



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. Objetos Interactivos
 - i. Trampas
 - ii. Plataformas de un sentido
 - iii. Creación del elevador
 - iv. Creación del ácido
5. Diseño conceptual de niveles (mock-ups)
 - a. Bosque
 - b. Pueblo
 - c. Alcantarilla
6. Implementación de mapa dentro de Unity de niveles
7. Implementación de checkpoints
8. Diseño de interacciones de texto de los NPCs
9. Implementación de NPCs Aliados
10. Implementación de una clase de enemigo
11. Implementación de la historia dentro del juego
12. Implementación de tienda de items
13. Implementación de mejoras (de estadísticas)
 - a. Dulces (Buffs Temporales)
 - b. Comida (Regeneración de Vida)
 - c. Muñecas quitapenas (Buffs permanentes)
14. Creación de escenas para el desarrollo de la historia o como indicadores
15. Implementar *backtracking* dentro del mundo
16. Implementación de puzles a resolver con desbloqueables

17. Diseño conceptual de la base de datos
 - a. Detección del dominio de datos
 - b. Restricciones de integridad
 - c. Análisis de datos relevantes a almacenar
18. Creación de la base de datos en MySQL
19. Inserción de datos del juego a la base de datos
20. Recopilación de información de usuarios desde la página Web
21. Diseñar la página web
22. Modelar una página de inicio
23. Embeber el juego dentro de la página Web
24. Creación de Manual de Usuario dentro de la página Web
25. Mostrar datos en gráficas embebidas en la página Web
26. Montar la Página Web y la base de datos en un servidor en línea
27. Creación de la API
28. Sistemas de dropeo

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
9. Creación de gráficas para el juego
 - a. *Background*
 - b. Animaciones
 - c. Enemigos
 - d. NPCs
 - e. Jugador
10. Creación de Sonidos y Música para el juego
11. Selección de una fuente de texto de unity

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	5 Diseño conceptual de niveles 11 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 existen	
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 NF 9.1 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	10.- Implementación de una clase “enemigo”	
Requerimientos relacionados	13 Implementación de mejoras (de estadísticas) 15 Implementación de backtracking dentro del mundo	
Meta en contexto	Crear enemigos que interactúen en batalla contra el jugador.	
Precondiciones	Sprites de los diferentes enemigos, creación del jugador y de la barra de vida.	
Condiciones de éxito	El enemigo ataca al jugador y le hace una cierta cantidad de daño que se verá reflejado en su barra de salud dependiendo del tipo de adversario que sea (momia, calaverita, charro negro).	
Condiciones de fallo	El enemigo no interactúa con el jugador ni le quita puntos de vida.	
Actor Primario	Motor de Juego de Unity.	
Actor Secundario	Usuario.	
Trigger	El jugador se aproxima.	
Flujo Principal	Paso	Acción
	1	Se crea la clase base de “enemigo”.
	2	El resto de enemigos derivan de esta clase base los comportamientos que deben seguir (movimiento, daño que hacen al jugador, puntos de vida del enemigo, velocidad).
	3	El enemigo entra en contacto con el jugador dependiendo de la posición en la que se encuentre.
	4	El enemigo hace daño al jugador quitándole una cierta cantidad de puntos de vida que se verán reflejados en su barra de salud.

Nombre de caso de uso	4.6.1 Desarrollo de trampas	
Requerimientos relacionados	4.2 Desarrollo del movimiento del personaje 4.3 Desarrollo de Armas 4.6.2 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	4.6.2.- Desarrollo de tablas de presión.	
Requerimientos relacionados	4.2 Desarrollo del movimiento del personaje 4.3 Desarrollo de Armas 4.6.1 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.f.2 Desarrollo de plataformas de un solo sentido dentro del escenario de Unity.	
Requerimientos relacionados	6 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	4.f.3 Creación del elevador	
Requerimientos relacionados	6 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	4.f.4 Creación de ácido.	
Requerimientos relacionados	6 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.- Diseño conceptual de Niveles.
Requerimientos relacionados	5. a - Bosque

	5. b - Pueblo 5. 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	El nivel no es interactivo ni atractivo para el usuario.	
Actor Primario	Administrador.	
Actor Secundario	Ninguno.	
Trigger	Comenzar a dibujar cada uno de los niveles.	
Flujo Principal	Paso	Acción
	1	Plantear la idea de los niveles que se desean tener para el juego.
	2	Plasmar cada una de estas ideas en bocetos preliminares de lo que será cada nivel.
	3	Dibujar la distribución de cada uno de los elementos interactivos/ítems que se encontrarán dentro del nivel.

Nombre de caso de uso	6.- Implementación de mapa dentro de unity
Requerimientos relacionados	5. 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	7.- Implementación de checkpoint	
Requerimientos relacionados	6.- 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	8 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	Diseño de interacciones de texto de los NPCs	
Meta en contexto	Implementar los NPCs (Non Playable Characters) para que interactúen con el usuario dentro del juego y que lo puedan ayudar en el progreso de este.	
Precondiciones	Creación de sprites de los diferentes NPCs: ajolote, mariachis. Asimismo, tener el script de C# que añada la interacción al objeto "NPC".	
Condiciones de éxito	El NPC es capaz de interactuar con el usuario dentro del juego.	
Condiciones de fallo	El NPC no es interactivo y no presenta ninguna conexión con el usuario.	
Actor Primario	Motor de Juego de Unity.	
Actor Secundario	Usuario.	
Trigger	El usuario entra dentro del rango del NPC para interactuar con él.	
Flujo Principal	Paso	Acción
	1	Asignar el script de C# correspondiente al objeto NPC, así como el sprite indicado y su caja de texto.
	2	Agregar al script del comportamiento del jugador lo que tiene que hacer a la hora de entrar en contacto con el tag "NPC".
	3	El jugador presiona "F" para interactuar con el NPC.
	4	El texto del NPC es desplegado de la manera correcta y se da continuidad a la historia.

Nombre de caso de uso	11.- Implementación de la historia dentro del juego
-----------------------	---

Requerimientos relacionados	3 Diseño del concepto del juego	
Meta en contexto	Desarrollar una historia atractiva para los usuarios que pueda ser implementada dentro del motor de juego Unity.	
Precondiciones	Tener un guión base con la idea general de la historia que se desea contar a lo largo del desarrollo del juego.	
Condiciones de éxito	La historia es atractiva para el usuario y se implementa de manera exitosa dentro del motor de juego Unity.	
Condiciones de fallo	La historia no tiene una buena narrativa y no es atractiva para el usuario.	
Actor Primario	Administrador.	
Actor Secundario	Motor de Juego de Unity.	
Trigger	El usuario carga el juego.	
Flujo Principal	Paso	Acción
	1	Crear el guión con la idea base de la historia que quiere que se cuente dentro del juego.
	2	Implementar los niveles dentro del juego que sirvan a la narrativa que se tiene pensada.
	3	El jugador progresa a través del juego y va experimentando la historia contada.

Nombre de caso de uso	15 Implementación de backtracking	
Requerimientos relacionados	16 Implementación de puzzles a resolver con desbloqueables 7 Implementación de checkpoints 6 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 puzzles.	
Precondiciones	Implementación de mapa dentro de Unity junto con puzzles a resolver con desbloqueables.	
Condiciones de éxito	El usuario puede desbloquear las zonas luego de cumplir ciertos requerimientos(puzzles 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	5 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	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	17 Diseño conceptual de la base de datos 18 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	27.- 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 23 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
--	---	--

Nombre de caso de uso	4.3.- Desarrollo de armas	
Requerimientos relacionados	4.3.1 Desarrollo del sistema de cambio de armas 4.3.2 Trompo 4.3.3 Balero 4.3.4 Espada	
Meta en contexto	El usuario puede realizar ataques con alguno de los tres tipos de arma.	
Precondiciones	Creación previa de la clase de jugador y sus características, así como el escenario jugable, barra de salud del jugador y los sprites que se asignarán a cada arma.	
Condiciones de éxito	El personaje puede cambiar entre cada arma (trompo, balero, espada) para poder usarlas en combate y atacar a cada enemigo.	
Condiciones de fallo	El arma no se despliega al momento de equiparla y no sirve para atacar a los enemigos.	
Actor Primario	Usuario	
Actor Secundario	Motor de Juego de Unity	
Trigger	El usuario presiona el clic izquierdo del <i>mouse</i> para poder atacar.	
Flujo Principal	Paso	Acción
	1	Crear un objeto nuevo en la escena principal, el cual debe tener sus componentes correspondientes (<i>box collider 2D</i> , <i>rigidbody 2D</i>).
	2	Crear un script de código en C# con una variable “daño”, la cual indica cuánto daño le hace a los enemigos, así como la detección de que encontró a un enemigo para atacar. Así, este script será asignado a cada tipo de arma.
	3	Crear un script de código en C# que detecte el arma que tiene el jugador equipada y, si se usa el <i>scroll</i> del <i>mouse</i> , que cambia al arma siguiente dentro de la lista de armas.

Nombre de caso de uso	4.6.3.- Desarrollo del elevador	
Requerimientos relacionados	NF 9.1 Background NF 9.2 Animaciones	
Meta en contexto	El usuario puede utilizar una plataforma que funcione como elevador para poder acceder a nuevas áreas dentro del mapa del juego.	
Precondiciones	Creación previa de la clase de personaje, plataforma que funcione como elevador y el mapa jugable.	
Condiciones de éxito	El jugador pisa el ascensor y se eleva para poder acceder a una nueva zona dentro del mapa del juego.	
Condiciones de fallo	El ascensor no se eleva cuando detecta al jugador.	
Actor Primario	Usuario	
Actor Secundario	Motor de Juego de Unity	
Trigger	El usuario Pisa el elevador.	
Flujo Principal	Paso	Acción
	1	Crear un objeto nuevo en la escena principal, el cual debe tener sus componentes correspondientes (<i>box collider 2D, rigidbody 2D</i>).
	2	Crear un script en C# para indicarle al objeto cuando el usuario entre en contacto con este, activando así la animación de “subir” de la plataforma.
	3	Con la animación se modifica la posición tanto del jugador como de la plataforma (elevador).

Nombre de caso de uso	4.f.4.- Desarrollo del ácido	
Requerimientos relacionados	NF 9.a Background NF 9.b Animaciones	
Meta en contexto	El ácido implementado dentro del mapa “elimina” al jugador.	
Precondiciones	Creación previa de la clase de jugador y sus características, así como el escenario jugable, barra de	

	salud del jugador y el sprite que tendrá el ácido.	
Condiciones de éxito	La barra de vida del jugador baja a cero cuando este entra en contacto con el ácido.	
Condiciones de fallo	El ácido no funciona de la manera correcta, ya sea, funciona como plataforma, no baja los puntos de vida, etc.	
Actor Primario	Usuario	
Actor Secundario	Motor de Juego de Unity	
Trigger	El usuario entra en contacto con el ácido.	
Flujo Principal	Paso	Acción
	1	Crear un objeto tipo <i>tilemap</i> en la escena principal e insertar los componentes <i>compose collider 2D</i> y <i>tilemap collider 2D</i> .
	2	Crear un script en C# dentro del cual se detecte que el jugador entró en contacto con el objeto (ácido) , haciendo que la variable de “vida” dentro de la barra del salud del jugador sea 0.
	3	Crear un <i>rule tile</i> para indicar en qué <i>tile</i> hay una animación de movimiento, pudiendo así emplearlo en un <i>tile palette</i> .

Nombre de caso de uso	7 Implementación de checkpoints
Requerimientos relacionados	20 Inserción de datos del juego a la base de datos 28 Creación de la API 8 Diseño de interacciones de texto de los NPCs
Meta en contexto	El usuario al entrar en contacto con un <i>checkpoint</i> , es capaz de guardar el progreso que lleva recorrido dentro del juego.
Precondiciones	Creación previa de la clase de jugador y sus características, así como el escenario jugable y el sprite que tendrá el <i>checkpoint</i> .
Condiciones de éxito	El jugador es capaz de guardar su progreso que lleva en el juego a través de los <i>checkpoints</i> .
Condiciones de fallo	El <i>checkpoint</i> no es capaz de guardar el progreso del jugador.

Actor Primario	Usuario	
Actor Secundario	Motor de Juego de Unity, DataBase Management System	
Trigger	El usuario presiona la tecla <i>F</i> al estar en el rango del <i>checkpoint</i> .	
Flujo Principal	Paso	Acción
	1	Creación de scripts en C# para poder hacer las funciones correspondientes (<i>queries</i> , <i>gets</i> , <i>update</i>), dentro de la base de datos que está conectada al API.
	2	Crear el objeto de “checkpoint” dentro del mundo y agregar la funcionalidad de guardar la posición del jugador dentro del mundo del juego, así como el arma que se tiene equipada, ítems en inventario.
	3	Almacenar los elementos actualizados a través del <i>checkpoint</i> dentro de la base de datos.
	4	Desplegar una caja de texto en la que se indique que su progreso dentro del juego fue guardado de manera exitosa.

Nombre de caso de uso	12 Implementación de tienda de ítems
Requerimientos relacionados	20 inserción de datos del juego a la base de datos 28 Creación de la API 8 Diseño de interacciones de texto de los NPCs{
Meta en contexto	El usuario puede comprar ítems que le ayudan a su progreso en la historia del juego.
Precondiciones	Creación previa de la clase de jugador y sus características, así como el escenario jugable, los sprites correspondientes a cada elemento que se venderá dentro de la tienda, contador de monedas y la interfaz de usuario.
Condiciones de éxito	El jugador puede acceder a la tienda y emplear las monedas conseguidas a lo largo del juego para poder comprar diferentes ítems que le ayuden a progresar en la historia.
Condiciones de fallo	El jugador no es capaz de interactuar con la tienda y, por

	lo tanto, no puede comprar nada extra que le ayude a progresar dentro del juego.	
Actor Primario	Usuario	
Actor Secundario	Motor de Juego de Unity, DataBase Management System	
Trigger	El usuario presiona la tecla <i>F</i> al entrar en el rango de la tienda.	
Flujo Principal	Paso	Acción
	1	Creación de scripts en C# para poder hacer las funciones correspondientes (<i>queries</i> , <i>gets</i> , <i>update</i>), dentro de la base de datos que está conectada al API.
	2	Creación de la tienda como objeto interactivo dentro del mapa del juego.
	3	Creación de script en C# dentro del cual se detecte que el jugador presionó la <i>F</i> dentro del rango de la tienda, desplegando así su interfaz con los items en venta.
	4	Verificación del número de monedas que tiene el usuario cuando quiera comprar algo, sabiendo así si tiene la cantidad suficiente para obtenerlo.
	5	Añadir al inventario del jugador el ítem que adquirió y guardarlo también dentro de la base de datos.

Nombre de caso de uso	13 Implementación de mejoras
Requerimientos relacionados	4.4 Desarrollo de la interfaz de usuario 12 Implementación de tienda de items
Meta en contexto	El usuario puede consumir objetos que le dan ventajas a su personaje, como aumentar sus puntos de salud o poder infligir más daño a los enemigos.
Precondiciones	Creación previa de la clase de jugador y sus características, así como el escenario jugable y los sprites correspondientes a los diferentes objetos.
Condiciones de éxito	El jugador consume el objeto y se le aplica cierta mejora

	(dependiendo del ítem consumido).	
Condiciones de fallo	El usuario no sufre ninguna modificación en sus estadísticas a la hora de consumir el objeto.	
Actor Primario	Usuario	
Actor Secundario	Motor de Juego de Unity, Sistema gestor de la base de datos.	
Trigger	El usuario presiona la tecla <i>E</i>	
Flujo Principal	Paso	Acción
	1	Creación de un script en C# que añada la mejora correspondiente a cada uno de los objetos.
	2	Al momento de consumir el objeto, se actualiza la estadística correspondiente del jugador dependiendo del ítem consumido (al presionar la tecla <i>E</i> con el objeto equipado).
	3	Actualizar dentro de la base de datos el valor de la estadística del jugador modificada.

Nombre de caso de uso	17 Diseño conceptual de la base de datos	
Requerimientos relacionados	3 Concepto del juego	
Meta en contexto	Tener un esquema del modelo entidad y relacional dentro del contexto del juego.	
Precondiciones	Ninguna	
Condiciones de éxito	El esquema relacional cumple las 3 formas normales.	
Condiciones de fallo	No hay restricciones de integridad	
Actor Primario	El administrador	
Actor Secundario	Ninguno	
Trigger	El desarrollo de los diagramas en draw.io	
Flujo Principal	Paso	Acción

	1	Análisis de los posibles datos presentes del juego y de los usuarios.
	2	Ver los datos que serán llaves primarias y foráneas.
	3	Realizar el esquema relacional con las entidades y relaciones identificadas conforme a las primera 3 forma normales.

Nombre de caso de uso	18 Creación de la base de datos en MySQL	
Requerimientos relacionados	3 Concepto del juego 7 Implementación de checkpoint	
Meta en contexto	Poder almacenar y consultar datos del juego y del usuario para ser analizados.	
Precondiciones	17 Diseño conceptual de la base de datos	
Condiciones de éxito	Los scripts son ejecutados correctamente.	
Condiciones de fallo	Los scripts presentan errores.	
Actor Primario	DataBase Management System	
Actor Secundario	Administrador	
Trigger	Al abrir el sistema gesto de la base de datos	
Flujo Principal	Paso	Acción
	1	La creación de un script de tipo sql en donde se crean tablas del esquema relacional.
		Declaración de los motores de almacenamiento, formato de codificación de caracteres Unicode.
	2	En un script separado se crean las inserciones de datos “dummy” para verificar que las tablas aceptan datos adecuados.

Nombre de caso de uso	28 Creación de la API
Requerimientos relacionados	23 Embeber el juego dentro de la página WEB

	12 Implementación de tienda de items 13 Implementación de mejoras 7 Implementación de checkpoints	
Meta en contexto	Establecer la comunicación entre sitio web, la base de datos y el juego.	
Precondiciones	tener el prototipo funcional del juego así como la versión final de la base de datos. Asimismo, se requiere de la instalación de node y npm dentro de los equipos de trabajo.	
Condiciones de éxito	La API envía y recibe información dentro de las distintas áreas de trabajo (web, unity, base de datos).	
Condiciones de fallo	La API no puede establecer comunicación.	
Actor Primario	Motor de juego de Unity, Página Web y DataBase management System	
Actor Secundario	Administrador	
Trigger	Al utilizar javascript y node para la creación de API	
Flujo Principal	Paso	Acción
	1	Crear un proyecto de node module.
	2	Crea el archivo de javascript con el que se establece la conexión entre la base de datos y Unity, así como el puerto de comunicación.
	3	Creación de los diferentes JS para la implementación de endpoints con el fin de poder realizar los queries respectivos.
	4	En el sistema gestor de la base de datos crear un usuario que pueda acceder a ciertas características de estas, como las vistas, modificar valores, etc... Asimismo, se establecen la ip y el puerto de comunicación.

Nombre de caso de uso	29 Sistema de <i>dropeo</i>
Requerimientos relacionados	27 Creación de la API 18 Creación de la base de datos en MySQL 10 Implementación de una clase de enemigo

	13 implementación de mejoras	
Meta en contexto	El usuario, al eliminar un enemigo, recibe un objeto aleatorio consumible.	
Precondiciones	Creación previa de la clase de jugador y sus características, así como el escenario jugable, la clase de enemigo y la base de datos implementada.	
Condiciones de éxito	El jugador obtiene un ítem consumibles tras eliminar un enemigo.	
Condiciones de fallo	El usuario tras eliminar a un enemigo , no obtenga ningún ítem dropeado por él.	
Actor Primario	Usuario	
Actor Secundario	Motor de Juego de Unity y sistema gestor de la base de datos	
Trigger	Al eliminar un enemigo	
Flujo Principal	Paso	Acción
	1	Utilizando una corrutina, llamar a las funciones correspondientes (<i>querys, gets, update</i>) para trabajar con la tabla llamada <i>loot table</i> , dentro de la cual se genera un número aleatorio para elegir un elemento dentro de la misma.
	2	El elemento elegido de manera aleatoria de la tabla se despliega dentro del juego a la hora de que el enemigo muere, permitiendo así que el jugador agarre el objeto en cuestión.

Historias de Usuario (12)

Como un jugador del personaje...

Juego

Como un jugador, quiero poder utilizar diferentes armas que me sirvan para pelear contra diferentes enemigos y jefes.

Juego

Como jugador quiero que la asignación de botones tenga un sentido de ser, al igual que los periféricos que haya sean con propósito y que sean consistentes, si es que no es así es mejor que no haya ninguno para poder un gamepad o hacerlo versátil.

Juego

Como jugador quiero una sensación de término, un resultado. Al momento de volverlo a jugar que pueda tener diferentes finales.

+ Añada una tarjeta

Product Backlog (6) [50f; 17nf]

Web

22.-Diseñar la página web

0/2

Web

23.-Modelar una página de inicio

Web

Juego

24.-Embeber el juego dentro de la página Web

Web

25.-Creación de Manual de Usuario dentro de la página Web

Base de Datos

Web

26.-Mostrar datos en gráficas embebidas en la página Web

+ 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 Frías Pina (13 Mayo - 19 Mayo)

Sprint 3- Alan (20 Mayo - 26 Mayo)

Sprint 4- Alex (27 Mayo - 2 Junio)

Sprint 5- Alex^2 (5 Junio - 14 Junio)

+ Añada una tarjeta

Pendiente

Base de Datos

General

Web

27.-Montar la Página Web y la base de datos en un servidor en línea

Juego

29.- Sistema de dropeo

Web

Juego

24.- Embeber el juego dentro de la página Web

+ Añada una tarjeta

En Revision

+ Añada una tarjeta

Terminados

2

AH

Sprint 4- Andrea Alexandra Barrón Córdova (27 Mayo - 2 Junio)

Juego

4,5.-Desarrollo de los menús

3/3

OM

Web

22.-Diseñar la página web

1

0/2

AH

Web

23.-Modelar una página de inicio

1

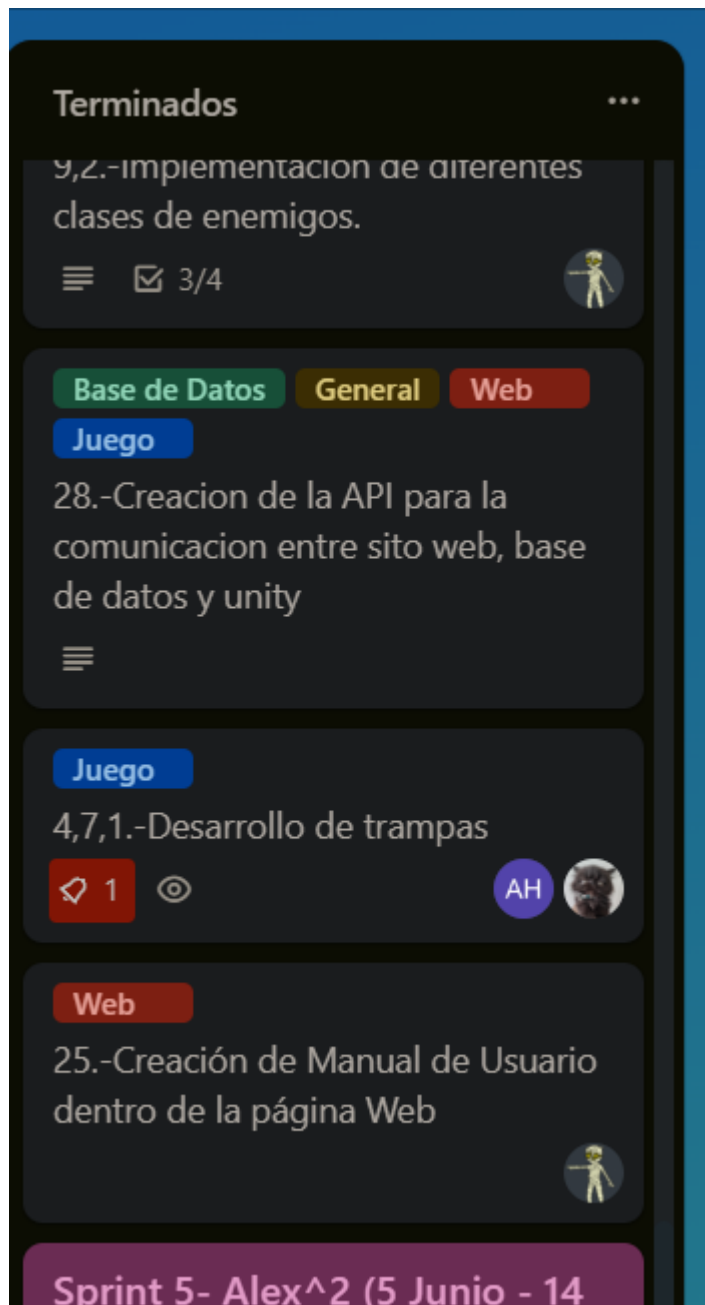
AH

Base de Datos

18.-Creación de la base de datos en MySQL

MP

+ Añada una tarjeta



Enlace del trello:

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