

Especificación de requerimientos de software

Guasacaca Studios

Samantha Covarrubias Figueroa

Cristina Alessandra González

Valeria Martínez Martínez

Mau Tumalán

Fernanda Colomo

Historias de usuario

Cliente	Historia	Requerimientos
General	<ul style="list-style-type: none">Como cliente busco que se desarrolle un videojuego RPG	<ul style="list-style-type: none">Elaborar diagrama UML.Elaborar casos de uso.Elaborar diagramas de flujo de actividad.
	<ul style="list-style-type: none">Como cliente busco que se expliquen los controles y jugabilidad del juego	<ul style="list-style-type: none">Elaborar manual de instrucciones en el juegoImplementar manual de instrucciones en el juego
Socio Formador	<ul style="list-style-type: none">Como cliente busco que el jugador encuentre una experiencia entretenida e innovadora.	<ul style="list-style-type: none">Crear sprites del videojuegoBuscar e implementar música para el videojuegoBuscar e implementar efectos de sonido para el videojuego
	<ul style="list-style-type: none">Como cliente busco que exista una historia interesante.	<ul style="list-style-type: none">Creación de NPCsImplementación de NPCs.Creación de cutscenes.Implementación de cutscenes.
	<ul style="list-style-type: none">Como cliente busco que el videojuego implemente distintas mecánicas RPG	<ul style="list-style-type: none">Implementación de Skilltree para personaje principal (chef)Elementos de personalización visual de personaje principal (chef)
	<ul style="list-style-type: none">Como cliente busco que el usuario encuentre un deseo de mejorar su desempeño, manteniendo un equilibrio entre la recompensa de ganar y la frustración de perder.	<ul style="list-style-type: none">Implementación de dificultadesImplementación de clases en los aliados para permitir diferentes acercamientos a las estrategias realizadas dentro del nivel.Implementación de puntaje de recompensas para SkillTree
	<ul style="list-style-type: none">Como cliente quiero que existan elementos dentro del juego que reten al usuario a buscar la mejor	<ul style="list-style-type: none">Implementación de clases en los enemigos para permitir diferentes acercamientos a las estrategias realizadas dentro

	<p>estrategia y sea retado para pasar el nivel.</p>	<p>del nivel.</p> <ul style="list-style-type: none"> Implementación de dificultades en los diferentes enemigos
Videojuego	<ul style="list-style-type: none"> Como cliente quiero visualizar la evolución del protagonista, personajes aliados y enemigos durante el juego. 	<ul style="list-style-type: none"> Diseño de evoluciones del personaje acorde a su nivel en el SkillTree Diseño de evoluciones de aliados acorde a las mejoras desbloqueadas en el SkillTree Diseño de evoluciones de enemigos acorde a la dificultad. Diseño de evoluciones de enemigos acorde a progreso en el juego.
	<ul style="list-style-type: none"> Como cliente quiero que el videojuego implemente mecánicas de estrategia en tiempo real 	<ul style="list-style-type: none"> Implementación de selección de personajes aliados y movimiento de estos Implementación de aparición de enemigos y movimiento de estos Implementación de time-delay (que el tiempo vaya más lento mientras se selecciona la posición del personaje) Implementación de ataque entre personajes aliados y enemigos Implementación de ataque de enemigos a la base (carrito de comida)
	<ul style="list-style-type: none"> Como cliente quiero que el usuario tenga que defender e interactuar con una base 	<ul style="list-style-type: none"> Implementación de carrito de comida Implementación de menú de interacción para creación de comida Implementación de cooldown de chef durante la creación de comida
	<ul style="list-style-type: none"> Como cliente quiero que el videojuego tenga la menor cantidad de periféricos (controles) y que tengan una razón de ser. 	<ul style="list-style-type: none"> Implementación de uso de periféricos sencillos para la jugabilidad del juego.
	<ul style="list-style-type: none"> Como cliente quiero que el juego sea rejugable. 	<ul style="list-style-type: none"> Implementación de distintas nacionalidades. Implementación de bifurcaciones en el SkillTree.

Base de datos	<ul style="list-style-type: none"> • Como cliente quiero que la información del jugador se guarde en un repositorio de información. 	<ul style="list-style-type: none"> • Implementar la base de datos con la herramienta de MYSQL.
	<ul style="list-style-type: none"> • Como cliente quiero que la información se pueda insertar, actualizar y eliminar en la base de datos. 	<ul style="list-style-type: none"> • Implementación del modelo CRUD en la base de datos.
	<ul style="list-style-type: none"> • Como cliente quiero que se pueda enviar y recibir información en la base de datos, acerca del videojuego en la página web. 	<ul style="list-style-type: none"> • Permitir la comunicación entre la BD, la página y el juego mediante las API's correspondientes
	<ul style="list-style-type: none"> • Como cliente quiero que se pueda acceder a la base de datos fácilmente. 	<ul style="list-style-type: none"> • Conectar la base de datos a la página web y el juego
	<ul style="list-style-type: none"> • Como cliente quiero que se pueda almacenar información de checkpoints, inventario, ítems, escenarios, vida ataque, velocidad, mana en tablas separadas. 	<ul style="list-style-type: none"> • Crear el diagrama entidad relación para diseñar el modelo. • Identificar las distintas tablas a implementar.
Web	<ul style="list-style-type: none"> • Cómo cliente quiero que la página web cuente con un ambiente que sea atractivo visualmente y que sea interactivo. 	<ul style="list-style-type: none"> • Diseñar distintas opciones de posibles prototipos de páginas web • Implementar el diseño front end que el cliente elija • Asegurarse de tener una página dinámica (interactuable)
	<ul style="list-style-type: none"> • Cómo cliente quiero que exista un apartado para iniciar sesión en el juego por medio de la página web. 	<ul style="list-style-type: none"> • Añadir una nueva sección dónde se pueda cargar el unity (uso de API) • Mediante el API comunicar la base de datos con el juego de Unity para poder hacer partidas distintas
	<ul style="list-style-type: none"> • Como cliente quiero que el juego se pueda jugar desde la página web 	<ul style="list-style-type: none"> • Crear el apartado para el juego de Unity y llamarlo desde el código en JS.
	<ul style="list-style-type: none"> • Como cliente quiero que en la página web se muestren estadísticas y gráficas con los datos 	<ul style="list-style-type: none"> • Implementación de estadísticas • Crear la comunicación de la BD y el código de la página

	guardados en el juego.	para poder tener resultados visuales de las estadísticas (API)
	<ul style="list-style-type: none"> • Como cliente quiero que exista un apartado con un manual de usuario en la página. 	<ul style="list-style-type: none"> • Crear el apartado • Implementarlo como parte del front end de la página

“Product backlog” (Numeración de requerimientos):

1. Elaborar diagrama UML.
2. Elaborar casos de uso.
3. Elaborar diagramas de flujo de actividad.
4. Elaborar manual de instrucciones en el juego
5. Implementar manual de instrucciones en el juego
6. Crear sprites del videojuego
7. Buscar e implementar música para el videojuego
8. Buscar e implementar efectos de sonido para el videojuego
9. Creación de NPCs
10. Implementación de NPCs.
11. Creación de cutscenes.
12. Implementación de cutscenes.
13. Implementación de Skilltree para personaje principal (chef)
14. Elementos de personalización visual de personaje principal (chef)
15. Implementación de dificultades
16. Implementación de clases en los aliados para permitir diferentes acercamientos a las estrategias realizadas dentro del nivel.
17. Implementación de puntaje de recompensas para SkillTree
18. Implementación de clases en los enemigos para permitir diferentes acercamientos a las estrategias realizadas dentro del nivel.
19. Implementación de dificultades en los diferentes enemigos
20. Diseño de evoluciones del personaje acorde a su nivel en el SkillTree
21. Diseño de evoluciones de aliados acorde a las mejoras desbloqueadas en el SkillTree
22. Diseño de evoluciones de enemigos acorde a la dificultad.
23. Diseño de evoluciones de enemigos acorde a progreso en el juego.
24. Implementación de selección de personajes aliados y movimiento de estos
25. Implementación de aparición de enemigos y movimiento de estos
26. Implementación de time-delay (que el tiempo vaya más lento mientras se selecciona la posición del personaje)
27. Implementación de ataque entre personajes aliados y enemigos
28. Implementación de ataque de enemigos a la base (carrito de comida)
29. Implementación de carrito de comida
30. Implementación de menú de interacción para creación de comida
31. Implementación de cooldown de chef durante la creación de comida
32. Implementación de uso de periféricos sencillos para la jugabilidad del juego.
33. Implementación de distintas nacionalidades.
34. Implementación de bifurcaciones en el SkillTree.
35. Implementar la base de datos con la herramienta de MYSQL.
36. Implementación del modelo CRUD en la base de datos.
37. Permitir la comunicación entre la BD, la página y el juego mediante las API's

correspondientes

38. Conectar la base de datos a la página web y el juego
39. Crear el diagrama entidad relación para diseñar el modelo.
40. Identificar las distintas tablas a implementar.
41. Diseñar distintas opciones de posibles prototipos de páginas web
42. Implementar el diseño front end que el cliente elija
43. Asegurarse de tener una página dinámica (interactuable)
44. Añadir una nueva sección dónde se pueda cargar el unity (uso de API)
45. Mediante el API comunicar la base de datos con el juego de Unity para poder hacer partidas distintas
46. Crear el apartado para el juego de Unity y llamarlo desde el código en JS.
47. Implementación de estadísticas
48. Crear la comunicación de la BD y el código de la página para poder tener resultados visuales de las estadísticas (API)
49. Crear el apartado
50. Implementarlo como parte del front end de la página

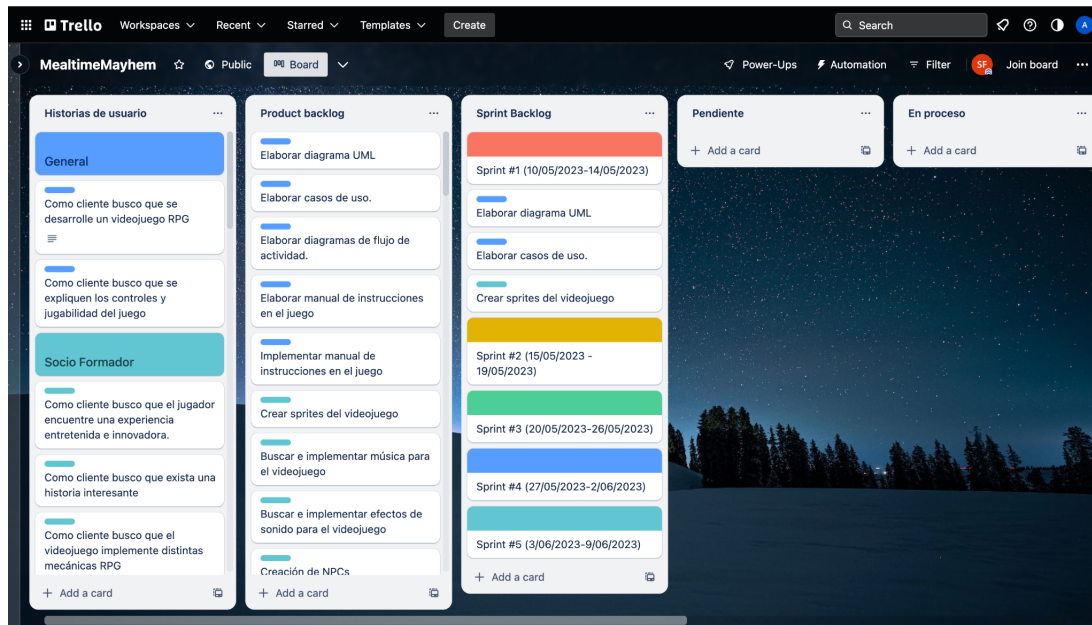
“Sprint backlog” con las tareas a ser realizadas en el primer sprint:

<https://trello.com/invite/b/Dtdwlh8r/ATTI4391f493d2473d255296fd70497099a27B890CAC/mealtimemayhem>

Sprint #1 (10/05/2023 - 14/05/2023)

- Elaborar diagrama UML
- Elaborar casos de uso
- Crear sprites del videojuego

Captura de pantalla del tablero virtual



Diagramas de casos de uso:

📎 Diagrama UML.png

Descripción de cada uno de los casos de uso del diagrama UML:

Caso de uso	Elaborar diagrama UML
Requerimientos relacionados	N/A
Meta en contexto	Elaborar un diagrama UML que tenga requerimientos pertinentes para el correcto desarrollo del videojuego.
Precondiciones	Definir las mecánicas y desarrollo del videojuego para encontrar todos los posibles requisitos para su desarrollo.
Condición para un fin exitoso	Diagrama completo, con todos los requerimientos necesarios para el desarrollo exitoso del videojuego.
Condición para un fin fallido	Diagrama modificado, con cambios por hacer para incluir todos los requerimientos no contemplados.
Actores primarios	Desarrolladores
Actores secundarios	Administrador del videojuego
Trigger	Los desarrolladores deben realizar y desambiguar todos los requerimientos acorde a las necesidades del administrador del videojuego.
Flujo Principal	
Pasos	Acciones
1	Definir completamente las mecánicas y demás características del videojuego.
2	Definir completamente los requerimientos que serán necesarios para su realización.
3	Desarrollar el diagrama UML que incluya todos los requerimientos previamente definidos.
Extensiones	N/A

Caso de uso	Elaborar casos de uso
Requerimientos relacionados	1
Meta en contexto	Diagramas con todos los requerimientos claramente definidos.
Precondiciones	Haber definido claramente los requerimientos necesarios en cada área y sus respectivos actores.
Condición para un fin exitoso	Diagrama con requerimientos definidos acorde a su área, con el fin de evitar modificaciones a lo largo del proyecto.
Condición para un fin fallido	Diagramas que requieran de modificación, debido a falta o mala representación de información como áreas o actores correspondientes.
Actores primarios	Desarrolladores
Actores secundarios	Administradores
Trigger	Los desarrolladores deben definir los requerimientos acorde a las necesidades de los administradores.
Flujo Principal	
Pasos	Acciones
1	Escuchar activamente las necesidades de cada administrador.
2	Definir completamente los requerimientos que serán necesarios en cada área.
3	Desarrollar diagramas de uso para cada área y sus respectivos actores.
Extensiones	N/A

Caso de uso	Elaborar diagramas de flujo de actividad
Requerimientos relacionados	1, 2
Meta en contexto	Los diagramas de flujo de actividad muestran gráficamente el flujo de desarrollo del videojuego y las conexiones entre las administraciones.
Precondiciones	Contar con los requerimientos claramente definidos, diagramas de caso de uso UML.
Condición para un fin exitoso	Se muestra correctamente en los diagramas de flujo, la relación de los requerimientos con las distintas administraciones del videojuegos que cumplen correctamente con todas las necesidades
Condición para un fin fallido	Se necesitan realizar modificaciones a los diagramas ya que no se muestra correctamente la relación entre las administraciones
Actores primarios	Desarrolladores
Actores secundarios	Administradores
Trigger	Los desarrolladores deberán escuchar las necesidades de los administradores, definir los requerimientos adecuados y relacionarlos entre sí con un diagrama de flujo de actividades
Flujo Principal	
Pasos	Acciones
1	Escuchar activa a las necesidades de cada administración
2	Definición de los requerimientos con sus respectivos diagramas de casos de uso UML
3	Desarrollar el diagrama de flujo con todas las actividades
Extensiones	N/A

Caso de uso	Elaborar manual de instrucciones en el juego
Requerimientos relacionados	1, 2, 3
Meta en contexto	Crear un manual que permita al jugador tener el conocimiento necesario para navegar por el juego.
Precondiciones	Establecer y definir claramente las mecánicas, su funcionamiento e interacciones, así como los personajes para elaborar un manual detallado.
Condición para un fin exitoso	Elaboración de un manual detallado, con el funcionamiento del juego que permita al usuario desarrollar un entendimiento lo suficientemente profundo del mismo para que su navegación sea agradable.
Condición para un fin fallido	No es comprensible el funcionamiento del juego debido a una redacción inapropiada de las instrucciones.
Actores primarios	Desarrolladores
Actores secundarios	Usuario
Trigger	Estando en la página web, el usuario hace clic en la sección de instrucciones, causando que estas se desplieguen.
Flujo Principal	
Pasos	Acciones
1	Definir las mecánicas, interacciones y funcionamiento.
2	Redactar de manera detallada cada instrucción.
Extensiones	N/A

Caso de uso	Implementar manual de instrucciones en el juego
Requerimientos relacionados	4
Meta en contexto	Elaborar un manual sobre cómo jugar el videojuego, con definiciones y ejemplificaciones de los ítems, cómo se gana, cómo se pierde, etc.
Precondiciones	Definición adecuada de las técnicas, estadísticas y elementos del juego. Además de la implementación en Unity que define el movimiento del personaje.
Condición para un fin exitoso	Se explica adecuadamente en el manual la dinámica de juego y cómo se gana, además del uso de los ítems y las teclas o controles físicos.
Condición para un fin fallido	No se explican las reglas adecuadamente en el manual para lograr una partida exitosa.
Actores primarios	Desarrolladores
Actores secundarios	Usuario
Trigger	El usuario hace uso de las instrucciones en el juego para completar el juego; los desarrolladores se encargan de incluir las instrucciones e información adecuadas en el manual.
Flujo Principal	
Pasos	Acciones
1	Definir totalmente el juego y sus mecánicas.
2	Realizar la implementación en Unity para la determinación de las teclas/controles del juego.
3	Elaborar un manual con todas las instrucciones necesarias.
Extensiones	N/A

Caso de uso	Crear sprites del videojuego
Requerimientos relacionados	N/A
Meta en contexto	Elaborar los sprites que serán parte del videojuego, pantallas, personajes con movimiento, ítems, etc.
Precondiciones	Haber definido una apariencia para los personajes, mecánicas, obstáculos y ambiente de juego.
Condición para un fin exitoso	Los sprites pueden ser usados adecuadamente en el resto del desarrollo del videojuego.
Condición para un fin fallido	Se requieren modificaciones a los sprites para su correcta implementación en Unity y en el videojuego en general.
Actores primarios	Desarrolladores
Actores secundarios	Administrador del videojuego
Trigger	El administrador de videojuego define qué tipo de juego busca. Los desarrolladores crean a partir de esto un boceto para los sprites del videojuego.
Flujo Principal	
Pasos	Acciones
1	Definir completamente el videojuego, sus mecánicas, ítems y apariencia general.
2	Corroborar la congruencia entre las necesidades del administrador y lo desarrollado por el equipo.
3	Crear los sprites para el videojuego.
Extensiones	N/A

Caso de uso	Buscar e implementar música para el videojuego
Requerimientos relacionados	N/A
Meta en contexto	Buscar e implementar música libre de derechos en congruencia con la historia y el ambiente del juego.
Precondiciones	Definir las mecánicas y entornos del videojuego.
Condición para un fin exitoso	Existe una congruencia entre las mecánicas y entornos del juego con las piezas musicales que se ocupan.
Condición para un fin fallido	El formato de las piezas no es el apropiado o no existe una relación entre estas y el ambiente del juego.
Actores primarios	Desarrolladores
Actores secundarios	Usuario
Trigger	Diferentes cambios de escena son reconocidos por el usuario gracias a la música. Los desarrolladores hacen esto posible por medio de la búsqueda e implementación de esta.
Flujo Principal	
Pasos	Acciones
1	Definir completamente las mecánicas y entornos del videojuego.
2	Clasificar las mecánicas y entornos de acuerdo a su rol en el juego.
3	Buscar e implementar la música que tiene congruencia con las mecánicas y entornos previamente definidos.
Extensiones	N/A

Caso de uso	Buscar e implementar efectos de sonido para el videojuego
Requerimientos relacionados	N/A
Meta en contexto	Buscar e implementar sonidos en formato MIDI en congruencia con las acciones y movimientos del juego.
Precondiciones	Definir las mecánicas, movimientos y entornos del videojuego.
Condición para un fin exitoso	Existe una congruencia entre las acciones y movimientos del juego con los MIDI que se ocupan.
Condición para un fin fallido	El formato de los MIDI no es el apropiado o no existe una relación entre estos y las mecánicas del juego.
Actores primarios	Desarrolladores
Actores secundarios	Usuario
Trigger	Diferentes acciones son reconocidas por el usuario gracias a los MIDI. Los desarrolladores hacen esto posible por medio de la búsqueda e implementación de los mismos.
Flujo Principal	
Pasos	Acciones
1	Definir completamente las mecánicas y movimientos del videojuego.
2	Clasificar los movimientos y mecánicas de acuerdo a su representación/rol en el juego.
3	Buscar e implementar los sonidos MIDI que representen los movimientos y mecánicas previamente definidos.
Extensiones	N/A

Caso de uso	Creación de NPCs
Requerimientos relacionados	1
Meta en contexto	Crear a los personajes NPC que ayudan al desarrollo de la historia y a la navegación dentro del juego.
Precondiciones	Crear una historia con trasfondo para el juego y establecer las instrucciones.
Condición para un fin exitoso	Crear los NPCs que permitan el desarrollo de la historia y ayuden al usuario principal.
Condición para un fin fallido	Crear NPCs que no muestren información clara o que no tengan un uso adecuado.
Actores primarios	Desarrolladores
Actores secundarios	Usuario
Trigger	El usuario llega a un punto del juego donde se encuentra a un NPC, el cual debe mostrar el mensaje o la instrucción correspondiente.
Flujo Principal	
Pasos	Acciones
1	Definir mensajes/instrucciones a ser mostradas.
2	Designar NPCs congruentes con el juego y el mensaje que se quiere dar.
3	Diseñar y crear los NPCs.
Extensiones	N/A

Caso de uso	Implementación de NPCs
Requerimientos relacionados	9
Meta en contexto	Encontrar un rol adecuado para los personajes NPCs que ayuden a la experiencia de juego del usuario.
Precondiciones	Definir totalmente las mecánicas y desarrollo del videojuego.
Condición para un fin exitoso	Los NPCs tienen un papel relevante y real dentro de la historia además de que interactúan con el jugador principal.
Condición para un fin fallido	Los NPCs no tienen un rol relevante, no muestran información adecuada o no son implementados.
Actores primarios	Desarrolladores
Actores secundarios	Usuario
Trigger	Los desarrolladores deben realizar NPCs que al interactuar con el usuario den información relevante y sean parte de la historia del videojuego.
Flujo Principal	
Pasos	Acciones
1	Definir completamente el videojuego y las interacciones NPCs - usuario.
2	Producir NPCs relevantes a la historia y desarrollo del videojuego.
3	Ensamblar el videojuego considerando el rol de los NPCs en la historia y desarrollo.
Extensiones	N/A

Caso de uso	Creación de cutscenes
Requerimientos relacionados	N/A
Meta en contexto	Implementar escenas que sirvan para añadir coherencia al desarrollo de la historia.
Precondiciones	Definir totalmente la historia del videojuego, determinando a los personajes adecuados.
Condición para un fin exitoso	Las cutscenes tienen un papel importante dentro de la historia y ayudan al jugador a sumergirse más dentro del juego.
Condición para un fin fallido	Las cutscenes no tienen sentido dentro de la historia.
Actores primarios	Desarrolladores
Actores secundarios	Usuario
Trigger	Al terminar un nivel el usuario es presentado con una cutscene que hila la historia con el siguiente nivel. Los desarrolladores diseñan una cutscene congruente con la historia.
Flujo Principal	
Pasos	Acciones
1	Definir completamente la historia del juego.
2	Diseñar las cutscenes relevantes.
3	Crear las cutscenes por medio de un programa de diseño o animación.
Extensiones	N/A

Caso de uso	Implementación de cutscenes
Requerimientos relacionados	N/A
Meta en contexto	Implementar escenas que sirvan para añadir coherencia al desarrollo de la historia.
Precondiciones	Definir totalmente la historia del videojuego, determinando a los personajes adecuados.
Condición para un fin exitoso	Las cutscenes tienen un papel importante dentro de la historia y ayudan al jugador a sumergirse más dentro del juego.
Condición para un fin fallido	Las cutscenes son implementadas en puntos donde no son relevantes.
Actores primarios	Desarrolladores
Actores secundarios	Usuario
Trigger	Al terminar un nivel el usuario es presentado con una cutscene que hila la historia con el siguiente nivel. Los desarrolladores diseñan una cutscene congruente con la historia.
Flujo Principal	
Pasos	Acciones
1	Definir completamente la historia del juego.
2	Diseñar las cutscenes relevantes.
3	Unir las cutscenes al desarrollo del juego en la sección de la historia donde se requiere de estas.
Extensiones	N/A

Caso de uso	Implementación de Skilltree para personaje principal (chef)
Requerimientos relacionados	N/A
Meta en contexto	Implementar sistema de mejoras para el personaje principal.
Precondiciones	Definir las habilidades que tendrá el personaje principal.
Condición para un fin exitoso	El personaje principal tiene un cambio en cuanto a habilidades dentro del juego y en relación a la mecánicas de este gracias al uso del skilltree.
Condición para un fin fallido	El personaje principal no presenta algún cambio o cambio positivo dentro del juego después de hacer uso del skilltree.
Actores primarios	Desarrolladores
Actores secundarios	Usuario
Trigger	Los desarrolladores diseñan un skilltree con opciones pertinentes al desarrollo del juego, del cual el usuario puede hacer uso al comienzo de este.
Flujo Principal	
Pasos	Acciones
1	Definir completamente las mejoras pertinentes al personaje principal dentro del juego.
2	Implementar las mejoras del skilltree en Unity.
3	Comprobar el funcionamiento adecuado de las mejoras planteadas.
Extensiones	N/A

Caso de uso	Elementos de personalización visual de personaje principal (chef)
Requerimientos relacionados	N/A
Meta en contexto	Lograr un nivel de personalización del personaje principal característico de un RPG.
Precondiciones	Definir el personaje principal y sus posibles apariencias.
Condición para un fin exitoso	El personaje principal es personalizable con consistencia a lo largo del desarrollo del juego.
Condición para un fin fallido	El personaje principal no es personalizable o no mantiene un nivel de personalización constante a lo largo del juego.
Actores primarios	Desarrolladores
Actores secundarios	Usuario
Trigger	Los desarrolladores deben realizar un sistema de personalización del personaje principal a medida que el usuario vaya avanzando en el juego.
Flujo Principal	
Pasos	Acciones
1	Definir al personaje principal y los aspectos que serán personalizables de este.
2	Crear visualmente las personalizaciones pertinentes.
3	Implementar en Unity los componentes visuales junto con los aspectos técnicos que permitan hacer uso de estos componentes.
Extensiones	N/A

Caso de uso	Implementación de dificultades
Requerimientos relacionados	N/A
Meta en contexto	Lograr un cambio de dificultad por cada horda y nivel que el jugador supera.
Precondiciones	Definir totalmente las mecánicas y desarrollo del videojuego.
Condición para un fin exitoso	El usuario se encuentra con un nivel de dificultad creciente conforme supera ciertos desafíos durante su experiencia al jugar.
Condición para un fin fallido	El usuario se encuentra con un juego de dificultad lineal, sea muy fácil o muy difícil, lo que afecta su experiencia de juego.
Actores primarios	Desarrolladores
Actores secundarios	Usuario
Trigger	Al superar cada horda o nivel el usuario enfrenta una dificultad mayor para superar los desafíos que se le presentan, la cual es designada por los desarrolladores.
Flujo Principal	
Pasos	Acciones
1	Diseñar la estructura del juego, principalmente los niveles y hordas por niveles.
2	Determinar cómo será representada la dificultad a incrementar por horda y nivel.
3	Implementar los cambios previamente determinados en dificultad al juego en Unity.
Extensiones	N/A

Caso de uso	Implementación de clases en los aliados para permitir diferentes acercamientos a las estrategias realizadas dentro del nivel
Requerimientos relacionados	N/A
Meta en contexto	Crear las clases correspondientes a los personajes que acompañan al personaje principal con sus distintas habilidades
Precondiciones	Definir totalmente las mecánicas y desarrollo del videojuego.
Condición para un fin exitoso	Los aliados (acompañantes del personaje principal) tienen diferentes habilidades que ayudan al personaje principal a combatir a los enemigos dentro del juego
Condición para un fin fallido	Los aliados no cuentan con habilidades variadas o su funcionamiento no es el correcto.
Actores primarios	Desarrolladores
Actores secundarios	Usuario
Trigger	Los desarrolladores deben realizar aliados que al interactuar con el juego, se genere en el usuario un sentido de estrategia dependiendo de las habilidades de los aliados elegidos
Flujo Principal	
Pasos	Acciones
1	Definir completamente el las habilidades de cada tipo de aliado
2	Establecer la forma en la que se mostrarán las habilidades de los aliados
3	Implementar las habilidades de cada aliado con los aspectos técnicos correspondientes en Unity
Extensiones	N/A

Caso de uso	Implementación de puntaje de recompensas para SkillTree
Requerimientos relacionados	N/A
Meta en contexto	Lograr un sistema de puntaje el cuál sirva para poder mejorar al personaje principal por medio del SkillTree
Precondiciones	Definir totalmente las mecánicas de puntuación y canje de puntos en el juego
Condición para un fin exitoso	Las mejoras del SkillTree se canjean exitosamente por medio de los puntos obtenidos a lo largo del juego.
Condición para un fin fallido	Las mejoras no se pueden canjear o implementar de forma adecuada en el juego.
Actores primarios	Desarrolladores
Actores secundarios	Usuario
Trigger	Los desarrolladores deben realizar un método en el juego donde el jugador pueda por medio de los puntos obtenidos en el juego, canjearlos para mejorar con el SkillTree a su personaje
Flujo Principal	
Pasos	Acciones
1	Definir completamente las mejoras disponibles
2	Establecer un sistema de puntuación dentro del juego
3	Implementar el método de canje de puntos para desbloquear las mejoras previamente establecidas
Extensiones	N/A

Caso de uso	Implementación de clases en los enemigos para permitir diferentes acercamientos a las estrategias realizadas dentro del nivel
Requerimientos relacionados	N/A
Meta en contexto	Lograr un sistema de ataque por parte de los enemigos que permita derrotar a estos estratégicamente
Precondiciones	Definir totalmente las mecánicas y desarrollo del videojuego.
Condición para un fin exitoso	Los enemigos tienen un esquema de ataque estructurado que permite al jugador crear estrategias para derrotarlos
Condición para un fin fallido	Los enemigos no contienen ataques estructurados por lo que es muy difícil implementar una estrategia favorable por el usuario
Actores primarios	Desarrolladores
Actores secundarios	Usuario
Trigger	Los desarrolladores deben realizar enemigos que sigan un orden de ataque el cual involucre al usuario a desarrollar estrategias que le permitan vencer a los enemigos de la mejor manera posible
Flujo Principal	
Pasos	Acciones
1	Definir completamente el comportamiento de los enemigos en el juego
2	Establecer las formas de daño que pueden recibir y hacer los enemigos
3	Implementar las mecánicas necesarias para la creación de la estructura de ataque de los enemigos
Extensiones	N/A

Caso de uso	Implementación de dificultades en los diferentes enemigos
Requerimientos relacionados	N/A
Meta en contexto	Lograr un cierto nivel de dificultad para cada tipo de enemigo en el juego
Precondiciones	Definir totalmente las mecánicas, items y enemigos en el juego
Condición para un fin exitoso	Cada tipo de enemigo cuenta con cierto nivel de daño dependiendo de su clasificación
Condición para un fin fallido	Todos los enemigos generan el mismo nivel de daño por lo que el nivel de dificultad es constante en todo momento en el juego
Actores primarios	Desarrolladores
Actores secundarios	Usuario
Trigger	Los desarrolladores deben realizar varios tipos de enemigos que generen diferentes niveles de daño haciendo que la dificultad varíe dependiendo del enemigo al que el usuario ataque
Flujo Principal	
Pasos	Acciones
1	Definir los distintos tipos de enemigos
2	Establecer los rangos de daño que genera cada tipo de enemigo
3	Implementar diversos niveles con la variedad de enemigos en Unity
Extensiones	N/A

Caso de uso	Diseño de evoluciones del personaje acorde a su nivel en el SkillTree
Requerimientos relacionados	N/A
Meta en contexto	Implementar un sistema de evolución donde se muestren las mejoras actuales del personaje .
Precondiciones	Definir totalmente las mecánicas y desarrollo del videojuego.
Condición para un fin exitoso	Se muestra la evolución por las mejoras del personaje principal en el juego
Condición para un fin fallido	No se muestra claramente la evolución por las mejoras del personaje principal en el juego
Actores primarios	Desarrolladores
Actores secundarios	Usuario
Trigger	Los desarrolladores deben realizar un sistema de evolución del personaje de acuerdo a las mejoras del personaje
Flujo Principal	
Pasos	Acciones
1	Establecer el tipo de mejoras que se pueden hacer por el SkillTree
2	Definir el tipo de evolución que se mostrará al realizar las mejoras por medio del SkillTree
3	Implementar la evolución del personaje principal
Extensiones	N/A

Caso de uso	Diseño de evoluciones de aliados acorde a las mejoras desbloqueadas en el skilltree
Requerimientos relacionados	N/A
Meta en contexto	Diseñar un cambio en las habilidades de los aliados al jugador principal en relación a las mejoras realizadas en el skilltree.
Precondiciones	Definir las mejoras presentadas en el skilltree y su posible aplicación a los personajes aliados.
Condición para un fin exitoso	Los personajes aliados tienen una mejora en habilidad propiciada por una elección en el skilltree.
Condición para un fin fallido	Los personajes aliados no presentan mejoras en habilidad o las mejoras no mantienen congruencia con las seleccionadas en el skilltree.
Actores primarios	Desarrolladores
Actores secundarios	Usuario
Trigger	Los desarrolladores deben implementar la extensión de las habilidades en el skilltree a los aliados, las cuales serán aprovechadas por el usuario en su experiencia a lo largo del juego.
Flujo Principal	
Pasos	Acciones
1	Definir completamente el skilltree a ser usado en el juego.
2	Adaptar las mejoras ofrecidas en el skilltree para el personaje principal a sus aliados.
3	Desarrollar el uso de las mejoras previamente definidas en Unity a lo largo del videojuego para los personajes aliados.
Extensiones	N/A

Caso de uso	Diseño de evoluciones de enemigos acorde a la dificultad
Requerimientos relacionados	N/A
Meta en contexto	Diseñar un cambio en las habilidades de los enemigos conforme una dificultad base designada.
Precondiciones	Diseñar a los enemigos y el momento dentro del desarrollo del videojuego en el que se presentarán.
Condición para un fin exitoso	Los enemigos presentan diferentes habilidades base a lo largo del juego.
Condición para un fin fallido	Los enemigos no presentan habilidades iniciales o no presentan un cambio en habilidades a lo largo del juego.
Actores primarios	Desarrolladores
Actores secundarios	Usuario
Trigger	Los desarrolladores deben implementar las evoluciones de los diferentes enemigos con los cuales el usuario se encontrará a lo largo del juego.
Flujo Principal	
Pasos	Acciones
1	Definir a los enemigos que se encontrarán a lo largo del juego.
2	Diseñar los cambios que presentarán los diversos tipos de enemigos.
3	Implementar las evoluciones de los enemigos a lo largo del desarrollo de la historia, hordas y niveles del juego.
Extensiones	N/A

Caso de uso	Diseño de evoluciones de enemigos acorde a progreso en el juego
Requerimientos relacionados	N/A
Meta en contexto	Diseñar un cambio en las habilidades de los enemigos conforme al progreso del usuario en el juego.
Precondiciones	Definir los cambios/evoluciones que pueden presentar los enemigos ocupando su dificultad base como referencia.
Condición para un fin exitoso	Los enemigos presentan un cambio en dificultad conforme al progreso del usuario en el juego.
Condición para un fin fallido	Los enemigos no presentan habilidades o no presentan un cambio incremental en habilidades a lo largo del juego.
Actores primarios	Desarrolladores
Actores secundarios	Usuario
Trigger	Los desarrolladores deben implementar las evoluciones de los diferentes enemigos con los cuales el usuario se encontrará a lo largo del juego en relación a su progreso.
Flujo Principal	
Pasos	Acciones
1	Definir a los enemigos que se presentarán a lo largo del juego.
2	Diseñar los cambios que presentarán los enemigos en relación al progreso del usuario.
3	Implementar las evoluciones de los enemigos en Unity conforme al punto en el que se encuentra el usuario dentro del juego.
Extensiones	N/A

Caso de uso	Implementación de selección de personajes aliados y movimientos de estos
Requerimientos relacionados	N/A
Meta en contexto	Establecer un entorno de selección de personajes aliados y movimientos para el jugador principal.
Precondiciones	Definir los personajes aliados y movimientos que estos pueden tener.
Condición para un fin exitoso	Los personajes aliados y sus posibles movimientos pueden ser seleccionados dentro del juego.
Condición para un fin fallido	Los personajes aliados no pueden ser seleccionados o únicamente pueden ser seleccionados los personajes sin posibilidad de elección de movimientos.
Actores primarios	Desarrolladores
Actores secundarios	Usuario
Trigger	Los desarrolladores deben implementar un entorno de selección que permita al usuario designar a sus aliados y sus movimientos al inicio del juego.
Flujo Principal	
Pasos	Acciones
1	Definir las elecciones para aliados y movimientos.
2	Diseñar el entorno dentro del juego que permita realizar una selección fácil y completa para el usuario.
3	Implementar en Unity el entorno y las posibles elecciones designadas previamente.
Extensiones	N/A

Caso de uso	Implementación de aparición de enemigos y movimiento de estos
Requerimientos relacionados	N/A
Meta en contexto	Determinar los momentos y lugares en los que los enemigos aparecerán dentro del juego además de los movimientos que tendrán dentro de este.
Precondiciones	Definir a los enemigos y los posibles movimientos que pueden tener dentro del juego.
Condición para un fin exitoso	Los enemigos aparecen en los momentos y lugares adecuados además de que presentan un movimiento acorde al diseño del juego.
Condición para un fin fallido	Los enemigos aparecen en momentos o lugares arbitrarios o presentan movimientos incongruentes con el diseño del juego.
Actores primarios	Desarrolladores
Actores secundarios	Usuario
Trigger	El usuario se encuentra con los diferentes enemigos activos a lo largo de su experiencia de juego, los cuales tendrán su aparición y movimientos determinados por los desarrolladores.
Flujo Principal	
Pasos	Acciones
1	Definir a los enemigos y los posibles movimientos que pueden tener en el juego.
2	Establecer congruencia entre los momentos, lugares y movimientos de los enemigos y el resto del desarrollo del juego.
3	Implementar los aspectos previamente determinados de los enemigos en Unity, cuidando mantener coherencia entre todos los rubros considerados.
Extensiones	N/A