



Tecnológico de Monterrey

Especificación de requerimientos de software

TC2005B: Construcción de software y toma de decisiones

Grupo 501

Andrea Alexandra Barrón Córdova A01783126

Mario Ignacio Frías Pina A01782559

Alan Anthony Hernández Pérez A01783347

Oswaldo Ilhuicatzí Mendizábal A01781988

Alejandro Fernandez del Valle Herrera A01024998

Bajo la instrucción de

Esteban Catillo Juarez

14 de mayo de 2023

Historias de Usuario

Las siguientes historias son transcritos casi textuales de los requerimientos sobre los cuales hablamos con los clientes.

Recopilación y base de datos

Yo como cliente me gustaría almacenar checkpoints, inventario que ocupas, así como tener si es posible algo en los escenarios o incluso los puntos de vida, ataque, mana, porque es información que se debe recuperar, incluso escenarios estaría bien.

En cuanto a las bases de datos me gustaría tener tablas separadas, siendo una base de datos relacional. Quiero sentirme seguro, con tablas que se entienden sin explicar, bonitas, cosas malas no.

Estadísticas sobre el uso del juego. que te permitan como desarrollador saber como juegan los usuarios para saber cómo mejorarlo arma que más se usa, cuántas veces mueres, juegan, etc.

Página web

Quiero que dentro de la página haya una explicación del juego, un manual de como jugarlo, las estadísticas y gráficos.

Me gustaría que usaran css javascript y igs (tentador).

Que la página tenga una conexión con la base de datos.

Videojuegos

Si es que hay periféricos con propósito y que sean consistentes, menor cantidad de periféricos que se ocupen de la mejor manera. Preferente que no tenga click para poder usar un gamepad o hacerlo “versátil”. En resumen asignación de botones con un sentido de ser

Tiene que tener una historia, comenzar y terminar, tener una sensación de término, resultado (ganar o perder) y pueda tener diferentes finales.

Que tu personaje puede evolucionar, una capacidad de decisión para que sea rejugable. En el tema de ser rejugable se pueden como escape rooms para que sea divertido.

Debe tener cutscenes.

Requerimientos Funcionales

1. Desarrollo de Documentación inicial
2. *Creación de Diagramas de UML*
 - a. *ER*
 - b. *Casos de Uso*
 - c. *Actividad*
3. Diseño del concepto del juego
4. Desarrollo de mecánicas básicas
 - a. Movimiento de cámara
 - b. Movimiento del personaje
 - i. dash
 - c. Combate
 - i. Cambio de Armas
 - ii. Trompo
 - iii. Balero
 - iv. Espada
 - d. Interface
 - e. Menús
 - f. Enemigo básico
 - g. Objetos Interactuables
 - i. Trampas
 - ii. Tablas de presión
 - iii. Plataformas de un sentido
5. Creación de gráficas para el juego
 - a. *Background*
 - b. Animaciones
 - c. Enemigos
 - d. NPCs
 - e. Jugador
6. Diseño conceptual de niveles (mock-ups)
 - a. Bosque
 - b. Pueblo
 - c. Alcantarilla
7. Implementación dentro de Unity de niveles
8. Implementación de checkpoints
9. Diseño de interacciones de texto de los NPCs
 - a. Implementación de NPCs Aliados
 - b. Implementación de clases de Enemigos
10. Implementación de la historia dentro del juego
11. Implementación de tienda de items
12. Implementación de mejoras (de estadísticas)

- a. Dulces (Buffs Temporales)
 - b. Comida (Regeneración de Vida)
 - c. Muñecas quitapenas (Buffs permanentes)
- 13. Creación de escenas para el desarrollo de la historia o como indicadores
- 14. Implementar decisiones en mejoras del personaje
- 15. Implementar *backtracking* dentro del mundo
- 16. Implementación de puzzles a resolver con desbloqueables
- 17. Creación de Sonidos y Música para el juego
- 18. Diseño conceptual de la base de datos
 - a. Detección del dominio de datos
 - b. Restricciones de integridad
 - c. Análisis de datos relevantes a almacenar
- 19. Creación de la base de datos en MySQL
- 20. Creación de tablas de dropeo, debe de estar lista antes de que inicie el juego y es solo de consulta, y no de inserción por usuarios.
- 21. Inserción de datos del juego a la base de datos
- 22. Recopilación de información de usuarios desde la página Web
- 23. Diseñar la página web
- 24. Modelar una página de inicio
- 25. Embeber el juego dentro de la página Web
- 26. Creación de Manual de Usuario dentro de la página Web
- 27. Mostrar datos en gráficas embebidas en la página Web
- 28. Montar la Página Web y la base de datos en un servidor en línea

Requerimientos No Funcionales

- 1. Juego de genero RPG
- 2. Aprender el uso de MySQL
- 3. Aprender el uso de Unity
- 4. Aprender el uso de HTML
- 5. Aprender el uso de CSS
- 6. Aprender el uso de JS
- 7. Aprender el uso de Git
- 8. Uso de Mouse + Teclado

Actores

1. Usuario
2. DataBase Management System
3. Motor de Juego de Unity
4. Página Web
5. Administrador

Tablas de Casos de Usos

Nombre de caso de uso	3.- Diseño del concepto del juego	
Requerimientos relacionados	6 Diseño conceptual de niveles 10 Implementación de la historia dentro del juego	
Meta en contexto	Tener una idea clara de cómo se va a desarrollar el juego. Permitir que todos los integrantes del equipo sepan qué se debe de desarrollar y cómo es que debe de ser desarrollado, para que a la hora del desarrollo, sea claro y no se gaste tiempo haciendo correcciones.	
Precondiciones	Tener una idea de para qué es el juego y la historia.	
Condiciones de éxito	Se tiene un documento que permite ver cuales son las mecánicas y conceptos usados en el juego.	
Condiciones de fallo	El documento explicando los conceptos del juego se encuentra incompleto / no se entiende / no existe	
Actor Primario	Administrador	
Actor Secundario	Ninguno.	
Trigger	Ya se cuenta con un elemento	
Flujo Principal	Paso	Acción
	1	Reunirse para terminar de definir los conceptos
	2	Crear un primer draft de lo que se busca que este en el juego
	3	Hacer las mecánicas del juego
	4	Compartir las ideas con el cliente
	5	Realizar cambios necesarios

Nombre de caso de uso	4.1.-Desarrollo de movimiento de la cámara	
Requerimientos relacionados	4.2 Desarrollo del movimiento del personaje 4.4 Desarrollo de la interfaz de usuario 5.1 Creación de gráficas de fondo	
Meta en contexto	El jugador puede, de forma fácil, navegar y entender el mundo sin resultar aturdido.	
Precondiciones	Unity debe tener un jugador controlable.	
Condiciones de éxito	El jugador no piensa en la cámara.	
Condiciones de fallo	El jugador se mareo o tiene complicaciones al ver abajo o al moverse.	
Actor Primario	Motor de Juego de Unity.	
Actor Secundario	Usuario.	
Trigger	La cámara sigue al jugador.	
Flujo Principal	Paso	Acción
	1	Se implementa <i>Cinemachine</i> .
	2	La cámara sigue al jugador.
	3	Se afina parámetros para poder tener un flujo que sirva con el tipo de movimiento que vamos a tener.
	4	Se agregan elementos necesarios para ubicar mejor al jugador.

Nombre de caso de uso	4.2.1.- Desarrollo de la habilidad de dash del personaje	
Requerimientos relacionados	4.2 Desarrollo del movimiento del personaje	
Meta en contexto	El jugador puede usar la mecánica dash dentro de combate o como habilidad de movimiento.	
Precondiciones	Tener movimiento izquierda y derecha y brinco terminados.	
Condiciones de éxito	El jugador presiona un botón y se crea un dash en la posición opuesta.	
Condiciones de fallo	El jugador no puede hacer dash o se atora al hacerlo.	
Actor Primario	Motor de Juego de Unity.	
Actor Secundario	Usuario.	
Trigger	El jugador presiona la tecla “CTRL”.	
Flujo Principal	Paso	Acción
	1	El jugador ingresa una entrada para dash.
	2	Se agrega una fuerza en la dirección contraria.
	3	Se hace un cooldown.
	4	Se refina el movimiento y la fuerza correcta.

Nombre de caso de uso	4.3.1 Desarrollo del sistema de cambio de armas	
Requerimientos relacionados	4.3 Desarrollo de armas	
Meta en contexto	El usuario cambia su arma actual dentro del juego dentro de las diferentes disponibles.	
Precondiciones	El personaje debe de tener por lo menos 2 armas entre las cuales puede cambiar para poder hacer la elección.	
Condiciones de éxito	El personaje cambia de arma.	
Condiciones de fallo	El personaje no puede cambiar de arma aunque se tenga más de una opción.	
Actores Primarios	Usuario	
Actores Secundarios	Motor de Juego de Unity.	
Trigger	El usuario presiona el botón de cambio de arma.	
Flujo Principal	Paso	Acción
	1	El usuario presiona el botón de cambio de armas.
	2	Se recopila la información del arma actual y la de la siguiente arma dentro de la base de datos.
	3	El motor de juego desactiva el arma actual del personaje.
	4	El motor de juego activa la siguiente arma en la lista de armas.

Nombre de caso de uso	4.4 Desarrollo de la Interfaz de Usuario	
Requerimientos relacionados	4.d Interfaz	
Meta en contexto	Tener un indicador visual de la barra de vida, contador de monedas y selección de arma visible en todo momento dentro del juego.	
Precondiciones	Creación previa de los siguientes sprites: barra de vida, monedas y su contador.	
Condiciones de éxito	Los elementos seleccionados se muestran de manera constante en pantalla.	
Condiciones de fallo	Los elementos mencionados no se muestran en pantalla.	
Actor Primario	Motor de Juego de Unity.	
Actor Secundario	Ninguno.	
Trigger	El usuario carga la partida.	
Flujo Principal	Paso	Acción
	1	El usuario se autentifica y le da al botón de “cargar la última partida guardada”
	2	Se recopila la información actual del indicador de barra de vida (nivel de salud), la cantidad de monedas recolectadas hasta ese momento y el arma o armas equipadas dentro de la base de datos.
	3	El motor de juego activa los contadores, acumuladores y estadísticas del jugador
	4	El usuario recibe daño de algún enemigo o se cura, y derivado de estos eventos, la información se actualiza en la base de datos, agarra una moneda o cambia de arma
	5	Se actualizan los valores dentro de la base de datos

Nombre de caso de uso	4.5 Desarrollo de los Menús	
Requerimientos relacionados	4.e. Menús	
Meta en contexto	Implementar los menús dentro del juego.	
Precondiciones	Tener las escenas del juego y los sprites del menú creadas.	
Condiciones de éxito	Los menús y los botones dentro de este pasan a las escenas correspondientes.	
Condiciones de fallo	Los botones dentro del menú inicial no cumplen su función designada.	
Actor Primario	Motor de Juego Unity.	
Actor Secundario	Usuario.	
Trigger	El usuario carga la partida.	
Flujo Principal	Paso	Acción
	1	Se crean las escenas del juego.
	2	Se crea el elemento “menú” dentro de la jerarquía del motor de juego Unity.
	3	Se ligan cada botón con la escena a la que debe ser dirigida dentro del juego.

Nombre de caso de uso	4.6.- Desarrollo de una clase enemigo básico	
Requerimientos relacionados	9.2 Implementación de diferentes clases de enemigos. 12 Implementación de mejoras (de estadísticas) 15 Implementación de backtracking dentro del mundo	
Meta en contexto	Poder crear enemigos de forma rápida y sencilla para poder agregar una variedad grande de enemigos.	
Precondiciones	Tener una forma de dañar al jugador	
Condiciones de éxito	Se tiene un enemigo realizado con esta clase base.	
Condiciones de fallo	No se puede crear un enemigo utilizando esta clase base.	
Actor Primario	Motor de Juego de Unity.	
Actor Secundario	Ninguno.	
Trigger	El jugador se aproxima.	
Flujo Principal	Paso	Acción
	1	Se diseña la creación de enemigos.
	2	Se crea un enemigo sencillo que implementa la clase.
	3	Se programa que el enemigo se acerque al jugador basado en su posición usando la clase genérica.
	4	El enemigo hace daño al jugador al hacer colisión.
	5	Se establece estatus de vida y cantidad de daño.
	6	Se crea un script para ver como muere.

Nombre de caso de uso	4.7.1 Desarrollo de trampas	
Requerimientos relacionados	4.2 Desarrollo del movimiento del personaje 4.3 Desarrollo de Armas 4.7.2 Desarrollo de tablas de presión.	
Meta en contexto	El jugador tiene que ver ciertos elementos para poder crear zonas donde pueda caer, o encontrar zonas escondidas.	
Precondiciones	Tener un tileset listo.	
Condiciones de éxito	Se pueden agregar trampas visuales en el juego	
Condiciones de fallo	Las trampas son más de 5% diferentes al piso normal y son fácilmente detectables.	
Actor Primario	Motor de Juegos de Unity.	
Actor Secundario	Usuario.	
Trigger	El jugador cae en una trampa.	
Flujo Principal	Paso	Acción
	1	Se crea un nuevo tilemap encima del de piso.
	2	Se agrega al tilemap los mismos lugares que las trampas.
	3	Se establece una diferencia mínima entre las trampas y el piso.

Nombre de caso de uso	4.7.2.- Desarrollo de tablas de presión.	
Requerimientos relacionados	4.2 Desarrollo del movimiento del personaje 4.3 Desarrollo de Armas 4.7.1 Desarrollo de trampas	
Meta en contexto	Tener un componente interactivo en el mundo que mande a llamar alguna acción cuando el jugador o algo interactúe con el.	
Precondiciones	Tener un jugador controlable, con trompos y elementos adicionales.	
Condiciones de éxito	Al acercarse el jugador o aventar algún elemento la tabla se activa.	
Condiciones de fallo	El jugador no activa la tabla al estar encima de ella o se activa sin que el jugador este en la tabla.	
Actor Primario	Motor de Juegos de Unity.	
Actor Secundario	Usuario.	
Trigger	Cuando el jugador o un objeto se pone en contacto con su Collision Trigger, o bien un arma del jugador.	
Flujo Principal	Paso	Acción
	1	Se crean el sprite y el trigger para la tableta.
	2	Se programa una lista de objetos con los que tiene contacto.
	3	Cuando un objeto entre a la tabla se agregue a la lista y cuando salga se remueva de esta.
	4	Cuando la lista está vacía se desactiva la tabla, de otra manera se activa.
	5	Se crea una lista de las etiquetas con las que puede interactuar para poder utilizarla en distintos escenarios.

Nombre de caso de uso	4.7.3 Desarrollo de plataformas de un solo sentido dentro del escenario de Unity.	
Requerimientos relacionados	7 Implementación de mapa dentro de Unity.	
Meta en contexto	Un elemento iterativo para la progresión de la exploración del mundo.	
Precondiciones	Selección de sprites para mostrar como plataformas.	
Condiciones de éxito	El jugador puede acceder a la plataforma traspasándola desde abajo, y al pasarla pueda caer en ella.	
Condiciones de fallo	El jugador no atraviesa la plataforma o no la puede pisar.	
Actor Primario	Motor de Juego de Unity.	
Actor Secundario	Usuario.	
Trigger	El jugador cae en la plataforma o la atraviesa con un salto.	
Flujo Principal	Paso	Acción
	1	Se inserta la plataforma a juego.
	2	Se cambian sus atributos para que sea de un solo sentido.
	3	Se cambia el ángulo del borde de la plataforma.

Nombre de caso de uso	5.1 Creación de gráficas de fondo.
-----------------------	------------------------------------

Requerimientos relacionados	7 Implementación de mapa dentro de unity.	
Meta en contexto	Tener fondos para poder reflejar el ambiente.	
Precondiciones	Ninguna.	
Condiciones de éxito	El elemento se ve reflejado en el mapa y cambia conforme el jugador entra en diferentes zonas.	
Condiciones de fallo	El fondo se transpone, ocasionando que oculte al jugador y elementos decorativos.	
Actor Primario	Motor de Juego de Unity.	
Actor Secundario	Ninguno.	
Trigger	El usuario carga la partida.	
Flujo Principal	Paso	Acción
	1	Se buscan fondos en internet.
	2	Se descarga el fondo y se coloca en la carpeta de assets.
	3	El motor de juego refleja el asset como fondo.

Nombre de caso de uso	5.3.- Creación de graficas para los enemigos	
Requerimientos relacionados	4.6 Desarrollo de una clase enemigo básico	
Meta en contexto	Tener visuales atractivos para representar enemigos.	
Precondiciones	Tener una lista de los enemigos que queremos agregar.	
Condiciones de éxito	Se tienen imágenes en formato de los enemigos.	
Condiciones de fallo	No se cuentan con las imágenes de los enemigos.	
Actor Primario	Motor de Juego de Unity.	
Actor Secundario	Ninguno.	
Trigger	El jugador ve un enemigo.	
Flujo Principal	Paso	Acción
	1	Se crean visuales de los enemigos.
	2	Se comparan para ver como se ven en el juego.
	3	Se importan a unity .

Nombre de caso de uso	5.4 Creación de gráficas para NPCs	
Requerimientos relacionados	6 Diseño conceptual de niveles.	
Meta en contexto	Tener todos los diseños/sprites de los personajes que estarán situados en la aldea y no sean jugadores.	
Precondiciones	Tener bocetos de los NPCs.	
Condiciones de éxito	Tener diferentes NPCs con sus respectivos sprites y animación adecuada.	
Condiciones de fallo	Las animaciones de los NPCs no funcionan.	
Actor Primario	Motor de Juego de Unity.	
Actor Secundario	Ninguno.	
Trigger	Al ingresar al proyecto de unity	
Flujo Principal	Paso	Acción
	1	Se pasan los dibujos a Photoshop.
	2	Se detalla el diseño.
	3	Se importa el diseño a los assets de Unity.
	4	Se crea un objeto en unity.
	5	Se inserta el diseño de photoshop al objeto para que sea hijo del mismo.
	6	Se aplica capsule collider 2 al objeto hijo.
	7	Se utiliza el animation con 2D skeletal de Unity para darle las animaciones correspondientes a los NPCs.
	8	Se envía al objeto padre en prefabs.

Nombre de caso de uso	5.5 Creación de gráficas para el jugador	
Requerimientos relacionados	5 Creación de gráficos	
Meta en contexto	Crear los sprites y gráficos del personaje principal, el cual será controlado por el usuario.	
Precondiciones	Tener los sprites del personaje principal: posición de ataque y de defensa, movimiento lateral izquierdo / derecho, movimiento arriba / abajo, salto.	
Condiciones de éxito	El personaje muestra las animaciones de acuerdo a la acción que esté realizando.	
Condiciones de fallo	El personaje no muestra las animaciones correspondientes.	
Actor Primario	Motor de juego Unity .	
Actor Secundario	Usuario	
Trigger	Iniciar el juego	
Flujo Principal	Paso	Acción
	1	Crear los sprites de las acciones del personaje principal: posición de ataque y de defensa, movimiento lateral izquierdo / derecho, movimiento arriba / abajo, salto.
	2	Crear el objeto “jugador” dentro del motor de juego Unity.
	3	Asignar los sprites a la acción correspondiente que realiza el jugador dentro del juego.

Nombre de caso de uso	6.- Diseño conceptual de Niveles.	
Requerimientos relacionados	6. a - Bosque 6. b - Pueblo 6. c - Alcantarilla	
Meta en contexto	Diseñar el layout (distribución de los elementos dentro de los niveles) que tendrá cada una de las zonas del juego.	
Precondiciones	Crear los modelos de los niveles: bosque, pueblo y alcantarilla.	
Condiciones de éxito	El nivel es atractivo e interesante para el usuario.	
Condiciones de fallo		
Actor Primario	Administrador.	
Actor Secundario	Ninguno.	
Trigger		
Flujo Principal	Paso	Acción

Nombre de caso de uso	7.- Implementación de mapa dentro de unity	
Requerimientos relacionados	6. Diseño conceptual de Niveles.	
Meta en contexto	Tener un mapa con distintos lugares para que el jugador pueda navegar y donde se puedan agregar trampas, enemigos y amigos.	
Precondiciones	Tener los tilemaps listos, tener las trampas listas, tener colisiones listas, y tener los sprites listos.	
Condiciones de éxito	El mapa está completo.	
Condiciones de fallo	El mapa está incompleto o no se pueden agregar los elementos descritos en el diseño.	
Actor Primario	Motor de Juegos de Unity.	
Actor Secundario	Ninguno.	
Trigger	El usuario carga el nivel.	

Flujo Principal	Paso	Acción
	1	Se crea el mundo.
	2	Se pinta todo el mapa.
	3	Se agregan las luces.
	4	Se agregan trampas.
	5	Se agregan decoraciones finales.
	6	Se agregan fondos.

Nombre de caso de uso	8.- Implementación de checkpoints.	
Requerimientos relacionados		
Meta en contexto		
Precondiciones		
Condiciones de éxito		
Condiciones de fallo		
Actor Primario	Motor de Juego de Unity.	
Actor Secundario	Usuario.	
Trigger		
Flujo Principal	Paso	Acción

Nombre de caso de uso	9 Diseño de las interacciones de textos los NPCs	
Requerimientos relacionados	4.5 Desarrollo de los Menús 10.- Implementación de la historia dentro del juego	
Meta en contexto	Un elemento textual asignado a los NPC que se despliegue cuando se pulse una tecla del teclado o cuando se acerca a un activador de evento, dependiendo si es enemigo o aliado.	
Precondiciones	Tener el diseño de la Interfaz gráfica	
Condiciones de éxito	Cada NPCs despliega diferentes diálogos de textos, entendibles y legibles, cuando se presiona la tecla asignada.	
Condiciones de fallo	No se despliegan los diálogos de texto en el momento indicado o son ilegibles para el usuario.	
Actor Primario	Administrador.	
Actor Secundario	Ninguno.	
Trigger	Cuando se presione la tecla asignada o el jugador se acerque al NPC.	
Flujo Principal	Paso	Acción
	1	Se escribe en un documento los diálogos de cada NPCs.
	2	Se crean canvas dentro de unity.
	3	Se insertan los diálogos correspondientes en los diferentes canvas.
	4	Se colocan los canvas en los objetos de los NPCs como objetos hijos.
	5	Se crean scripts para poder esconder y mostrar los diálogos de los NPC.

Nombre de caso de uso	9.1 Implementación de NPCs aliados	
Requerimientos relacionados		
Meta en contexto		
Precondiciones		
Condiciones de éxito		
Condiciones de fallo		
Actor Primario	Motor de Juego de Unity.	
Actor Secundario	Usuario.	
Trigger		
Flujo Principal	Paso	Acción

Nombre de caso de uso	9.2 Implementación de diferentes clases de enemigos	
Requerimientos relacionados		
Meta en contexto		
Precondiciones		
Condiciones de éxito		
Condiciones de fallo		
Actor Primario	Motor de Juego de Unity.	
Actor Secundario	Ninguno.	
Trigger		
Flujo Principal	Paso	Acción

Nombre de caso de uso	10.- Implementación de la historia dentro del juego	
Requerimientos relacionados		
Meta en contexto		
Precondiciones		
Condiciones de éxito		
Condiciones de fallo		
Actor Primario	Motor de Juego de Unity.	
Actor Secundario	Ninguno.	
Trigger		
Flujo Principal	Paso	Acción

Nombre de caso de uso	15 Implementación de backtracking	
Requerimientos relacionados	16 Implementación de puzles a resolver con desbloqueables 8 Implementación de checkpoints 7 implementación de mapa dentro de unity	
Meta en contexto	Desarrollar áreas que sean inaccesibles la primera vez que sean recorridas por el usuario. Al progresar con sus habilidades el jugador será capaz de volver a estas zonas y acceder a ellas usando dichas habilidades o resolviendo puzles.	
Precondiciones	Implementación de mapa dentro de Unity junto con puzles a resolver con desbloqueables.	
Condiciones de éxito	El usuario puede desbloquear las zonas luego de cumplir ciertos requerimientos(puzles o uso de una habilidad específica)y que obtenga recompensas.	
Condiciones de fallo	Las áreas no se desbloquean después de cumplir los requerimientos o son accesibles antes de cumplirlos.	
Actor Primario	Usuario.	
Actor Secundario	Motor de Juego de Unity.	
Trigger	El usuario cumple con los requerimientos o obtiene las	

	habilidades necesarias.	
Flujo Principal	Paso	Acción
	1	Se modifica el mapa para diseñar áreas inaccesibles para el usuario.
	2	Se implementan los puzles ya realizados en las áreas diseñadas.
	3	Se genera un script para saber cuando el usuario superó el puzle.
	4	Se asigna el script al objeto de piso.

Nombre de caso de uso	16.- Implementación de puzles a resolver con desbloqueables.	
Requerimientos relacionados	6 Diseño conceptual de Niveles. 4.3 Desarrollo de armas.	
Meta en contexto	Que el jugador tenga la oportunidad de regresar en el mapa para acceder a nuevas áreas a través de puzles que se puedan resolver con objetos obtenidos a lo largo del juego. Con esto se da la oportunidad de backtracking (elemento del género Metroidvania).	
Precondiciones	Debe de existir un objeto desbloqueable para que este pueda ser utilizado.	
Condiciones de éxito	El jugador puede resolver de manera exitosa el puzzle.	
Condiciones de fallo	El jugador no resuelve de manera exitosa el puzzle.	
Actor Primario	Usuario.	
Actor Secundario	Motor de Juego de Unity.	
Trigger	El jugador obtiene el objeto que sirve para desbloquear las nuevas zonas dentro del mapa del juego.	
Flujo Principal	Paso	Acción
	1	El usuario obtiene el objeto para desbloquear la nueva zona dentro del mapa del juego.
	2	Se actualiza la base de datos al almacenar los datos del objeto obtenido.
	3	El motor de juego Unity desbloquea las zonas

		del mapa que anteriormente estaban bloqueadas.
	4	El usuario regresa en a las zonas dentro del mapa a las que no podía acceder con anterioridad.
	5	El usuario accede a las nuevas zonas usando el objeto desbloqueado.

Nombre de caso de uso	17 Creación de sonidos y música para el juego.	
Requerimientos relacionados	6 Diseño conceptual de niveles.	
Meta en contexto	Durante la ejecución del juego, se reproducen sonidos ambientales y música para darle ambientación al videojuego .	
Precondiciones	Ninguna.	
Condiciones de éxito	Los sonidos y música se reproducen con buena calidad de sonido.	
Condiciones de fallo	No se escuchan los sonidos y música o tienen distorsiones en el audio.	
Actor Primario	Motor de Juego de Unity.	
Actor Secundario	Ninguno.	
Trigger	Al ingresar a proyecto de unity	
Flujo Principal	Paso	Acción
	1	Se busca sonidos y músicas en diferentes páginas en los cuales posean licencia libre.
	2	Se descargan los sonidos/músicas seleccionados.
	3	Se importan a un editor de sonido.
	4	Se utilizan editores de sonido, para la modificación de los sonidos descargados y generar nuevos sonidos.
	5	Se pasan los audios a unity en la carpeta de assets.
	6	Se crean diversos audio source en las escena .

	7	Se asignan los audios a cada audio source.
	8	Se inserta el elemento de audio listener.

Nombre de caso de uso	21 Inserción de datos del juego a la base de datos	
Requerimientos relacionados	18 Diseño conceptual de la base de datos 19 Creación de la base de datos en MySQL 20 Creación de tablas de dropeo, debe de estar lista antes de que inicie el juego y es solo de consulta, y no de inserción por usuarios.	
Meta en contexto	Los usuarios pueden comparar las estadísticas de otras personas. Al igual los desarrolladores puedan ver cómo mejorar el juego y como se está utilizando las herramientas actuales.	
Precondiciones	Tener una base de datos establecida.	
Condiciones de éxito	Los datos se insertan por completo en el juego.	
Condiciones de fallo	No hay datos dentro del juego.	
Actor Primario	DataBase Management System.	
Actor Secundario	Motor de Juego de Unity.	
Trigger	El usuario carga la partida.	
Flujo Principal	Paso	Acción
	1	Se construye una base de datos.
	2	La base de datos, implementada en la página web, se actualiza con las acciones de otros usuarios.
	3	Se inserta la base de datos dentro del código de la página

Nombre de caso de uso	22.- Recopilación de información de usuarios desde la página web.	
Requerimientos relacionados	4.3 Desarrollo de armas. 9.1 Implementación de NPCs aliados. 12 Implementación de mejoras (de estadísticas). 20 Creación de tablas de dropeo, debe de estar lista antes de que inicie el juego y es solo de consulta, y no de inserción por usuarios.	
Meta en contexto	Poder obtener estadísticas sobre el uso del juego para que el usuario sepa cómo juegan otros usuarios y para que el desarrollador sepa cómo mejorarlo.	
Precondiciones	Tener una base de datos para las diferentes referencias.	
Condiciones de éxito	La página web se conecta con la base de datos y recopila la información. Muestra las estadísticas.	
Condiciones de fallo	La página no se conecta con la base de datos.	
Actor Primario	DataBase Management System	
Actor Secundario	Página Web	
Trigger	El usuario entra a la página.	
Flujo Principal	Paso	Acción
	1	El usuario entra a la página.
	2	El usuario se registra con nombre y contraseña
	3	La base de datos recibe esa información para añadirlo a la tabla de usuarios
	4	El usuario juega el juego
	5	Se almacenan las acciones del usuario conforme este toma decisiones.

Nombre de caso de uso	23.- Diseño de la página web	
Requerimientos relacionados	24 Modelar una página de inicio 26 Creación de un manual de usuario dentro de la página web	
Meta en contexto	Establecer una interfaz para que el usuario pueda navegar.	
Precondiciones	Ninguna.	
Condiciones de éxito	La página responde a los comandos de entrada del usuario, como desplazarse o hacer clic.	
Condiciones de fallo	La página no se despliega.	
Actor Primario	Administrador.	
Actor Secundario	Ninguno.	
Trigger	El usuario carga la página usando el enlace de esta.	
Flujo Principal	Paso	Acción
	1	El usuario entra a la página.
	2	El usuario interactúa con la página (ya sea desplazándose o haciendo clic en los elementos
	3	La página se desplaza, si no acciona el video o juego dentro de la página, o cambia a otra página.

Nombre de caso de uso	24.- Modelación de una página de inicio	
Requerimientos relacionados	23 Diseñar la página web	
Meta en contexto	Establecer una interfaz para dar la bienvenida al usuario.	
Precondiciones	Tener una página web.	
Condiciones de éxito	La página responde a los comandos de entrada del usuario, como desplazarse o hacer clic.	
Condiciones de fallo	La página no se despliega.	
Actor Primario	Administrador.	
Actor Secundario	Página Web.	

Trigger	El usuario carga la página usando el enlace de esta.	
Flujo Principal	Paso	Acción
	1	El usuario entra a la página.
	2	El usuario interactúa con la página (ya sea desplazándose o haciendo clic en los elementos
	3	La página se desplaza, sino acciona el video o juego dentro de la página, o cambia a otra página.

Nombre de caso de uso	25.- Embeber el juego dentro de la página web	
Requerimientos relacionados	3-17 Todos los que tengan que ver con el juego. 23 Diseñar la página web. 27 Mostrar datos en gráficas embebidas en la página web. 28 Montar la página web y la base de datos en un servidor en línea.	
Meta en contexto	El usuario puede jugar el juego accediendo desde la página web.	
Precondiciones	Diseño de la página web y creación del juego.	
Condiciones de éxito	Se puede jugar el juego dentro de la página, al igual que recopilar datos.	
Condiciones de fallo	No se puede cargar el juego.	
Actor Primario	Página Web.	
Actor Secundario	Motor de Juego de Unity.	
Trigger	El usuario abre la página con el juego.	
Flujo Principal	Paso	Acción
	1	El usuario entra a la página.
	2	El usuario se mueve a la pestaña donde está el juego.
	3	El usuario hace clic para empezar una nueva partida o cargar la última partida guardada.
	4	Se recopila la información y la partida empieza.

Nombre de caso de uso	26.- Creación de un manual de usuario dentro de la página web	
Requerimientos relacionados	23 Diseñar la página web	
Meta en contexto	El usuario entiende cómo funciona el juego para poder jugarlo.	
Precondiciones	Tener una página web diseñada.	
Condiciones de éxito	La página web muestra el manual.	
Condiciones de fallo	La página web no muestra el manual.	
Actor Primario	Página Web.	
Actor Secundario	Administrador.	
Trigger	El usuario carga la página usando el enlace de esta.	
Flujo Principal	Paso	Acción
	1	El usuario entra a la página.
	2	El usuario interactúa con la página desplazándose hacia la sección de manual.
	3	La página muestra el manual de usuario.

Nombre de caso de uso	27.- Mostrar datos en gráficas embebidas en la página web	
Requerimientos relacionados	18 Diseño conceptual de la base de datos 19 Creación de la base de datos en MySQL 21 Inserción de datos del juego a la base de datos	
Meta en contexto	Los usuarios pueden comparar las estadísticas de otras personas. De igual forma, los desarrolladores puedan ver cómo mejorar el juego y como se está utilizando las herramientas actuales.	
Precondiciones	Tener una página web y una base de datos.	
Condiciones de éxito	La página web muestra las gráficas.	
Condiciones de fallo	La página web no muestra estadísticas.	
Actor Primario	Página Web.	
Actor Secundario	DataBase Management System.	

Trigger	El usuario consulta las estadísticas o se actualizan a base de otros usuarios, por ejemplo usar un arma o matar un enemigo.	
Flujo Principal	Paso	Acción
	1	Un usuario toma una decisión en el juego.
	2	La acción es registrada en la base de datos.
	3	La base de datos se comunica con la página web.
	4	Se traduce la información a una manera gráfica.
	5	La página web muestra las gráficas.

Nombre de caso de uso	28.- Montar la página web y la base de datos en un servidor en línea	
Requerimientos relacionados	18 Creación de la base de datos en MySQL 21 Inserción de datos del juego a la base de datos 24 Diseñar la página web 27 Montar datos en gráficas embebidas en la página Web	
Meta en contexto	Cualquier persona puede acceder a la página web.	
Precondiciones	Tener la página web montada junto, al igual que la base de datos.	
Condiciones de éxito	Se puede acceder a la página web en el servidor.	
Condiciones de fallo	La página no es accesible.	
Actor Primario	DataBase Management System.	
Actor Secundario	Ninguno.	
Trigger	Se accede al servidor.	
Flujo Principal	Paso	Acción
	1	La página web se construye junto con la base de datos.
	2	El servidor se conecta con la página.
	3	El servidor publica la página.

Nombre de caso de uso	4.2 Desarrollo del movimiento del personaje	
Requerimientos relacionados	4.2.1 Desarrollo de la habilidad de dash del personaje. 7 Implementación de mapa dentro de unity. 5.5 Creación de gráficas para el jugador. 5.2 Creación de animaciones. 4.4 Desarrollo de la interfaz del usuario.	
Meta en contexto	El usuario puede moverse a través del mundo usando a su personaje.	
Precondiciones	Creación previa de un asset de un personaje y un escenario de mundo para poder mostrar.	
Condiciones de éxito	El personaje se mueve de lado a lado, saltar y puede esquivar	
Condiciones de fallo	El personaje no es responsivo a los comandos de entrada del usuario	
Actor Primario	Usuario	
Actor Secundario	Motor de Juego de Unity	
Trigger	El usuario presiona las teclas “a” y “d” para el movimiento de lado a lado, “ctrl” para esquivar o espacio para saltar.	
Flujo Principal	Paso	Acción
	1	El usuario presiona el botón para accionar el movimiento
	2	El personaje registra el movimiento
	3	La interfaz muestra el movimiento a través de la animación

Historias de Usuario (12) ...

Juego

Como cliente quiero que el juego sea rpg, por ende que el personaje pueda evolucionar y que haya capacidad de decisión.

Juego

Como cliente quiero que el juego sea divertido: que sea rejugable, tenga cutscenes y, en el caso de metroidvania, tenga algo como escape rooms.

Base de Datos

Juego

Como un desarrollador, quiero tener el juego en Unity comunicándose con una base de datos, para poder utilizar sus herramientas y simplificar el desarrollo y los cambios.

Base de Datos

Yo como cliente me gustaría almacenar checkpoints, inventario.

+ Añada una tarjeta

Product Backlog (41) [33f; 8nf] ...

🕒

AH

Juego

5,2.-Creación de animaciones

🕒

AH

Juego

5,4.-Creación de gráficas para NPCs

🕒

AH

Juego

5,5.-Creación de gráficas para el jugador

OM

Juego

6.-Diseño conceptual de Niveles. Mock-ups, dibujos, etc.

☰

☑ 0/3

OM

Juego

+ Añada una tarjeta

Sprint Backlog ...

SPRINT 0 - Alejandro Fernández del Valle Herrera (22 Abril - 28 Abril)

Sprint 1 - Oswaldo Ilhuicatzí Mendizábal (29 Abril - 12 Mayo)

Sprint 2 - Mario Ignacio Frias Pina (13 Mayo - 19 Mayo)

Juego

4,3,1.-Desarrollo del sistema de cambio de armas

MP

Juego

4,2.-Desarrollo del movimiento del personaje

🟢

Juego

+ Añada una tarjeta

Pendiente ...

+ Añada una tarjeta

En Desarrollo ...

+ Añada una tarjeta

En Revision ...

+ Añada una tarjeta

Terminados



SPRINT 1- Alejandro Fernández del Valle Herrera - (22 Abril - 28 Abril)

Juego

Desarrollo de movimiento de la cámara

Juego

Desarrollo del movimiento del personaje

Juego

Desarrollo de Armas

✓ 3/3

Juego

Desarrollo de la Interface de Usuario

✓ 1/6

Juego

Desarrollo de plataformas de un

+ Añada una tarjeta



MySQL Tablas



Personaje

idPersonaje

vida

ataque

velocidad

espada?

balero?

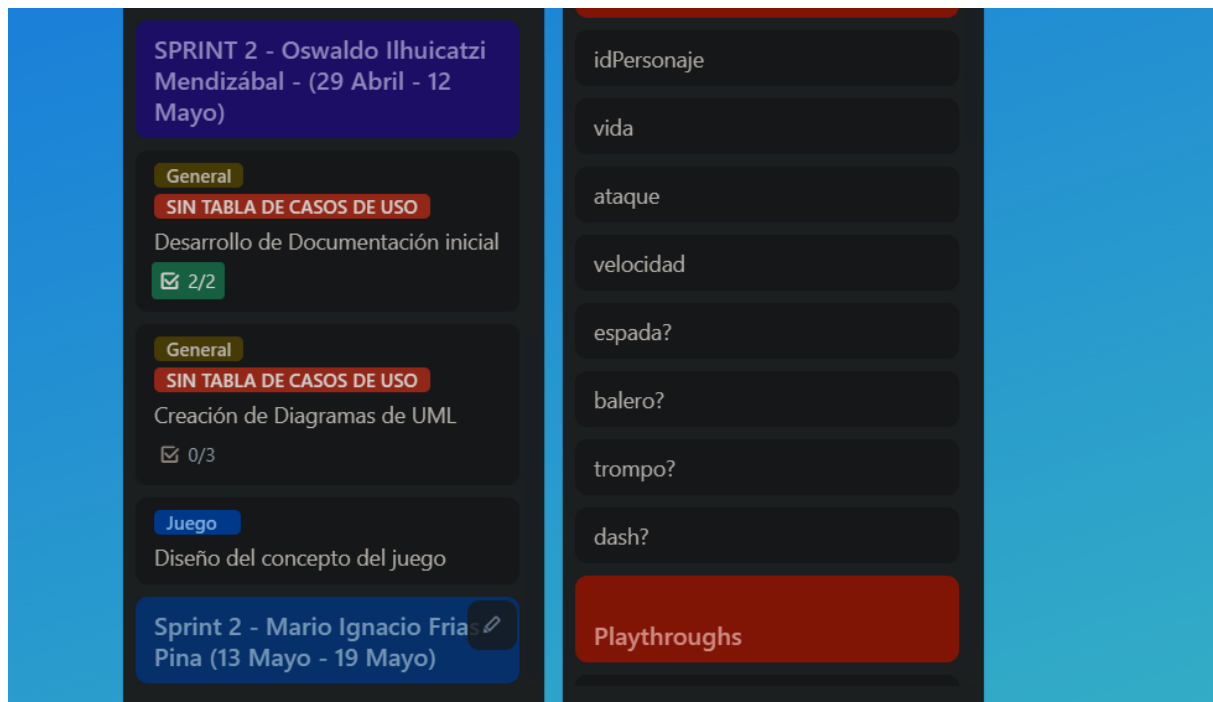
trompo?

dash?

Playthroughs

+ Añada una tarjeta





Enlace del trello:

<https://trello.com/invite/b/hLOBSSiz/ATTI9191e2d208f4f3d6f295e6b35d58be4770FC4F15/principal>