

FLEETSTUDIOS



Logo original hecho para el proyecto

Nombre del grupo	Flota Studios
Titulación	Titulados en la Escuela Técnica de Informática
Grupo de la titulación	A Computadores D Informática
Grupo reducido	GR1
Numero de grupo	2
Repositorio Github (url)	https://github.com/ocxpke/FleetStudios
Espacio Trello (url)	https://trello.com/invite/b/dhYyIAOf/ATTIfc8a9ecc386642ca4090a9e62ec1592689B22737/gamedevelopment
Memoria del proyecto	https://1drv.ms/w/s!AiAMSw7wbdNogfsYbCG4ct8-rt8wMg?e=GWvBkt

MIEMBROS

N.º de miembro	Nombre	Apellidos	Correo electrónico UMA
1	Clara	León Becerra	clari6260@gmail.com
2	Francisco	León Jaime	duendeeditorial@uma.es
3	José Antonio	Ramírez Girón	joseramirez@uma.com
4	Álvaro	Rodríguez Samper	alroper04@gmail.com
5	Andrea	Soler Ortega	061083213@uma.es
6	Regina	Velasco Soto	rvelasco@uma.es

TABLA DE CONTENIDOS

-Sección 1: Introducción. Hay que explicar el problema que se intenta solucionar con el sistema software a desarrollar.

El objetivo será desarrollar una versión del conocido juego de hundir la flota, incorporando nuevas características que ofrezcan una versión redefinida y más completa.

Se ha podido observar que, con el auge de los videojuegos, al ser estos más llamativos y en constante renovación, este tipo de juegos de mesa han quedado “obsoletos” pues son juegos lineales y repetitivos que nunca consiguen sorprender con alguna novedad que enganche al jugador.

Tampoco existe ninguna plataforma líder para llevar al online este clásico juego.

Nosotros buscamos aprovechar esta oportunidad y sacar un proyecto que esperemos que triunfe en el mercado gracias a su constante soporte y actualizaciones, novedades y comunidad que lo apoye. Nuevas modalidades (diferentes tableros), modo campaña, diferentes personalizaciones (mapas, skins, ...), y una gran variedad de nuevas e impresionantes armas que se podrán obtener a través de jugar, ir obteniendo recompensas y superando logros.

-Sección 2: Roles. Indicar los roles que tiene cada integrante del equipo. Cada uno debe tener dos roles, y cada rol debe estar cubierto, al menos, por dos compañeros. [Recordad que tener un rol significa ser responsable de esa tarea, pero todos los integrantes del equipo deben saber de todas las tareas.]

CONCEPTO O IDEA: Todos

DISEÑO: Álvaro, Regina y Clara

PLANIFICACIÓN: Paco, Andrea, Álvaro

PROGRAMACIÓN: José Antonio, Álvaro

ILUSTRACIÓN: Andrea, Clara, Regina

PRUEBA Y DEPURACIÓN: Todos

BANDA SONORA: José Antonio, Paco

REDACCIÓN: Regina, Paco

- Sección 3: Gestión del Riesgo. Hay que identificar los riesgos del proyecto y, para cada uno, indicar: tipo de riesgo, descripción del riesgo, probabilidad (muy baja, baja, moderada, alta o muy alta), efectos del

riesgo (catastrófico, serio, tolerable o insignificante) y estrategia para mitigarlo.

RIESGO 1:

Riesgo: Competencia directa contra los pioneros del juego.

Descripción del riesgo: Estamos entrando directamente en la zona de confort de una empresa que ya lleva tiempo fija y con el control en esta parte del sector de dicho producto por ende hay mucho por remar en contra.

Probabilidad: Muy alta.

Efectos del riesgo: Catastróficos.

Estrategia para mitigarlo: Aprovechar las ideas revolucionarias que queremos implementar y no dejar el juego obsoleto y sin actualizaciones como nuestros competidores.

RIESGO 2:

Riesgo: Incumplimiento de los plazos.

Descripción del riesgo: No ser capaces de cumplir los plazos pactados con los inversores y anunciados a los clientes finales.

Probabilidad: Baja.

Efectos del riesgo: Catastróficos.

Estrategia para mitigarlo: Establecer puntos claves en el proyecto junto a una fecha asociada e ir completando estos “checkpoints” con tiempo y planificación.

RIESGO 3:

Descripción del riesgo: Robo de datos de nuestros clientes.

Probabilidad: Media.

Efectos del riesgo: Catastróficos.

Estrategia para mitigarlo: Establecer un plan de ciberseguridad que identifique riesgos y aporte soluciones para evitarlo.

RIESGO 4:

Descripción del riesgo: Un integrante se sale del proyecto antes de finalizarlo.

Probabilidad: Baja.

Efectos del riesgo: Dificultad para encontrar con rapidez a la persona idónea. Tolerables.

Estrategia para mitigarlo: Redistribución del trabajo. Activar el protocolo comunicado en el contrato de trabajo en grupo.

RIESGO 5:

Descripción del riesgo: El cliente requiere un cambio que provoca un gran rediseño.

Probabilidad: Moderada.

Efectos del riesgo: Serios.

Estrategia para mitigarlo: Comunicarnos constantemente con el cliente para que se mantenga informado y se encuentre de acuerdo con la dirección del proyecto y su visión.

RIESGO 6:

Descripción del riesgo: Personal clave con los permisos del repositorio/documento está enfermo o no está disponible.

Probabilidad: Muy baja.

Efectos del riesgo: Tolerables.

Estrategia para mitigarlo: Asegurarnos desde el inicio que todos los miembros podemos acceder y modificar cualquier plataforma clave.

RIESGO 7:

Descripción del riesgo: La idea inicial del proyecto no permite la suficiente expansión.

Probabilidad: Moderada.

Efectos del riesgo: Serios.

Estrategia para mitigarlo: Buscar funciones que no tengan necesariamente que ver directamente con el tema principal del proyecto, pero permitan fusionarse.

RIESGO 8:

Descripción del riesgo: Se subestimó inicialmente la carga de trabajo que supone el desarrollo software que queríamos plantear.

Probabilidad: Media.

Efectos del riesgo: No cumplir plazos de entrega. Catastróficos.

Estrategia para mitigarlo: Realizar nueva valoración sobre cargas de trabajo, asignación y redefinición de plazos.

RIESGO 9:

Descripción del riesgo: La base de datos no es capaz de procesar el suficiente volumen de transacciones por segundo.

Probabilidad: Baja.

Efectos del riesgo: Serios.

Estrategia para mitigarlo: Actualizar la base de datos a un plan que permita más velocidad.

RIESGO 10:

Descripción del riesgo: Reestructuración de la empresa por problemas financieros provoca cambios directos en la organización del proyecto.

Probabilidad: Baja.

Efectos del riesgo: Reducción del presupuesto inicial. Catastróficos.

Estrategia para mitigarlo: Rediseño de objetivos para adaptarse al nuevo escenario y reasignación de tareas.

RIESGO 11:

Descripción del riesgo: Se cae el servidor de la página que ofrezca el servicio de distribución del proyecto en un momento crítico.

Probabilidad: Baja.

Efectos del riesgo: Interrupción del servicio. Catastróficos.

Estrategia para mitigarlo: Ir siempre con un margen de tiempo a la hora de la entrega y preparación del proyecto.

RIESGO 12:

Descripción del riesgo: A la hora de actualizar el código en el repositorio de GitHub encontrarse con un problema de incompatibilidad del código (debido a dos miembros trabajando al mismo tiempo) que no sepamos solucionar.

Probabilidad: Baja.

Efectos del riesgo: Bajos.

Estrategia para mitigarlo: Intentar no trabajar en el mismo instante desde dos ordenadores diferentes. En caso de suceder, que discutan los integrantes los beneficios de los cambios con el fin de decantarse por una implementación u otra.

RIESGO 13:

Descripción del riesgo: Diferencias de opinión entre los miembros del grupo a la hora de implementar o diseñar el proyecto.

Probabilidad: Moderada.

Efectos del riesgo: Tolerables.

Estrategia para mitigarlo: Mantener la comunicación en todo momento para que todos se sientan cómodos con la dirección en la que se encamina el proyecto. En caso de suceder, resolver la cuestión por medios democráticos.

RIESGO 14:

Descripción del riesgo: El desarrollo usando Eclipse puede plantear problemas sobre configuración, compatibilidad de complementos, gestión de dependencias y mantenimiento que pudieran afectar al rendimiento.

Probabilidad: Baja.

Efectos del riesgo: Tolerables

Estrategia para mitigarlo: familiarizarse con las mejores prácticas de Eclipse, mantener un entorno de desarrollo bien configurado y actualizado, y asegurarse de que cualquier complemento utilizado sea compatible y esté bien mantenido.

RIESGO 15:

Descripción del riesgo: Problemas financieros a la hora de expandir la velocidad de transacciones por segundo de la base de datos o a la hora de contratar un plan necesario en el software utilizado

Probabilidad: Baja.

Efectos del riesgo: Serios.

Estrategia para mitigarlo: Incluir en la planificación la posible necesidad de estas expansiones para poder presupuestarlo de antemano y realizar las gestiones convenientes para la obtención de préstamos.

RIESGO 16:

Descripción del riesgo: El índice de fallos es mayor al esperado

Probabilidad: Moderado.

Efectos del riesgo: Tolerable.

Estrategia para mitigarlo: Reestructurar la planificación o alternatively buscar una forma de sustituir la implementación de esa herramienta por otra.

- **Sección 4: Planificación.** Se debe indicar el modelo de proceso software elegido (y por qué se ha elegido) y pegar capturas de pantalla de los tableros Trello con las tareas identificadas y su organización. Si se han usado módulos (*Power-Ups*), pegar también capturas de pantalla con lo que muestran. Explicad brevemente las tareas.

Hemos decidido implementar una metodología tipo 'agile' por ser la que mejor se ajusta a nuestras necesidades, por brindarnos la posibilidad de adaptación a pequeños cambios que surjan durante el desarrollo del proyecto, además de la libertad que nos ofrece al explotar las fortalezas individuales de cada miembro.

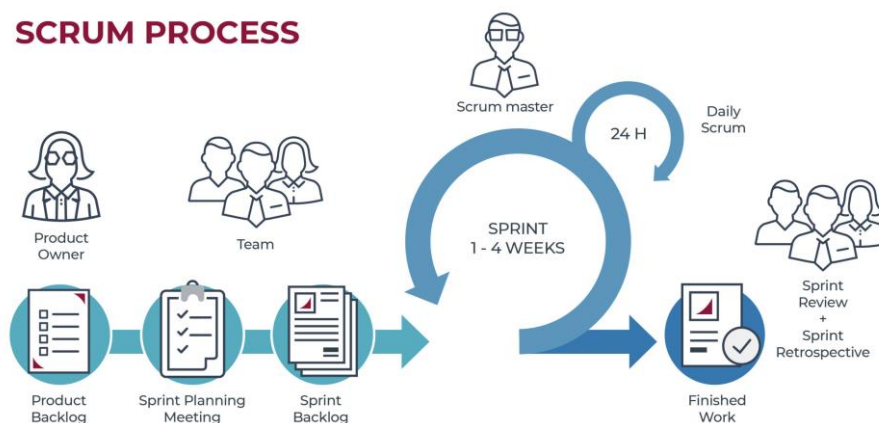
Analizando las metodologías 'agile', nos hemos decantado por la metodología *SCRUM*, por ser una metodología ampliamente usada en el sector tecnológico a la hora de desarrollar software. Asimismo, nos permite desarrollar proyectos de un tamaño pequeño/mediano, como viene a ser el propuesto por *FLEETSTUDIOS*.

Al aplicar la metodología SCRUM a nuestro proyecto tuvimos que aplicar los roles ya predefinidos dentro de *SCRUM*:

Product Owner --> Andrea Soler Ortega.

Scrum Master --> Jose Ramírez Girón.

Team --> Clara León Becerra, Francisco León Jaime, Álvaro Rodríguez Samper, Regina Velasco Soto.



El *product owner* se pondrá en contacto con el cliente, en este caso el profesor, el cual transmitirá qué espera del proyecto a desarrollar, aspiración e ideas a implementar. El *product owner* será el encargado de generar el **product backlog**, que será necesario para transmitir la idea del cliente al equipo.

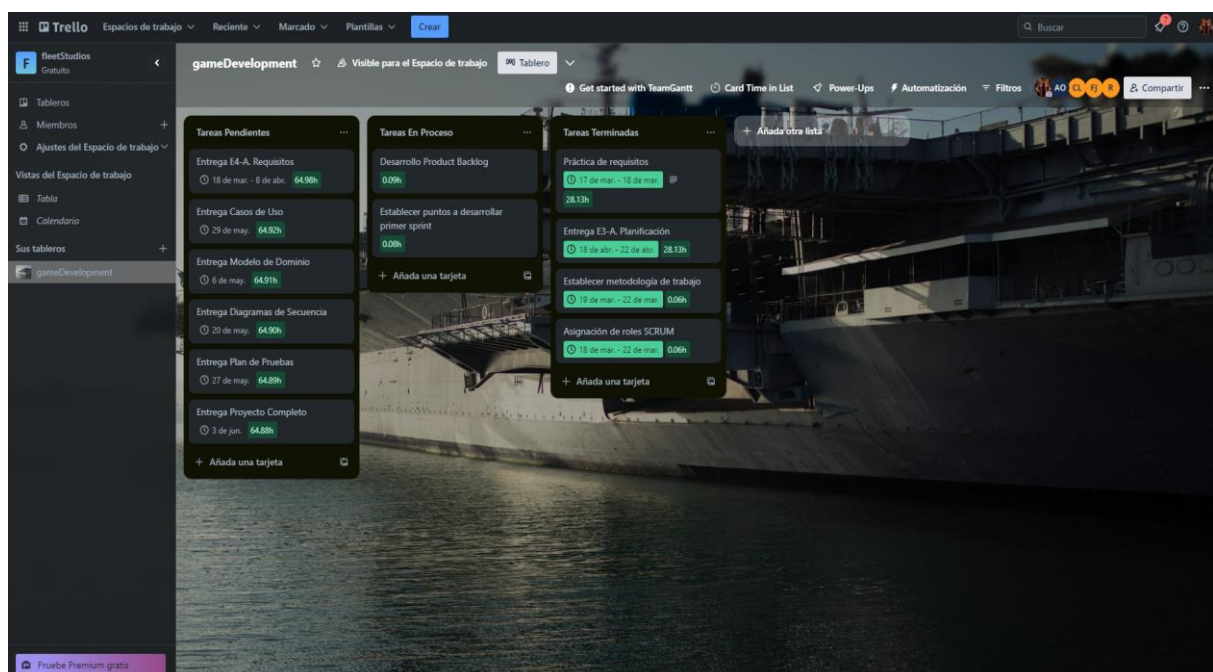
Una vez transmitida la idea, establecidos los puntos a desarrollar, componentes necesarios y establecido el orden de evolución de la aplicación, en resumen, una vez organizado el desarrollo del proyecto, nos tocará establecer los **sprints** los cuales serán de 1 semana. Además, al final de cada uno, el scrum master se encargará de tener una pequeña reunión con cada miembro del *team* para que puedan compartir sus reflexiones acerca del desarrollo del producto, y que mejoras implementaría. Aquí,

en los **sprints** ,cada miembro del equipo escogerá una tarea dependiendo de sus fortalezas. Cabe recalcar que, durante los sprints, las reuniones con el cliente pueden tener lugar y generar cambios en el proyecto.

Al finalizar cada **sprint**, se generará el **sprint backlog** que será desarrollado por el *scrum master*, donde se recogerán los siguientes puntos y requisitos a desarrollar por el equipo para el siguiente **sprint**.

Al finalizar el proyecto se tendrá un último **sprint review** y se generará un **burndown chart** para ver el desempeño del equipo a lo largo del desarrollo del proyecto. Y, por último, una charla con el cliente (profesor) para mostrar el resultado final del proyecto.

En conclusión, por las razones definidas anteriormente y teniendo en cuenta los diferentes procesos software existentes, creemos que este es el que mejor se adapta al nuestro.



En el momento de la realización del tablero, solo tenemos las entregas del proyecto según están indicadas en la planificación de la asignatura. Según vayamos avanzando se irán añadiendo nuevas tareas.



Se han habilitado estos *power-ups* para ayudar a la hora del desarrollo del proyecto. A medida que se vaya avanzando en el proyecto se generarán y se aplicaran más reglas al tablero y se habilitarán los *power-ups* necesarios.

Las tareas expuestas en el tablero se describirán en 3 tablas, las cuales describen su estado. A estas, se le añadirán más tablas durante el desarrollo del proyecto.

- Sección 5: Requisitos. Añadir los requisitos como se explica al final de las diapositivas. Es decir, identificar con tarjetas de colores los RF y los RNF (obligatorios y opcionales) en forma de historias de usuario. Se valorará positivamente que los requisitos lleven asociadas pruebas de aceptación. Cada historia de usuario debe tener un identificador (por ejemplo, RF1) y un título. Pegar como imagen el diagrama de requisitos hecho en Visual Paradigm (en dicho diagrama no se indica el texto de las historias de usuario, solamente el identificador y título). Cada grupo debe definir como mínimo tantos RFs como 5x su número de miembros, y tantos RNFs como su número de miembros. Por ejemplo, un grupo de 6 miembros debe definir al menos 30 RFs y 6 RNFs.

REQUISITOS FUNCIONALES

RF01 Colocación de barcos: Como podrá disponer el usuario los barcos y ver si esta colocación es correcta.

RF02 Detectar bombardeo o hundimiento: Método que detectara si el barco ha sido golpeado o si este se ha hundido.

RF03 Bombardear: Simple hecho de realizar un movimiento de ataque en el mapa dadas unas coordenadas.

RF04 Fin de la partida: Acción que tendrá lugar con una victoria dentro del juego, en términos de este, o por finalización del cronometro.

RF05 Turnos jugadores: Intercambio de mapas entre esta secuencia de acciones donde cada jugador vera su flota y el despliegue de esta.

RF06 Mostrar puntuación: Pequeño display de información en la interfaz, en el cual el usuario podrá ver su información.

RF07 Registro usuarios: Acción que se realizara en el menú principal donde un nuevo jugador se registrara.

RF08 Mostrar tableros: Display de la pantalla donde se podrá ver como el propio jugador ha colocado sus barcos.

RF09 Acceder tienda: Opción del menú donde esta lanzara una serie de ítems aleatorios obtenibles a cambio de monedas.

RF10 Envío notificaciones: Notificaciones que se mostraran cuando el usuario tenga que realizar algún movimiento o cambie su status dentro del juego.

RF11 Mostrar chat: Disponibilidad de hablar con el adversario a traves de una ventana flotante, chat privado por partida.

RF12 Inicio partida: Acción que contemplara el usuario registrado al estar en el menú principal.

RF13 Salta temporizador: Cuando el contrincante no hace ningún movimiento durante 20 minutos la partida se termina.

RF14 Inicia temporizador: Para prevenir inactividad empezará un temporizador tras el último movimiento.

RF15 Obtener datos de usuario: Queremos recibir del usuario un nick único, un correo y una contraseña.

RF16 Guardar sesión: Que el usuario pueda guardar su progreso en la tienda y logros si se ha registrado.

RF17 Completar logros: Si se ha registrado, el usuario debe poder visualizar y recibir sus logros en su cuenta. Por ejemplo: "Hundiste 50 barcos", "Acertaste durante 10 turnos seguidos"...

RF18 Comprar: El usuario debe poder usar en la tienda las monedas que tenga en saldo para personalizar la interfaz, la apariencia de los barcos y su perfil.

RF19 Recibir recompensas: El usuario debe de recibir recompensas completando logros. Estas deben aumentar su saldo de monedas de la tienda.

RF20 Editar perfil: El usuario debe poder consultar y modificar los datos de su perfil en todo momento.

RF21 Borrar cuenta: El usuario debe de poder darse de baja del servicio.

RF22 Integración de redes sociales: El usuario puede conectar sus cuentas de redes sociales para invitar amigos, compartir logros.

RF23 Añadir amigos: El usuario debe poder registrar como amistades a otros usuarios para ver sus perfiles y puntuaciones, así como establecer una partida.

RF24 Sistema de torneos: El usuario podrá ver el ranking de puntuaciones de jugadores que decidan jugar.

RF25 Tutorial: Mostrará un tutorial interactivo que guíe a nuevos jugadores a través del juego. Enseñará tanto las reglas como a colocar barcos, realizar ataques.

RF26 Seguridad de datos: Deberá garantizar la seguridad y privacidad de los datos de los usuarios.

RF27 Soporte técnico: Permitir hacer reportes a los usuarios de sus problemas con la aplicación.

RF28 Recuperar batalla guardada: Para recuperar una partida donde se dejó.

RF29 Vender: El usuario desde su inventario podrá vender objetos que no necesite no de los cuales necesite su valor.

RF30 Inicio usuario: El usuario podrá iniciar sesión para ver su inventario, estadística, etc...

RF31 Maquina como jugador: Un modo de juego en el que solo juega 1 persona real, con varias dificultades.

RF32 Modo de juego: Opción que el jugador podrá elegir y esta vera varias opciones ya descritas aquí.

RF33 Pruebas unitarias: Tests que se realizaran relacionados con el código limpio y una buen practica que se implementara.

RF34 Inventario: El usuario dispondrá de un apartado donde vera sus objetos obtenidos.

RF35 Items: La tienda e inventario se compondrán de estos serán objetos los cuales ayudarán en el desarrollo de la partida.

RF36 Seleccionar casilla: Al hacer clic en una casilla del tablero, se registra un intento de ataque en esa posición.

RF37 Abandonar: Se implementa un botón que da la opción a uno de los jugadores de abandonar la partida

RF38 Retroalimentación visual: Después de cada ataque, las casillas del tablero se resaltan para indicar si hubo un acierto, un fallo o un hundimiento.

RF39 Opciones de pantalla: Al iniciar el juego, se ofrece la opción de modo de juego para elegir entre varias variables como el brillo.

RF40 Barcos restantes: se muestra de manera clara la cantidad de barcos restantes del oponente.

RF41 Deshacer movimientos: Se implementa un botón de "deshacer" que permite al jugador revertir su último movimiento antes de confirmar el turno.

RF42 Notificaciones de turno: Se muestra claramente en la interfaz del juego cuando es el turno de un jugador, junto con un sonido distintivo que indica el comienzo de su turno.

RF43 Modificación del sonido: Se incluye un menú de opciones donde los jugadores pueden ajustar el volumen de los efectos de sonido y la música, así como activar o desactivar cada uno según sus preferencias.

RF44 Comprar proyectiles especiales: Proyectiles anti-defensa. Se pueden comprar proyectiles especiales que traspasen niveles de defensa.

RF45 Comprar defensas especiales: Defensas antiproyectil. Para evitar un número determinado de impactos de proyectiles se pueden comprar defensas especiales

RF46 Activar defensa especial: Para proteger un barco se puede activar un tipo de defensa especial.

RF47 Activar proyectil especial: Para superar defensas especiales enemigas se puede activar lanzamiento de proyectil especial.

RF48 Objetivo no alcanzado: Barcos no alcanzados de forma reiterada.

RF49 Bonificación de defensa: Por varios intentos consecutivos de ataques rechazados.

REQUISITOS NO FUNCIONALES

RNF1 Uso de memoria: Cuenta memoria se consume al salvar datos, cuanto ocupa la BBDD y como de eficiente se administra esa memoria.

RNF2 Tiempo de respuesta BBDD: Crucial para la sesión de los jugadores, en caso de torneo que estas sean eficaces y eficientes.

RNF3 Interfaz de usuario: Elementos que se dispondrán por pantalla al usuario que vera este, según sus necesidades, y como podrá interactuar

RNF4 Tiempo de respuesta del usuario: El usuario podrá, siempre dentro del rango temporal del turno, realizar su movimiento cuando quiera.

RNF5 Música: Elemento que se utilizara para brindar un mayor disfrute de experiencia al usuario.

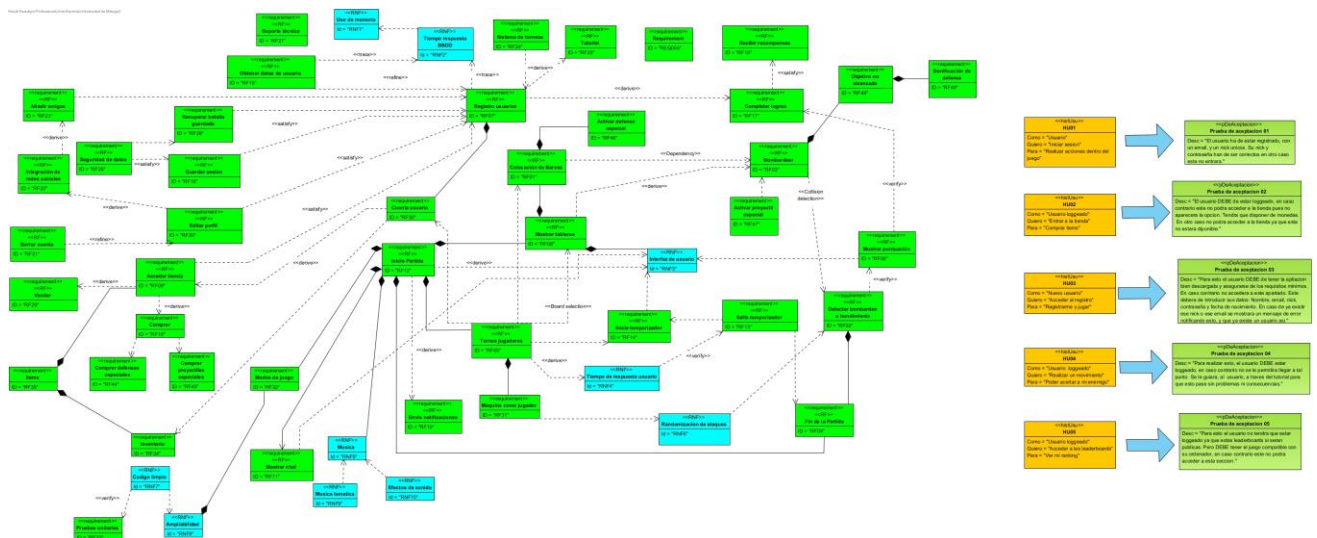
RNF6 Randomización ataques: La máquina como jugador tendrá varios modos de ataque entre los cuales estará el 'random mode' el cual puede que nunca acierte.

RNF7 Código limpio: Técnicas que se aplicaran a nuestro código para que este sea legible y práctico para aquel que intente trabajar con él.

RNF8 Ampliabilidad: Relacionado con el punto anterior, posibilidad de añadir módulos de expansión a nuestro proyecto para un mayor disfrute.

RNF9 Música de fondo: Música relacionada con el mapa del juego, hay que destacar que música engloba a todos los sonidos, por ende, no son iguales.

RNF10 Efectos de sonido: Estos serán un indicador y aparecerán cuando se golpe o falle un blanco, u ocurran otras acciones.



- **Sección 6: Herramientas software usadas durante la realización del proyecto.** Indicar las herramientas software que se han usado para desarrollar el proyecto hasta la fecha. Cualquier tipo de herramienta: de comunicación, trabajo colaborativo, elaboración de documentos, etc.

JAVA - JAVA SWING – ECLIPSE – MYSQL – SQL – TRELLO – GIT – GITHUB -
PHOTOSHOP – WHATSAPP – DISCORD – WORD – SKETCHBOOK