**Resumen de la Planificación   
del desarrollo del Proyecto**

**Metodología ágil de desarrollo**

Vamos a llevar a cabo el desarrollo del videojuego con una variante personalizada de lo que puede ser una metodología ágil.  
En la que vamos a tener en todos los inicios de semana una reunión, en la que los miembros del grupos pongan en común todos los avances realizados en la semana.

**Fases de desarrollo**

Vamos a definir tantas fases como semanas dure el desarrollo del videojuego.   
Y en cada una de ellas habrá una serie de actividades a realizar por parte de los integrantes del equipo. Siendo publicadas las mismas antes de empezar la semana (domingo/lunes).

**Desarrollo iterativo**

En función de como se vayan desarrollando esas actividades, el jefe del proyecto, planificará las actividades de la siguiente fase, para ajustar la carga de trabajo y ajustarnos a lo desarrollado en cada momento, lo que disminuye la complejidad y aumentas las probabilidades de conseguir un videojuego completo y consistente… y no fallar en el intento.

Es decir, durante el transcurso de una fase, los miembros del equipo irán implementando sus tareas, apoyándose en el resto del equipo y en el jefe de proyecto. Y el jefe del proyecto irá revisando el trabajo realizado, con el objetivo de (1) ayudar y corregir lo desarrollado y (2) ver la complejidad de lo desarrollado, los problemas y si se termina a tiempo, para poder planificar las tareas de la siguiente fase.

**Fechas clave**

* 21/04/2022 Presentación del diseño del videojuego
* 19/05/2022 Presentación de primera versión del videojuego
* 03/06/2022 Publicación del videojuego completo

**Herramientas utilizadas Fases del desarrollo**

Discord

* Reuniones
* Salas de trabajo
* Mensajería

GitHub

* Sistema de control de versiones
* Fases y actividades
* Tablero de actividades

**Fase 1**

**Primer mapa // Carlos**

Crear un entorno inicial mínimo (poco más amplio de lo que pueda verse en la pantalla), con sprites randoms (dan igual, solo son de prueba, ya usaremos los buenos en el mapa bueno). Que contenga al personaje inicial (y la posición en la que debe de respawnear), uno o varios enemigos (y la posición en la que deben respawnear) y un entorno completo, con paredes y obstáculos, y tocar mínimamente la decoración. Todo esto tiene que ir por capas, es decir, todos los obstáculos que no se puedan atravesar, en capa de obstáculos; obstáculos que no se puedan atravesar pero que influyan al personaje en otra capa diferente (e incluso en varias según afecte al personaje) y así con todos los obstáculos, objetos y personajes.

El objetivo: Saber como se diseñan los mapas de cara a que los diseñadores gráficos sepan iniciar todo el proceso en la fase producción. Y que los programadores sepan como tratar los mapas correctamente antes de iniciar dicha fase. Además de adentrarnos en el mundillo para orientarnos en el diseño del videojuego.

**Mundo del videojuego // Fátima**

Desarrollar por escrito la historia principal en la que se va a desarrollar nuestro videojuego.

* Describir al personaje principal: sus características, ambiciones… porque es el protagonista y que pinta en la historia. Detallando (de forma secundaria) sus mecánicas y habilidades.  
  IDEM con los posibles acompañantes que tenga el protagonista, en caso de que existan; con los NPC; y con los enemigos (como mínimo tiene que haber 2 enemigos básicos y 1 jefe, aunque tampoco debemos de hacer muchos más tipos, ya que son en cierta medida innecesarios y aumentan la complejidad).
* Describir el mundo en el que se va a desarrollar y con ello dar una idea de los posibles mapas que vamos a crear.
* Desarrollar toda la trama del videojuego, con un inicio, un nudo y un desenlace. En la que se vean envueltos de forma correcta todos los personajes y entornos.
* Incorporar en la historia (personajes, mapas y trama) los Objetivos de Desarrollo Sostenibles de la ONU.

El objetivo: Tener una primera versión completa del diseño del juego. Para poder completar la fase de diseño de forma correcta y a tiempo. Y poder seguir desarrollando las siguientes fases por parte del resto del equipo de forma correcta.

Nota 1: Antes de iniciar esta tarea, contactar con el jefe de proyecto, para concertar una reunión. Se deben delimitar ciertos puntos que afecta directamente su desarrollo.

Nota 2: No hace falta enmarcar el diseño grafico del juego, aunque si delimitar. Y para nada, hay que buscar/crear los sprites.

**Coordinar Programadores fase 1// Nico**

Coordinar el desarrollo de cada una de las actividades de programación de esta fase (fase 1): Protagonista, objetos e inventario y mascotas.  
Ayudarles a desarrollar las funciones y transmitir a los otros programadores los avances. Deberá de impartir un cierto nivel de programación, para que todas las clases y funciones, tengan el mismo estilo y calidad.   
Además de asegurarse que se sieguen las buenas practicas de programación. Y que se desarrollan en tiempo y forma a lo estipulado.

Como mínimo, deberá concertar una reunión a lo largo de la fase con cada uno de los programadores para realizar las tareas ya descritas. Y que será independiente a las reuniones que concierte el jefe de proyecto.

Desarrollar el diagrama de clases de todo lo desarrollado. De la forma más adecuada y limpia posible.

**Protagonista // Nico**

Desarrollar la clase Protagonista. Esta tendrá muchísima funcionalidad y dependerán muchas otras clases de la misma. Pero por ahora solo centrarse en el correcto movimiento del personaje. En el laboratorio del día 24, hay un enlace a un codigo de ejemplo de como mover al personaje correctamente. Mirarlo y utilizarlo.  
Moverlo con las teclas A,S,D,W.

Tener en cuenta, que aunque sea el protagonista, no deja de ser un personaje más.

Obviamente hay que desarrollarla con buenas prácticas, TODO ORDENADO, SEPARADO Y DOCUMENTADO (comentar cada clase y función completamente, y dentro de cada función, comentar las líneas de codigo que se consideren necesarias).

El objetivo: Producción del videojuego. Crear la clase principal, con los métodos esenciales para poder desarrollar distintos códigos de otras clases y funcionalidades.

Nota 1: coordinarse con el resto de programadores para desarrollar las diferentes clases coherentemente.

Nota 2: Todo desde un entorno vacío, es decir, una ventana con el background y ya. Todos los sprites randoms, usar los que se quiera de la librería de Arcade Python. Cuando tengamos los buenos ya los intercambiaremos.

**Objetos e Inventario // Mario**

Desarrollar la clase objetos. Estos objetos se podrán coger por el personaje principal y se almacenarán en su inventario. Por lo que habrá que desarrollar también la clase inventario. Aunque esta de momento solo será para almacenar los objetos. Es decir, no hace falta desarrollar el tratamiento, uso y cualquier otra cosa que se pueda hacer con el inventario. Solo servirá para mostrar que hemos guardado algo (y que se vea claramente que algo hemos guardado).  
En cuanto a la clase Objetos, mencionar que se debe de ir implementando ya bien. Teniendo en cuanta que va a ver diferentes tipos de objetos, con diferentes características. Y que se deben de desarrollar al menos un par de objetos distintos, por ejemplo, las curas y la munición.

La clase Objeto debe de estar desarrollada lo más completa posible. Con la mayor cantidad de parámetros útiles posibles y la mayor cantidad de funciones útiles posibles. O dicho de otro modo, con la mayor cantidad de funcionalidad que se pueda llegar a conseguir para un objeto del juego (esto va de la mano con el trabajo de imaginación del desarrollador, contando con el equipo de Diseño y del Jefe de proyecto).  
Y obviamente con buenas prácticas, TODO ORDENADO, SEPARADO Y DOCUMENTADO (comentar cada clase y función completamente, y dentro de cada función, comentar las líneas de codigo que se consideren necesarias).

El objetivo: Producción del videojuego. Enmarcar mecánicas, para enmarcar el GDD.

Nota 1: coordinarse con el resto de programadores para desarrollar las diferentes clases coherentemente. Sobre todo con el programador jefe, que es el encargado de la clase Protagonista y los objetos e inventario, por decirlo de algún modo son del personaje protagonista.   
Usar el personaje y su movimiento (clase protagonista), pero sino está desarrollada, usar el código de ejemplo del lab, que es independiente a lo que aquí se debe desarrollar.

Nota 2: Todo desde un entorno vacío, es decir, una ventana con el background y ya. Todos los sprites randoms, usar los que se quiera de la librería de Arcade Python. Cuando tengamos los buenos ya los intercambiaremos.

**Mascotas // Daniel**

Desarrollar la clase mascotas. Esta mascota deberá de incorporar una inteligencia artificial que se encargue de seguir a nuestro Personaje Protagonista por el mapa. Que aunque suene chungo, una IA, no es más que un algoritmo (función) que según las coordenadas de nuestro protagonista y sus movimientos se mueva como corresponda.  
Esta clase tendrá más funcionalidad, pero por ahora solo hay que centrarse en este aspecto.

Obviamente hay que desarrollarla con buenas prácticas, TODO ORDENADO, SEPARADO Y DOCUMENTADO (comentar cada clase y función completamente, y dentro de cada función, comentar las líneas de codigo que se consideren necesarias).

El objetivo: Producción del videojuego. Enmarcar mecánicas, para enmarcar el GDD.

Nota 1: coordinarse con el resto de programadores para desarrollar las diferentes clases coherentemente. Sobre todo con el programador jefe, que es el encargado de la clase Protagonista y las mascotas, deben de seguir el movimiento del mismo.  
Usar el personaje y su movimiento (clase protagonista), pero sino está desarrollada, usar el código de ejemplo del lab, que es independiente a lo que aquí de debe desarrollar.

Nota 2: Todo desde un entorno vacío, es decir, una ventana con el background y ya. Todos los sprites randoms, usar los que se quiera de la librería de Arcade Python. Cuando tengamos los buenos ya los intercambiaremos.

**Testear fase 1 // Rubén**

Analizar el código desarrollado por lo programadores en las actividades de esta fase (fase 1): Protagonista, objetos e inventario y mascotas.  
Corrigiendo (comunicando cada uno de los cambios) y verificando todas estas clases y sus funciones.

Desarrollar las diferentes pruebas unitarias para validar el correcto funcionamiento de todos los métodos. En busca de que en situaciones esperadas y básicas, se ejecute correctamente. Y explorar en situaciones que se alejen un poco de lo que el usuario deba hacer o pueda hacer, en busca de comportamientos extraños del código.

El objetivo: asegurarse de que el videojuego es completo y robusto.

**Fase 2**

**Apoyo en “Mundo del videojuego 2” y “Arte del videojuego” // Carlos**

Ayudar a la diseñadora jefe en sus actividades de la fase 2 de desarrollo según delegue.  
Céntrate sobre todo en la del “Arte del videojuego” (léetela) ya que has trabajado con el editor de mapas y puede que tengas una perspectiva interesante.

Ponte en contacto con la diseñadora jefe.

**Arte del videojuego// Fátima**

Desarrollar por escrito el arte visual que va a incorporar nuestro videojuego. Describir el genero visual y los aspectos esperados que van a tener los personajes y entorno (ambiente, obstáculos, construcciones, etc).   
También se puede completar con una primera versión de sprites. (Centrarse sobre todo en el personaje principal, sus acompañantes y los enemigos.)

**Mundo del videojuego 2 // Fátima**

Con el feedback por parte del jefe de proyecto de lo entregado en la actividad anterior (“Mundo del videojuego”), completar la historia: Añadir, eliminar, modificar y COMPLETAR aún más la lo que ya se tiene.

**Coordinar Programadores fase 2 // Nico**

IDEM que “Coordinar Programadores fase 1” pero con todo lo desarrollado tanto en esta fase como en las anteriores.

**Implementar mapa // Nico**

Tenemos ya una primera prueba sobre los mapas. Implementarlo en Arcade Python.   
Tener el mapa operativo y que se pueda utilizar todo lo desarrollado hasta ahora.

Como entiendo que es semana santa, y tienes otras actividades, desde el principio esta tarea cuenta con dos partes (es decir, dos semanas). Así que con investigar como se haría y jugar un poco me sirve. Pero piensa que si lo haces todo ahora, menos tienes que hacer la siguiente y te puedes centrar un poco más con la de “Coordinas Programadores”.

**Mecánica de disparar // Mario**

Implementar la mecánica de disparar en nuestros personajes.

Para ello, crear la clase balas. E implementar un método en la clase Personaje (por ahora Protagonista) que trate a esta nueva clase. Y que devuelva lo que tenga que devolver a la clase principal para que se impriman las cosas.

**Mecánica de vida // Mario**

Implementar en la clase Personaje (por ahora Protagonista) un sistema de vida y un método que permita imprimir por pantalla la vida del personaje.

Como por ahora solo tenemos al protagonista, el método se enfocará en imprimir en una parte de la pantalla (a decidir por ti, aunque puede cambiar, según vayamos incorporando nuevas informaciones) la vida que tiene el personaje, con el diseño y forma que quieras.

De forma adicional, crea otro método (independiente al anterior) que imprima la vida, con el formato que quieras, encima del Sprite. Esto se implementará cuando hagamos a los enemigos y demás personajes. Pero así ya lo tenemos hecho.

**Testear fase 2 // Rubén**

IDEM que “Testear fase 1” pero con todo lo desarrollado tanto en esta fase como en las anteriores.

**Fase 3**

**Diseño de niveles // Fátima y Carlos**

Ya que tenemos los primeros borradores de la historia del juego y de su diseño. Realizar algunos diseños de los mapas. Para ello, usar cualquier herramienta (o herramientas). Puede ser con un Paint, o simplemente una hoja de papel… lo que queráis. En la que me pongáis un primer boceto del primer nivel (en caso de que hagamos un juego por niveles) o de la primera zona (en caso de que hagamos un juego de mundo abierto). Esto último depende, de lo que decidamos todos, sobre todo vosotros, que sois los diseñadores. Y de como vayamos diseñando el mapa y de como lo vayamos implementando. Pero ahora, como el mapa que hagáis va a ser mínimo, no debería de ser una preocupación grave.  
Por cierto, no hace falta dibujar bien sabes, con que cada cosa se entienda, o en caso de que no se entienda esté comentada que es, más que suficiente.

Esto es para que vayamos haciéndonos una idea de como vamos a hacer los mapas y que elementos vamos a utilizar. Además de tener una idea general del mapa, y ya cuando estemos seguros y entre todos hayamos dado el visto bueno, implementarlo de verdad. Para reducir los cambios en el mapa, que se tardará más que en un simple boceto (aunque no estaremos salvado al 100% de tener que modificar algo).

Esa es la parte uno. La parte dos, es completar el borrador del diseño del juego, con lo que se os vaya ocurriendo mientras trabajáis. Tanto por escrito, como buscando los sprites necesarios.

Y la parte tres. Implementar un poco del mapa, en la herramienta Tiled Map Editor. Cuanto más y más completo, mejor. Obvio, antes de que implementéis, enseñadme el boceto, o parte del boceto, aunque no es necesario, es solo para ahorraros tiempo en caso de que algo no me guste o no sea lo que hay que hacer y yo pueda avisaros.   
No tenemos todavía el feedback por parte de los programadores del primer mapa de prueba desarrollado en la actividad “Primer mapa”. Así que no sé si tenéis que cambiar algo de lo allí desarrollado. Pero no debería ser problema, ya que para cuando querías implementarlo, debería de estar la respuesta… aunque sea mínima. Además, de que creo que estaba bien y sino, poco había que cambiar.

Recuerda, Fátima, que eres la diseñadora jefe, así que organiza un poco todo el trabajo.

**Coordinar Programadores fase 3 // Nico**

IDEM que “Coordinar Programadores fase 1” pero con todo lo desarrollado tanto en esta fase como en las anteriores.

**Implementar mapa 2 // Nico**

Terminar de implementar el mapa de prueba desarrollado en la tarea “Primer mapa”. Tiene que estar al completo.   
Debes de comunicarte con el equipo de diseñadores para comentarles los hallazgos realizados, tanto en esta actividad como en la actividad de la fase uno. Básicamente porque van a ponerse a desarrollar el mapa en esta fase y necesitan saber si lo desarrollado en el mapa de prueba está bien.

**Mascotas 2 // Daniel**

Terminar de desarrollar el funcionamiento de la Inteligencia artificial de la mascota.

Implementarle a la mascota un método, que cuando se pulse una tecla del teclado (pongamos por ejemplo la Q), deje de seguir al protagonista y tenga la posibilidad de moverse con las teclas de dirección (A,S,D,W).  
Es decir, el protagonista se queda quieto y es a la mascota a la que controlamos.  
Para ello, echar un vistazo a como controlamos el movimiento del protagonista, ya que es exactamente igual.

**Testear fase 3 // Rubén**

IDEM que “Testear fase 1” pero con todo lo desarrollado tanto en esta fase como en las anteriores.

**Fase 4**

**Diseño de niveles 2 // Fátima y Carlos**

Seguir con el diseño de los mapas:

* Modificar y añadir lo que sea necesario en los bocetos del mapa. Con lo que se proponga en la reunión de esta semana, y con cualquier cosa que os pueda decir yo (o cualquier compañero) al revisarlo con detenimiento.
* Terminar de bocetar en caso de que no se haya terminado

Ir implementando el mapa en el editor de mapas. Me gustaría tener algo del primer nivel, en plan, lo más completo y vistoso posible.

Reunir los sprites:

* Ahora que ya tenemos el boceto y un primer borrador del arte del juego. Quiero que me subáis al GitHub (en las carpetas adecuadas como os dije) todos los sprites que se necesiten para hacer los mapas. Y con esto, indirectamente, es, buscar los sprites que vais a ir necesitando para hacerlo. Obviamente la lista no se va a cerrar con esta tarea, seguramente se seguirán buscando según se vayan necesitando y ocurriendo, pero para ya ir estableciendo los sprites.
* Ordenarme y nombrarme todos los sprites que se suban, como os dije. Que se entiendan que son simplemente con el nombre.  
  Quizá, enumerarlos, como dije, no, por el tema de que seguramente se añadan y se lie. Pero ordenarlos por carpetas y hacer que sean poquitos en cada lado y del mismo tipo si, ya que así se entienden mejor y están ordenados, que es lo mismo que numerarlos.  
  Sabéis, tipo, carpetas como: Caminos (y guardar todos los tipos de caminos), terrenos (y guardar el césped, el agua, etc), edificios, etc.
* Ya que os vais a poner con el tema de los sprites del mapa, ponedme los sprites necesarios para animar a los personajes. Al menos el del protagonista, la animación de andar y de atacar.

Fatima, eres la diseñadora jefa, organiza todo esto como consideres más oportuno.

**Coordinar Programadores fase 4 // Nico**

IDEM que “Coordinar Programadores fase 1” pero con todo lo desarrollado tanto en esta fase como en las anteriores.

**Menús // Nico**

Implementarme un sistema de gestión del juego. Que se inicie en una pantalla de inicio, y que dándole a un botón o una tecla o lo que sea, pase al juego. Y que cuando se pulse otra tecla (ya que todavía no tenemos la muerte del personaje) se pase al menú de fin de partida. En este último menú, que haya dos opciones, volver al menú de inicio, o al juego.   
No sé si nos dará tiempo a implementar puntos de guardado, porque cuando le demos a reiniciar en este último menú, lo suyo sería que se volviese ahí. Pero de momento, como no lo tenemos, con iniciarme el juego de nuevo (setup) suficiente, luego se cambia en cero coma, en caso de que lo hagamos claro. Lo digo por si te lo preguntabas.

**Objetos e inventario 2 // Mario**

Quiero que me implementes las armas del personaje. Son 4, la espada de fuego, la espada de aire, la espada de agua y la espada de electricidad. (Realmente es la misma espada, pero que cambia de habilidad con magia, modo varita de mago y atacas con lo que invoques). Y con ello la munición.  
  
Y ahora hablando un poco en cristiano, es terminar lo de disparar, como te dije. Que mirase que arma tiene, y si tiene munición de esa arma, ya que es finita, pues que dispare un tipo de bala en concreto. Con un Sprite en concreto, un daño, una velocidad y demás.?  
No sé si tienes que implementar todo como clases y demás, eso lo juzgas tú. Incluso, si tienes que cambiar algo de la clase Bullet. Todo esto, lo último digo, lo menciono, porque se me está empezando a ir la cabeza para hacer alguna cosita chula con los elementos… y por eso digo, que puede que la clase Bullet sufra cambios y se crean alguna que otra clase complementaria.  
Pero bueno, no te lio, eso lo hablamos cuando quedemos y avances con esto primero, solo te lo mencionaba para abrirte un poco la mente.

**Mascotas 3 // Daniel**

Terminar de desarrollar todo el control de la mascota que no esté acabado de las otras dos tareas anteriores relacionadas con esta.

E implementarme en ese mismo método, que la mascota solo siga al protagonista si tiene visión del mismo. También hay un ejemplo en el apartado de los ejemplos que te dejé en la actividad anterior.  
  
Para concretarte un poco más, esto último, no va para la mascota en si. Es para el tema de los enemigos, que cuando vean al protagonista vayan a por él. Pero como no tenemos los enemigos implementados, hazlo así. Ya que así es súper fácil para ti. Y luego no cuesta absolutamente nada separar el código, ya que en cierta medida va a ser el mismo.

**Testear fase 4 // Rubén**

IDEM que “Testear fase 1” pero con todo lo desarrollado tanto en esta fase como en las anteriores.