

Board Game Manager



Curso: 2ºDAM O

Alumno: Miguel Pastor Pérez

Tutor: Jose Ramón García Sevilla

INDICE

1.	Introducción.....	4
	Breve descripción del proyecto.....	4
	Breve descripción del contenido de la memoria	4
2.	Estudio de mercado	5
	Contexto del sector productivo	5
	Análisis de competencia y DAFO	6
	Segmentación del mercado	7
	Modelo de negocio	7
	Ubicación	7
3.	Marketing	8
	Producto o servicio.....	8
	Precio.....	8
	Promoción.....	8
	Distribución	9
4.	Forma jurídica	10
5.	Recursos humanos	11
6.	Análisis de costes	12
	Tabla de costes	12
	Cálculo del umbral de rentabilidad	12
7.	Inversión inicial.....	13
8.	Fuentes de financiación	14
9.	Viabilidad económica	15
	Plan de tesorería.....	15
	Cuenta de resultados	16

INDICE

10.	Análisis de requisitos.....	17
	Requisitos funcionales	17
	Requisitos no funcionales.....	18
	Metodología de desarrollo. Fases del proyecto. Tareas y plazos de ejecución.....	19
11.	Diseño.....	21
	Diseño de datos	21
	Entidad – Relación.....	21
	Diseño lógico – relacional / Estructura de la base de datos	22
	Diseño funcional.....	23
	Diagrama de clases	23
	Diagrama de casos de usos	24
	Diseño de interfaces.....	25
12.	Implementación.....	26
	Tecnologías a emplear.....	26
	Requerimientos, comparativa y justificación de las tecnologías elegidas.....	26
	Descripción de las tecnologías utilizadas en el prototipo	28
	Diario de desarrollo	30
13.	Pruebas.....	38
	Procedimiento de evaluación, seguimiento y control del proyecto	38
	Procedimientos para la participación de los usuarios en la evaluación del proyecto	39
14.	Conclusiones.....	40
15.	Bibliografía	41

Introducción

Breve descripción del proyecto

Buscamos una solución para mejorar los juegos de mesa tecnológicamente. La principal idea en que se basa nuestro proyecto es ofrecer un seguimiento de partidas y estadísticas. A continuación, hablaremos de nuestra empresa, aplicación y la idea de negocio más detalladamente:

- La empresa **Board Score Technologies** es la empresa que ha decidido desarrollar una aplicación cuyo objetivo prioritario es darse a conocer y crecer en el mundo de los juegos de mesa, sin restarle importancia a los beneficios de la aplicación, que también son importantes para el mantenimiento de la empresa.
- Nuestra aplicación es **Board Game Manager**, esta se centra en el control y almacenamiento de partidas de juegos de mesa, basándose en la idea de que en estos juegos no hay una buena forma de tener un historial de las partidas y estadísticas, así se podría tener un seguimiento para las personas que quieran avanzar al siguiente nivel.
- Como idea de negocio nuestra aplicación estará diseñada para juegos comunes y su forma de contar puntos también será general, una vez esta aplicación se dé a conocer se espera realizar pactos con empresas de juegos de mesa añadiendo una interfaz y un apartado específico para su juego a cambio de que la empresa anuncie en esta nuestra aplicación.

Breve descripción del contenido de la memoria

En esta memoria se analizará el proyecto tanto en el apartado de empresa como en el técnico.

En el apartado de empresa veremos sus competencias, buscaremos la forma correcta de venderlo, las posibilidades en el mercado y una estimación de sus posibles beneficios y gastos.

Y en el apartado técnico estableceremos los requisitos específicos de nuestro proyecto, la metodología óptima para este, su diseño de datos, funcional y de interfaces, las tecnologías que usaremos y sus pruebas.

Una vez analizado los dos apartados podremos llegar a una conclusión sobre nuestro proyecto y su futuro.

Estudio de mercado

Contexto del sector productivo

El sector productivo al que va dirigido nuestro producto sería el sector servicios o sector terciario, en el subsector ocio más concretamente dirigido al de juegos de mesa.

Este sector lleva creciendo año tras año gracias al impulso que tuvo en la pandemia. Según los datos de **NPD Group** la empresa de investigación del mercado global que se especializa en el seguimiento y análisis de diversas industrias, este sector ha incrementado sus ingresos en un 7%, además podemos observar los resultados de **Asmodee** una de las mayores distribuidoras de juegos de mesa del mundo, que ha logrado facturar más de 20 millones de euros en 2023 en España.

(ejecutivos, 2024).

La gráfica más actual que he encontrado es la siguiente, que muestra las ventas de **Devir**, la empresa con grandes juegos reconocidos mundialmente como *Catan* y *Carcassonne*.

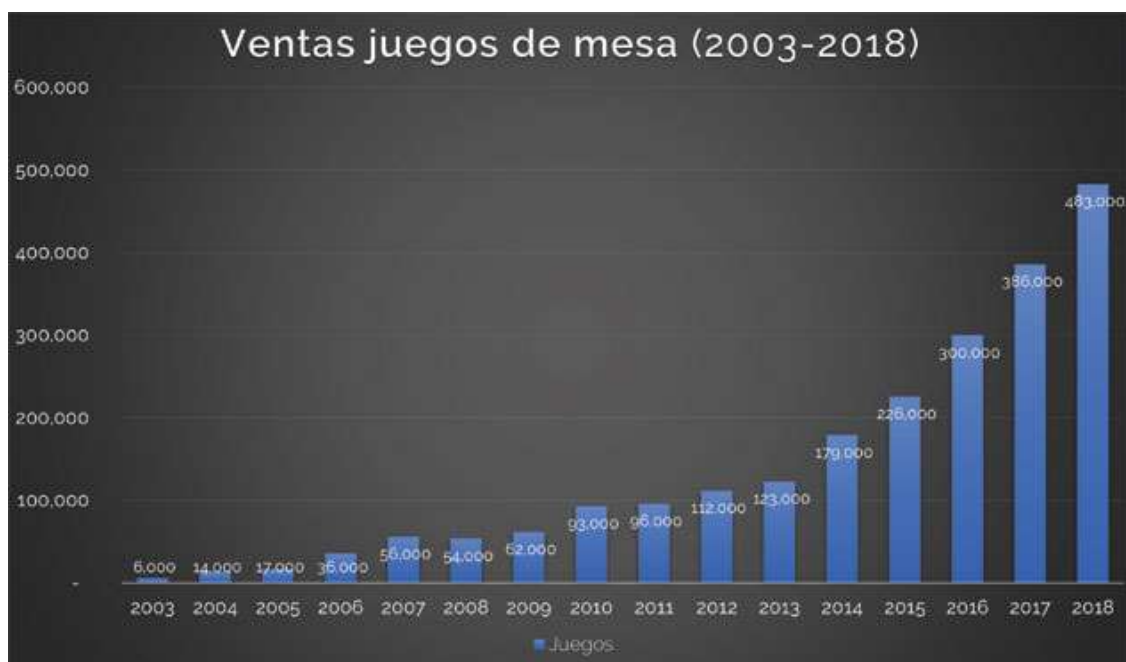


Figura 1: Grafica ventas de juegos de mesa de **Devir**, Fuente: (Juguetes b2b, 2019)

Análisis de competencia y DAFO

Hemos realizado un estudio general de la competencia y hemos encontrado dos tipos de competidores:

1. **Board Game Stats**, son los mayores competidores, tienen una aplicación muy completa que lleva bastantes años que además cuenta con una alta valoración según sus usuarios. Pero su mayor debilidad es su alto coste, que es superior a los cinco euros.
2. El resto de los competidores encontrados son aplicaciones sencillas con la fortaleza de que son gratuitas a consta de sus debilidades como diseños simples y menos funcionalidades, la mayoría solo cuentan con solo el propio conteo de la partida.



Figura 2: Análisis DAFO, Fuente: Propia

Segmentación del mercado

Para nuestro proyecto hemos establecido los siguientes criterios de segmentación del mercado.

Demográficamente no tenemos grandes restricciones, en este factor el rango de edad sería entre los 12 años que es una edad en la que se empieza a tener acceso a las tecnologías hasta los 50 puesto que hay una gran parte de la gente más mayor que no está habituada al uso de estas.

En el criterio psicográfico, el proyecto se centrará en las personas sociables y competitivas, ya que gracias a nuestra aplicación tendrán una motivación adicional para seguir reuniéndose y evitar jugar muy esporádicamente, ayudando a crear comunidades de juegos de mesa que utilicen complementariamente nuestra aplicación fomentando la competición.

Modelo de negocio

Nuestro modelo de negocio se basa en la venta online, en el que el cliente compra nuestro producto, además para darlo a conocer ofreceremos una versión limitada, para que el usuario objetivo conozca las comodidades y beneficios que le otorga nuestra aplicación y crear la necesidad de comprar el producto.

Ubicación

Hemos decidido trabajar desde un *coworking* que nos brinde las comodidades necesarias, después de realizar una busca exhausta en la zona hemos encontrado el coworking Coalte situado en la calle del Marqués de Molins en Alicante, que nos ofrece comodidades tales como wifi, cableado RJ-45, aire acondicionado, impresora, bebidas frías, entre otros. Solo por un precio de 90€ al mes.

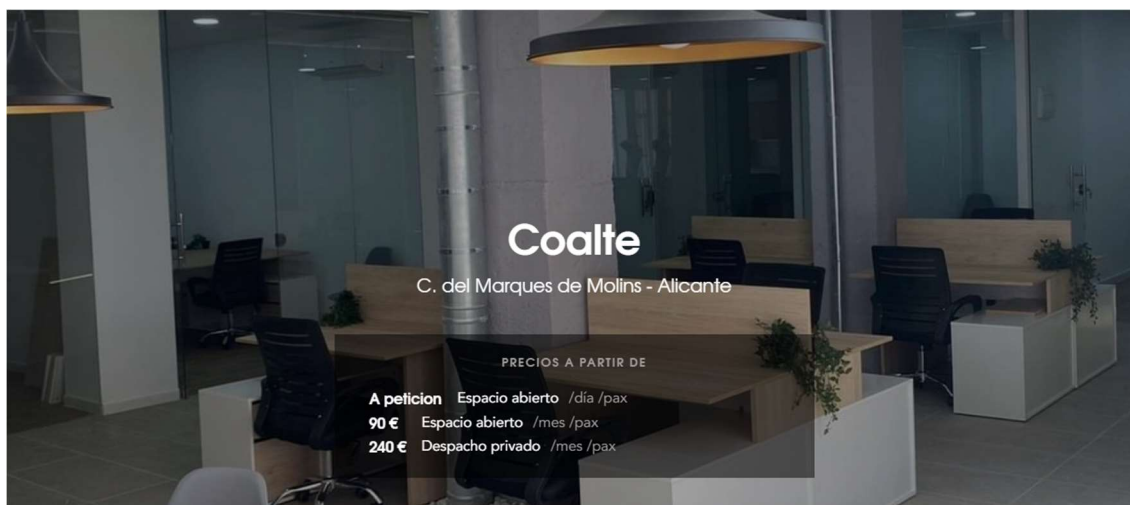


Figura 3: Espacio de trabajo, Fuente: (Workin space, 2024)

Marketing

Producto o servicio

Nuestro producto es una aplicación para controlar partidas de juegos de mesa. Una vez finalizada la partida, se almacenan y se actualizan las estadísticas de los jugadores.

La aplicación está diseñada para facilitar el seguimiento de partidas de diferentes juegos de mesa, sus principales funciones son gestionar partidas, ver el historial de partidas y ver las estadísticas de los jugadores.

Precio

Nuestra aplicación ofrece dos opciones:

- La opción gratuita con funcionalidades limitadas, como la cantidad de partidas que se pueden tener almacenadas y los jugadores, que en ambos casos serán de un máximo de cuatro. De esta forma, el cliente puede probar el producto, pero sin lograr su principal.
- La otra opción es la versión completa sin limitaciones con un costo de 2,97 euros que ofrece una experiencia completa y sin restricciones.

Promoción

La principal promoción será un intercambio de beneficios entre los juegos de mesa y nuestra aplicación dando a conocer el juego de mesa en nuestra aplicación con una interfaz personalizada y funciones específicas para gestionar sus partidas, a cambio de que ellos promocionen nuestra aplicación en su juego.

Pero como esto será en un futuro cuando la aplicación ya sea un poco reconocida al inicio se hará promoción en redes sociales personales de la empresa, los momentos importantes para hacer promoción son, antes del lanzamiento, el propio lanzamiento de la aplicación y antes de lanzar una gran actualización.

Para aumentar la visibilidad de nuestro proyecto irá destinado a promocionar nuestras redes sociales un total de 3000 euros anuales.

Distribución

Nuestro producto se distribuirá a través de la **Google Play Store** primero y una vez avance y se adaptará la aplicación a iOS también se distribuirá en la **App Store**.

Para distribuirlo en la **Google Play Store** se necesita una cuenta de desarrollador que se consigue con un único pago de 23,42€ y en la **App Store** igual pero la cuenta de desarrollador se obtiene con un pago anual de 92,72€.

Ambas se llevan un porcentaje de cada venta, la **Google Play Store** se lleva un 30% pero por estadísticas suele tener más ventas, mientras que la **App Store** solo se lleva un 15% dado a que esta distribuidora no genera tantas ventas.

Forma jurídica

Dado que constituir una sociedad puede ser más costoso y complejo en cuanto a los trámites administrativos he optado por darme de alta como empresario individual puesto que para formar una sociedad hará falta un capital inicial más elevado que podemos ahorrarnos. Analizaremos las ventajas y desventajas:

Ventajas:

- **Es ideal para empresas de pequeña escala**, ofreciendo una estructura ágil y adaptable.
- **Menos tramites y gestiones**, simplificando la gestión y reduciendo la carga administrativa.
- **Costos más bajos**, además de lo antes mencionado se ahorrarán todos los gastos relacionados con la creación de una entidad legal separada del empresario.

Desventajas:

- **Limitaciones de expansión del negocio** por las limitaciones de tiempo, recursos y habilidades individuales.
- **Falta de separación patrimonial**, no hay una separación entre los activos personales y los de tu negocio y esto es un riesgo dado a que los problemas financieros de tu negocio podrían poner tus bienes personales en riesgo.

(Gencat, 2022).

Recursos humanos

En el apartado de recursos humanos es fundamental para definir como se gestionarán los aspectos relacionados con el personal dentro de la empresa. Dado que en la empresa se centra en un trabajador autónomo y por el momento no dispone de más trabajadores en este apartado no se podrá tratar sobre temas relacionados con estos y tampoco realizar el organigrama de la empresa, puesto que no tendría sentido.

En cuanto a los impuestos como autónomo gracias a empezar por primera vez podremos elegir la opción de la tarifa plana de 80€ durante los primeros 12 meses, los siguientes meses según los beneficios variará, pero haciendo los cálculos con los posibles beneficios y gastos será alrededor de 300€.

(Dobano, 2023).

Análisis de costes

Tabla de costes

Costes fijos	Euros (mensual)	Euros (anual)
Remuneración mensual	1.400 €	16.800 €
Gastos de marketing digital	-	3000 €
Coworking	90 €	1080 €
Distribución	-	116,14 €
Cuota Autónomo	80 €	960 €
Suma total anual:		21.956,14 €

Para aclarar el análisis de costes hay que tener en cuenta que el impuesto del primer año es de 80€ mientras que los siguientes ascenderá a 300€.

Cálculo del umbral de rentabilidad.

El cálculo del umbral de rentabilidad se realiza dividiendo los costes por el beneficio al vender un producto, para determinar cuánto se debe vender para cubrir los gastos y empezar a generar beneficios.

Nuestros costes son de 21.956,14 euros y nuestro ingreso por producto es de 2.97 euros, por lo tanto, necesitaremos vender **7393** (7392.63) unidades para empezar a generar beneficios, hay que tener en cuenta que cada venta tiene un coste de distribución que se tendrá en cuenta en el plan de tesorería.

Inversión inicial

Dado a que hemos buscado un coworking que nos aporta todo lo que necesitamos para trabajar en nuestro entorno lo único en lo que tendremos que invertir el dinero al empezar es en un equipo en el que podamos trabajar.

Concepto	Vida útil	Coste total	Amortización anual
Portátil	3 años	600 €	200 €
Total		600 €	200 €

Fuentes de financiación

Dado que nuestro proyecto se estima que empezará a ser autosuficiente tras varios meses, necesitaremos tener una inversión que pueda cubrir estos meses y tener un margen suficiente, después de hacer unos cálculos se estima que con aproximadamente 15.300€ el proyecto podría estar totalmente costado durante los primeros 6 meses.

Concepto	Euros (anual)
Ahorros	3.000 €
Crowdfunding	12.500 €
Total	15.500 €

Para financiar nuestro proyecto haremos un crowdfunding gracias a *Kickstarter* con una meta de 12.500 euros, dado a que esta plataforma es de las más populares, además esta plataforma no te ayuda solo a recaudar fondos, sino que también dispone de marketing y promoción integrados que pueden dar una amplia visibilidad y obtener la validación del mercado.

Viabilidad económica

Plan de tesorería

Entradas	Enero	Febrero	Marzo	Abril	Mayo	Junio	Julio	Agosto	Septiembre	Octubre	Noviembre	Diciembre	Total
Ahorros empresario	3.000,00 €	- €	- €	- €	- €	- €	- €	- €	- €	- €	- €	- €	3.000,00 €
Crowdfunding	12.500,00 €	- €	- €	- €	- €	- €	- €	- €	- €	- €	- €	- €	12.500,00 €
Ventas (Play Store)	- €	- €	3.671,36 €	1.880,38 €	1.311,20 €	896,98 €	1.227,76 €	4.440,20 €	2.372,08 €	1.779,06 €	980,42 €	1.644,96 €	20.204,40 €
Ventas (App Store)	- €	- €			2.753,52 €	1.436,36 €	935,72 €	1.972,76 €	3.131,98 €	1.644,96 €	1.194,98 €	1.469,14 €	14.539,42 €
Total entradas	15.500,00 €	- €	3.671,36 €	1.880,38 €	4.064,72 €	2.333,34 €	2.163,48 €	6.412,96 €	5.504,06 €	3.424,02 €	2.175,40 €	3.114,10 €	50.243,82 €
Salidas	Enero	Febrero	Marzo	Abril	Mayo	Junio	Julio	Agosto	Septiembre	Octubre	Noviembre	Diciembre	Total
Comisión (Play Store)	- €	- €	1.101,41 €	564,11 €	393,36 €	269,09 €	368,33 €	1.332,06 €	711,62 €	533,72 €	294,13 €	493,49 €	6.061,32 €
Comisión (App Store)	- €	- €	- €	- €	413,03 €	215,45 €	140,36 €	295,91 €	469,80 €	246,74 €	179,25 €	220,37 €	2.180,91 €
Sueldo	1.400,00 €	1.400,00 €	1.400,00 €	1.400,00 €	1.400,00 €	1.400,00 €	1.400,00 €	1.400,00 €	1.400,00 €	1.400,00 €	1.400,00 €	1.400,00 €	16.800,00 €
Gastos de marketing digital	3.000,00 €	- €	- €	- €	- €	- €	- €	- €	- €	- €	- €	- €	3.000,00 €
Coworking	90,00 €	90,00 €	90,00 €	90,00 €	90,00 €	90,00 €	90,00 €	90,00 €	90,00 €	90,00 €	90,00 €	90,00 €	1.080,00 €
Equipo informático	600,00 €	- €	- €	- €	- €	- €	- €	- €	- €	- €	- €	- €	600,00 €
Distribución	116,14 €	- €	- €	- €	- €	- €	- €	- €	- €	- €	- €	- €	116,14 €
Impuesto Autónomo	80,00 €	80,00 €	80,00 €	80,00 €	80,00 €	80,00 €	80,00 €	80,00 €	80,00 €	80,00 €	80,00 €	80,00 €	960,00 €
Total salidas	5.286,14 €	1.570,00 €	2.671,41 €	2.134,11 €	2.376,39 €	2.054,55 €	2.078,69 €	3.197,97 €	2.751,42 €	2.350,46 €	2.043,37 €	2.283,86 €	30.798,37 €
Entradas menos salidas	10.213,86 €	- 1.570,00 €	999,95 €	- 253,73 €	1.688,33 €	278,79 €	84,79 €	3.214,99 €	2.752,64 €	1.073,56 €	132,03 €	830,24 €	19.445,45 €
Saldo en el banco c.c	10.213,86 €	8.643,86 €	9.643,81 €	9.390,08 €	11.078,41 €	11.357,20 €	11.442,00 €	14.656,98 €	17.409,62 €	18.483,18 €	18.615,21 €	19.445,45 €	19.445,45 €

Figura 4: Plan de tesorería, Fuente: Propia

Cuenta de resultados

Cuenta resultados	Año 1	Cuenta resultados	Año 1
Ingresos de explotación		Gastos de explotación	
Ventas Ingresos	34.743,82 €	Remuneración mensual	16.800,00 €
-	- €	Gastos de marketing digital	3.000,00 €
-	- €	Coworking	1.080,00 €
-	- €	Amortización Equipo Informático	200,00 €
-	- €	Distribución	116,14 €
-	- €	Impuesto Autónomo	960,00 €
-	- €	Comisiones	8.242,23 €
Total de ingresos de explotación	34.743,82 €	Total de gastos de explotación	30.398,37 €
Ingresos financieros		Gastos financieros	
-	- €	-	- €
Total de ingresos de financieros	- €	Total de gastos de financieros	- €
Resultados de explotación	4.345,45 €		
Resultados financieros	- €		
Resultados antes de impuestos	4.345,45 €		

Análisis de requisitos

Requisitos funcionales

La aplicación será un gestor de partidas de juegos de mesa. Para ello, necesitaremos tener control sobre el juego que se está jugando y los jugadores de la partida. Con esto, podremos gestionar una partida, añadiendo opciones que pueden ser necesarias a lo largo de ella, como decidir algo al azar, llevar el control de a qué jugador le toca el turno y realizar algún cálculo básico. Además, cuando la partida finalice, se tendrá que poder determinar el ganador. Más adelante, se especificarán más concretamente las funcionalidades.

Sobre el juego, cabe destacar que en la primera versión nuestro proyecto se centra en el control de partidas de juegos de mesa más estándares como pueden ser la mayoría de los juegos de cartas, dados, etc. Aun así, se podrá usar para cualquier juego que el resultado dependa de una puntuación.

Para otros juegos que tienen muchas variables se harán versiones específicas de la aplicación para abarcar las diferentes variables que sean necesarias para cada uno, pero en estos casos de juegos más complejos, la primera versión de la aplicación podría aportar todas sus funcionalidades y podría ser utilizada como apoyo llevando la puntuación simple en vez de todas las variables del juego.

Además, la primera versión de nuestra aplicación será para Android, pero una vez lanzada gracias a nuestras tecnologías se hará una adaptación de esta para iOS. Para nuestra aplicación necesitaremos las siguientes funcionalidades:

- Cargar partida guardada.
- Se podrá elegir un juego predefinido.
- Se podrá elegir un juego del usuario permitiendo hacer un CRUD de este.
- Gestión de los usuarios permitiendo realizar un CRUD de estos.
- Generar orden de juego aleatorio o no hacerlo.
- Control de turnos por jugador o por ronda.
 - Modificación de puntos.
 - Tirar dados.
 - Cambio de turno de jugador.
 - Cambio de ronda.
 - Invertir el orden.
 - Guardar partida.
 - Finalizar la partida.
 - Anular la partida.
- Resultados de la partida.

- Historial de partidas con búsqueda
 - Por fecha.
 - Por jugadores.
- Lista de jugadores búsqueda ordenada.
 - Por victorias generales.
 - Por victorias de un juego específico.

Requisitos no funcionales

- Rendimiento.
 - Tiempo de carga bajo.
 - Tiempo de respuesta al realizar acciones bajo.
- Usabilidad.
 - Interfaz intuitiva y fácil de usar.
 - Uniformidad en el diseño y navegación.
- Escalabilidad.
 - Flexibilidad para agregar nuevos juegos.
 - Posibilidad de ampliar el número de plataformas disponibles.
- Compatibilidad.
 - Soporte en la gran mayoría de dispositivos Android.

Metodología de desarrollo. Fases del proyecto.

Tareas y plazos de ejecución.

Hemos elegido la metodología incremental, puesto que el avance progresivo de esta nos facilitará el desarrollo para poder ofrecer distintas versiones de nuestro producto e ir recibiendo retroalimentación de estos, además de ir captando los primeros compradores.

Gracias al avance progresivo antes mencionado esta es metodología nos otorgará los siguientes beneficios:

- **Reducción del riesgo**, al dividir el proyecto los errores serán más fáciles de detectar y así evitar errores graves.
- **Mejora continua temprana**, permitiendo mejorar desde el principio el producto constantemente gracias al aprendizaje sobre la marcha y la retroalimentación.
- **Flexibilidad ante cambios**, dando la capacidad de adaptarse fácilmente a modificaciones en condiciones o requisitos sobre la marcha.

En este modelo incremental esta segmentado por incrementos, los primeros incrementos definidos son:

- **Desarrollo inicial**: Se centrará en sacar una aplicación que pueda gestionar una partida de un juego de mesa casual como una partida de cartas, dados o cualquier juego que solo necesite el control de turnos o rondas y puntos para **Android**.
- **Primer incremento**: Este incremento adaptará el primer desarrollo en **Android** para **iOS**.
- **Segundo incremento**: Se añadirán formas específicas de administrar partidas de juegos más elaborados y complejos que no solo haga falta la puntuación, gestionando así muchas más variables para **Android**.
- **Tercer incremento**: En este incremento se adaptará el segundo incremento para **iOS**.
- **Cuarto incremento**: Los tres juegos más jugados recibirán una interfaz personalizada tanto para **Android** como **iOS**.

Las fases definidas dentro de un incremento son las siguientes:

1. **Análisis de los requerimientos.** En esta primera etapa se realizará una revisión detallada de los requisitos establecidos y se determinará con los objetivos del incremento y sus características claves.
2. **Diseño del sistema y programa.** Una vez definidos los requisitos de la fase incremental se realizará el diseño, desde la arquitectura del sistema hasta la interfaz de usuario, de esta forma habrá una sólida estructura para empezar con la implementación.
3. **Codificación.** En esta fase se llevará a cabo la implementación de las funcionalidades anteriormente definidas siguiendo el diseño anteriormente definido, enfocándose en hacer código sostenible con el tiempo.
4. **Pruebas y validación.** Después de realizar la codificación en la etapa se realizarán pruebas para detectar y corregir posibles errores o fallos en el sistema, garantizando así el correcto funcionamiento del proyecto.
5. **Despliegue.** Para terminar en la última etapa se llevará a cabo el despliegue del incremento en producción, asegurándonos de realizar una transición suave para los usuarios de nuestra aplicación.

El desarrollo inicial, es decir la primera versión de nuestro proyecto tendrá los siguientes plazos de ejecución.

Fases	Dificultad	Duración
Análisis de los requerimientos	Alta	7 días
Diseño del sistema y programa	Alta	10 días
Codificación	Media-Baja	30 días
Pruebas y validación	Media-Alta	14 días
Despliegue	Baja	3 días
Tiempo requerido total:		64 días

(Informatica metodologias, 2017)

Diseño

Diseño de datos

Entidad – Relación

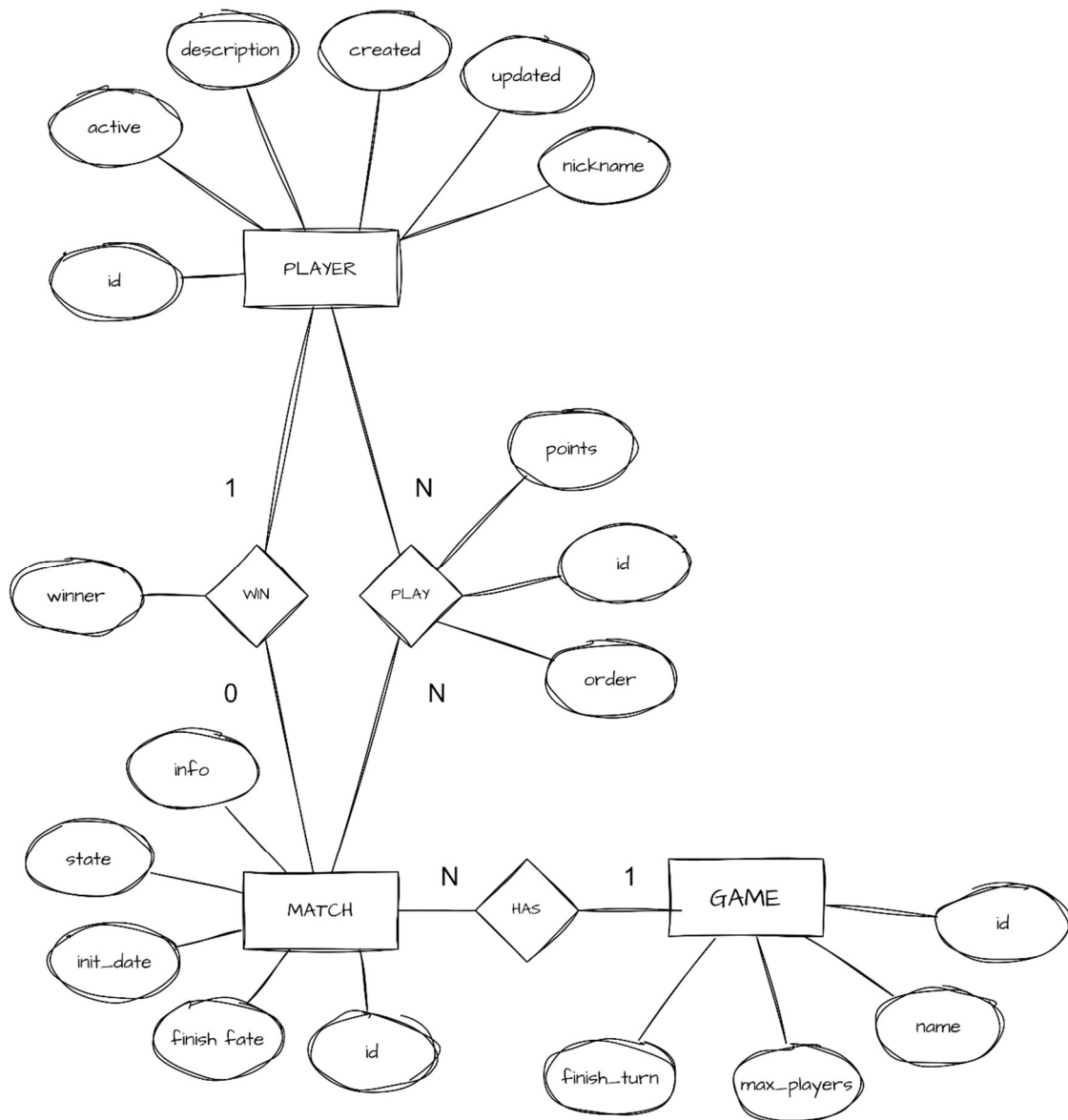


Figura 5: Diseño entidad relación, Fuente: Propia

Diseño lógico – relacional / Estructura de la base de datos

- player (id, nickname, created, updated, description, active):
 - PK: id
- score (id, player_id, match_id, points, order):
 - PK: id
 - FK: player_id → player
 - FK: match_id --> match
- match (id, game_id, state, winner, info, init_date, finish_date):
 - PK: id
 - FK: winner → player
 - FK: game_id → game
- game (id, name, max_players, finish_turn)
 - PK: id

Diseño funcional

Diagrama de clases

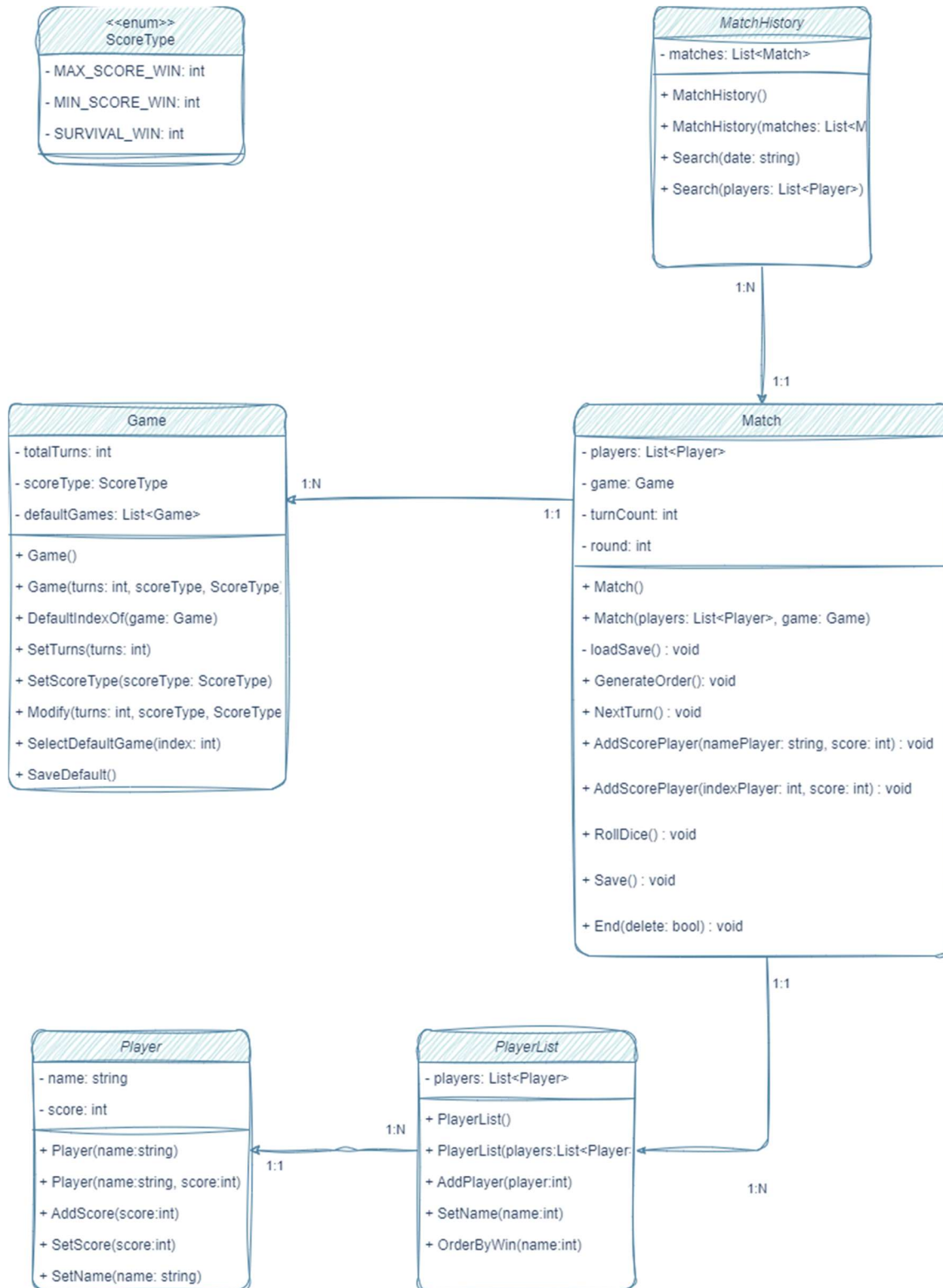


Figura 6: Diagrama de clases, Fuente: Propia

Diagrama de casos de usos

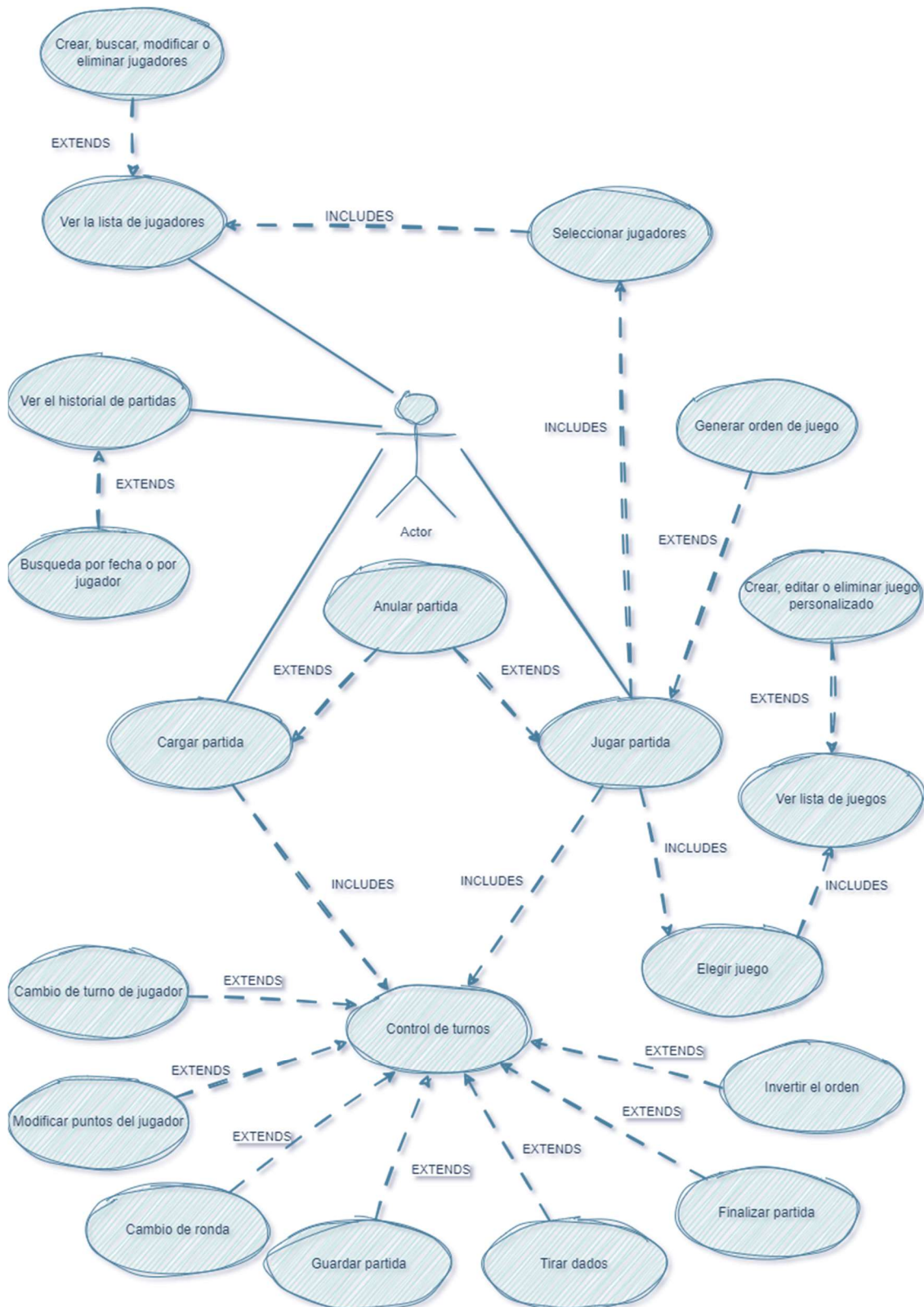


Figura 7: Diagrama de casos de uso, Fuente: Propia

Diseño de interfaces

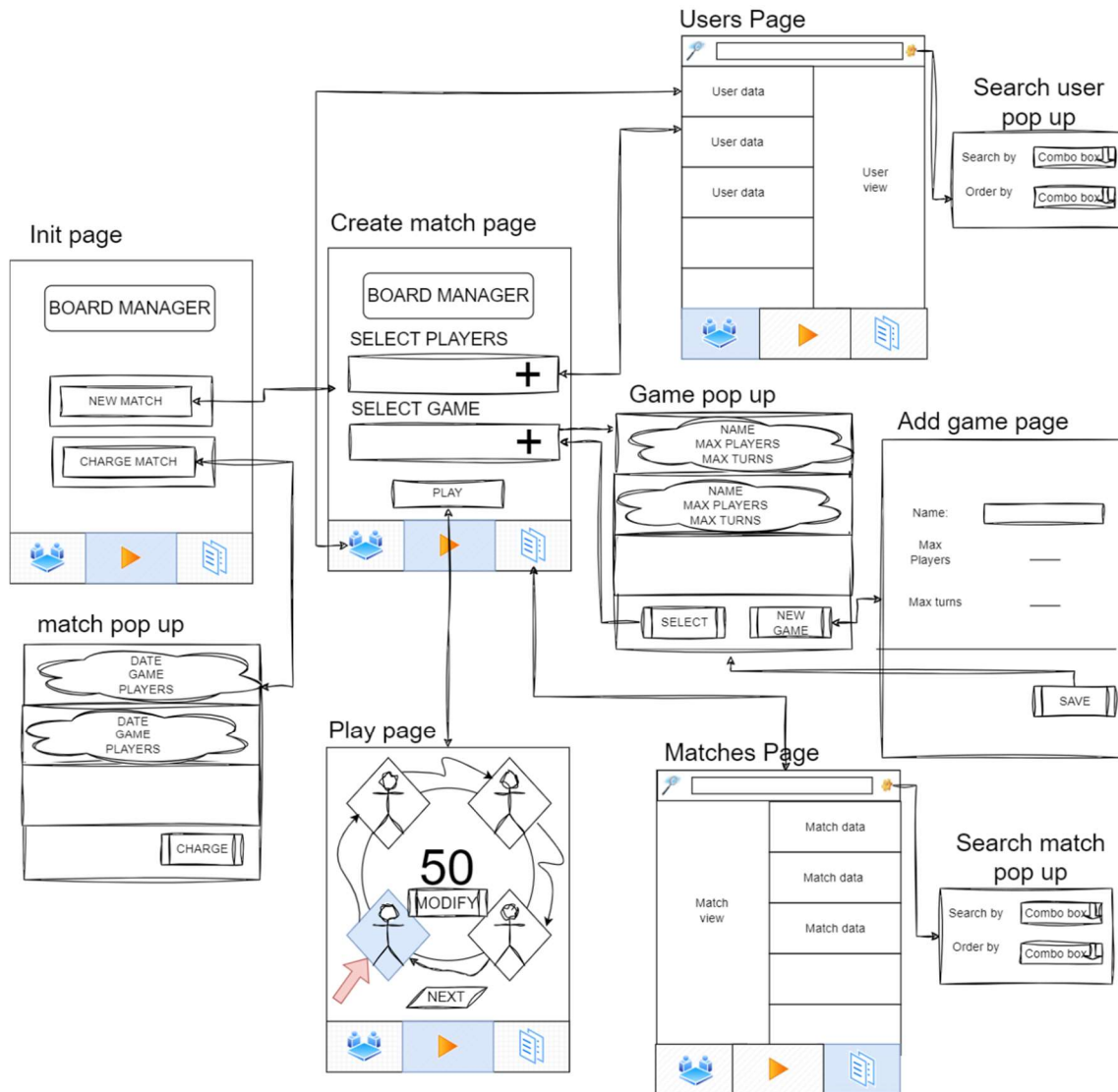


Figura 8: Diseño de interfaces, Fuente: Propia

Implementación

Tecnologías a emplear

Requerimientos, comparativa y justificación de las tecnologías elegidas.

Para el desarrollo de este proyecto, necesitaremos tecnologías que puedan cumplir con los siguientes requerimientos:

- **Desarrollo multiplataforma:** Necesitaremos una solución que nos permita crear nuestra aplicación para **Android** y **iOS** y más prescindible la posibilidad de hacerlo web.
- **Rendimiento rápido y fluido:** Le damos bastante prioridad a que nuestra aplicación tenga la cantidad mínima posible de interrupciones, con tiempos de carga bajos y una respuesta instantánea a las interacciones del usuario.
- **Interfaz de usuario innovadora:** Buscamos una interfaz de usuario que sea atractiva y diferente al resto.
- **Gran comunidad y recursos:** Dado que la empresa no tiene experiencia desarrollando con ninguna tecnología que cumpla los requisitos necesarios es importante contar con una comunidad activa de desarrolladores y una gran cantidad de recursos disponibles.

Las tecnologías que cumplen con los requerimientos son:

- **Flutter:** Sobresale gracias a su amplia gama de opciones a la hora diseñar y animar dando facilidad a crear interfaces innovadoras, sus aplicaciones son robustas y escalables además ofrece un rendimiento rápido y fluido.
- **React Native:** Destaca por su desarrollo rápido y su gran comunidad de apoyo, además se desarrolla en JavaScript.
- **Xamarin:** Se caracteriza por su rendimiento nativo, integración con el ecosistema de Microsoft y Visual Studio y flexibilidad en el diseño de interfaz de usuario.

Después de comparar las distintas opciones, hemos decidido utilizar **Flutter** junto con el lenguaje de programación **Dart**, que es el principal en el desarrollo de aplicaciones con **Flutter**.

También hemos tenido que hacer un análisis de las posibles formas de manejar nuestros datos y que además sea compatible con el resto de las tecnologías elegidas en nuestro caso **Flutter** y **Dart**.

Las opciones son:

- **Firestore**: Es una base de datos NoSQL en tiempo real proporcionada por Google, almacena los datos en la nube.
- **SQLite**: Es una biblioteca de base de datos embebida, ligera y con todas las funcionalidades SQL, almacena los datos en local.
- **Drift**: Es una biblioteca ORM para SQLite que facilita el uso de base de datos relacionales en Flutter, también almacena los datos en local.

Para elegir entre estas me he basado en lo siguiente:

- Usar **Firestore**, tiene el beneficio de que los datos se almacenan en la nube, pero tiene varios inconvenientes, como que es necesario tener conexión a internet, además al no ser de código abierto traería un costo adicional.
- Las otras opciones son bastante similares, dado que ambas son de código abierto, almacenan los datos en local, la diferencia radica en que **Drift** es un ORM, puede ser más complejo y menos simple, por lo que la curva de aprendizaje será mayor y para este proyecto específicamente preferimos la rapidez y menor abstracción que nos aporta usar directamente **SQLite**.
- Además, utilizar un **NoSQL** podría traer problemas que soluciona las bases de datos relacionales, puesto que podríamos asegurarnos de que no se dupliquen datos más fácilmente, además las consultas que accedan a diferentes tablas podrían volverse más complejas y la dificultad de mantener la integridad referencial entre las tablas sería un problema no justificado dado que no necesitamos los beneficios de una base de datos **NoSQL**.

En el siguiente apartado, se tratará más específicamente las tecnologías elegidas, es decir Flutter, Dart y SQLite y los motivos por los que estas son las indicadas para nuestro proyecto.

Descripción de las tecnologías utilizadas en el prototipo

Las tecnologías usadas en este proyecto son:



Figura 9: Logo Flutter, Fuente: (Flutter, 2024)

- **Flutter** es un framework de código abierto de Google para crear aplicaciones **multiplataforma**, agradables a la vista y compiladas de forma nativa a partir de un único código base gracias a esto se reduce significativamente el tiempo y los recursos necesarios para crear y mantener aplicaciones para diferentes sistemas operativos. El framework proporciona un conjunto de widgets personalizables y herramientas de diseño que permiten crear **interfaces de usuario innovadoras**, además la facilidad para implementar animaciones y transiciones fluidas hacen que las aplicaciones de **Flutter** sean visualmente atractivas.



Figura 10: Logo Dart, Fuente: (Dart, 2024)

- **Dart** es un lenguaje de programación versátil que admite tanto programación orientada a objetos como funcional, también desarrollado por Google, diseñado para ofrecer un **gran rendimiento** gracias a su compilador **Ahead-of-Time (AOT)** que es un método de compilación en el que el código se traduce a código de máquina antes de la ejecución. Tiene una sintaxis clara y concisa, similar a la de otros lenguajes populares, lo que lo hace **fácil de aprender**.



Figura 11: Logo SQLite, Fuente: (Sqlite, 2024)

- **SQLite** es una biblioteca de bases de datos embebida, liviana y de código abierto que implementa un motor de base de datos SQL autónomo y sin servidor, gracias a esto tendremos **portabilidad** en cualquier sistema operativo, muy necesario para el **desarrollo multiplataforma**.

En nuestro proyecto usaremos el paquete **sqflite** de **Flutter**, gracias a su antigüedad el paquete cuenta con una gran documentación y comunidad, recibe actualizaciones regularmente, ofrece un bajo consumo de recursos y un rendimiento alto además al no tener ninguna capa de abstracción intermedia puedes optimizar las consultas permitiendo realizar consultas SQL directamente.

Diario de desarrollo

Semana 1 – Instalación (01/04/2024-07/04/2024)

Tareas realizadas

Se ha realizado la instalación del SDK de Flutter, Dart y adaptado Visual Studio con Android Studio que son necesarias para usar Flutter y el entorno de Visual Studio Code donde se desarrollará el proyecto y se compilara la aplicación en el sistema operativo respectivo, también se ha añadido al entorno de desarrollo las respectivas extensiones de ambas tecnologías. Una vez finalizado se ha importado el paquete de sqflite para finalizar la instalación de todas las tecnologías antes mencionadas.

Dificultades encontradas

Una vez instalado todo lo necesario al intentar ejecutar la aplicación no encontraba Visual Studio Code el SDK de Flutter.

Al comprobar el correcto funcionamiento de la instalación de Flutter, el comando “flutter doctor” avisaba de que había varios problemas, uno con Visual Studio, dando problemas con las versiones y dos en Android Studio faltaba hacer una instalación aparte faltaban comandos de Android Studio que necesitaba Flutter y aceptar una licencia.

Solución

Para resolver el problema de Visual Studio Code después de buscar información he encontrado la solución, que era añadir una variable de entorno con la ruta al SDK.

El problema de la instalación de Flutter de Visual Studio, el comando indicaba correctamente la solución y se ha solucionado usando el installer de Visual Studio y añadiendo componentes específicos que faltaban.

El primer problema de Android Studio después de intentar solucionarlo desde Android Studio y no conseguirlo he tenido que buscar manualmente en el navegador los comandos que faltaban que eran los cmd-line-tools y se los he añadido en la ruta correspondiente ejecutándolos.

El segundo problema al parecer iba relacionado con el primero, gracias a los comandos que hemos añadido ahora el comando “flutter doctor --android-licenses” funciona correctamente, y podremos aceptar las licencias finalizando así la instalación.

Semana 2 – Conocimientos básicos Dart (08/04/2024-14/04/2024)

Tareas realizadas

Esta semana ha sido dedicada a aprender los conocimientos básicos necesarios para trabajar con Dart que a su vez es necesario para usar Flutter.

Se ha empezado por los fundamentos básicos de Dart, como las sintaxis de las variables, funciones, estructuras de control y clases también hemos comprobado los tipos de datos y colecciones que ofrece el lenguaje.

Además, en Dart las operaciones asíncronas son explícitas y se usan habitualmente para realizar operaciones como llamadas a la base de datos, de manera eficiente y sin bloquear la interfaz del usuario, por lo tanto, hay que entender su funcionamiento y aprender su uso dado que es algo muy necesario para el desarrollo de nuestro proyecto.

Dificultades encontradas

Los conceptos avanzados de la programación asíncrona en Dart y su aplicación en casos prácticos eran más complejos de lo esperado y ha habido dificultades para manejar correctamente la asincronía sin bloquear la interfaz de usuario.

Solución

Tras ver la documentación y buscar varias explicaciones se puso en práctica para ganar experiencia y entender la programación asíncrona y sus casos de uso.

Semana 3 – Conocimientos básicos Flutter (15/04/2024-21/04/2024)

Tareas realizadas

La semana se ha centrado en adquirir conocimientos básicos sobre Flutter, más específicamente centrado en el desarrollo Android.

Lo primero que hemos aprendido es la estructuración básica de una aplicación Flutter, puesto que al crear un proyecto se crean predefinidas varias carpetas y puede ser un poco complicado.

Una vez aprendida la estructuración hemos aprendido que en Flutter todo son widgets, que son los componentes de Flutter, por lo tanto, una aplicación de Flutter se forma con un árbol de widgets.

Tras poner en práctica algún widget y su funcionamiento, hemos terminado aprendiendo navegación entre pantallas y gestión del estado, que es la forma de manejar cambios en la interfaz de usuario de Flutter.

Dificultades encontradas

Las dificultades encontradas en Flutter han sido con la interfaz gráfica, ya que este tipo de interfaces suelen tener algún editor visual y no se suele modificar directamente el código, en cambio en Flutter se maneja directamente desde el código.

Solución

Gracias a las funciones hot reload y hot restart de Flutter que nos permite refrescar rápidamente la interfaz de usuario sin necesidad de recompilar toda la aplicación, gracias a esto se acelera el proceso de desarrollo.

Una vez optimizado el desarrollo hace falta practicar y familiarizarse con la sintaxis de Flutter.

Semana 4 – Base de datos (22/04/2024-28/04/2024)

Tareas realizadas

Para empezar el proyecto necesitaremos la base de datos, y cómo es local en la base de datos hay que integrar con Flutter y Dart, para hacerlo usaremos el paquete sqflite.

Lo primero que definiremos son las tablas de las bases de datos con sentencias SQL en funciones y las instanciamos abriendo una base de datos, además teniendo en cuenta que necesitaremos tener datos para probar la aplicación así que prepararemos otra función que le añadiremos que añada columnas a las tablas anteriormente creadas.

Dificultades encontradas

La sentencia que inicia la base de datos solo se deberá ejecutar una vez en nuestra aplicación, porque si no daría siempre error antes de iniciarse porque ya estén creadas o las sobrescribirá iniciándose.

La base de datos necesitaremos solo tener una instancia en nuestra aplicación para asegurarnos de la integridad.

Solución

Las dos opciones se han arreglado en la configuración de la base de datos, en el paquete ambos casos estaban contemplados por lo tanto con una configuración correcta se han solucionado fácilmente.

El primero se ha solucionado añadiendo la función al “onCreate” para que solo se ejecute cuando no hay base de datos en la ruta definida.

Y el segundo se ha modificado “singleInstance” a true, para que nuestra base de datos solo tenga una instancia.

Semana 5 – Inicio del desarrollo (29/04/2024-05/05/2024)

Tareas realizadas

Esta semana se empieza el desarrollo inicial donde se desarrollarán a un nivel más complejo los conocimientos básicos de Dart y Flutter y lo juntaremos con la base de datos anteriormente creada.

Necesitaremos una página inicial con los siguientes componentes:

- Una barra superior transparente donde ponga el nombre de la aplicación.
- Una imagen donde poner el logo ya que es la página inicial.
- Dos botones, uno que nos redirige a una página para crear una partida de un juego de mesa y el otro que nos mostrará las partidas que no han terminado para poder reanudarla.
- Una barra de navegación baja para dirigirnos a futuras páginas que no estarán implementadas en el prototipo, pero será necesario tener en cuenta ese espacio para que el prototipo se adapte al resultado final.
- Un fondo con gradiente.

Dificultades encontradas

Aun poniendo la barra superior en transparente tiene un mínimo de color.

Los botones y la barra de navegación por defecto no concuerdan con nuestra aplicación ni con el diseño que buscamos.

Solución

Para la barra superior se ha tenido que poner que el cuerpo se extienda por la barra, puesto que si esto no se especifica el cuerpo no está por debajo de la barra y al hacerlo transparente no tiene el color del cuerpo.

Para los botones se ha buscado un paquete que ofrezca botones más adecuados y además se ha usado en ellos una animación al pulsarlos.

La barra de navegación baja se ha usado un paint, que se usa para pintar entre los puntos indicados con la curvatura que se indique para darle una forma más original.

Semana 6 – Página crear partida (06/05/2024-12/05/2024)

Tareas realizadas

En la semana seis se ha hecho una página en la que buscas en la lista de jugadores y en la lista de juegos, seleccionas los que vayan a jugar la partida y con los datos seleccionados crea la partida insertándola a la base de datos.

Dificultades encontradas

El widget de Flutter en el que mostramos la lista solo permite hacer clic en un elemento mientras que para la lista de jugadores necesitaremos poder seleccionar varios.

Solución

Usando el paquete `multi_select_flutter` hemos conseguido poder seleccionar varios jugadores en el mismo widget y de esta forma poder crear la partida con todos sus jugadores.

Semana 7 – Página de la partida (13/05/2024-19/05/2024)

Tareas realizadas

Esta semana es muy relevante, las tareas realizadas han sido algunas de las funcionalidades de la gestión de la partida, una vez cargue la pantalla sacará de la base de datos las puntuaciones de los jugadores y su orden, sabrá qué partido o por el parámetro que se le pasa o cogiendo el ultimo insertado en su ausencia. Con estos datos mostraremos un DropDownButton, para mostrar al jugador que le toca, y su puntuación, además si hacemos clic en el se muestra el resto de los jugadores, necesitaremos un campo para introducir los cambios en la puntuación, un botón para modificarla, dos botones para ir un turno adelante o atrás y otro para guardar la partida

Dificultades encontradas

Una vez terminada la primera versión hemos realizado dos modificaciones, los botones que cambian el turno y el de modificar la puntuación.

Solución

Los botones que cambian el turno ahora calculan el siguiente usando el resto del total, de esta manera cada vez que sobrepase los límites pasará a la posición opuesta.

La funcionalidad añadida al botón de modificar puntuación es que saldrá un mensaje avisándonos que se ha modificado y automáticamente cambiara al siguiente jugador.

Semana 8 – Pruebas (20/05/2024-26/05/2024)

Tareas realizadas

En la última semana una vez terminado el prototipo se ha dedicado a tiempo a usar la aplicación como la usaría cualquier usuario, detectando de esta forma posibles mejoras para un futuro en cuanto a usabilidad y algún error sobre las funcionalidades que se han aplicado durante estas últimas semanas

Dificultades encontradas

Tras probar la página que crea un partido nos hemos dado cuenta que en el caso hipotético en el que se añada muchos usuarios ocupaba más del límite de espacio. Y la página se podía corromper.

Solución

La solución para arreglar el problema del espacio añadiendo usuarios ha sido un poco tediosa, puesto que la primera solución ha sido añadir un `SingleChildScrollView` al que ponerle de hijo nuestro widget anterior donde se deriva nuestra página, pero dado que no estaba establecido el tamaño de nuestro `Container`, este ocupaba media página y la otra la dejaba en blanco, para solucionar este comportamiento, el `Container` lo hemos tenido que envolver en otro widget padre que era un `ConstrainedBox` un widget en el que hemos podido añadir un constraint para que ocupara como mínimo la altura de la página es decir la página entera, de esta forma siguiendo la estructura se ha podido arreglar correctamente.

Pruebas

Procedimiento de evaluación, seguimiento y control del proyecto

Para asegurar que el software este construido correctamente hemos decidido realizar varias pruebas.

La primera será una prueba de rendimiento, dado que el arranque de nuestra aplicación es un momento crítico en el que se carga gran parte de esta y además se abre la conexión a la base de datos, teniendo en cuenta todo esto, aunque necesite bastante tiempo para arrancar gracias a estas pruebas podemos optimizarlo de la mejor manera y comprobar que tenga un arranque eficiente a mediados qué va avanzando el proyecto.

Después se realizarán varias pruebas unitarias sobre las funciones que actúan sobre la base de datos, de esta forma se comprobará que los datos se manejan correctamente que es fundamental para que la aplicación funcione correctamente.

Ambas pruebas serán automáticas y se realizarán gracias al paquete flutter-test de Flutter.

Procedimientos para la participación de los usuarios en la evaluación del proyecto

Para conocer la valoración del usuario de nuestra aplicación se fomentará la participación del usuario demostrándose desde un primer momento del proyecto que se les escuchará y que las peticiones harán que la aplicación mejore continuamente para satisfacer sus necesidades.

El método con el que recibiremos la mayor cantidad de retroalimentación será a través de las redes sociales. La aplicación tendrá en las redes sociales más comunes perfiles en los que creará una gran comunidad.

Las interacciones planificadas de las redes sociales serán las siguientes:

- Se subirán regularmente publicaciones de los nuevos avances de la aplicación.
- Votaciones en las que el usuario pueda elegir cuál será la implementación final entre las opciones.
- Encuestas en la que el usuario pueda decidir entre opciones en que se centre las siguientes actualizaciones.
- Se subirán post de preguntas y respuestas en los que podamos resolver dudas de los usuarios.
- Cada cierto tiempo compartiremos un enlace en el que puedan valorar nuestra aplicación.

Los métodos anteriormente mencionados se valorarán como pruebas de aceptación del usuario, pruebas de usabilidad y pruebas de cumplimiento de requisitos.

Conclusiones

En el apartado de empresa del proyecto se ha analizado correctamente el mercado y sus competidores, se ha realizado un estudio para llegar alcanzar a la máxima cantidad de clientes y se ha realizado una estimación de los gastos y beneficios teniendo en cuenta los requisitos legales de la empresa.

Y en el apartado técnico se ha establecido correctamente los requisitos que se tenían que cumplir y una metodología para conseguirlos, también se ha diseñado la aplicación, tanto como se iba a estructurar dentro del proyecto, la forma de almacenar los datos y cómo debería ser visualmente. Una vez definido todo se han elegido las tecnologías con las que se podría cumplir los objetivos anteriormente mencionados y se ha explicado el desarrollo del prototipo y los inconvenientes que han sucedido, también se han planificado las pruebas que serán necesarias para validar la aplicación.

Una vez que se ha realizado este proyecto mi valoración personal es que tiene un hueco en el mercado y que si se ejecuta correctamente llegando a la gente podría llegar a generar beneficios en no mucho tiempo.

Una posible mejora una vez terminado es que se pueda usar en varios dispositivos a la vez, algo que se ha planteado, pero no se le veía una mejora clara dado al hecho de que tengan que apuntar todo el curso de la partida, esto puede ser negativo puesto a que en los juegos de mesa es más común que una persona sea la que lleve el recuento de las puntuaciones de los demás, aun así, la aplicación podría llegar a ser más completa con más formas de ser usada.

Bibliografía

Dart. (01 de 05 de 2024). *Dart*. Obtenido de Dart: <https://dart.dev/>

Dobaño, R. (27 de 12 de 2023). *Quipu*. Obtenido de getquipu.com:
<https://getquipu.com/blog/cuanto-cuesta-darse-de-alta-como-autonomo/>

ejecutivos. (30 de 1 de 2024). *ejecutivos*. Obtenido de ejecutivos:
<https://ejecutivos.es/noticias/la-industria-de-los-juegos-de-mesa-crece-un-7-en-espana/>

Flutter. (01 de 05 de 2024). *Flutter*. Obtenido de Flutter: <https://flutter.dev/>

Gencat. (12 de 07 de 2022). *canalempresa.gencat*. Obtenido de canalempresa.gencat:
https://canalempresa.gencat.cat/es/01_que_voleu_fer/02_comencar_un_negoci/crear-empresa-constitucio-tramits/vull_ser_autonom/ser-autonom-avantatges-i-inconvenients/

Informatica metodologias. (11 de 09 de 2017). *Informatica metodologias*. Obtenido de Informatica metodologias:
<https://informaticametodologias.wordpress.com/2017/09/11/metodologia-incremental/>

Juguetes b2b. (08 de 07 de 2019). *interempresas.net*. Obtenido de interempresas.net:
<https://www.interempresas.net/Juguetes/Articulos/330708-Devir-artifice-del-exito-del-juego-de-mesa-moderno.html>

Sqlite. (01 de 05 de 2024). *Sqlite*. Obtenido de Sqlite: <https://www.sqlite.org/>

Workin space. (2024). *workin.space*. Obtenido de workin.space:
<https://workin.space/es/coworking/espana/valenciana/alicante/coalte>