



**Tecnológico  
de Monterrey**

## **Construcción de Software y toma de decisiones**

TC2005B.400

### **Especificación de Requerimientos de Software**

28 de marzo 2022

Diego Araque | A01026037

Alvaro Garcia | A01781511

Salomon Dabbah | A01028445

Marco Torres | A01025334

1. Obtenga una lista de historias de usuario del cliente o dueño del producto (product owner) siguiendo la metodología de SCRUM.

[https://miro.com/welcomeonboard/dmpuOGIXbW0yRDVQYTNHVnRFbVRPclRCSE9mTm5qQ09kTndUU2tQVFNiWFVPcEJ5cGY1Um4zNFBSUThYTUZ1V3wzNDU4NzY0NTIyMDg3NjEwMzY2?invite\\_link\\_id=90453309845](https://miro.com/welcomeonboard/dmpuOGIXbW0yRDVQYTNHVnRFbVRPclRCSE9mTm5qQ09kTndUU2tQVFNiWFVPcEJ5cGY1Um4zNFBSUThYTUZ1V3wzNDU4NzY0NTIyMDg3NjEwMzY2?invite_link_id=90453309845)



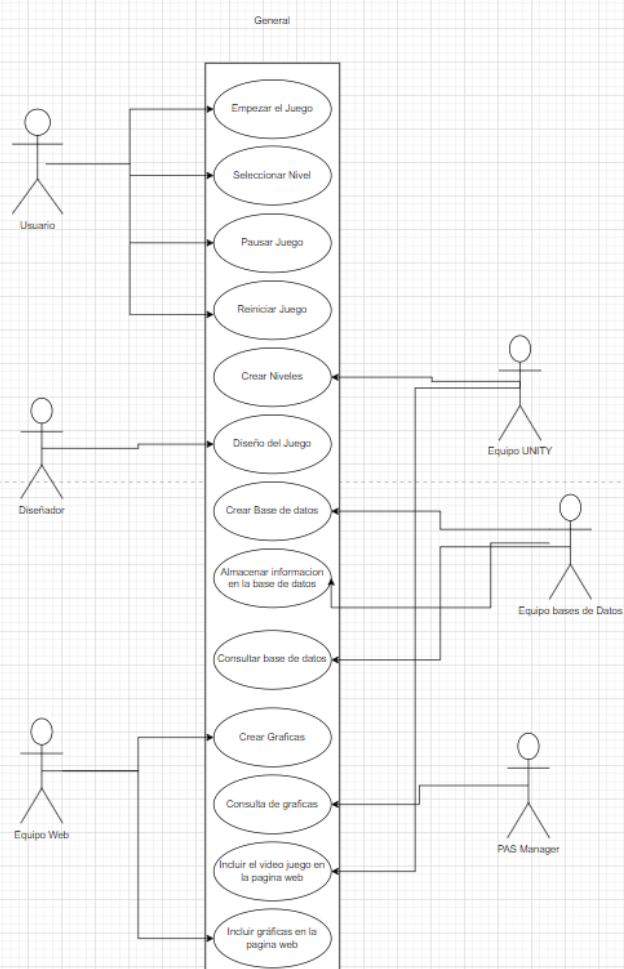
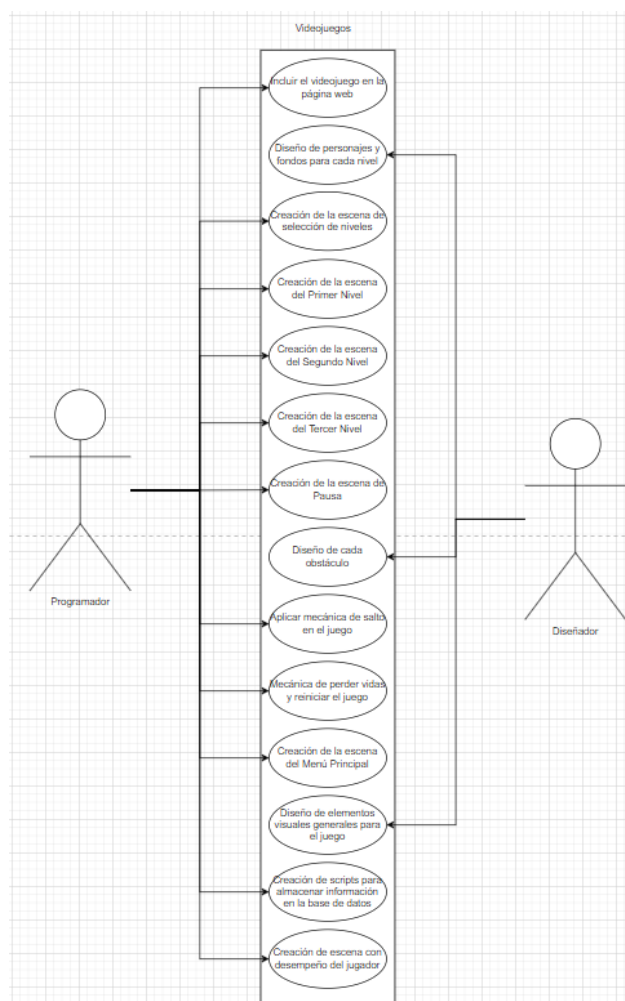
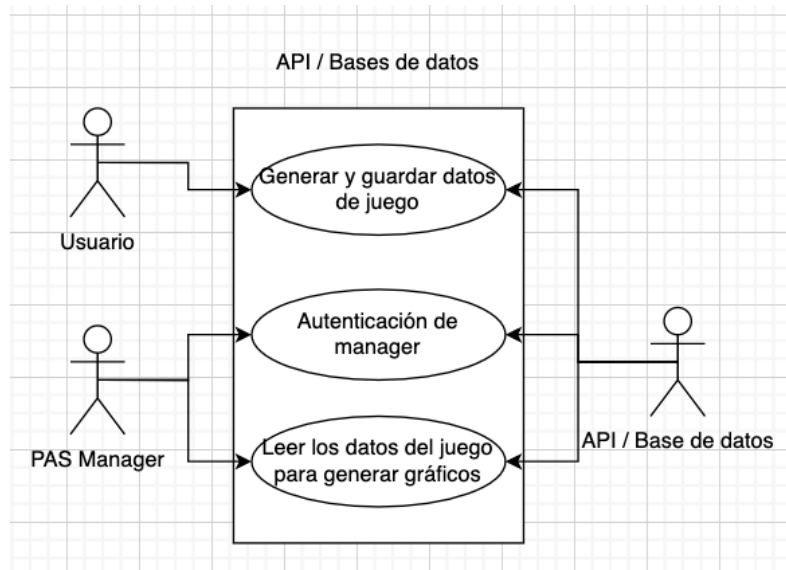
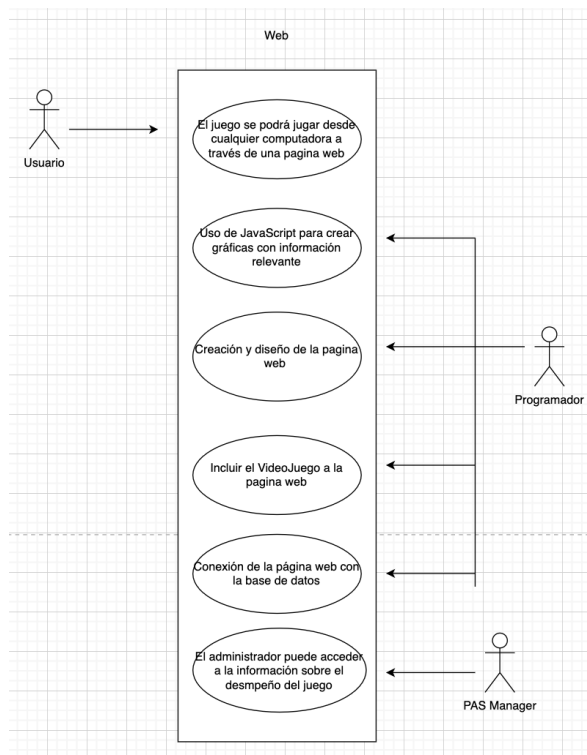
2. De la lista anterior, filtre los requerimientos funcionales y plásmelos en la lista de tareas principales de SCRUM (product backlog).

<https://trello.com/b/1o5HLABF/sprint-1>



3. Cree un diagrama de casos de uso de UML que muestre a los actores y casos asociados al sistema propuesto para el socio formador.

[https://app.diagrams.net/?src=about#G1CfA4G5ha25ergAAZQHEdRO8e\\_mydx6E0](https://app.diagrams.net/?src=about#G1CfA4G5ha25ergAAZQHEdRO8e_mydx6E0)



4. Para cada caso de uso (óvalos en el diagrama) desarrolle una tabla descriptiva como las vistas en clase, donde se haga énfasis en los actores, disparadores del caso y flujo de ejecución.

Use case name	Creación de la escena del primer nivel
Related Requirements	Creación de la escena del segundo nivel Creación de la escena del tercer nivel Aplicar mecánica de salto en el juego Mecánica de perder vidas y reiniciar el juego Diseño de elementos visuales generales para el juego Diseño de cada obstáculo
Goal in Context	Crear el primer nivel del juego con un buen diseño y mecánicas
Preconditions	Tener los diseños de elementos visuales listos al igual que los obstáculos
Successful End Condition	Un buen nivel que funciona y es divertido
Failed End Condition	Un nivel con mal diseño y funcionalidad
Primary Actors	Desarrolladores
Secondary Actors	Diseñadores
Trigger	El jugador elige el primer nivel del juego

Main Flow	
Step	Action
1	Fijar el fondo y la cámara del juego
2	Añadir los sprites (personaje, bloques y obstáculos)
3	Asignarle a cada sprite un rigidbody y designar capsule colliders
4	Crear los siguientes scripts: <ul style="list-style-type: none"> <li>- Character controller</li> <li>- Player movement</li> </ul>
5	Añadir la música al juego
6	Hacer pruebas

Use case name	Creación de la escena del segundo nivel
Related Requirements	Creación de la escena del primer nivel Creación de la escena del tercer nivel Aplicar mecánica de salto en el juego Mecánica de perder vidas y reiniciar el juego Diseño de elementos visuales generales para el juego Diseño de cada obstáculo
Goal in Context	Crear el segundo nivel del juego con un buen diseño y mecánicas
Preconditions	Tener los diseños de elementos visuales listos al igual que los obstáculos Completar la escena y los scripts del primer nivel
Successful End Condition	Un buen nivel que funciona y es divertido
Failed End Condition	Un nivel con mal diseño y funcionalidad
Primary Actors	Desarrolladores
Secondary Actors	Diseñadores
Trigger	El jugador elige el segundo nivel del juego

Main Flow	
Step	Action
1	Fijar el fondo y la cámara del juego
2	Añadir los sprites (personaje, bloques y obstáculos)
3	Asignarle a cada sprite un rigidbody y designar capsule colliders
4	Asignar los siguientes scripts: <ul style="list-style-type: none"> <li>- Character controller</li> <li>- Player movement</li> </ul>
5	Añadir la música al juego
6	Hacer pruebas

Use case name	Creación de la escena del tercer nivel
Related Requirements	Creación de la escena del primer nivel Creación de la escena del segundo nivel Aplicar mecánica de salto en el juego Mecánica de perder vidas y reiniciar el juego Diseño de elementos visuales generales para el juego Diseño de cada obstáculo
Goal in Context	Crear el tercer nivel del juego con un buen diseño y mecánicas
Preconditions	Tener los diseños de elementos visuales listos al igual que los obstáculos Completar la escena y los scripts del primer nivel
Successful End Condition	Un buen nivel que funciona y es divertido
Failed End Condition	Un nivel con mal diseño y funcionalidad
Primary Actors	Desarrolladores
Secondary Actors	Diseñadores
Trigger	El jugador elige el tercer nivel del juego

Main Flow	
Step	Action
1	Fijar el fondo y la cámara del juego
2	Añadir los sprites (personaje, bloques y obstáculos)
3	Asignarle a cada sprite un rigidbody y designar capsule colliders
4	Asignar los siguientes scripts: <ul style="list-style-type: none"> <li>- Character controller</li> <li>- Player movement</li> </ul>
5	Añadir la música al juego
6	Hacer pruebas

Use case name	Diseño de cada obstáculo
Related Requirements	Diseño de elementos visuales generales para el juego
Goal in Context	Crear los obstáculos y los bloques con notas para que el jugador pierda vidas cuando toque los obstáculos
Preconditions	Tener el fondo de cada uno de los niveles listos para agregarlos al juego así como el piso del mismo
Successful End Condition	
Failed End Condition	
Primary Actors	Diseñadores
Secondary Actors	Desarrolladores
Trigger	Cuando el jugador toca un obstáculo pierde una vida

Main Flow	
Step	Action
1	Diseñar los bloques cada uno con su respectiva nota y tamaño
2	Asignarle a cada sprite un rigidbody y designar capsule colliders
3	Hacer el script para que cuando se toque un obstáculo se pierda una vida
4	Hacer pruebas

Main Flow	
Step	Action
1	Crear un endpoint en la base de datos para recibir los datos dependiendo de los parámetros que se pasen.
2	Crear una función en el cliente que obtenga los datos y que se ejecute cuando
3	Hacer pruebas.

Use case name	Mecánica de perder vidas y reiniciar el juego
Related Requirements	<p>Aplicar mecánica de salto en el juego</p> <p>Creación de variables y scripts para almacenar información relevante del juego en la base de datos.</p> <p>Diseño de cada obstáculo</p> <p>Diseño de la escena del primer nivel</p> <p>Diseño de la escena del segundo nivel</p> <p>Diseño de la escena del tercer nivel</p>
Goal in Context	Poder darle el funcionamiento esperado a cada nivel del juego
Preconditions	Tener los obstáculos y la mecánica de salto en el juego
Successful End Condition	El jugador tiene 5 vidas y pierde una cada vez que toca un obstáculo. Si pierde todas las vidas pierde el juego y este puede ser reiniciado
Failed End Condition	El jugador no pierde vidas cuando toca un obstáculo y el nivel no finaliza/reinicia cuando pierde todas las vidas
Primary Actors	Desarrolladores
Secondary Actors	-no hay actores secundarios
Trigger	Cuando el jugador toca un obstáculo

Main Flow
-----------



Step	Action
1	Creación del script de pérdida de vidas
2	Creación del script de finalización/reiniciación de nivel
3	Integración de los scripts en el juego
4	Hacer pruebas

Use case name	Creación de la escena del menú principal
Related Requirements	Uso de unity para la creación del videojuego Diseño de elementos visuales generales para el juego Creación de la escena de selección de niveles
Goal in Context	Mostrar al usuario que ya se encuentra en el juego y darle la opción de ir a la pantalla de selección de niveles
Preconditions	Tener unity instalado. Tener algunos assets para el menú y tener TextMesh Pro instalado en Unity (diseño de texto).
Successful End Condition	El menú se ve estético y tiene un botón que redirecciona a la escena de selección de niveles.
Failed End Condition	No funciona el botón de “Play” o el menú no es estético
Primary Actors	Desarrolladores
Secondary Actors	Diseñadores
Trigger	Cuando el jugador toca el botón de “Play”

Main Flow	
Step	Action
1	Creación de una nueva escena
2	Estilización de la escena

3	Programación e integración del botón de cambio de escena
4	Hacer pruebas

Use case name	Creación de la escena de pausa
Related Requirements	Uso de unity para la creación del videojuego Diseño de elementos visuales generales para el juego Diseño de la escena del primer nivel Diseño de la escena del segundo nivel Diseño de la escena del tercer nivel
Goal in Context	Poder pausar y resumir el juego en cualquier punto de un nivel
Preconditions	Tener las escenas de los tres niveles donde al menos esté sucediendo un evento que se pueda pausar
Successful End Condition	El jugador selecciona el botón de pausa que lo lleva a otra escena, donde puede resumir el juego donde se quedó.
Failed End Condition	El botón de pausa no redirecciona al menú de pausa. El juego no se pausa correctamente. El juego no se resume correctamente
Primary Actors	Desarrolladores
Secondary Actors	Diseñadores
Trigger	Cuando el jugador selecciona el botón de pausa o de continuar juego

Main Flow	
Step	Action
1	Añadir el botón de pausa en los niveles y crear el script para las funciones de pausa del juego
2	Crear y diseñar la escena de pausa

3	Integrar el script con el juego
4	Hacer pruebas

Use case name	Mecánica de salto
Related Requirements	Uso de unity para la creación del videojuego Diseño de los personajes Mecánica de perder vidas Diseño de cada obstáculo Diseño de la escena del primer nivel Diseño de la escena del segundo nivel Diseño de la escena del tercer nivel
Goal in Context	Poder cumplir el objetivo general del juego, que el usuario salta con cada nota
Preconditions	Tener las escenas de los tres niveles, el personaje y los obstáculos
Successful End Condition	El jugador presiona espacio y el personaje saltara. Si se presiona por más tiempo, el salto estará prolongado.
Failed End Condition	La tecla de espacio no está funcionando o al presionarlo por más tiempo, el salto es igual
Primary Actors	Desarrolladores
Secondary Actors	Diseñadores
Trigger	Cuando el jugador presiona la tecla de espacio, el jugador salta

Main Flow	
Step	Action
1	Crear el script en donde se especifique lo que debe pasar cuando se presiona espacio.
2	Jugar con las velocidades y tiempos
3	Integrar el script con el juego
4	Hacer pruebas

Use case name	Creación de la escena que aparece al completar un nivel, que tenga información sobre el desempeño del jugador.
Related Requirements	Creación de la escena del primer nivel Creación de la escena del segundo nivel Creación de la escena del tercer nivel Aplicar mecánica de salto en el juego Mecánica de perder vidas y reiniciar el juego Diseño de elementos visuales generales para el juego Diseño de cada obstáculo
Goal in Context	Display del score al terminar el nivel
Preconditions	Algún nivel que ya este funcional para comprobar que si se genera el resultado correcto
Successful End Condition	Display del score
Failed End Condition	No te direcciona a una nueva escena con el score
Primary Actors	Desarrolladores
Secondary Actors	Diseñadores
Trigger	El jugador termina el nivel o pierde

Main Flow	
Step	Action
1	Crear la escena en donde se encuentre toda la información necesaria
2	Crear un script que al terminar el juego o perder todas las vidas te direcciona a esta escena
3	Integrar el script con el juego
4	Hacer pruebas

Use case name	Diseño de elementos visuales generales para el juego.
Related Requirements	Diseño de cada obstáculo Diseño de la escena del primer nivel Diseño de la escena del segundo nivel Diseño de la escena del tercer nivel
Goal in Context	Crear un buen diseño para los elementos de juego y el mismo juego.
Preconditions	Saber las funcionalidades del juego Saber los diferentes elementos que debe de tener el juego.
Successful End Condition	Buen diseño visual del juego.
Failed End Condition	Diseño atascado y no balanceado del juego con elementos no llamativos.
Primary Actors	Diseñadores
Secondary Actors	Desarrolladores
Trigger	

Main Flow	
Step	Action
1	Definir cuáles son los elementos que habrá en el juego.
2	Hacer el diseño de cada elemento.
3	Hacer pruebas de cómo se verán todos los elementos juntos.

Use case name	Diseño de personajes y fondos para cada nivel.
Related Requirements	Diseño de cada obstáculo Diseño de elementos visuales generales para el juego. Diseño de la escena del primer nivel Diseño de la escena del segundo nivel Diseño de la escena del tercer nivel

Goal in Context	Crear un buen diseño para los personajes del juego y los fondos.
Preconditions	Sabar los diferentes niveles que debe de tener el juego.
Successful End Condition	Buen diseño visual de los personajes y fondos del juego.
Failed End Condition	Diseño atascado y no balanceado del juego con personajes y fondos no llamativos.
Primary Actors	Diseñadores
Secondary Actors	Desarrolladores
Trigger	

Main Flow	
Step	Action
1	Definir cuáles son los niveles que habrá en el juego.
2	Hacer el diseño o buscar los personajes y fondos.
3	Hacer pruebas de cómo se verán .

Use case name	Creación de la base de datos
Related Requirements	Crear el esquema de entidad relación Normalizar el esquema de entidad relación
Goal in Context	Almacenar información importante del juego
Preconditions	Tener un esquema de entidad relación totalmente normalizado
Successful End Condition	Base de datos que almacena datos importantes para los managers
Failed End Condition	No se presenta información relevante en la base de datos
Primary Actors	Desarrolladores

Secondary Actors	MySQL
Trigger	Comandos en MySQL Workbench

Main Flow	
Step	Action
1	Tener una idea de que datos se quieren almacenar
2	Crear el esquema de entidad relación
3	Normalizar a la tercera forma normal
4	Con comandos de MySQL crear lo establecido en los pasos anteriores

Use case name	Crear esquema de entidad relación
Related Requirements	Normalizar esquema de entidad relación Crear base de datos
Goal in Context	Tener en un esquema todos los atributos que vamos a requerir
Preconditions	Tener una idea general de que analisis se quiere hacer
Successful End Condition	Tener un esquema de entidad relación en UML 2.0
Failed End Condition	No tener el esquema en UML
Primary Actors	Desarrolladores
Secondary Actors	Herramientas de modelado
Trigger	Selección de distintas tablas y relaciones para la creación del esquema

Main Flow	
Step	Action
1	Tener idea general de que se quiere almacenar y analizar

2	Usar una app de modelado
3	Crear el esquema
4	Revisarlo entre todos los desarrolladores

Use case name	Normalizar el esquema de entidad relación
Related Requirements	Crear esquema de entidad relación Crear base de datos
Goal in Context	Que todas nuestras tablas carezcan de información redundante y malas prácticas
Preconditions	Que el esquema de entidad relación ya exista
Successful End Condition	Un esquema sin redundancia y hecho con buenas prácticas
Failed End Condition	Un esquema con redundancia y hecho con malas prácticas
Primary Actors	Desarrolladores
Secondary Actors	Herramientas de modelado
Trigger	Detección de redundancia, falta de llaves primarias, dependencias transitivas y funcionales, entre muchas otras.

Main Flow	
Step	Action
1	Esquema de entidad relación ya creado
2	Búsqueda de malas prácticas
3	Corrección de las malas prácticas
4	Revisarlo al final para comprobar que todo esté en perfectas condiciones

Use case name	Normalizar el esquema de entidad relación
Related Requirements	Crear esquema de entidad relación



	Crear base de datos
Goal in Context	Que todas nuestras tablas carezcan de información redundante y malas prácticas
Preconditions	Que el esquema de entidad relación ya exista
Successful End Condition	Un esquema sin redundancia y hecho con buenas prácticas
Failed End Condition	Un esquema con redundancia y hecho con malas prácticas
Primary Actors	Desarrolladores
Secondary Actors	Herramientas de modelado
Trigger	Detección de redundancia, falta de llaves primarias, dependencias transitivas y funcionales, entre muchas otras.

Main Flow	
Step	Action
1	Esquema de entidad relación ya creado
2	Búsqueda de malas prácticas
3	Corrección de las malas prácticas
4	Revisarlo al final para comprobar que todo esté en perfectas condiciones

Use case name	Generar y guardar datos del juego
Related Requirements	Creación de la Base de Datos en MySQL Conectar la base de datos con la API Creación de variables y scripts para almacenar información relevante del juego en la base de datos
Goal in Context	Hacer que los jugadores puedan generar datos de sus partidas como los lugares donde fallan, o en número de intentos para

	pasar un nivel, y almacenar esos datos en la base de datos.
Preconditions	Tener una base de datos conectada a una API.
Successful End Condition	Se guardan los datos necesarios en las partidas de los jugadores.
Failed End Condition	No se guardan los datos en la base de datos o se guardan datos erróneos
Primary Actors	Desarrolladores
Secondary Actors	
Trigger	Cuando un jugador realiza una acción que necesite ser almacenada se guardará en un array local y cuando el jugador terminé la partida se enviarán todos los datos a la api y se almacenarán en la base de datos.

Main Flow	
Step	Action
1	Definir qué eventos del juego son los que generarán datos.
2	Definir los posibles valores que puede generar cada evento
3	Programar el almacenamiento de los datos localmente.
4	Programar el envío a la api de los datos almacenados una vez que se concluya la partida.

Use case name	Autenticación de Manager
Related Requirements	Creación de la Base de Datos en MySQL Conectar la base de datos con la API
Goal in Context	Hacer que los managers puedan acceder por medio de autenticación a la sección de gráficas y estadísticas.

Preconditions	Tener una base de datos conectada a una API.
Successful End Condition	Los managers pueden crear una cuenta y se pueden autenticar con ella.
Failed End Condition	La autenticación o creación de cuentas falla.
Primary Actors	Desarrolladores
Secondary Actors	
Trigger	Cuando un manager quiere acceder a la sección de gráficas y estadísticas

Main Flow	
Step	Action
1	Crear un endpoint en la base de datos para la creación de nueva cuenta de manager.
2	Diseñar en el UI el proceso de sign up o log in.
3	Hacer pruebas.

Use case name	Leer los datos del juego para generar gráficos
Related Requirements	Creación de la Base de Datos en MySQL Conectar la base de datos con la API Creación de variables y scripts para almacenar información relevante del juego en la base de datos
Goal in Context	Poder enviar los datos almacenados en la base de datos al cliente para poder generar las gráficas correspondientes.
Preconditions	Tener una base de datos conectada a una API.
Successful End Condition	Se envían los datos de la base de datos al cliente de forma ordenada y limitada.

Failed End Condition	No se envían los datos.
Primary Actors	Desarrolladores
Secondary Actors	
Trigger	Cuando se accede a la sección de gráficas y estadísticas.

Use case name	El administrador puede acceder a la información sobre el desempeño del juego
Related Requirements	Creación de la Base de Datos en MySQL Conectar la base de datos con la API Conexión de la página web con la base de datos Creación y diseño de la página web
Goal in Context	Que los administradores puedan acceder a información relevante sobre el desempeño de los usuarios al jugar.
Preconditions	Juego terminado que envía información sobre el jugador a la base de datos y se conecta a la página web.
Successful End Condition	Los administradores pueden visualizar la información del juego correctamente
Failed End Condition	Los administradores no pueden visualizar la información del juego correctamente
Primary Actors	Desarrolladores
Secondary Actors	-
Trigger	El administrador inicia sesión para ver los datos del juego

Main Flow	
Step	Action
1	Crear sistema de autenticación.
2	Registrar una cuenta para el administrador e integrar el código necesario para que se muestre la información del juego al iniciar sesión.

- <https://trello.com/b/1o5HLABF/sprint-1>



Sprint Backlog

Requerimiento Funcional

Videojuegos

Creación de la escena del Primer Nivel. Historia de Usuario #22

Creación de Niveles

2

DF

MS

SD

Requerimiento Funcional

Videojuegos

Creación de la escena de Pausa. Historia de Usuario #2 Pausar el Juego

DF

Requerimiento Funcional

Videojuegos

Diseño de cada obstáculo. Historia de Usuario #11 Diseño del Juego

MS

Requerimiento Funcional

Videojuegos

Búsqueda de 3 canciones y sus respectivas partituras. Historias de Usuario #11 Diseño del Juego

+ Add a card