



Universidad del Bío-Bío
Facultad de Ciencias Empresariales
Dept. de Sistemas de Información

Desarrollo de una aplicación web para la comunidad de Re-Volt America

Proyecto de título para optar al título de Ingeniero de Ejecución en
Computación e Informática

José Benavente

Lunes 18 de diciembre, 2023

Índice

Abstracto	4
Dedicatoria	5
Agradecimientos	6
Resumen	7
Introducción	8
1. Estudio del Problema	9
1.1. Definiciones, Siglas y Abreviaciones	9
1.2. Historia de Re-Volt y sus Comunidades	9
1.3. La Comunidad de Re-Volt America	12
1.4. Contexto del Problema	13
1.4.1. Re-Volt y Partidas en Línea	13
1.4.1.1. Autos y Categorías	13
1.4.1.2. Sala de Espera	14
1.4.1.3. Carreras en Línea	15
1.4.2. Re-Volt: OpenGL y los Session Logs	16
1.4.3. Funcionamiento Interno de Re-Volt America	18
1.4.3.1. Sistema de Temporadas, Rankings y Sesiones	18
1.4.3.2. Paquete de Contenido de RVA	19
1.4.3.3. Sistema de Puntuación	19
1.4.3.4. Cálculo de Resultados	21
1.4.3.5. Multiplicadores por Auto	23
1.5. Sistema Actual	24
1.6. Problemática Actual	24
1.6.1. Diagrama de la Situación en la Actualidad	25
1.7. Propuesta de solución	26
1.8. Soluciones similares disponibles	26
1.8.1. Aplicación para Cálculo de Puntos de Re-Volt I/O	26
1.9. Justificación del Problema	27
2. Proyecto	28
2.1. Objetivo General del Proyecto	28
2.2. Objetivos Específicos del Proyecto	28
2.3. Metodología de Desarrollo	28
2.4. Técnicas y Notaciones	30
2.5. Estándares de Documentación	30

2.6. Software, Frameworks y Lenguajes Utilizados	30
3. Factibilidad	32
3.1. Factibilidad Técnica	32
3.1.1. Conocimientos de los Usuarios	32
3.1.2. Disponibilidad Profesional	32
3.1.3. Despliegue y Servidor	33
3.2. Factibilidad Operativa	33
3.3. Factibilidad Económica	34
3.3.1. Tablas de Costos	34
3.3.2. Flujo de Caja	35
3.3.2.1. Contexto e Indicadores Económicos	35
3.3.2.2. Puesta en Marcha	35
3.3.2.3. Cálculo del Valor Actual Neto	36
3.4. Conclusión de Factibilidad	37
4. Arquitectura de Re-Volt America	38
5. Requerimientos del Software	39
5.1. Límites	39
5.2. Caracterización de los Usuarios	39
5.3. Objetivo General del Software	40
5.3.1. Objetivos Específicos del Software	40
5.4. Requerimientos Funcionales del Software	41
5.5. Requerimientos No Funcionales del Software	43
5.6. Interfaces Internas de Salida	47
5.7. Interfaces Externas de Salida	48
6. Análisis Funcional	49
6.1. Actores	49
6.2. Casos de Uso	51
6.2.1. Diagramas de Casos de Uso	51
6.2.2. Especificación de los Casos de Uso	55
6.2.3. Detalle de los Casos de Uso	57
6.3. Modelo de Datos	66
6.4. Esquema de la Base de Datos	68
6.5. Diseño de Interfaz	78
6.5.1. Fuentes	85
6.5.2. Paleta de Colores	85
6.6. Diseño de Arquitectura	85
6.7. Estructura del código	85
6.7.1. Backend	86
6.7.2. Frontend	86

7. Plan de Capacitación, Implementación y Puesta en Marcha	87
7.0.1. Estado del Proyecto	87
7.0.2. Proyecciones del Proyecto	87
8. Conclusión del Proyecto	88
8.0.1. Proyecciones del Proyecto	89
9. Anexos	90
9.1. Anexos de Pruebas de Aceptación	90
9.2. Anexos de Recopilación de Información	90
9.3. Anexo Diccionario de Datos	91
9.4. Anexo Aspectos de Gestión de Proyectos	91
9.4.1. Anexo Carta Gantt	91
9.4.2. Riesgos de Alto Nivel, Impacto y Estrategia	92
9.4.3. Anexo Estimación Casos de Uso	92
9.4.4. Anexo Resumen de Esfuerzo	92
9.5. Anexos Retrospectiva del Proyecto	92
9.5.1. Anexo Iteraciones en el Desarrollo	92

Abstracto

A

Dedicatoria

El presente proyecto está dedicado a toda la comunidad de Re-Volt, en especial a quienes forman parte del grupo de jugadores de Re-Volt America.

Sin lugar a dudas, durante los últimos años, este juego pasó de ser un simple pasatiempo a convertirse en algo muy importante para mí a nivel personal. Muchas de las personas que he conocido a través de Re-Volt se han convertido, ya a estas alturas, en buenos amigos a quienes valoro muchísimo. Todos han sabido siempre apreciar mi trabajo, y por eso les estoy infinitamente agradecido. Sin ustedes, nada de lo que amo hacer tendría trascendencia alguna. De todo corazón, muchas gracias. Es por todo lo anterior que no podría dedicar este trabajo a nadie si no a ustedes.

Agradecimientos

En primer lugar, se extiende un agradecimiento formal a los siguientes desarrolladores quienes, de una forma u otra, han contribuido a la base de código del presente proyecto:

- Marco Roth. Por ayudar con la migración del proyecto a "esbuild", y por resolver problemas con HAML.
- Nicolás Duque. Por probar la instalación del proyecto en plataformas de Linux, testear el proyecto en su fase beta, y por ayudar con la implementación de traducciones.
- Luigi Riccio.
- Esteban Martinez
- Henrique Gomes Britos

También se les extiende un agradecimiento a todos quienes han hecho una donación voluntaria, sin importar el monto que fuese, al desarrollador principal de este proyecto a través de GitHub Sponsors durante el desarrollo de esta tesis:

- Vicente Aguilera.
- Gabriel Carnielli.
- Dario Chaile.
- Benjamín Contreras.
- Benjamín Ferrada.
- Josafat Jiménez.
- Mateusz Kobylański.
- Jorge Matamala.
- Benjamín Mosso.
- Leandro Rodríguez.
- Juan Pablo Rosas.

Resumen

VENDER REVOLT AMERICA:

RV AMERICA HA QUEDADO ATRAS EN CUANTO A SU SOPORTE Y GRACIAS A ESTE PROYECTO CRECERÁ Y SE VOLVERÁ MÁS MASIVO ETC...

El presente informe trata de una aplicación que busca centralizar toda la información relacionada a las sesiones multijugador del videojuego Re-Volt celebradas por la comunidad de Re-Volt America, la cual busca mejorar la experiencia de usuario para los administradores encargados de mantener los registros de resultados y tablas de puntuación de Re-Volt America actualizadas, además de servir como un cambio revolucionario para todos aquellos que buscan conseguir una experiencia competitiva dentro del videojuego.

Introducción

El presente informe contiene las especificaciones técnicas correspondientes al desarrollo del proyecto de titulación de la carrera de Ingeniería de ejecución Informática titulado “Desarrollo de una aplicación web para la comunidad de Re-Volt America”.

El documento se organiza en varios capítulos, los cuales van desde la definición del problema, contexto situacional, y desafíos a enfrentar, hasta una propuesta de solución completa en donde se entra en profundo detalle acerca de la forma en que esta propuesta pretende resolver la situación en cuestión.

ESPECIFICAR CONTENIDO DE CADA CAPITULO POR VENIR. HACERLO CON TODOS LOS CAPITULOS

Capítulo 1

Estudio del Problema

1.1. Definiciones, Siglas y Abreviaciones

Como aclaración inicial, **Re-Volt America** es un nombre propio de comunidad, el cual se escribe sin acentuar la letra "e" como vendría normalmente acentuada en el idioma español. El anglicismo de "America" es completamente intencional, ya que el nombre proviene del idioma inglés americano.

A continuación, se definirán algunos conceptos relevantes en el contexto del videojuego Re-Volt y de la comunidad de Re-Volt America:

- Re-Volt: El videojuego Re-Volt, publicado por Acclaim Studios en 1999.
- RV: Abreviación para "Re-Volt".
- Re-Volt: OpenGL: Reescritura moderna de Re-Volt, basada en OpenGL.
- RVGL: Abreviación de "Re-Volt: OpenGL".
- Re-Volt I/O: Comunidad europea de Re-Volt.
- Re-Volt America: La comunidad americana de Re-Volt.
- RVA: Abreviación para "Re-Volt America".
- Sesión: Serie de carreras de RVGL en línea, en donde dos o más personas compiten en carreras multijugador.
- Session Log: Archivo separado por comas, el cual contiene un registro crudo de los resultados de las carreras jugadas en una sesión de RVGL.

1.2. Historia de Re-Volt y sus Comunidades

Re-Volt America, en su expresión más simple, es una comunidad de jugadores del videojuego Re-Volt, el cual fue lanzado originalmente en el año 1999 por Acclaim Studios en Londres. Re-Volt es un videojuego de carreras y simulador arcade de autos a control

remoto, el cual explora una premisa en donde dichos autos compiten en carreras de radio control en ambientes como museos, supermercados, barcos, sitios de construcción, entre otros. Esto combinado con una mecánica de objetos que pueden ser recogidos por dichos autos para atacar a los competidores, obtener más velocidad, entre otras ventajas.

El arte original de la caratula del juego puede ser apreciado en la figura FIGNUM.



El primero de septiembre del año 2004, Acclaim Studios se declara en banca rota, y cesa permanentemente todo el desarrollo y mantenimiento que en algún momento proveyó a Re-Volt y a su comunidad. Este suceso, a lo largo de los años, dio lugar a muchas comunidades segmentadas del juego en el internet de ese entonces. Con el tiempo, nuevos sitios y proyectos comenzaron a surgir, tales como el portal web de Re-Volt Race, una página de Re-Volt que se dedicaba a organizar partidas online y mantener tablas de resultados para los jugadores, o Re-Volt: OpenGL (RVGL), una re-escritura moderna del Re-Volt original que todos conocían, ahora disponible para plataformas modernas y otros sistemas operativos además de Windows, como Linux, MacOS, e incluso una versión para dispositivos Android.

Dentro de lo anteriormente enmarcado, aparece en el año 2015 la comunidad de Re-Volt I/O, cuyo logotipo se puede apreciar en la figura FIGNUM. Esta comunidad estaba formada por un grupo de jugadores de Re-Volt, principalmente europeos, quienes incursionaron por primera vez en intentar crear una plataforma estable para el videojuego y su comunidad de jugadores. Este sería un lugar en donde cualquiera que quisiera disfrutar del juego podría encontrar guías de ayuda, tutoriales, descargas y demás contenido para poder instalar y jugar Re-Volt en su computador o dispositivo móvil.



En sus inicios, Re-Volt I/O adoptó a RVGL como la distribución estándar de Re-Volt que ofrecería a sus jugadores, haciéndole ganar público y reconocimiento al proyecto publicando enlaces de descarga directos en su página web (re-volt.io), además de entregar soporte y mantener hilos de discusión relacionados con RVGL y sus actualizaciones en su foro oficial (forum.re-volt.io).

En adición a lo anterior, RVGL no era tan sólo una versión modernizada del Re-Volt original, sino que también traía consigo el aspecto más importante que tiene Re-Volt en la actualidad, y el cual mantiene unida y activa a su comunidad en general: el modo multijugador u online. Dicho modo no sólo permitía a los jugadores correr carreras en línea, sino que, además, extendía soporte para que miembros de la comunidad pudiesen diseñar sus propios autos y pistas de manera personalizada, agrandando así, de manera casi infinita, el repertorio de contenido descargable para Re-Volt.

Re-Volt I/O adoptó un sistema en donde su administración elige ciertos autos y pistas hechos por la comunidad cada ciertos meses. De esta forma, todos estos autos y pistas, elegidos a votación, terminan juntos en un paquete de contenido de extensión para RVGL, el cual Re-Volt I/O se encarga de distribuir para que sus usuarios lo descarguen y puedan jugar en línea. De manera habitual, tener este paquete de contenido es obligatorio para poder jugar en las sesiones multijugador organizadas por Re-Volt I/O, lo cual lo convertiría en un estándar para los jugadores que quisieran incorporarse a la comunidad en toda su extensión.

Fue así como Re-Volt I/O, entre finales del 2015 y mediados del 2017, logró consolidarse y llegar a más jugadores que nunca, formando una comunidad activa de amantes del juego quienes, espontáneamente, se reunían a jugar en línea durante la semana utilizando un paquete de contenido adicional para RVGL, el cual todos debían descargar e instalar por separado para poder jugar. Eventualmente, estas partidas en línea adquirieron un horario definido con fechas y horas acordadas con antelación, para así facilitar la asistencia de los jugadores a los eventos de carreras.

En la actualidad, Re-Volt I/O sigue siendo la comunidad de Re-Volt más grande en términos de jugadores y escala, pero en si todas las comunidades de Re-Volt están unidas y se ayudan unas con otras. Después de todo, se trata de un juego nicho, en donde todos intentan hacerlo accesible y fácil de entender para quienes deseen formar parte de su comunidad.

1.3. La Comunidad de Re-Volt America

Si bien Re-Volt I/O fue, durante muchos años, la única comunidad grande de Re-Volt a nivel mundial, no fue mucho después de su gran auge que comenzarían a formarse los demás grupos que, a día de hoy, tienen gran relevancia en la escena multijugador de Re-Volt y que, además, cuentan con un numeroso público y gran actividad. Dentro de estas nuevas comunidades se encuentra Re-Volt America, la comunidad de Re-Volt que abarca a todos los jugadores del continente americano, especialmente de latinoamérica. El logotipo oficial de Re-Volt America, o RVA para abreviar, puede apreciarse en la figura FIGNUM.



La comunidad de Re-Volt America es concebida originalmente en el año 2017, bajo el nombre de Re-Volt Tournament. No fue hasta después de un par de años que esta sería renombrada a Re-Volt America, debido a la procedencia de sus jugadores, la cual era tanto de norte america como de sudamerica.

En el presente año 2023, Re-Volt America cuenta con una gran cantidad de jugadores activos, y con un sistema de puntuación único en la escena de Re-Volt y sus comunidades en línea. Este complejo sistema de puntuación, y su funcionamiento sostenido durante los últimos 6 años, son la base del problema que busca solucionar este proyecto de título.

Con el pasar del tiempo, este sistema se ha convertido en algo muy difícil de mantener para los administradores de la comunidad, tanto a nivel logístico como técnico.

1.4. Contexto del Problema

1.4.1. Re-Volt y Partidas en Línea

Como ya se mencionó anteriormente, Re-Volt es un videojuego de carreras el cual, gracias al surgimiento de RVGL y sus comunidades impulsoras, es jugado mayoritariamente en línea. Pero, ¿a qué nos referimos con "jugar en línea"? Para poder entender este concepto, tenemos que ir a lo que es una carrera en términos conceptuales, y las implicaciones que estas conllevan dentro de un contexto competitivo.

Para poder jugar en línea, cada jugador debe elegir un nombre de usuario, el cual puede incluso variar de partida en partida. Esto se hace una vez que ingresa al juego y avanza en el menú hasta llegar al selector de nombre de usuario en forma de neumático. El nombre que el jugador ingrese aquí será el nombre de usuario con el que se identificará a la hora de ser ingresado a los resultados de cada carrera en la que participe. El selector de nombre de usuario se puede apreciar a continuación en la figura FIGNUM.



1.4.1.1. Autos y Categorías

En Re-Volt, existen diferentes tipos de autos que pueden ser elegidos por el jugador. En el juego original, estos autos se clasifican en categorías según su velocidad máxima, aceleración, peso, y desempeño general en pista. Las categorías originales, ordenadas desde los autos más lentos, hasta los más rápidos, son las siguientes:

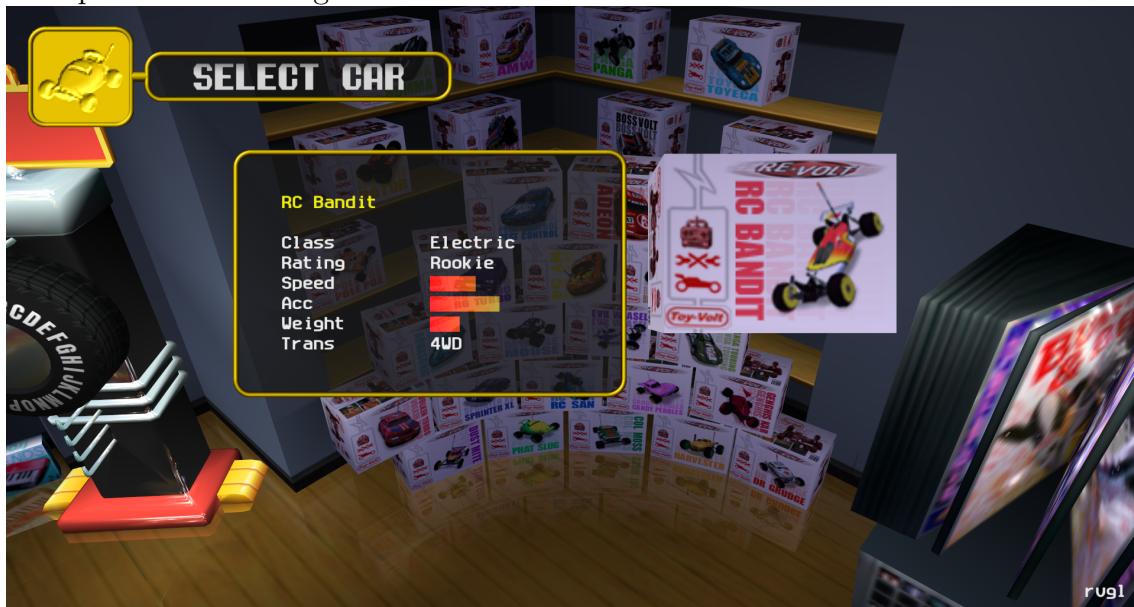
- Rookie (Novato).
- Amateur (Amateur).

- Advanced (Avanzado).
- Semi-Pro (Semi-Profesional).
- Pro (Profesional).

Además de las categorías originales, también existen categorías especiales que han surgido a partir del contenido creado por la comunidad de Re-Volt y RVGL. Estas categorías son las siguientes:

- Super-Pro
- Clockwork

La categoría o "Rating" de cada auto puede apreciarse desde el menú del juego, tal como se puede ver en la figura FIGNUM.



De manera habitual, el anfitrión define una categoría de auto con la cual se jugará la sesión. De esta forma, todos los jugadores que se conecten deberán utilizar autos de la categoría definida.

Teniendo en cuenta lo anterior, las sesiones suelen ser nombradas a partir de su categoría asociada. Por ejemplo, si para una sesión se decide jugar autos de la categoría "Pro", es normal que esta sea titulada "Pros Session", o "Pro Races" al momento de ser anunciada.

1.4.1.2. Sala de Espera

Una vez que el usuario elige su auto, este es llevado a la sala de espera, la cual viene representada por la figura FIGNUM.



Una vez llegados a este punto, sólo se debe esperar a que el anfitrión de la sesión de inicio a las carreras.

1.4.1.3. Carreras en Línea

Normalmente, en las partidas multijugador, suelen jugarse muchas carreras de manera consecutiva. A estas series de partidas en línea se les conoce como "sesiones". Cada sesión de Re-Volt consiste en una cantidad predefinida de carreras en pistas determinadas y con cierta clase de autos. Estas determinaciones las realiza el anfitrión de la partida, quien las comunica públicamente de manera oportuna para que todos aquellos que deseen participar tengan en cuenta todas las características de la sesión que van a jugar.

Las siguientes dos ilustraciones presentan las instancias clave dentro del juego. En la ilustración FIGNUM, se puede apreciar la perspectiva del jugador al momento de jugar Re-Volt. Luego, en la ilustración FIGNUM, se puede ver la tabla de resultados que se muestra por pantalla a medida que los corredores finalizan la carrera.



1.4.2. Re-Volt: OpenGL y los Session Logs

Para ayudar a llevar una cuenta fiable de todas las carreras jugadas, RVGL introduce una funcionalidad que permite a los jugadores obtener un registro escrito de los resultados de cada carrera. Este registro viene en forma de un archivo separado por comas, el cual es conocido por la comunidad como Session Log. Este es el archivo que utilizan los organizadores y administradores de RVA para calcular los resultados oficiales de las sesiones.

En la ilustración FIGNUM, presentada a continuación, puede apreciarse un extracto del Session Log de una sesión cualquiera, el cual ha sido importado desde Microsoft Excel para una visualización más clara. Este extracto representa una sola de las carreras jugadas en la sesión. Entiéndase que, inmediatamente después de la última línea, vendría

escrita la carrera siguiente, y así sucesivamente con todas las demás.



The screenshot shows a Microsoft Excel spreadsheet titled "session_2023-12-02_16-03-24.csv". The table has columns A through G. Rows 1-3 show session metadata: Version (RVGL 23.1030a1), Session (12/02/2023 16:05), and Results (School's Out! 1 R). Row 4 defines the columns: #, Player, Car, Time, BestLap, Finished, and Cheating. Rows 5-15 list player results: 1 B (Genghis Kar, 0.090741, 0.032639, TRUE, FALSE); 2 ROD (Naranja Turbo, 0.096991, 0.036088, TRUE, FALSE); 3 KILABARUS (HSF-1, 0.101331, 0.039039, TRUE, FALSE); 4 WATYNECC (Genghis Kar, 0.098819, 0.036053, TRUE, FALSE); 5 URV (Condor GRV, 0.099676, 0.040891, TRUE, FALSE); 6 SWIMPY (Tesla, 0.101146, 0.036169, TRUE, FALSE); 7 INS (Condor GRV, 0.101678, 0.041389, TRUE, FALSE); 8 INIGO (Albatross GT, 0.104792, 0.034352, TRUE, FALSE); 9 BALESZ (RC Phink, 02:23:026, 0.036829, TRUE, FALSE); 10 NARU (Ciagnik, 0.110694, 0.041366, TRUE, FALSE); 11 ANTONIUS (Volken Turbo, 0.114005, 0.03463, TRUE, FALSE).

	A	B	C	D	E	F	G
1	Version	RVGL 23.1030a1	P2P	Client			
2	Session	12/02/2023 16:05	INS	Arcade	3	TRUE	
3	Results	School's Out! 1 R		11			
4	#	Player	Car	Time	BestLap	Finished	Cheating
5	1	B	Genghis Kar	0.090741	0.032639	TRUE	FALSE
6	2	ROD	Naranja Turbo	0.096991	0.036088	TRUE	FALSE
7	3	KILABARUS	HSF-1	0.101331	0.039039	TRUE	FALSE
8	4	WATYNECC	Genghis Kar	0.098819	0.036053	TRUE	FALSE
9	5	URV	Condor GRV	0.099676	0.040891	TRUE	FALSE
10	6	SWIMPY	Tesla	0.101146	0.036169	TRUE	FALSE
11	7	INS	Condor GRV	0.101678	0.041389	TRUE	FALSE
12	8	INIGO	Albatross GT	0.104792	0.034352	TRUE	FALSE
13	9	BALESZ	RC Phink	02:23:026	0.036829	TRUE	FALSE
14	10	NARU	Ciagnik	0.110694	0.041366	TRUE	FALSE
15	11	ANTONIUS	Volken Turbo	0.114005	0.03463	TRUE	FALSE

A continuación, la tabla TABLENUM es una especificación de los elementos relevantes de un Session Log. Algunos de dichos elementos son utilizados por RVA para calcular los resultados acumulados de las sesiones, para así obtener una tabla con posiciones finales para cada jugador al final de cada sesión.

Elementos de un Session Log	
Id	Descripción
B1	Versión de RVGL de la sesión jugada.
C1-D1	Protocolo de conexión de la sesión.
B2	Fecha y hora de apertura de la sesión.
C2	Nombres del jugador anfitrión de la sesión.
D2	Tipo de colisión entre autos, seleccionado por el anfitrión.
E2	Número de vueltas de esta carrera.
B3	Nombre de la pista de esta carrera.
C3	Número de jugadores que comenzaron la carrera. Si un jugador se desconecta en mitad de la carrera, este número no se ve afectado.
B5-B15	Nombres de los jugadores que participaron de la carrera, agregados de arriba hacia abajo por orden de llegada a la línea de meta.
C5-C15	Nombres de los autos utilizados por cada jugador en esta carrera.
D5-D15	Tiempo total de la carrera de cada jugador.
E5-E15	Mejor vuelta de cada jugador.
F5-F15	TRUE si el jugador finalizó la carrera. FALSE si el jugador se desconectó en mitad de la carrera.
G5-G15	TRUE si los archivos locales del jugador no corresponden a los archivos del anfitrión. FALSE si estos coinciden. Por ejemplo, "Cheating" es TRUE si un jugador modificó localmente la velocidad de su auto.

1.4.3. Funcionamiento Interno de Re-Volt America

1.4.3.1. Sistema de Temporadas, Rankings y Sesiones

Históricamente, Re-Volt America se ha dedicado a organizar sesiones multijugador de Re-Volt para su público, así como también se ha preocupado de llevar la cuenta de los resultados de cada carrera que se ha jugado en ellas. Esto lo hace mediante un sistema de temporadas.

Todos los jugadores que, en algún momento u otro, han participado en las sesiones online organizadas por Re-Volt America, han sido indexados en la base de datos de jugadores que mantiene la comunidad. De esta forma, sus victorias, puntos y otras estadísticas asociadas han sido preservadas a lo largo del tiempo.

Para conseguir un sistema atractivo para sus jugadores, Re-Volt America organiza sus sesiones en una serie de rankings, los cuales consisten en 28 sesiones multijugador cada uno, jugándose una sesión al día. Al cumplirse 6 de estos rankings, se completa lo que en RVA se conoce como una temporada.

Cada sesión organizada por RVA consiste en 20 carreras, las cuales se juegan en 20 pistas diferentes. Es por esto que, en una sesión, se producen 20 sets de resultados (1 por carrera).

Cada año, en promedio, se inician dos temporadas, y cada una es nombrada en base a su año de inicio, y su periodo en relación a otras temporadas. Por ejemplo, si en el año 2023 se da inicio a una temporada en el mes de enero, esta vendría a llamarse "2023", ya que terminará dentro del mismo año en el que inició. Por otra parte, si la temporada

inicia en el mes de diciembre de 2023, esto quiere decir que terminará dentro del año 2024, por lo que pasaría a llamarse "2023-24".

Es así como Re-Volt America, desde el año 2017, hasta el presente, ha conseguido consolidarse como la comunidad de Re-Volt predilecta para los jugadores tanto de norteamérica como América latina.

1.4.3.2. Paquete de Contenido de RVA

Re-Volt America cuenta con un paquete de contenido extra para RVGL, el cual es construido por sus administradores y organizadores al principio de cada temporada.

Este paquete consiste en una selección de autos y pistas para RVGL hechos por la comunidad, los cuales tienen el objetivo de extender el contenido original del juego, y así proveer a los usuarios con una experiencia más enriquecedora y variada al momento de jugar en línea.

El paquete de contenido de RVA, más conocido como RVA Pack, es, técnicamente, un conjunto de archivos correspondientes a los autos y pistas que este busca agregar. Estos archivos son mandatorios para poder participar de las sesiones organizadas por RVA. Esto quiere decir que, para que un usuario pueda entrar a una sesión de RVA, este debe tener instalado el RVA Pack en su instalación de RVGL local.

1.4.3.3. Sistema de Puntuación

Como se ha aludido a lo largo del estudio del problema, la comunidad de Re-Volt America cuenta con su propio sistema de puntuación, el cual le sirve para generar estadísticas por jugador, tablas de resultados al final de cada sesión, y demás métricas que, finalmente, promueven la competitividad y enriquecen la experiencia de los usuarios.

El funcionamiento de dicho sistema de puntuación es el siguiente: dependiendo de la posición de cada jugador, al final de cada carrera, a estos se les asigna un puntaje. El mapeo de puntos para una carrera con menos de 10 participantes viene definido por la siguiente función que relaciona posiciones con puntos, respectivamente:

$$f_1 = \begin{cases} pos & f_1(pos) \\ 1 \rightarrow 15 \\ 2 \rightarrow 12 \\ 3 \rightarrow 10 \\ 4 \rightarrow 7 \\ 5 \rightarrow 5 \\ 6 \rightarrow 4 \\ 7 \rightarrow 2 \\ 8 \rightarrow 2 \\ 9 \rightarrow 1 \end{cases}$$

Por otro lado, en caso de que la carrera cuente con 10 jugadores o más, el mapeo de puntos sería el siguiente:

$$f_2 = \begin{cases} pos & f_2(pos) \\ 1 \rightarrow 20 \\ 2 \rightarrow 16 \\ 3 \rightarrow 12 \\ 4 \rightarrow 10 \\ 5 \rightarrow 8 \\ 6 \rightarrow 8 \\ 7 \rightarrow 6 \\ 8 \rightarrow 4 \\ 9 \rightarrow 2 \\ 10 \rightarrow 2 \\ 11 \rightarrow 1 \\ 12 \rightarrow 1 \\ 13 \rightarrow 1 \\ 14 \rightarrow 1 \\ 15 \rightarrow 1 \\ 16 \rightarrow 1 \end{cases}$$

Por lo tanto, el conjunto que define los posibles puntajes obtenibles viene definido de la siguiente manera:

$$P = \{1, 2, 4, 5, 6, 7, 8, 10, 12, 15, 16, 20\}$$

Si se tiene que " n " es la cantidad de jugadores que participaron de una carrera, entonces:

$$A = \{1, \dots, n\} \subset \mathbb{N}$$

$$n = |A|$$

Teniendo en cuenta lo anterior, es posible definir una función que, al recibir una posición final de un jugador en una carrera determinada, retorne la cantidad de puntos correspondientes a dicha posición. Esta función vendría definida a continuación, en donde " pos " es la posición final del jugador en la carrera en cuestión.

$$f : pos \in A \rightarrow \begin{cases} f_1(pos) & \text{if } n < 10 \\ f_2(pos) & \text{if } n \geq 10 \end{cases} \in P$$

Al finalizar una sesión, los puntajes de cada jugador en cada carrera se suman, lo que entrega un total de puntos por jugador. Una vez sumados, los puntajes finales son normalizados, y sometidos a diferentes procesos de ajuste propios del sistema de RVA. Una vez ajustados, estos pasan a ser los puntajes finales de la sesión de cada jugador.

1.4.3.4. Cálculo de Resultados

Una vez procesados, los resultados son llevados a una tabla final, tal como se puede apreciar en la figura FIGNUM.

#	Fecha/Sesión	PF	MEX	FG2	S02	DEL	TH1	S01	CT2	SC1	WS2	MD	SM2	FG1	PV	GEX	TGS	TSZ	RT	MM	M3	CTI	PP	PA	CC	MP	PO	
122	5	17/02/2023	1	BGM	1	3	1	4	3	3	1	4	1	1	2	3	2	1	1	1	1	2	5	2.11	215	19	0.95	9.68
					<i>Rebound 4X4</i>																							
123	2	TOMDOM	3	2	4	5	2	7	4	3	4	2	1	7	4	6	5	4	2	3	5	2	3.75	139	20	1	3.71	
124					<i>B59</i>																							
125	3	ROG_97	1	3	7	1	4	2	1	2	7	7	5	5	2	4	3	3	4	1	3.44	140	18	0.9	3.66			
126					<i>Albatross GT</i>																							
127	4	OSVALDOX	6	4	6	2	3	8	8	4	5	6	1	5	3	6	5	5	2	4	4.61	143	18	0.9	2.79			
128					<i>Manko XS</i>																							
129	5	LOQ	5	5	8	3	10	9	8	7	7	8	6	2	8	3	6	7	6	4	9	6.37	112	19	0.95	1.67		
130					<i>Panorama Junior</i>																							
131	6	WALUIGIMB	7	6	7	1	4	1	7	9	3	9	3	1	6	7	5.07	108	14	0.7	1.49							
132					<i>Trirraut Get Air</i>																							
133	7	ESTEBANMZ	2	8	5	5	5	6	9							2	2	4	1	1	8	4.46	84	13	0.65	1.22		
134					<i>Panorama Col. Moss</i>																							
135	8	ASURA	4	8	6	2	7	8	6	2	10	3	4	8	7	8	7	5	6	3	6	5.79	73	19	0.95	1.20		
136					<i>Hot Spl. MosP sprinter XL</i>																							
137	9	ALE OWO	2	7	5	6	8	6	5	6	5											5.56	37	9	0.45	0.30		
138					<i>RC Bandit</i>																							
139	10	TIOROTTI											5	6	8	4	3	4	8			5.43	28	7	0.35	0.18		
140					<i>Sprinter XL</i>																							
141	11	BOMB FACTORY	8	9	9	9	9	10														9.00	7	6	0.3	0.02		
142					<i>Sprinter XE1 Gekko Volken Turbo</i>																							
143	12	BURDANG																				3.00	7	1	0.05	0.01		
144					<i>Dr. Gru</i>																							
145	13	PHON7X																				7.00	2	1	0.05	0.00		
146					<i>Volken T</i>																							
147																												

Parámetros				
Nombre	Abreviación	Descripción		
Posición Promedio	PP	La suma de las posiciones del jugador, dividida por su cantidad de carreras corridas.		
Puntaje Acumulado	PA	La suma total de los puntos correspondientes a las posiciones obtenidas por el jugador.		
Carreras Corridas	CC	La cantidad de carreras corridas por el jugador.		
Multiplicador por Participación	MP	La cantidad de carreras corridas por el jugador, dividida por la cantidad de carreras totales de la sesión.		
Puntaje Oficial	PO	El puntaje acumulado del jugador, multiplicado por un factor normalizador de 0.1.		

De acuerdo con la tabla anterior, los cálculos de PA, PP y MP pueden ser representados utilizando las siguientes fórmulas, respectivamente, en donde "pos" es la posición del jugador en cada carrera, y "tot" la cantidad total de carreras de la sesión.

$$PA = \sum_{i=1}^{CC} x = x + f(pos_i)$$

$$PP = \frac{\left(\sum_{i=1}^{CC} x = x + pos_i \right)}{CC}$$

$$MP = \frac{CC}{tot}$$

A partir de estos parámetros, podemos calcular el puntaje oficial, o PO, obtenido por el jugador en la sesión en cuestión.

$$PO = \left(\frac{PA}{PP} \cdot MP \right) \cdot 0.1$$

1.4.3.5. Multiplicadores por Auto

Como ya se mencionó anteriormente, al principio de cada temporada, se agregan y se eliminan autos del pack de RVA. Asimismo, RVA introduce el concepto de multiplicadores para dichos autos por temporada.

De acuerdo con el sistema de RVA, el puntaje obtenido por un jugador al finalizar una carrera es multiplicado por el multiplicador del auto que utiliza en dicha carrera. De esta forma, el puntaje obtenido por el jugador incrementa o disminuye dependiendo del multiplicador asociado a su auto.

Los multiplicadores de cada auto son un reflejo de su potencial para desempeñarse en carreras en línea en comparación a los demás autos de su categoría. Por ejemplo, si un auto es increíblemente rápido, fácil de manejar y no tiene dificultad alguna, entonces a este se le asocia con un multiplicador bajo, para así nivelar los puntos que, potencialmente, pueda obtener un jugador al utilizarlo. Por el contrario, si un auto es muy lento, difícil de manejar, y simplemente es peor que los demás autos de su categoría, entonces se le asocia con un multiplicador alto, para así otorgar puntos extra a quien lo utilice.

La decisión de qué autos son buenos y qué autos son malos, en comparación con los demás de su categoría, es tomada por la administración de RVA en base a pruebas de manejo para los autos, y la experiencia previa en temporadas anteriores. A partir de esta decisión, se le otorga un multiplicador a cada auto.

Considerando los distintos multiplicadores que puede tener cada auto, la fórmula para el cálculo del PA es modificada ligeramente, en donde "*mul*" es el multiplicador del auto asociado.

$$PA = \sum_{i=1}^{CC} x = x + f(pos_i) \cdot mul_i$$

Como se explicó anteriormente, existen varias categorías de autos en Re-Volt. Si bien las sesiones de RVA corresponden a una categoría a la vez, el sistema admite que jugadores elijan autos hasta 3 clases por debajo de la categoría de la sesión. Por ejemplo, si la sesión es publicada como de categoría "Pro", un jugador puede elegir un auto de categoría "Semi-Pro".

De acuerdo con el sistema de RVA, por cada categoría de diferencia entre la sesión y el auto elegido por el jugador, a este se le otorga un bono de 0.25, el cual se adiciona al

multiplicador de su auto. Si en el ejemplo anterior la sesión era "Pro" y el auto "Semi-Pro", entonces al multiplicador del auto se le suma directamente 0.25. Si la diferencia fuese de 2 categorías, entonces se le sumaría 0.5.

Teniendo en cuenta lo anterior, la fórmula es modificada nuevamente. En este caso, la variable "*delta*" representa el bono asignado al jugador en la carrera.

$$PA = \sum_{i=1}^{CC} x = x + f(pos_i) \cdot (mul_i + delta_i)$$

1.5. Sistema Actual

1.6. Problemática Actual

Hoy en día, RVA utiliza una aplicación llamada "RVA-Points", la cual está hecha exclusivamente para procesar los Session Logs generados por RVGL y transformarlos en archivos separados por comas que contienen los resultados de las sesiones en el formato de RVA.

Una vez procesados los Session Logs, RVA lleva la cuenta de sus temporadas y jugadores utilizando Microsoft Excel como una pseudo base de datos para registrarlos, y Dropbox para la sincronización de archivos entre administradores.

Cada temporada, los administradores manejan un documento maestro de Excel, el cual, en resumidas cuentas, es utilizado para registrar los resultados de cada sesión y, a la vez, llevar la cuenta de los puntos por jugador. Esto se consigue utilizando macros y demás componentes característicos de Excel.

Los documentos relacionados a cada temporada son archivados utilizando una carpeta compartida en Dropbox.

Actualmente, el proceso diario para calcular y publicar los resultados de cada sesión es llevado a cabo, de manera semiautomática, por los administradores de RVA. Este proceso consiste en los siguientes pasos:

1. Una vez finalizada la sesión, se busca el Session Log generado por RVGL, y este llega a los administradores de RVA.
2. Se revisan los nombres de usuario manualmente. Esto se hace en caso de que algún jugador haya utilizado un nombre ligeramente distinto al que usa normalmente.
3. El Session Log de la sesión es importado desde "RVA-Points". Este programa procesa el Session Log, y lo transforma en resultados ordenados en el formato de RVA.
4. También desde "RVA-Points", los resultados procesados son exportados a otro archivo separado por comas.
5. Se copian los contenidos del archivo exportado al documento maestro de la temporada actual.

6. Se agrega cierta información de manera manual, como la fecha y número de la sesión.
7. Se revisa manualmente que los nombres de usuario de la sesión correspondan a los nombres que ya figuran en el documento maestro.
8. Se publica una fotografía de la tabla de resultados de la sesión en el Discord de RVA.
9. Se publica una fotografía del ranking actualizado con los datos de la sesión en el Discord de RVA.

A raíz de este largo y tedioso proceso de cálculo y manejo de resultados, se ha dado origen a este proyecto de título, como una oportunidad de mejorar dicho proceso, y poder ofrecer así una mejor experiencia de usuario tanto para los jugadores de RVA, como para los organizadores de la comunidad, quienes dedican su tiempo y esfuerzo a mantener estos registros históricos al día para todos.

1.6.1. Diagrama de la Situación en la Actualidad

Teniendo en cuenta el contexto del problema, la situación en la actualidad puede representarse a través del siguiente diagrama BPMN:



1.7. Propuesta de solución

Debe explicar en términos generales cómo las TIC pueden resolver o mejorar la(s) problemática identificada y quienes serán los usuarios principales, que tecnología se utilizaría para dar soporte a la propuesta.

1.8. Soluciones similares disponibles

A continuación, se describen las soluciones disponibles que pueden ser catalogadas como similares al proyecto que se presenta.

1.8.1. Aplicación para Cálculo de Puntos de Re-Volt I/O

Existe un trabajo similar hace varios años, el cual fue desarrollado por la comunidad europea de RVGL: Re-Volt I/O.

Este trabajo se trata de una aplicación web que permite a los administradores de Re-Volt I/O importar los resultados de las sesiones multijugador, y poder visualizarlos dentro de la misma página; sin embargo, dicho proyecto no cuenta con ningún tipo de interconexión entre resultados, lo que quiere decir que cada sesión de carreras publicada en dicho sitio es independiente de otras. Debido a lo anterior, este proyecto no cuenta con perfiles de usuario, y por ende no permite visualizar estadísticas de ningún tipo, ni tampoco relacionar tablas de resultados entre si.

En general, esta solución fue concebida para ser algo simple y rápido que sirviera para calcular y renderizar resultados de manera oportuna, y no con una visión de persistencia en mente. A continuación, en la ilustración FIGNUM, puede apreciarse la tabla de resultados generada por la aplicación web de Re-Volt: I/O.



1.9. Justificación del Problema

Actualmente, no existe ninguna plataforma que permita a los usuarios registrarse o visualizar resultados y estadísticas de las sesiones multijugador organizadas por la comunidad. Lo anterior significa que, cuando se juegan partidas online, no es posible obtener de manera automática rankings, estadísticas por vehículo y menos por usuario, ya que no hay forma de vincular, de manera definitiva, a los jugadores a través de un perfil dentro del juego. Es por ello que un usuario de nuestra comunidad no puede saber cuántas carreras o sesiones ha ganado con cierto auto, o en cierta pista, cómo se compara al resto, qué porcentaje de carreras ha perdido, ganado, etc.

Capítulo 2

Proyecto

2.1. Objetivo General del Proyecto

Desarrollar una aplicación web para Re-Volt America, la comunidad del continente americano formada alrededor del videojuego de carreras Re-Volt, 1999.

2.2. Objetivos Específicos del Proyecto

1. Elaborar una propuesta que consiga atender las necesidades y problemas de los usuarios de Re-Volt America, en relación con el almacenamiento y visualización de resultados de partidas online, además de proveer visibilidad a la comunidad en general.
2. Diseñar la solución de software de procesamiento de datos de sesiones multijugador de Re-Volt, creando interfaces que les permitan a los jugadores visualizar los resultados oficiales de las sesiones de carreras en línea, además de estadísticas personales.
3. Implementar la aplicación web, la cual permitirá a los organizadores de sesiones de carreras en línea subir y publicar los resultados de dichas carreras, además de realizar el lazo entre jugadores y cuentas de usuario de esta. El software implementado permitirá, a su vez, procesar dicha información subida a la web, y así mostrar a los usuarios finales una vista clara de sus resultados en carreras y sus estadísticas personales.

2.3. Metodología de Desarrollo

Para poder definir la metodología de desarrollo a utilizar, primero se debe tener en cuenta la tabla de riesgos asociada, la cual puede apreciarse a continuación.

Tabla de Riesgos			
Experiencia en el Problema	Alta	X	Se tienen años de experiencia con la comunidad de RVA y su sistema interno.
	Baja		
Tamaño del Problema	Grande	X	La cantidad de funcionalidades a implementar es muy alta.
	Pequeño		
Complejidad del Problema	Complejo	X	El sistema de puntos de RVA es difícil de comprender y manejar completamente.
	Simple		
Tamaño del Software	Grande	X	El software a construir requiere muchas funcionalidades.
	Pequeño		
Complejidad Software	Complejo	X	El software debe implementar cálculos complejos (ratios, promedios, etc.).
	Simple		
Experiencia Software	Alta	X	Se tiene una alta experiencia desarrollando software para RVA y su sistema de puntos.
	Baja		
Modularidad Funcional	Existe	X	Las funcionalidades pueden implementarse por separado y luego integrarse.
	No existe		

Según la evaluación del proyecto, se concluyó que este tiene un riesgo total asociado bajo, por lo que se contaría con la libertad de utilizar cualquier tipo de metodología de desarrollo para llevarlo a cabo. Como se cuenta con esta libertad de elección, se busca sacar provecho de ella, y se ha seleccionado una metodología de desarrollo iterativa, a través de la cual se desarrollará en ciclos. Estos ciclos permitirán al software evolucionar a medida que se recibe retroalimentación por parte de los usuarios, corrigiendo errores y mejorando detalles a medida que se progresó en el desarrollo de la aplicación.

Según el contexto en el que se desarrollará esta aplicación, los usuarios de la comunidad tendrán una importante incidencia en el testeo y uso diario de la misma, lo cual implica que estarán presentes durante el proceso de implementación de varias de las funcionalidades que se pretenden lograr con esta propuesta. Se sabe también con antelación que los usuarios estarán dispuestos a ayudar con las pruebas y el testeo de la aplicación. Esta opción fue seleccionada ya que:

1. Permite más flexibilidad en cuanto a lo que se planea desarrollar como aplicación web, debido a que a medida que avance el proyecto es muy probable que surjan cambios o nuevas ideas a partir de la retroalimentación recibida de la comunidad.
2. Debido a la naturaleza del proyecto, utilizar una metodología iterativa es muy beneficioso ya que ésta permitirá una constante supervisión de los cambios realizados al software, lo que finalmente se traducirá en menos errores, y un producto final que se ajuste y esté a la altura de las necesidades de la comunidad.
3. Con una metodología iterativa será posible contar con prototipos que la comunidad pueda comenzar a utilizar, y a su vez adaptarse a las interfaces y funcionalidades de la aplicación mientras esta se encuentra en desarrollo. Esto es muy importante ya que las sesiones multijugador suelen organizarse a diario, por lo que mientras antes se cuente con una solución funcional, antes podrá la comunidad comenzar a llevar un registro histórico de las partidas online que organiza.

2.4. Técnicas y Notaciones

- Diagrama de Casos de Usos.
- BPMN para modelar el proceso de negocio actual.
- Carta Gantt para la planificación inicial del proyecto.
- Patrón de diseño MVC (Modelo, Vista, Controlador).

2.5. Estándares de Documentación

- Adaptación Basada en IEEE Software Test Documentation Std 829-1998.
- Adaptación Basada en IEEE Software Requirements Specifications Std 830-1998.

2.6. Software, Frameworks y Lenguajes Utilizados

A continuación se lista el software, frameworks y lenguajes de programación, marcado y estilos utilizados para la realización de este proyecto.

Para efectos del siguiente listado, los nombres de las herramientas, frameworks y lenguajes se han redactado en negrita, seguidos de paréntesis en itálica que contienen el número de la versión asociada a cada ítem.

Lenguajes

- **Ruby** (*3.2.2*): Lenguaje de programación de alto nivel.
- **HAML** (*6.2.3*): Lenguaje de marcado para la abstracción de HTML.
- **Sass** (*6.0*): Lenguaje de extensión para CSS.

Software

- **MongoDB** (*7.0.3*): Base de datos orientada a documentos JSON.
- **Redis** (*7.0.12*): Almacenamiento en memoria, utilizado para el caché de datos.
- **RubyMine** (*2023.2.2*): Entorno de desarrollo integrado especializado para el trabajo con aplicaciones en Ruby, específicamente para Ruby on Rails.
- **Rake** (*13.1*): Librería de Ruby para la definición de tareas interdependientes.
- **MongoDB Compass** (*1.39.0*): Visor para bases de datos de MongoDB.
- **RedisInsight** (*2.30.0*): Visor para el almacenamiento del caché en Redis.
- **Docker Desktop** (*4.21.0*): Visor y gestor de contenedores de Docker, en formato de aplicación de escritorio multiplataforma.
- **NodeJS** (*16.13.0*): Entorno de servidor multiplataforma utilizado para la conversión de archivos en runtime.

- **Yarn** (*1.22.21*): Gestor de paquetes para JavaScript.
- **Docker** (*24.0.2*): Tecnología que permite crear y utilizar contenedores. Para efectos de este proyecto, es utilizado con el fin de probar el software desarrollado en distribuciones de Linux determinadas.
- **Termius** (*8.7.2*): Cliente SSH.
- **Git/Git Bash** (*2.34.1*): Sistema de control de versiones.
- **Ubuntu LTS** (*18.04.6*): Subsistema de Linux para Windows.

Frameworks

- **Ruby on Rails** (*7.1*): Framework para desarrollo de aplicaciones web full-stack.
- **Jekyll** (*4.0.0*): Framework para desarrollo de aplicaciones web estáticas escrito en Ruby.
- **Bootstrap** (*4.4.1*): Framework para la creación, manejo de elementos visuales y la responsividad en aplicaciones web.

Capítulo 3

Factibilidad

3.1. Factibilidad Técnica

3.1.1. Conocimientos de los Usuarios

Para el correcto funcionamiento de la aplicación propuesta, es de esperar que los distintos miembros del staff de RVA, quienes serán los principales usuarios del software, deban tener determinados conocimientos para poder operarla con éxito.

Gracias a que, naturalmente, el staff de RVA lleva un largo tiempo manejando la comunidad y los resultados de las sesiones organizadas, no hace falta mayor capacitación técnica en cuanto al funcionamiento del cálculo de puntos, multiplicadores de autos, y demás características específicas del sistema de RVA. Para efectos de factibilidad técnica, el staff ya cuenta con los conocimientos necesarios para poder migrar a un sistema que sólo busca mejorar y facilitar los procesos actuales.

En cuanto a la aplicación propuesta en este proyecto y su funcionamiento operacional, el staff será habituado al nuevo sistema mediante un video de inducción a la nueva plataforma web. También se asignará un periodo de prueba para que puedan utilizar la página ellos mismos, y así puedan adaptarse fácilmente.

3.1.2. Disponibilidad Profesional

Para el desarrollo de la aplicación propuesta, se necesita del trabajo de un profesional en el área del desarrollo de software, el cual sea capaz de satisfacer las necesidades de RVA y cumplir los objetivos de desarrollo propuestos.

Para efectos de este proyecto, se cuenta tanto con el tiempo profesional, como también con los conocimientos técnicos requeridos.

Adicionalmente, se cuenta con el equipo físico para poder desarrollar la aplicación, como puede ser un computador, acceso a internet y demás software para desarrollo. A continuación, se presentan tablas de especificación de los equipos físicos y el software con el que se cuenta para desarrollar la aplicación.

Equipos Físicos	
Nombre	Acceso
Windows PC	Equipo personal
MacBook Pro M1	Equipo personal

Software	
Nombre	Acceso
Notepad++	Software libre
MongoDB Compass	Software libre
RedisInsight	Software libre
Git/Git Bash	Software libre
Ubuntu LTS	Software libre
RubyMine	Licencia de estudiante
Termius	Licencia de estudiante
Microsoft Excel	Licencia de estudiante

3.1.3. Despliegue y Servidor

Para realizar el despliegue de la aplicación a desarrollar, se ha escogido un servidor de tipo VPS (Servidor Virtual Privado). Las especificaciones técnicas de dicho servidor son las siguientes:

VPS	
Característica	Detalle
Proveedor	DigitalOcean
Región	Nueva York
Sistema Operativo	Ubuntu 22.04 (LTS) x64
Tipo de CPU	Intel Regular
Número de vCPUs	1 CPU
Memoria	2 GB
Almacenamiento (SSD)	50 GB
Transferencia	2 TB

El software instalado en este servidor para el despliegue de la aplicación es el mismo especificado en la sección 2.6.

3.2. Factibilidad Operativa

En cuanto a la factibilidad operativa de este proyecto, se sabe que el staff de Re-Volt America tiene gran disposición al cambio, ya que el sistema antiguo no solamente les demanda demasiado tiempo y esfuerzo, sino que, además, resulta poco preciso y muy propenso a errores en su uso diario debido a la gran cantidad de pasos manuales que este conlleva.

La factibilidad operativa es fácilmente demostrada al comparar la solución propuesta con el sistema que esta pretende reemplazar. Vale decir que, actualmente, quienes hacen uso del sistema de hojas de cálculo maestras en Excel y la aplicación de RVA-Points para el procesamiento de las sesiones, verían su trabajo facilitado en todos los sentidos al contar con una plataforma que se encargue de llevar la cuenta de todo, procesar los resultados y mantener los rankings y temporadas al día de manera automática y confiable.

3.3. Factibilidad Económica

A continuación se presenta un detalle de tablas de costos y el flujo de caja asociado al proyecto.

3.3.1. Tablas de Costos

En esta sección se presentan dos tablas de costos relacionadas al proyecto. La primera tabla contiene todo el software utilizado para el desarrollo de este proyecto, mientras que la segunda tabla contempla los costos inherentes de RVA, combinando aquellos costos adquiridos que vienen de antes del despliegue a producción del software que se ha desarrollado.

Software		
Nombre	Acceso	Precio (anual)
Notepad++	Software libre	\$0
MongoDB Compass	Software libre	\$0
RedisInsight	Software libre	\$0
Git/Git Bash	Software libre	\$0
Ubuntu LTS	Software libre	\$0
RubyMine	Licencia de estudiante	\$0
Termius	Licencia de estudiante	\$0
Microsoft Excel	Licencia de estudiante	\$0

Costos de Producción		
Nombre	Proveedor	Precio (anual)
VPS	DigitalOcean	\$125.000
Mailer	Postmark	\$156.000
Dominio (rva.lat)	Namecheap	\$22.600
Git/Git Bash	Software libre	\$0
Ubuntu LTS	Software libre	\$0
RubyMine	Licencia de estudiante	\$0
Termius	Licencia de estudiante	\$0
Microsoft Excel	Licencia de estudiante	\$0

Cabe destacar que, dentro de los costos de producción, tanto el precio anual del VPS como el del dominio de RVA son costos adquiridos que han sido mantenidos desde antes

de la puesta en marcha de este proyecto. Su mención en la tabla anterior es puramente una formalidad.

3.3.2. Flujo de Caja

Para la redacción del flujo de caja, en primer lugar se hablará de los indicadores económicos utilizados para llegar a él, y luego se calculará el Valor Actual Neto (VAN) a 5 años de la puesta en marcha del proyecto.

La valorización del tiempo de trabajo será determinada a partir de promedios y estimaciones que pueden encontrarse hoy en día en el mercado del desarrollo de software. Las referencias serán mencionadas oportunamente.

3.3.2.1. Contexto e Indicadores Económicos

La Tasa Mínima Aceptable de Rendimiento (TMAR) para proyectos de desarrollo de software, a la cual llamaremos " r ", ha sido calculada para este proyecto a partir de la inflación promedio anual reportada por el Banco Central de Chile, y la prima de riesgo asociada a proyectos tecnológicos reportada por el Standish Group International en 2011 (CHAOS Manifesto).

Inflación Promedio Anual	4.3 %
Tasa Prima de Riesgo	21 %

Con la inflación promedio anual y la tasa prima de riesgo, es posible calcular el valor de la TMAR asociada al proyecto.

$$r = 0.04 + 0.21 + (0.04 \cdot 0.21) = 0.2584 \approx 26\%$$

3.3.2.2. Puesta en Marcha

A partir de los datos obtenidos en la subsección 3.3.2.1, es posible realizar un cálculo del valor actual neto (VAN) del proyecto. En este caso, todos los costos asociados figurarán como un ahorro para la comunidad de RVA, ya que debemos considerar que todo lo realizado vendría representar dinero que, potencialmente, la comunidad hubiese tenido que gastar en otro profesional y recursos de no haberse desarrollado el software en cuestión.

La valorización del tiempo empleado en el desarrollo de software a lo largo del proyecto está basada en el sueldo promedio de un analista programador en Chile según Talent.com (\$900.000 mensual; \$5.538 por hora), ajustado a las horas de trabajo efectivas empleadas en el proyecto, las cuales fueron 4 horas de trabajo efectivo durante 6 días de la semana por mes (4 semanas) de desarrollo.

$$\text{Desarrollo} = \$5.538 \cdot (4 \text{ horas} \cdot 6 \text{ días} \cdot 4 \text{ semanas}) = \$531.648$$

Además del tiempo valorizado, también se ha estimado una valorización de los costos de internet y electricidad asociados con el proyecto, basados en su costo promedio mensual y dividiéndolos por la mitad, para así obtener un valor realista.

$$Internet = \frac{\$20.000}{2} = \$10.000$$

$$Electricidad = \frac{\$40.000}{2} = \$20.000$$

A continuación, se presenta una tabla que contiene los gastos mensuales estimados de la puesta en marcha del proyecto:

Costos	
Desarrollo	\$531.648
Internet	\$10.000
Electricidad	\$20.600
Total	\$561.648

Una vez calculado el total de gastos para la puesta en marcha, es posible extraerlo a 4 meses y calcular la inversión inicial del proyecto a través de la siguiente fórmula.

$$I_0 = \$561.648 \cdot 4 \text{ meses} = \$2.246.592$$

3.3.2.3. Cálculo del Valor Actual Neto

Siguiendo con la lógica de la valorización de las horas de trabajo efectivas, se tiene que, una vez terminado el proyecto (al final de los 4 meses), la cantidad de horas efectivas necesarias para mantener el proyecto a lo largo del tiempo disminuirá sustancialmente.

Se dirá que, si antes eran 4 las horas efectivas en 6 días de la semana, ahora sólo será 1 hora por 5 días de la semana, ya que el trabajo requerido por la comunidad ya no será de desarrollo de software, sino que de mantenimiento, realización de mejoras oportunas, y soporte para la plataforma.

$$Mantenimiento = \$5.538 \cdot (1 \text{ horas} \cdot 5 \text{ días} \cdot 4 \text{ semanas}) = \$110.760$$

Además de lo anterior, se realiza un ajuste proporcional del costo de internet y electricidad en función de las valorizaciones del tiempo de desarrollo y mantenimiento calculadas.

$$\frac{Desarrollo}{Mantenimiento} = \frac{\$531.648}{\$110.760} = 4.8$$

$$Internet = \frac{\$10.000}{4.8} = \$2.083$$

$$Electricidad = \frac{\$20.600}{4.8} = \$5.208$$

Si proyectamos a 5 años, podemos calcular el Valor Actual Neto (VAN) de ahorro para el proyecto.

	0	1	2	3	4	5
Mantenimiento		\$1.329.120	\$1.329.120	\$1.329.120	\$1.329.120	\$1.329.120
Internet		\$24.996	\$24.996	\$24.996	\$24.996	\$24.996
Electricidad		\$62.496	\$62.496	\$62.496	\$62.496	\$62.496
Hosting		-\$126.312	-\$126.312	-\$126.312	-\$126.312	-\$126.312
Flujo		\$1.290.300	\$1.290.300	\$1.290.300	\$1.290.300	\$1.290.300
Inv. Inicial	\$2.246.592					
Flujo Total	\$6.451.500					

Como clarificación general para la siguiente fórmula, el número de años de proyección vendrá definido como " n ", y " C " representará los flujos de caja.

Asimismo, recordar también que " r " representa la TMAR del proyecto calculada en la subsección 3.3.2.1, e " I_0 " representa la inversión inicial calculada en la subsección 3.3.2.2.

$$VAN = I_0 + \sum_{t=i}^n \frac{C_t}{(1+r)^t} = \$5.646.624$$

[https://www.bcentral.cl/web/banco-central/contenido/-/detalle/informe-de-politica-monetaria-septiembre-2023 \(INFLACIÓN\)](https://www.bcentral.cl/web/banco-central/contenido/-/detalle/informe-de-politica-monetaria-septiembre-2023 (INFLACIÓN))

[https://www.immagic.com/eLibrary/ARCHIVES/GENERAL/GENREF/ChaosManifest_-2011.pdf \(TASA DE RIESGO\)](https://www.immagic.com/eLibrary/ARCHIVES/GENERAL/GENREF/ChaosManifest_-2011.pdf (TASA DE RIESGO))

<https://radio.uchile.cl/2023/04/13/cuentas-de-electricidad-de-hogares-de-mayor-consumo-subiran-entre-10-y-16/>

<https://www.ine.gob.cl/estadisticas/economia/transporte-y-comunicaciones/estructura-del-transporte-por-carretera/2020/05/16/conexiones-a-internet-en-hogares-del-pa>

<https://cl.talent.com/salary?job=analista+programador>

3.4. Conclusión de Factibilidad

Para concluir, podemos decir que en cuanto a los distintos tipos de factibilidad evaluados se cuenta con usuarios técnicamente capaces y con la disponibilidad profesional necesaria para el desarrollo del proyecto. Por el lado operativo, existe disposición al cambio por parte de los administradores y el staff en general. Económicamente, a partir de los costos y beneficios recopilados, se ha obtenido que existe un valor económico sustancial asociado al proyecto.

Gracias al análisis realizado en los puntos anteriores, se puede concluir que el proyecto es factible.

Capítulo 4

Arquitectura de Re-Volt America

Capítulo 5

Requerimientos del Software

5.1. Límites

- El software no permitirá relacionar automáticamente cualquier nombre de jugador con su perfil de usuario registrado en ella.
- El software no registrará estadísticas de manera retroactiva para jugadores no registrados.
- El software no permitirá visualizar carreras de manera individual.
- Las estadísticas por jugador no pueden ser editadas manualmente, sino que la modificación de estas debe ser un efecto de subir o eliminar sesiones.
- Las sesiones eliminadas no se podrán visualizar. Esto quiere decir que las sesiones que se eliminan no podrán ser accedidas nunca más una vez borradas.
- Los usuarios no podrán cancelar sus cuentas por si mismos.
- El software no permitirá editar los modelos de sesiones de manera manual.

5.2. Caracterización de los Usuarios

Como se ha mencionado a lo largo de este informe, los usuarios a quienes apunta este proyecto de título son aquellos que forman parte de la comunidad de Re-Volt America. Estos usuarios, de manera general, se caracterizan por:

- Pertenencia a un grupo etario entre 14 y 26 años.
- Competencias técnicas en uso de software similares.
- Familiaridad con el juego y el sistema de puntos de RVA.

Por otra parte, tenemos la definición de quienes pueden ser clasificados como usuarios del software, así como la frecuencia con la que utilizarán la plataforma y su rol dentro de la misma:

Usuarios		
Rol	Nivel	Descripción
Administrador	Primario	Utiliza todas las funciones de la página, como subir autos, pistas, sesiones y manejo general del software. Configura el software para el resto de los usuarios.
Organizador	Primario	Utiliza las funciones específicamente relacionadas a la subida y manejo de sesiones en la página.
Moderador	Secundario	Utiliza las funciones específicas de manejo de usuarios, tal como la edición de perfiles o aplicación de infracciones.
Jugador	Terciario	Utiliza la página sólo para visualizar la información que esta ofrece.

5.3. Objetivo General del Software

El sistema manejará información del proceso de cálculo de resultados de carreras para que la comunidad centralice los rankings y estadísticas por usuario dentro del mismo, es decir, reducirá el trabajo manual requerido actualmente para este procesamiento de resultados, y así hará más eficiente todo el proceso que conlleva mantener los rankings actualizados.

5.3.1. Objetivos Específicos del Software

- El sistema permite que los administradores de la comunidad puedan subir los archivos de resultados generados por RVGL y, de esta forma, los resultados son generados automáticamente dentro de la aplicación. Esto elimina el tiempo de los organizadores de calcular los resultados manualmente.
- El sistema permite que los usuarios puedan ver sus estadísticas en tiempo real, lo que elimina la necesidad de llevar la cuenta de manera manual por cada uno de ellos.
- El sistema permite enlazar los nombres de usuario utilizados en las sesiones multi-jugador de RVGL a perfiles dentro de la aplicación, lo cual hace posible la recopilación y atribución de métricas individuales por jugador, y agiliza la visualización de resultados.
- El sistema permite llevar un registro histórico de manera automática según se suben y se procesan los archivos de resultados en la aplicación, lo cual elimina completamente la necesidad de la comunidad de mantener toda esta información actualizada de manera manual.

5.4. Requerimientos Funcionales del Software

Módulo de Registro de Autos de Re-Volt America	
Id	Descripción
RF_01	La plataforma contará con un módulo de creación de autos. Dentro de este módulo, los autos deben ser relacionados con una temporada. No podrán existir autos con nombres duplicados dentro de la misma temporada. Los autos deben ser importados desde un archivo separado por comas, el cual debe contar con todos los parámetros obligatorios del modelo de auto. Sólo los administradores pueden crear autos.
RF_02	La plataforma contará con un módulo de visualización de autos. El listado estará separado por temporadas y por clases de autos. Este módulo estará disponible para cualquier tipo de usuario.
RF_03	La plataforma contará con un módulo de edición de un auto. Los autos deben contar con todos los parámetros obligatorios para ser modificados. Este módulo estará disponible sólo para los administradores.
RF_04	La plataforma contará con un módulo de eliminación de un auto. La eliminación no tendrá efectos secundarios a nivel de la base de datos, ya que la aplicación estará preparada para manejar excepciones cuando las entradas de corredores estén asociadas a un auto que no existe. Este módulo sólo estará disponible para administradores.
RF_05	La plataforma contará con un módulo de visualización de un sólo auto. En esta vista se podrán ver todos los parámetros del auto. Este módulo estará disponible para cualquier tipo de usuario.

Módulo de Registro de Pistas de Re-Volt America	
Id	Descripción
RF_06	La plataforma contará con un módulo de creación de pistas. Dentro de este módulo, las pistas deben ser relacionados con una temporada. No podrán existir pistas con nombres duplicados dentro de la misma temporada. Las pistas deben contar con todos los parámetros obligatorios del modelo de pista. Sólo los administradores pueden crear pistas.
RF_07	La plataforma contará con un módulo de visualización de pistas. El listado estará separado por temporadas y paginado. Este módulo estará disponible para cualquier tipo de usuario.
RF_08	La plataforma contará con un módulo de edición de una pistas. Las pistas deben contar con todos los parámetros obligatorios para ser modificadas. Este módulo estará disponible sólo para los administradores.
RF_09	La plataforma contará con un módulo de eliminación de una pista. La eliminación no tendrá efectos secundarios a nivel de la base de datos, ya que la aplicación estará preparada para manejar excepciones cuando las carreras estén asociadas a una pista que no existe. Este módulo sólo estará disponible para administradores.
RF_10	La plataforma contará con un módulo de visualización de una sola pista. En esta vista se podrán ver todos los parámetros de la pista. Este módulo estará disponible para cualquier tipo de usuario.

Módulo de Registro de Temporadas de Re-Volt America	
Id	Descripción
RF_11	La plataforma contará con un módulo de creación de temporadas. Las temporadas deben contar con todos los parámetros obligatorios. Al momento de crear una temporada, sus 6 rankings asociados deberán crearse automáticamente. Sólo los administradores pueden crear temporadas.
RF_12	La plataforma contará con un módulo de visualización de temporadas. El listado contendrá todas las temporadas, las cuales vendrán ordenadas por fecha de inicio. Este módulo estará disponible para cualquier tipo de usuario.
RF_13	La plataforma contará con un módulo de edición de una temporada, las cuales deben contar con todos los parámetros obligatorios para ser modificados. Este módulo estará disponible sólo para los administradores.
RF_14	La plataforma contará con un módulo de eliminación de una temporada. Al eliminarse una temporada, se eliminarán también todos sus rankings asociados. Al mismo tiempo, todas las sesiones asociadas a dichos rankings también serán eliminadas. Las estadísticas globales de cada jugador también serán substraídas de sus perfiles. Este módulo sólo estará disponible para administradores.
RF_15	La plataforma contará con un módulo de visualización de una sola temporada. En esta vista se podrán ver los rankings asociados y la tabla de resultados de la temporada. Este módulo estará disponible sólo para cualquier tipo de usuario.

Módulo de Visualización de Rankings de Re-Volt America	
Id	Descripción
RF_16	La plataforma contará con un módulo de visualización de rankings, el cual se encontrará dentro de cada visualización de temporada. Este módulo estará disponible para cualquier tipo de usuario.
RF_17	La plataforma contará con un módulo de visualización de un solo ranking. En esta vista se podrán ver todas las sesiones asociadas al ranking, junto con su tabla de resultados acumulados. Este módulo estará disponible para cualquier tipo de usuario.

Módulo de Sesiones de Re-Volt America	
Id	Descripción
RF_18	La plataforma contará con un módulo de subida de sesiones en forma de archivo separado por comas. El archivo separado por comas deberá ser un Session Log generado por RVGL. Este módulo deberá permitir al usuario seleccionar un archivo separado por comas. Las sesiones deben contar con todos los parámetros obligatorios. Sólo los administradores pueden crear sesiones.
RF_19	La plataforma contará con un módulo de visualización de sesiones. El listado contendrá las últimas sesiones del ranking vigente de la temporada actual. Este módulo estará disponible para cualquier tipo de usuario.
RF_20	La plataforma contará con un módulo de eliminación de una sesión. La eliminación gatilla la sustracción de las estadísticas de los usuarios que participaron en ella. Este módulo sólo estará disponible para administradores.
RF_21	La plataforma contará con un módulo de visualización de una sola sesión. En esta vista se podrán ver todos los parámetros de la sesión, ordenados en el formato de RVA. Este módulo estará disponible para cualquier tipo de usuario.

5.5. Requerimientos No Funcionales del Software

La presente sección hablará de los requerimientos no funcionales de la aplicación desarrollada. Todos los requerimientos no funcionales se relacionarán con uno o más atributos. Si un atributo aplica a un requerimiento no funcional, eso quiere decir que el requerimiento contribuye a la calidad del software desarrollado a través de ese atributo. Todos los atributos listados están basados en la norma ISO 25010.

RNF_01		
Descripción	La plataforma contará con una API REST, la cual estará diseñada para ser consumida por aplicaciones de terceros. Todos los endpoints estarán disponibles sólo para lectura por parte de terceros.	
Atributo	Aplica	Especificación
Adecuación Funcional	X	La API contribuye a la corrección funcional, ya que facilita la obtención de datos precisos del sistema a terceros.
Eficiencia de Desempeño		
Compatibilidad	X	La API contribuye a la coexistencia con otras piezas de software independientes, ya que permite a dicho software consumir información del sistema en tiempo real.
Usabilidad		
Fiabilidad	X	La API contribuye a la madurez del software, ya que es gracias a ella que el sistema puede satisfacer las necesidades de los usuarios que consumen información del mismo.
Seguridad	X	Gracias al diseño de la API, sólo se exponen endpoints de lectura, por lo que esta contribuye a la confidencialidad e integridad de la información.
Mantenibilidad		
Portabilidad		

RNF_02		
Descripción	<p>La plataforma contará con una separación interna de roles a nivel de software. Estos roles serán: administrador, organizador, moderador y jugador, siendo el último denotado por no tener ninguno de los otros roles. El rol del primer administrador del sistema deberá ser asignado de manera interna. Una vez que existe un administrador, este podrá modificar los roles de todos los usuarios desde la misma página web.</p>	
Atributo	Aplica	Especificación
Adecuación Funcional	X	Contribuye a la completitud y pertinencia funcional, ya que, gracias a contar con una separación de privilegios de usuario, el sistema puede proveerles opciones específicas basado en la jerarquía de roles definida.
Eficiencia de Desempeño		
Compatibilidad		
Usabilidad		
Fiabilidad		
Seguridad	X	La presencia de roles en el sistema contribuye directamente a la autenticidad, ya que con ellos el sistema es capaz de decernir si el usuario debiera poder realizar una determinada acción o no basándose en su identidad dentro del software.
Mantenibilidad		
Portabilidad		

RNF_03		
Descripción	El código estará versionado a través de Git, utilizando GitHub como plataforma de almacenamiento en la nube para el control de versiones. El proyecto será de código abierto, y se contará con un sistema de integración continua para el software implementado directamente en GitHub. A través de este sistema se podrá ser desplegar el software de manera automática cuando se detecten cambios en una rama de Git determinada.	
Atributo	Aplica	Especificación
Adecuación Funcional		
Eficiencia de Desempeño		
Compatibilidad	X	Contribuye a la coexistencia del software con otros sistemas, ya que, gracias a estar versionado en Git y ser de código abierto, los mantenedores y desarrolladores de otras piezas de software podrán tener acceso a toda la información técnica que necesiten para poder integrar sus productos con la plataforma de RVA exitosamente.
Usabilidad	X	Existe una contribución al reconocimiento de la adecuación, ya que un usuario, al poder visualizar el historial de desarrollo completo en Git, podrá entender si el software, a nivel técnico, es adecuado para sus necesidades.
Fiabilidad	X	Gracias al sistema de despliegue que detalla, este requerimiento no funciona contribuye a la madurez, disponibilidad y tolerancia a fallos del sistema. Al contar con un sistema de despliegue y versionado del proyecto, los desarrolladores pueden trabajar en él y hacer llegar actualizaciones muchísimo más rápido al entorno de producción, y por ende a los usuarios finales del software.
Seguridad	X	Contribuye al no repudio, ya que, gracias a que el control de versiones lleva registro de todo lo que se modifica en la base de código, será posible rastrear malas decisiones o errores con facilidad.
Mantenibilidad	X	Este requerimiento contribuye a la modularidad, reusabilidad analizabilidad y a la capacidad del software para ser modificado y probado. Todo esto gracias a que los cambios realizados en el software son documentados a través del control de versiones.
Portabilidad	X	Contribuye a la adaptabilidad del software, y a la capacidad del mismo para ser instalado. Como todo el software es público y se encuentra en Git, analizarlo para adaptarlo resulta muy sencillo. Por otra parte, el sistema de despliegue facilita enormemente el instalar la aplicación en un entorno determinado.

5.6. Interfaces Internas de Salida

Id	Nombre	Detalle de Datos
IN_01	Car	name, speed, accel, weight, multiplier, folder_name, category, stock, season
IN_02	Track	name, short_name, difficulty, lenght, folder_name, stock, season
IN_03	Season	name, start_date, end_date, current, racer_result_entries
IN_04	Ranking	number, racer_result_entries, season
IN_05	Session	number, host, version, physics, protocol, pickups, date, teams, category, session_log_data, ranking, races, racer_result_entries
IN_06	Race	track_name, laps, racers_count, racer_entries
IN_07	RacerEntry	car_name, position, username, time, best_lap, finished, cheating, laps, racers_count
IN_08	RacerResultEntry	username, country, session_count, race_count, positions_sum, average_position, obtained_points, official_score, participation_multiplier, team
IN_09	User	username, encrypted_password, reset_password_token, reset_password_sent_at, remember_created_at, sign_in_count, current_sign_in_at, last_sign_in_at, current_sign_in_ip, last_sign_in_ip, confirmation_token, confirmation_token, confirmed_at, confirmation_sent_at, unconfirmed_email, failed_attempts, unlock_token, locked_at, admin, mod, organizar, locale, country, profile, stats

5.7. Interfaces Externas de Salida

Id	Nombre	Detalle de Datos	Medio de Salida
OUT_01	Car	name, speed, accel, weight, multiplier, folder_name, category, stock, season	Pantalla
OUT_02	Track	name, short_name, difficulty, lenght, folder_name, stock, season	Pantalla
OUT_03	Season	name, start_date, end_date, current, racer_result_entries	Pantalla
OUT_04	Ranking	number, racer_result_entries, season	Pantalla
OUT_05	Session	number, host, version, physics, protocol, pickups, date, teams, category, session_log_data, ranking, races, racer_result_entries	Pantalla, Archivo CSV.
OUT_06	User	username, admin, mod, organizer, locale, country, profile, stats	Pantalla

Capítulo 6

Análisis Funcional

6.1. Actores

En el software desarrollado, cada rol dentro del staff de RVA ha sido considerado como un actor diferente. Para el contexto del presente proyecto, el cargo de cada actor corresponde a su nombre, lo que quiere decir que si un actor lleva por nombre "Administrador", se entenderá que su cargo es "Administrador de la comunidad de RVA". Sólo los jugadores no cuentan con un cargo dentro de RVA.

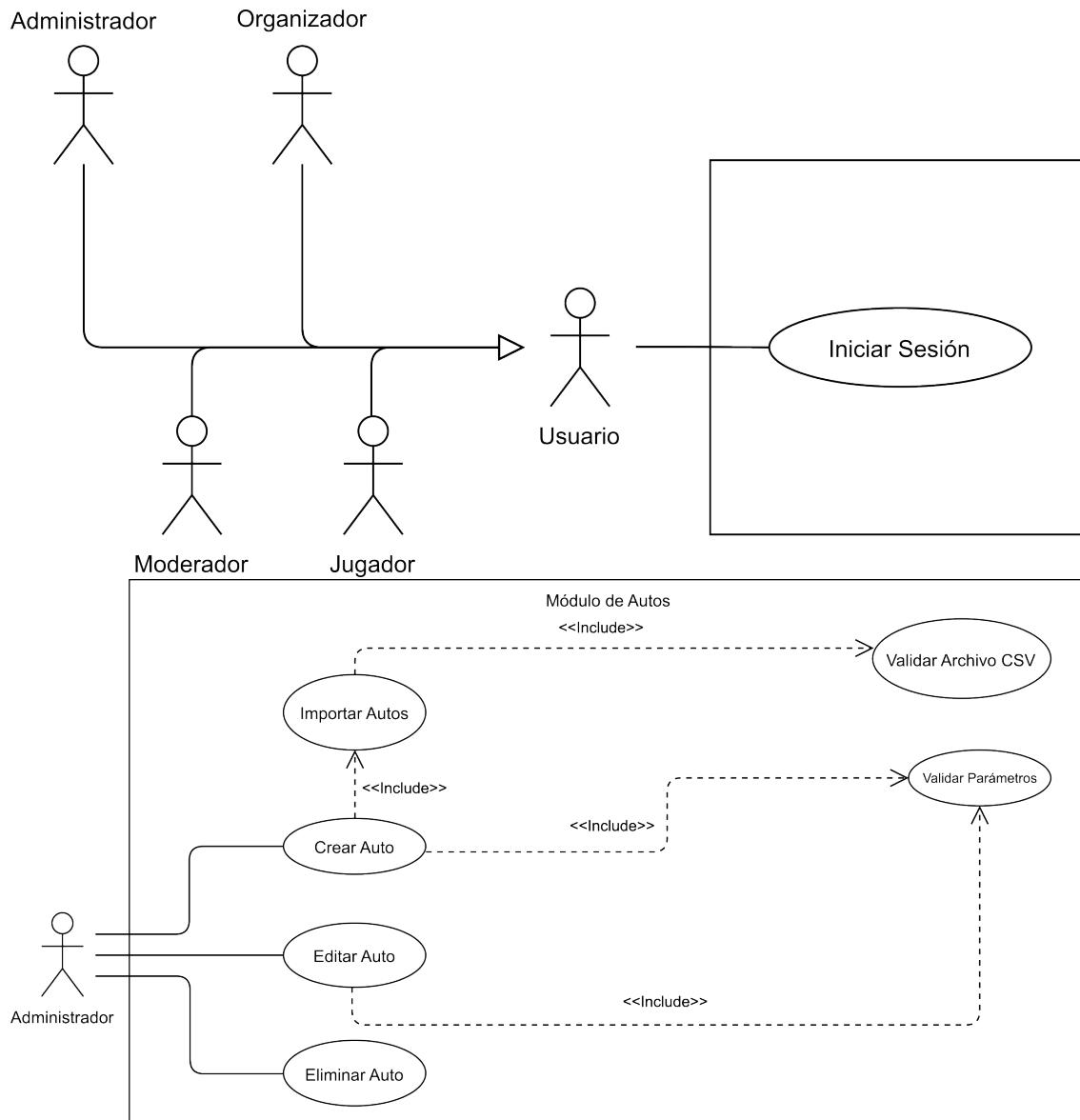
La especificación de todos los actores se puede encontrar a continuación en la tabla TABLENUM.

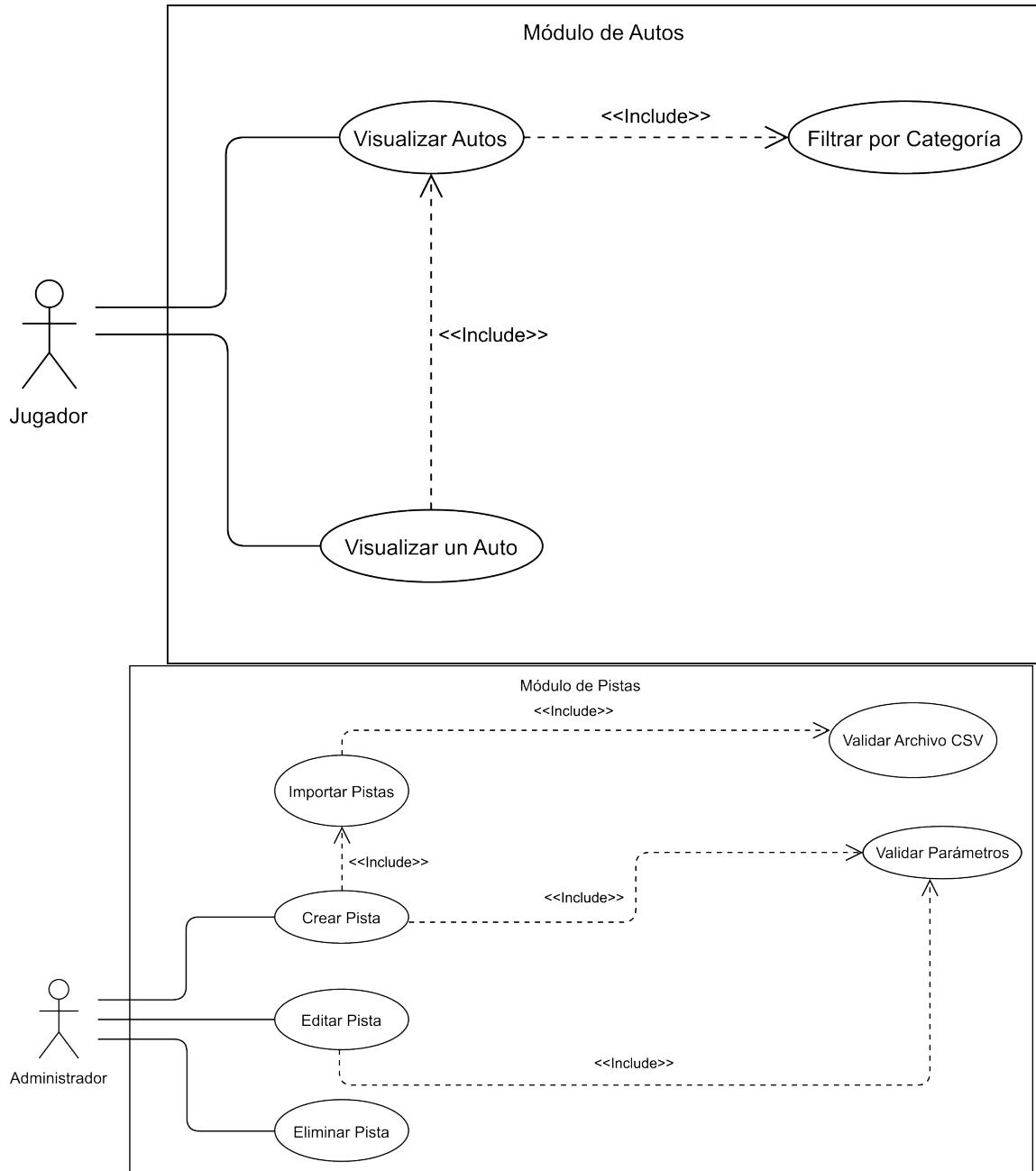
Actores			
Actor	Función	Conocimientos	Privilegio
Administrador	Cumple con todas las funciones dentro de la comunidad como mantener autos, pistas, sesiones y manejo general del software. Configura el software para el resto de los usuarios.	Requiere conocimiento sobre como funciona el sistema en su totalidad, comprendiendo también el contexto en el que funciona RVA como comunidad.	Máximo.
Organizador	Cumple con funciones específicamente relacionadas al manejo de sesiones multijugador y el procesamiento de los puntajes.	Requiere conocimiento específico de como subir archivos de sesión a la aplicación web, y un contexto general de como funciona RVA.	Intermedio (módulo de sesiones).
Moderador	Utiliza las funciones específicas de manejo de usuarios, tal como la edición de perfiles o aplicación de infracciones.	Requiere conocimientos que se limitan al manejo de usuarios dentro de la web.	Intermedio (manejo de usuarios).
Jugador	Utiliza la página sólo para visualizar la información que esta ofrece.	No requiere conocimientos técnicos más allá de iniciar sesión.	Ninguno.

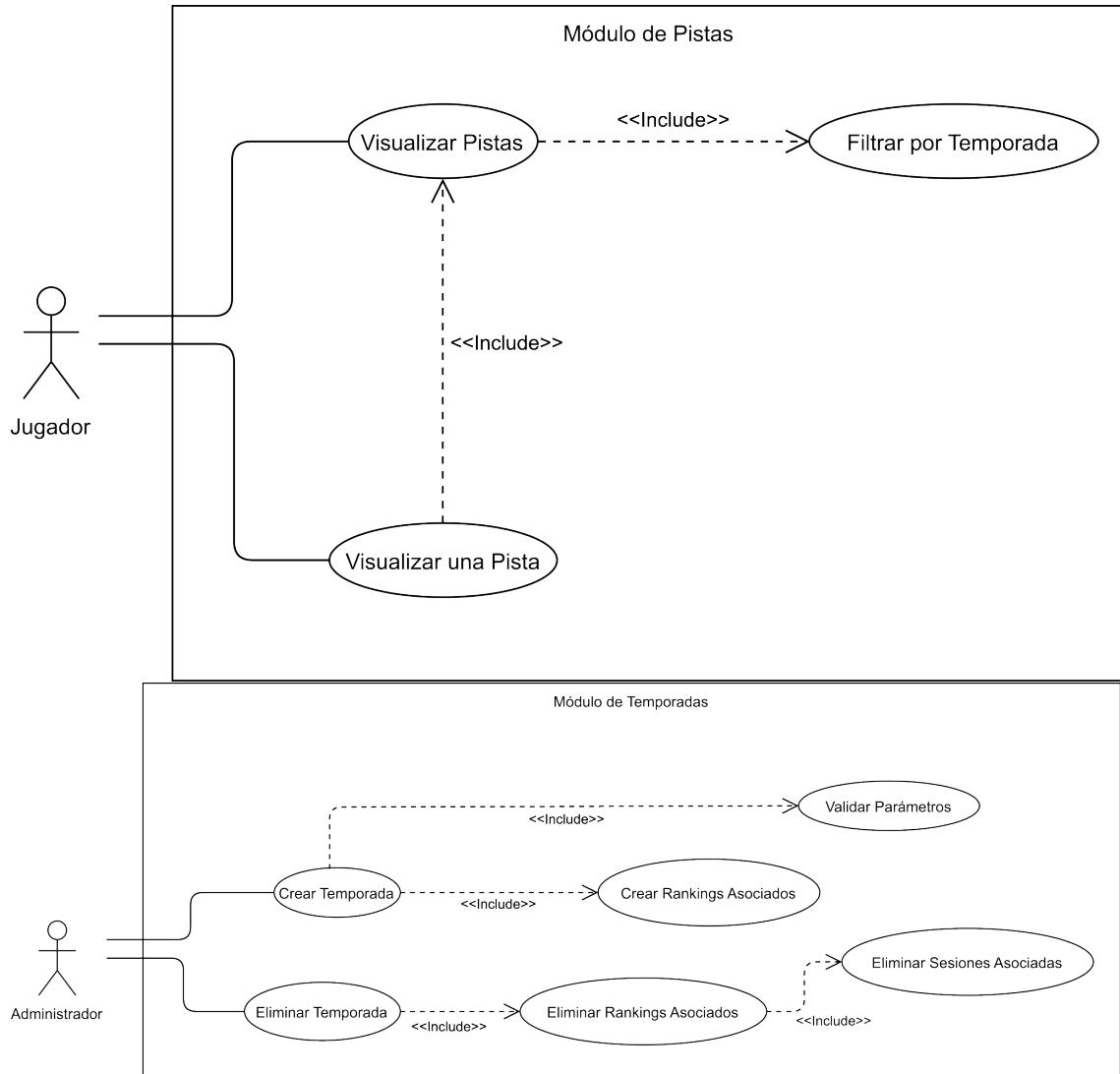
6.2. Casos de Uso

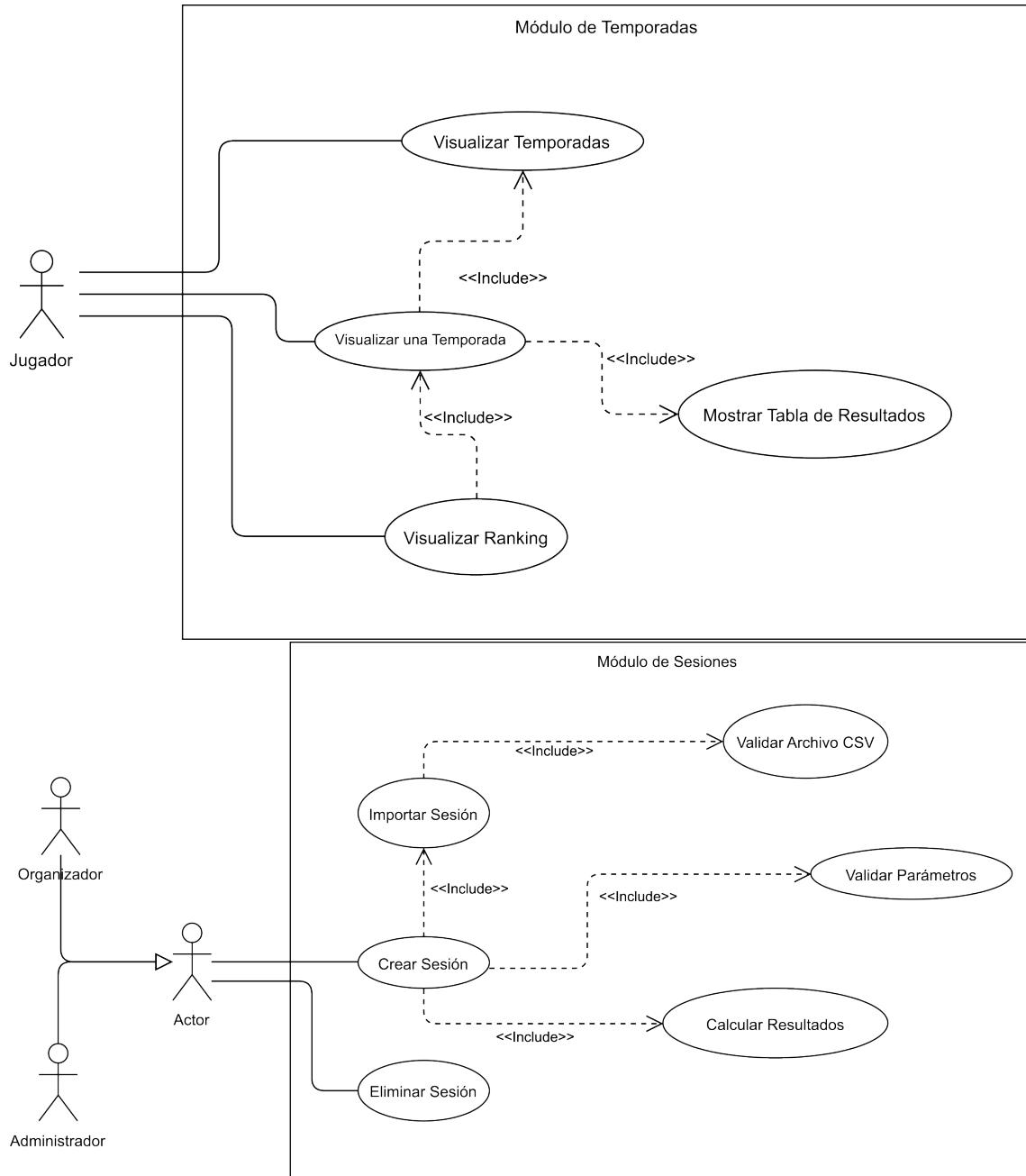
Esta sección contiene todos los diagramas de casos de uso relevantes para el proyecto. Todos los casos de uso cuentan con su respectiva especificación y detalle según corresponda.

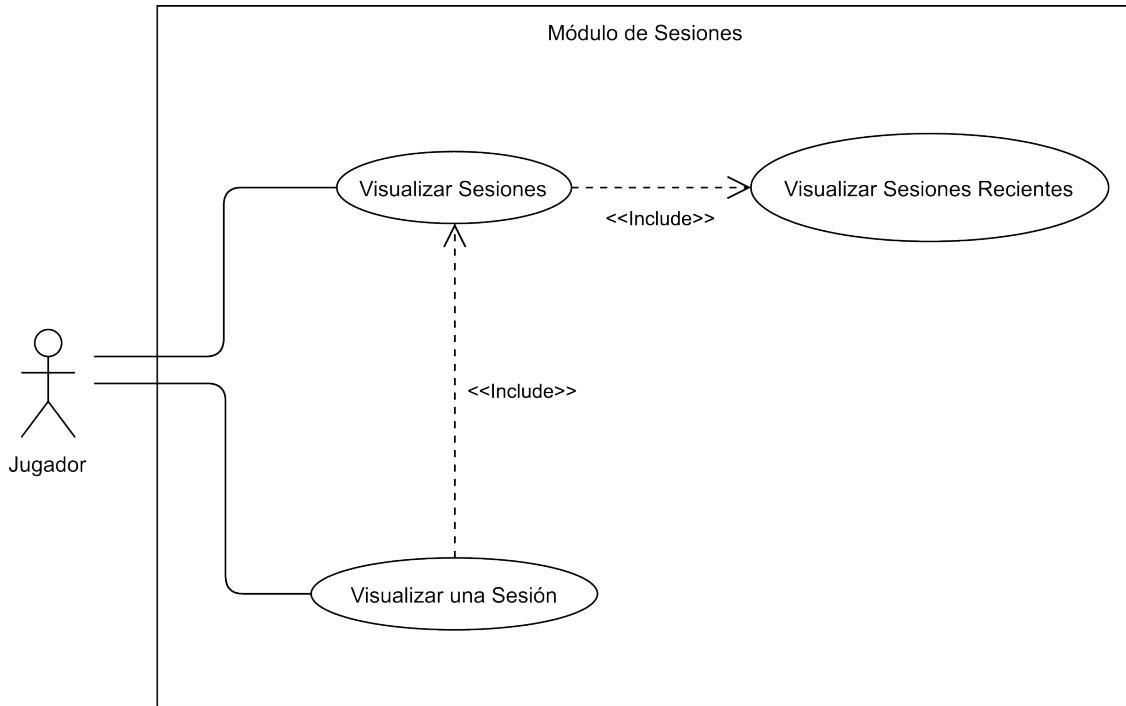
6.2.1. Diagramas de Casos de Uso











6.2.2. Especificación de los Casos de Uso

En esta sección se detallan todos los casos de uso diagramados en la subsección 6.2.1. De acuerdo con la jerarquía de actores presentada en la sección 6.1, se entenderá que, si un actor de tipo "Jugador" puede realizar una acción específica, entonces cualquier otro de los actores con cargo podrá realizarla también.

A continuación, la tabla TABLENUM contiene los nombres y actores asociados a cada uno de los diagramas de casos de uso. Los casos de uso cuyos actores y nombre han sido marcados en negrita serán detallados posteriormente.

Casos de Uso		
Id	Actor	Nombre
CU_01	Administrador, Moderador, Organizador y Jugador	Iniciar Sesión
CU_02	Administrador	Importar Autos
CU_03	Administrador	Crear Auto
CU_04	Administrador	Editar Auto
CU_05	Administrador	Eliminar Auto
CU_06	Administrador	Validar Archivo CSV
CU_07	Administrador	Validar Parámetros
CU_08	Jugador	Visualizar Autos
CU_09	Jugador	Visualizar un Auto
CU_10	Jugador	Filtrar por Categoría
CU_11	Administrador	Importar Pistas
CU_12	Administrador	Crear Pista
CU_13	Administrador	Editar Pista
CU_14	Administrador	Eliminar Pista
CU_15	Administrador	Validar Archivo CSV
CU_16	Administrador	Validar Parámetros
CU_17	Jugador	Visualizar Pistas
CU_18	Jugador	Visualizar una Pista
CU_19	Jugador	Filtrar por Temporada
CU_20	Administrador	Crear Temporada
CU_21	Administrador	Eliminar Temporada
CU_22	Administrador	Validar Parámetros
CU_23	Administrador	Crear Rankings Asociados
CU_24	Administrador	Eliminar Rankings Asociados
CU_25	Administrador	Eliminar Sesiones Asociadas
CU_26	Jugador	Visualizar Temporadas
CU_27	Jugador	Visualizar una Temporada
CU_28	Jugador	Visualizar Ranking
CU_29	Jugador	Mostrar Tabla de Resultados
CU_30	Administrador y Organizador	Importar Sesión
CU_31	Administrador y Organizador	Crear Sesión
CU_32	Administrador y Organizador	Eliminar Sesión
CU_33	Administrador y Organizador	Validar Archivo CSV
CU_34	Administrador y Organizador	Validar Parámetros
CU_35	Administrador y Organizador	Calcular Resultados
CU_36	Jugador	Visualizar Sesiones
CU_37	Jugador	Visualizar una Sesión
CU_38	Jugador	Visualizar Sesiones Recientes

6.2.3. Detalle de los Casos de Uso

A continuación, se presentan tablas de detalle para cada uno de los casos de uso listados en la subsección 6.2.2. Aquellos casos de uso que fueron marcados en negrita anteriormente también contarán con un detalle de su flujo de eventos básico.

CU_01_INICIAR_SESION (Usuario)	
Pre-Condiciones: El usuario debe estar en la página web. El usuario debe haberse registrado en la plataforma.	
Post-Condiciones: El usuario inicia sesión en la plataforma.	
Flujo de Eventos Básicos	
Usuarios: Administrador, Organizador, Moderador y Jugador	Sistema
	1. Renderiza la pantalla de inicio de sesión.
2. Ingresa su correo y contraseña, y luego pulsa el botón para iniciar sesión.	3. Valida la información ingresada por el usuario.
	4. Sesión iniciada. Redirecciona al usuario a la página principal.
Flujo de Eventos Alternativo	
Usuarios: Administrador, Organizador, Moderador y Jugador	Sistema
	3 (b). Si las credenciales son incorrectas, el sistema muestra un mensaje de error.
	4 (b). Vuelve al paso 2 del flujo básico.

CU_02_IMPORTAR_AUTOS (Administrador)	
Pre-Condiciones: El administrador debe haber iniciado sesión con sus credenciales.	
Post-Condiciones: El sistema crea los autos y los guarda en la base de datos.	
Flujo de Eventos Básicos	
Usuario: Administrador	Sistema
	1. Renderiza el formulario de importación de autos.
2. Selecciona el archivo CSV con la información de los autos y la temporada. Luego pulsa el botón para importar.	3. Valida el archivo CSV con la información de los autos. Valida si los datos son correctos. Verifica si ya existe un auto con el mismo nombre relacionado con la misma temporada.
	4. Autos creados. Redirecciona al administrador a la página de importación de autos.
Flujo de Eventos Alternativo	
Usuario: Administrador	Sistema
	3 (b). Si el archivo seleccionado no es válido, el sistema muestra un mensaje de error.
	4 (b). Vuelve al paso 2 del flujo básico.

CU_03_CREAR_AUTO (Administrador)
Pre-Condiciones: El administrador debe haber iniciado sesión con sus credenciales.
Descripción: El sistema guarda el modelo de auto en la base de datos con los datos ingresados por el administrador.

CU_04_EDITAR_AUTO (Administrador)
Pre-Condiciones: El administrador debe haber iniciado sesión con sus credenciales.
Descripción: El sistema actualiza el modelo de auto en la base de datos con los datos ingresados por el administrador.

CU_05_ELIMINAR_AUTO (Administrador)
Pre-Condiciones: El administrador debe haber iniciado sesión con sus credenciales.
Descripción: El sistema elimina el modelo de auto en la base de datos.

CU_06_VALIDAR_ARCHIVO_CSV
Pre-Condiciones: Debe ser activado por CU_02 .
Descripción: El sistema lee el archivo y verifica que éste esté en el formato adecuado, y que contenga datos válidos para crear modelos de autos.

CU_07_VALIDAR_PARAMETROS

Pre-Condiciones: Debe ser activado por **CU_03**.

Descripción: Verifica que los parámetros cumplan con las restricciones del modelo de auto.

CU_08_VISUALIZAR_AUTOS (Jugador)

Pre-Condiciones: No tiene.

Descripción: El sistema renderiza los autos por pantalla.

CU_09_VISUALIZAR_UN_AUTO (Jugador)

Pre-Condiciones: No tiene.

Post-Condiciones: El sistema renderiza un auto por pantalla.

Flujo de Eventos Básicos

Usuario:	Sistema
	1. El sistema renderiza un enlace a la página del auto, ya sea una imagen o texto clickeable.
2. El usuario clickea el enlace.	
	3. El sistema renderiza una visualización del auto seleccionado por el usuario.

CU_10_FILTRAR_POR_CATEGORIA

Pre-Condiciones: Debe ser activado por **CU_08**.

Descripción: El sistema verifica la categoría de los autos y la utiliza para separarlos en diferentes secciones de la plataforma.

CU_11_IMPORTAR_PISTAS (Administrador)	
Pre-Condiciones: El administrador debe haber iniciado sesión con sus credenciales.	
Post-Condiciones: El sistema crea las pistas y las guarda en la base de datos.	
Flujo de Eventos Básicos	
Usuario: Administrador	Sistema
	1. Renderiza el formulario de importación de pistas.
2. Selecciona el archivo CSV con la información de las pistas y la temporada. Luego pulsa el botón para importar.	3. Valida el archivo CSV con la información de las pistas. Valida si los datos son correctos. Verifica si ya existe una pista con el mismo nombre relacionada con la misma temporada.
	4. Pistas creadas. Redirecciona al administrador a la página de importación de pistas.
Flujo de Eventos Alternativo	
Usuario: Administrador	Sistema
	3 (b). Si el archivo seleccionado no es válido, el sistema muestra un mensaje de error.
	4 (b). Vuelve al paso 2 del flujo básico.

CU_12_CREAR_PISTA (Administrador)
Pre-Condiciones: El administrador debe haber iniciado sesión con sus credenciales.
Descripción: El sistema guarda el modelo de pista en la base de datos con los datos ingresados por el administrador.

CU_13_EDITAR_PISTA (Administrador)
Pre-Condiciones: El administrador debe haber iniciado sesión con sus credenciales.
Descripción: El sistema actualiza el modelo de pista en la base de datos con los datos ingresados por el administrador.

CU_14_ELIMINAR_PISTA (Administrador)
Pre-Condiciones: El administrador debe haber iniciado sesión con sus credenciales.
Descripción: El sistema elimina el modelo de pista en la base de datos.

CU_15_VALIDAR_ARCHIVO_CSV
Pre-Condiciones: Debe ser activado por CU_11 .
Descripción: El sistema lee el archivo y verifica que éste esté en el formato adecuado, y que contenga datos válidos para crear modelos de pistas.

CU_16_VALIDAR_PARAMETROS

Pre-Condiciones: Debe ser activado por **CU_12**.

Descripción: Verifica que los parámetros cumplan con las restricciones del modelo de pista.

CU_17_VISUALIZAR_PISTAS (Jugador)

Pre-Condiciones: No tiene.

Descripción: El sistema renderiza las pistas por pantalla.

CU_18_VISUALIZAR_UNA_PISTA (Jugador)

Pre-Condiciones: No tiene.

Post-Condiciones: El sistema renderiza una pista por pantalla.

Flujo de Eventos Básicos

Usuario:	Sistema
	1. El sistema renderiza un enlace a la página de la pista, ya sea una imagen o texto clickeable.
2. El usuario clickea el enlace.	
	3. El sistema renderiza una visualización de la pista seleccionada por el usuario.

CU_19_FILTRAR_POR_TEMPORADA

Pre-Condiciones: Debe ser activado por **CU_17**.

Descripción: El sistema verifica la temporada de las pistas y la utiliza para renderizar las pistas de la temporada actual.

CU_20_CREAR_TEMPORADA (Administrador)	
Pre-Condiciones: El administrador debe haber iniciado sesión con sus credenciales.	
Post-Condiciones: El sistema crea la temporada y sus rankings asociados.	
Flujo de Eventos Básicos	
Usuario: Administrador	Sistema
	1. Renderiza el formulario de creación de temporada.
2. Llena los campos del formulario de creación de temporada.	3. Valida los datos ingresados por el administrador. 4. El sistema crea la temporada. 5. El sistema crea 6 rankings y los relaciona con la temporada recién creada. 6. Temporada y rankings asociados creados. Redirecciona al administrador a la página de la temporada recién creada.
Flujo de Eventos Alternativo	
Usuario: Administrador	Sistema
	3 (b). Si los datos ingresados son inválidos, el sistema muestra un mensaje de error. 4 (b). Vuelve al paso 2 del flujo básico.

CU_21_ELIMINAR_TEMPORADA (Administrador)	
Pre-Condiciones: El administrador debe haber iniciado sesión con sus credenciales.	
Post-Condiciones: El sistema elimina la temporada, sus rankings asociados y todas las sesiones asociadas a dichos rankings.	
Flujo de Eventos Básicos	
Usuario: Administrador	Sistema
	1. Renderiza las opciones de administrador en la vista de temporada.
2. El administrador clica el botón de eliminar.	3. El sistema busca y elimina todas las sesiones de todos los rankings de la temporada eliminada. 4. El sistema elimina todos los rankings asociados a la temporada. 5. El sistema elimina la temporada. 6. Temporada eliminada. Redirecciona al administrador a la página de las temporadas.

CU_22_VALIDAR_PARAMETROS
Pre-Condiciones: Debe ser activado por CU_20 .
Descripción: Verifica que los parámetros cumplan con las restricciones del modelo de temporada.

CU_23_CREAR_RANKINGS_ASOCIADOS

Pre-Condiciones: Debe ser activado por **CU_20**.

Descripción: El sistema crea 6 modelos de ranking y los asocia a la temporada que se está creando en **CU_20**.

CU_24_ELIMINAR_RANKINGS_ASOCIADOS

Pre-Condiciones: Debe ser activado por **CU_21**.

Descripción: El sistema elimina los 6 modelos de ranking asociados a la temporada que se está eliminando en **CU_20**.

CU_25_ELIMINAR_SESIONES_ASOCIADAS

Pre-Condiciones: Debe ser activado por **CU_24**.

Descripción: El sistema elimina todos los modelos de sesión asociados con el ranking que se está eliminando en **CU_24**.

CU_26_VISUALIZAR_TEMPORADAS (Jugador)

Pre-Condiciones: No tiene.

Descripción: El sistema renderiza las temporadas por pantalla.

CU_27_VISUALIZAR_UNA_TEMPORADA (Jugador)

Pre-Condiciones: No tiene.

Descripción: El sistema renderiza una temporada por pantalla.

CU_28_VISUALIZAR_RANKING (Jugador)

Pre-Condiciones: No tiene.

Descripción: El sistema renderiza un ranking por pantalla.

CU_29_VISUALIZAR_TABLA_DE_RESULTADOS (Jugador)

Pre-Condiciones: Debe ser activado por **CU_27**.

Descripción: El sistema renderiza una tabla de resultados asociada a la temporada.

CU_30_IMPORTAR_SESION (Administrador y Organizador)	
Pre-Condiciones: El administrador u organizador debe haber iniciado sesión con sus credenciales.	
Post-Condiciones: El sistema crea la sesión y la guarda en la base de datos. El sistema también calcula y suma las estadísticas a la temporada y ranking de la sesión en cuestión. De la misma forma, suma los resultados calculados a las estadísticas individuales de los jugadores que participaron en ella.	
Flujo de Eventos Básicos	
Usuarios: Administrador y Organizador	Sistema
	1. Renderiza el formulario de importación de sesión.
2. Selecciona el archivo CSV (Session Log), la temporada y el ranking. Luego pulsa el botón para importar.	3. Valida el archivo Session Log. Valida si los datos son correctos.
	4. Sesión creada. Redirecciona al administrador u organizador a la página de visualización de la sesión importada.
Flujo de Eventos Alternativo	
Usuario: Administrador y Organizador	Sistema
	3 (b). Si el archivo seleccionado no es válido, el sistema muestra un mensaje de error.
	4 (b). Vuelve al paso 2 del flujo básico.

CU_31_CREAR_SESION (Administrador y Organizador)	
Pre-Condiciones: El administrador u organizador debe haber iniciado sesión con sus credenciales.	
Descripción: El sistema guarda el modelo de sesión en la base de datos. El sistema calcula y suma las estadísticas de cada jugador a sus perfiles, y también a los resultados acumulados de los modelos de ranking y temporada asociados a la sesión.	

CU_32_ELIMINAR_SESION (Administrador y Organizador)	
Pre-Condiciones: El administrador u organizador debe haber iniciado sesión con sus credenciales.	
Post-Condiciones: El sistema elimina la sesión. El sistema sustrae las estadísticas asociadas a la sesión de los perfiles de los jugadores que participaron en ella.	
Flujo de Eventos Básicos	
Usuario: Administrador y Organizador	Sistema
	1. Renderiza las opciones de administrador en la vista de sesión.
2. El administrador u organizador clica el botón de eliminar.	3. El sistema calcula los resultados de la sesión.
	4. El sistema busca los perfiles de cada jugador que participó en la sesión.
	5. El sistema sustrae los resultados calculados de cada perfil de jugador, respectivamente.
	6. Sesión eliminada. Redirecciona al administrador u organizador a la página principal.

CU_33_VALIDAR_ARCHIVO_CSV
Pre-Condiciones: Debe ser activado por CU_30 .
Descripción: El sistema lee el archivo y verifica que éste esté en el formato adecuado, y que contenga datos válidos para crear un modelo de sesión.

CU_34_VALIDAR_PARAMETROS
Pre-Condiciones: Debe ser activado por CU_31 .
Descripción: Verifica que los parámetros cumplan con las restricciones del modelo de sesión y sus sub-modelos.

CU_35_CALCULAR_RESULTADOS
Pre-Condiciones: Debe ser activado por CU_31 .
Descripción: Calcula los resultados de la sesión, transformándolos de datos crudos del Session Log a resultados en el formato de RVA.

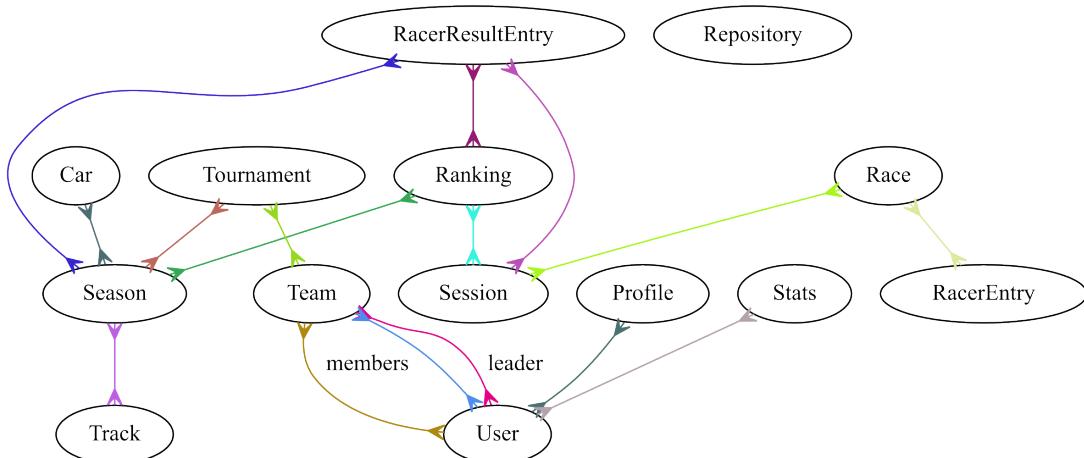
CU_36_VISUALIZAR_SESIONES (Jugador)
Pre-Condiciones: No tiene.
Descripción: El sistema renderiza las sesiones por pantalla.

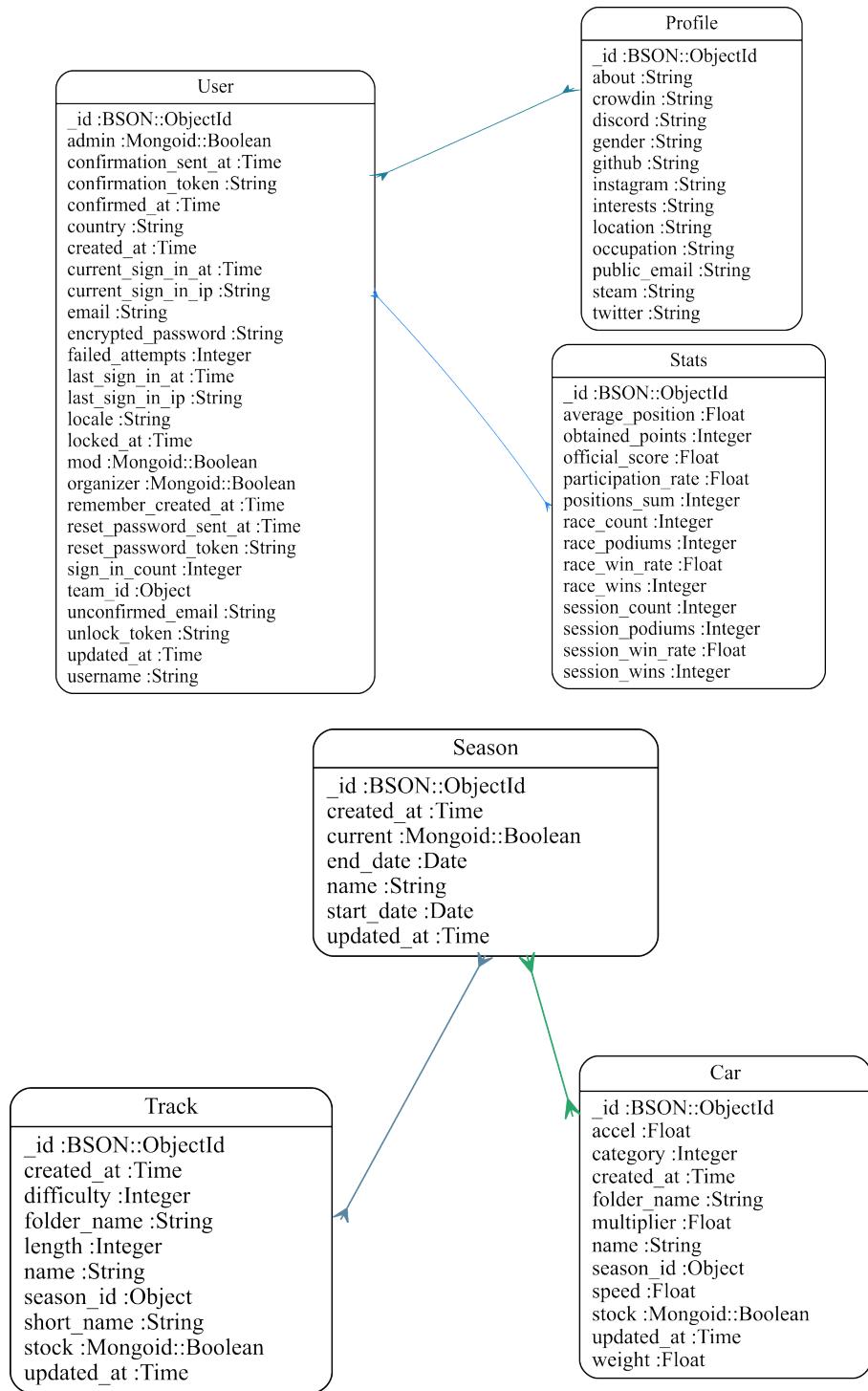
CU_37_VISUALIZAR_UNA_SESION (Jugador)	
Pre-Condiciones: No tiene.	
Post-Condiciones: El sistema renderiza una sesión por pantalla.	
Flujo de Eventos Básicos	
Usuario: Jugador	Sistema
	1. El sistema renderiza un enlace a la página de una sesión.
2. El usuario clickea el enlace.	
	3. El sistema calcula los resultados de la sesión y los transforma en resultados en el formato de RVA.
	3. El sistema renderiza la sesión por pantalla.

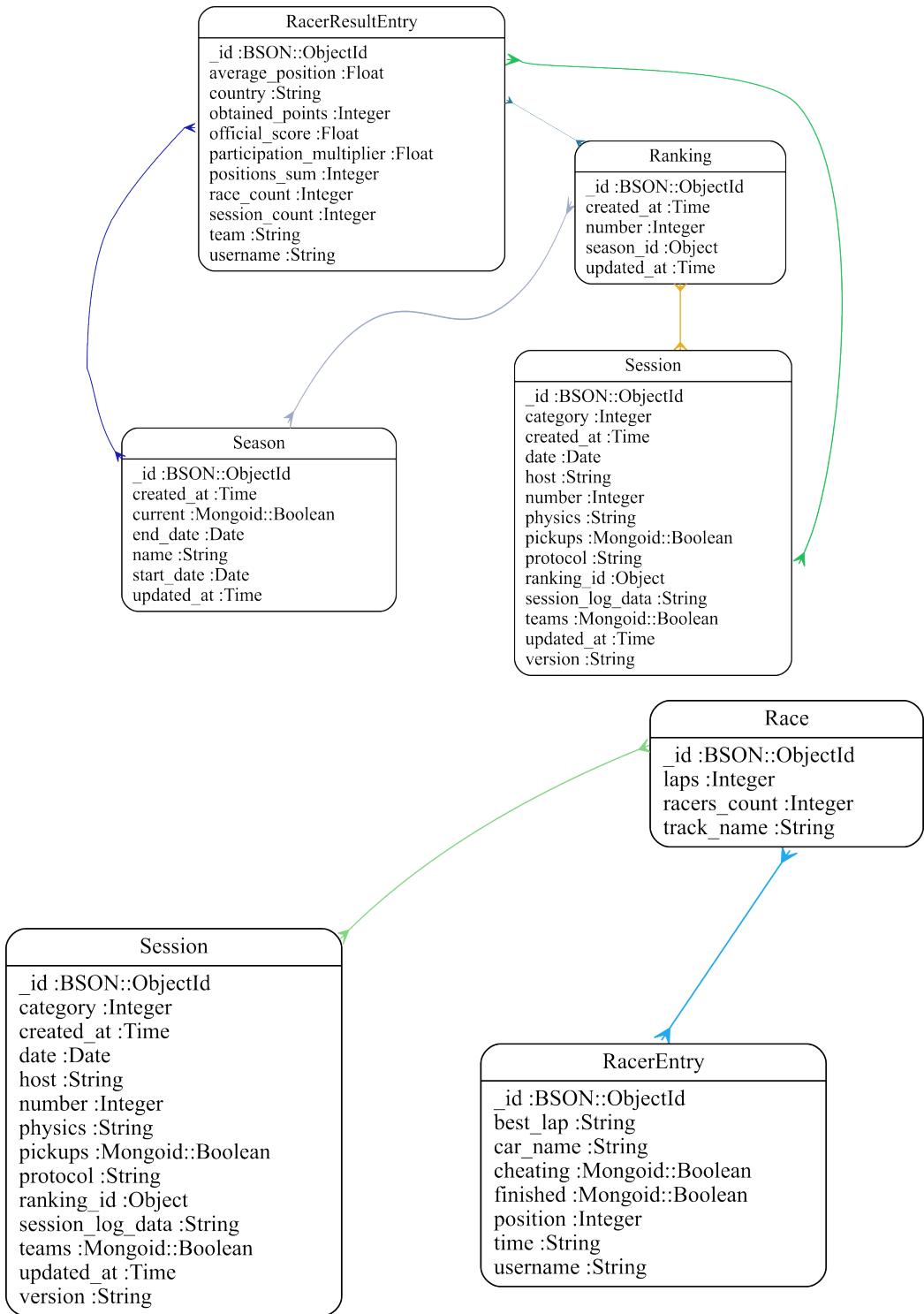
CU_38_VISUALIZAR_SESIONES_RECIENTES (Jugador)
Pre-Condiciones: Debe ser activado por CU_36.
Descripción: El sistema las últimas sesiones, buscando la temporada más reciente por fecha, y luego buscando la última sesión del ranking más alto posible (primero al sexto).

6.3. Modelo de Datos

Diagrama con Modelo de datos no relacional:







6.4. Esquema de la Base de Datos

Todas las bases de datos dentro de MongoDB están prefijadas utilizando el término “rv”. Por ejemplo, la base de datos que almacena las colecciones de autos se llama “rv_cars”, la de los usuarios “rv_users”, etc.

```

class Season
  include Mongoid::Document
  include Mongoid::Timestamps
  include Mongoid::MultiParameterAttributes

  store_in :database => 'rv_seasons'

  has_many :rankings
  has_many :cars
  has_many :tracks

  embeds_many :racer_result_entries

  accepts_nested_attributes_for :racer_result_entries

  field :name, :type => String
  field :start_date, :type => Date
  field :end_date, :type => Date
  field :current, :type => Boolean, :default => false

  validates_presence_of :name
  validates_presence_of :start_date
  validates_presence_of :current

  validates_uniqueness_of :name
end

```

Listing 1: Representación en código del modelo de Season de RVA.

EXPLICAR NOMENCLATURA DE COLECCIONES DE REDIS

A continuación, se describen los datos de la base de datos mediante archivos de definición de modelos de Ruby con mongoid:

```
class Ranking
  include Mongoid::Document
  include Mongoid::Timestamps

  store_in :database => 'rv_rankings'

  belongs_to :season
  has_many :sessions

  embeds_many :racer_result_entries

  accepts_nested_attributes_for :racer_result_entries

  field :number, :type => Integer

  validates_presence_of :number
end
```

Listing 2: Representación en código del modelo de Ranking de RVA.

```

class Session
  include Mongoid::Document
  include SessionLogUploader::Attachment(:session_log)
  include Mongoid::Timestamps

  store_in :database => 'rv_sessions'

  belongs_to :ranking
  embeds_many :races
  embeds_many :racer_result_entries

  accepts_nested_attributes_for :racer_result_entries

  field :number, :type => Integer
  field :host, :type => String
  field :version, :type => String
  field :physics, :type => String
  field :protocol, :type => String
  field :pickups, :type => Boolean
  field :date, :type => Date
  field :teams, :type => Boolean
  field :category, :type => Integer
  field :session_log_data, :type => String

  validates_presence_of :number
  validates_presence_of :host
  validates_presence_of :version
  validates_presence_of :physics
  validates_presence_of :protocol
  validates_presence_of :date
  validates_presence_of :session_log
  validates_presence_of :teams
  validates_presence_of :category
end

```

Listing 3: Representación en código del modelo de Session de RVA.

```

class User
  include Mongoid::Document
  include Mongoid::Timestamps

  store_in :database => 'rv_users'

  USERNAME_REGEX = /\A([A-Z0-9_]{1,16}|[0-9a-f]{24})\z/

  validates :email, :presence => true, :uniqueness => true
  validates :username, :presence => true,
    :uniqueness => true,
    :length => { :minimum => 2, :maximum => 16 }
  validates_format_of :username, :with => USERNAME_REGEX

  belongs_to :team, :optional => true
  has_one :team

  embeds_one :profile
  embeds_one :stats
  accepts_nested_attributes_for(
    :profile,
    :update_only => true,
    :allow_destroy => false
  )
  accepts_nested_attributes_for(:stats,
    :update_only => true,
    :allow_destroy => false
  )

  before_create :create_profile
  before_create :create_stats

  field :admin, :type => Boolean, :default => false
  field :mod, :type => Boolean, :default => false
  field :organizer, :type => Boolean, :default => false
  field :locale, :type => String, :default => 'en_us'
  field :country, :type => String
end

```

Listing 4: Representación en código del modelo de Session de RVA.

```

class Race
  include Mongoid::Document
  include Mongoid::Attributes::Dynamic

  embedded_in :session

  embeds_many :racer_entries

  field :track_name, :type => String
  field :laps, :type => Integer
  field :racers_count, :type => Integer

  validates_presence_of :track_name
  validates_presence_of :laps
  validates_presence_of :racers_count
end

```

Listing 5: Representación en código del modelo Profile.

```

class RacerEntry
  include Mongoid::Document
  include Mongoid::Attributes::Dynamic

  embedded_in :race

  field :car_name, :type => String
  field :position, :type => Integer
  field :username, :type => String
  field :time, :type => String
  field :best_lap, :type => String
  field :finished, :type => Mongoid::Boolean
  field :cheating, :type => Mongoid::Boolean

  validates_presence_of :car_name
  validates_presence_of :position
  validates_presence_of :username
  validates_presence_of :time
  validates_presence_of :best_lap
  validates_presence_of :finished
  validates_presence_of :cheating
end

```

Listing 6: Representación en código del modelo Profile.

```

class Profile
  include Mongoid::Document

  embedded_in :user

  field :about, :type => String
  field :gender, :type => String
  field :public_email, :type => String
  field :location, :type => String
  field :discord, :type => String
  field :github, :type => String
  field :instagram, :type => String
  field :crowdin, :type => String
  field :steam, :type => String
  field :twitter, :type => String
  field :occupation, :type => String
  field :interests, :type => String
end

```

Listing 7: Representación en código del modelo Profile.

```

class Stats
  include Mongoid::Document

  embedded_in :user

  field :race_wins, :type => Integer
  field :race_win_rate, :type => Float
  field :race_podiums, :type => Integer
  field :race_count, :type => Integer
  field :positions_sum, :type => Integer
  field :session_wins, :type => Integer
  field :session_win_rate, :type => Float
  field :session_podiums, :type => Integer
  field :session_count, :type => Integer
  field :average_position, :type => Float
  field :participation_rate, :type => Float
  field :official_score, :type => Float
  field :obtained_points, :type => Integer
end

```

Listing 8: Representación en código del modelo Stats.

```

class Car
  include Mongoid::Document
  include Mongoid::Timestamps

  store_in :database => 'rv_cars'

  belongs_to :season

  field :name, :type => String
  field :speed, :type => Float
  field :accel, :type => Float
  field :weight, :type => Float
  field :multiplier, :type => Float
  field :folder_name, :type => String
  field :category, :type => Integer
  field :stock, :type => Boolean, :default => false

  validates_presence_of :name
  validates_presence_of :speed
  validates_presence_of :accel
  validates_presence_of :weight
  validates_presence_of :multiplier
  validates_presence_of :folder_name
  validates_presence_of :category
  validates_presence_of :stock
end

```

Listing 9: Representación en código del modelo Car.

```
class Track
  include Mongoid::Document
  include Mongoid::Timestamps

  store_in :database => 'rv_tracks'

  belongs_to :season

  field :name, :type => String
  field :short_name, :type => String
  field :difficulty, :type => Integer
  field :length, :type => Integer
  field :folder_name, :type => String
  field :stock, :type => Boolean, :default => false

  validates_presence_of :name
  validates_presence_of :short_name
  validates_presence_of :difficulty
  validates_presence_of :length
  validates_presence_of :folder_name
  validates_presence_of :stock
end
```

Listing 10: Representación en código del modelo Track.

```
class RacerResultEntry
  include Mongoid::Document

  embedded_in :season
  embedded_in :ranking
  embedded_in :session

  field :username, :type => String
  field :country, :type => String
  field :session_count, :type => Integer
  field :race_count, :type => Integer
  field :positions_sum, :type => Integer
  field :average_position, :type => Float
  field :obtained_points, :type => Integer
  field :official_score, :type => Float
  field :participation_multiplier, :type => Float
  field :team, :type => String

  validates_presence_of :username
  validates_presence_of :country
end
```

Listing 11: Representación en código del modelo Track.

6.5. Diseño de Interfaz

The screenshot shows the homepage of the Re-Volt America website. At the top, there's a navigation bar with links for Play, Tracks, Cars, Seasons, League, Downloads, and BGM. Below the header is a large banner featuring a purple Re-Volt car on a track with the text "Re-Volt America" and "The Website for the American community of Re-Volt, 1999". A "Downloads »" button is visible. The main content area has three sections: "Community" (with a thumbnail of a race), "Competition" (with a thumbnail of a race), and "Reliability" (with a thumbnail of a red car). Each section has a brief description and a "Play »" button. Below these are two more sections with thumbnails: "Community" and "Competition", which are identical to the ones above. At the bottom, there's a "Recent Sessions" sidebar listing five recent races: "Random Races" (Hosted by MACACOSKY, October 19th, 2023), "SuperPro Races" (Hosted by BGM, October 18th, 2023), "Advanced Races" (Hosted by GFORCE, October 17th, 2023), "Pro Races" (Hosted by GFORCE, October 16th, 2023), and "Random Races" (Hosted by MACACOSKY, October 15th, 2023). The footer contains links for Policies (Rules, Terms of Service, Privacy Policy), Organization (Staff Listing, Development Logs, Report a Bug), Trademark (Assets, Sponsor), and Find Us (Discord, GitHub, Crowdin).

Re-Volt America
The Website for the American community of Re-Volt, 1999
Download the game and meet players from all over the Americas!

[Downloads »](#)

Community

We are an amazing Re-Volt community mainly made up of South American players, but you will also find people from all over the world playing online with us.

[Play »](#)

Competition

In Re-Volt America we race in a seasonal format. Throughout each season, you will be able to race online daily and score points for every race you participate in!

[Seasons »](#)

Reliability

Re-Volt America has been designed with sustainability and scalability in mind. Most of the features you see here you will not find in any other Re-Volt community.

[About Us »](#)

Community

We are an amazing Re-Volt community mainly made up of South American players, but you will also find people from all over the world playing online with us.

[Play »](#)

Competition

In Re-Volt America we race in a seasonal format. Throughout each season, you will be able to race online daily and score points for every race you participate in!

[Seasons »](#)

Reliability

Re-Volt America has been designed with sustainability and scalability in mind. Most of the features you see here you will not find in any other Re-Volt community.

[About Us »](#)

Recent Sessions

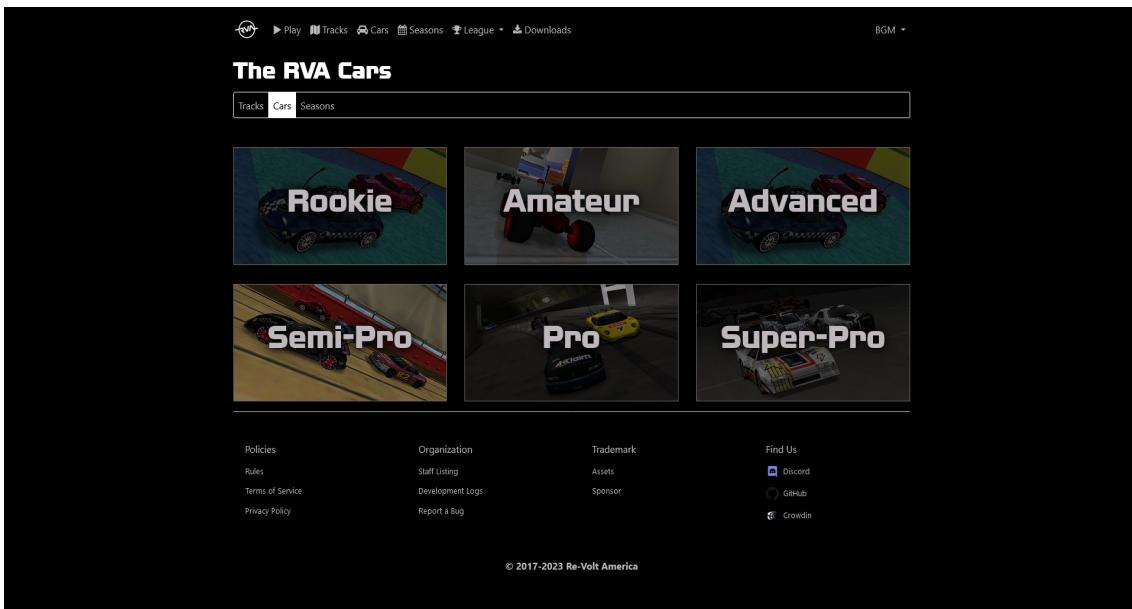
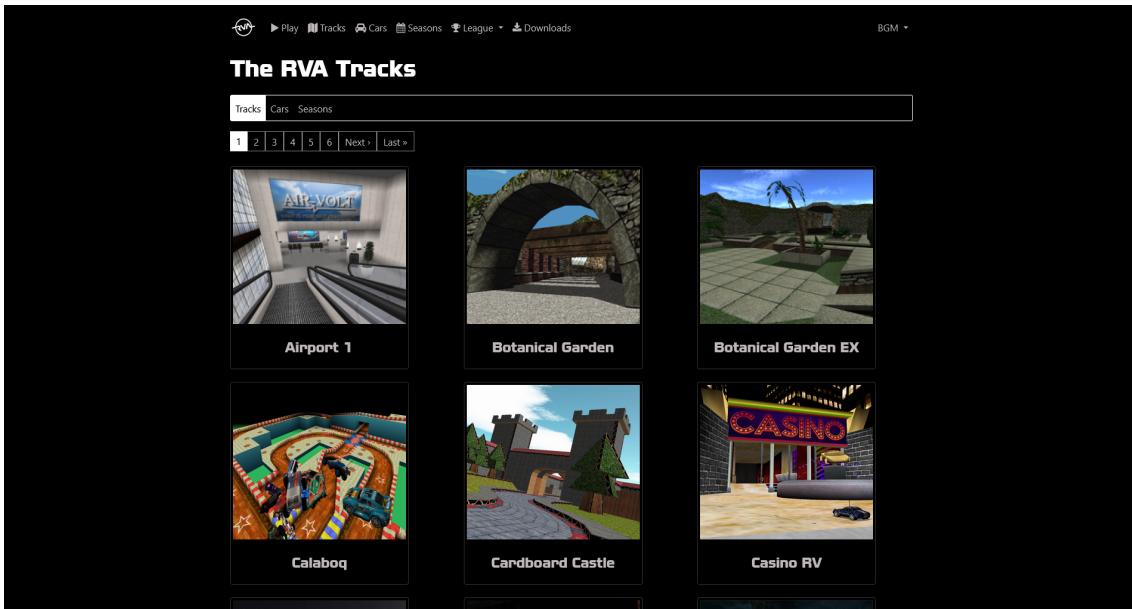
18 Random Races Hosted by MACACOSKY	October 19th, 2023
17 SuperPro Races Hosted by BGM	October 18th, 2023
16 Advanced Races Hosted by GFORCE	October 17th, 2023
15 Pro Races Hosted by GFORCE	October 16th, 2023
14 Random Races Hosted by MACACOSKY	October 15th, 2023

Policies
Rules
Terms of Service
Privacy Policy

Organization
Staff Listing
Development Logs
Report a Bug

Trademark
Assets
Sponsor

Find Us
Discord
GitHub
Crowdin



The RVA Cars - Rookie

Tracks Cars **Rookie** Seasons

Search... Classes ▾

RC Bandit 	Dr Grudge 	Sprinter XL 	E-Razzr
Speed 51.7 km/h	Speed 51.3 km/h	Speed 50.7 km/h	Speed 47.2 km/h
Acc 2.09 ms ²	Acc 2.54 ms ²	Acc 3.20 ms ²	Acc 3.10 ms ²
Weight 1.20 Kg	Weight 1.40 Kg	Weight 0.90 Kg	Weight 1.10 Kg
Multiplier 0.7	Multiplier 0.7	Multiplier 0.7	Multiplier 0.7

Hurricane 	Iso-77A 	Kanberra Kruiser 	Harvester
----------------------	--------------------	-----------------------------	----------------------

The RVA Seasons

Tracks Cars **Seasons**

What are RVA Seasons?

In RVA, we follow a seasonal format for online racing. Every Season consists of 6 Rankings, and each ranking consists of 28 Sessions, 18 of which are for Single races and the other 10 for Team races. Players score points for every race.

Current Season Learn More

All Seasons

Season 2023
2023-10-02 — Present

Policies Organization Trademark Find Us

Rules Staff Listing Assets Discord

Terms of Service Development Logs Sponsor GitHub

Privacy Policy Report a Bug Crowdin

© 2017-2023 Re-Volt America

Play Tracks Cars Seasons League Downloads

BGM

Tracks Cars Seasons Season 2023 Ranking

2023 Leaderboard

#	Country	Racer	Ranking 1	Ranking 2	Ranking 3	PA	MP	PO	CC	FJ
			Ranking 4	Ranking 5	Ranking 6					
1	ESP	YUN				3372	0.93	129.71	315	17
2	ESP	MACACOSKY	3.83			2697	0.91	79.78	255	14
3	CUB	ROG	5.24			2075	0.86	45.94	308	18
4	COL	WALUIGIMB	5.68			2235	0.86	41.17	242	14
5	COL	CEMMANUEL	4.82			1727	0.88	38.44	193	11
6	COL	ESTEBANMZ	4.41			1481	0.8	37.26	177	11
7	ECU	LOQ	5.96			1845	0.84	34.09	253	15
8	BRA	GFORCE	4.1			1093	0.96	30.16	77	4
9	BRA	FER	6.25			1440	0.88	25.81	247	14
10	ESP	BGM	2.03			500	0.98	24.69	39	2
11	PER	MIGHTY	5.67			1088	0.89	21.47	143	8
12	COL	TOMODOM	6.79			664	0.78	16.7	78	5

Play Tracks Cars Seasons Season 2023 Ranking 1

BGM

Sessions (18)

1 2 3 Next > Last <

18	Random Races	Hosted by MACACOSKY	October 19th, 2023
17	Super-Pro Races	Hosted by BGM	October 18th, 2023
16	Advanced Races	Hosted by GFORCE	October 17th, 2023
15	Pro Races	Hosted by GFORCE	October 16th, 2023
14	Random Races	Hosted by MACACOSKY	October 15th, 2023
13	Rookie Races	Hosted by MACACOSKY	October 14th, 2023
12	Super Pro Races	Hosted by MACACOSKY	October 13th, 2023
11	Semi-Pro Races	Hosted by YUN	October 12th, 2023

Ranking 1

#	Country	Racer	PP	PA	MP	PO	CC	FJ
1	ESP	YUN	3.35	3372	0.93	129.71	315	17

Play Tracks Cars Seasons League Downloads BGM

Tracks Cars Seasons > Season 2023 > Session 18

Random Races (Session 18)

Version: RVGL 23.0602a1
Connection: P2P [Server]
Mode: Arcade
Pickups: Enabled
Teams: No
[Download Session Log](#)

Admin Actions

Delete Session

#	Date	FP Racer	MM	SM2	SV2	DTI	SAN	T2	SW2	STR	FG2	MR	TW1	PLX	MY1	API	MD	DT2	SV1	TWM	BG	MY2	PP	PA	CC	MP	PO	
18	2023-10-19	1 MACACOSKY	1	6	2	1	1	3	7	1	2	1	3	2	5	2	1	2	1	2	2	2	2.35	220	20	1.0	9.36	
		2 YUN		2	4	2	2	1	1	4	1	4	1	1	1	2	6	3	1	4	1	4	1	2.25	214	19	0.95	9.04
		3 TOMDOM	3	3	1	4	3	2	3	6	3	6	6	3	1	5	2	3	2	4	3	3	3.3	178	20	1.0	5.39	
		4 WALUIGIMB								4	2	5	5	2	4	7	3	4	5	7		5	2.65	83	12	0.6	1.88	
		5 ESTEBANMZ	2	8	3	8	8	4	2	3	4	3	10	5	3								3.15	90	13	0.65	1.86	
		6 CEMMANUEL														4	3	3	1	4			0.75	47	5	0.25	1.57	
		7 ROG	4	4	5	6			6	7	6	2	4	6	8	8	6	6	5	7	5	7	5.1	85	18	0.9	1.5	
		8 ASURA	7	1	7	7	6	8	10	8	8	9	5	8	4	1	5						4.7	65	15	0.75	1.04	

Download Session Log

#	Date	FP Racer	MM	SM2	SV2	DTI	SAN	T2	SW2	STR	FG2	MR	TW1	PLX	MY1	API	MD	DT2	SV1	TWM	BG	MY2	PP	PA	CC	MP	PO	
18	2023-10-19	1 MACACOSKY	1	6	2	1	1	3	7	1	2	1	3	2	5	2	1	2	1	2	2	2	2.35	220	20	1.0	9.36	
		2 YUN		2	4	2	2	1	1	4	1	4	1	1	1	2	6	3	1	4	1	4	1	2.25	214	19	0.95	9.04
		3 TOMDOM	3	3	1	4	3	2	3	6	3	6	6	3	1	5	2	3	2	4	3	3	3.3	178	20	1.0	5.39	
		4 WALUIGIMB								4	2	5	5	2	4	7	3	4	5	7		5	2.65	83	12	0.6	1.88	
		5 ESTEBANMZ	2	8	3	8	8	4	2	3	4	3	10	5	3								3.15	90	13	0.65	1.86	
		6 CEMMANUEL														4	3	3	1	4			0.75	47	5	0.25	1.57	
		7 ROG	4	4	5	6			6	7	6	2	4	6	8	8	6	6	5	7	5	7	5.1	85	18	0.9	1.5	
		8 ASURA	7	1	7	7	6	8	10	8	8	9	5	8	4	1	5						4.7	65	15	0.75	1.04	
		9 NICK	5	7	8	5	4	6	8	9	9	7	8	7	9	4	8	8	8	6	7	6	6.95	64	20	1.0	0.92	
		10 GHOSTERCAT		7	5	5	5	10	10	7	9	6	7	7	7	6	5	6					5.1	54	15	0.75	0.79	
		11 FER	6	5	6	3	5	7	9	10	7	8	9	10									4.25	40	12	0.6	0.56	

FP: Final Position
PO: Official Score
PP: Average Position
CC: Race Count
PA: Accumulated Score
MP: Participation Multiplier

The screenshot shows the BGM profile page with the "General" tab selected. At the top, there's a navigation bar with icons for Play, Tracks, Cars, Seasons, League, and Downloads. Below the navigation is the BGM logo with a flag icon. A "Administrator" button is visible. On the left, a summary box displays the rank (10th), win rate (0.51), sessions (2), races (39), and wins (20). To the right, a "Session History" section lists two entries: "Super Pro Races" (17 sessions, October 18th, 2023) and "Amateur Races" (6 sessions, October 7th, 2023). Below these sections are tabs for "General" and "Stats". Under "General", there are links to GitHub (BGMP), Crowdin (BGM), Discord (bgm), Email (jose@bgm.cl), Occupation (Software Developer), Interests (RvGL), and About BGM (Nothing here yet...). At the bottom, there are links for Policies, Rules, Terms of Service, Organization (Staff Listing, Development Logs), Trademark (Assets, Sponsor), and Find Us (Discord, GitHub).

The screenshot shows the BGM profile page with the "Stats" tab selected. The layout is identical to the "General" tab, but the content is focused on statistics. The summary box shows the same metrics: rank (10th), win rate (0.51), sessions (2), races (39), and wins (20). The "Race Stats" section includes: 20 wins, 34 podiums, 0.51 win rate; 2 sessions played, 39 races played. The "Overall Stats" section includes: 2 session wins, 2 session podiums, 1.0 session win rate; 500 obtained points, 24.69 official score, 2.03 average position; 0.98 participation rate.

The screenshot shows a dark-themed web application interface. At the top, there are navigation links: Play, Tracks, Cars, Seasons, League, Downloads, and a BGM dropdown. Below the header is a section titled "Global Stats" with a subtitle "Sorted by Wins". A table displays 11 rows of race data, each with a rank, country flag, racer name, wins, podiums, win ratio, average position, race count, obtained points, and official score.

#	Country	Racer	Wins	Podiums	Win Ratio	Average Pos.	Race Count	Obtained Points	Official Score
1	YUN	YUN	90	197	0.29	3.35	315	3372	129.71
2	MACACOSKY	MACACOSKY	52	131	0.2	3.83	255	2697	79.78
3	ROG	ROG	24	94	0.08	5.24	308	2075	45.94
4	LOQ	LOQ	21	58	0.08	5.96	253	1845	34.09
5	ESTEBANMZ	ESTEBANMZ	21	76	0.12	4.41	177	1481	37.26
6	BGM	BGM	20	34	0.51	2.03	39	500	24.69
7	GFORCE	GFORCE	19	36	0.25	4.1	77	1093	30.16
8	WALUIGIMB	WALUIGIMB	17	61	0.07	5.68	242	2236	41.17
9	CEMMANUEL	CEMMANUEL	16	69	0.08	4.82	193	1727	38.44
10	MIGHTY	MIGHTY	11	45	0.08	5.67	143	1088	21.47
11	BURDANG	BURDANG	10	30	0.09	5.83	104	634	13.96

The screenshot shows a dark-themed web application interface. At the top, there are navigation links: Play, Tracks, Cars, Seasons, League, Downloads, and a BGM dropdown. Below the header is a section titled "Upload Session" with several input fields: "Browse..." (with "No file selected."), "Select Season" (dropdown), "Select Ranking" (dropdown), "Select Category" (dropdown), and checkboxes for "Teams" and "Import". At the bottom, there are links for Policies (Rules, Terms of Service, Privacy Policy), Organization (Staff Listing, Development Logs, Report a Bug), Trademark (Assets, Sponsor), and Find Us (Discord, GitHub, Crowdin). A copyright notice at the very bottom reads "© 2017-2023 Re-Volt America".

6.5.1. Fuentes



6.5.2. Paleta de Colores

6.6. Diseño de Arquitectura

El proyecto, en su estado actual, hace uso de un servidor propio, el cual contiene los servicios web, bases de datos y caché. Todo en una sola máquina. Independientemente de donde se termine alojanda, la aplicación web estará disponible en "<https://rva.lat/>".

Además de esto, la planificación contempla dos servicios externos, que actualmente son proveídos por GitHub pages, los cuales sirven como repositorios de almacenamiento de datos masivos. Dichos repositorios se encargan actualmente de servir información y assets como las imágenes de las pistas y autos que la web ofrece a los usuarios:

- <https://tracks.rva.lat/>: Repositorio de pistas de RVA.
- <https://cars.rva.lat/>: Repositorio de autos de RVA.

A continuación, se muestra un diagrama que ilustra todo el proceso de interacción entre servicios y usuarios:

Tal como se puede apreciar en la ilustración anterior, la arquitectura que da soporte a la aplicación web de RVA se concentra en un servidor, con dos almacenes de datos. Podemos ver que los usuarios juegan la sesión, el host de la sesión sube el Session Log a la web, y los usuarios pueden visitar la misma web para revisar los resultados

6.7. Estructura del código

El proyecto, al ser una aplicación hecha en el framework de Ruby on Rails, sigue patrón de MVC (Model View Controller), o modelo, vista, controlador. El árbol de

directorio se ve de la siguiente manera:

6.7.1. Backend

Directorio	Detalle
controllers	Contiene todos los controladores de Rails, los cuales manejan todas las peticiones web del usuario.
helpers	Contiene todas las clases utilitarias, las cuales pueden ser utilizadas para asistir a las clases de modelos, vistas y controladores.
javascript	Todo el código de JavaScript utilizado por la aplicación a nivel web se almacena en este directorio.
models	Contiene todas las clases que modelan y envuelven los datos almacenados en la base de datos de la aplicación.
services	Contiene todas las clases de servicio de la aplicación. Las clases de servicio son aquellos procesos complejos compuestos de mucho código, los cuales se abstractan en servicios para no polucionar los controladores que los requieren.
uploaders	Contiene todas las clases relacionadas con la subida de archivos.
config	Contiene toda la configuración de la aplicación, como las rutas, inicializadores de librerías, configuración de despliegue, declaración de los entornos de Rails (development, production), configuración de la base de datos y archivos de idioma para la localización del proyecto.

6.7.2. Frontend

Directorio	Detalle
views	Contiene todas las vistas de la aplicación. Todo lo que el usuario ve en la aplicación web tiene su archivo de vista que lo representa.
assets	Todas las imágenes, hojas de estilo y demás artefactos requeridos por la aplicación para su correcto funcionamiento y diseño.
public	Todos los archivos estáticos servidos por la página pueden encontrarse aquí. Cosas como las páginas de error (404, 422, 500, etc.), estilos compilados, entre otros.

Capítulo 7

Plan de Capacitación, Implantación y Puesta en Marcha

7.0.1. Estado del Proyecto

Actualmente, el proyecto se encuentra finalizado.

EXPLICAR TODO LO QUE SE TERMINÓ

Además de lo anterior, se ha programado e implementado completamente la lógica operativa interna de Re-Volt America dentro del sistema, por lo que este puede recibir archivos Session Log, procesarlos y almacenarlos correctamente, mostrando al usuario una vista interpretada de los resultados de la sesión.

7.0.2. Proyecciones del Proyecto

Capítulo 8

Conclusión del Proyecto

Para finalizar este informe, a continuación, se plantean una serie de conclusiones a las que se llegó luego de llevar a cabo el desarrollo e implementación del software:

- En relación con elaborar una propuesta que consiguiera atender las necesidades y problemas de los usuarios de Re-Volt America, con respecto al almacenamiento y visualización de resultados de partidas online, podemos decir que se logró cumplir dicho objetivo gracias al software desarrollado.
- De acuerdo con el objetivo que hablaba de diseñar una solución de software de procesamiento de datos de sesiones multijugador oficiales de las sesiones de carreras en línea, además de estadísticas personales, podemos decir que es un objetivo que fue cumplido completamente ya que, tanto el procesamiento de datos como las estadísticas por usuario fueron exitosamente implementadas.
- En relación con el objetivo de implementar una aplicación web que permitiera a los organizadores de sesiones de carreras en línea subir y publica los resultados de dichas carreras podemos decir que, en conclusión, dicho objetivo fue cumplido con éxito, ya que el software, incluso en su estado actual, ya puede realizar la importación de Session Logs, el cálculo y procesamiento de resultados en el formato de RVA, y finalmente mostrar dichos resultados por pantalla a quien los solicite desde la web.
- El proyecto ha permitido demostrar, incluso en su estado actual, las competencias que se esperan de un ingeniero de ejecución en computación e informática, ya que he aplicado la identificación de necesidades, análisis y el diseño de soluciones informáticas para Re-Volt America, logrando desarrollar una solución que le permite a la comunidad tener un mejor manejo de sus proceso internos, registros de datos más fiables y una experiencia de usuario muchísimo mejor que con la que contaban al trabajar con su sistema original.
- Bajo mi percepción, el proyecto fue realmente enriquecedor desde el punto de vista del desarrollo de software puesto que, dentro del área que fue comprendido pude aplicar diversas tecnologías, técnicas de diseño de software, despliegue de aplicaciones y aplicación de estándares de calidad, todo en un mismo contexto que

cierra de manera redonda el ciclo de desarrollo que se espera pueda ser alcanzado por un ingeniero de software.

8.0.1. Proyecciones del Proyecto

Dentro de las proyecciones del proyecto etc etc

Capítulo 9

Anexos

9.1. Anexos de Pruebas de Aceptación

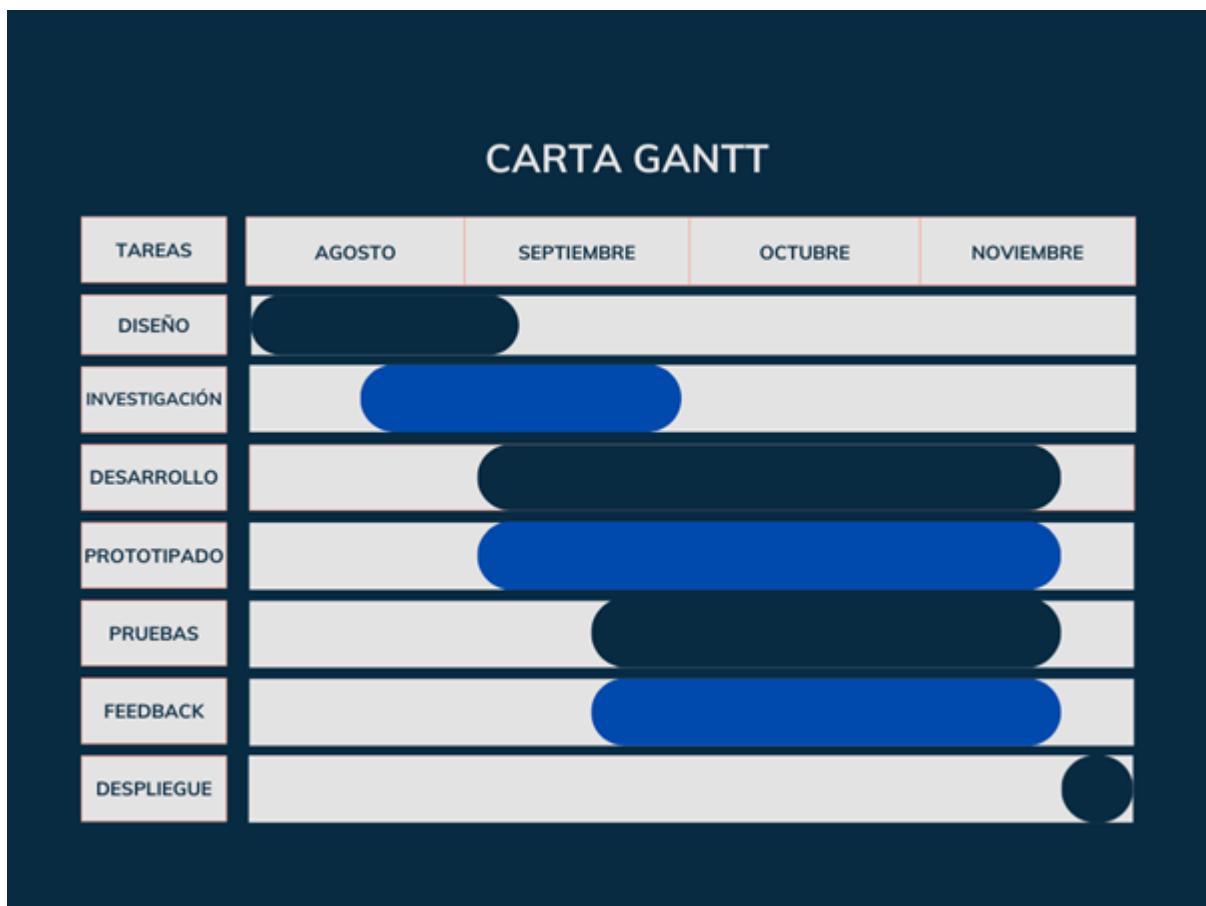
9.2. Anexos de Recopilación de Información

REUNIONES Y RECOLECCION DE INFORMACION DE USO ETC

9.3. Anexo Diccionario de Datos

9.4. Anexo Aspectos de Gestión de Proyectos

9.4.1. Anexo Carta Gantt



9.4.2. Riesgos de Alto Nivel, Impacto y Estrategia

9.4.3. Anexo Estimación Casos de Uso

9.4.4. Anexo Resumen de Esfuerzo

Actividad	Número de Horas
Preparación del proyecto	20
Desarrollo del módulo de autos	50
Desarrollo del módulo de pistas	50
Desarrollo del módulo de sesiones	80
Desarrollo del módulo de temporadas	80
Corrección de errores de código	40
Despliegue de la aplicación	27
Control de versiones	37
Total	384

9.5. Anexos Retrospectiva del Proyecto

9.5.1. Anexo Iteraciones en el Desarrollo

Funcionalidad	Fecha	Retroalimentación
Módulo de Autos (1)	12/11/2023	Añadir visualización para todos los autos y un sólo auto.
Módulo de Autos (2)	12/11/2023	Agregar los autos tanto a la navegación como a la sub-navegación de la aplicación
Módulo de Autos (3)	21/11/2023	Corregir problemas internos del modelo de autos
Módulo de Pistas (1)	11/11/2023	Corregir problemas de validación en el modelo de Pista
Módulo de Pistas (2)	12/11/2023	Agregaron las pistas tanto a la navegación como a la sub-navegación de la aplicación
Módulo de Pistas (3)	28/11/2023	Mejorar la relación a modelos de Pista desde la visualización del modelo de Sesión.
Módulo de Temporadas (1)	19/11/2023	Generar modelos de Ranking asociados a la temporada automáticamente después de crearla
Módulo de Temporadas (2)	30/11/2023	Eliminar entradas de resultado de jugador asociadas a la temporada si estas tienen todos sus parámetros en cero
Módulo de Sesiones (1)	19/11/2023	Determinar el número de cada sesión automáticamente una vez creada
Módulo de Sesiones (2)	28/11/2023	Añadir botón de descarga para el Session Log asociado a cada sesión
Módulo de Sesiones (3)	29/11/2023	Añadir acciones de administrador a la vista de sesión, tales como eliminar