

Grupo 6 - Especificación de Requerimientos de Software

Introducción

Todo lo que se mostrará en este documento se puede ver en el siguiente link de Trello:

- Trello: <https://trello.com/b/bXUPj0Aa/sprint-1-grupo-6>

Historias de Usuario

Octavio

El proyecto se compone de 4 partes principales: el juego, la página web, la base de datos (B.D.s), y la conexión entre las anteriores:

- Página web:
 - Se espera que desarrollen una página que muestre la información general del juego.
 - El juego debe estar funcionando en la página web, y debe haber otra más que muestre los datos del juego que se recolectaron mediante gráficas (eso lo veremos más adelante).
- Bases de datos:
 - Debe registrar los datos básicos para que el juego funcione; así como los datos que van almacenándose conforme el jugador avanza en el juego.
- Videojuego:
 - Se espera que sea un RPG
 - Con una mecánica de juego que fomente una toma estratégica de decisiones.

- Parte de los datos que se tienen que almacenar (en las B.D.s) tienen que ver con esas decisiones que se van tomando.
- Conexión entre éstas:
 - Se deben diseñar APIs que sean medios de comunicación entre las tres partes del proyecto.

Esteban Castillo

- Yo como cliente, me gustaría tener la info. distribuida en tablas; sin datos faltantes, ni duplicados.
 - Columnas: B.D., Web, Videojuego.
- Req. Funcional - Guardar nombre de jugador, poderes... y el tipo de dato. Los datos a almacenar, y el tipo.
- Ya que se sabe qué datos existen, crear un modelo Entidad-Relación. en el cual se haga una “3ra forma normal”. Crear el modelo con UML. En un inicio no hay que crear las tablas, pero sí tener claro las definiciones conceptuales.
- Ya con el modelo, crear las tablas en MySQL con código. Agregando restricciones de integridad (como llaves primarias).
- Llenar las tablas con info. genérica (dummy data) para probarlas, y darse una idea. Así, lo que sigue es empezar a definir las consultas del front/back-end.
- Modelo conceptual - Aquellos programas y datos en una laptop, que operan los dispositivos.

Gilberto

- Poder tener varios perfiles de juego, e iniciar sesión con ellos.
- Poder jugar el juego durante una "sesión" y poder almacenar el progreso en el juego.

- Poder revisar en la página web estadísticas sobre información del juego que me permitan formar mejores estrategias.

Socio formador

Personalmente mi enfoque es más hacia la parte de la experiencia de usuario, es decir, que tu juego sea algo que "se divierta de jugar" y que "proporcione un reto al jugador así como hacer válidas sus decisiones y que tengan algún efecto en su experiencia" además de que sea algo que pueda ser "rejugable, es decir, que mi primera vez jugando pueda ser diferente a la segunda en base a mis decisiones".

Si tuviera que decir algo sobre bases de datos te sugiero save files, puesto que al tener rejugabilidad se puedan guardar diversos estados del juego y mis decisiones para continuar más adelante. Por el lado de diseño web, que en tu página pueda entender de qué trata tu juego, que me vendas la idea y posiblemente sería bueno un "how to play" en caso de que el juego sea algo complejo.

Requerimientos

Bases de datos

- Definir programas y datos del dispositivo (laptop) con los que se trabajará el modelo conceptual. – **NO FUNCIONAL; HISTORIA USUARIO (Esteban)**
- Diagramas de bases de datos. Entidad de relación UML – **NO FUNCIONAL; HISTORIA USUARIO (Esteban)**
- Sistema de guardado, checkpoints en la base de datos – **FUNCIONAL; HISTORIA USUARIO (Esteban)**
- Se tendrá un sistema de registro de usuarios – **FUNCIONAL; HISTORIA USUARIO (Esteban)**

- El dominio deberá tener un Host – **FUNCIONAL; HISTORIA USUARIO (Esteban)**
- Las diferentes tablas de las Bases de Datos serán: Jefes (Atributos); Personaje jugador (Atributos); Escenarios (Atributos); Puntajes de las partidas – **FUNCIONAL; HISTORIA USUARIO (Esteban)**
- Debe haber tablas con información genérica – **FUNCIONAL; HISTORIA USUARIO (Esteban)**
- Definir consultas provenientes del Front/Back-end – **NO FUNCIONAL; HISTORIA USUARIO (Esteban)**
- Agregar restricciones de integridad a las tablas (Una vez definido el modelo) a – **NO FUNCIONAL; HISTORIA USUARIO (Esteban)**
- Sistema de estadísticas – **FUNCIONAL; HISTORIA USUARIO (Gilberto)**
- Definir datos básicos a registrar en la BD – **NO FUNCIONAL; HISTORIA USUARIO (Octavio)**
- Definir datos no-básicos almacenables en la BD, y almacenarlos conforme se progrese en el juego – **NO FUNCIONAL; HISTORIA USUARIO (Octavio)**
- Definir datos básicos a registrar en la BD – **NO FUNCIONAL; HISTORIA USUARIO (Octavio)**
- Debe haber un sistema de almacenamiento de datos en la BD – **FUNCIONAL; HISTORIA USUARIO (Octavio)**
- Diseño de APIs que comuniquen BD, página web y juego. – **FUNCIONAL; HISTORIA USUARIO (Octavio)**
- Definición del método de envío de datos de la página web y videojuego a la BD. – **FUNCIONAL; HISTORIA USUARIO (Octavio)**

Videojuego

- Crear una landpage de login para el juego – **NO FUNCIONAL; HISTORIA USUARIO (Gilberto)**
- Debe haber un sistema de selección de perfiles – **FUNCIONAL; HISTORIA USUARIO (Gilberto)**
- Conexión del almacenamiento de progreso con los perfiles seleccionables y creables. – **FUNCIONAL; HISTORIA USUARIO (Gilberto)**
- Entregable del juego; que se pueda usar/interactuar . – **FUNCIONAL; HISTORIA USUARIO (Gilberto)**
- Entregable del juego; que se pueda usar/interactuar. – **FUNCIONAL; HISTORIA USUARIO (Gilberto)**
- Sistema de estadísticas – **FUNCIONAL; HISTORIA USUARIO (Gilberto)**
- El usuario debe poder formular estrategias con base en sus estadísticas – **NO FUNCIONAL; HISTORIA USUARIO (Gilberto)**
- Debe haber conexión entre el juego y página web – **NO FUNCIONAL; HISTORIA USUARIO (Octavio)**
- Definir mecánica principal del juego; la cual debe fomentar la toma de decisiones – **NO FUNCIONAL; HISTORIA USUARIO (Octavio)**
- Diseño de APIs que comuniquen BD, página web y juego. – **FUNCIONAL; HISTORIA USUARIO (Octavio)**
- Definición del método de envío de datos de la página web y videojuego a la BD. – **FUNCIONAL; HISTORIA USUARIO (Octavio)**
- Implementación de mecánicas – **FUNCIONAL; HISTORIA USUARIO (SOCIO FORMADOR)**
- Definición de sistema de decisiones que afecte a la experiencia. – **NO FUNCIONAL; HISTORIA USUARIO (SOCIO FORMADOR)**

- Investigación de mecánicas atrapantes en videojuegos. – **NO FUNCIONAL; HISTORIA USUARIO (SOCIO FORMADOR)**
- Definición de decisiones en las que podrá tomar parte el usuario. – **NO FUNCIONAL; HISTORIA USUARIO (SOCIO FORMADOR)**
- Depurar mecánicas monótonas. – **NO FUNCIONAL; HISTORIA USUARIO (SOCIO FORMADOR)**
- Ramificar decisiones del juego realizadas por el usuario. – **NO FUNCIONAL; HISTORIA USUARIO (SOCIO FORMADOR)**
- Definir datos extrapolables a una nueva partida una vez concluida cierta partida. – **NO FUNCIONAL; HISTORIA USUARIO (SOCIO FORMADOR)**
- Implementar tutorial de mecánicas del juego. – **FUNCIONAL; HISTORIA USUARIO (SOCIO FORMADOR)**
- Implementar introducción del juego en la página web – **FUNCIONAL; HISTORIA USUARIO (SOCIO FORMADOR)**

Diseño web

- Crear un tutorial del juego (Video) – **NO FUNCIONAL; HISTORIA USUARIO (Gilberto)**
- Sistema de estadísticas – **FUNCIONAL; HISTORIA USUARIO (Gilberto)**
- Conexión entre página web y BD – **FUNCIONAL; HISTORIA USUARIO (OCTAVIO)**
- Desarrollo de página web que muestre información general del juego. – **FUNCIONAL; HISTORIA USUARIO (OCTAVIO)**
- Diseño de APIs que comuniquen BD, página web y juego. – **FUNCIONAL; HISTORIA USUARIO (OCTAVIO)**

- Diseño estético de la página Web. – **NO FUNCIONAL; HISTORIA USUARIO (OCTAVIO)**
- Implementación del diseño a la página por medio de CSS y HTML. – **FUNCIONAL; HISTORIA USUARIO (OCTAVIO)**
- Definición del método de envío de datos de la página web y videojuego a la BD. – **FUNCIONAL; HISTORIA USUARIO (OCTAVIO)**

Restricciones

- Tiempo para terminar el trabajo: 10 semanas
- Presupuesto: 0 pesos
- Recursos humanos: equipo de 6 personas
- Habilidades de los integrantes del grupo
- Fecha de la entrega final: inicios de mayo
- Ritmo de trabajo: Un sprint por semana
- Género: RPG

Particular de Trello

Sprint Backlog

- Es un subconjunto del *product backlog* (requerimientos), que indica lo que será realizado durante un Sprint. A quién, y la fecha
- Para 13/03/23 - Tener requerimientos listos; con respectivas partes (como HTML y B.D.s) al igual que tipos (funciona, non-func., restriccc.)
- Sprint 1 (para 13/03/23) - Tener historias de usuario listas (falta Socio Formador)

Tareas a realizar ('To-Do')

- Entregable 2 (4.2.2 Actividad Propuesta Simulador)
- Entregable Especificación de Requerimientos de Software
- Mecánicas e ilustraciones
- Idea del juego
- Listado de assets
- DIAGRAMA ENTIDAD RELACIÓN (TODOS)

Tareas completadas

- Pasar tareas a realizar del sprint backlog
- Listado de escenas (y créditos), y listado de assets

https://docs.google.com/document/d/1BeCugrPZyDeLMqwaU7XOCCwPoAnRrkDb_aOggGUDRsBW/edit?usp=sharing

- Idea del juego(Natalia)
- Mecánicas del juego(Luis)
- Ilustraciones(Luis)
- Listado de assets(Miguel)
- Listado de escenas(David)
- Corrección de historias de usuario(Daniel)
- Clases y diagramas(Joaquin)
- Diagrama entidad relación(todos)

En proceso

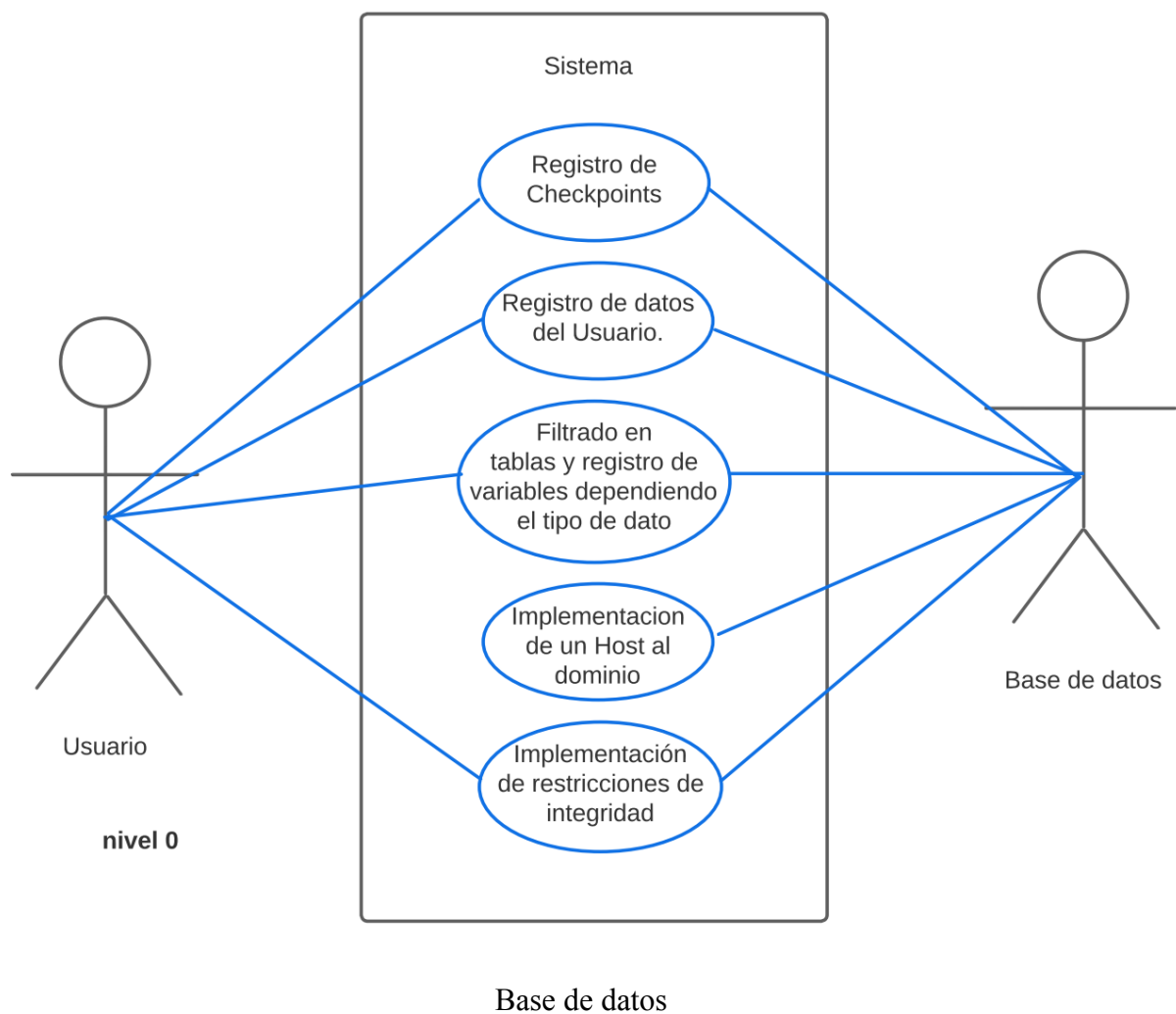
- Tener las historias del usuario
- Definir product backlog
- Definir el sprint backlog
- Especificación de requerimientos de software

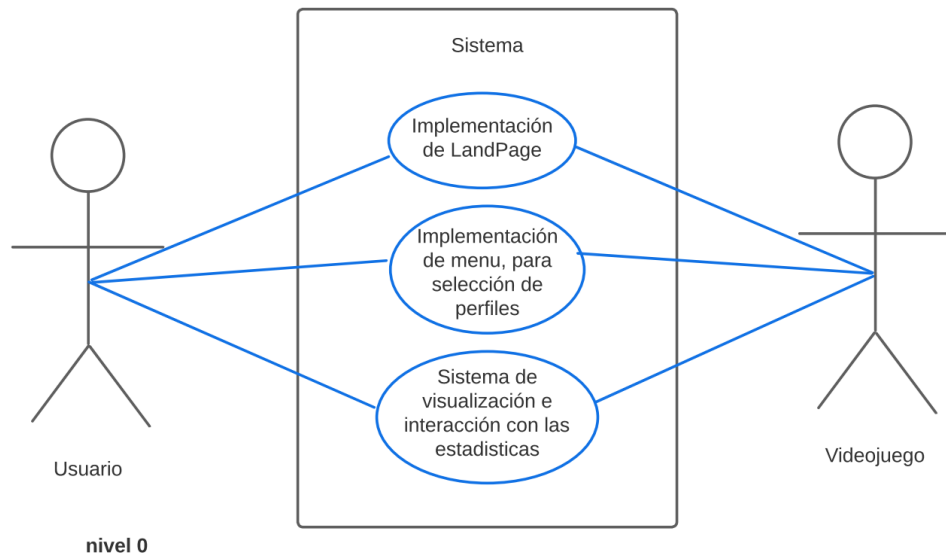
En revisión

- Revisar la evidencia; si ésta ya está en el repositorio y se logró, se pasa a "DONE".
- De lo contrario, se 'regresa'
- En cada tarjeta, para marcarla como terminada, hay que poner el link de la 'evidencia' (link, imagen, PDF; etc.)
- Definir página de arquitectura

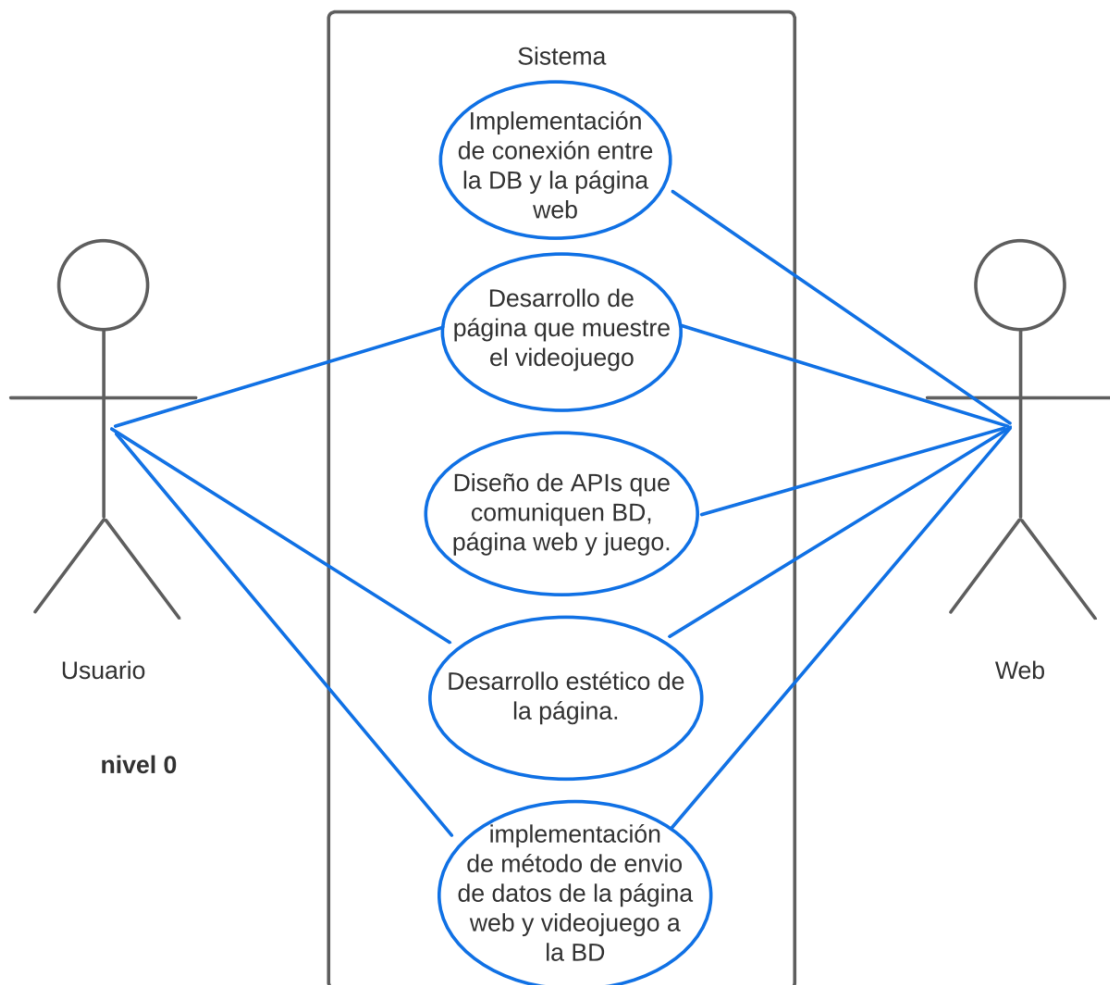
UML y Trello

UML





Diseño de videojuego



Desarrollo Web

Tablas de casos de uso

Base de datos

Nombre de caso de uso	Registro de Checkpoints	
Actores primarios	Jugador	
Actores Secundarios	Administrador de base de datos	
Disparadores del caso	El jugador llega a un lugar seguro dónde al interactuar con un objeto la partida se guarda hasta ese punto en la base de datos.	
Flujo de Ejecución	Paso	Acción
	1	El jugador interactúa con el objeto.
	2	El juego obtiene los valores float e int. Como la posición en “x” y “y” del jugador, la vida del jugador, los puntos de mana, el número identificador de las armas, poderes, game runtime, etc...
	3	El juego se comunica con la Base de datos por medio de una API como administrador e introduce los valores previamente mencionados en la tabla del jugador.

Nombre de caso de uso	Registro de Datos de Usuario	
Actores primarios	Usuario	
Actores Secundarios	Administrador de base de datos	
Disparadores del caso	El usuario se registra en la página web con su correo y contraseña, con la que se le vinculará su progreso en el juego.	
Flujo de Ejecución	Paso	Acción
	1	El usuario introduce su correo electrónico y

		contraseña en el formulario.
	2	Se verifican los datos. Validando la dirección de correo electrónico y que no se haya usado anteriormente.
	3	Se envía la información para ser almacenada en la base de datos del juego.

Nombre de caso de uso	Filtrado en tablas y registro de variables dependiendo el tipo de dato	
Actor primario	Administrador de la base de datos	
Disparadores del caso	Cuando en la API una parte del código desea encontrar y/o mostrar datos que cumplen con ciertas condiciones.	
Flujo de Ejecución	Paso	Acción
	1	De acuerdo a lo que necesité el videojuego o página web, por medio de la programación de la API, podremos hacer consultas SQL de los datos que se desean obtener. Por ejemplo: SELECT * FROM swords WHERE damage < X;
	2	Para registrar variables, podremos hacerlo por medio de un endpoint “/registro”, el usuario podrá enviar una solicitud POST con los datos que desea registrar. Por ejemplo: INSERT INTO usuarios (nombre, correo_electronico) VALUES ('Juan', 'juan@example.com');

Nombre de caso de uso	Implementación de un host al dominio	
Actores primarios	Administrador de base de datos	
Disparadores del caso	Cuando se desea almacenar y gestionar la información del juego.	
Flujo de Ejecución	Paso	Acción
	1	Se selecciona la base de datos que se va a utilizar para el juego, en este caso Mysql.
	2	Se debe configurar la base de datos para configurarla tiene que ser accesible desde el host. Esto puede requerir la configuración de permisos de acceso, configuración de usuarios y contraseñas, etc.
	3	Se debe configurar el host para que pueda conectarse a la base de datos. Esto puede implicar la configuración de puertos y protocolos de comunicación.
	4	Una vez que la base de datos y el host están configurados, se debe desarrollar el código necesario para que el juego se pueda conectar y utilizar la base de datos.

Nombre de caso de uso	Implementación de restricciones de seguridad	
Actores primarios	Desarrollador web	
Actores Secundarios	Administrador, usuario	
Disparadores del caso	Cuando se necesite proteger la información y garantizar la seguridad y privacidad de los datos del usuario y del sitio web	
Flujo de Ejecución	Paso	Acción
	1	Identificar los posibles riesgos y vulnerabilidades

		en la página web y el videojuego, como inyección de código, robo de información, acceso no autorizado, entre otros.
	2	Establecer y aplicar medidas de seguridad adecuadas para cada riesgo identificado, como el uso de protocolos seguros (HTTPS)

Diseño de Videojuego

Nombre de caso de uso	Implementación de menú para la selección de perfiles.	
Actores primarios	Usuario, desarrolladores y diseñadores de videojuegos y base de datos (nosotros)	
Actores Secundarios	Alojamiento web (Google Cloud)	
Disparadores del caso	El usuario ingresa a la página, se registra o inicia sesión y se dirige al apartado del videojuego.	
Flujo de Ejecución	Paso	Acción
	1	Muestra la interfaz de usuario diseñada, dónde se incluyen las opciones para seleccionar un perfil existente o crear uno nuevo. Por medio de un formulario se puede crear un perfil nuevo o seleccionar una lista de perfiles existentes.
	2	Cuando un usuario selecciona la opción de crear un nuevo perfil, la página web o aplicación debe permitirles ingresar información como nombre, correo electrónico, contraseña y cualquier otro campo personalizado que sea necesario.

	3	Se accede a la base de datos para guardar los datos proporcionados o para validar y acceder a la información ya antes almacenada del perfil como el nombre del perfil, correo electrónico, contraseña.
	4	En caso de ser la primera vez que el usuario ingresa al juego (es decir de no tener progreso guardado) se le muestran opciones de personalización, el usuario debe poder acceder a un menú de personalización que les permita editar los detalles de su perfil y al terminar guardar la información en la base de datos.

Nombre de caso de uso	Implementación de Landpage	
Actores primarios	Usuario, desarrolladores y diseñadores de videojuegos y web (nosotros)	
Actores Secundarios	Alojamiento web (Google Cloud)	
Disparadores del caso	El usuario ingresa a la página, se registra o inicia sesión y se dirige al apartado del videojuego.	
Flujo de Ejecución	Paso	Acción
	1	Crear sketches del landing page a mano o por medio de herramientas digitales, para tener una idea clara de los objetivos y estrategias. Definir elementos visuales y de contenido de la página, como el encabezado, descripción, botones, imágenes, videos y formularios de registro.
	2	Implementación técnica mediante el uso de los

		editores HTML, CSS y Javascript, para crear y dar el formato deseado a la página.
	3	Publicación en un servidor web y alojamiento en una dirección URL. Se puede alojar en el mismo sitio web principal o en un dominio diferente. Es importante asegurarse de que la página sea fácilmente accesible para los visitantes y que tenga una carga rápida.

Nombre de caso de uso	Sistema de visualización e interacción con las estadísticas	
Actores primarios	Usuario, desarrolladores y diseñadores de videojuegos (nosotros)	
Actores Secundarios	Alojamiento web (Google Cloud)	
Disparadores del caso	El usuario ingresa a la página, se registra o inicia sesión y se dirige al apartado de mecánicas del juego	
Flujo de Ejecución	Paso	Acción
	1	Crear sección UI en donde se muestre la información en la pantalla durante el juego, como por ejemplo la salud del jugador, tipo de arma, gemas recogidas, etc.
	2	Crear sección de mapas y mini-mapas: muestran la ubicación del jugador y los objetivos en el juego.
	3	Sección que muestre el progreso del jugador a lo largo del juego, como por ejemplo el nivel alcanzado, las habilidades desbloqueadas, las misiones completadas, etc.

Desarrollo Web

Nombre de caso de uso	Implementación de Conexión entre base de datos y página web	
Actores primarios	Desarrollador web, Administrador de la base de datos	
Actores Secundarios	Proveedor de alojamiento web (Google Cloud), Usuario final	
Disparadores del caso	El usuario ingresa a la página web.	
Flujo de Ejecución	Paso	Acción
	1	Por medio de la API en Node.js. (Dónde se programa la configuración de la conexión, el acceso y manipulación de datos, la autenticación y autorización para acceder a la base de datos, el procesamiento de datos, la generación de informes y estadísticas.) Se establece la conexión y se comparte información de la página web a la base de datos y viceversa a través de una serie de endpoints.

Nombre de caso de uso	Desarrollo de página que muestre el videojuego	
Actores primarios	Desarrollador web, Diseñador Gráfico, Creador de Videojuegos	
Actores Secundarios	Usuario	
Disparadores del caso	El usuario navega hacia la dirección donde el videojuego se mostrará	
Flujo de Ejecución	Paso	Acción
	1	Una vez con la parte del backend desarrollada se puede comenzar con el esquema de la página web.
	2	Una vez que tenemos el esqueleto de la página web

		podemos comenzar a establecer la parte del desarrollo estructural de la página(html) para después poder poner una estética adecuada para la página(css).
--	--	--

Nombre de caso de uso	Diseño de APIs que comuniquen BD, página web y juego.	
Actores primarios	Sistema de la página web, sistema del videojuego	
Actores Secundarios	Base de datos, desarrollador	
Disparadores del caso	Cargar o subir nuevas estadísticas	
Flujo de Ejecución	Paso	Acción
	1	Diseñamos y desarrollamos las APIs necesarias para comunicar la base de datos, la página web y el juego.
	2	Implementamos las APIs en el sistema de la página web y el sistema del videojuego, y configura la conexión con la base de datos.

Nombre de caso de uso	Desarrollo estético de la página.	
Actores primarios	Desarrolladores y diseñadores web	
Actores Secundarios	Usuario	
Disparadores del caso	El usuario navega en la página web	
Flujo de Ejecución	Paso	Acción
	1	Se elabora un esqueleto (html) de lo que se necesita en la página web y cómo

		quiere que se vea estéticamente.
	2	Una vez que se tiene elaborado el esqueleto se puede pasar al desarrollo del documento css en el cuál se podrá diseñar la estética que se estableció previamente.

Nombre de caso de uso	Implementación de método de envío de datos de la página web y videojuego a la BD	
Actores primarios	Desarrollador web, Desarrollador del videojuego	
Actores Secundarios	Administrador de la Base de datos	
Disparadores del caso	El usuario realiza un acción en el videojuego que genera datos y la página almacena	
Flujo de Ejecución	Paso	Acción
	1	El usuario interactúa con la página web o el videojuego, generando datos a partir de sus acciones.
	2	El sistema de la página web o el videojuego captura los datos generados por el usuario y los prepara para ser enviados.

Trello

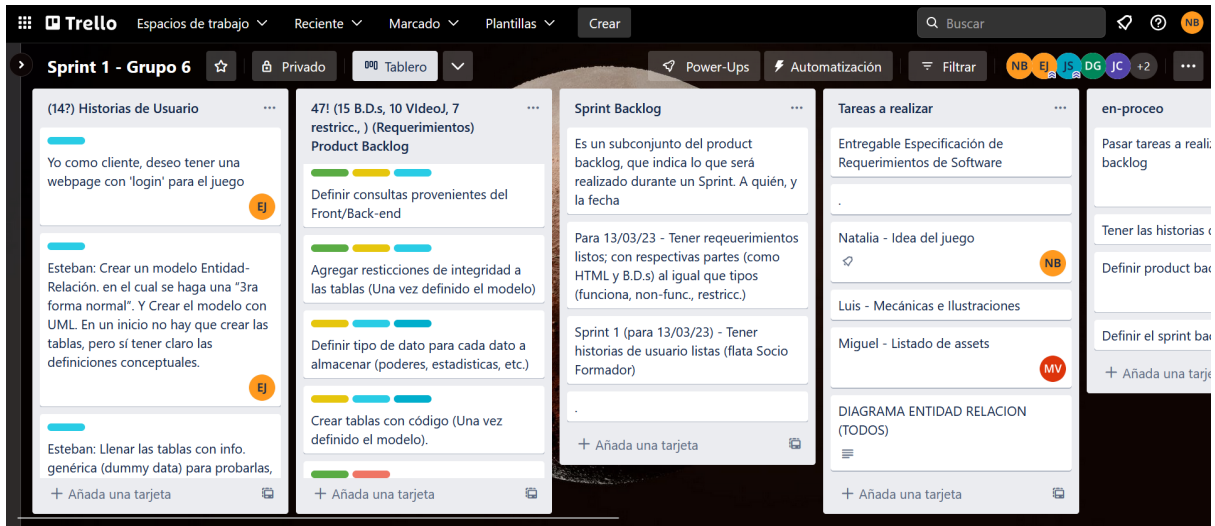


Imagen 1 – Trello pantallazo



Imagen 2 – Trello pantallazo