



## GR2 06. NITRO DRIFT

### DOBLE GRADO EN MATEMÁTICAS E INGENIERÍA INFORMÁTICA

- 1 - ALBERTO RAMIREZ COLLADO - [alberto.rc0612@uma.es](mailto:alberto.rc0612@uma.es)
- 2 - ÁLVARO PEÑA BARRAGÁN - [alvaropb02@uma.es](mailto:alvaropb02@uma.es)
- 3 - ÁNGEL MANUEL SORIA GIL - [angelmanuelsoriagil@uma.es](mailto:angelmanuelsoriagil@uma.es)
- 4 - FRANCISCO JAVIER MOLINA CUENCA - [pacomolina@uma.es](mailto:pacomolina@uma.es)
- 5 - JUAN CARLOS ALCAUSA LUQUE - [jcalcausal@uma.es](mailto:jcalcausal@uma.es)
- 6 - PABLO JULIÁN CAMPOY FERNÁNDEZ - [pabloodjdr@uma.es](mailto:pabloodjdr@uma.es)

# **ÍNDICE DE CONTENIDOS**

Introducción.....	3
Roles.....	4
Gestión de riesgos.....	5
Planificación.....	6
Requisitos.....	10
Herramientas software.....	15

## **INTRODUCCIÓN**

En el mundo del entretenimiento digital, los juegos de carreras han sido durante mucho tiempo una opción popular entre los jugadores de todas las edades. Sin embargo, o bien no difieren demasiado unos de otros, o bien carecen de algunas funcionalidades que pueden resultar atractivas al usuario.

Por tanto, pensamos que existe una creciente demanda de experiencias de juego innovadoras, y por ello, una oportunidad para desarrollar un nuevo videojuego de carreras en 2D que combine la jugabilidad clásica con elementos modernos y originales.

Así pues, el sistema software a desarrollar, es un juego de carreras en 2D que tiene como objetivo principal ofrecer una experiencia de juego envolvente, en la que los jugadores puedan competir entre sí en distintas modalidades; con una jugabilidad intuitiva y gráficos tradicionales.

Al abordar esta necesidad en el mercado de los juegos de carreras, el sistema software a desarrollar se posiciona como una opción atractiva entre los jugadores; pues pensamos es una alternativa fresca y entretenida.

## ■ ROLES

En este proyecto se han establecido cinco roles principales. Estos son los siguientes:

- **Coordinador del proyecto.** Este rol será asumido por Francisco Javier Molina Cuenta y Pablo Julián Campoy Fernández.
- **Diseño del proyecto.** Asumido por Alberto Ramírez Collado y Álvaro Peña Barragán.
- **Requisitos.** Asumido por Ángel Manuel Soria Gil y Juan Carlos Alcausa Luque.
- **Programador.** De ello nos encargamos todos los miembros del equipo. Cada uno asumirá una parte de la programación.
- **Tester.** De nuevo, todos los miembros del equipo asumimos esa responsabilidad y probaremos el código de otros compañeros para garantizar que el índice de errores sea mínimo.

# GESTIÓN DE RIESGOS

RIESGO	DESCRIPCIÓN	TIPO DE RIESGO	PROBABILIDAD	EFFECTO
Riesgo 1	El software empleado puede resultar obsoleto o no cumplir ciertos requisitos mínimos para el desarrollo del proyecto.	Tecnológico	Moderada	Tolerable
Riesgo 2	Elección errónea de las herramientas empleadas para el desarrollo del proyecto.	Tecnológico Herramientas	Alta	Tolerable
Riesgo 3	Falta de experiencia técnica que pueda condicionar el desarrollo del proyecto.	Personal	Muy Alta	Serio
Riesgo 4	Disponibilidad limitada del personal debido a otros compromisos.	Personal	Moderada	Tolerable
Riesgo 5	Reestructuraciones de la organización del proyecto	Organización	Baja	Tolerable
Riesgo 6	Cambios de requisitos con un rediseño importante.	Requisitos	Moderada	Tolerable
Riesgo 7	El cliente no aprecia la magnitud de los requisitos.	Requisitos	Moderado	Tolerable

Riesgo 8	Subestimación del tiempo necesario para el desarrollo del software.	Estimación	Baja	Serio
Riesgo 9	Subestimar el tiempo necesario para depuración del código.	Estimación	Baja	Serio

<b>RIESGO</b>	<b>ESTRATEGIA DE MITIGACIÓN</b>
Riesgo 1	Realizar un estudio de requisitos correcto para poder elegir en función a este un software que se ajuste a las necesidades del proyecto.
Riesgo 2	Cambiar las herramientas empleadas por otras más idóneas.
Riego 3	Formación y desarrollo de conocimientos necesarios para completar el proyecto.
Riesgo 4	Cada rol será cubierto por más de un miembro del equipo y mantener un compromiso con el proyecto.
Riego 5	Posibles cambios de roles, enfoques abiertos para el desarrollo y mantener una documentación asociada elaborada para facilitar estos cambios.
Riesgo 6	Modularización del proyecto para que los cambios de requisitos afecten a la menor parte posible del proyecto y se les pueda plantear una rápida y efectiva solución.
Riesgo 7	Tratar de explicar por qué ciertas propuestas no son viables y llegar a un consenso entre los requisitos del cliente y las posibilidades del desarrollo.
Riesgo 8	Llevar a cabo una buena planificación/organización de las tareas a realizar y evitar retrasos en su correspondiente realización.
Riesgo 9	Llevar a cabo una buena planificación/organización de las tareas a realizar y evitar retrasos en su correspondiente realización.

## ■**PLANIFICACIÓN**

En el proyecto a realizar se ha de decidir el modelo de planificación para llevar a cabo. Nos debatimos entre dos principales opciones: el modelo de gestión predictiva clásica y el modelo de gestión reactiva o ágil.

El primero de ellos se basa en la planificación de una ruta centrada en no desviarse de ella y seguirla al pie de la letra hasta el fin del proyecto, con escenarios planteados para evitar cualquier imprevisto posible. Por otra parte, la segunda es aquella en la que el camino seguido va cambiando centrándose en adaptarse ante los imprevistos, más que evitarlos, buscando tener estrategias para afrontarlos de forma adecuada en caso de que surjan.

La principal razón por la que nos hemos decantado por la planificación ágil es la flexibilidad a la hora de establecer las distintas tareas; ya que se trata de un proyecto muy novedoso para nosotros y además, la creación de un videojuego da pie a un gran proceso creativo; junto con los imprevistos que ello conlleva. Independientemente de los hechos inesperados que podamos tener de por medio, nuestra disponibilidad suele ser variable en el tiempo debido a nuestros peculiares horarios y cargas de trabajo. Es entonces por nuestra falta de experiencia en proyectos y por los imprevistos que podrían surgir que consideramos que el modelo de gestión ágil será más beneficioso para nuestro propósito.

Vamos a pasar a ver las tareas identificadas hasta el momento y describirlas.

- **Diseño del logo del proyecto.**
- **Realizar gestión de riesgos.** 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.
- **Elaboración de los modelos de personajes.** Realizar los diseños conceptuales de los personajes que se podrán seleccionar en el juego. Como se ha mencionado, son diseños conceptuales. Dada la naturaleza retro-pixelada de nuestro proyecto el acabado de los ‘sprites’ será ligeramente diferente.

- **Identificar perfiles de usuario y funcionalidades del juego.** En esta tarea se busca indagar en más ideas para el proyecto y formalizarlas para tener una ideas claras de qué vamos a hacer en el juego planeado. Ejemplos de esto son los distintos modos de juego que se prevé que haya como modo un jugador (modo clásico, contrarreloj, por tokens), modo multijugador o modo creador (creación de nuevos mapas y recorridos).
- **Realizar planificación del proyecto.** 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.
- **Identificar plataformas para el desarrollo del videojuego.** Debemos informarnos sobre el desarrollo de videojuegos en general para escoger la plataforma software adecuada; y de esta forma, minimizar los riesgos que una mala decisión conlleva.
- **Realizar esbozo de la interfaz del juego.** Crear un boceto gráfico en el que se plasme cómo se verán los principales menús del juego. Detallando botones e identificando elementos que lleven asociados una funcionalidad como puede ser un botón de selección de modo de juego.
- **Análisis de requisitos.** Se debe identificar, documentar y validar las necesidades y expectativas de un sistema, producto o servicio. Implica comprender y definir claramente qué funcionalidades, características y restricciones debe tener el sistema para satisfacer los objetivos del cliente o usuario final.
- **Programación.** Consistirá en la programación del juego y posteriormente, conforme se avance en el proyecto se crearán y detallarán tareas específicas para ello.
- **Test.** Consistirá en el testeo del juego y de los elementos programados. Conforme se avance en el proyecto se crearán y detallarán tareas específicas para ello.

Además, se han empleado los siguiente power-ups: Google Drive y Burndown Chart for Trello. Se incluyen imágenes de ello, así como del tablero de Trello.

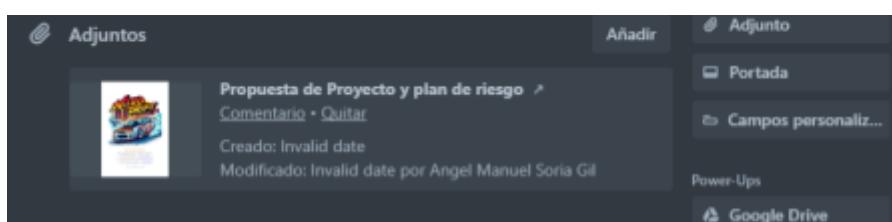


Figura 1: empleo del power-up de Google Drive

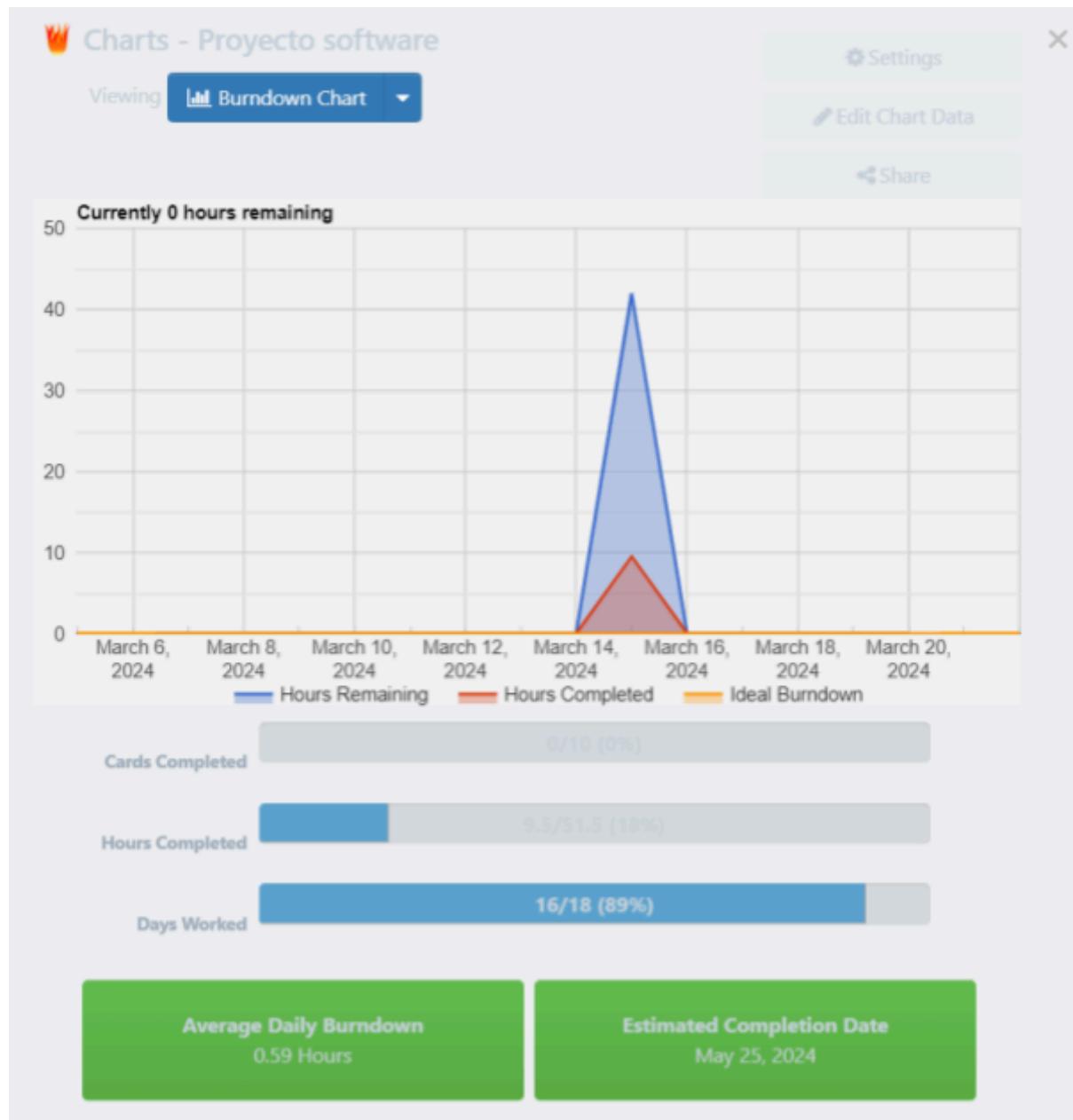


Figura 2: Gráfica mostrada por Burndown Chart for Trello

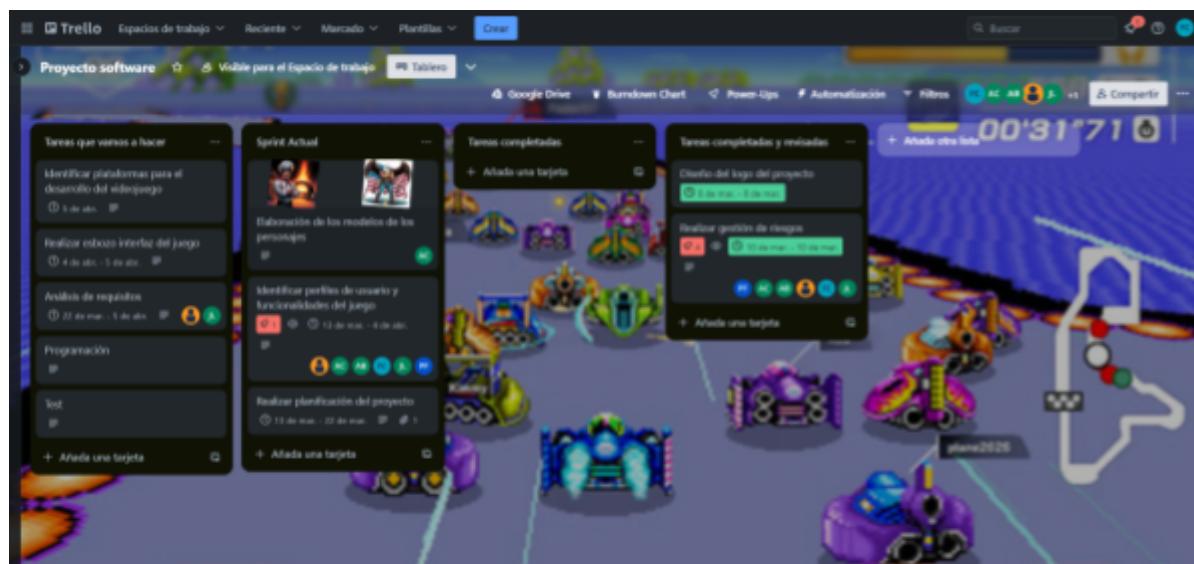


Figura 3: Tablero de Trello

# REQUISITOS

## RF1: Integración con plataformas de streaming

**Como** aficionado al streaming,  
**quiero** que el juego se integre con plataformas populares de streaming como Twitch o YouTube Gaming.

**Para** poder transmitir en directo mis carreras y compartir mi experiencia con otros jugadores.

## RF2: Sistema de mejora de vehículos

**Como** jugador,  
**quiero** poder mejorar las habilidades y atributos de mi vehículo a medida que avanza el juego.  
**Para** aumentar mis posibilidades de ganar carreras.

## RF3: Sistema de clasificación y puntuación

**Como** administrador del juego,  
**quiero** implementar un sistema de clasificación y puntuación.  
**Para** registrar y comparar el rendimiento de los jugadores en las carreras para fomentar la competencia y la superación personal.

## RF4: Sistema de monetización equilibrado

**Como** equipo de monetización,  
**queremos** implementar un sistema de microtransacciones y contenido descargable que sea equilibrado y justo para los jugadores.  
**Para** garantizar una experiencia de juego satisfactoria sin crear desequilibrios o ventajas injustas.

## RF5: Sistema de Replay

**Como** desarrollador del juego,  
**quiero** implementar un sistema de Replay que permita a los jugadores revisar y compartir reproducciones de carreras.  
**Para** que puedan analizar su desempeño y compartir momentos destacados con otros jugadores.

## RF6: Ligas entre amigos

**Como** jugador social,  
**quiero** tener un modo de competir con mis amigos y desafiarlos.  
**Para** añadir competitividad y alicientes que te hagan jugar.

## RF7: Valoración de circuitos y personajes

**Como** administrador del juego,  
**quiero** que los usuarios puedan valorar circuitos y personajes.  
**Para** tener feedback y trabajar en mejoras en función de la opinión de los jugadores.

## RF8: Control de música y sonidos

**Como** jugador,  
**quiero** poder pausar el juego.  
**Para** configurar si quiero tener la música de fondo activa para tener mayor concentración en el juego o ponerme mi propia música.

## RF9: Marcadores

**Como** administrador del juego,  
**quiero** mostrar al jugador la distancia, tiempo y tokens obtenidos en la partida actual y el récord que haya conseguido hasta ahora.  
**Para** que tenga un modo de ver su progresión en la carrera y comparar con lo conseguido hasta el momento en otras partidas.

## RF10: Multijugador offline

**Como** equipo de desarrollo,  
**queremos** permitir un modo multijugador en un único dispositivo.  
**Para** que los jugadores puedan seguir disfrutando en zonas remotas sin conexión.

#### RF11: Sistema de guardado automático

**Como** jugador,  
**quiero** que al final de cada carrera o conjunto de carreras el juego haga un guardado automático.  
**Para** evitar la pérdida del progreso realizado por el jugador.

#### RF16: Jerarquía en los circuitos

**Como** jugador,  
**quiero** que se presenten distintos circuitos que se vayan desbloqueando progresivamente al completar los anteriores, estableciendo una jerarquía entre ellos.  
**Para** tener el juego organizado en niveles permitiendo avanzar en él hasta completarlo.

#### RF12: Personalización de los recorridos

**Como** jugador,  
**quiero** que se me ofrezca la posibilidad de personalizar circuitos y elegir cuales quiero recorrer y en qué orden.  
**Para** obtener una experiencia de juego rica.

#### RF17: Notificación de errores

**Como** administrador del juego,  
**quiero** poder registrar de alguna manera los errores que se produzcan durante una partida de algún jugador concreto.  
**Para** que en caso de producirse un bug o bloquearse el juego se informe al equipo de desarrollo incluyendo información de la máquina en la que se estaba ejecutando y las circunstancias en que se ha producido (mapa, vehículo, modo de juego, etc.).

#### RF13: Lógica de asignación de objetos

**Como** diseñador de juegos,  
**quiero** implementar una lógica en la asignación de objetos de las cajas misteriosas que dependa de la posición del jugador en la carrera.  
**Para** proporcionar un equilibrio y una experiencia de juego justa y emocionante.

#### RF18: Control de daños de los vehículos

**Como** desarrollador del juego,  
**quiero** que se muestre el daño sufrido por los vehículos durante las carreras en función de los accidentes que se produzcan (colisiones con otros vehículos y obstáculos).  
**Para** tenerlos en cuenta a la hora de realizar una clasificación global de jugadores y partidas.

#### RF14: Personalización de la dificultad

**Como** desarrollador del juego,  
**quiero** incluir la posibilidad de que cada jugador pueda personalizar su propia dificultad en el juego.  
**Para** que los jugadores puedan adaptar la experiencia de juego a sus preferencias y habilidades.

#### RF19: Modo de juego de entrenamiento

**Como** jugador,  
**quiero** que se ofrezca un modo de juego "contra la máquina" en la que un único jugador pueda participar sin conexión con otros jugadores ficticios.  
**Para** tener la posibilidad de realizar entrenamientos.

#### RF15: Logros y premios

**Como** jugador,  
**quiero** tener una serie de metas a corto plazo que me sirvan de orientación para mejorar mis habilidades en el juego.  
**Para** motivar la progresión y el compromiso del jugador con el juego.

#### RF20: Personalización de los mapas de juego

**Como** equipo de desarrollo,  
**quiero** incluir opciones de personalización dentro de cada uno de los mapas. Por ejemplo, poder cambiar el tiempo (modo lluvia/soleado/nublado...).  
**Para** hacerlos más realistas.

### RF21: Habilidades

**Como** desarrollador del juego,  
**quiero** poder tener habilidades con los que el coche  
 haga movimientos especiales.  
**Para** hacer el juego más frenético.

### RF22: Estadísticas de vehículo

**Como** jugador,  
**quiero** poder elegir entre distintos vehículos que me  
 proporcionen distintas estadísticas.  
**Para** personalizar las estadísticas en función del  
 mapa.

### RF23: Misiones diarias

**Como** administrador del juego,  
**quiero** implementar un sistema de misiones diarias.  
**Para** que los jugadores se vean incentivados a jugar  
 de manera continua y puedan avanzar de manera  
 más rápida.

### RF24: Carrera semanal

**Como** desarrollador del juego,  
**quiero** implementar una carrera semanal.  
**Para** que los jugadores luchen por la clasificación en  
 la carrera.

### RF25: Perfil personalizable

**Como** jugador,  
**quiero** personalizar mi perfil con distintas fotos de  
 perfiles y tarjetas de visita con mis estadísticas de  
 carrera y logros.  
**Para** que el resto de jugadores puedan visitar mi  
 perfil y ver mis logros.

### RF26: Selección de personaje

**Como** jugador,  
**quiero** que se me ofrezca la posibilidad de elegir entre varios  
 personajes, teniendo cada uno de ellos unas estadísticas  
 distintas.  
**Para** poder tener una experiencia de juego más variada.

### RF27: Sistema de estadísticas generales

**Como** desarrollador del juego,  
**quiero** poder establecer unas estadísticas generales como  
 horas jugadas, tokens conseguidos, personaje y vehículo más  
 usado, así como el circuito que más veces se ha jugado.  
**Para** que cada jugador pueda llevar un seguimiento de su  
 experiencia.

### RF28: Sistema de eventos por tiempo limitado

**Como** administrador del juego,  
**quiero** desarrollar eventos especiales por tiempo limitado, que  
 ofrezcan recompensas exclusivas y desafíos únicos para  
 completar individualmente o con otros jugadores.  
**Para** fomentar la competitividad o la cooperatividad,  
 dependiendo del tipo de evento.

### RF29: Personalización de los controles

**Como** jugador,  
**quiero** tener la posibilidad de modificar la sensibilidad y  
 disposición de los controles, así como la opción de poder  
 emplear distintos dispositivos de entrada.  
**Para** adaptar la experiencia de juego a mis preferencias y  
 necesidades.

### RF30: Sistema de menús intuitivos

**Como** desarrollador del juego,  
**quiero** implementar menús sencillos de navegar y entender,  
 que presenten las distintas opciones de forma clara y con una  
 estética que resulte agradable y simple.  
**Para** facilitar la experiencia del jugador.

### RF31: Personalización de objetos

**Como** jugador,  
**quiero** tener la posibilidad de restringir qué tipo de objetos  
 puedo obtener de las cajas misteriosas.  
**Para** adaptar mi estrategia de juego a mis preferencias y  
 habilidades.

### RF32: Restricción de uso de objetos

**Como** administrador del servidor,  
**quiero** tener la capacidad de prohibir el uso de objetos  
 obtenidos de las cajas misteriosas en determinadas partidas.  
**Para** agregar variedad y desafío a las carreras y promover  
 estrategias alternativas.

#### RNF1: Accesibilidad para personas con discapacidad

**Como equipo de desarrollo,**  
**queremos** diseñar el juego de manera que sea accesible para personas con discapacidades visuales, auditivas o motoras.

**Para garantizar** que todos los jugadores puedan disfrutar del juego.

#### RNF4: Compatibilidad con distintos sistemas operativos

**Como equipo de desarrollo,**  
**queremos** que el juego esté disponible en distintos sistemas operativos (Windows, Linux, MacOS...).

**Para asegurar** que los controles en cada uno de ellos se adapten correctamente.

#### RNF2: Controles intuitivos

**Como equipo de desarrollo,**  
**queremos** tener unos controles que se aprendan a usar con solamente una partida.  
**Para minimizar** el tiempo de aprendizaje y maximizar la jugabilidad.

#### RNF5: Accesibilidad para distintas generaciones de dispositivos

**Como equipo de desarrollo,**  
**queremos** que el juego sea accesible para dispositivos que no sean únicamente de última generación.  
**Para aumentar** la cantidad de jugadores que puedan disfrutar del juego.

#### RNF3: Manejo fluido

**Como equipo de desarrollo,**  
**queremos** que los controles sean fluidos y se refleje bien la naturaleza retro del videojuego en los movimientos de sus distintos personajes y transportes.  
**Para garantizar** una mejor experiencia de juego.

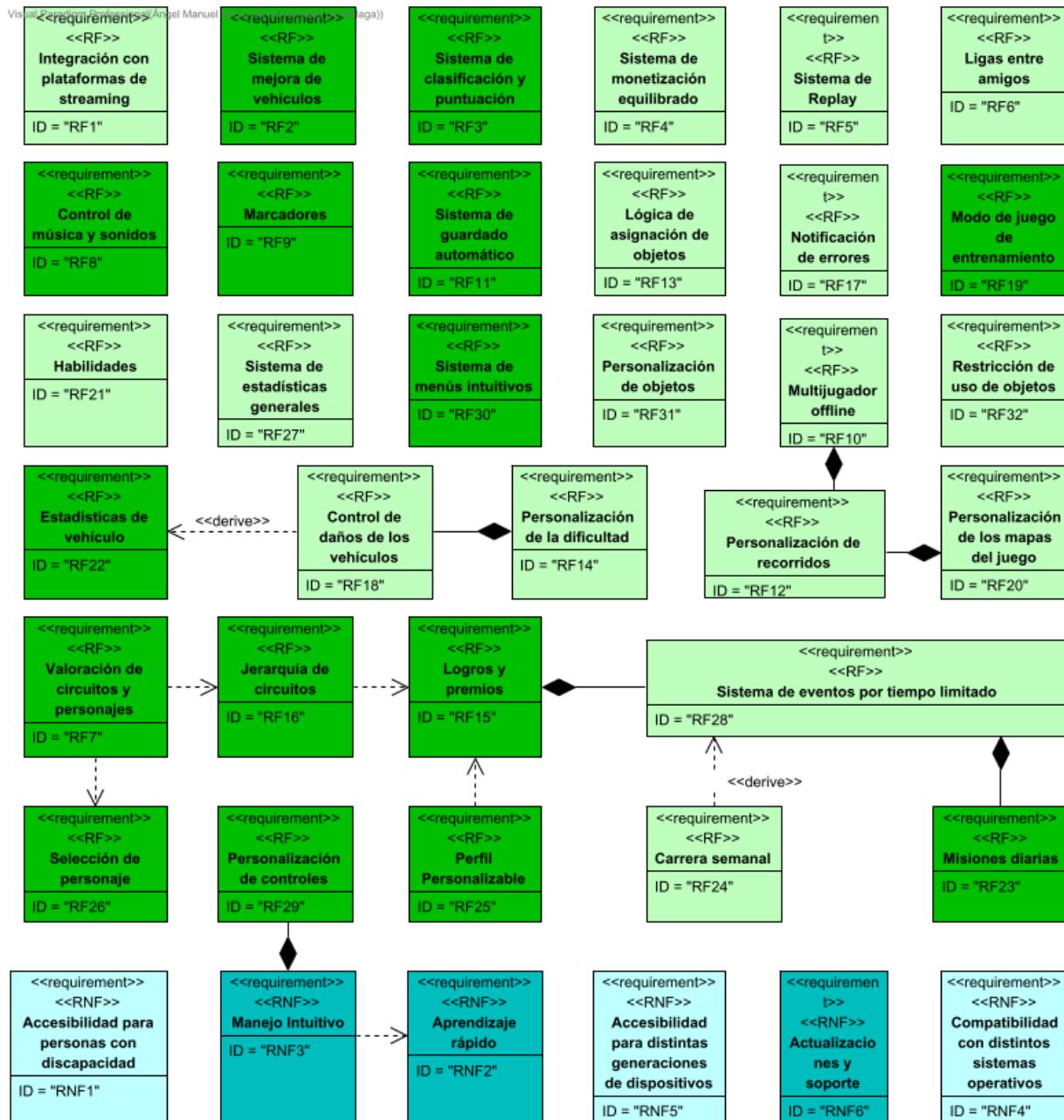
#### RNF6: Actualizaciones y soporte

**Como equipo de desarrollo,**  
**queremos** que el juego reciba actualizaciones regulares, tanto de contenido como de corrección de errores.  
**Para mantener** la calidad y la atención del consumidor el mayor tiempo posible.

#### Pruebas de aceptación RF8.

- Durante toda la partida, se verifica que se muestre un botón de pausa en la esquina superior derecha de la pantalla.
- Se confirma que al pulsar el botón de pausa, el juego se detenga adecuadamente. Se verifica que si no se pulsa el botón de pausa, el juego continúe sin interrupciones.
- Al pausar el juego, se verifica la aparición de un botón de encendido/apagado para controlar la música.
- Se comprueba que al mantener la música encendida, ésta continúe reproduciéndose sin interrupciones.
- Se verifica que al seleccionar la opción de apagar la música, ésta se detenga de manera adecuada.

Recogemos finalmente todos los requisitos y relaciones en el diagrama mostrado en la página siguiente.



## **HERRAMIENTAS SOFTWARE**

Entre las herramientas software empleadas hasta la fecha en el proyecto destacamos dos grupos principales:

- **Upper Case** como:

- **Google Drive.** Empleado como gestor de archivos y herramienta de trabajo colaborativo para la compartición de documentos
- **BingAI.** Empleada para el diseño de imágenes como el logo de grupo.
- **Trello.** Empleado para la planificación del proyecto y como gestor de actividades o calendario.
- **Visual Paradigm.** Empleado en la elaboración del diagrama de requisitos y modelar las relaciones entre ellos.

- **Lower Case** como:

- **GitHub.** En él guardamos las diferentes versiones del código del proyecto y la documentación asociada a este.