LÁSER TAG

**App:**

¿Qué hace?:

La App registra los datos de usuario y las puntuaciones. En ella se configuran las armas, las vidas, los puntos de vida y se establecen los equipos y los diferentes modos de juego.

Recibe:

* Puntuaciones <– **GameController**
* ID´s emparejadas de armas y sensores <– **GameController**
* Señal de fin de juego <– **GameController**

Envía:

* ID de usuario –> **GameController**
* Configuración de armas –> **GameController**
* Configuración de equipos ­–> **GameController**
* Modo de juego –> **GameController**

**Weapon:**

¿Qué hace?:

La clase Weapon posee todas las funcionalidades relacionadas con el funcionamiento del arma, como la emisión del disparo, la recarga o la selección entre disparo automático/manual. Controla los leds de estado y envía sus diferentes estados a la clase GameController.

Recibe:

* Configuración de armas <– **GameController**

Envía:

* Confirmación de disparo (para estadísticas)–> **Statics**

Ejecuta:

* Función de disparo con intervalo de tiempo variable (man/auto)
* Conteo de balas en el cargador (salvo en modo infinito)
* Conteo del número de cargadores disponibles (salvo en modo infinito)
* Recarga del cargador
* Recarga del numero de cargadores
* Desactiva el arma en caso de muerte <– **GameController**

**Player:**

¿Qué hace?:

La clase Player se encarga de gestionar todas las funciones relacionadas con el estado del jugador, como monitorizar el número de vidas restantes, descontar salud al recibir un disparo o rellenarla al conseguir un botiquín, o activar la inmunidad.

Recibe:

* Configuración de vida <– **GameController**
* Confirmacion de impacto <– **GameController**<– **Sensores**

Envía:

* Confirmación de la muerte ­–­> **GameController­**–>**Statics/**–­>**Weapon** /–­> **Sensores**
* Responsable de la muerte –­> **GameController­**–> **Statics (Esto no va aquí, es GameController quien recibe la info directamente de los sensores)**