# Reporte Grupal

## Integrantes

José Sosa, Gabriel Sarria, Andrés Pichamba

## Descripción general

El proyecto consiste en un videojuego donde varios jugadores pueden interactuar y dispararse entre si a través de un mapa. Sus scores son registrados y una vez que mueren son restituidos en otra posición. Los jugadores también pueden hablar entre si mediante un sistema de chat, al igual que conectarse o desconectarse en cualquier momento.

## Base de Datos

La base de datos del juego fue desarrollada a través de Mongo. Los jugadores pueden registrar sus cuentas y utilizar la función de Sign Up incluida en el documento de HTML en caso de que no estén previamente registrados en la base de datos. A la vez, si un jugador ya se ha registrado puede optar por la función de Sign In y utilizar su cuenta ya existente. Cabe destacar que solo los administradores pueden realizar modificaciones en la base de datos, más no los jugadores directamente.

## Implementación del Juego

El juego fue desarrollado principalmente a través de Javascript, también utilizando al elemento de Canvas para poder dibujar al mapa, los jugadores, las barras de estado, la puntuación y el sistema de chat.

Primero, mediante Javascript se pueden crear funciones de clases para así generar a los jugadores. De esa forma se puede crear a un jugador genérico, y simplemente crear a jugadores nuevos que heredan esas mismas características, pero con un id diferente, y referentes a un nuevo jugador. Javascript también permite que la interacción entre ambos jugadores sea más fácil, al igual que cuando se disparan entre si.

Javascript también facilita los cálculos de posición, tanto de los jugadores, como de las balas. Se pueden usar además principios de herencia para que la bala siempre se oriente con la dirección y ángulo del mouse. Así mismo, se pueden definir elementos a través de funciones para revisar las teclas que han sido presionadas, enviando esa información al servidor.

## Create y Read

Cuando se inicializa a un jugador, este adquiere un nuevo ID, y a su vez hereda un paquete de inicialización con los atributos de posición, la cantidad de vida en la barra, y su score. Estos jugadores son partes de clases y son almacenados en un arreglo, donde se envía el dato al servidor. Esto a su vez inicializa al socket con la función de conexión. Con esto se revisa si el jugador está conectado, y luego se evalúa a las distintas interacciones que tiene, como las teclas que presiona, los mensajes que escribe en el cuadro de chat, y los disparos que realiza. Así mismo, se creó un paquete que aloja a los otros paquetes para así enviar la lista de jugadores del arreglo. Esto facilita la evaluación de datos cuando hay varios jugadores conectados.

Luego de que estos paquetes son evaluados en el servidor, los jugadores heredan los atributos de Javascript de la aplicación de HTML y son dibujados en la pantalla. La función de setInterval permite en este punto revisar también que no aparezcan elementos de un jugador que no ha sido creado, o a la vez, que se ha desconectado.

## Update

La función de Update lee los datos del paquete de inicialización almacenado en el arreglo, y dibuja a los jugadores que estén conectados. Una vez que son creados, la función de Update dibuja a al jugador y hace que la bala inicializada en ese elemento del arreglo herede esa posición del jugador.

Las funciones de colisión de las balas con el jugador son llamadas desde las mismas balas por facilitar el principio de herencia para comparar a la posición. Cuando un jugador muere, su vida se hace equivalente a -1, y cuando es menor que 0, genera a un nuevo jugador en una nueva posición, removiendo al anterior. Con esto, se puede generar que el jugador que ha muerto pueda crearse de nuevo, pero se lo remueve de la posición para que no aparezca en el mismo lugar y vuelva a ser disparado inmediatamente. También se reinicia la barra de vida del jugador que ha muerto. A la vez, esta función de Update le otorga un punto más al score del otro jugador, y se llama al paquete de Update con un push revisando los datos del jugador.

Finalmente, la función de Update revisa un intervalo donde se revisa a todos los elementos del socket, y aquí se llama al paquete de inicialización, Update y al de Delete que será explicado abajo. Estos paquetes son a la vez inicializados a través de arreglos para poder evaluar a cada uno de los elementos.

## Delete

Los jugadores en si se eliminan cuando se desconectan, e igualmente se eliminan a las balas. El paquete de Delete en si elimina al socket de ese jugador y todos sus atributos, incluyendo las balas, y se registra dentro de la lista con todos los sockets. Este paquete entonces se evalúa como un arreglo al final de la actualización de cada intervalo. Aquí de nuevo se aprovecha a los sockets para poder eliminar más fácilmente, tanto al jugador como a los elementos que le corresponden. A su vez, la función de desconectar también llama al paquete de eliminar, borrando al jugador de la lista. La función sin embargo no borra al jugador de la base de datos.

## Envío de mensajes

## Sistema de ingreso