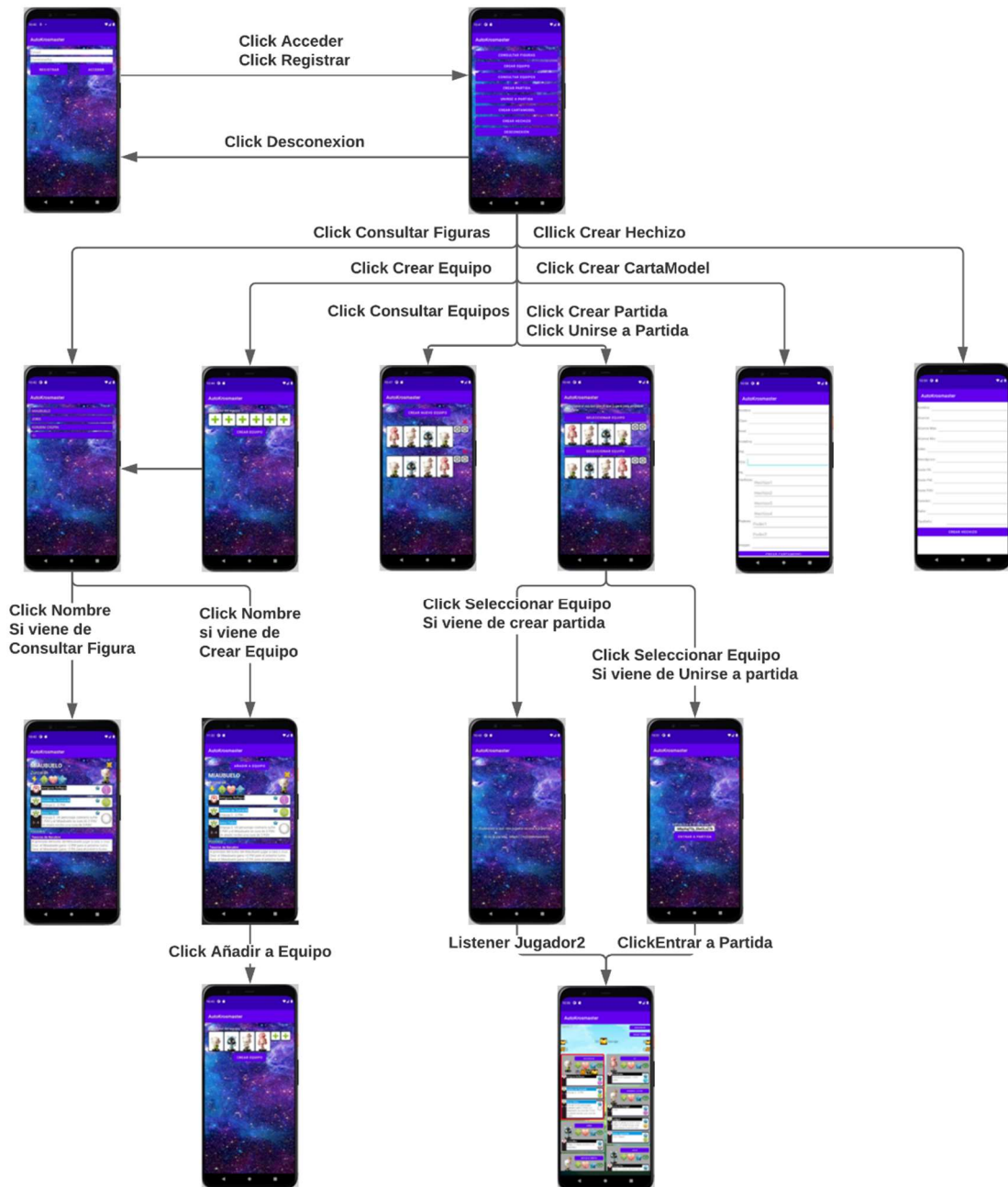


Figura 26: Diseño de pantallas

7.3.2 Pantalla de Registro / Acceso

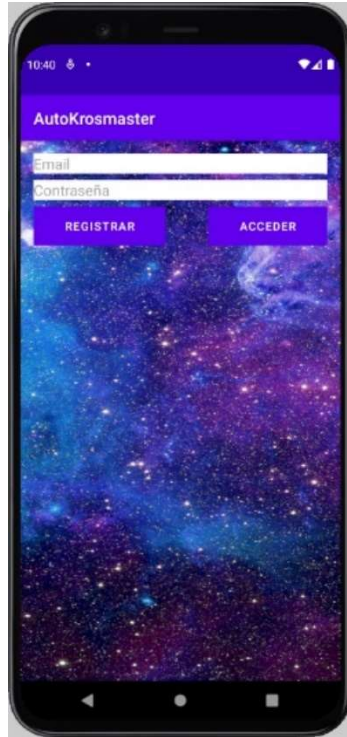


Figura 27: Pantalla de Registro / Acceso

Esta es la pantalla principal a la que será dirigido el usuario una vez se instale la aplicación. podrá utilizar un email y una contraseña para registrarse. La gestión del registro será usando Firebase Authentication. En caso de que el registro / acceso sea correcto, el usuario será redirigido a la pantalla inicial. En caso contrario, aparecerá una ventana emergente informándole de cual ha sido el error.

Esta pantalla sólo será accesible en caso de que el usuario no esté autenticado. En caso de que ya esté autenticado, el usuario será redirigido automáticamente a la pantalla inicial.

7.3. Pantalla Inicial



Figura 28: Pantalla Inicial

Esta pantalla es la pantalla principal de la aplicación, desde ella los usuarios podrán:

- Acceder a consultar figuras
- Acceder a crear equipo
- Acceder a consultar equipos
- Acceder a crear partida
- Acceder a unirse a partida

Los usuarios administradores podrán también:

- Acceder a crear cartamodel
- Acceder a crear hechizo

Finalmente, los usuarios podrán:

- Desconectarse de su cuenta para volver a la pantalla de registro / acceso

7.3. Pantalla de consultar equipos



Figura 29: Pantalla de consultar equipos

En esta pantalla los usuarios podrán acceder a una lista de los nombres de todas las figuras registradas en la base de datos.

Los usuarios podrán hacer clic en cada uno de estos nombres para abrir la pantalla de figura de cada uno de ellos.

7.3. Pantalla de figura

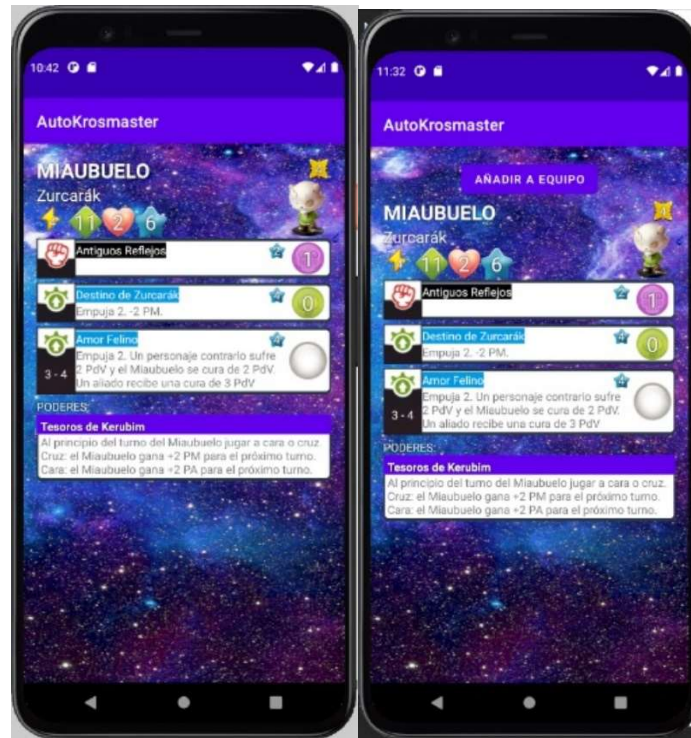


Figura 30: Pantalla de figura

En esta pantalla se mostrarán todos los detalles de una figura seleccionada previamente. Estos detalles son:

- Una imagen de la figura
- El nombre
- La raza
- El nivel
- La iniciativa
- Los PM
- Los PdV
- Los PA
- La lista de hechizos
- La lista de poderes

Cada hechizo contendrá:

- El tipo de Lanzamiento
- El alcance
- El nombre

- El color
- La descripción
- El coste (tanto de PA, PM o PdV) mostrando sólo los campos que sean necesarios
- El daño o la curación realizada
- El tipo de daño o curación (si es que tiene)

Si el usuario provenía de la actividad "crear equipo" la pantalla de figura mostrará un campo extra "añadir a equipo" que devolverá al usuario a la pantalla de crear equipo y llenará la figura que se quería seleccionar en el campo correspondiente.

7.3. Pantalla de crear equipo

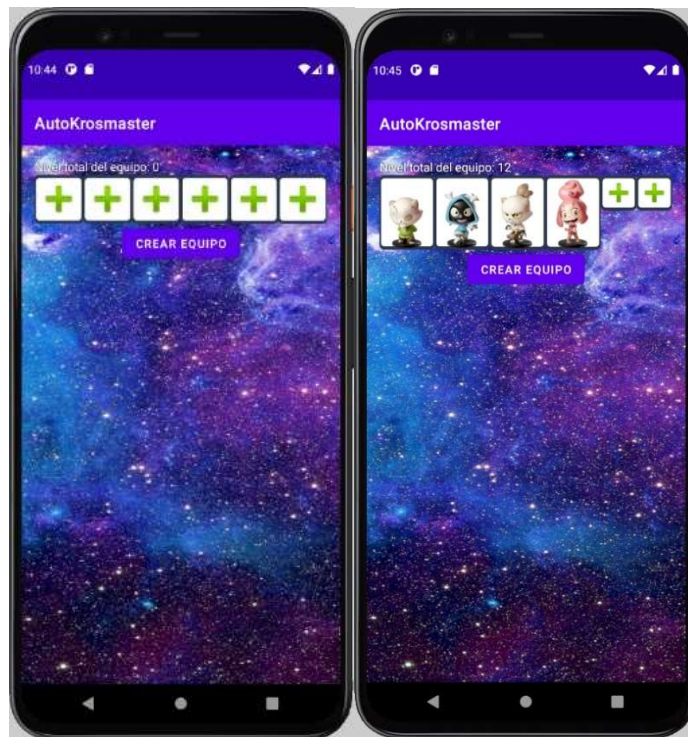


Figura 31: Pantalla de crear equipo

Esta pantalla permitirá a los usuarios crear un equipo de máximo 6 personajes (según las normas establecidas).

En todo momento se mostrará el nivel total del equipo.

Al hacer clic en los iconos de "+" el usuario será redirigido a la pantalla de selección de figura.

Una vez el usuario seleccione la figura, esta será colocada en la posición correspondiente.

En caso de que el nivel total del equipo sea mayor que 12, el texto de nivel total del equipo cambiará a rojo.

Finalmente, el usuario podrá hacer click en el botón "crear equipo" para añadir ese equipo a su lista de equipos jugables. En caso de que el nivel total del equipo sea mayor que 12, el equipo no se creara y el usuario será informado a través de una ventana emergente que el nivel del equipo no puede ser mayor de 12.

7.3. Pantalla de consultar equipo



Figura 32: Pantalla de consultar equipo

En esta pantalla el usuario verá una lista con todos los equipos que ha creado anteriormente.

El usuario podrá dirigirse a la pantalla de crear equipo pulsando el botón "crear nuevo equipo".

El usuario podrá eliminar alguno de sus equipos haciendo click en la cruz roja en la esquina superior derecha de cada uno de ellos.

7.3. Pantalla de seleccionar equipo



Figura 33: Pantalla de seleccionar equipo

En esta pantalla el usuario podrá seleccionar alguno de los equipos que ya tiene creados para jugar con él.

En caso de que se provenga de la actividad crear partida será dirigido a la Pantalla de crear partida.

En caso de que se provenga de la actividad unirse a partida será dirigido a la Pantalla de unirse a partida.

7.3. Pantalla de crear partida



Figura 34: Pantalla de crear partida

Esta pantalla mostrará al usuario un id que tiene que compartir con otro usuario para poder iniciar la partida. La actividad quedará a la espera hasta que otro usuario acceda a la partida y, cuando eso pase, el usuario será redirigido a la pantalla de partida.

7.3. Pantalla de unirse a partida



Figura 35: Pantalla de unirse a partida

Esta pantalla mostrará al usuario el texto "introduzca el ID de la partida" con un campo rellenable y un botón para entrar a partida. El usuario deberá rellenar ese campo con el id proveído por otro usuario que esté en la actividad de crear partida y hacer clic en el botón "entrar a partida".

En caso de que esa partida no exista, aparecerá una ventana emergente informando al usuario de que no se ha podido acceder a la partida.

En caso de que la partida exista, se redirigirá al usuario a la pantalla de partida.

7.3. Pantalla de partida



Figura 36: Pantalla de partida

Esta pantalla es la pantalla de partida. Se mostrará a dos usuarios a la vez, desde ahora J1 (jugador 1) y J2 (jugador 2)

Esta pantalla contiene:

- Un texto indicando el turno actual
- Un botón "modificar" que permite a cualquiera de los usuarios modificar cualquiera de los campos de la partida de forma manual
- Un botón "pasar turno" que permite al usuario que está jugando actualmente pasar el turno de su figura. En caso de que sea el turno de su última figura jugable, el turno pasará a la primera figura del otro usuario.
- Un icono de GG Salvaje que está visible si el GG salvaje aún no ha sido asignado y desaparece cuando este es asignado.
- Un icono con un texto mostrando el número de GG del J1, este irá incrementando y decreciendo pertinentemente.
- Un icono con un texto mostrando el número de GG del J2, este irá incrementando y decreciendo pertinentemente.

- Un icono con un texto mostrando el número de Kamas del J1, este irá incrementando y decreciendo pertinentemente.
- Un icono con un texto mostrando el número de Kamas del J2, este irá incrementando y decreciendo pertinentemente.
- Una lista con todas las figuras de J1 a la izquierda y todas las figuras de J2 a la derecha.
- Una lista con las Recompensas demoníacas actualmente disponibles para comprar, distribuidas por precio.

Las dos listas de figuras son independientemente deslizantes y resaltan el turno de la figura actual en rojo.

contiene para cada figura del jugador:

- La imagen de la miniatura
- Un botón con el nombre de la figura que abre la interfaz de la figura para poder ver todos sus detalles
- Un botón que sirve para aplicar los cambios a los campos editables de PM PdV y PA
- Los PM, PdV, PA y Alcance Adicional actuales para esa figura, que pueden ser editados en cualquier momento
- Una lista con cuatro botones de acciones adicionales que solo se mostrará si es el turno de esa figura y el usuario es el jugador a la que esa figura pertenece
 - Botón de dar puñetazo: al hacer clic espera a que el usuario haga click en una figura enemiga para infligir un PdV y reducir en cinco los PA de la figura que lanza el hechizo. Si la figura no tiene suficientes PA para realizar la acción, aparecerá una ventana emergente informando al usuario y no se realizará la acción.
 - Botón de recoger Kama: al hacer clic reduce en uno los PA de la figura actual e incrementa en uno los Kamas del jugador actual. Si la figura no tiene suficientes PA para realizar la acción, aparecerá una ventana emergente informando al usuario y no se realizará la acción.
 - Botón comprar Recompensa Demoníaca: al hacer clic espera a que el usuario haga clic en una recompensa demoníaca para comprarla y reducir en uno los PA de la figura que lanza el hechizo. Si la figura no tiene suficientes PA o Kamas para realizar la acción, aparecerá una ventana emergente informando al usuario y no se realizará la acción

- Botón comprar GG: al hacer clic reduce en diez los Kamas del jugador actual e incrementa en uno los GG del jugador actual. Si la figura no tiene suficientes Kamas para realizar la acción, aparecerá una ventana emergente informando al usuario y no se realizará la acción.
- Una lista con todos los hechizos que puede realizar esa figura. Estos hechizos podrán clicarse en caso de que sea el turno de esa figura y el usuario es el jugador a la que esa figura pertenece.

Al hacer clic en algún hechizo, espera a que el usuario haga click en una figura enemiga para infligir los efectos del hechizo y reducir los PA que cuesta de la figura que lanza el hechizo. Si la figura no tiene suficientes PA para realizar la acción, aparecerá una ventana emergente informando al usuario y no se realizará la acción.

Para cada hechizo se mostrará:

- El tipo de Lanzamiento
- El alcance
- El nombre
- El color
- La descripción
- El coste (tanto de PA, PM o PdV) mostrando sólo los campos que sean necesarios
- El daño o la curación realizada
- El tipo de daño o curación (si es que tiene)
- Una lista con todas las recompensas demoníacas y dados equipados en esa figura

Al llegar los GG de uno de los dos jugadores a 0, una ventana emergente aparecerá para ambos jugadores informando que la partida ha terminado y quien es el ganador. Al hacer click en aceptar, los jugadores serán redirigidos a la pantalla inicial.

7.3. Pantalla de crear carta



Figura 37: Pantalla de crear carta

Esta pantalla solo será accesible para los usuarios administradores. Desde ella se puede entrar una nueva carta en la base de datos rellenando todos los campos necesarios.

Se permite dejar campos vacíos ya que es requerido para algunas cartas.

7.3. Pantalla de crear hechizo



Figura 38: Pantalla de crear hechizo

Esta pantalla solo será accesible para los usuarios administradores. Desde ella se puede entrar un nuevo hechizo en la base de datos rellorando todos los campos necesarios.

Se permite dejar campos vacíos ya que es requerido para algunas cartas.

8. Diseño de base de datos

En este apartado se suelen esperar diagramas conceptuales de las base de datos, así como diagramas del modelo lógico relacional.

En nuestro caso, se ha utilizado Firebase Realtime Database para todos los datos, el cual no está basado en tablas sino en un JSON.

Por otro lado se ha utilizado Firebase storage para almacenar las imágenes.

Para ello, lo que se hará a continuación será mostrar la distribución de los datos con una breve explicación de las reglas que los atañen:

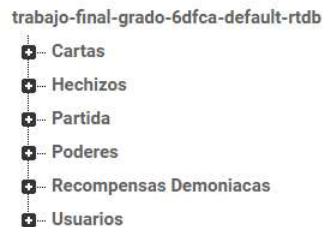


Figura 39: Estructura Firebase padre

Esta es la estructura principal de nuestra base de datos. En ella podemos observar las carpetas:

8.1 Cartas

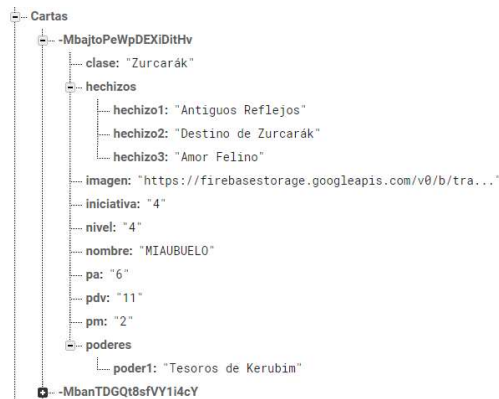


Figura 40: Estructura Firebase Cartas

Cada carta tendrá un identificador único aleatorio y dentro de ese identificador tendremos los campos:

- Clase: la clase de la carta
- Hechizos: dentro de hechizos tendremos una lista con la posición del hechizo como llave y el nombre del hechizo como valor
- Imagen: el valor de imagen será la dirección proveída por Firebase Storage de la imagen de esa carta
- Iniciativa: la iniciativa de la carta
- Nivel: el nivel de la carta
- Nombre: el nombre de la carta
- PA: los Puntos de acción máximos de la carta
- PdV: los Puntos de vida máximos de la carta
- PM: los Puntos de movimiento máximos de la carta
- Poderes: dentro de poderes tendremos una lista con la posición del poder como llave y el nombre del hechizo como valor

Reglas

Lectura: Todos los usuarios registrados

Escritura: Los usuarios administradores

8.2 Hechizos

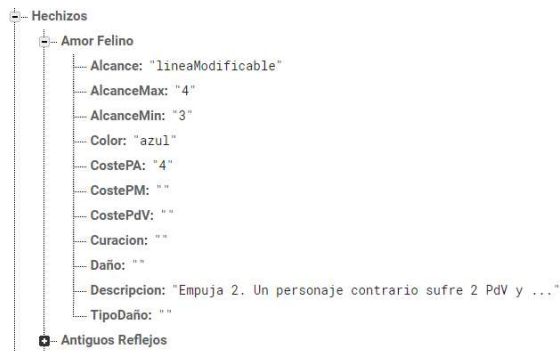


Figura 41: Estructura Firebase Hechizos

Cada hechizo tendrá el nombre como identificador único. Dentro de cada hechizo tendremos:

- Alcance: El tipo de lanzamiento de ese hechizo. Será una cadena de caracteres que posteriormente en la aplicación será transformada a la imagen que podemos ver a la izquierda de cada hechizo
- Alcance máximo: un número que representa el número de casillas máximo donde puede lanzarse el hechizo

- Alcance Mínimo: un número que representa el número de casillas mínimo donde puede lanzarse el hechizo
- Color: el color del hechizo, que representa las restricciones de lanzamiento en turnos y partida del hechizo
- Coste PA: el coste de Puntos de Acción de lanzar el hechizo
- Coste PM: el coste de Puntos de Movimiento de lanzar el hechizo
- Coste PdV: el coste de Puntos de Vida de lanzar el hechizo
- Curación: el número de Puntos de Vida que cura el hechizo
- Daño: el número de Puntos de Vida que inflige el hechizo
- Descripción: una descripción del hechizo que puede incluir efectos adicionales de este
- Tipo daño: el elemento con el que se inflige el daño o se cura (neutral, agua, tierra, fuego, aire, curación)

Reglas

Lectura: Todos los usuarios registrados

Escritura: Los usuarios administradores

8.3 Partida



Figura 42: Estructura Firebase Partida

Cada Partida tendrá un identificador único. Dentro de partida encontraremos:

- GG salvaje: la información del GG salvaje; true si ya se ha gastado y false si no
- Jugador 1 y Jugador2: almacena la información de los jugadores
- Recompensas Demoniacas: almacena una lista de las recompensas demoníacas actualmente disponibles, con la posición como llave y el nombre de la recompensa demoníaca como valor
- Turno: el turno actual de la partida

- Turno figura: el turno de la figura dentro del turno actual de la partida
- Turno Jugador: si el turno actual es del Jugador 1 o del Jugador 2
- Click1: si se ha hecho clic en algún hechizo; true si sí y false si no
- Click1Hechizo: el nombre del hechizo en el que se ha hecho clic

Estos dos últimos parámetros sirven para almacenar la información del click del hechizo para que la partida pueda escuchar el click de a que figura se deberá infligir los efectos de ese hechizo.

- Dentro de Jugador1 y Jugador2 encontraremos:

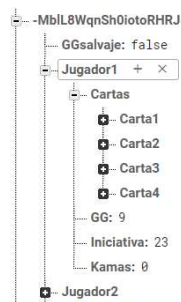


Figura 43: Estructura Firebase Partida – Jugador

- Cartas: una lista con todas las cartas del jugador
- GG: los Galones de Gloria actuales del jugador
- Iniciativa: la iniciativa total de todo el equipo del jugador
- Kamas: el número de Kamas que posee actualmente el jugador
- Dentro de cartas encontraremos las mismas características que en la estructura padre – carta con el añadido de:



Figura 44: Estructura Firebase Partida – Jugador - Carta

- Alcance Temporal: Los puntos de Alcance extra que tenga la carta en ese momento (pueden ser negativos)
- Dado 1: El primer dado que tiene asignado la carta en ese momento
- Dado2: El segundo dado que tiene asignado la carta en ese momento
- PA Temporal: Los puntos de Acción extra que tenga la carta en ese momento (pueden ser negativos)
- PM Temporal: Los puntos de Movimiento extra que tenga la carta en ese momento (pueden ser negativos)
- PdV Temporal: Los puntos de Vida extra que tenga la carta en ese momento (pueden ser negativos)
- Eliminada: Si la carta ha sido eliminada de la partida; true si sí false si no
- Id: El identificador único de la estructura padre – carta

Reglas

Lectura: Los usuarios que participan en la partida, los usuarios administradores

Escritura: Los usuarios que participan en la partida, los usuarios administradores

8.4 Poderes

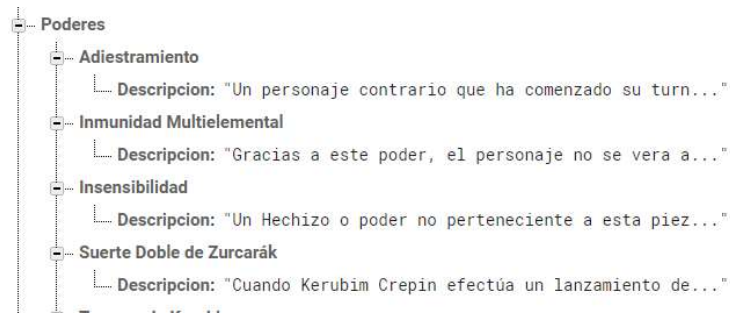


Figura 45: Estructura Firebase Poderes

Dentro de Poderes tendremos una lista de todos los poderes con el nombre como identificador único conteniendo dentro la llave "Descripción" con el valor de la descripción del poder.

Reglas

Lectura: Todos los usuarios registrados

Escritura: Los usuarios administradores

8.5 Recompensas Demoniacas

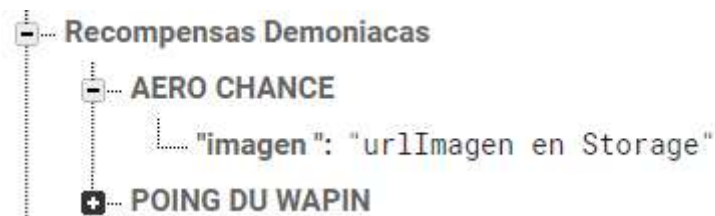


Figura 46: Estructura Firebase Recompensas Demoniacas

Dentro de Recompensas Demoniacas tendremos una lista de todos los poderes con el nombre como identificador único conteniendo dentro la llave "imagen" con el valor de la dirección proveída por Firebase Storage de la imagen de esa carta.

Reglas

Lectura: Todos los usuarios registrados

Escritura: Los usuarios administradores

8.6 Usuarios



Figura 47: Estructura Firebase Usuarios

Dentro de Usuarios tendremos una lista de todos los usuarios con un identificador único aleatorio.

Dentro de cada usuario tendremos una lista de todos los equipos que ha creado con un identificador único aleatorio.

Dentro de cada Equipo tendremos una lista con la posición de la figura como llave con el valor de la dirección proveída por Firebase Storage de la imagen de esa carta.

Reglas

Lectura: Cada usuario puede acceder a su usuario, los usuarios administradores pueden acceder a todo

Escritura: Cada usuario puede acceder a su usuario, los usuarios administradores pueden acceder a todo.

8.7 Firebase Storage

Se Usará Firebase Storage para guardar las imágenes de las Figuras.

Estas estarán almacenadas dentro de la carpeta "Figuras" como se puede ver en la figura siguiente:



Figura 48: Firebase Database

En formato png:



Figura 49: Firebase Database / Figuras

Y el propio Firebase Database nos proveerá de la dirección para acceder a estas imágenes como se puede ver subrayado en amarillo en la figura siguiente:



Figura 50: Firebase Database / Figuras / MIAUBUELO.png

9. Presupuesto

Coste de recursos humanos

Precio/hora que se pagará en función de la tarea. Sueldos medios en España obtenidos de glassdoor.es:

Profesional	€/Año	€/Hora
Gestor	40.000 €	25 €/h
Analista/Diseñador	35.000 €	21,8 €/h
Programador/Testador	28.000 €	17,5 €/h
Técnico de Sistemas	24.000 €	15 €/h

Tabla 4: Precio/Hora en función de la tarea

Tarea	Tiempo	Profesional	Precio Total
Gestión	50 horas (10% de las horas totales)	Gestor	1250€
Contratar plataformas necesarias (Host, BBDD, Servidor HTTP)	5 horas	Técnico de Sistemas	75€
Diseño	100 horas	Diseñador	2180€
Programación	150 horas	Programador + Testador	2625€
Documentación	50 horas	Programador	875€
Testing	150 horas	Testador	2625€
Despliegue	5 horas	Técnico de Sistemas	75€
Análisis de objetivos	40 horas	Analista	872€
Total	550 horas		10.577€

Tabla 5: Precio total de las faenas del proyecto

Coste de recursos materiales

Recurso	Precio	Precio (6 meses)	Detalles
Estación de trabajo	750,89 €	97,47€	750.90 € Totales
Coste de alquiler del local	150 €/mes	900€	Oficina buscada des de la página web idealista.com https://www.idealista.com/inmueble/88798183/
Coste de energía	15 €/mes	90 €	Precio calculado a partir del consumo que tiene un ordenador, 4 fluorescentes LED y un router a la hora multiplicado por cuatro horas al día que es el que se trabajará, multiplicado por 5 días a la semana y multiplicado por 4/25 semanas que tiene un mes.
Coste de conexión a internet	14 €/mes	84€	Tarifa Superior Movistar [11]
Total		1.171,47€	

Tabla 6: Coste de recursos materiales

Detalles del coste de la estación de trabajo:

Objeto	Precio	Tiempo de vida	Precio (6 meses)	Link
Mesa	39€	4 años	4,87€	https://www.ikea.com/es/es/p/linnmon-adils-mesa-blanco-efecto-roble-tinte-blanco-blanco-s89215672/
Silla	59€	4 años	7,37€	https://www.ikea.com/es/es/p/flintan-silla-trabajo-vissle-negro-20336841/
Portátil	647,90€	4 años	80,98€	https://www.pccomponentes.com/hp-250-g7-intel-core-i5-1035g1-8gb-256gb-ssd-156

Ratón	4,99€	2 años	1,25€	https://www.pccomponentes.com/logitech-b100-rat-n-negro
-------	-------	--------	-------	---

Tabla 7: Detalles de coste de la estación de trabajo

Coste de licencias, suministros y servicios

Recurso	Precio	Precio (6 meses)	Link
Joc de taula Krosmaster: Blast	44,90€	44,90€	https://www.ankama-shop.com/en/krosmaster/2519-krosmaster-blast.html
Publicación en Google Play	25 €	25€	Contratación de un servidor dedicado a ionos.es
Licencia Windows 10 Pro	260€/año	130 €	https://www.microsoft.com/es-es/store/b/windows
IntelliJ IDEA Ultimate	500€/año	250 €	https://www.jetbrains.com/es-es/store/#commercial?billing=yearly
Total		449,90 €	

Tabla 8: Coste de licencias, suministros y servicios

Coste Total

Coste de Recursos humanos: 10.577€

Coste de recursos materiales: 1.171,47€

Coste de licencias, suministros y servicios: 449,90 €

Total: 12.198,37€

10. Conclusiones

Una vez finalizado el proyecto, se debe analizar si se han cumplido los diferentes objetivos que se tenía en mente:

El primer objetivo era crear una herramienta de seguimiento de la partida para evitar el error humano. Después de las pruebas realizadas con jugadores novicios jugando con y sin la aplicación podemos concluir que este objetivo ha sido alcanzado.

El segundo objetivo era crear una herramienta que disminuya la cantidad de elementos que es necesario llevar para la realización de una partida. Después de las pruebas realizadas realizando partidas con y sin la aplicación podemos concluir que este objetivo ha sido alcanzado.

El tercer objetivo era crear un catálogo donde ver todas las miniaturas coleccionables. Este objetivo es alcanzable, ya que parte de las miniaturas coleccionables ya están introducidas en el catálogo, sin embargo, en la previsión de el proyecto no se contempló el numero de horas necesarias introduciendo todos los datos y con la planificación actual, ese numero de horas no ha sido asumible.

La aplicación ha cumplido con todos los requisitos funcionales y tecnológicos que se habían propuesto. A pesar de eso, la profundidad con la que se han tratado estos, no ha sido la máxima posible, ya que sería necesario tratar cada hechizo, cada poder y cada recompensa demoníaca por separado y ese trabajo no ha sido asumible debido al tiempo. La forma de solucionar esta falta de profundidad en la automatización ha sido dándole a los usuarios herramientas para modificar los atributos personalmente, pudiendo así realizar ellos esa parte del trabajo personalmente.

A nivel personal, la realización de este proyecto ha sido un reto tanto para la gestión del tiempo como por el propio reto de desarrollar una aplicación de este nivel de complejidad desde cero. Algunas cosas se podrían haber hecho mejor y otras se han hecho lo mejor que se ha podido, pero en general el trabajo realizado ha producido un resultado satisfactorio.

Aprender a trabajar con firebase ha sido algo muy estimulante. Los servicios están muy bien desarrollados y documentados. Por otro lado, trabajar con Android Studio no ha sido tan satisfactorio a causa de las RecyclerViews. La forma de tratar con listas de datos variables en Android se tiene que hacer a traves de estas y añaden una complejidad al proyecto que,

en caso de tener que usar Nested RecyclerViews (RecyclerViews dentro de RecyclerViews) crece exponencialmente y dificultan todo el proceso de programar.

11. Referencias

- [1]. lastlevel.es (2015, mayo), Krosmaster Arena: Reglamento (versión2.1). [Online]. Disponible en: https://www.lastlevel.es/distribucion/krosmaster/Reglas_Krosmaster_v2.1.pdf
- [2] Miller, M. (2013, diciembre 31). Top Ten Tabletop Games Of 2013. [Online]. Disponible en: <https://www.gameinformer.com/b/features/archive/2013/12/31/top-ten-tabletop-games-of-2013.aspx?PostPageIndex=2>
- [3] Martínez, Jesús (Consultado 2021, enero 25). Krosarchive.es. [Online]. Disponible en: <https://krosarchive.es/ES/seasons>
- [4] Federico, Josh (Consultado 2021, enero 25) Krosfinder.com. [Online]. Disponible en: <https://web.archive.org/web/20200117002959/http://krosfinder.com/>
- [5] Jorge Ruiz-Vanoye, Alejandro Fuentes-Penna, Ocotlán Díaz-Parra i cuatro autores más “*Metodologías de Desarrollo de Software*”. Edición 1. Editorial Académica Dragón Azteca, 2017
- [6] Popal, Nabila y Reith, Ryan (2021, Abril, 28) Smartphone Market Share. [Online] Disponible en: <https://www.idc.com/promo/smartphone-market-share/os>
- [7] Múltiples autores (2021, Mayo 21) Android. [Online] Disponible en: <https://es.wikipedia.org/wiki/Android>
- [8] Developer.android.com (2021, Abril) Notas de la versión de Android Studio. [Online]Disponible en: <https://developer.android.com/studio/releases>
- [9] Developer.android.com (2021, Abril) Todo lo que necesitas para realizar compilaciones en Android. [Online] Disponible en: <https://developer.android.com/studio/features>
- [10] firebase.google.com (2020, diciembre 18) Firebase Realtime Database. [Online] Disponible en: <https://firebase.google.com/docs/database>
- [12] Movistar (2021, febrero 1) Tarifas móviles Empresas - Telefónica. [Online] Disponible en: https://empresas.telefonica.es/landings?custom_emp=10097392935_kwd-305034232452&sem=900107056&tc_alt=4863&n_okw=tarifas%20movil%20movistar%2

[Oempresas_b_c_100532189519_10097392935_kwd-305034232452_&&keyword=tarifas%20movil%20movistar%20empresas&gclid=CjwKC_AiA65iBBhB-EiwAW253W_CdtSlyJuIFmhTJTFeyzFo69MU_E9wy6DDc7_7AIl7Q2VhkvovqZhoCxK_YQAvD_BwE&gclsrc=aw.ds](https://www.emprendedores.es/gestion/requisitos-legales-app-lanzar-aplicacion/)

[13] Emprendedores, Redacción. (2015, noviembre 23). Requisitos legales que debe cumplir una app. [online] Disponible en: <https://www.emprendedores.es/gestion/requisitos-legales-app-lanzar-aplicacion/>

[14] Khawas, Chunnu & Shah, Pritam. "Application of Firebase in Android App Development-A Study". *International Journal of Computer Applications (0975 – 8887)*, Volumen 179, No.46, pp. 1-5, junio 2018

[15] Navdeep Singh, "Study Of Google Firebase API For Android", *IJIRCCE (An ISO 3297: 2007 Certified Organization)* Vol. 4, Issue 9, September 2016

[16] Fernández, Yenisleidy y Diaz, Yanette. " Patrón Modelo-Vista-Controlador." *Revista Telem@tica*. Vol. 11. No. 1, pp. 47-57, enero-abril, 2012

[17] giovanni, jonathan (2018, noviembre 29). MVP_Firebase_Client. [Online] Disponible en: https://github.com/jonathan-giovanni/MVP_Firebase_Client

[18] Hohensee, Barbara. *Introducción A Android Studio. Incluye Proyectos Reales Y El Código Fuente*. Edición 1. Babelcube Inc, 2014.

[19] Tamplin, James(2016, mayo 18) Firebase se expande para convertirse en una plataforma de aplicaciones unificada. [Online] Disponible en: <https://firebase.googleblog.com/2016/05/firebase-expands-to-become-unified-app-platform.html>

[20] firebase.google.com (2020, diciembre 18) Firebase Authentication. [Online] Disponible en: <https://firebase.google.com/docs/auth>

[21] firebase.google.com (2021, marzo 04) Cloud Storage para Firebase. [Online] Disponible en: <https://firebase.google.com/docs/storage>

