

Infinity Gates

Project: Infinity Gates



SOLAR SOFTWARE



Project: Infinity Gates
Juan José Verdera Infante
Andrew Nauzet Beltrán Pérez
Proyecto de desarrollo de aplicaciones web

2. Índice de contenidos

Infinity Gates	1
2. Índice de contenidos	2
3. Introducción	4
3.1. Resumen del proyecto	4
3.2. Abstract	5
4. Análisis	5
4.1. Identificación del problema/necesidad	5
4.2. Búsqueda de información	5
4.2.1. The Binding of Isaac	5
4.2.2. Slay the Spire	5
4.2.3. Pokémon	6
4.2.4. Final Fantasy	6
4.3. Estado del arte	6
4.4. Razones que justifican el proyecto	6
5. Definición	7
5.1. Alternativas y propuestas de solución	7
5.2. Requisitos	7
5.3. Objetivos y alcance del proyecto	8
6. Diseño	8
6.1. Mecánica de juego	8
6.1.1. Flujo de Mazmorra	8
6.1.2. Flujo de combate	9
6.2. Esquemas	10
6.3. Sketches	10
6.4. Wireframes	10
6.5. Mockups	10
6.6. Diagramas UML	10
6.6.1. Diseño provisional de la base de datos relacional de infinite gates	11
6.6.2. Diagrama de flujo de combate	12
7. Planificación	13
7.1. Recursos	13
7.1.1. Materiales	13
7.1.2. Herramientas	13
7.1.3. Tecnologías	13
7.1.4. Servicios	14
7.1.5. Recursos humanos	14
7.1.6. Estimación temporal	14
7.2. Tareas a llevar a cabo	14
7.2.1. Elaboración de documentación del proyecto y de la aplicación	14
7.2.2. Desarrollo de aplicación cliente	14
7.2.3. Creación de servidor	14
7.2.4. Servidor de base datos	15

7.3. Metodologías	16
7.4. Estimación de costes	17
7.4.1. Gastos iniciales	17
7.4.2. Mensualidades	17
7.5. Validación	17
8. Conclusiones	18
8.1. ¿Se han cumplido los objetivos?	18
8.2. Dificultades encontradas	18
8.2.1. Un workframe nuevo	18
8.2.2. No somos diseñadores	18
8.2.3. Trabajar con móviles	18
8.3. Posibles mejoras	18
9. Bibliografía	18
10. Anexos	19
10.1. Memoria de ejecución y evaluación	19
10.2. Instrucciones de uso	19
10.3. documentación	19
10.4. etc	19

3. Introducción (Revisar completo)

3.1. Resumen del proyecto

Infinity Gates es un juego del género Roguelike, en este proyecto hemos juntado varias ideas de distintos juegos del género para crear nuestro juego con nuestro estilo combinado; pensado principalmente para usuarios de móvil(Aunque también estará disponible para otras plataformas), para la elaboración de este proyecto utilizamos tecnologías que nos permiten elaborar nuestra aplicación en múltiples plataformas simultáneas, usando React Native. Todo esto con el objetivo obtener un producto que pueda satisfacer las necesidades de entretenimiento de un público en concreto, aquellas personas que le puedan importar nuestro producto.

3.2. Abstract

Infinity Gates is a game of the genre Roguelike, in this project we have merge various ideas from different games of the genre to create our own with our own combined style; thinking mainly for the mobile users(Although it will be available for other platforms), for the making of this project we use technologies that allow us to develop our app in different platforms simultaneously, using React Native. Everything with the main goal of making a product that could satisfy the needs of entertainment for the customers that will concern us about our product.

4. Análisis(Revisar completo)

4.1. Identificación del problema/necesidad

La necesidad es el entretenimiento humano y la creación de un juego de un estilo único multijugador que no existe en el mercado actual; que pueda satisfacer a una variedad de públicos que puedan estar buscando nuestro juego para su disfrute.

4.2. Búsqueda de información

Para la inspiración de nuestro proyecto hemos estado valorando anteriormente distintas opciones y posibilidades que realizar para este trabajo, al final nos quedamos con esta idea y para su consecuente diseño y desarrollo nos hemos basado en otros juegos populares del género, tales como The Binding of Isaac, Slay the Spire, Pokémon y Final Fantasy; a continuación explicaremos las influencias de cada uno de estos en nuestra idea.

4.2.1. The Binding of Isaac

De este juego hemos sacado la idea de que cada una de las “runs” sea independiente la una de la otra, es decir, se nos ocurrió que en una run podrías obtener objetos o herramientas que se puedan utilizar solo dentro de la propia run, pero conservaremos el valor que hacer runs a pesar de no vencerlas, es decir, fuera de la propia dungeon el jugador tendrá su propio equipo con el que poder influenciar cómo va de preparado a la mazmorra, dando importancia a la preparación antes de hacer una run y a que se puedan utilizar herramientas dentro de esta.

4.2.2. Slay the Spire

Este juego nos inspiró enormemente en cómo diseñar un Roguelike disfrutable para móvil, hablando en general de esta inspiración, es cierto que hemos sacado bastantes cosas de este juego que no tiene otro similar, ni en estilo ni en éxito, pero, el problema es que este es un juego pensado exclusivamente para una persona, al contrario que nuestro propio juego, siguiendo el razonamiento de “Si una persona lo disfruta, con dos personas disfrutas el doble”.

En específico de este juego hemos sacado nuestro sistema de combate que será explicado posteriormente, está basado en ese mismo juego y recoge influencias de otros juegos de cartas de combate como turnos como puedan ser Hearthstone o Legends of Runaterra. El sistema de mazmorras es muy parecido al de Slay the Spire, pero con un par de adiciones y mejoras para hacer la aventura más dinámica y cada run distinta la una de la otra y por último también vamos a utilizar habilidades pasivas que se desarrollan fuera de la mazmorra, esto de nuevo le dará importancia a las decisiones del jugador, pero tampoco planeamos que hayan decisiones claramente mejores que otras.

4.2.3. Pokémon

Aunque en los juegos anteriores existen diferentes personajes para el uso de cada run, hay un elemento que no se emplea en los anteriores y es usar diferentes personajes con mecánicas distintas en dungeons, Pokémon es el ejemplo más claro de este estilo, en Pokémon tu puedes hacer combates individuales e ir cambiando de “personajes”, pero también puedes hacer

combates de 2vs2 o 1vsN, hemos pensado en el sistema de combate y lo hemos descartado pero gracias a pensar en ello se nos ocurrió la idea de que cada Dungeon puede tener diferentes modalidades como por ejemplo: 1, 2 ,3 ,4 personajes simultáneamente y el nivel de desafío o el diseño de la dungeon cambiaría de acorde al número de personajes, además de que esto nos permitirá añadir interacciones entre personajes, esto sobretodo satisface la necesidad de los usuarios coleccionistas, es decir, aquellos usuarios que prefieren coleccionar personajes o tener un equipo que desarrollar en un juego, tal y como es Pokémon.

4.2.4. Final Fantasy

Final Fantasy es un juego que no se puede considerar como Roguelike por carecer elementos importantes del género, aún así todos los géneros tienen partes de otros géneros, en este caso escogemos el elemento RPG de esta saga, los personajes de nuestros jugadores podrán gozar de diferentes habilidades según su clase y que podrán ser desbloqueadas por nivel además, el jugador podrá equipar diferentes piezas de equipo a su personaje, el equipo podrá dar diferentes propiedades o ayudas al jugador en sus aventuras y ayudarán a definir a qué se puede enfrentar y dar la posibilidad de establecer “Builds” empleando habilidades y equipo que puedan satisfacer al público softcore que les guste este tipo de cosas, haciendo que puedas optimizar tu personaje, pero también simplemente puedes escoger lo que más te gusta e ir con ello, sin obligar a seguir reglas predeterminadas hechas por la comunidad.

4.3. Estado del arte (situación actual)

El género Roguelike consiste en una gran variedad de juegos distintos que acaban siendo muy influyentes a nivel personal y bastante queridos, pero el género tiene una deficiencia y es que no hay casi elecciones en cuando se trata de jugar con amigos a este tipo de juegos, que los juegos multijugador sean competitivos o cooperativos mantienen la vida de un juego sin necesidad de grandes actualizaciones o paquetes de contenido de algún tipo sean gratis o de pago, en nuestro caso creando un juego que junta el elemento online con la aventura solitaria del Roguelike y poder compartir la experiencia con amigos alargaría la vida del juego y podría atraer una buena parte del público por los motivos anteriores, además pensando en que nuestro juego está pensado para dispositivos móviles y en los jugadores casuales que conforman la inmensa mayoría del mercado al que queremos llegar, es decir, teniendo en cuenta lo anterior, haciendo un juego de partidas cortas pero disfrutable es la clave para ganarnos a este público tan grande.

4.4. Razones que justifican el proyecto

Nuestro proyecto se cimenta en las bases de que no existe un juego igual al que nosotros planteamos, eso significa que estamos creando un producto innovador, como se ha explicado en apartados anteriores, creemos que puede tener éxito por el amplio mercado al que apuntamos y el público objetivo que pueda estar interesado en nuestro producto que consideramos como una mejora de varios juegos creando uno distinto de ellos, la gente que ya está familiarizada con esta idea simplemente podrá cambiarse a usar más nuestro producto del que ya usaran anteriormente y algo que planeamos con este juego añadiendo la interacción entre jugadores es crear una comunidad integrada sin necesidad de encontrar grupos por internet con los que poder jugar o hablar del juego, haciendo esto dentro del propio juego cimenta la relación entre jugadores y su relación con el juego.

5. Definición(Por revisar)

5.1. Alternativas y propuestas de solución

Como se ha mencionado anteriormente, nuestra idea es una que no existe actualmente específicamente como la hemos desarrollado, pueden existir juegos parecidos pero no uno igual, pero pueden ser de géneros diferentes o con objetivos distintos, como por ejemplo Divinity: Original Sin 2, puede ser visto como un juego RPG en el que pueden jugar hasta 4 jugadores al mismo tiempo en el que cada uno lleva su personaje y van de aventuras, pero si como contraste ponemos nuestra idea nos damos cuenta que una vez que acabas la aventura de Divinity: Original Sin 2 no hay nada más, el cooperativo es solo para 4 personas y no interactúas con gente por fuera de tu grupo con el que juegas, estas son diferencias a grandes rasgos de nuestro juego con otro, que hay más diferencias específicas como que nuestro juego se basa en el combate por turno de cartas y no en el combate del juego que hemos usado de ejemplo que se basa en turnos por puntos de acción.

Como otras propuestas para solucionar el vacío que cubre la idea que desarrollamos hemos pensado en básicamente la misma realización pero con diferentes componentes y cambios en cosas como el sistema de combate, el sistema de combate por turnos con cartas es algo por lo que nos decidimos al final dada la comodidad de móvil, la simpleza y el toque de aleatoriedad que le presta al juego haciendo que tengas que pensar qué debes hacer.

5.2. Requisitos

Aunque nuestro objetivo y requisitos distan bastante uno del otro, creemos que podríamos cumplir con los requisitos con un par de meses de tiempo, pero lo que hemos planeado para nuestro juego es lo siguiente:

- Sistema de identificación y registro
- Sistema de creación de personajes
- Sistema de mazmorras (1, 2, 3, 4 jugadores)
- Sistema social (Chat global, amigos, guilds, mazmorras cooperativas)
- Sistema de administración de cuentas
- Sistemas de administración de personajes
- Plataforma de pago
- Despliegue en plataformas móviles y de escritorio
- Interfaz multilenguaje(Inglés, Chino simplificado, Español, Alemán, Japonés)

5.3. Objetivos y alcance del proyecto

Nuestro objetivo en este proyecto es una versión funcional del sistema de cliente, servidor, api y base de datos, esta versión contendrá los siguientes objetivos:

- Registro e identificación
- Creación de personajes
- Mazmorra de un jugador de un solo personaje
- Equipamiento de personajes
- Inventario de cuenta

- Despliegue en móvil de la versión funcional

Creemos que nuestro proyecto es bastante grande y complejo de realizar especialmente porque estaremos aprendiendo un workframe del lado del cliente con el que no hemos trabajado anteriormente y todavía en nuestro diseño inicial para empezar el desarrollo tenemos varios puntos con los que todavía dudamos si cambiarlos incluso a mitad de proyecto, como el motor de base de datos, en principio usamos MongoDB, pero quizá al ser una base de datos no relacional nos hace replantearnos nuestra elección, dado que sí que se pueden hacer relaciones en MongoDB pero no son unas relaciones reales.

Nuestros propios objetivos están sujetos a cambios principalmente por los factores de los que dependemos, que es el tiempo que tengamos disponible para poder realizar la ejecución hasta la presentación, es por ello que esta lista es por lo menos lo mínimo que se podrá ver en la demo y será ampliable si podemos plantearnos nuevas metas después de cumplir las actuales.

6. Diseño(Por revisar)

6.1. Mecánica de juego

6.1.1. Flujo de Mazmorra

Al darle a jugar entraremos en una run, tendremos diferentes rutas que seguir hasta llegar al boss final de la run, en las rutas habrán salas, las cuales pueden ser, (**combate**, **combate élite**, **tesoro**, **tienda**, **hoguera**, **misión**, **sala misteriosa**).

Combate: será una lucha contra enemigos normales que darán como recompensa por acabar con ellos las siguientes cosas:

- cartas para añadir al mazo actual
- oro para la run actual

Combate élite: será una lucha contra enemigos más difíciles con una mecánica especial, ejemplo: cada vez que recibe daño aumenta su ataque. Darán como recompensa las siguientes cosas:

- cartas para añadir al mazo actual
- oro para la run actual
- Aolita(dinero global de la app)
- pasiva aleatoria

Tesoro: sala donde se conseguirá:

- pasiva aleatoria
- oro para la run actual

Tienda: sala donde se podrá comprar con el oro de la run actual lo siguiente:

- cartas para la run actual
- pasivas
- pociones
- Aolita(dinero global de la app)

Hoguera: permitirá realizar alguna de las siguientes opciones:

- eliminar carta del mazo actual
- curar un porcentaje de la vida del personaje
- aumentar la vida máxima del personaje esta run

- mejorar una carta del mazo

Misión: sala donde se encomendara al usuario realizar una misión en una mazmorra definida pues al completarla darán recompensas:

- Aolita(dinero global de la app)
- desbloquear equipo
- desbloquear pasivas

Sala misteriosa: en esta sala podremos encontrarnos...

- eventos especiales, por ejemplo un goblin amigable que nos ofrece un trato
- una sala de tesoro
- una sala de tienda
- un combate

6.1.2. Flujo de combate

Inicio de combate

Un combate será iniciado cuando el jugador tenga un encuentro con un enemigo que podría ser dado por casillas de: combate, misión, evento especial, sala misteriosa, combates de élite o llegando a la parte final de la mazmorra.

Combate por turnos

El combate será dado por turno, para jugador/es, turno para enemigos.

En el turno de jugador que será el primero, el jugador tomará un número aleatorio de cartas, cada carta tendrá un valor de punto de acción, es decir, el jugador está limitado por el número de puntos de acción (Cada clase podría llamarse de otra forma ej: Guerrero-Stamina), los puntos de acción se utilizan para utilizar cartas/acciones.

Se cambiará de turno automáticamente cuando el jugador no puede utilizar cartas por los siguientes motivos:

- No tener más cartas
- El coste de las cartas supera los puntos de acción disponibles

Una vez finalizado el turno, si quedan puntos de acción disponibles, se irán a un pool de puntos de acción extra con un máximo de puntos guardables definido.

Después del turno del jugador, tomará turno de los enemigos, los enemigos en su turno dependiendo de sus habilidades podrán:

- Atacar
- Colocar un debuff
- Preparar una acción de múltiples turnos

6.2. Esquemas

lorem ipsum dolor sit amet.

6.3. *Sketches* (bocetos)

lorem ipsum dolor sit amet.

6.4. *Wireframes* (esquemas de página o plano de pantalla)

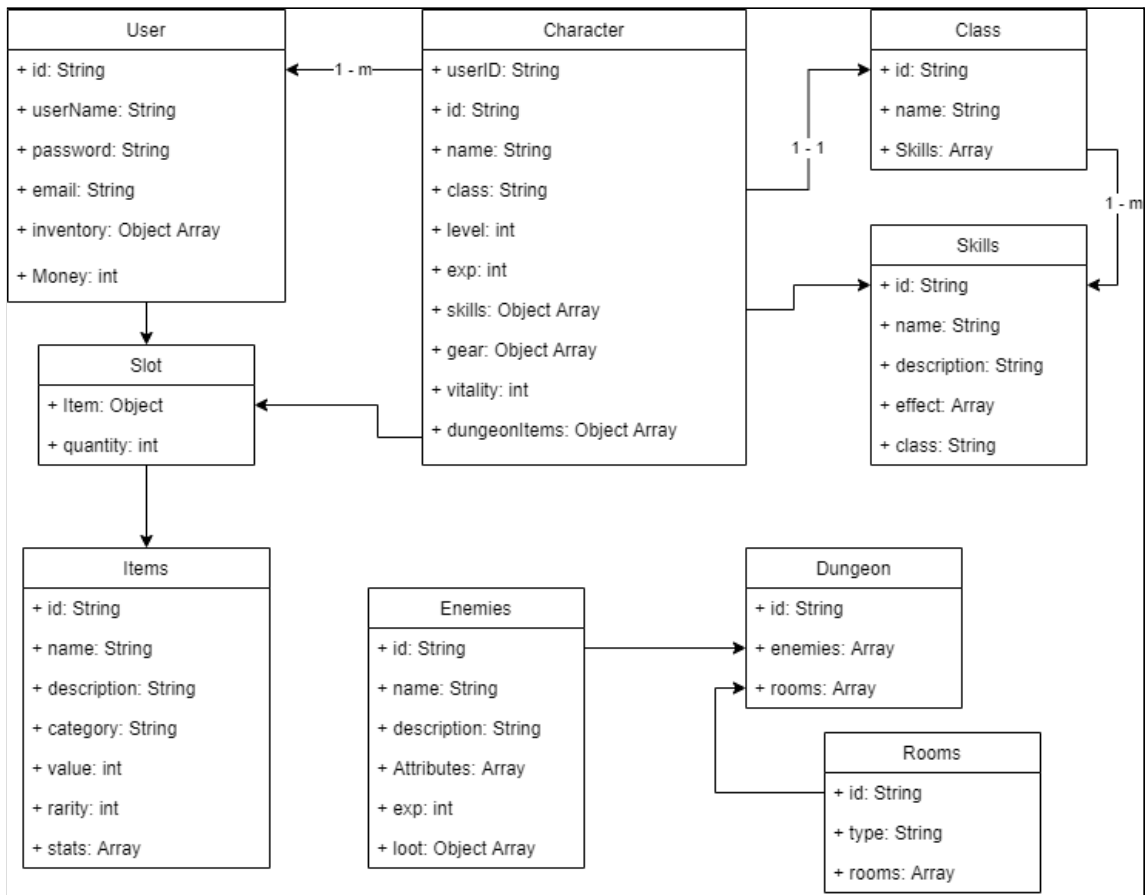
lorem ipsum dolor sit amet.

6.5. *Mockups* (prototipos)

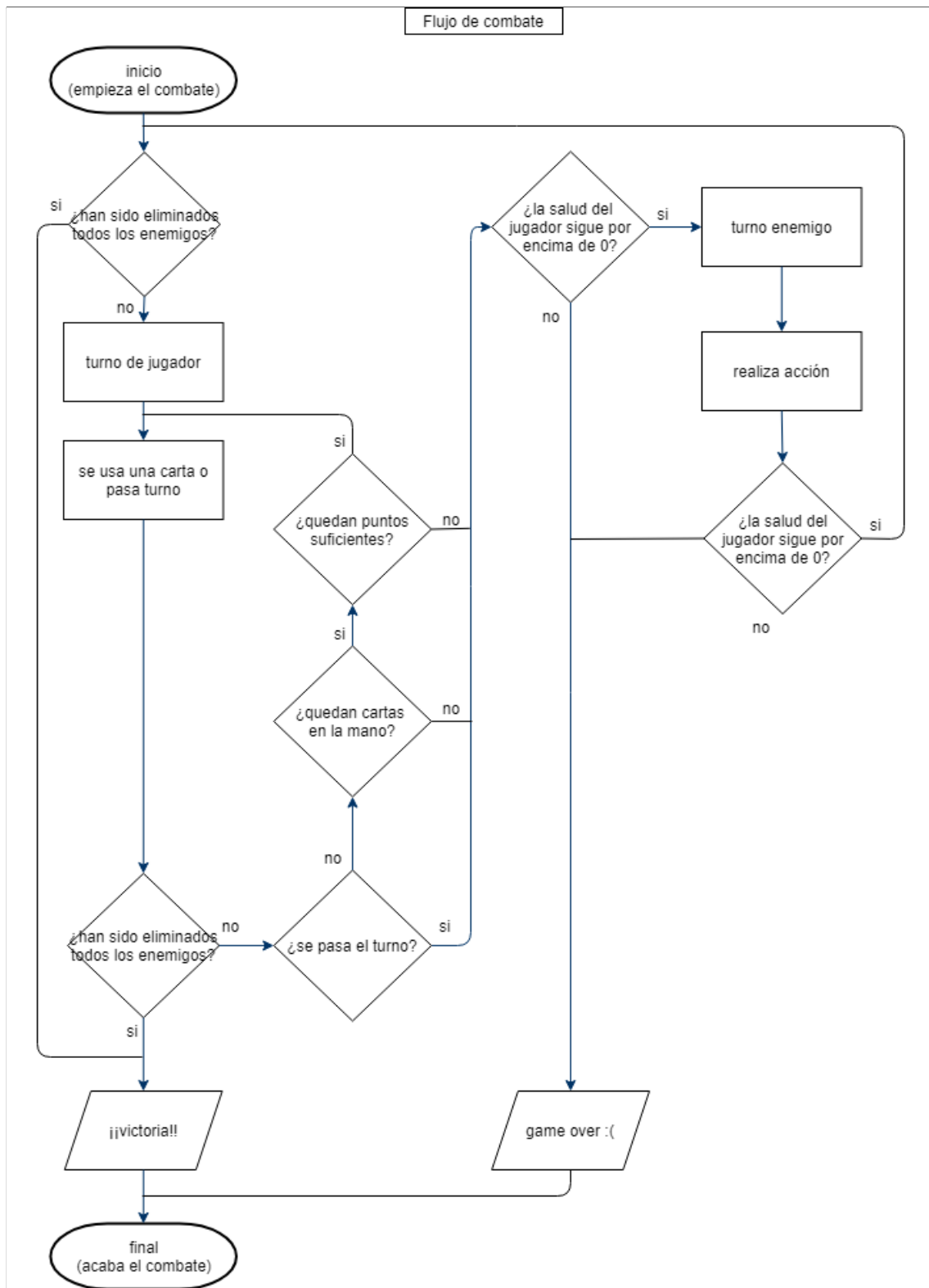
lorem ipsum dolor sit amet.

6.6. Diagramas UML (de casos de uso, de flujo, de clases, despliegue, componentes, etc.)

6.6.1. Diseño provisional de la base de datos relacional de infinite gates (Revisar)



6.6.2. Diagrama de flujo de combate



7. Planificación (revisar)

7.1. Recursos

7.1.1. Materiales (equipos informáticos, etc.)

- 2 Equipos informáticos: Serán necesarios para que nuestro equipo de programadores puedan trabajar en paralelo sin necesidad de depender de tener solo una máquina.
- 1 Dispositivo móvil para pruebas: En una prueba de despliegue real será necesario utilizar al menos un dispositivo móvil de sistema operativo Android, sería opcional obtener otro dispositivo móvil iOS para pruebas.
- Raspberry Pi 4B

7.1.2. Herramientas (editores, IDEs, *software*, etc.)

- Visual Studio Code: VSCode será nuestro editor de código debido a su simpleza, estilo visual, la capacidad de adaptar nuestro entorno de trabajo a las necesidades del proyecto usando addons o personalizando el propio
- Photoshop: Se ha utilizado para dibujar los primeros bocetos de diseño sobre las vistas de la aplicación, creación de logo entre otras expresiones de arte.
- Adobe xd: La herramienta seleccionada para la creación de wireframes/mockups

7.1.3. Tecnologías (lenguajes de programación, sistemas operativos, sistemas gestores de bases de datos, etc.)

- Windows 10 profesional: Sistema operativo en el que se programará el código, debido a su disponibilidad y familiaridad de los programadores con el sistema operativo.
- JavaScript: El lenguaje de programación que se utiliza en todas las webs del mundo, es por ello por lo que nuestra aplicación web debe utilizar JavaScript, además de ser un lenguaje con el que los programadores web deben de estar familiarizados, lo mejor de JavaScript es estar orientado a eventos, la cantidad de workframes y repositorios de paquetes de la comunidad con los que podemos trabajar.
- Node.js: Workframe para JavaScript del lado del servidor, en un juego el punto principal de la relación cliente-servidor es la interacción entre ambos, es por eso que Node.js al venir de JavaScript nos hace pensar que esta es nuestra mejor opción dado que parece encajar perfectamente con nuestra idea.
- React Native: React Native es la versión de lenguaje híbrido del workframe React, el workframe más utilizado de JavaScript, ha sido escogida con el workframe de Expo por un motivo que nos ahorra tiempo de programación y es que React Native es un lenguaje de programación que sirve para cualquier plataforma, dado que cuando se compila se convierte en código html, css y JavaScript que puede leer un navegador web o se traduce a Java que lo puede leer un dispositivo Android
- MongoDB: Este motor de base de datos es el más utilizado en cuanto a proyectos realizados con los objetivos que nosotros tenemos, es decir, es especialmente bueno para juegos, debido a su funcionamiento, aunque encontramos varios inconvenientes y es que nos gustaría más un modelo relacional y con schemas, es por eso que nos planteamos cambiar de motor de base de datos

7.1.4. Servicios (*hosting*, control de versiones, *quality assurance* o *testing*)

- Github: El control de versiones más utilizado en el mundo de la programación que se podría considerar standard también como repositorio para alojar el control de versiones.
- Servicio de hosting para base de datos: El servicio a elegir dependerá de qué motor de base de datos utilizamos para nuestro proyecto ya que muchas veces el gestor de uno ofrece el propio servicio para hostear una base de datos y utilizarla, al contrario que nuestro servidor intermediario, si que creemos que un servicio externo sería más seguro

7.1.5. Recursos humanos

- 2 Programadores: Sin programadores no podemos ni empezar a hacer un prototipo para el proyecto, podrían incluirse otros tipos de especialidades como música, modelaje, diseño... pero para el alcance del proyecto no planeamos emplear ninguno de los anteriores
- Reviewers: En cualquier cantidad serán necesarios para obtener un feedback de nuestro proyecto

7.1.6. Estimación temporal

- Despliegue modelo funcional: 1 mes
 - desarrollo de documentación inicial: 1 semana
 - backend: 4 días
 - front end 2 semanas
- Despliegue de alpha: 2 meses
- Despliegue de Beta: 3 meses
- Despliegue versión completa: 3 meses

7.2. Tareas a llevar a cabo

7.2.1. Elaboración de documentación del proyecto y de la aplicación

Será necesario terminar el documento actual para obtener una imagen clara del diseño y desarrollo de nuestra aplicación con la que empezaremos a trabajar, además de esta documentación del proyecto en sí mismo vamos a necesitar elaborar una documentación sobre la propia aplicación y cómo funciona cada método, el manejo de la api, la estructura de la base de datos.

7.2.2. Desarrollo de aplicación cliente

Para la elaboración del proyecto será necesario hacer una aplicación del lado del cliente que pueda funcionar en dispositivos de escritorio y móviles, además de que se pueda poder descargar o instalar de alguna manera, para ello se tendrá que hostear la versión web en un dominio y las versiones de móvil se podrán descargar utilizando una apk, por ejemplo por lo menos para la versión de Android.

7.2.3. Creación de servidor

La creación de un servidor y su API para la comunicación entre base de datos y cliente es algo que debe ser construido en orden para que toda la aplicación funcione correctamente, la API

servirá para definir la comunicación y como se trata esta misma, además de ser capaces de poder asegurarnos que la información enviada a nuestro servidor y base de datos es fiable y utilizable en caso de tener usuarios con intenciones malignas sobre nuestro programa.

El servidor como se ha escrito anteriormente en las tecnologías que se van a utilizar va a ser node debido a que su lenguaje base está muy preparado para los eventos y la reacción antes estos lo cual en un juego nos parece bastante importante.

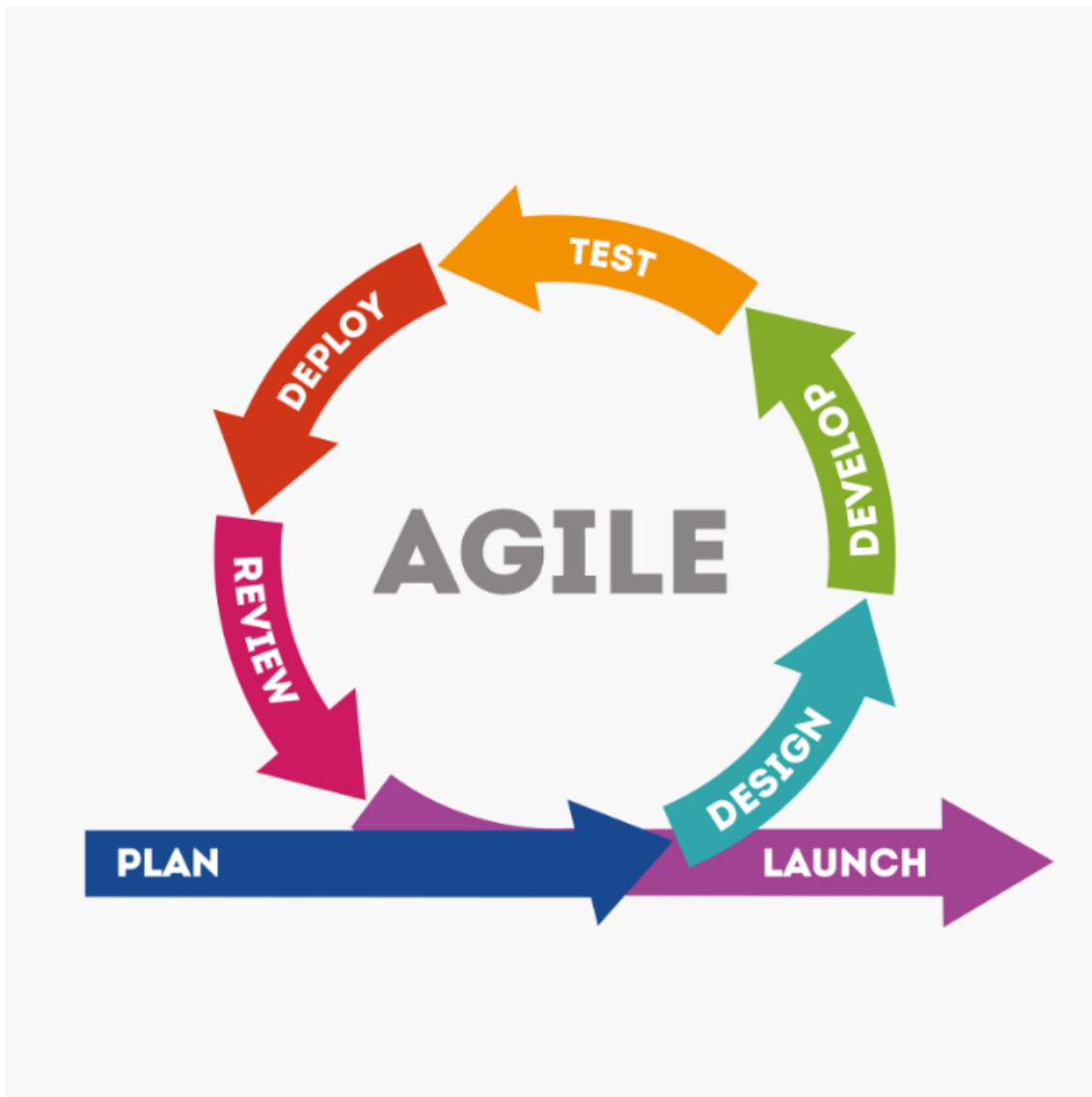
7.2.4. Servidor de base datos

Un lugar donde poder hostear nuestra base de datos será imperativo tenerlo dado que es donde se guardará la información de usuarios, en este caso sí que usaremos un servicio externo, dado que el servidor lo podemos migrar en cualquier momento, una base de datos es más complicado. Será fácil migrar la base de datos o cambiarla durante el proceso de desarrollo inicial dado que todavía no tendremos grandes volúmenes de datos guardados, lo cual nos haría replantearnos si cambiar de motor de base de datos y servicio de hosting para la misma.

7.2.5. Cronograma

lorem ipsum dolor sit amet.

7.3. Metodologías



Este esquema representa completamente el cómo vamos a desarrollar nuestro proyecto, nos basamos en la programación funcional para ir construyendo modelos funcionales cada vez más funcionales.

Explicando nuestra metodología con más profundidad, nosotros tenemos un plan, un objetivo que hacer y al que llegar como mínimo, diseñamos el cómo va a ser, los desarrollamos, vamos probando los módulos, después probamos cómo funciona en conjunto y revisamos nuestro proyecto, así continuamente hasta llegar a una aplicación que se pueda lanzar al mercado, nos gusta tener un plan flexible e introspectivo dado que al ser un proyecto tan grande debemos asegurarnos que el conjunto siga funcionando y pueda soportar nuevas funcionalidades.

El cómo hacemos nuestro diseño se basa en nuestro plan, para rediseñar algo nos basamos en pruebas y opiniones externas además de influencias que podamos obtener o nuevas ideas.

Nuestro método de desarrollo se basa en el peer-2-peer para la escritura de código del cuál no estamos familiarizados, pensamos que con este método podemos sobrepasar los obstáculos que se nos ponen en el camino relacionados con aprender un nuevo lenguaje o en nuestro caso un nuevo workframe.

Para el desarrollo regular de otros componentes de nuestra aplicación simplemente trabajaremos de forma paralela en partes distintas de la aplicación, ya que estaremos mucho más familiarizados con los objetivos que tenemos que llegar y cómo realizarlos y para la review del código que escribamos por separado simplemente lo comprobará el otro además de que individualmente tendremos que asegurarnos que un módulo hace y cumple lo que debe de hacer aunque sea de forma mínima.

El despliegue de la actualización en las fases de desarrollo consistirá en el alojamiento virtual de servidor, base de datos y la ejecución en emuladores de la versión de cliente y la aplicación en el cliente debe de estar en un repositorio al que podamos acceder de esta forma podremos trabajar a distancia y con diferentes ritmos de trabajo, y en caso de no estar trabajando con nuestros equipos habituales podremos acceder a todo nuestro proyecto en distintas localizaciones sin ningún tipo de problema.

7.4. Estimación de costes

Los costes han sido estimados a la fecha de Marzo de 2021.

7.4.1. Gastos iniciales

Para desarrollar el código no se requieren equipos informáticos de gran potencia, por ello podríamos prescindir de tarjeta gráfica dedicada, cada uno nos podría costar unos 550€.

El dispositivo móvil Android para pruebas ha de ser de gama media como mucho, queremos que todo el mundo pueda jugar sin problemas a nuestro juego, el cual no debería pedir mucho para funcionar correctamente, su coste debería rondar los 150€.

En cuanto al dispositivo móvil iOS, en caso de querer realizar pruebas nos aumentaría bastante el coste así que obtener uno de hace un par de años podría ser buena idea, el modelo más barato actualmente que vende Apple oficialmente cuesta 449,01€.

La Raspberry Pi 4B tiene varios modelos que se diferencian en la ram, 2GB con un coste de 28,88€, 4GB con un coste de 45,38€ y la de 8GB con un coste de 61,88€.

El coste inicial estimado es desde 1.128,88€ a 1.960,89€.

7.4.2. Mensualidades

(Se podría mirar el crear otro apartado para pagos únicos, ejemplo, un equipo de traductores o diseñador gráfico para la interfaz)

(Photoshop, Adobe XD) Para empresas cada una 29,39€, para estudiantes 19,66€.

(por confirmar si requiere indicarse un coste para este apartado) Programadores, 2000€/mes?

Artista musical, disponer de un profesional para el sonido del juego suena muy bien pero para este proyecto vamos a prescindir de ello ya que no planeamos crear ninguna banda sonora...(completar o cambiar esto en un futuro que queda feillo)

Artista gráfico, aunque no es un proyecto muy grande, necesitaríamos tener una interfaz propia bonita y diseños de cartas únicos, disponer de un artista gráfico sería primordial para una versión avanzada del proyecto, contratar uno al comienzo del desarrollo de la beta para trabajar junto al equipo sería una opción a considerar.

Traductores

7.5. Validación (test unitarios y test de integración)

lorem ipsum dolor sit amet.

8. Conclusiones

8.1. ¿Se han cumplido los objetivos?

Hemos cumplido el alcance que nos habíamos propuesto.

8.2. Dificultades encontradas

Como dificultades que hemos encontrado podemos destacar las siguientes, aunque algunas ya habíamos anticipado desde la planificación que iban a ser un obstáculo que superar.

8.2.1. Un workframe nuevo

Trabajar sin un guía con algo completamente desconocido aunque divertido es complicado y tiene sus momentos, el desconocimiento implica muchos puntos en contra algunos peores que otros, principalmente no conocemos las capacidades totales de lo que estábamos haciendo, cuál era el límite o la mejor forma de realizar algo, aunque nos enorgullece el resultado al que hemos llegado.

8.2.2. No somos diseñadores

El cómo nos hemos llegado a adaptar a esta situación de que no somos expertos en materia ni mucho menos ninguno de los integrantes del equipo se podría decir que tiene un buen nivel en cualquiera de las áreas de dibujo, música y modelaje, pero con las herramientas adecuadas, fallo y error además de sumar la experiencia previa que aunque no nos hiciese profesionales en los campos sí que facilita que podamos llegar a una resolución mejor cuando se trata del apartado gráfico de nuestra aplicación.

8.2.3. Trabajar con móviles

Ha sido la primera vez que trabajamos con móviles o un despliegue multiplataforma y para solventar este problema pues decidimos utilizar caminos que nos facilitaban esto mismo, como

por ejemplo el workframe del lado del cliente que utilizamos para el despliegue que nos permite compatibilidad con el mismo código en distintas plataformas y al principio la verdad es que nos encontrábamos perdidos de cómo empezar a escribir el lado del cliente pensando en un móvil, no resultó ser tan complicado pero no conocíamos el primer paso a dar.

8.3. Posibles mejoras

Obtener un diseñador gráfico, dibujante, músico, modelador y traductores.

9. Bibliografía

lorem ipsum dolor sit amet.

10. Anexos

10.1. Memoria de ejecución y evaluación

lorem ipsum dolor sit amet.

10.2. Instrucciones de uso

lorem ipsum dolor sit amet.

10.3. documentación

lorem ipsum dolor sit amet.

10.4. etc

lorem ipsum dolor sit amet.