



Universidad de Valladolid

ESCUELA DE INGENIERÍA INFORMÁTICA (SEGOVIA)

Grado en Ingeniería Informática de Servicios y Aplicaciones

**Gestor de stints para carreras en simuladores
(MyRacingManager)**

Alumno: Sergio Hernando Mateo

Tutor: José Vicente Álvarez Bravo

Gestor de stints para carreras en simuladores. Sergio Hernando Mateo.

“The winner ain’t the one with the fastest car, it’s the one who refuses to lose”

-Dale Earnhardt

“You miss 100% of the shots you don’t take”

-Wayne Gretzky

Agradecimientos

A mi tutor José Vicente por el tiempo invertido en apoyarme en la realización de este proyecto.

A mi familia, por mostrarme día a día que si te esfuerzas, puedes conseguir todo lo que te propongas.

A mis amigos y compañeros de carrera, por todos los grandes momentos que hemos vivido y viviremos.

Y en especial a Carlos, compañero de fatigas durante toda la carrera y sin el cual, probablemente, no estaría escribiendo estas palabras.

A todos, muchas gracias.

Resumen

El objetivo de este proyecto es el desarrollo de una aplicación informática que permita la gestión de los stints realizados en carreras de simulación online. Esta aplicación permitirá la gestión que van a llevar a cabo tanto los equipos formados por los diferentes pilotos, como los stints que realizará cada piloto en el transcurso de una carrera virtual.

Palabras clave: stint/s, gestión, piloto, equipo, carrera virtual.

Abstract

This project's objective is developing a desktop application that allows the user to manage the stints done by a driver on an online simulation race. This application will allow team management, as well as the stint management for online races.

Key words: stint/s, management, driver, team, online race.

Índice

Agradecimientos	4
Resumen	6
Abstract	6
Índice	8
Índice de tablas	12
Índice de figuras	15
Capítulo 1: Introducción	17
Motivación	20
Objetivos y alcance	20
Árbol de características	22
Estado del arte	23
JRT (Joel Real Timing) (iRacing)	23
Hoja Excel (iRacing)	24
McLaren ATLAS (McLaren)	24
Pi ToolBox (Cosworth)	25
Entorno tecnológico	26
Visual Studio	26
GitHub	26
NuGet	26
WPF (Windows Presentation Foundation)	27
SQLite	27
SQLite Browser	27
C# (C Sharp)	27
XAML (eXtensible Application Markup Language)	28
Extended WPF Toolkit (Librería)	28
Selenium WebDriver	28
Usuarios de la aplicación	29
Piloto de iRacing	29
Piloto de simulación	29
Estrategas	29
Jefes de equipo	29
Fan de carreras en general	29
Organización del documento	30
Contenido del CD-ROM	31

Capítulo 2: Planificación, presupuesto y estimación	33
Metodología	34
Estimación	35
Planificación	40
Presupuesto	42
Presupuesto en personal	42
Presupuesto en hardware	42
Presupuesto en software	43
Capítulo 3: Análisis	45
Requisitos de usuario	46
Requisitos funcionales	58
Listado de requisitos funcionales	58
Requisitos no funcionales	60
Disponibilidad	60
Accesibilidad	60
Usabilidad	60
Seguridad	61
Restricciones	61
Atributos de calidad	61
Requisitos de información	61
Listado de requisitos de información	61
Casos de uso	64
Listado de casos de uso	64
Diagramas de casos de uso	87
Capítulo 4: Diseño	91
Arquitectura lógica	92
Arquitectura física	93
Diagramas de clases	94
Diagramas de secuencia	96
Diagrama entidad-relación de la base de datos	99
Diagrama relacional base de datos	100
Diccionario de datos	101
Diseño interfaz de usuario	105
Capítulo 5: Implementación	117
Análisis página web iRacing (Scraping de datos)	118
Generación de horarios	122
Elección de color de fuente	124
Capítulo 6: Pruebas	127

Caja negra	128
Capítulo 7: Manuales de usuario (instalación)	139
Capítulo 8: Manuales de usuario	142
Capítulo 9: Conclusiones y futuras mejoras	152
Conclusiones	153
Mejoras a corto plazo	153
Mejoras a largo plazo	153
Bibliografía y Webgrafía	155
Bibliografía	156
Webgrafía	156
Anexos	159
Anexo I: Glosario de términos de carreras de coches	160

Índice de tablas

Tabla 1: Tabla de objetivos	21
Tabla 2: Tabla de suposiciones	22
Tabla 3: Tabla de dependencias	22
Tabla 4: Tabla comparativa entre herramientas de seguimiento de carreras	26
Tabla 5: Tabla comparativa de uso entre los diferentes usuarios de la aplicación	31
Tabla 6: Tabla de salidas/entradas/consultas/fi/fe	37
Tabla 7: Tabla de complejidad entradas y consultas	38
Tabla 8: Tabla de complejidad salidas	38
Tabla 9: Tabla de complejidad ficheros externos e internos	38
Tabla 10: Tabla de calculo puntos de función no ajustados	39
Tabla 11: Tabla de complejidades	40
Tabla 12: Horario de trabajo	41
Tabla 13: Cálculo presupuesto en personal	43
Tabla 14: Cálculo presupuesto en hardware	43
Tabla 15: Cálculo presupuesto en software	44
Tabla 16: Lista de requisitos de usuario	47
Tabla 17: US-01: Crear piloto con ID	48
Tabla 18: US-02: Crear piloto sin ID	49
Tabla 19: US-03: Eliminar piloto	49
Tabla 20: US-04: Editar piloto	50
Tabla 21: US-05: Crear evento	50
Tabla 22: US-06: Editar evento	51
Tabla 23: US-07: Eliminar evento	51
Tabla 24: US-08: Crear carrera	52
Tabla 25: US-09: Eliminar carrera	52
Tabla 26: US-10: Crear equipo	53
Tabla 27: US-11: Editar equipo	53
Tabla 28: US-12: Añadir piloto a equipo	54
Tabla 29: US-13: Eliminar piloto de equipo	54
Tabla 30: US-14: Eliminar equipo	55
Tabla 31: US-15: Insertar credenciales de iRacing	55
Tabla 32: US-16: Eliminar credenciales de iRacing	56
Tabla 33: US-17: Editar disponibilidad del piloto	56
Tabla 34: US-18: Crear horario manualmente	57
Tabla 35: US-19: Generar horario automáticamente	57
Tabla 36: US-20: Cambiar idioma del programa	58
Tabla 37: US-21: Consultar información del programa	58
Tabla 38: Tabla de requisitos funcionales	61
Tabla 39: Lista de requisitos de información	62

Tabla 40: RI-01: Email	63
Tabla 41: RI-02: Datos evento	63
Tabla 42: RI-03: Datos piloto	63
Tabla 43: RI-04: Datos equipo	64
Tabla 44: RI-05: Datos carrera	64
Tabla 45: RI-06: Datos disponibilidad	64
Tabla 46: RI-07: Datos relación piloto-equipo	64
Tabla 47: RI-08: Credenciales de iRacing	65
Tabla 48: Lista de casos de uso	66
Tabla 49: CU-01: Crear piloto con ID	67
Tabla 50: CU-02: Crear piloto sin ID	68
Tabla 51: CU-03: Eliminar piloto	69
Tabla 52: CU-04: Editar piloto	70
Tabla 53: CU-05: Crear evento	71
Tabla 54: CU-06: Editar evento	72
Tabla 55: CU-07: Eliminar evento	73
Tabla 56: CU-08: Crear carrera	74
Tabla 57: CU-09: Eliminar carrera	75
Tabla 58: CU-10: Crear equipo	76
Tabla 59: CU-11: Editar equipo	77
Tabla 60: CU-12: Añadir piloto a equipo	78
Tabla 61: CU-13: Eliminar piloto de equipo	79
Tabla 62: CU-14: Eliminar equipo	80
Tabla 63: CU-15: Insertar credenciales de iRacing	81
Tabla 64: CU-16: Eliminar credenciales de iRacing	82
Tabla 65: CU-17: Editar disponibilidad del piloto	83
Tabla 66: CU-18: Crear horario manualmente	84
Tabla 67: CU-19: Generar horario automáticamente	85
Tabla 68: CU-20: Cambiar idioma del programa	86
Tabla 69: CU-21: Consultar información del programa	87
Tabla 70: Tabla especificación entidad Event	102
Tabla 71: Tabla especificación entidad Driver	103
Tabla 72: Tabla especificación entidad Team	103
Tabla 73: Tabla especificación entidad Race	104
Tabla 74: Tabla especificación entidad Availability	105
Tabla 75: Tabla especificación entidad DrivesFor	105
Tabla 76: Interfaz pestaña events	106
Tabla 77: Interfaz pestaña drivers	107
Tabla 78: Interfaz pestaña teams	108
Tabla 79: Interfaz pestaña races	109
Tabla 80: Interfaz pestaña schedules	110
Tabla 81: Interfaz ventana edit driver availability	111

Tabla 82: Interfaz ventana add driver to team	112
Tabla 83: Interfaz ventana error	113
Tabla 84: Interfaz ventana add driver with ID	114
Tabla 85: Interfaz ventana enter iRacing credentials	115
Tabla 86: Interfaz ventana about	116
Tabla 87: Interfaz opciones	116
Tabla 88: Interfaz ventana change language	117
Tabla 89: PCN-01: Prueba caja negra creación de evento	129
Tabla 90: PCN-02: Prueba caja negra creación piloto sin ID	130
Tabla 91: PCN-03: Prueba caja negra creación piloto con ID	131
Tabla 92: PCN-04: Prueba caja negra creación de equipo	132
Tabla 93: PCN-05: Prueba caja negra adición de piloto a equipo	133
Tabla 94: PCN-06: Prueba caja negra eliminación de piloto de equipo	134
Tabla 95: PCN-07: Prueba caja negra creación de carrera	135
Tabla 96: PCN-08: Prueba caja negra eliminar carrera	135
Tabla 97: PCN-09: Prueba caja negra editar disponibilidad	136
Tabla 98: PCN-10: Prueba caja negra cambiar idioma	137
Tabla 99: PCN-11: Prueba caja negra guardar credenciales	138

Índice de figuras

Figura 1: Variación de ingresos y audiencia en los e-Sports	19
Figura 2: Max Verstappen corriendo en iRacing	20
Figura 3: Rubens Barrichello posa para una foto en twitter	20
Figura 4: Árbol de características	23
Figura 5: Imagen de la pantalla del JRT en acción	24
Figura 6: Imagen calculador de datos excel	25
Figura 7: Imagen del visor de horarios de excel	25
Figura 8: Imagen de una de las aplicaciones de ATLAS	25
Figura 9: Imagen de una de las herramientas de Pi ToolBox	26
Figura 10: Logo Visual Studio	27
Figura 11: Logo GitHub	27
Figura 12: Logo NuGet	27
Figura 13: Logo WPF	28
Figura 14: Logo SQLite	28
Figura 15: Logo SQLite Browser	28
Figura 16: Logo C#	28
Figura 17: Logo XAML	29
Figura 18: Logo Xceed	29
Figura 19: Logo Selenium	29
Figura 20: Esquema de desarrollo de una aplicación ágil	35
Figura 21: Diagrama de Gantt planificación	42
Figura 22: Diagrama de casos de uso gestión de pilotos	88
Figura 23: Diagrama de casos de uso gestión de equipos	89
Figura 24: Diagrama de casos de uso gestión de eventos	89
Figura 25: Diagrama de casos de uso gestión de carreras	90
Figura 26: Diagrama casos de uso gestión de opciones	90
Figura 27: Arquitectura Lógica	93
Figura 28: Arquitectura Física	94
Figura 29: Diagrama de clases de presentación	95
Figura 30: Diagrama de clases de modelo	96
Figura 31: Diagrama de secuencia editar disponibilidad	97
Figura 32: Diagrama de secuencia guardar credenciales de iRacing	98
Figura 33: Diagrama de secuencia añadir piloto con ID	99
Figura 34: Diagrama Entidad-Relación	100
Figura 35: Diagrama relacional base de datos	101
Figura 36: Boceto pestaña Events	106
Figura 37: Boceto pestaña Drivers	107
Figura 38: Boceto pestaña Teams	108
Figura 39: Boceto pestaña Races	109

Figura 40: Boceto pestaña Schedules	110
Figura 41: Boceto ventana Edit driver availability	111
Figura 42: Boceto ventana Add driver to team	112
Figura 43: Boceto ventana Error	113
Figura 44: Boceto ventana Add driver with ID	114
Figura 45: Boceto ventana Enter iRacing credentials	115
Figura 46: Boceto menu Options	116
Figura 47: Boceto ventana About	116
Figura 48: Boceto ventana Language Selection	117
Figura 49: Análisis página login iRacing.com	120
Figura 50: Jerarquía de elementos página membersite iRacing.com	121
Figura 51: Análisis página membersite iRacing.com	122
Figura 52: Algoritmo de creación de horarios	125
Figura 53: Algoritmo de elección entre colores de fuente	126
Figura 54: Pruebas de caja negra	129
Figura 55: Prueba creación de evento	130
Figura 56: Prueba creación de piloto sin ID	131
Figura 57: Prueba creación de piloto con ID	132
Figura 58: Prueba creación de equipo	133
Figura 59: Prueba adición de piloto a equipo	134
Figura 60: Prueba eliminación piloto de equipo	135
Figura 61: Prueba crear carrera	136
Figura 62: Prueba eliminar carrera	136
Figura 63: Prueba editar disponibilidad	137
Figura 64: Prueba cambiar idioma	138
Figura 65: Prueba guardar credenciales	139
Figura 66: Estructura de la carpeta “TFG_Sergio_Hernando_Mateo_SQLite”	141
Figura 67: Manual de cambio de idioma	145
Figura 68: Manual manual de información sobre el TFG	145
Figura 69: Manual de gestión de eventos	146
Figura 70: Manual de gestión de pilotos	147
Figura 71: Manual de búsqueda de piloto con ID	148
Figura 72: Manual de inserción de credenciales	148
Figura 73: Manual de gestión de equipos	149
Figura 74: Manual de añadir un piloto a equipo	150
Figura 75: Manual de gestión de carreras	151
Figura 76: Manual de gestión de horarios	151
Figura 77: Manual de edición de disponibilidad	152

1. Capítulo 1: Introducción

Los e-sports, un término que se escucha mucho últimamente, sobre todo en los círculos de gente más joven, son aquellos videojuegos de carácter competitivo que se juegan de acuerdo a una serie de normativas estipuladas por los desarrolladores y/o entidades organizadoras de eventos.

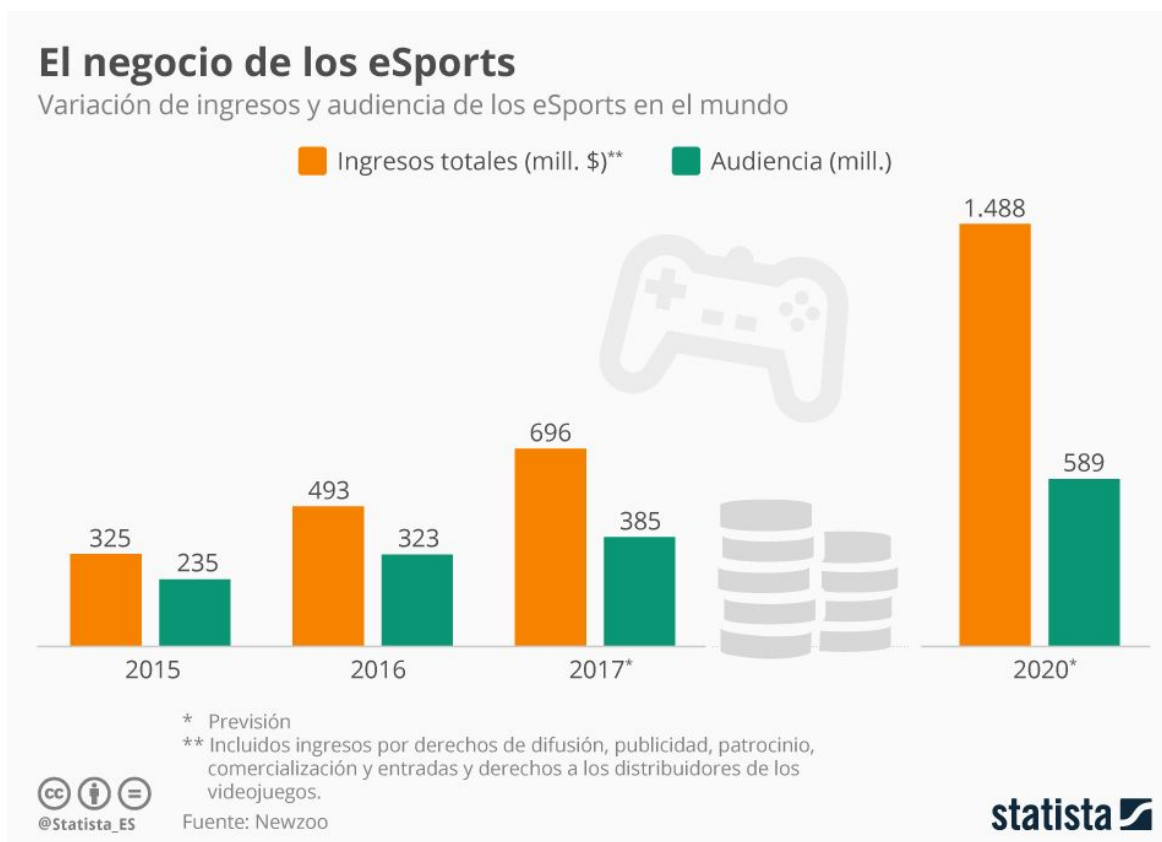


Figura 1. Variación de ingresos y audiencia de los e-Sports

Existen muchos géneros de videojuegos en los que se pueden encontrar competiciones de e-Sports. Uno de los géneros con mayor crecimiento tanto en nuestro país como en el mundo es el conocido como simracing. El simracing reúne a todos los juegos competitivos que intentan crear una simulación virtual de carreras reales en las cuales pueden competir los jugadores respetando unas normas establecidas. Dentro de este género existen muchos títulos, desde los más conocidos como Gran Turismo o Forza Motorsport, hasta los menos como Richard Burns Rally o Nascar Racing 2003.

Como reconoce Dom Duhan en la web www.motorsport.com en el artículo "Insight: Why it's time to take the Sim Racing business seriously?", *Los juegos de carreras tienen el poder de formar nuevos aficionados al automovilismo. De hecho, muchos jóvenes crecen jugando a los eSports sin seguir esas categorías del mundo real, no están interesados en el motorsport tradicional a través de las rutas tradicionales. Y dado que el automovilismo es un deporte que requiere una cantidad enorme de inversión incluso a nivel de karting, ¿por qué no entrar en el simracing?*

El videojuego que representa el estándar a seguir por los e-sports en el simracing, es el conocido como iRacing. iRacing es un videojuego con un servicio de suscripción mensual, en el cual se reúnen todo tipo de aficionados al mundo del motor y las carreras para disfrutar de una experiencia de simulación lo más parecida posible a la de aquellos afortunados que la pueden tener de verdad.

Con un “prizepool” de más de 500.000\$ repartido entre diferentes competiciones, iRacing es considerado para muchos un simple pasatiempo, pero para otros, su forma de ingresos.

iRacing también es considerado como una herramienta de entreno para muchos pilotos reales, gracias a su gran precisión en la simulación tanto de los coches como de los circuitos. Si participas en iRacing puedes competir con miles de jugadores de todas partes del mundo, incluyendo a pilotos de la talla de Max Verstappen (7 victorias en F1), Rubens Barrichello (11 victorias en F1) o Raffaele Marciello (Campeón de la F3 Europea y las GT Blancpain Series y Blancpain Sprint Series) entre otros.



Figura 2. Max Verstappen corriendo en iRacing.



Figura 3. Rubens Barrichello posa para una foto en twitter.

Un competidor de iRacing tiene la oportunidad de participar en diferentes eventos, que representan carreras de resistencia virtuales, y que tienen lugar, normalmente, una semana antes de que estos se realicen en la vida real. Estos eventos van desde las prestigiosas 24 Horas de LeMans (Tercer fin de semana de Junio, Circuit de la Sarthe, Le Mans, Francia), hasta eventos no tan conocidos en el continente europeo como los Chilli Bowl Nationals (2 semanas después de Navidad, Tulsa Expo Center, Tulsa, Oklahoma, EEUU) o los 1000 Kilómetros de Bathurst (Segundo fin de semana de Octubre, Mount Panorama Circuit, Bathurst, Nueva Gales del Sur, Australia).

En estos eventos de resistencia, dos o más pilotos se turnan el control de un mismo coche para conseguir ser los primeros en el momento que finaliza el tiempo establecido por la organización. Este tipo de carreras conllevan un enorme trabajo de gestión. Por desgracia,

los programas informáticos usados por los equipos profesionales para organizar los stints o turnos de los pilotos, la extracción de tiempos oficiales o la telemetría del coche, no están disponibles para el público en general y tampoco para los competidores de iRacing.

1.1. Motivación

Atendiendo a lo expuesto anteriormente, en este TFG se presenta el desarrollo de MyRacingManager (MRM), una aplicación que permite a cualquier persona desempeñar el rol de jefe de un equipo de carreras, dándole la oportunidad de añadir pilotos, crear equipos y eventos, y crear y gestionar los turnos de conducción de dichos pilotos. De esta forma, se facilitará el acceso a todos aquellos que lo deseen, a una herramienta que les permita disfrutar de las experiencias de carreras de resistencia.

1.2. Objetivos y alcance

El objetivo principal es el desarrollo de una aplicación de escritorio, que simplifique lo máximo posible todos los procesos de gestión de pilotos, equipos y eventos, y que presente una interfaz sencilla y amigable para el usuario. Un requisito a tener en cuenta a la hora de definir el alcance de la aplicación será el hecho de que el usuario final deberá tener un cierto conocimiento sobre el funcionamiento de una carrera de coches. En el Anexo I se facilita un glosario de términos utilizados en este proyecto que pueden ser confusos si no se está familiarizado con las carreras de coches de simulación. A continuación, en la tabla 01 se desglosa este objetivo principal en los siguientes objetivos específicos:

ID	Objetivo
OBJ-01	El programa permitirá la gestión de pilotos.
OBJ-02	El programa permitirá la gestión de equipos
OBJ-03	El programa permitirá la gestión de eventos
OBJ-04	El programa permitirá la gestión de carreras
OBJ-05	El programa permitirá la gestión de horarios de pilotaje
OBJ-06	El programa podrá ser usado por personas tanto hispanohablantes como angloparlantes

Tabla 1. Tabla de objetivos

En este sentido, y como complemento a los objetivos, se muestran las suposiciones (tabla 2) y las dependencias (tabla 3) que se han tenido en cuenta en la realización de este proyecto. Con respecto a las dependencias, decir que estas definen aspectos ajenos a la aplicación que pueden afectar a su rendimiento y su funcionamiento, y cuyos cambios no pueden ser controlados.

ID	Suposición
SUP-1	Se supone que el usuario de la aplicación tiene un mínimo conocimiento sobre el funcionamiento y la gestión de carreras de resistencia

Tabla 2. Tabla de suposiciones

ID	Dependencia
DEP-1	La creación y extracción de datos de los pilotos desde la página de iRacing dependerá de que el árbol DOM de esta no cambie y esta esté disponible.

Tabla 3. Tabla de dependencias

1.2.1. Árbol de características

El árbol de características nos ayuda a representar las diferentes funcionalidades de nuestra aplicación e identificar las tareas relacionadas con estas. La figura a continuación (Figura 4) muestra el diagrama de características en el que se detallan los aspectos más relevantes de la aplicación:

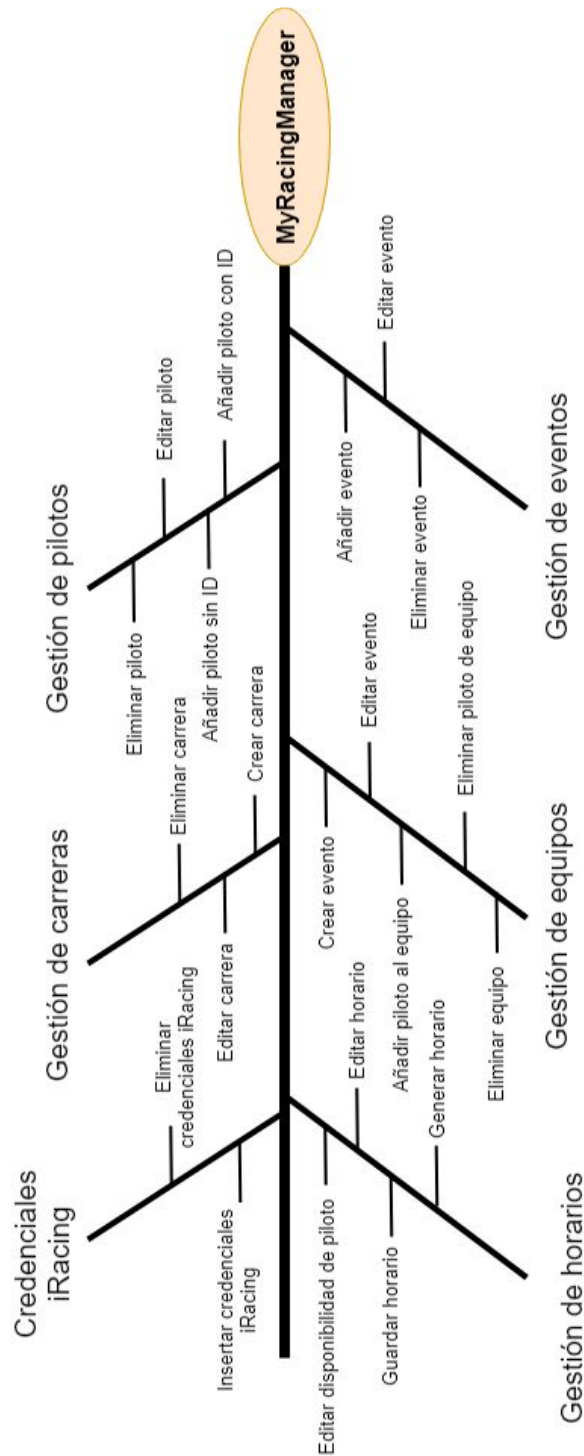


Figura 4. Árbol de características.

- Gestión de eventos: Detalla las acciones que el usuario puede realizar respecto a los eventos. Permite crear, editar y eliminar un evento.
- Gestión de pilotos: Detalla las acciones que el usuario puede realizar respecto a los pilotos. Permite crear piloto sin ID, crear piloto con ID, editar piloto y eliminar piloto.
- Gestión de equipos: Detalla las acciones que el usuario puede realizar respecto a los equipos. Permite crear, editar y eliminar un equipo, así como a su vez añadir y eliminar pilotos de dicho equipo.
- Gestión de carreras: Detalla las acciones que el usuario puede realizar respecto a las carreras. Permite crear, editar y eliminar una carrera.
- Gestión de horarios: Detalla las acciones que el usuario puede realizar respecto a los horarios. Permite generar y editar horarios, a la vez que editar la disponibilidad de los pilotos que participan en una carrera.
- Gestión de credenciales de iRacing: Detalla las acciones que el usuario puede realizar respecto a sus credenciales de inicio de sesión de iRacing. Permite insertar y eliminar las credenciales.

1.3. Estado del arte

A día de hoy no existe una aplicación software para iRacing que cumpla con los objetivos señalados con anterioridad. Sin embargo, sí que existen herramientas que permiten una gestión parcial que nos pueden servir para contextualizar nuestro proyecto. A continuación se describen las más importantes.

1.3.1. JRT (Joel Real Timing) (iRacing)

JRT es una aplicación desarrollada por Joel Guez, miembro de la comunidad de iRacing, que permite ver el estado de la carrera en tiempo real a través de un navegador web. Ofrece opciones como la visualización del mapa del circuito o la realización capturas de pantalla.

L	C	#	NAME	IR	Lic	REL	Δ	GAP	LAST	BEST	LC	SPD	Apex	Max	St	PIT	Lane	Stop	INC
1	1	13	Vortex SimRacing HPD1 Julien Oligo	5638	A 3.28	+43'41		2'07.680	2'04.706	356	133.5	136.2	198.9	11	14	1'20	18.3	167	
2	2	1	Radicals Online Wyatt Gooden	4479	A 4.81	+31'14	+12'27	2'08.774	2'04.857	350	261.4	136.1	201.8	12	14	1'33	31.1	225	
3	3	73	7even 3hree Motorsport #1 (HPD) Francisco Javier Gutierrez	4271	B 3.16	+26'56	+16'45	2'08.458	2'05.556	348	263.0	138.8	189.8	2	13	1'36	33.3	177	
4	4	012	GTL-VRT Gold Grzegorz Opalka	3684	A 3.35	+13'55	+29'46	2'13.704	2'06.696	343	157.6	105.1	263.8	17	13	1'36	31.6	121	
5	5	28	Abbandonati Racing Team Dennis Hirschka	3489	A 3.20	+4'27	+39'14	2'08.835	2'04.922	339	149.5	98.2	213.3	9	14	1'32	32.1	231	
6	6	14	Vortex SimRacing HPD2 Denis Garsse	5762	B 3.03		+43'41	2'12.440	2'04.825	337	154.7	119.4	261.3	17	14	1'35	33.5	140	
7	7	21	GTEYE Racing Max Wilson	1661	B 3.28	-19'32	+44'01	3'11.090	2'06.234	337	254.5	63.3	224.9	1	14	1'22	26.3	338	
8	8	515	Friction Racing // 24h Assault Frank Levick IV	6563	P 4.59	+8'55	+30'46	PIT	2'04.420	336		113.8	144.2	--	16	13'13	13'13	174	
9	9	18	Foracer Coanda Simsport #16 Markus Lendermann	5350	A 4.49	-29'29	+1h13	2'19.938	2'19.137	323	227.9	133.9	154.5	16	11	1'57	54.2	372	
10	10	7	Pure Racing Team Andreas Altmannberger	6084	A 3.27	-30'27	+1h14	2'20.926	2'19.761	323	106.7	54.1	215.4	24	11	1'56	54.3	650	
11	11	3	IneX Racing Dennis Hirschka	6482	B 4.00	-30'52	+1h14	2'25.551	2'19.227	322	218.4	112.6	237.1	3	12	1'50	48.2	348	

Figura 5. Imagen de la pantalla del JRT en acción.

1.3.2. Hoja Excel (iRacing)

Hoja de Excel creada por un miembro de la comunidad de iRacing que permite la creación y compartición de horarios de pilotaje manualmente.

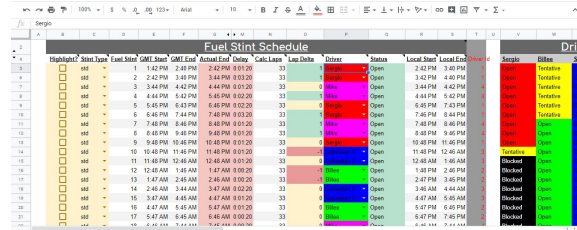
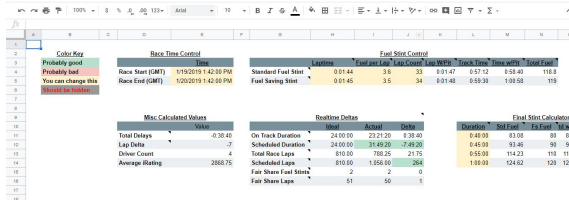


Figura 6. Imagen del calculador de datos excel. Figura 7. Imagen del visor de horarios de la hoja excel.

1.3.3. McLaren ATLAS (McLaren)

ATLAS es un paquete software creado para obtener, mostrar y analizar los datos obtenidos de sistemas de seguimiento de datos utilizados en el mundo de los deportes de motor. Creado, licenciado y mantenido por McLaren Applied Technologies. Usado por ingenieros en carreras reales.

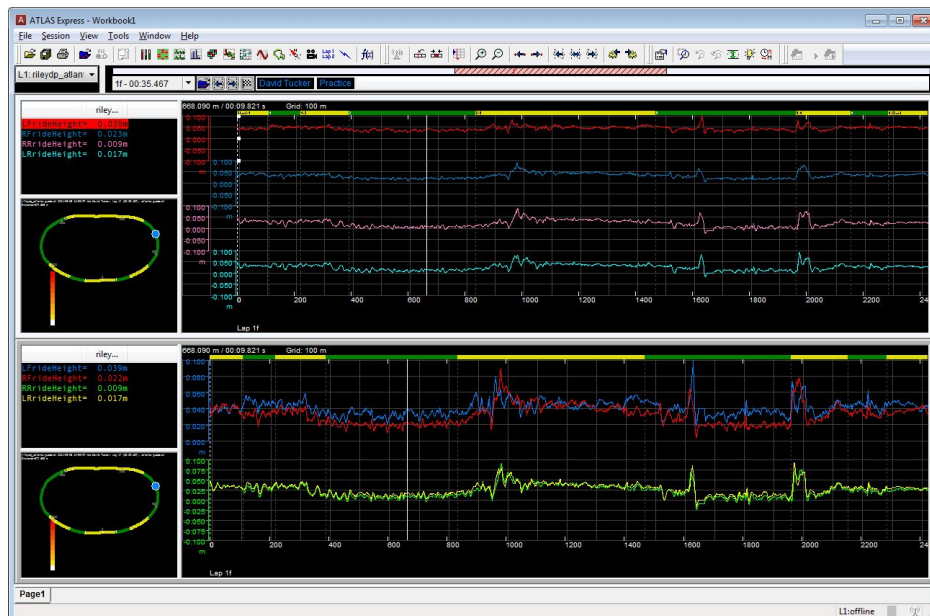


Figura 8. Imagen de una de las aplicaciones de ATLAS.

1.3.4. Pi ToolBox (Cosworth)

Pi ToolBox es una suite de potentes herramientas de análisis de datos entre otras aplicaciones entre las que se encuentran aplicaciones de telemetría, análisis de video y trazadas óptimas. Creado y licenciado por Cosworth Electronics. Usado por ingenieros reales.



Figura 9. Imagen de la pantalla de una de las herramientas de Pi ToolBox.

En la Tabla 04 se muestra una comparativa entre las diferentes herramientas que se han presentado y las funcionalidades de nuestro proyecto (MRM).

Herramienta	Seguimiento de estado de carrera	Telemetría de tiempo real	en Gestión de stints	Gestión de pilotos	de Gestión de equipos
JRT	✓	✓	✗	✗	✗
Hoja Excel	✓	✗	✓	✓	✗
McLaren ATLAS	✓	✓	●	●	●
Pi ToolBox	✓	✓	●	●	●
MRM	✗	✗	✓	✓	✓

Tabla 4. Tabla comparativa entre herramientas de seguimiento de carreras

✓ Disponible

✗ No disponible

● No hay datos

1.4. Entorno tecnológico

En este apartado se describen las herramientas, plataformas, frameworks y librerías que se han usado en la creación de este software.

1.4.1. Visual Studio



Visual Studio es un Entorno de Desarrollo Integrado o IDE para sistemas operativos Windows que permite programar en diferentes lenguajes como C++, C# o Visual Basic entre muchos otros. Este entorno de desarrollo es necesario para utilizar las tecnologías usadas.

Figura 10: Logo Visual Studio

1.4.2. GitHub



GitHub es una plataforma de desarrollo software colaborativa que utiliza el sistema de control de versiones Git. GitHub permite almacenar código de forma pública o privada. Este sistema de control de versiones nos ayuda tanto a tener un repositorio donde se encuentre una copia de seguridad, como a controlar los cambios realizados en la aplicación.

Figura 11: Logo GitHub

1.4.3. NuGet



NuGet es un gestor de paquetes para .NET. El cliente de NuGet proporciona la habilidad de consumir y producir paquetes. Este gestor facilita enormemente la instalación de extensiones en los proyectos realizados en Visual Studio

Figura 12: Logo NuGet

1.4.4. WPF (Windows Presentation Foundation)



WPF permite el desarrollo de interfaces potentes cogiendo características de Windows y aplicaciones web. WPF permite dividir el código del programa entre interfaz con el lenguaje declarativo XAML y la lógica de negocio con uno de los lenguajes de .NET, en este caso C#. WPF ha sido elegido por su facilidad de uso, capacidades de extensión, y rendimiento en Windows.

Figura 13: Logo WPF

1.4.5. SQLite



SQLite es un sistema de gestión de bases de datos relacionales de dominio público. Al contrario de la arquitectura cliente-servidor, el motor SQLite no es un proceso independiente con el que el programa se comunica, sino que SQLite se enlaza con el programa pasando a formar parte de la estructura del mismo. SQLite almacena la base de datos completa en un solo fichero en la máquina host. SQLite ha sido elegido por su facilidad de uso y integración con el programa frente a una base de datos tradicional.

Figura 14: Logo SQLite

1.4.6. SQLite Browser



SQLite Browser es una herramienta que permite crear, diseñar y editar bases de datos compatibles con SQLite. Esta herramienta será utilizada para realizar pruebas en la fase de desarrollo y monitorizar los cambios realizados a los registros de la base de datos.

Figura 15: Logo SQLite Browser

1.4.7. C# (C Sharp)



C# Es un lenguaje orientado a objetos derivado de C/C++ desarrollado por microsoft para formar parte de su plataforma .NET. Su sintaxis básica es similar a la de Java aunque incluye mejoras derivadas de otros lenguajes. C# ha sido elegido por su compatibilidad con los elementos de WPF y su capacidad de extensión.

Figura 16: Logo C#

1.4.8. XAML (eXtensible Application Markup Language)



XAML es un lenguaje declarativo basado en XML. XAML fue diseñado para ser usado con la plataforma de desarrollo .NET en los elementos que tienen interacción con la pantalla. XAML ha sido elegido por ser el lenguaje usado para la presentación de los elementos WPF.

Figura 17: Logo XAML

1.4.9. Extended WPF Toolkit (Librería)



Extended WPF Toolkit es una librería disponible para WPF que amplía las capacidades, características y elementos disponibles a la hora de crear interfaces con el lenguaje XAML. Extended WPF ha sido elegido por ser la mejor librería de extensión de WPF disponible de forma gratuita.

Figura 18: Logo Xceed

1.4.10. Selenium WebDriver



Selenium es una herramienta de automatización de buscadores web. El principal uso de la herramienta es la realización de tests automáticos en portales web. En esta aplicación se utilizará como herramienta de scraping web para la extracción de datos de la página web de iRacing.com. Selenium ha sido elegido por la facilidad que ofrece a la hora de realizar un escaneo del árbol DOM de una web.

Figura 19: Logo Selenium

1.5. Usuarios de la aplicación

Con el objetivo de un correcto diseño de la interfaz, es importante definir los principales usuarios de la aplicación y sus necesidades. También es importante definir las tareas que tendrá cada tipo de usuario en el momento de usar la aplicación.

1.5.1. Piloto de iRacing

Definimos a un piloto de iRacing como cualquier jugador de iRacing que utilice de forma completa la aplicación. Este tipo de usuario define también el usuario sobre el que se realizan las pruebas de usabilidad, debido a que es el más abundante y accesible. Los pilotos de iRacing son los usuarios más comunes de la aplicación. Debido a que el total de la aplicación está dirigido a ellos, las tareas que éstos realicen serán las que definan el total de tareas de la aplicación.

1.5.2. Piloto de simulación

Se define como piloto de simulación a todo aquel jugador de un juego de simulación que no sea iRacing, el cual usará la aplicación de forma completa exceptuando las tareas propias del simulador iRacing. Este usuario supondrá un grueso mucho menor que el piloto de iRacing, debido a ser menos común el uso de estas herramientas en otros simuladores.

1.5.3. Estrategas

Definimos a un estratega como la persona encargada de definir la estrategia de carrera dentro de un equipo de carreras profesional en carreras no virtuales. Debido a la existencia de las herramientas profesionales ya mencionadas, se espera que este tipo de usuario no se vea atraído por las funcionalidades del software desarrollado, y por tanto, que esta aplicación no sea muy usada por este colectivo.

1.5.4. Jefes de equipo

Conocemos como Jefe de equipo a aquella persona encargada del manejo de las personas que forman un equipo. Como con el anterior tipo de usuario, se espera que éste no se vea atraído por las funcionalidades del software desarrollado, y por tanto, que esta aplicación no sea muy usada por este colectivo.

1.5.5. Fan de carreras en general

Gracias a la capacidad de seguimiento de equipos y pilotos que proporciona la herramienta, podrá ser usada por cualquier persona que desee realizar un seguimiento de los eventos, equipos y pilotos que existen en su círculo de interés.

Herramienta	Gestión de stints	Gestión de pilotos	Gestión de equipos
Piloto iRacing	✓	✓	✓
Piloto simulación	✓	✓	✓
Estrategas	●	✗	✗
Jefes de equipo	✗	●	●
Fan de carreras	●	✓	✓

Tabla 5. Tabla comparativa de uso entre diferentes usuarios de la aplicación

✓ Usará la funcionalidad ✗ No usará la funcionalidad ● Posiblemente

1.6. Organización del documento

- **Capítulo 1: Introducción:** En este capítulo se identifican los objetivos, la motivación, y el estado del arte en el que se enmarca el proyecto. Se estudian también las herramientas utilizadas para la realización del proyecto y se identifican los diferentes usuarios finales de la aplicación.
- **Capítulo 2: Planificación, presupuesto y estimación:** Capítulo en el cual se presenta la planificación temporal a partir de la estimación de costes mediante puntos de función de la aplicación.
- **Capítulo 3: Análisis:** En este capítulo se identifican los requisitos de usuario, los requisitos funcionales, los requisitos no funcionales, y los de información. También se especifican los casos de uso así como los diagramas de casos de uso que los relacionan entre ellos. También se realizan el diccionario de datos, así como los diagramas de clases, de secuencia y el diagrama entidad-relación de la base de datos.
- **Capítulo 4: Diseño:** Capítulo en el que se debaten los aspectos referentes al diseño de la aplicación, tanto desde un punto de vista lógico como desde un punto de vista físico. También se especifican el diseño de las interfaces de usuario que necesita el programa.
- **Capítulo 5: Implementación:** En este capítulo se discuten aspectos de la implementación que son relevantes y, por tanto, merecen una explicación para su mejor entendimiento.
- **Capítulo 6: Pruebas:** Capítulo en el cual se describen las pruebas que se han llevado a cabo durante la implementación de la aplicación. Las pruebas realizadas han sido pruebas de caja negra.
- **Capítulo 7: Manuales de instalación:** En este capítulo se muestran las instrucciones para la correcta instalación y ejecución del programa desarrollado.

- **Capítulo 8: Manuales de usuario:** Capítulo dedicado a dar las instrucciones de como usar la aplicación y se dan consejos y advertencias sobre las consecuencias de las acciones del usuario.
- **Capítulo 9: Conclusiones y futuras mejoras:** En este capítulo se exponen las conclusiones a las que se ha llegado durante la realización de este trabajo de fin de grado. Se exponen las futuras mejoras que se podrían aplicar al proyecto en un futuro, tanto cercano como lejano.
- **Bibliografía y Webgrafía:** En este apartado se muestran los elementos bibliográficos usados para obtener información útil para la realización del proyecto, así como páginas web visitadas y útiles.
- **Glosario:** En este anexo se exponen todos los contenidos los cuales no están relacionados con ninguno de los demás capítulos, o sirven para ampliar el contexto en el que se desarrolla este proyecto.

1.7. Contenido del CD-ROM

Con este Trabajo de Fin de Grado se entregan dos tipos de CD-ROM:

- CD-ROM Biblioteca: contiene la documentación del TFG para su almacenamiento en los sistemas de información de la UVA
- CD-ROM Tribunal: contiene tres carpetas:
 - Programa: Contiene el programa realizado, compilado. Es el Programa “Final”.
 - Código: Contiene todos los archivos del proyecto sin compilar.
 - Documentación: Contiene la documentación del TFG.

2. Capítulo 2: Planificación, presupuesto y estimación

2.1. Metodología

La metodología empleada en este proyecto es la metodología ágil. Agile permite realizar de una manera iterativa una serie de prototipos útiles del programa que nos ayudan a realizar el seguimiento del proyecto a la vez que nos permiten añadir nuevas funcionalidades.

En este caso se han llevado a cabo tres iteraciones, en las cuales se vuelven a realizar todos los procesos principales del desarrollo de una aplicación (Análisis, diseño, implementación y pruebas), siendo estas:

- 1ª Iteración: Desarrollo de las funcionalidades que nos permiten introducir datos sobre pilotos, equipos y eventos. Aplicación de técnica de scraping en la página web de iRacing. Desarrollo de los mecanismos de introducción de datos sobre una carrera o evento específicos.
- 2ª Iteración: Desarrollo de las funciones necesarias para ofrecer al usuario la opción de editar los horarios creados para cada carrera.
- 3ª Iteración: Desarrollo de algoritmo de creación de “stints” y interfaz de visualización de dichos turnos.



Figura 20. Esquema de desarrollo de una aplicación ágil.

2.2. Estimación

Para la estimación en temporal y monetaria del proyecto se usa una de las técnicas aprendidas durante el desarrollo del grado, llamada “estimación por puntos de función”, la cual consiste en varios pasos:

- El primer paso, es definir una serie de características de la aplicación y agruparlas en diferentes secciones como son Entradas, salidas, consultas, ficheros internos y ficheros externos.
- Tras ello, y siguiendo las tablas de complejidad Tabla 07/08/09, se asigna a cada una de las características una complejidad (baja/media/alta).
- Después de asignar la complejidad, y para extraer los puntos de función no ajustados, realizamos una suma ponderada con los pesos presentes en la Tabla 10, la cual nos dará los puntos de función no ajustados (PFNA).
- A continuación debemos extraer el Factor de Ajuste (FA), el cual se calcula dando peso a una serie de parámetros descritos en la Tabla 11 y usando su suma en la fórmula a continuación: $FA = (0,01 * Suma) + 0,65$.
- Con el FA definido, podremos definir los puntos de función como la multiplicación de los puntos de función no ajustados por el factor de ajuste ($PF = PFNA * FA$)
- Finalmente, con los puntos de función ajustados y el tiempo por cada punto de función, obtenemos el tiempo total de desarrollo del proyecto.

Entradas	Salidas	Consultas	F. Internos	F. Externos
Información sobre pilotos (Baja)	Listado general de pilotos (Baja)	Consulta general de un piloto (Baja)	Fichero información pilotos (Baja)	Manual de instalación (Media)
Información sobre eventos (Baja)	Listado de pilotos por equipo (Baja)	Consulta modificación de piloto (Media)	Fichero información equipos (Baja)	Manual de usuario (Media)
Información sobre equipos (Media)	Listado general de eventos (Baja)	Consulta general de evento (Baja)	Fichero información eventos (Baja)	
Información sobre carreras (Media)	Listado general de equipos (Media)	Consulta general de equipo (Baja)	Fichero información carreras (Baja)	
Información sobre stints (Alta)	Listado general de carreras (Media)	Consulta modificación de equipo (Media)	Fichero información stints (Baja)	
	Listado de stints por evento (Media)	Consulta general de carrera (Baja)		

		Consulta modificación de carrera (Media)		
		Consulta general de stints (Media)		
		Consulta modificación de stints (Alta)		

Tabla 6. Tabla de salidas/entradas/consultas/fi/fe

A continuación, se muestran las tablas usadas para asignar la complejidad a los diferentes tipos de datos.

Entradas y consultas	1 a 4 TED	5 a 15 TED	>15 TED
0-1 TER Accedidos	Baja	Baja	Media
2 TER Accedidos	Baja	Media	Alta
>2 TER Accedidos	Media	Alta	Alta

Tabla 7. Tabla de complejidad entradas y consultas

Salidas	1 a 5 TED	6 a 9 TED	>19 TED
0-1 TER Accedidos	Baja	Baja	Media
2-3 TER Accedidos	Baja	Media	Alta
>3 TER Accedidos	Media	Alta	Alta

Tabla 8. Tabla de complejidad salidas

Ficheros internos y externos	1 a 19 TED	20 a 50 TED	>50 TED
1 TER Accedidos	Baja	Baja	Media
2 a 5 TER Accedidos	Baja	Media	Alta
>5 TER Accedidos	Media	Alta	Alta

Tabla 9. Tabla de complejidad ficheros externos e internos

Siendo TER → Tipos de Elementos de Registro y TED → Tipos de Elementos de Datos

Seguidamente, con la complejidad asignada, obtenemos los puntos de función no ajustados mediante una suma ponderada de las cantidades con los pesos que aparecen a continuación:

Tipo de función	Complejidad	Total Complejidad	x	Total por tipo	Suma
Ficheros internos	Simple	x7		5	35
	Media	x10		0	0
	Alta	x15		0	0
Ficheros externos	Simple	x5		0	0
	Media	x7		2	14
	Alta	x10		0	0
Entradas de usuario	Simple	x3		2	6
	Media	x4		2	8
	Alta	x6		1	6
Salidas de usuario	Simple	x4		3	12
	Media	x5		3	15
	Alta	x7		0	0
Consultas de usuario	Simple	x3		4	12
	Media	x4		4	16
	Alta	x6		1	6
Total de puntos de función no ajustados:					130 PFNA

Tabla 10. Tabla de calculo puntos de función no ajustados

	Factor de ajuste	Complejidad
1	Comunicación de datos	0
2	Funciones distribuidas	0
3	Prestaciones	3
4	Gran uso de la configuración	1
5	Velocidad de las transacciones	2
6	Entrada on-line de datos	4
7	Diseño para la eficiencia del usuario final	5
8	Actualización de datos on-line	4
9	Complejidad del proceso lógico de la aplicación	4
10	Reusabilidad de código	1
11	Facilidad de instalación	4
12	Facilidad de operación	4
13	Localizaciones múltiples	1
14	Facilidad de cambios	1
	TOTAL:	34

Tabla 11. Tabla de complejidades

$$\text{Factor de Ajuste (FA)} = (0.01 \times 34) + 0.65 = 0,99$$

$$\text{Puntos de Función Ajustados} = 130 \times 0.99 = 128.7 \approx 129 \text{ PFA}$$

$$\text{Días de proyecto} = 129 \text{ Puntos de Función} * 1 \text{ Dia/Punto de Función} = 129 \text{ Días} = 4,3 \text{ Meses}$$

2.3. Planificación

Atendiendo a la estimación realizada de 4,3 meses, se realiza una planificación de las 3 iteraciones. Como se puede observar en el diagrama de Gantt más abajo, la primera iteración es la que ocupa la mayor parte del tiempo de proyecto, dado que es la iteración donde se asientan los fundamentos de la aplicación. También influye en el tiempo planificado que la primera iteración cuenta con una de las tareas de programación más importantes y tediosas del proyecto, así como un periodo de pruebas muy extensivo.

Dado a que en el momento en el que se haya terminado la primera iteración ya se estará más familiarizado con la herramienta, el desarrollo de las demás funcionalidades debería ser más rápido y eficiente.

Para la planificación se cuenta con que el trabajo se empezará a realizar el día de Marzo de 2019, y, siguiendo la estimación, se acabará el 10 de Julio de 2019.

El horario de trabajo será el siguiente:

LUNES	MARTES	MIÉRCOLES	JUEVES	VIERNES	SÁBADO	DOMINGO
4H	4H	4H	3H	4H	2H	0H

Tabla 12. Horario de trabajo

Con este horario y nuestra planificación, el proyecto tendría una duración aproximadamente de 396 horas.

A continuación se muestra el diagrama de Gantt para la planificación realizado con *Microsoft Project*

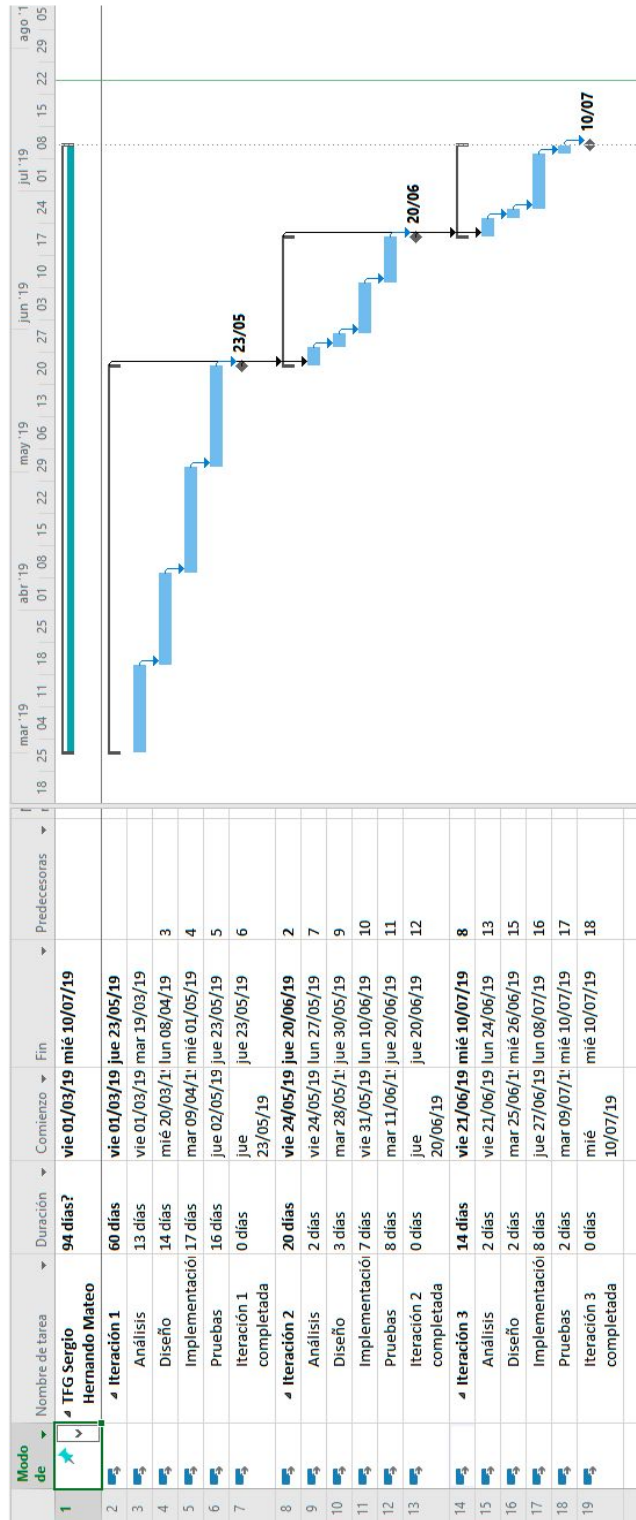


Figura 21. Diagrama de Gantt planificación

2.4. Presupuesto

Aunque el trabajo presentado ha sido realizado íntegramente por el autor de éste TFG, éste ha realizado diferentes funciones durante el desarrollo, y por ello, con el objetivo de obtener un presupuesto más ajustado a la realidad, se aplicarán los sueldos medios correspondientes a cada una de las funciones que se han realizado.

Para realizar el presupuesto de la aplicación nos basaremos en el tiempo estimado de realización de la aplicación de 4,3 meses.

2.4.1. Presupuesto en personal

El presupuesto en personal representa el coste económico del proyecto teniendo en consideración los distintos perfiles profesionales implicados en su desarrollo. Los datos de sueldo han sido extraídos de www.indeed.com.

Recurso usado	Cálculo usado para presupuestar
Analista	$1 \text{ analista} * \frac{4900 \text{€}}{\text{Mes}} * 2 \text{ Meses} = 9800\text{€}$
Programador	$1 \text{ programador} * \frac{1642 \text{€}}{\text{Mes}} * 2 \text{ Meses} = 3248\text{€}$
Tester	$1 \text{ tester} * \frac{1524 \text{€}}{\text{Mes}} * 0,3 \text{ Meses} = 457\text{€}$
TOTAL:	13505€

Tabla 13. Cálculo presupuesto en personal

2.4.2. Presupuesto en hardware

Para la elaboración del presupuesto relativo al hardware, se ha tenido en cuenta el precio medio de un ordenador con capacidades suficientes para el desarrollo del proyecto, en este caso, un ordenador de 1350€ con una vida útil de aproximadamente 3 años.

Recurso usado	Cálculo usado para presupuestar
Ordenador	$1 \text{ ordenador} * \frac{37,5 \text{€}}{\text{Mes}} * 4,3 \text{ Meses} = 150\text{€}$
TOTAL:	150€

Tabla 14. Cálculo presupuesto en hardware

2.4.3. Presupuesto en software

Para el presupuesto en software, se ha tenido en cuenta todos los gastos relacionados con el software o licencias software que se han debido adquirir para el desarrollo de proyecto.

Recurso usado	Cálculo usado para presupuestar
Microsoft Project 2019	$1 \text{ licencia} * 510\text{€} * 0,1\% \text{ uso} * 2 \text{ Meses} = 102\text{€}$
Cuenta iRacing testing	$3 \text{ meses} * 11\text{€/mes} = 33\text{€}$
TOTAL:	135€

Tabla 15. Cálculo presupuesto en software

Por tanto, el presupuesto total del proyecto, suma de los tres epígrafes anteriores, asciende a:
 $13505\text{€} + 150\text{€} + 135\text{€} = 13790 \text{€}$

3. Capítulo 3: Análisis

3.1. Requisitos de usuario

En este apartado se muestran los requisitos de usuario que se han ido extrayendo durante el transcurso de las tres iteraciones del proyecto. Estos requisitos de usuario deben describir el comportamiento funcional y no funcional del sistema de una forma tal que sean comprensibles por los usuarios de un sistema sin tener conocimiento técnico detallado.

US-ID	Iteración	Descripción
US-01	1	Crear piloto con ID
US-02	1	Crear piloto sin ID
US-03	1	Eliminar piloto
US-04	1	Editar piloto
US-05	1	Crear evento
US-06	1	Editar evento
US-07	1	Eliminar evento
US-08	1	Crear carrera
US-09	1	Eliminar carrera
US-10	1	Crear equipo
US-11	1	Editar equipo
US-12	1	Añadir piloto a equipo
US-13	1	Eliminar piloto de equipo
US-14	1	Eliminar equipo
US-15	1	Insertar credenciales de iRacing
US-16	1	Eliminar credenciales de iRacing
US-17	2	Editar disponibilidad del piloto
US-18	2	Crear horario manualmente
US-19	3	Generar horario automáticamente
US-20	3	Cambiar idioma del programa
US-21	3	Consultar información del programa

Tabla 16. Lista de requisitos de usuario

Seguidamente, se pasa a detallar cada uno de los requisitos de usuario:

US-01 Crear piloto con ID	
Descripción	El usuario puede crear un piloto introduciendo su ID de iRacing
Objetivo relacionado	OBJ-01
Prioridad	Alta
Iteración	1
Precondición	El ID de piloto introducido debe existir en la página web de iRacing
Postcondición	El piloto queda registrado junto con todos sus datos en el programa
Comentarios	

Tabla 17. US-01: Crear piloto con ID

US-02 Crear piloto sin ID	
Descripción	El usuario podrá crear un piloto introduciendo los datos del piloto
Objetivo relacionado	OBJ-01
Prioridad	Alta
Iteración	1
Precondición	
Postcondición	El piloto queda registrado junto con todos sus datos en el programa
Comentarios	

Tabla 18. US-02: Crear piloto sin ID

US-03 Eliminar Piloto	
Descripción	El usuario puede eliminar un piloto introducido en el programa
Objetivo relacionado	OBJ-01
Prioridad	Alta
Iteración	1
Precondición	El piloto debe existir en los datos del programa
Postcondición	El piloto queda eliminado del programa junto con todos sus datos
Comentarios	

Tabla 19. US-03: Eliminar piloto

US-04 Editar Piloto	
Descripción	El usuario puede editar un piloto ya guardado en el programa
Objetivo relacionado	OBJ-01
Prioridad	Alta
Iteración	1
Precondición	El piloto debe existir en los datos del programa
Postcondición	Los datos del piloto quedan cambiados en el programa
Comentarios	Los datos que se pueden editar de un piloto son: - Nombre

Tabla 20. US-04: Editar piloto

US-05 Crear evento	
Descripción	El usuario puede crear un evento
Objetivo relacionado	OBJ-03
Prioridad	Alta
Iteración	1
Precondición	
Postcondición	Los datos del evento quedan registrados en el programa
Comentarios	

Tabla 21. US-05: Crear evento

US-06 Editar evento	
Descripción	El usuario puede editar un evento
Objetivo relacionado	OBJ-03
Prioridad	Alta
Iteración	1
Precondición	Debe existir un evento dentro de los datos del programa
Postcondición	Los nuevos datos del evento quedan registrados en el programa
Comentarios	

Tabla 22. US-06: Editar evento

US-07 Eliminar evento	
Descripción	El usuario puede eliminar un evento
Objetivo relacionado	OBJ-03
Prioridad	Alta
Iteración	1
Precondición	Debe existir un evento dentro de los datos del programa
Postcondición	El evento queda eliminado de los datos del programa
Comentarios	

Tabla 23. US-07: Eliminar evento

US-08 Crear carrera	
Descripción	El usuario puede crear una carrera
Objetivo relacionado	OBJ-04
Prioridad	Alta
Iteración	1
Precondición	
Postcondición	Los datos de la carrera quedan registrados en el programa
Comentarios	

Tabla 24. US-08: Crear evento

US-09 Eliminar carrera	
Descripción	El usuario puede eliminar una carrera
Objetivo relacionado	OBJ-04
Prioridad	Alta
Iteración	1
Precondición	Debe existir una carrera en los datos del programa
Postcondición	La carrera queda eliminada del programa
Comentarios	

Tabla 25. US-09: Eliminar carrera

US-10 Crear equipo	
Descripción	El usuario puede crear un equipo dentro del programa
Objetivo relacionado	OBJ-02
Prioridad	Alta
Iteración	1
Precondición	Debe existir por lo menos 1 piloto dentro del programa
Postcondición	Queda creado un equipo con los datos especificados por el usuario
Comentarios	

Tabla 26. US-10: Crear equipo

US-11 Editar equipo	
Descripción	El usuario puede editar los datos de un equipo
Objetivo relacionado	OBJ-02
Prioridad	Alta
Iteración	1
Precondición	Debe existir un equipo en los datos del programa
Postcondición	Los datos del equipo quedan editados en el sistema
Comentarios	

Tabla 27. US-11: Editar equipo

US-12 Añadir piloto a equipo	
Descripción	El usuario puede añadir un piloto a un equipo
Objetivo relacionado	OBJ-02
Prioridad	Alta
Iteración	1
Precondición	Debe existir un equipo en los datos del programa Debe existir un piloto en los datos del programa
Postcondición	El piloto queda relacionado con el equipo
Comentarios	

Tabla 28. US-12: Añadir piloto a equipo

US-13 Eliminar piloto de equipo	
Descripción	El usuario puede eliminar un piloto de un equipo
Objetivo relacionado	OBJ-02
Prioridad	Alta
Iteración	1
Precondición	Debe existir un equipo con un piloto dentro de los datos del programa
Postcondición	Se elimina la relación del piloto con el equipo
Comentarios	

Tabla 29. US-13: Eliminar piloto de equipo

US-14 Eliminar equipo	
Descripción	El usuario puede eliminar un equipo que exista dentro del programa
Objetivo relacionado	OBJ-02
Prioridad	Alta
Iteración	1
Precondición	El equipo a eliminar debe existir dentro del programa
Postcondición	El equipo seleccionado queda eliminado de los datos del programa
Comentarios	

Tabla 30. US-14: Eliminar equipo

US-15 Insertar credenciales de iRacing	
Descripción	El usuario puede insertar sus credenciales de iRacing
Objetivo relacionado	OBJ-01
Prioridad	Alta
Iteración	1
Precondición	El usuario debe de poseer unas credenciales de iRacing válidas
Postcondición	Las credenciales de iRacing del usuario se quedan guardadas en el programa
Comentarios	

Tabla 31. US-15: Insertar credenciales de iRacing

US-16 Eliminar credenciales de iRacing	
Descripción	El usuario puede eliminar sus credenciales de iRacing
Objetivo relacionado	OBJ-01
Prioridad	Alta
Iteración	1
Precondición	Deben de existir unas credenciales de iRacing insertadas en el programa
Postcondición	Las credenciales de iRacing del usuario quedan eliminadas del programa
Comentarios	

Tabla 32. US-16: Eliminar credenciales de iRacing

US-17 Editar disponibilidad del piloto	
Descripción	El usuario puede editar la disponibilidad de un piloto para una carrera
Objetivo relacionado	OBJ-05
Prioridad	Alta
Iteración	2
Precondición	Debe existir una carrera en el sistema
Postcondición	La disponibilidad de un piloto queda guardada para su futuro uso en horarios
Comentarios	

Tabla 33. US-17: Editar disponibilidad del piloto

US-18 Crear horario manualmente	
Descripción	El usuario puede editar la disponibilidad de un piloto para una carrera
Objetivo relacionado	OBJ-05
Prioridad	Alta
Iteración	2
Precondición	Debe existir una carrera en el sistema
Postcondición	El horario de pilotaje queda guardado en el sistema
Comentarios	

Tabla 34. US-18: Crear horario manualmente

US-19 Generar horario automáticamente	
Descripción	El usuario puede crear un horario de pilotaje automáticamente
Objetivo relacionado	OBJ-05
Prioridad	Alta
Iteración	3
Precondición	Debe existir una carrera en el sistema Todos los pilotos del equipo deben tener su disponibilidad establecida Debe existir por lo menos un horario válido
Postcondición	El horario de pilotaje queda guardado en el sistema
Comentarios	

Tabla 35. US-19: Crear horario automáticamente

US-20 Cambiar el idioma del programa	
Descripción	El usuario puede cambiar el idioma del programa
Objetivo relacionado	OBJ-06
Prioridad	Baja
Iteración	3
Precondición	
Postcondición	El idioma del programa queda cambiado
Comentarios	

Tabla 36. US-20: Cambiar el idioma del programa

US-21 Consultar información del programa	
Descripción	El usuario puede consultar la información del programa
Objetivo relacionado	
Prioridad	Baja
Iteración	3
Precondición	
Postcondición	
Comentarios	

Tabla 37. US-21: Consultar la información del programa

3.2. Requisitos funcionales

A continuación se muestran los requisitos funcionales que se han ido extrayendo durante el transcurso de las tres iteraciones del proyecto. Los requisitos funcionales describen toda acción que el sistema ha de realizar.

3.2.1. Listado de requisitos funcionales

RF-ID	Iteración	Caso/s de uso/s relacionados	Descripción
RF-01	1	CU-01, CU-02,	El programa muestra al usuario la opción de crear un nuevo piloto
RF-02	1	CU-01	El programa extrae los datos de la página web de iRacing
RF-03	1	CU-01	El programa muestra al usuario los datos extraídos de la página oficial de iRacing
RF-04	1	CU-01	El programa crea un piloto con los datos extraídos
RF-05	1	CU-01, CU-02, CU-05, CU-06, CU-10, CU-11, CU-14, CU-15, CU-17, CU-21	El sistema muestra un mensaje de error
RF-06	1	CU-01, CU-15, CU-17, CU-20	El sistema cancela el proceso
RF-07	1	CU-02	El programa muestra los campos para que el usuario introduzca los datos del nuevo piloto
RF-08	1	CU-02	El programa guarda los datos del piloto sin ID en el sistema
RF-09	1	CU-03, CU-04	El sistema muestra al usuario los datos del piloto seleccionado
RF-10	1	CU-03	El sistema elimina el piloto seleccionado
RF-11	1	CU-04	El sistema guarda los nuevos datos del piloto
RF-12	1	CU-05	El sistema muestra la opción de crear un nuevo evento
RF-13	1	CU-05	El sistema guarda el nuevo evento
RF-14	1	CU-06	El sistema muestra un listado con los eventos disponibles

RF-15	1	CU-06	El sistema guarda los nuevos datos del evento
RF-16	1	CU-07	El sistema muestra la opción de eliminar evento
RF-17	1	CU-07	El sistema elimina el evento
RF-18	1	CU-08	El sistema muestra la opción de crear una carrera
RF-19	1	CU-08	El sistema guarda la carrera con todos los datos en el sistema
RF-20	1	CU-09	El sistema muestra la opción de eliminar carrera
RF-21	1	CU-09	El sistema elimina la carrera
RF-22	1	CU-10	El sistema muestra los campos a rellenar para crear un equipo
RF-23	1	CU-10	El sistema guarda la información del nuevo equipo
RF-24	1	CU-11	El sistema muestra los campos a editar de un equipo
RF-25	1	CU-11	El sistema guarda la información del equipo editado
RF-26	1	CU-12	El sistema muestra la ventana de añadir pilotos a equipo
RF-27	1	CU-12	El sistema añade el piloto al equipo
RF-28	1	CU-12, CU-13	El sistema actualiza la lista de pilotos disponibles
RF-29	1	CU-13	El sistema muestra un listado de pilotos del equipo
RF-30	1	CU-14	El sistema muestra la opción de eliminar equipo
RF-31	1	CU-14	El sistema elimina el equipo
RF-32	1	CU-15	El sistema muestra los campos para introducir las credenciales de iRacing
RF-33	1	CU-15	El sistema guarda las credenciales de iRacing
RF-34	1	CU-16	El sistema muestra la opción de eliminar las credenciales de iRacing
RF-35	1	CU-16	El sistema elimina las credenciales de iRacing
RF-36	2	CU-17	El sistema muestra el menú de edición de disponibilidad de un piloto
RF-37	2	CU-17	El sistema guarda la disponibilidad del piloto para la correspondiente carrera

RF-38	2	CU-18	El sistema muestra una lista con los stints que se han de completar en dicha carrera
RF-39	2	CU-18	El sistema guarda el horario creado manualmente por el usuario
RF-40	3	CU-19	El sistema genera el horario automáticamente
RF-41	3	CU-19	El sistema guarda el horario creado automáticamente
RF-42	3	CU-20	El sistema muestra los controles para el cambio del idioma
RF-43	3	CU-20	El sistema cambia el idioma de la aplicación
RF-44	3	CU-21	El sistema muestra la información del programa

Tabla 38. Tabla de requisitos funcionales

3.3. Requisitos no funcionales

A continuación se muestran los requisitos no funcionales que se han ido extrayendo durante el transcurso de las tres iteraciones del proyecto. Los requisitos no funcionales representan características o restricciones de la aplicación. Debido a su carácter poco específico, estos pueden estar sujeto a interpretaciones diferentes. Estos requisitos no funcionales se dividen en las siguientes categorías: Disponibilidad, accesibilidad, usabilidad, seguridad, restricciones y atributos de calidad.

3.3.1. Disponibilidad

RNF-01: La aplicación deberá estar disponible durante toda la ejecución de la misma y no permitir la parada inesperada del proceso de la aplicación.

3.3.2. Accesibilidad

RNF-02: La aplicación deberá ser accesible desde dispositivos windows.

3.3.3. Usabilidad

RNF-03: La aplicación deberá poder ser usada por un usuario con conocimientos sobre el funcionamiento de las carreras por turnos.

RNF-04: La aplicación deberá poder ajustarse a la resolución que elija el usuario proporcionando la mejor experiencia posible.

RNF-05: La aplicación deberá ofrecer una interfaz sencilla y clara para el uso.

3.3.4. Seguridad

RNF-06: La aplicación deberá ocultar en todo momento la contraseña de usuario.

RNF-07: La aplicación deberá acceder de forma segura y sin provocar denegaciones de servicio a la página web de iRacing.

RNF-08: La aplicación no deberá usar los datos del usuario para acceder a cualquier otra información que no sea la mostrada al usuario.

3.3.5. Restricciones

RNF-09: La aplicación deberá comprobar que un evento finalice después de empezar y no antes.

RNF-10: La aplicación deberá comprobar que la disponibilidad introducida es válida y en caso contrario actuar en consecuencia.

3.3.6. Atributos de calidad

RNF-11: El idioma de la aplicación será el inglés y español.

RNF-12: La aplicación deberá tener un estilo visual similar al de las demás aplicaciones.

3.4. Requisitos de información

Seguidamente se muestran los requisitos de información que se han ido extrayendo durante el transcurso de las tres iteraciones del proyecto. Estos requisitos de información representan los datos y estructuras de datos almacenados o utilizados por el sistema.

3.4.1. Listado de requisitos de información

ID	Nombre
RI-01	Email
RI-02	Datos evento
RI-03	Datos piloto
RI-04	Datos equipo
RI-05	Datos carrera
RI-06	Datos disponibilidad
RI-07	Datos relación piloto-equipo
RI-08	Credenciales iRacing

Tabla 39. Lista de requisitos de información

Seguidamente se pasa a describir los requisitos de información

RI-01 Email	
Descripción	El email de las credenciales deberá tener el formato ****@***.***
Información almacenada	- Email con formato ****@***.***

Tabla 40. RI-01: Email

RI-02 Datos evento	
Descripción	El sistema almacenará los datos de un evento
Información almacenada	<ul style="list-style-type: none"> - Fecha inicio - Fecha fin - Nombre - Imagen - Tipo de evento

Tabla 41. RI-02: Datos evento

RI-03 Datos piloto	
Descripción	El sistema almacenará los datos de un piloto
Información almacenada	<ul style="list-style-type: none"> - Nombre - iRating - sRating - ID - Color

Tabla 42. RI-03: Datos piloto

RI-04 Datos equipo	
Descripción	El sistema almacenará los datos de un equipo
Información almacenada	<ul style="list-style-type: none"> - Nombre - SOF

Tabla 43. RI-04: Datos equipo

RI-05 Datos carrera	
Descripción	El sistema almacenará los datos de una carrera
Información almacenada	<ul style="list-style-type: none"> - Equipo - Evento - Tiempo por vuelta - Medida de combustible - Combustible medio por vuelta - Tiempo por vuelta - Tiempo en boxes - Número de vueltas - Horario de pilotaje

Tabla 44. RI-05: Datos carrera

RI-06 Datos disponibilidad	
Descripción	El sistema almacenará los datos de una disponibilidad de un piloto en una carrera
Información almacenada	<ul style="list-style-type: none"> - Piloto - Carrera - Disponibilidad

Tabla 45. RI-06: Datos piloto

RI-07 Datos relación piloto-equipo	
Descripción	El sistema almacenará los datos de una relación de un piloto con un equipo
Información almacenada	<ul style="list-style-type: none"> - Piloto - Equipo

Tabla 46. RI-07: Datos piloto

RI-08 Credenciales iRacing	
Descripción	El sistema almacenará las credenciales de iRacing del usuario
Información almacenada	<ul style="list-style-type: none"> - Email - Contraseña

Tabla 47. RI-08: Credenciales iRacing

3.5. Casos de uso

A continuación se muestran los casos de uso que se han ido extrayendo durante el transcurso de las tres iteraciones del proyecto. Los casos de uso representan actividades que un usuario realiza dentro de las limitaciones de un sistema.

3.5.1. Listado de casos de uso

ID	Iteración	Nombre
CU-01	1	Crear piloto con ID
CU-02	1	Crear piloto sin ID
CU-03	1	Eliminar piloto
CU-04	1	Editar piloto
CU-05	1	Crear evento
CU-06	1	Editar evento
CU-07	1	Eliminar evento
CU-08	1	Crear carrera
CU-09	1	Eliminar carrera
CU-10	1	Crear equipo
CU-11	1	Editar equipo
CU-12	1	Añadir piloto a equipo
CU-13	1	Eliminar piloto de equipo
CU-14	1	Eliminar equipo
CU-15	1	Insertar credenciales de iRacing
CU-16	1	Eliminar credenciales de iRacing

CU-17	2	Editar disponibilidad del piloto
CU-18	2	Crear horario manualmente
CU-19	3	Generar horario automáticamente
CU-20	3	Cambiar idioma del programa
CU-21	3	Consultar información del programa

Tabla 48. Lista de casos de uso

Seguidamente, se procede a realizar la especificación de los casos de uso descritos

CU-01 Crear piloto con ID			
Descripción	El usuario de la aplicación puede crear pilotos con ID		
Iteración	1		
Precondiciones	El ID de piloto introducido debe existir en la página web de iRacing El usuario debe haber introducido sus credenciales de iRacing en el programa.		
Secuencia normal	Paso 1		El usuario selecciona la sección de pilotos
	Paso 2	RF-01	El programa muestra al usuario la opción de crear un nuevo piloto
	Paso 3		El usuario selecciona la opción de crear un nuevo piloto mediante ID de iRacing
	Paso 4		El usuario inserta el ID de iRacing del piloto elegido
	Paso 5	RF-02	El programa extrae los datos de la página web de iRacing
	Paso 6	RF-03	El programa muestra al usuario los datos extraídos de la página oficial de iRacing
	Paso 7		El usuario acepta los datos extraídos de la página web de iRacing
	Paso 8	RF-04	El programa creará un piloto con los datos extraídos
Postcondiciones	El piloto queda guardado en los datos de la aplicación		
Excepciones	Fallo a la hora de recuperar la información del piloto de la página de iRacing		
	Paso 1	RF-05	El sistema muestra un mensaje de error
	El usuario declina los datos extraídos de la página web de iRacing		

	Paso 1	RF-06	El sistema cancela el proceso
Frecuencia	Alta		
Comentarios			

Tabla 49. CU-01: Crear piloto con ID

CU-02 Crear piloto sin ID			
Descripción	El usuario de la aplicación puede crear pilotos sin ID de iracing		
Iteración	1		
Precondiciones	El Usuario debe disponer de los datos necesarios para crear el piloto.		
Secuencia normal	Paso 1		El usuario selecciona la sección de pilotos
	Paso 2	RF-01	El programa muestra al usuario la opción de crear un nuevo piloto
	Paso 3		El usuario selecciona la opción de crear un nuevo piloto sin IDI
	Paso 4	RF-07	El programa muestra los campos para que el usuario introduzca los datos del nuevo piloto
	Paso 5		El usuario inserta los datos del piloto que quiere registrar
	Paso 6		El usuario selecciona la opción de crear nuevo piloto
	Paso 7	RF-08	El programa guarda los datos del piloto sin ID en el sistema
Postcondiciones	El piloto queda guardado en los datos de la aplicación		
Excepciones	Fallo por falta de información sobre el piloto		
	Paso 1	RF-05	El programa muestra un mensaje de error
Frecuencia	Alta		
Comentarios			

Tabla 50. CU-02: Crear piloto sin ID

CU-03		Eliminar Piloto	
Descripción	El usuario de la aplicación puede eliminar pilotos		
Iteración	1		
Precondiciones	Debe de existir por lo menos un piloto registrado en la aplicación		
Secuencia normal	Paso 1		El usuario selecciona la sección de pilotos
	Paso 2		El usuario selecciona el piloto que quiere eliminar
	Paso 3	RF-09	El sistema muestra al usuario los datos del piloto seleccionado
	Paso 4		El usuario selecciona la opción de eliminar piloto
	Paso 5	RF-10	El sistema elimina el piloto seleccionado
Postcondiciones	El piloto queda eliminado de los datos de la aplicación		
Excepciones			
Frecuencia	Baja		
Comentarios	En la acción de borrado también se borra el piloto de los diferentes equipos en los que se encuentre, así como también se borran los stints que estén relacionados con este piloto.		

Tabla 51. CU-03: Eliminar piloto

CU-04		Editar Piloto	
Descripción	El usuario de la aplicación puede eliminar pilotos		
Iteración	1		
Precondiciones	Debe de existir por lo menos un piloto registrado en la aplicación		
Secuencia normal	Paso 1		El usuario selecciona la sección de pilotos
	Paso 2		El usuario selecciona el piloto que quiere editar
	Paso 3	RF-09	El sistema muestra al usuario los datos del piloto seleccionado
	Paso 4		El usuario edita los datos del piloto que desee
	Paso 5		El usuario selecciona la opción de guardar usuario
	Paso 6	RF-11	El sistema guarda los nuevos datos del piloto
Postcondiciones	Los nuevos datos del piloto quedan registrados en la aplicación		
Excepciones			
Frecuencia	Baja		
Comentarios	Solo podrán ser editados los datos del piloto personalizables, no podrán ser editados ni ID, sRating ni iRating.		

Tabla 52. CU-04: Editar piloto

CU-05 Crear evento			
Descripción	El usuario de la aplicación puede crear eventos		
Iteración	1		
Precondiciones			
Secuencia normal	Paso 1		El usuario entra en la sección de eventos
	Paso 2	RF-12	El sistema muestra la opción de crear un nuevo evento
	Paso 3		El usuario selecciona la opción de crear un nuevo evento
	Paso 4		El usuario inserta los datos del nuevo evento
	Paso 5		El usuario selecciona la opción de guardar el nuevo evento
	Paso 6	RF-13	El sistema guarda el nuevo evento.
Postcondiciones	El evento queda creado dentro del sistema.		
Excepciones	Fallo por falta de información sobre el evento		
	Paso 1	RF-05	El programa muestra un mensaje de error
Frecuencia	Media		
Comentarios			

Tabla 53. CU-05: Crear evento

CU-06 Editar Evento			
Descripción	El usuario de la aplicación puede editar eventos		
Iteración	1		
Precondiciones	Tiene que existir un evento creado anteriormente		
Secuencia normal	Paso 1		El usuario entra en la sección de eventos
	Paso 2	RF-14	El sistema muestra un listado con los eventos disponibles
	Paso 3		El usuario selecciona el evento a editar
	Paso 4		El usuario inserta los nuevos datos del evento
	Paso 5		El usuario selecciona la opción de guardar los nuevos datos del evento
	Paso 6	RF-15	El sistema guarda los nuevos datos del evento
Postcondiciones	El evento queda editado dentro del sistema.		
Excepciones	Fallo por falta de información sobre el evento		
	Paso 1	RF-05	El programa muestra un mensaje de error
Frecuencia	Baja		
Comentarios			

Tabla 54. CU-06: Editar evento

CU-07		Eliminar Evento	
Descripción	El usuario de la aplicación puede eliminar eventos		
Iteración	1		
Precondiciones	Debe de existir por lo menos un evento en la aplicación		
Secuencia normal	Paso 1		El usuario selecciona la sección de eventos
	Paso 2		El usuario selecciona el evento que quiere eliminar
	Paso 3	RF-16	El sistema muestra la opción de eliminar evento
	Paso 4		El usuario selecciona la opción de eliminar evento
	Paso 5	RF-17	El sistema elimina el evento
Postcondiciones	El evento queda eliminado de los datos de la aplicación		
Excepciones			
Frecuencia	Baja		
Comentarios	En la acción de borrado de un evento también se borran los stints y carreras de ese evento.		

Tabla 55. CU-07: Eliminar evento

CU-08		Crear carrera	
Descripción	El usuario de la aplicación puede crear carreras		
Iteración	1		
Precondiciones	Debe de existir por lo menos un evento y un equipo en la aplicación		
Secuencia normal	Paso 1		El usuario selecciona la sección de carreras
	Paso 2	RF-18	El sistema muestra la opción de crear una carrera
	Paso 3		El usuario selecciona el evento donde tiene lugar la carrera
	Paso 4		El usuario selecciona el equipo que participa en esa carrera
	Paso 5		El usuario introduce los datos de la carrera
	Paso 6		El usuario selecciona la opción de crear carrera
	Paso 7	RF-19	El sistema guarda la carrera con todos los datos en el sistema
Postcondiciones	La carrera queda guardada en el sistema		
Excepciones			
Frecuencia	Alta		
Comentarios			

Tabla 56. CU-08: Crear carrera

CU-09		Eliminar carrera	
Descripción	El usuario de la aplicación puede eliminar carreras		
Iteración	1		
Precondiciones	Debe existir por lo menos una carrera dentro del sistema		
Secuencia normal	Paso 1		El usuario selecciona la sección de carreras
	Paso 2		El usuario selecciona la carrera a eliminar
	Paso 3	RF-20	El sistema muestra la opción de eliminar carrera
	Paso 4		El usuario selecciona la opción de eliminar carrera
	Paso 5	RF-21	El sistema elimina la carrera
Postcondiciones	La carrera queda guardada en el sistema		
Excepciones			
Frecuencia	Alta		
Comentarios			

Tabla 57. CU-09: Eliminar carrera

CU-10			
Crear equipo			
Descripción	El usuario puede crear un nuevo equipo		
Iteración	1		
Precondiciones			
Secuencia normal	Paso 1		El usuario selecciona la pestaña equipos
	Paso 2		El usuario selecciona la opción de crear equipo
	Paso 3	RF-22	El sistema muestra los campos a rellenar para crear un equipo
	Paso 4		El usuario rellena los campos necesarios para la creación de un equipo
	Paso 5		El usuario selecciona la opción de añadir nuevo equipo
	Paso 6	RF-23	El sistema guarda la información del nuevo equipo
Postcondiciones	Un nuevo equipo está creado en el sistema		
Excepciones	Los datos introducidos no son válidos		
	Paso 1	RF-05	El programa muestra un mensaje de error.
Frecuencia	Alta		
Comentarios			

Tabla 58. CU-10: Crear equipo

CU-11 Editar equipo			
Descripción	El usuario puede editar un equipo		
Iteración	1		
Precondiciones	Debe existir un equipo dentro del sistema		
Secuencia normal	Paso 1		El usuario selecciona la pestaña equipos
	Paso 2		El usuario selecciona el equipo a editar
	Paso 3	RF-24	El sistema muestra los campos a editar de un equipo
	Paso 4		El usuario rellena los campos que desea editar del equipo
	Paso 5		El usuario selecciona la opción de editar equipo
	Paso 6	RF-25	El sistema guarda la información del equipo editado
Postcondiciones	Los datos del equipo quedan editados en el sistema		
Excepciones	Los datos introducidos no son válidos		
	Paso 1	RF-05	El programa muestra un mensaje de error.
Frecuencia	Alta		
Comentarios			

Tabla 59. CU-11: Editar equipo

CU-12 Añadir piloto a equipo			
Descripción	El usuario puede añadir un piloto a un equipo		
Iteración	1		
Precondiciones	Debe existir un equipo dentro del sistema Debe existir un piloto en el sistema		
Secuencia normal	Paso 1		El usuario selecciona la pestaña equipos
	Paso 2		El usuario selecciona el equipo al que añadir un piloto
	Paso 3	RF-26	El sistema muestra la ventana de añadir pilotos a equipo
	Paso 4		El usuario selecciona el piloto que desea añadir al equipo
	Paso 5		El usuario selecciona la opción de añadir piloto a equipo
	Paso 6	RF-27	El sistema añade el piloto al equipo
	Paso 7	RF-28	El sistema actualiza la lista de pilotos disponibles
Postcondiciones	Los pilotos quedan relacionados con el equipo		
Excepciones			
Frecuencia	Alta		
Comentarios			

Tabla 60. CU-12: Añadir piloto a equipo

CU-13		Eliminar piloto de equipo	
Descripción	El usuario puede eliminar un piloto de un equipo		
Iteración	1		
Precondiciones	Debe existir un equipo dentro del sistema Debe existir un piloto relacionado con el equipo		
Secuencia normal	Paso 1		El usuario selecciona la pestaña equipos
	Paso 2		El usuario selecciona el equipo del que quiere eliminar un piloto
	Paso 3	RF-29	El sistema muestra el listado de pilotos del equipo
	Paso 4		El usuario selecciona el piloto que desea eliminar del equipo
	Paso 5		El usuario selecciona la opción de eliminar piloto
	Paso 6	RF-30	El sistema elimina el piloto del equipo
	Paso 7	RF-28	El sistema actualiza la lista de pilotos disponibles
Postcondiciones	Los pilotos quedan relacionados con el equipo		
Excepciones			
Frecuencia	Alta		
Comentarios			

Tabla 61. CU-13: Eliminar piloto de equipo

CU-14 Eliminar equipo			
Descripción	El usuario puede eliminar un equip		
Iteración	1		
Precondiciones	Debe existir un equipo dentro del sistema		
Secuencia normal	Paso 1		El usuario selecciona la pestaña equipos
	Paso 2		El usuario selecciona el equipo que desea eliminar
	Paso 3	RF-30	El sistema muestra la opción de eliminar equipo
	Paso 4		El usuario selecciona la opción de eliminar equipo
	Paso 5	RF-31	El sistema elimina el equipo
Postcondiciones	El equipo queda eliminado del sistema		
Excepciones	Los datos introducidos no son válidos		
	Paso 1	RF-05	El sistema muestra un mensaje de error
Frecuencia	Media		
Comentarios			

Tabla 62. CU-14: Eliminar equipo

CU-15 Insertar credenciales de iRacing			
Descripción	El usuario puede insertar sus credenciales de iRacing		
Iteración	1		
Precondiciones	El usuario ha de tener una cuenta activa en iRacing		
Secuencia normal	Paso 1		El usuario selecciona la opción de insertar credenciales de iRacing en el programa
	Paso 2	RF-32	El sistema muestra los campos para introducir las credenciales de iRacing
	Paso 3		El usuario introduce las credenciales de iRacing
	Paso 4		El usuario selecciona guardar las credenciales de iRacing
	Paso 5	RF-33	El sistema guarda las credenciales de iRacing
Postcondiciones	Las credenciales del usuario quedan guardadas en el programa		
Excepciones	Las contraseñas no coinciden		
	Paso 1	RF-05	El sistema muestra un mensaje de error.
	El email tiene un formato incorrecto		
	Paso 1	RF-05	El sistema muestra un mensaje de error.
	El usuario cancela la introducción de credenciales		
	Paso 1	RF-06	El sistema cancela el proceso
Frecuencia	Baja		
Comentarios			

Tabla 63. CU-15: Introducir credenciales de iRacing

CU-16 Eliminar credenciales de iRacing			
Descripción	El usuario puede eliminar sus credenciales de iRacing		
Iteración	1		
Precondiciones	El usuario ha de tener una cuenta activa en iRacing Las credenciales de iRacing han de estar insertadas en el programa previamente		
Secuencia normal	Paso 1		El usuario selecciona la opción de insertar credenciales de iRacing en el programa
	Paso 2	RF-34	El sistema muestra la opción de eliminar las credenciales de iRacing
	Paso 3		El usuario selecciona la opción de eliminar las credenciales de iRacing
	Paso 4	RF-35	El sistema elimina las credenciales de iRacing
Postcondiciones	Las credenciales del usuario quedan eliminadas del programa		
Excepciones			
Frecuencia	Baja		
Comentarios			

Tabla 64. CU-16: Eliminar credenciales de iRacing

CU-17 Editar disponibilidad del piloto			
Descripción	El usuario de la aplicación puede editar la disponibilidad de un piloto durante una carrera		
Iteración	2		
Precondiciones	Debe existir por lo menos una carrera dentro del sistema Debe existir un piloto en el equipo que está registrado a la carrera		
Secuencia normal	Paso 1		El usuario selecciona un piloto de la lista en la pestaña de schedules
	Paso 2		El usuario selecciona la opción de editar disponibilidad
	Paso 3	RF-36	El sistema muestra el menú de edición de disponibilidad de un piloto
	Paso 4		El usuario selecciona la disponibilidad del piloto para cada uno de los stints de la carrera
	Paso 5		El usuario selecciona la opción de guardar disponibilidad del piloto para la carrera
	Paso 6	RF-37	El sistema guarda la disponibilidad del piloto para la correspondiente carrera.
Postcondiciones	La disponibilidad del piloto queda registrada en el programa		
Excepciones	El usuario selecciona la opción de cancelar disponibilidad		
	Paso 1	RF-06	El sistema cancela el proceso
	Paso 2	RF-05	El sistema muestra un mensaje de error
Frecuencia	Alta		
Comentarios			

Tabla 65. CU-17: Editar disponibilidad del piloto

CU-18		Crear horario manualmente	
Descripción	El usuario de la aplicación puede crear un horario de carrera manualmente		
Iteración	2		
Precondiciones	Debe existir por lo menos una carrera para poder crear algún horario		
Secuencia normal	Paso 1		El usuario selecciona una carrera para crear el horario de esta
	Paso 2	RF-38	El sistema muestra una lista con los stints que se han de completar en dicha carrera
	Paso 3		El usuario selecciona el piloto que desea que haga cada stint
	Paso 4		El usuario selecciona la opción de guardar el horario creado
	Paso 5	RF-39	El sistema guarda el horario creado manualmente por el usuario
Postcondiciones	El horario creado queda registrado en el programa		
Excepciones			
Frecuencia	Alta		
Comentarios			

Tabla 66. CU-18: Crear horario manualmente

CU-19		Generar horario automáticamente	
Descripción	El usuario de la aplicación puede crear un horario de carrera automáticamente		
Iteración	3		
Precondiciones	Debe existir por lo menos una carrera para poder crear algún horario		
Secuencia normal	Paso 1		El usuario selecciona una carrera para crear el horario de ésta
	Paso 2		El usuario selecciona la opción de generar horario automáticamente
	Paso 3	RF-40	El sistema genera el horario automáticamente
	Paso 4		El usuario selecciona la opción de guardar horario
	Paso 5	RF-41	El sistema guarda el horario creado automáticamente
Postcondiciones	El horario creado queda registrado en el programa		
Excepciones			
Frecuencia	Alta		
Comentarios			

Tabla 67. CU-19: Generar horario automáticamente

CU-20 Cambiar idioma del programa			
Descripción	El usuario puede cambiar el idioma del programa		
Iteración	3		
Precondiciones			
Secuencia normal	Paso 1		El usuario selecciona la opción de cambio de idioma
	Paso 2	RF-42	El sistema muestra los controles para el cambio de idioma
	Paso 3		El usuario selecciona el idioma que desea
	Paso 4		El usuario acepta el cambio de idioma
	Paso 5	RF-43	El sistema cambia el idioma de la aplicación
Postcondiciones	El idioma de aplicación queda cambiado al seleccionado por el usuario		
Excepciones	El usuario cancela el cambio de idioma		
	Paso 1	RF-06	El programa cancela el proceso
Frecuencia	Alta		
Comentarios			

Tabla 68. CU-20: Cambiar idioma del programa

CU-21 Consultar información del programa			
Descripción	El usuario puede cambiar el idioma del programa		
Iteración	3		
Precondiciones			
Secuencia normal	Paso 1		El usuario selecciona la opción de información sobre el programa
	Paso 2	RF-44	El sistema muestra la información sobre el programa
Postcondiciones			
Excepciones	Los datos introducidos no son válidos		
	Paso 1	RF-05	El programa muestra un mensaje de error.
Frecuencia	Alta		
Comentarios			

Tabla 69. CU-21: Consultar información del programa

3.5.2. Diagramas de casos de uso

Los diagramas de caso de uso representan una serie de acciones que se realizan por un usuario dentro de un sistema y muestran la relación entre los casos de uso de un sistema y sus actores.

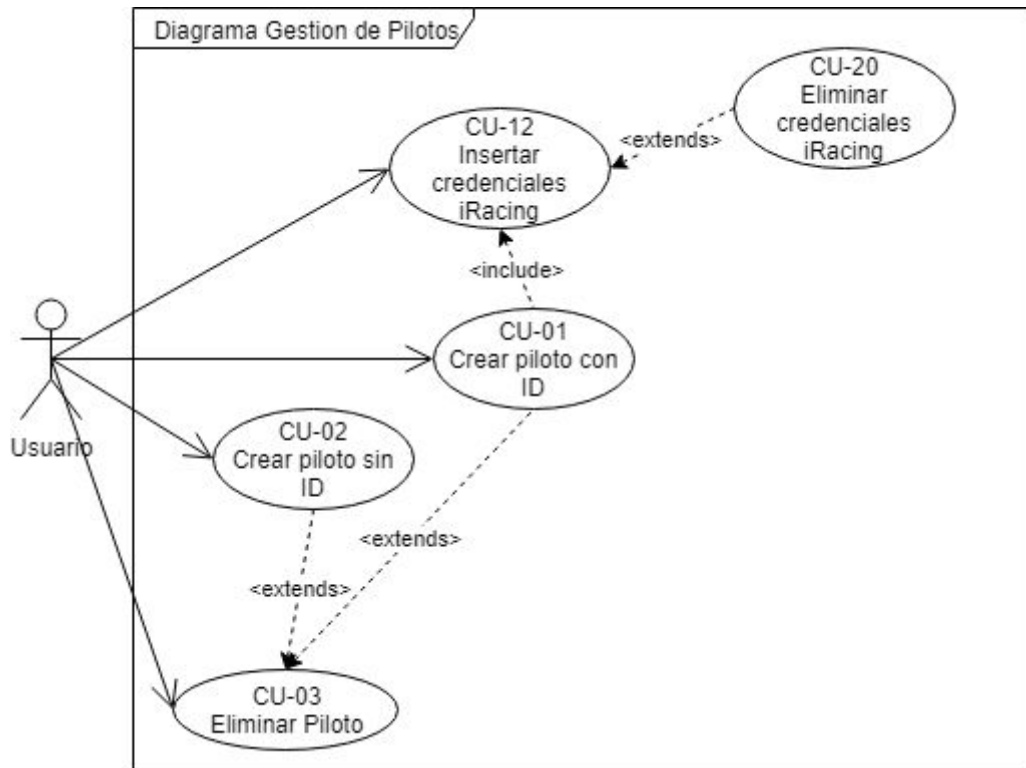


Figura 22. Diagrama casos de uso gestión de pilotos

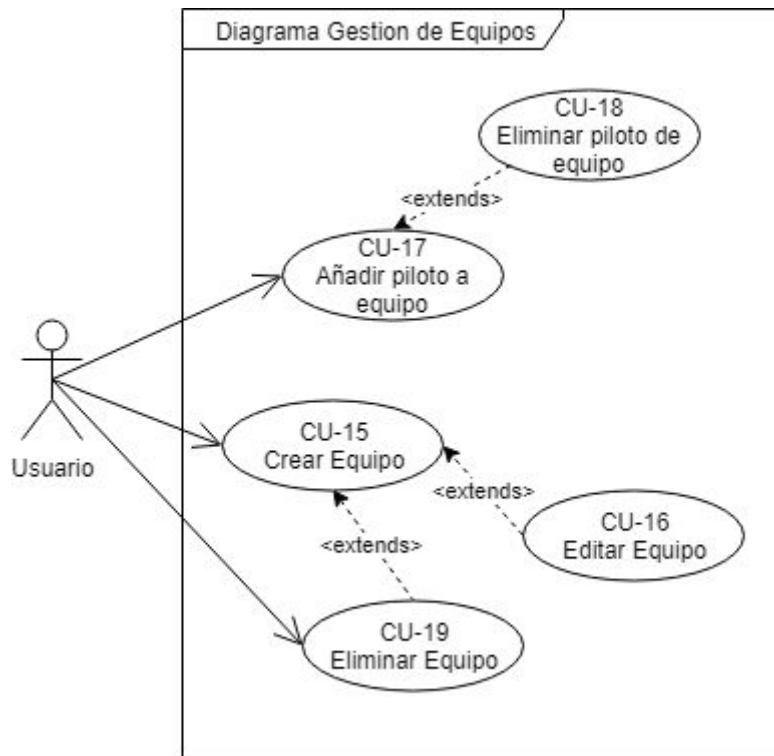


Figura 23. Diagrama casos de uso gestión de equipos

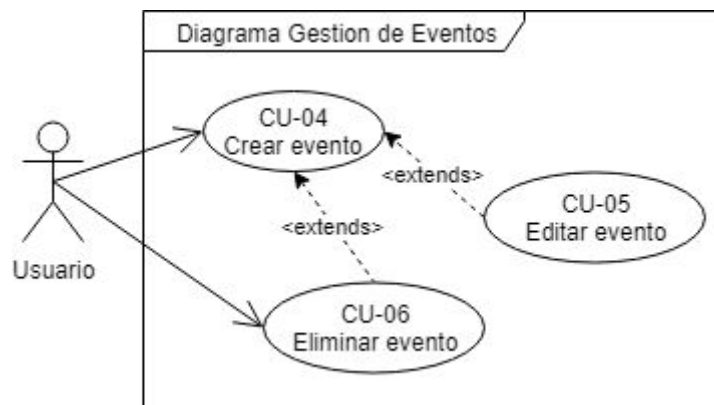


Figura 24. Diagrama casos de uso gestión de eventos

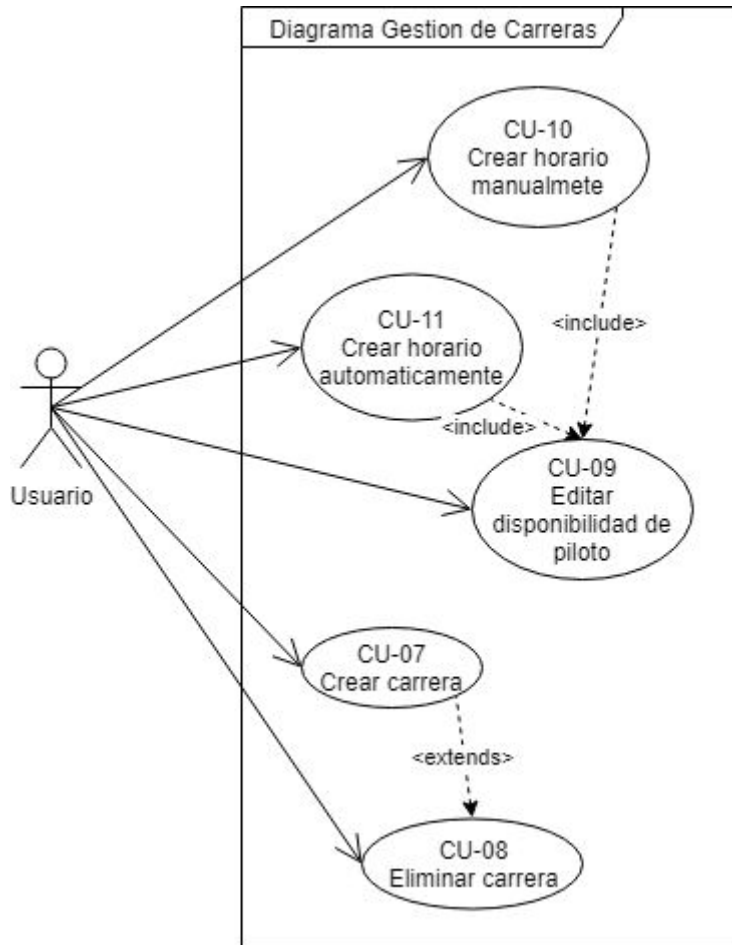


Figura 25. Diagrama casos de uso gestión de carreras

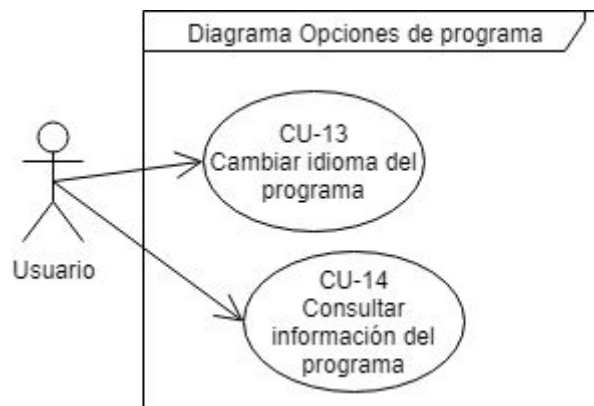


Figura 26. Diagrama casos de uso gestión de opciones

4. Capítulo 4: Diseño

4.1. Arquitectura lógica

La arquitectura lógica representa la organización del proyecto y las dependencias entre sus capas, paquetes y elementos en general.

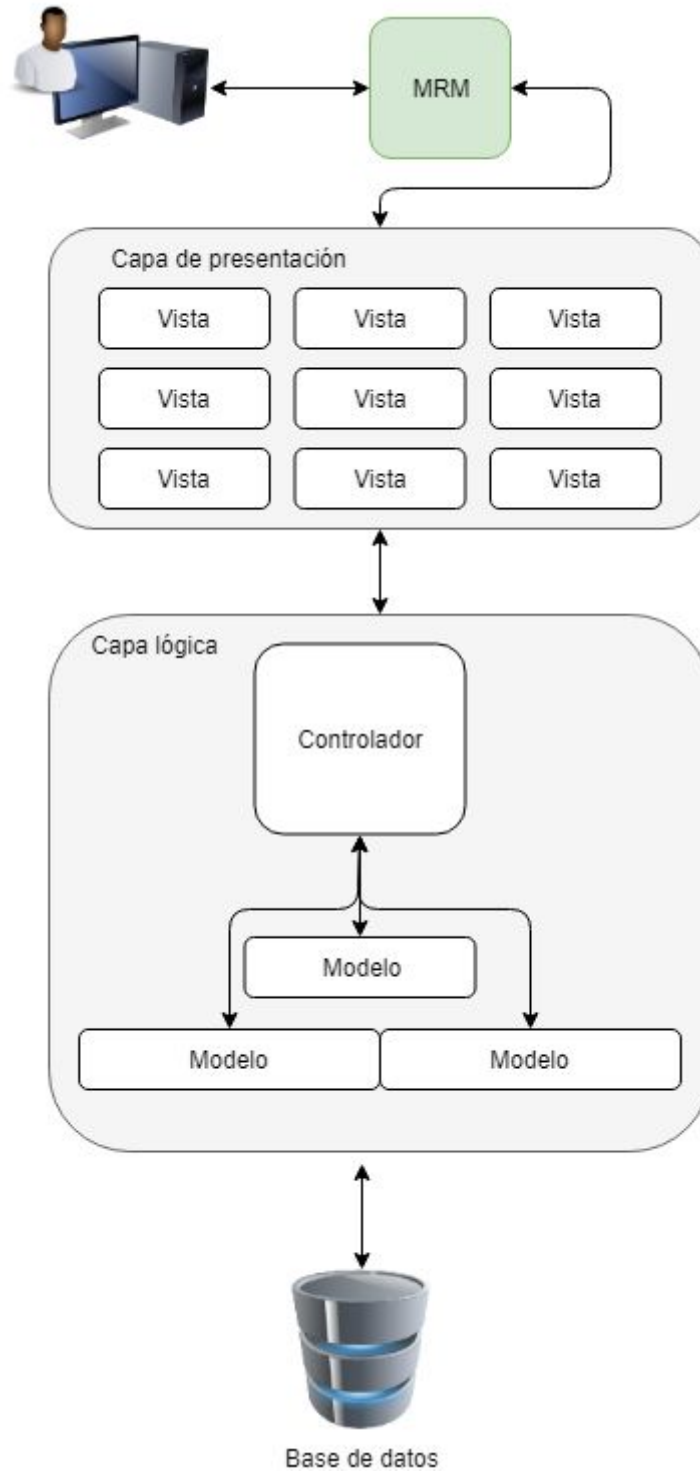


Figura 27. Arquitectura lógica

4.2. Arquitectura física

La arquitectura física representa los diferentes elementos físicos que participan en nuestro software, así como las diferentes tecnologías involucradas en el desarrollo de la aplicación.

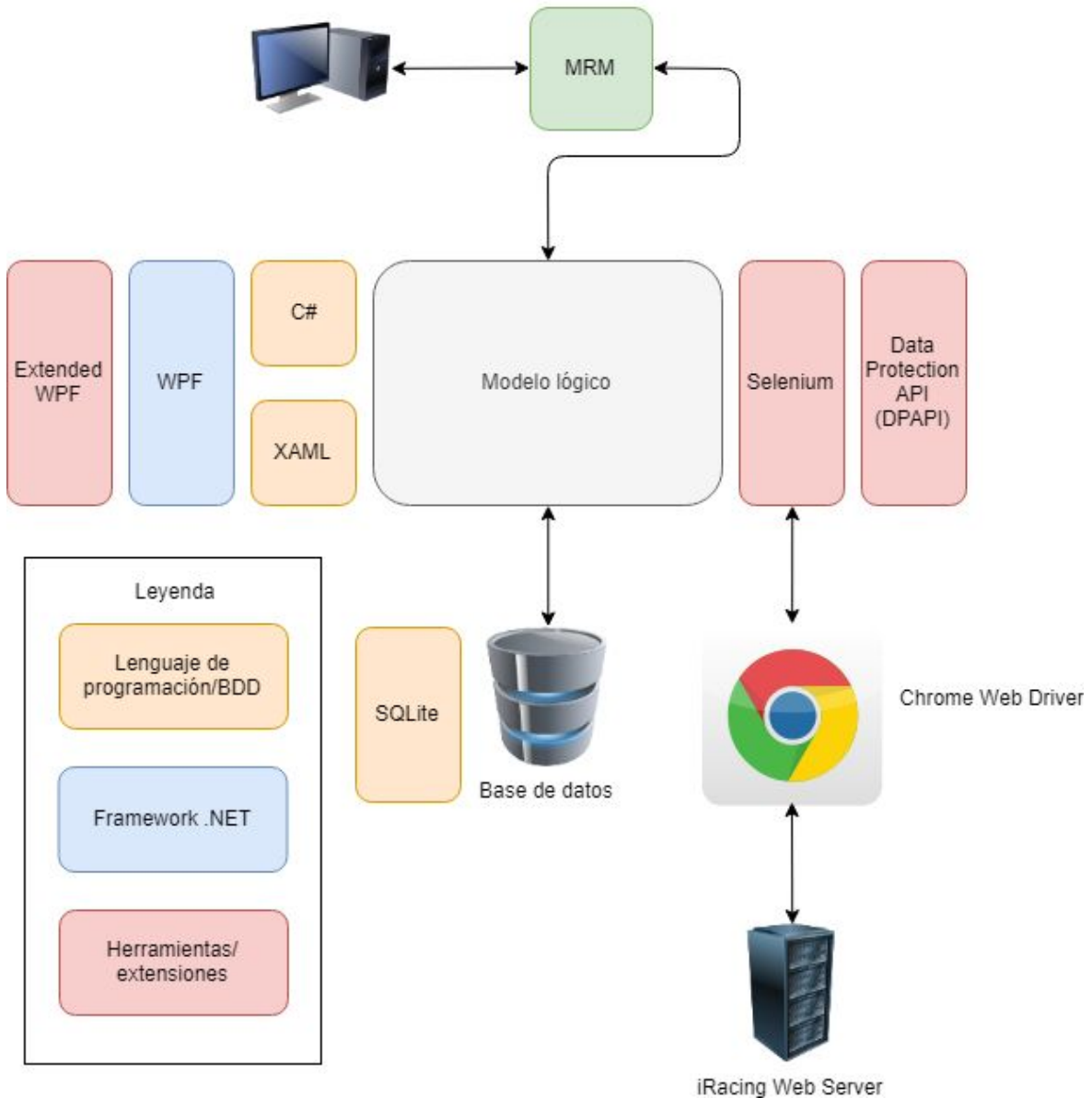


Figura 28. Arquitectura física

4.3. Diagramas de clases

La tecnología WPF utilizada tiene la característica de que relaciona cada clase de presentación con una clase del modelo. Debido a esto se genera muchas clases dentro del proyecto, por lo cual, se ha decidido dividir el diagrama de clases en dos. El primer diagrama representa las clases que están relacionadas con alguna capa de presentación y el segundo las clases que solo forman parte del modelo. Para asegurar la simplicidad del diagrama de clases de presentación no se incluyen los event handlers de cada control, los cuales sí que se encuentran dentro de la clase.

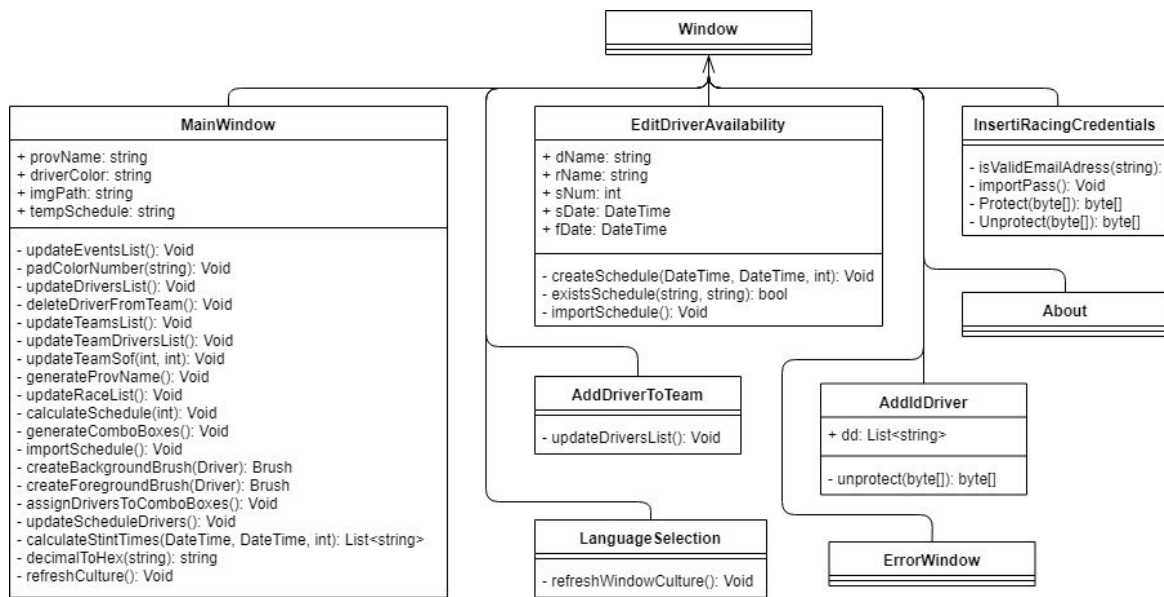


Figura 29. Diagrama de clases de presentación

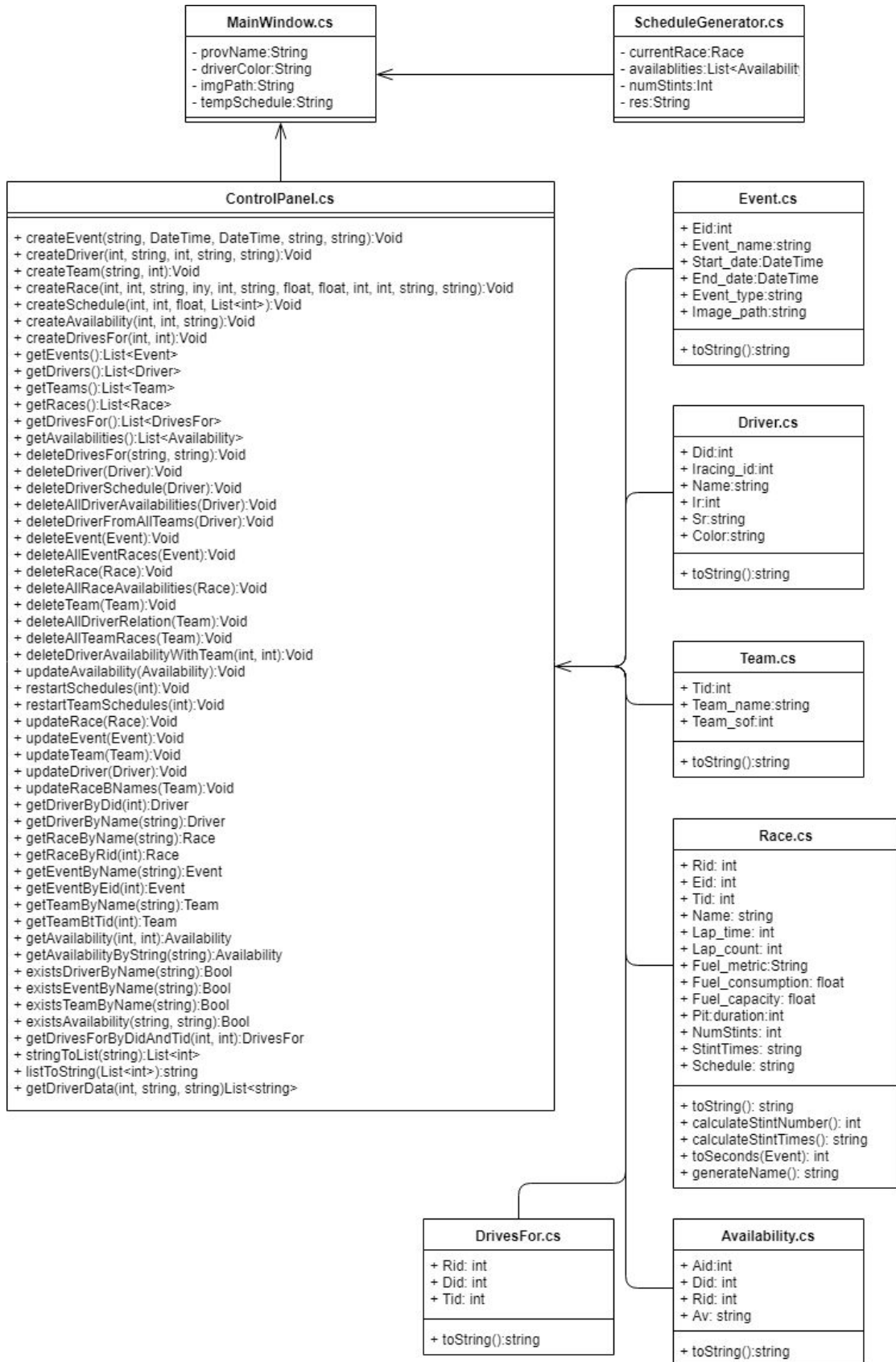


Figura 30. Diagrama de clases de modelo

4.4. Diagramas de secuencia

Los diagramas de secuencia muestran la interacción entre los diferentes módulos de la aplicación de una forma temporal y de forma que se diferencian los mensajes que se realizan entre ellos.

Debido a que muchos de los casos de uso tendrían diagramas de secuencia similares solo se realizan los diagramas más significativos.

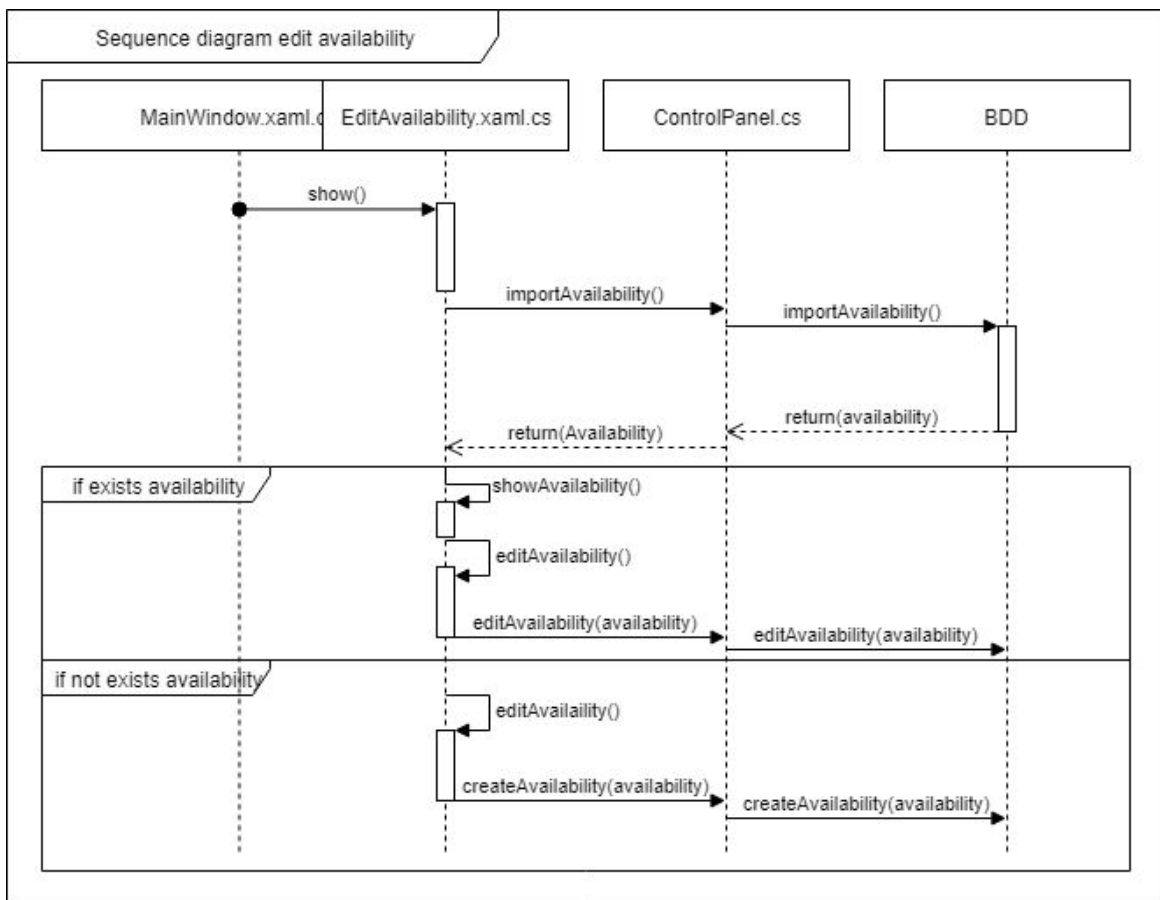


Figura 31. Diagrama de secuencia editar disponibilidad

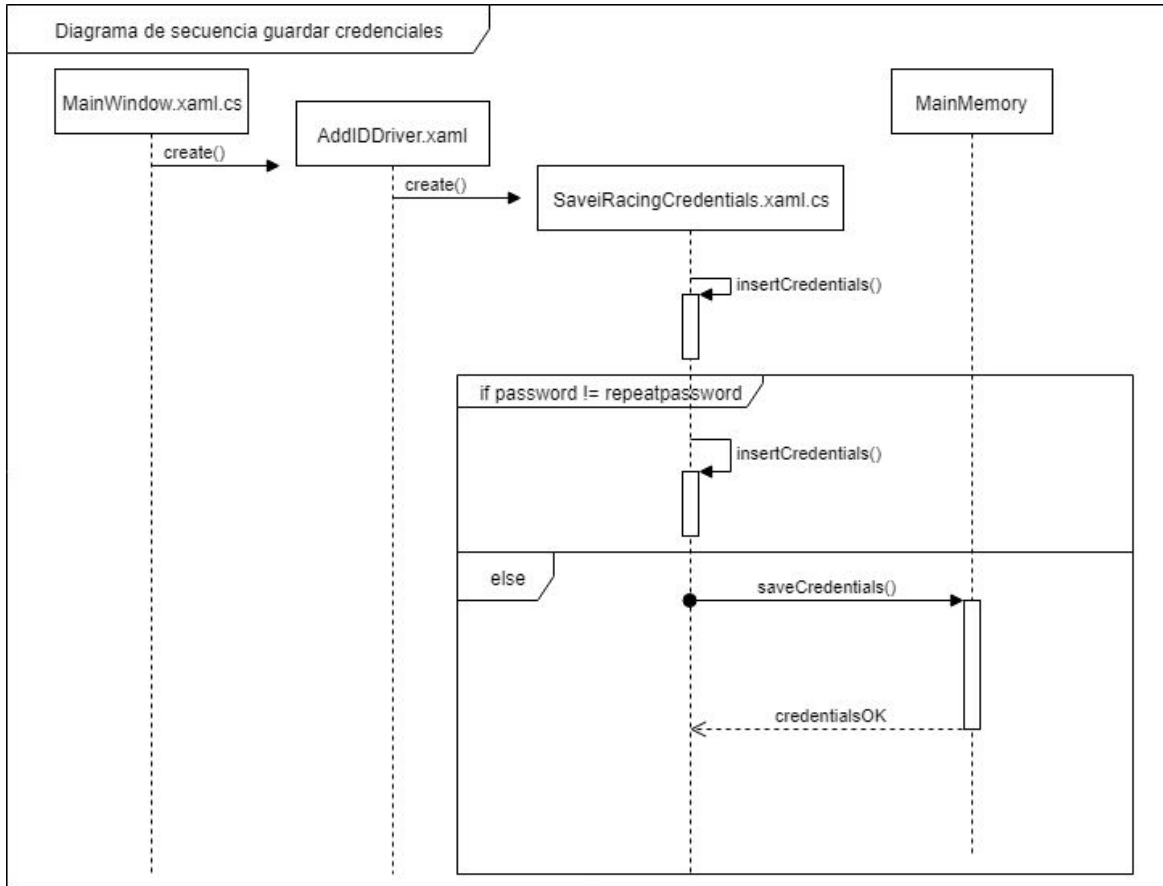


Figura 32. Diagrama de secuencia guardar credenciales de iRacing

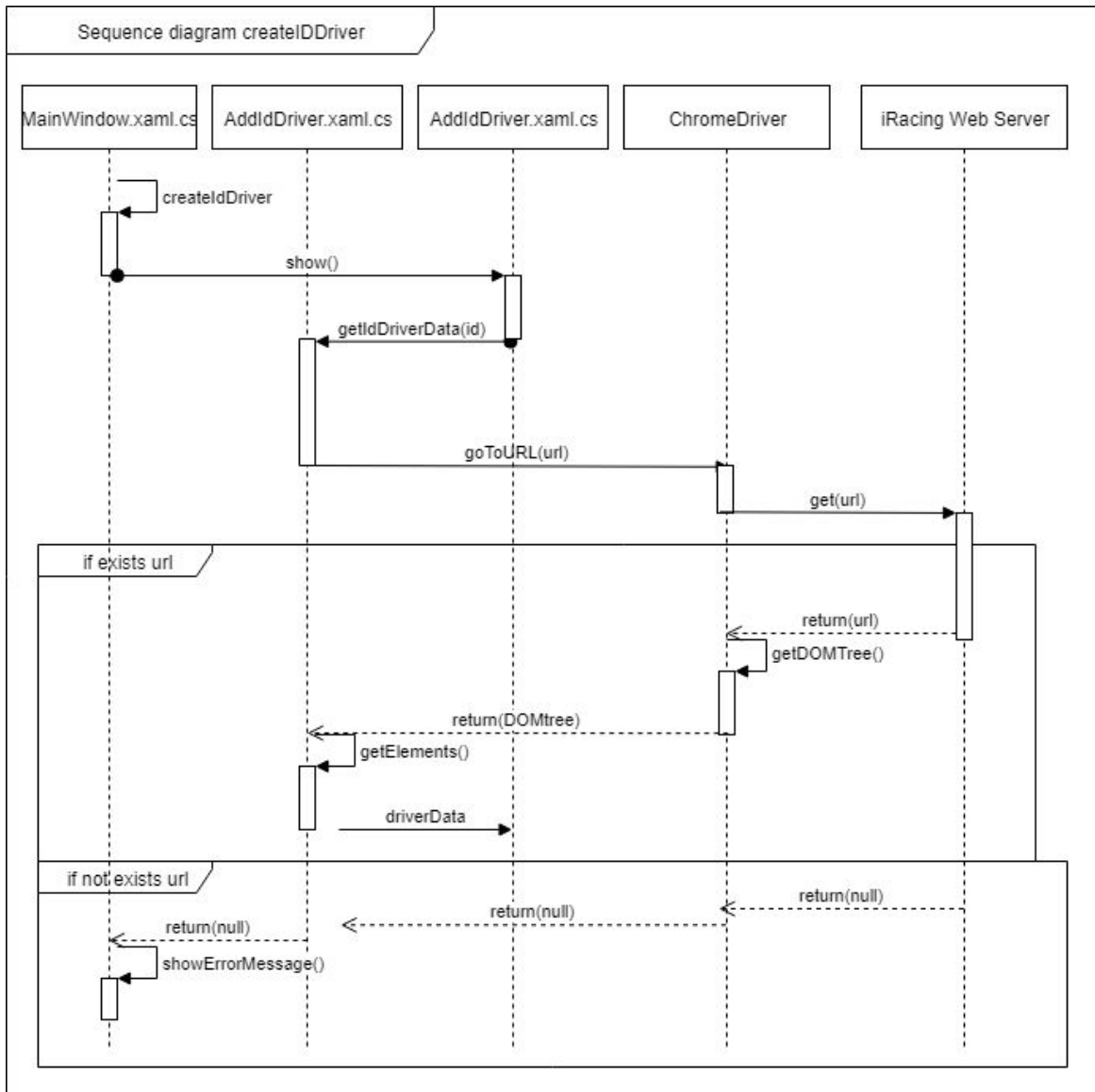


Figura 33. Diagrama de secuencia Añadir piloto con ID

4.5. Diagrama entidad-relación de la base de datos

El diagrama entidad-relación de la base de datos representa la relación entre las diferentes entidades de la base de datos, mostrando sus atributos. Seguidamente se muestra el perteneciente a la aplicación desarrollada.

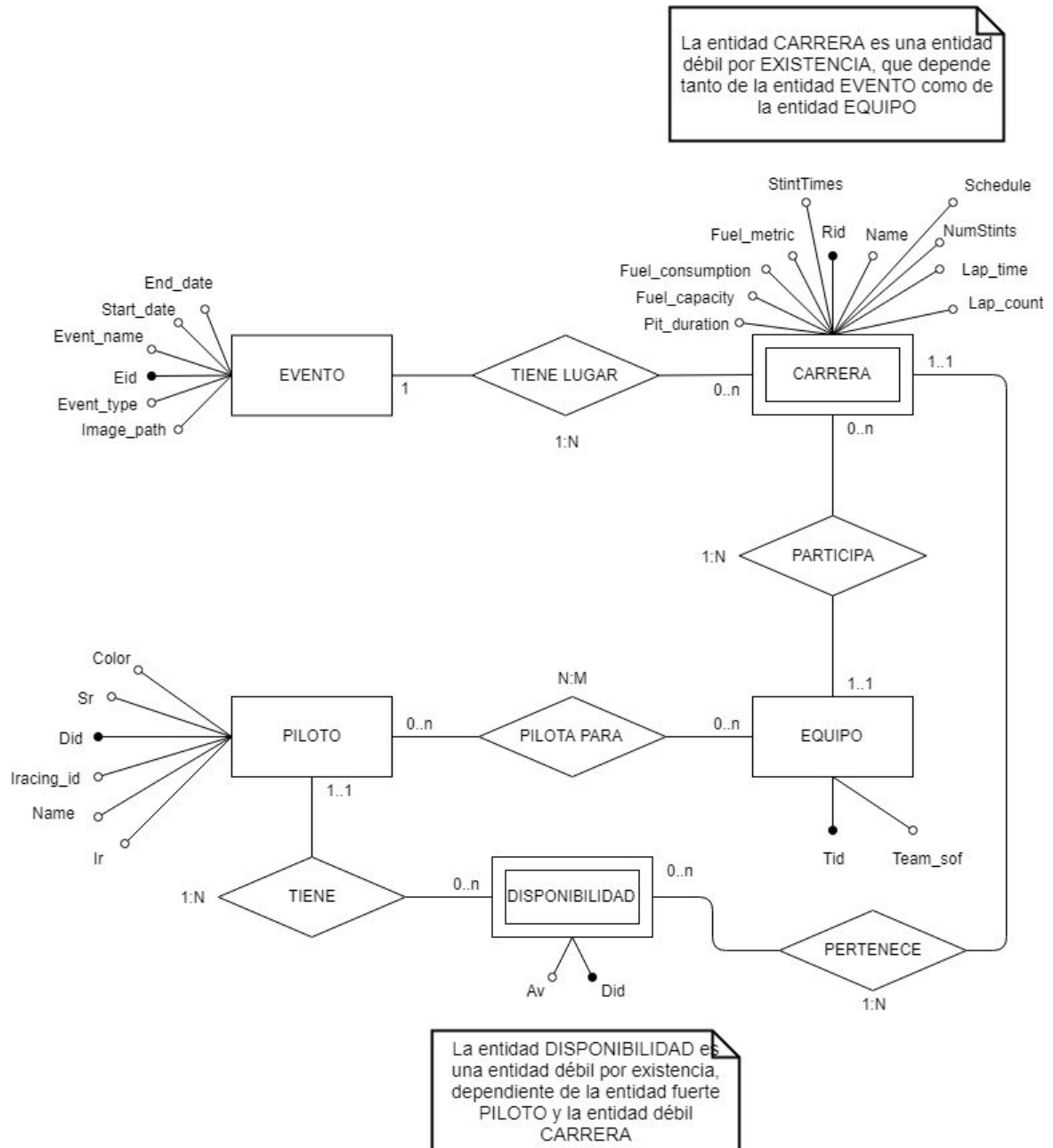


Figura 34: Diagrama Entidad-Relación.

4.6. Diagrama relacional base de datos

El diagrama relacional de la base de datos vincula entre sí las diferentes tablas que aparecen en la base de datos a través de la expansión de un campo en común, en este caso, la clave primaria de cada tabla.

Este diagrama relacional se puede extraer de el diagrama entidad-relación de una forma sencilla siguiendo una serie de normas:

- Las relaciones 1:N provocan una propagación de la clave primaria desde la tabla que contiene la relación 1..1 a la tabla que contiene la relación múltiple
- Las relaciones N:M provocan la creación de una nueva tabla que consiste de las dos claves primarias de las entidades que se relacionan, y de todos los atributos que forman esa relación si esta los contiene.
- Las relaciones 1:1 provocan una propagación de clave primaria, pero a diferencia de las relaciones 1:N, es indiferente desde qué tabla hacia qué tabla sea la propagación.

Siguiendo estas sencillas reglas, se genera el diagrama relacional que se muestra a continuación, formado por un total de 7 tablas, 6 pertenecientes a las entidades que forman la base de datos, y 1 (DrivesFor) perteneciente a la relación N:M PILOTA PARA.

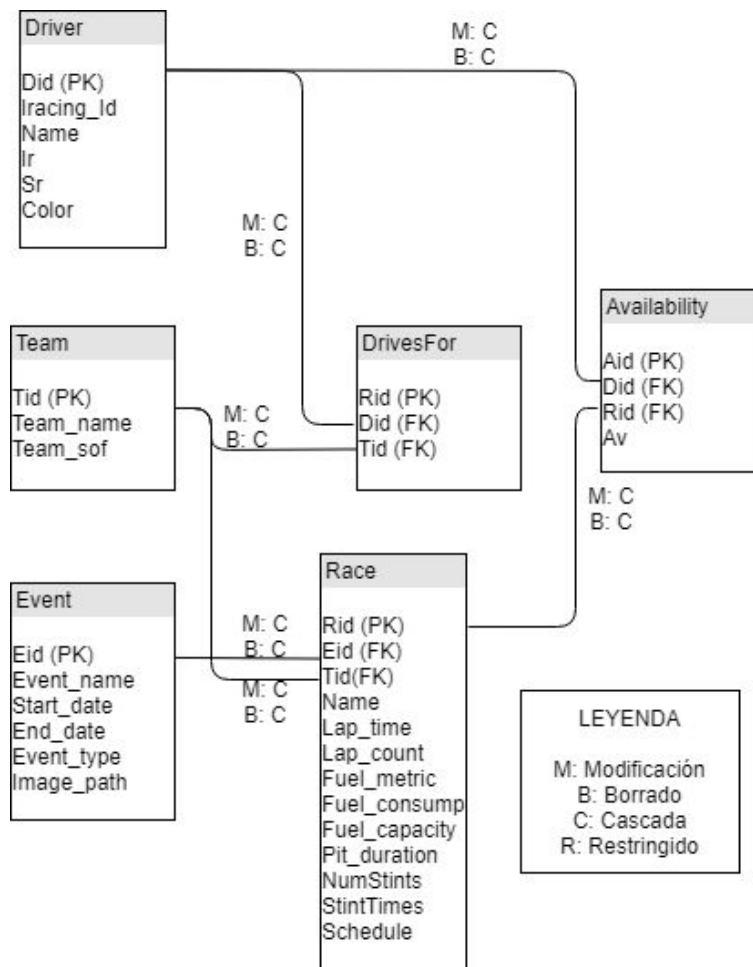


Figura 35. Diagrama relacional base de datos

4.7. Diccionario de datos

En el diccionario de datos a continuación se detallan todos los elementos que forman parte de los datos manejados en la aplicación. Este diccionario de datos será útil en el momento de necesitar el acceso a información sobre los datos manejados por la aplicación.

Event			
Nombre atributo	Tipo de atributo	Descripción	Comentarios
Eid	int	Id del evento	UNIQUE AUTOINCREMENT PRIMARY KEY
Event_name	string	Nombre del evento	
Start_date	DateTime	Fecha de inicio del evento	
End_date	DateTime	Fecha de fin del evento	
Event_type	string	Tipo de evento	
Image_path	string	URL relativa de la imagen del evento	

Tabla 70. Tabla especificación entidad Event

Driver			
Nombre atributo	Tipo de atributo	Descripción	Comentarios
Did	int	Id del piloto	UNIQUE AUTOINCREMENT PRIMARY KEY
Iracing_id	int	ID del piloto en iRacing	
Name	string	Nombre del piloto	
Ir	int	iRating del piloto en iRacing	
Sr	string	sRating del piloto en iRacing	
Color	string	Color identificativo del piloto	

Tabla 71. Tabla especificación entidad Driver

Team			
Nombre atributo	Tipo de atributo	Descripción	Comentarios
Tid	int	Id del equipo	UNIQUE AUTOINCREMENT PRIMARY KEY
Team_name	string	Nombre del equipo	
Team_sof	int	SOF del equipo	Para info sobre SOF mirar glosario de términos

Tabla 72. Tabla especificación entidad Team

Race			
Nombre atributo	Tipo de atributo	Descripción	Comentarios
Rid	int	Id de la carrera	UNIQUE AUTOINCREMENT PRIMARY KEY
Eid	int	Id del evento donde se disputa la carrera	FOREIGN KEY
Tid	int	Id del equipo que disputa la carrera	FOREIGN KEY
Name	string	Nombre de la carrera	
Lap_time	int	Tiempo por vuelta	
Lap_count	int	Número de vueltas	
Fuel_metric	string	Métrica de medida del combustible	Litros/Galones
Fuel_consumption	float	Consumo del combustible por vuelta	
Fuel_capacity	float	Capacidad del tanque de combustible	
Pit_duration	int	Duración de una parada en boxes	
Schedule	string	Horario de pilotaje	

Tabla 73. Tabla especificación entidad Race

Availability			
Nombre atributo	Tipo de atributo	Descripción	Comentarios
Aid	int	Id de la disponibilidad	UNIQUE AUTOINCREMENT PRIMARY KEY
Did	int	Id del piloto con disponibilidad	FOREIGN KEY
Rid	int	Id de la carrera de la disponibilidad	FOREIGN KEY
Av	string	Representación de la disponibilidad con números	0 → No disponible 1 → Disponible 2 → Tentativa

Tabla 74. Tabla especificación entidad Availability

DrivesFor			
Nombre atributo	Tipo de atributo	Descripción	Comentarios
Rid	int	Id de la relación piloto-equipo	UNIQUE AUTOINCREMENT PRIMARY KEY
Did	int	Id del piloto	FOREIGN KEY
Tid	int	Id del equipo	FOREIGN KEY

Tabla 75. Tabla especificación entidad DrivesFor

4.8. Diseño interfaz de usuario

En este apartado se muestran los bocetos realizados para la interfaz de la aplicación. Esta interfaz se ha realizado de forma que las diferentes tareas están compartimentalizadas para mejorar la experiencia de usuario. También se ha decidido realizar la aplicación en un tema oscuro (fondo oscuro y frente blanco) con algunos toques de color, con el objetivo de conseguir una cierta consistencia con las herramientas ya disponibles, con el objetivo de que el usuario sienta que se encuentra en un entorno familiar. La utilización de un tema oscuro en la aplicación tiene la ventaja añadida de que al ser aplicaciones comúnmente usadas en entornos con una iluminación baja o nula, un tema oscuro disminuye el cansancio de la vista del usuario.

Con el objetivo de una mejor visualización de las interfaces en la documentación impresa, se realizan los bocetos con un tema claro, pero con la misma distribución de controles que se encontrará en la aplicación final con tema oscuro.

DI-01 Pestaña "events"	
Definición	En esta pestaña se mostrará toda la información necesaria para la gestión de eventos
Activación	Al acceder a la aplicación / Al seleccionar la pestaña "Events"
Boceto	<p style="text-align: center;">Figura 36. Boceto pestaña Events</p>
Eventos	El usuario podrá crear un nuevo evento, editar uno ya existente, o eliminar un evento previamente introducido.

Tabla 76. Interfaz pestaña events

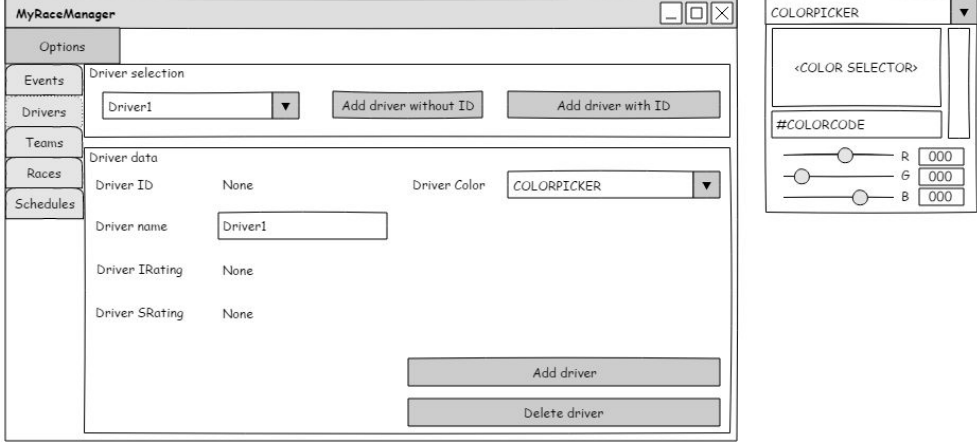
DI-02 Pestaña "drivers"	
Definición	En esta pestaña se mostrará toda la información necesaria para la gestión de pilotos
Activación	Al seleccionar la pestaña "Drivers"
Boceto	 <p style="text-align: center;">Figura 37. Boceto pestaña Drivers</p>
Eventos	El usuario podrá crear un nuevo piloto, tanto con ID, como sin ID, y eliminar un piloto.

Tabla 77. Interfaz pestaña drivers

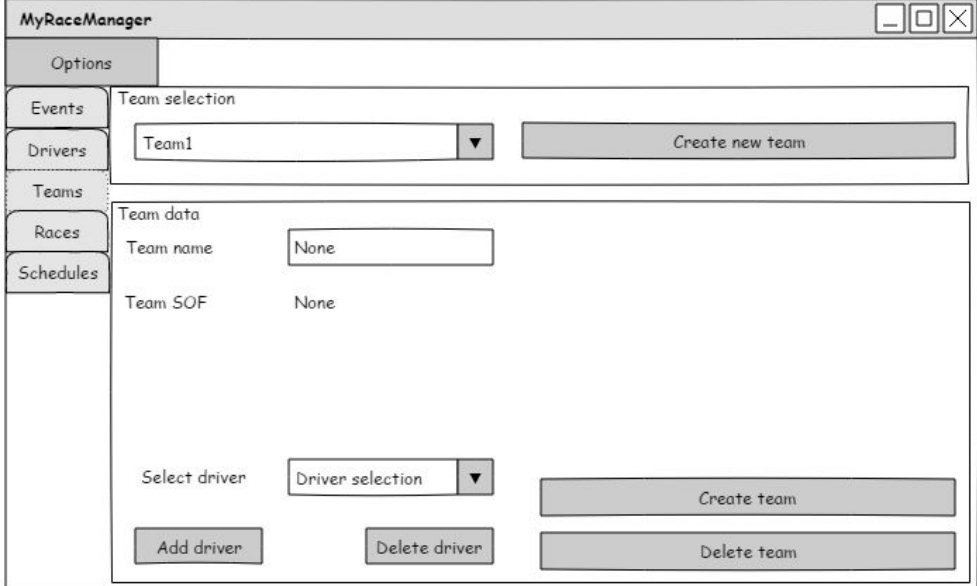
DI-03 Pestaña "teams"	
Definición	En esta pestaña se mostrará toda la información necesaria para la gestión de equipos
Activación	Al seleccionar la pestaña "Teams"
Boceto	 <p style="text-align: center;">Figura 38. Boceto pestaña Teams</p>
Eventos	El usuario podrá crear un nuevo equipo, editarlo, añadir pilotos, o eliminar un equipo anteriormente creado.

Tabla 78. Interfaz pestaña teams

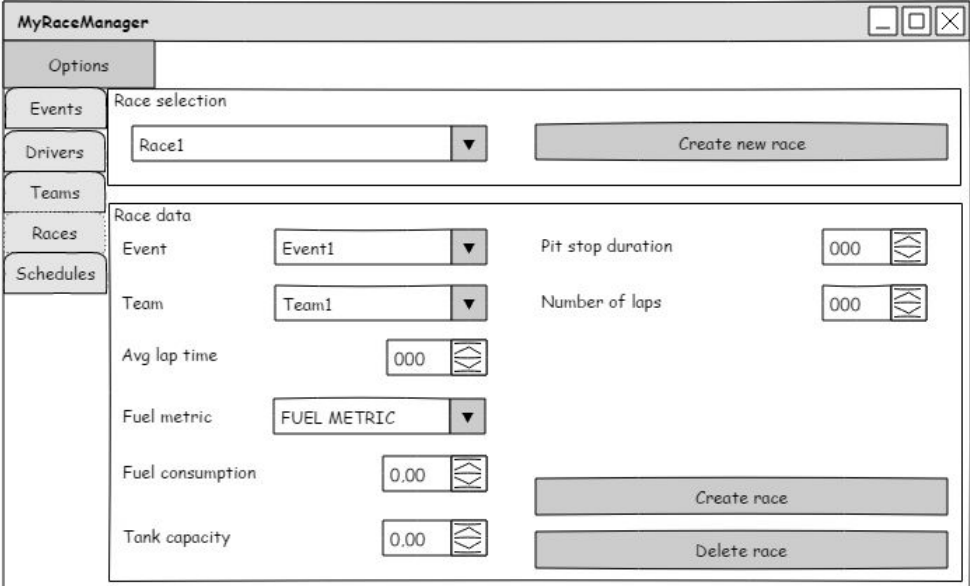
DI-04 Pestaña "Races"	
Definición	En esta pestaña se mostrará toda la información necesaria para la gestión de carreras
Activación	Al seleccionar la pestaña "Races"
Boceto	 <p style="text-align: center;">Figura 39. Boceto pestaña Races</p>
Eventos	El usuario podrá crear una carrera, eliminarla o editarla.

Tabla 79. Interfaz pestaña races

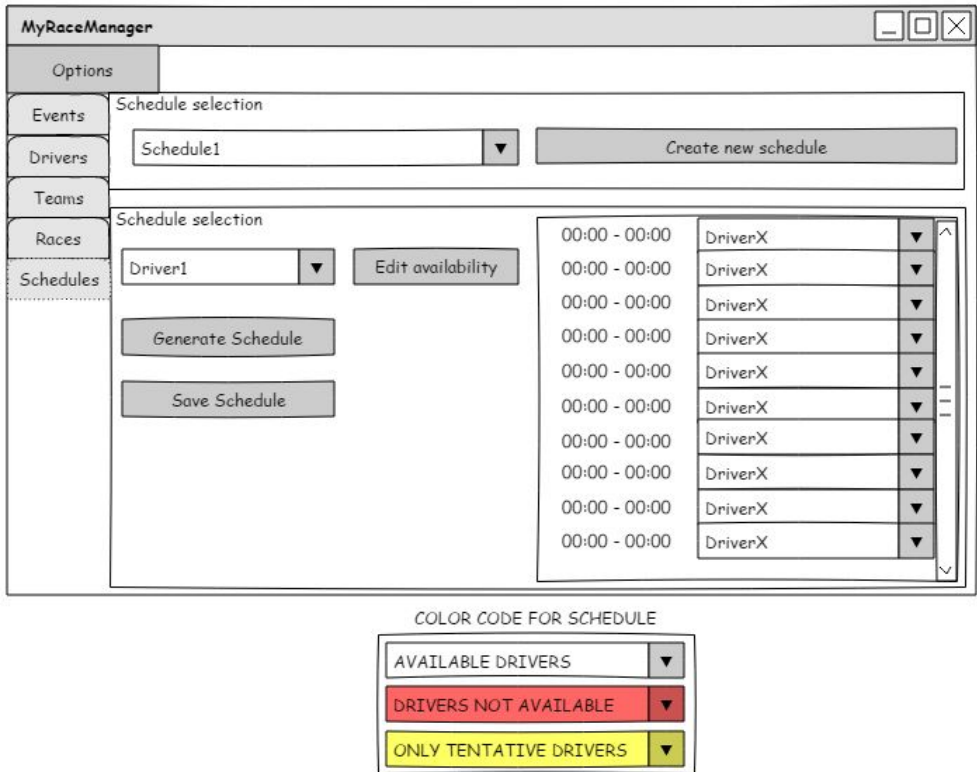
DI-05 Pestaña "Schedules"	
Definición	En esta pestaña se mostrará toda la información necesaria para la gestión de horarios de conducción
Activación	Al seleccionar la pestaña "Schedules"
Boceto	 <p style="text-align: center;">Figura 40. Boceto pestaña Schedules</p>
Eventos	El usuario podrá editar un horario de pilotaje, crearle automáticamente, y editar la disponibilidad de los pilotos del equipo para esa carrera.

Tabla 80. Interfaz pestaña schedules

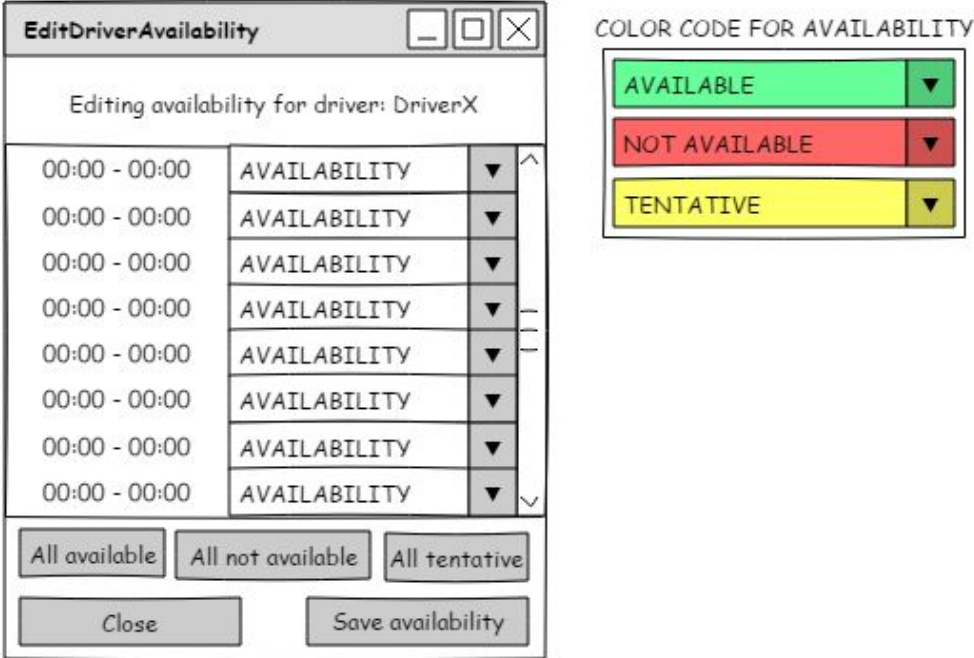
DI-06 Ventana "Edit driver availability"	
Definición	En esta ventana se mostrará toda la información necesaria para la edición de la disponibilidad de un piloto para una carrera
Activación	Al hacer clic en el botón "Edit driver availability"
Boceto	 <p>Figura 41. Boceto ventana edit availability</p>
Eventos	El usuario podrá editar la disponibilidad individualmente, y también podrá poner todas las disponibilidades en los tres modos disponibles en la aplicación (available, not available, tentative).

Tabla 81. Interfaz ventana edit driver availability

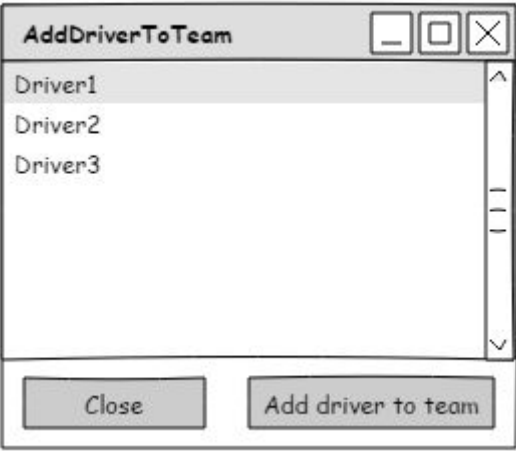
DI-07 Ventana "Add driver to team"	
Definición	En esta ventana se mostrará todos los controles necesarios para añadir pilotos a un equipo.
Activación	Al hacer clic en el botón "Add driver to team"
Boceto	 <p>Figura 42. Boceto ventana add driver to team</p>
Eventos	El usuario podrá añadir cualquier piloto que ya no esté dentro de un equipo, como parte de este.

Tabla 82. Interfaz ventana add driver to team

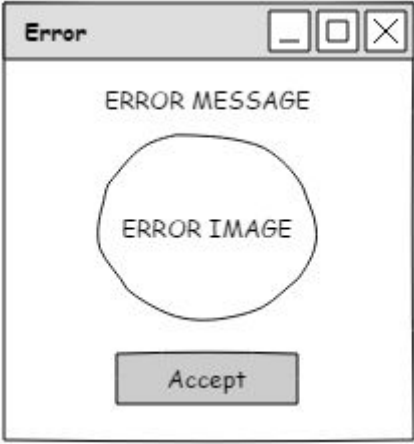
DI-08 Ventana "Error"	
Definición	Esta ventana mostrará un mensaje de error y una imagen de error
Activación	Al existir un error dentro del programa
Boceto	 <p>Figura 43. Boceto ventana error</p>
Eventos	El usuario podrá reconocer que conoce el error dando al botón "Accept"

Tabla 83. Interfaz ventana error

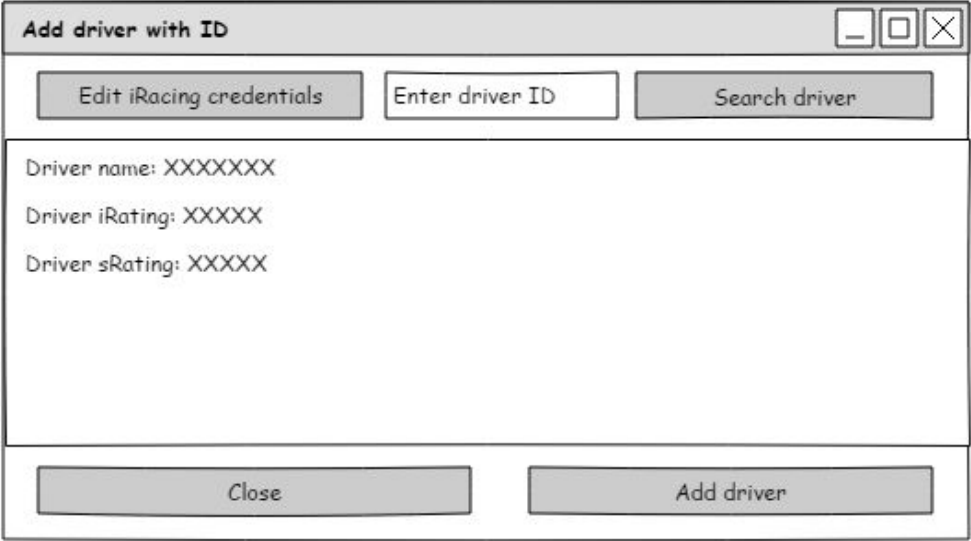
DI-09 Ventana "Add driver with ID"	
Definición	Esta ventana mostrará los controles para añadir un piloto mediante su ID de iRacing
Activación	Al hacer clic sobre el botón "Add driver with ID"
Boceto	 <p style="text-align: center;">Figura 44. Boceto ventana add driver with ID</p>
Eventos	El usuario podrá insertar sus credenciales de iRacing, así como introducir el ID de un piloto para buscar sus datos y añadir el piloto si se está conforme con los datos obtenidos.

Tabla 84. Interfaz ventana add driver with ID

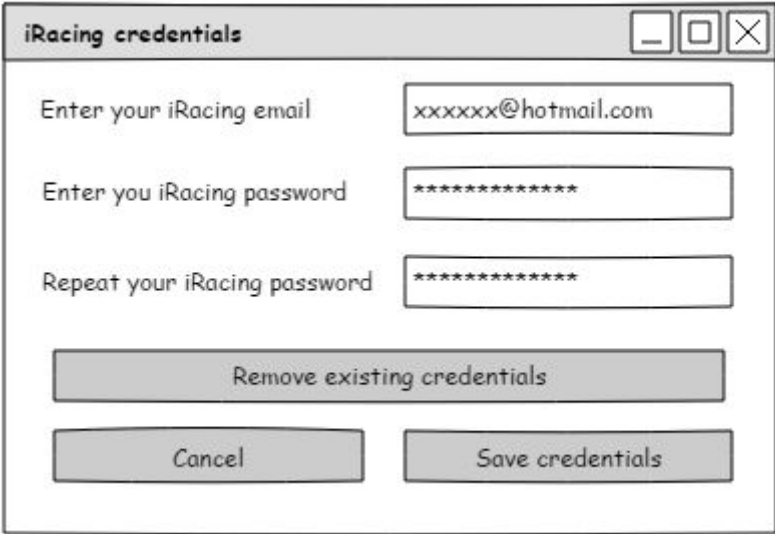
DI-10 Ventana “iRacing credentials”	
Definición	Esta ventana mostrará los controles para poder guardar las credenciales de inicio de sesión en iRacing del usuario para su posterior uso.
Activación	Al hacer clic sobre el botón “Edit iRacing credentials”
Boceto	 <p style="text-align: center;">Figura 45. Boceto ventana insert iRacing credentials</p>
Eventos	El usuario podrá insertar sus credenciales de inicio de sesión para su posterior uso, podrá eliminar sus credenciales.

Tabla 85. Interfaz ventana enter iRacing credentials

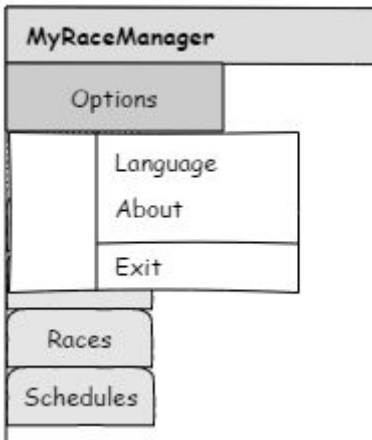
DI-11 Menu "Options"	
Definición	Esta pestaña permitirá al usuario acceder a diferentes funcionalidades de la aplicación como el cambio de idioma, una ventana de información, o permitirá otra forma de salir de la aplicación
Activación	Al hacer click sobre el ítem superior "Options"
Boceto	 <p>Figura 46. Boceto menu options</p>
Eventos	El usuario podrá seleccionar diferente opciones de la lista.

Tabla 86. Interfaz opciones

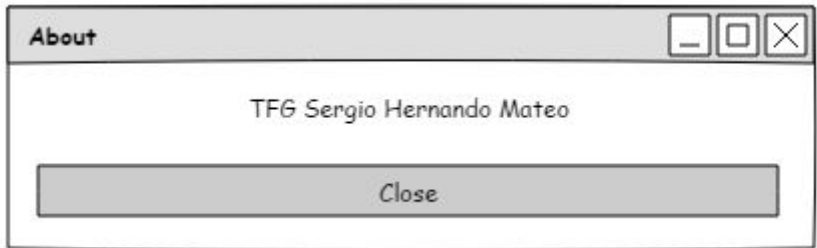
DI-12 Ventana "About"	
Definición	Esta ventana mostrará diferente información sobre la aplicación a los usuarios.
Activación	Al hacer click sobre el ítem "About" del menú "Options"
Boceto	 <p>Figura 47. Boceto ventana About</p>
Eventos	Ninguno

Tabla 87. Interfaz ventana "About"

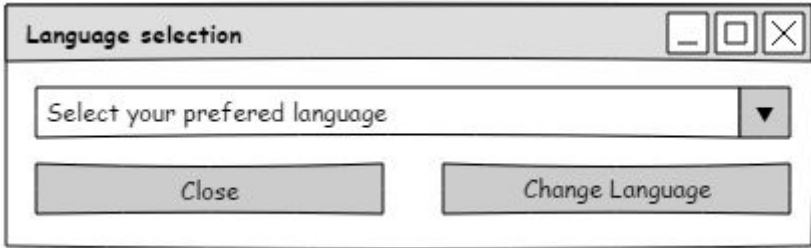
DI-13 Ventana "Change language"	
Definición	Esta ventana permitirá cambiar el idioma de la aplicación
Activación	Al hacer click sobre el ítem "Language" del menú "Options"
Boceto	 <p>Figura 48. Boceto ventana Language Selection</p>
Eventos	El usuario podrá cambiar el idioma de la aplicación seleccionando un idioma de la lista y haciendo click en el botón de "Change language"

Tabla 88. Interfaz ventana "Change language"

5. Capítulo 5: Implementación

En este apartado se discuten los aspectos relacionados con la implementación del programa que se consideran dignos de mención tanto por ser ciertamente interesantes, utilizar prácticas poco comunes, o simplemente para su mejor entendimiento.

5.1. Análisis página web iRacing (Scraping de datos)

Como pequeña introducción, el scraping de datos es una técnica de programación usada para la extracción de datos de páginas web que no pueden ser accedidos de otra forma (APIs/acceso a bases de datos). Esta técnica es principalmente utilizada para extraer datos de páginas tipo server render, es decir, aquellas en las que lo que se muestra en pantalla lo decide el servidor y no el usuario mediante llamadas a AJAX. Esta técnica puede ser ejecutada con infinidad de herramientas, pero en nuestro caso, necesitaremos una herramienta que no está específicamente diseñada para este propósito llamada Selenium, ya explicada anteriormente en el apartado de Entorno Tecnológico.. Esta necesidad viene por tener que interactuar con la página web objetivo para hacer login cada vez que se desee extraer datos. Los principales pasos de la técnica del Scraping son:

1. Identificación de la web objetivo
2. Identificación de los datos objetivo
3. Análisis de la estructura del árbol DOM de la web
4. Recorrido del árbol DOM con una herramienta
5. Extracción de los datos deseados
6. Almacenamiento de los datos
7. Presentación de los datos

A continuación, se muestra de la forma más clara posible el proceso que se ha seguido para analizar la página web de iRacing y extraer los datos de cada piloto. Para realizar el scraping con éxito, se deberá de interactuar con páginas completamente diferentes, la primera será la de login de iRacing, en la cual deberemos diferenciar tres elementos principales, el campo de usuario (class name = username), el campo de contraseña (class name = password), y el botón de submit del formulario (id = submit). Tras ello, deberemos introducir el usuario dentro del campo del usuario, la contraseña dentro del campo de contraseña, y tras ello, pulsar el botón de submit del formulario.

The image shows a screenshot of the iRacing.com login page. At the top, there is a red error message: "Invalid email address or password. Please try again." Below this, the "Sign In" section contains an "Email Address" field, a "Password" field, a checkbox for "Enable Auto Login for this Computer", and a "LOG IN" button. There are also links for "Recover Password" and "Sign Up". Below the "Sign In" section is a "Not Racing Yet?" section with a "SIGN UP" button. The bottom of the page shows copyright information for 2019 iRacing.com.

```
<!doctype html PUBLIC "-//W3C//DTD HTML 4.01//EN" "http://www.w3.org/TR/html4/strict.dtd">
<html>
  <head>...</head>
  <body onload="showFeedback(); setInterval('checkField()',50);">
    <div id="wrapper">
      <div id="feedback">...</div>
      <div id="content-sign-in" class="clearfix">
        <div class="inner clearfix">
          <div id="service">...</div>
          <script>...</script>
          <script src="/membersite/js/iracing_script.js?ver=201906241323013"></script>
          <style type="text/css">...</style>
          <div class="clearfix node">
            <h1>Sign In</h1>
            <form method="post" name="LOGIN" action="/membersite/Login">
              <input class="username" tabindex="1" required="required" name="username" type="email">
              <input class="password" tabindex="2" required="required" name="password" type="password">
              <div class="clearfix" style="width: 330px;">
                <div class="clearfix login-w">
                  <div class="login-right clearfix">
                    <input id="submit" class="log-in" tabindex="4" type="submit" value="Log in" onclick="submitLoginForm()
                    "> == $0
                    ::after
                  </div>
                <div class="login-left clearfix">...</div>
                ::after
              </div>
            </div>
            <!-- <p><strong>Purchased iRacing with Steam?</strong> <a href="javascript:void(0);"
            onclick="javascript:SteamLogin(this);">Log in here &raquo;</a></p>
            -->
          <p>...</p>
          ::after
        </div>
        <input class="hidden" type="hidden" name="utcoffset" value="-120">
        <input class="hidden" type="hidden" name="todaysdate" value>
      </form>
      ::after
    </div>
    <div class="clearfix node last">...</div>
    <script type="text/javascript">...</script>
    ::after
  </body>
</html>
```

Figura 49. Análisis página login iRacing.com

Tras la pagina de login nos encontramos que la página “membersite” tiene un sistema de clases más complejo, y que los elementos que nosotros buscamos solo tienen la etiqueta style, con lo cual, no podemos identificarlos mediante id o class name como antes, lo que nos obliga a identificar el elemento accesible más cercano y desde ahí trabajar para llegar a ellos.

Encontramos que elemento accesible más cercano es la imagen de licencia (class name = licenseImage) dentro del elemento roadTabContent (id = roadTabContent) (Este elemento roadTabContent es importante porque siempre escogemos el iRating y Safety rating que el piloto tenga en la categoría Road). Tras ello, tenemos que ir bajando en la jerarquía usando el método CssSelector eligiendo los hijos correctos que nos permitan acceder a los datos que necesitamos. En nuestro caso, la jerarquía de elementos tiene este aspecto:

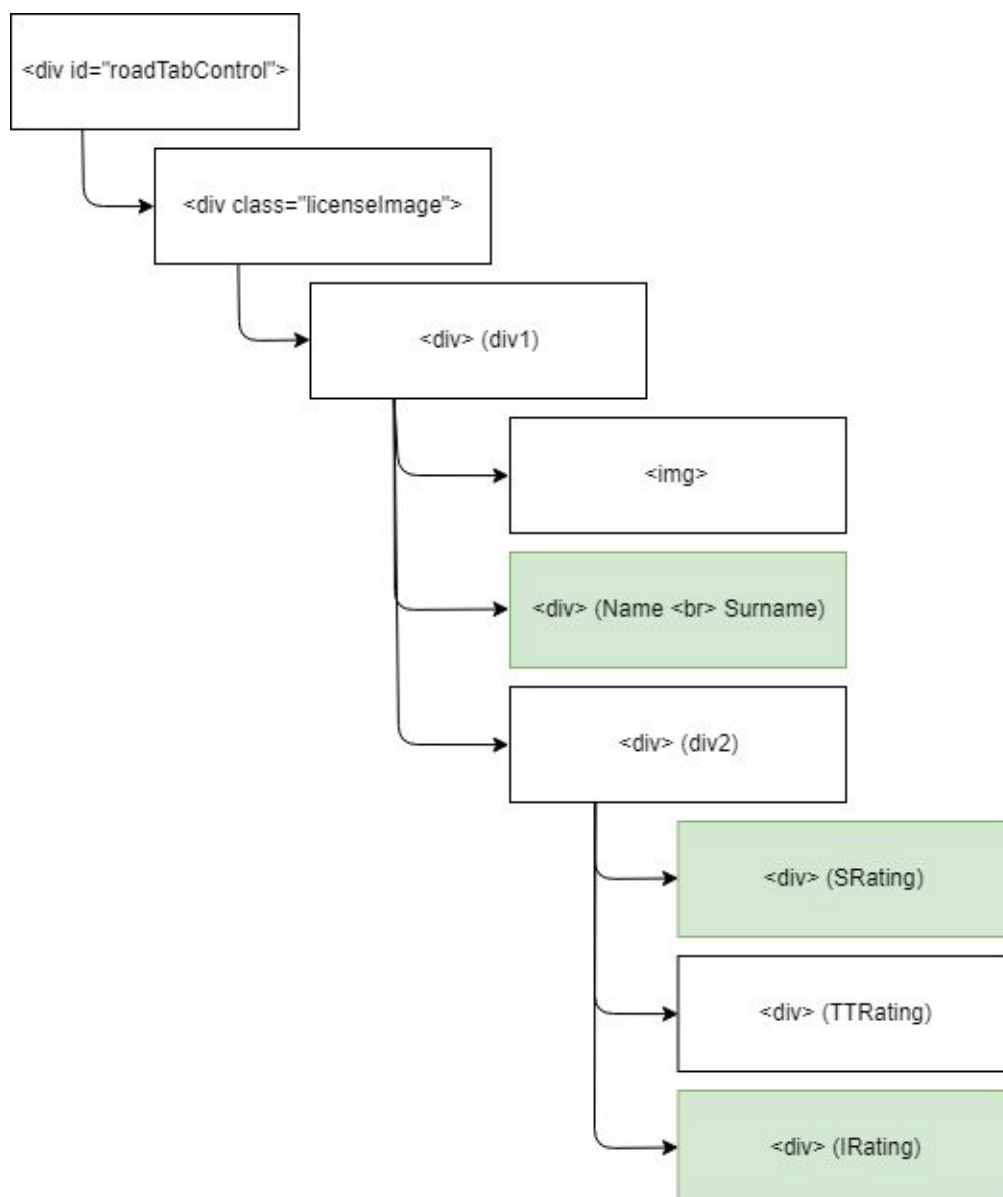


Figura 50. Jerarquía de elementos página membersite iRacing.com

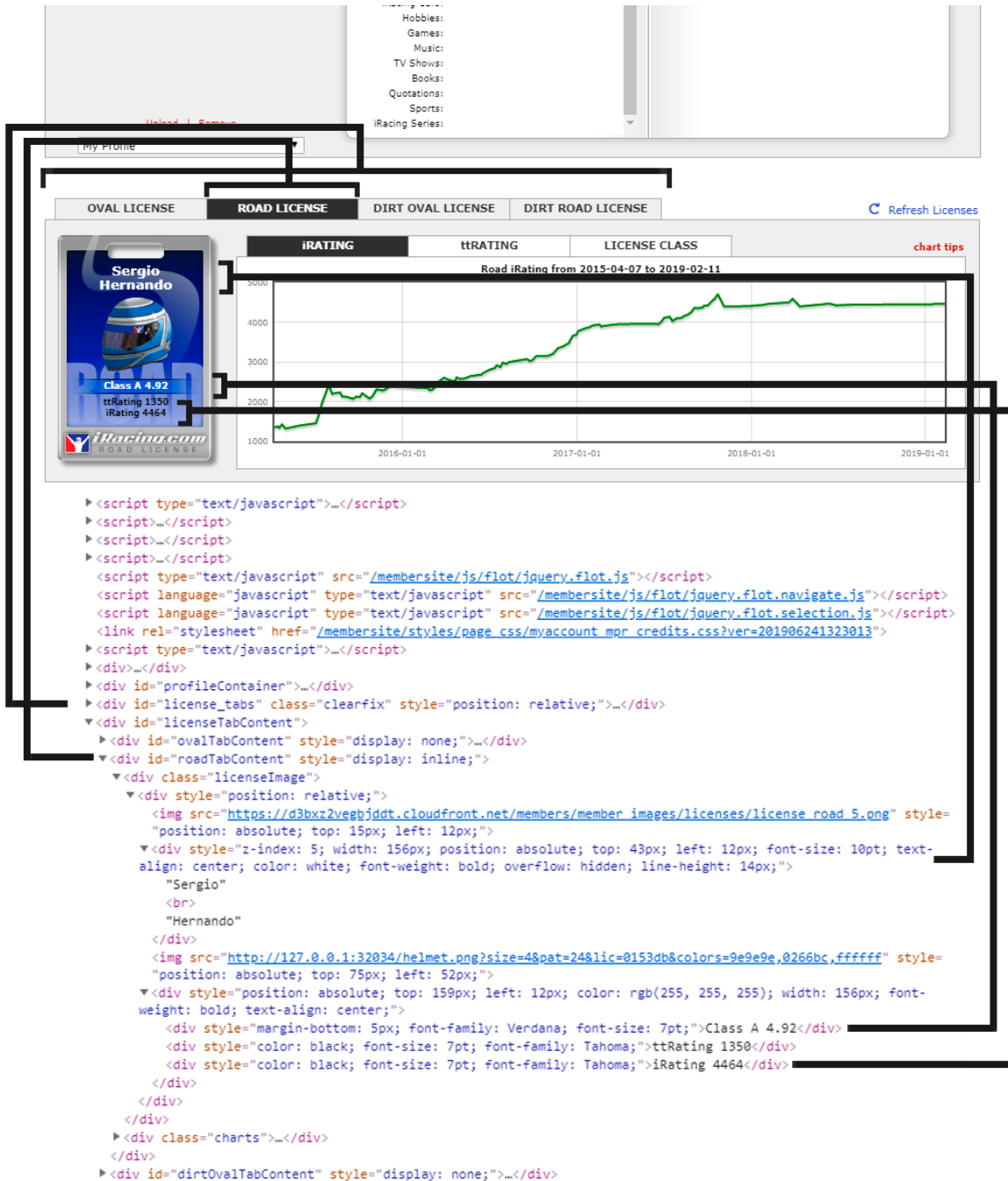


Figura 51. Análisis página membersite iRacing.com

Tras conseguir aislar los elementos extraídos de la página, los devolvemos en forma de lista a la ventana principal encargada de añadir al usuario con ID. En esta ventana, el usuario podrá escoger si quiere, o no, añadir al piloto a su programa.

5.2. Generación de horarios

En este apartado se discute el algoritmo de creación automática de horarios y las mejoras que se le podrán aplicar en un futuro.

Para este algoritmo se ha creado una clase completamente nueva llamada `ScheduleGenerator.cs`. Esta clase se caracteriza por no disponer de interfaz gráfica y por recibir como argumento el nombre de la carrera para la cual queremos generar el horario.

Este algoritmo accede en primer lugar a la carrera, de la cual se quiere generar el horario, y tras ello, se guardan también sus disponibilidades y se calculan el número de stints que se van a realizar en la carrera. A continuación, el algoritmo de creación, recorre todas las disponibilidades buscando la primera que esté disponible para añadir el id del piloto al horario. Si no encuentra ningún piloto que esté disponible, empieza a buscar un piloto que esté en tentativa de correr, y finalmente, si no encuentra ninguno en tentativa, no añade ningún id de piloto y sigue buscando.

Finalmente, cuando se han hecho el mismo número de bucles que de stints, si la longitud del horario generado es menor que el número de stints, entonces esto significa que el algoritmo no ha encontrado piloto disponible, o en tentativa, en alguno de los stints por rellenar, con lo cual se mostrará un mensaje de error para que el usuario revise los horarios de disponibilidad de los pilotos para tener por lo menos un piloto disponible o tentative en cada uno de los stints.

En caso de encontrar un horario completo, se devuelve a la pantalla principal para su importación a la interfaz de usuario.

Este algoritmo no es el más óptimo para la tarea, debido a que puede dejar a pilotos que están disponibles fuera de la carrera, y no es altamente personalizable, pero esta decisión se ha tomado para mantener el algoritmo fácil de explicar y entender.

```
class ScheduleGenerator
{
    Race currentRace;
    List<Availability> availabilities = new
List<Availability>();
    int numStints = 0;
    string res = "";

    public ScheduleGenerator(string raceName)
    {
        currentRace =
((MainWindow)Application.Current.MainWindow).getC().getRaceByName
(raceName);
    }
}
```

```
        foreach (Availability a in
((MainWindow)Application.Current.MainWindow).getC().getAvailabili
ties())
    {
        if (a.Rid == currentRace.Rid)
        {
            availabilities.Add(a);
        }
    }

    numStints =
currentRace.getStintNumber(((MainWindow)Application.Current.MainW
indow).getC().getEventByEid(currentRace.Eid).Event_type);

    for (int i = 0; i<numStints; i++)
    {
        // Search for available drivers
        foreach (Availability a in availabilities)
        {
            if (a.Av[i].Equals('1'))
            {
                res += a.Did + ",";
                break;
            }
        }

        // If available drivers not found, search for
tentative drivers
        if (res.Length == i*2)
        {
            foreach (Availability a in availabilities)
            {
                if (a.Av[i].Equals('2'))
                {
                    res += a.Did + ",";
                    break;
                }
            }
        }

        if (res.Length == i*2)
```

```
        {
            break;
        }
    }

    if (res.Length/2 < numStints)
    {
        List<string> err = new List<string>();
        err.Add(Properties.Resources.ScheduleNotCreated);
        ErrorWindow ew = new ErrorWindow(err);
        ew.ShowDialog();
        err.Clear();
    }
    else
    {
        ((MainWindow)Application.Current.MainWindow).tempSchedule = res;
    }
}
```

Figura 52. Algoritmo de creación de horarios.

5.3. Elección de color de fuente

A continuación se discute el algoritmo de elección del color de fuente, en este caso para los combo box que tienen fondos de diferentes colores. Este algoritmo elige entre un color de fuente blanco o negro dependiendo del color de fondo que se esté usando.

Para poder facilitar la reusabilidad de este algoritmo, creamos un nuevo método público, el cual llamaremos cada vez que queramos general el color de fuente respecto a otro. Este método devuelve un objeto Brush, el cual es del tipo usado en la asignación de colores tanto de fondo como de fuente para los controles del programa. El método toma como parámetro el piloto del cual se quiere acceder al color de fondo.

El primer paso es dividir el color del piloto almacenado con el formato 000000000 siendo cada 3 números una variable del color (RGB).

Tras ello, averiguamos la intensidad de cada color y las sumamos, y si el resultado es mayor a 186, usamos negro, y sino, blanco.

El valor 186 es un valor que se puede variar al gusto, pero tras varias pruebas se ha decidido mantener este valor por ser el que mejores resultados ha dado.

Este algoritmo es generalmente aceptado por la comunidad, pero existen casos en los que no cumple con las recomendaciones del W3C. Como en ningún momento se ha decidido que el programa deba seguir las guías del W3C en la selección de colores, no se ha usado la fórmula que si cumple con las guías del W3C con el objetivo de mantener la fórmula sencilla y fácil de comprender.

```
public Brush createForegroundBrush(Driver d)
{
    int r =
Convert.ToInt32(d.Color.ToString().Substring(0, 3));
    int g =
Convert.ToInt32(d.Color.ToString().Substring(3, 3));
    int b =
Convert.ToInt32(d.Color.ToString().Substring(6, 3));

    if(((r*0.299) + (g*0.587) + (b * 0.114)) > 186)
    {
        return Brushes.Black;
    }
    else
    {
        return Brushes.White;
    }
}
```

Figura 53. Algoritmo de elección entre colores de fuente.

6. Capítulo 6: Pruebas

Las pruebas son una parte fundamental del desarrollo de un proyecto software, ya que determinan el correcto funcionamiento de la aplicación. Deben ser realizadas por un tester y han de abarcar todos los aspectos de la aplicación. Éstas pruebas pueden ser de caja negra y/o de caja blanca.

6.1. Caja negra

Las pruebas de caja negra son utilizadas para comprobar las funcionalidades del código sin tener en cuenta los procesos que se llevan a cabo entre medias. En las pruebas de caja negra se indica las entradas de datos que se llevan a cabo y la salida de datos o resultado esperado.

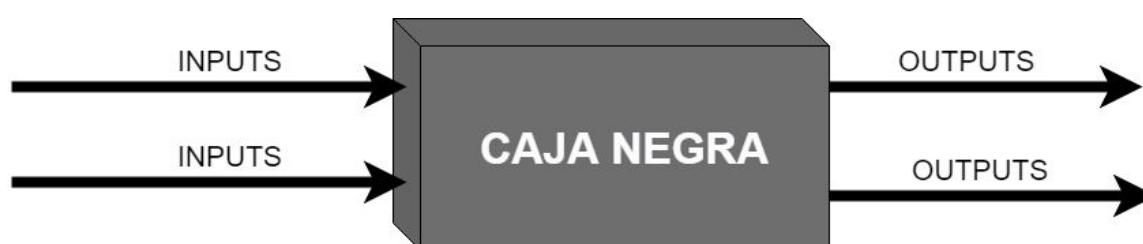


Figura 54. Pruebas de caja negra.

En las tablas mostradas en este apartado se muestran las pruebas de caja negra realizadas.

PCN-01 Creación de evento	
Descripción	Crear un evento
Precondiciones	
Entradas	Start date: 1/1/2020 10:00 Finish date: 1/1/2020 16:00 Road event with time ✓ Event name: New Event Image: No
Salidas	Start date: 1/1/2020 10:00 Finish date: 1/1/2020 16:00 Road event with time ✓ Event name: New Event Image: No
Resultado satisfactorio	Sí

Prueba	<div><p>Event Selection</p><p>New Event from 01/01/2020 10:00:38 to 01/01/2020 16:00:46. Event type: RT <input type="button" value="Create new event"/></p><p>Event Data</p><p>Start date: miércoles, 1 de enero de 2020 1 <input type="button" value="Event name"/> New Event</p><p>Finish date: miércoles, 1 de enero de 2020 1 <input type="button" value="Image of the event"/> Upload image</p><p><input type="radio"/> Road event with laps <input type="radio"/> Oval event with laps</p><p><input checked="" type="radio"/> Road event with time</p><p><input type="button" value="Edit selected event"/></p><p><input type="button" value="Delete selected event"/></p></div>
--------	--

Figura 55. Prueba creación de evento

Tabla 89. PCN-01: Prueba caja negra creación de evento

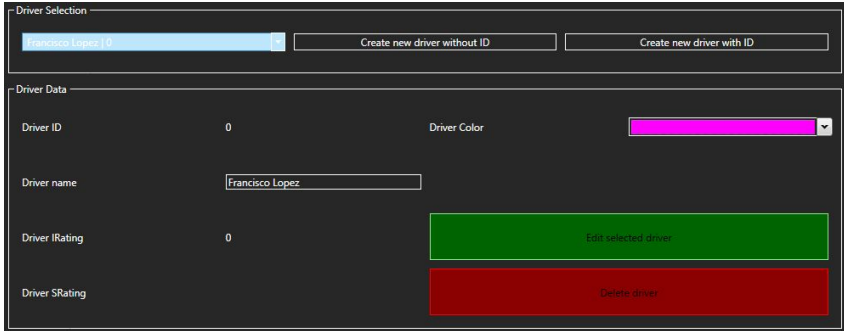
PCN-02 Creación de piloto sin ID	
Descripción	Crear un piloto sin ID de iRacing
Precondiciones	
Entradas	Driver name: Francisco Lopez Color: #FF00FF (Rosa)
Salidas	Driver name: Francisco Lopez Color: #FF00FF (Rosa)
Resultado satisfactorio	Sí
Prueba	 <p>Figura 56. Prueba creación de piloto sin ID</p>

Tabla 90. PCN-02: Prueba caja negra creación de piloto sin ID

PCN-03 Creación de piloto con ID	
Descripción	Crear un piloto con ID de iRacing
Precondiciones	El piloto ha de existir en el sistema de iRacing
Entradas	Driver ID: 190926 Color: #00FF00 (Verde)
Salidas	Driver ID: 190926 Driver name: Sergio Hernando Driver iRating: 4464 Driver SRating: Class A 4.92
Resultado satisfactorio	Sí
Prueba	 <p>Figura 57. Prueba creación de piloto con ID</p>

Tabla 91. PCN-03:Prueba caja negra creación de piloto con ID

PCN-04 Creación de equipo	
Descripción	Crear un equipo
Precondiciones	
Entradas	Nombre: Equipo Team
Salidas	Nombre: Equipo Team Team SOF: 0
Resultado satisfactorio	Sí
Prueba	 <p>Figura 58. Prueba creación de equipo</p>

Tabla 92. PCN-04::Prueba caja negra creación de equipo

PCN-05 Añadir piloto a equipo	
Descripción	Añadir un piloto al equipo
Precondiciones	Se ha de crear primero un equipo y guardarlo en la base de datos Ha de existir algún piloto dentro de la base de datos
Entradas	Add driver to team: Sergio Hernando 190926
Salidas	Nombre: Equipo Team Team SOF: 4464 Driver List: - Sergio Hernando 190926
Resultado satisfactorio	Sí
Prueba	 <p>Figura 59. Prueba adición piloto a equipo</p>

Tabla 93. PCN-05: Prueba caja negra adición de piloto a equipo

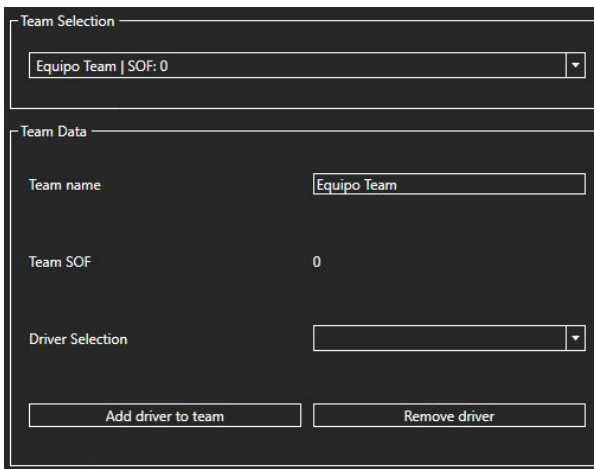
PCN-06 Eliminar piloto de equipo	
Descripción	Eliminar un piloto de un equipo
Precondiciones	Se ha de crear primero un equipo y guardarlo en la base de datos con algún piloto que pertenezca a ese equipo
Entradas	Remove driver from team: Sergio Hernando 190926
Salidas	Nombre: Equipo Team Team SOF: 0 Driver List: -
Resultado satisfactorio	Sí
Prueba	 <p>Figura 60. Prueba eliminación piloto de equipo</p>

Tabla 94. PCN-06: Prueba caja negra eliminación piloto de equipo

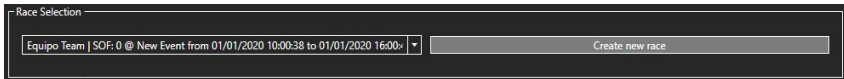
PCN-07 Crear carrera	
Descripción	Crear una carrera que relacione un equipo y un evento
Precondiciones	Ha de existir algún evento en el sistema Ha de existir algún equipo en el sistema
Entradas	Event: New Event Team: Equipo Team Average lap time (seconds): 100 Fuel metric: Litres Average Fuel Consumption: 2,7 Tank fuel capacity: 100 Full pit stop duration: 60
Salidas	Event: New Event Team: Equipo Team Average lap time (seconds): 100 Fuel metric: Litres Average Fuel Consumption: 2,7 Tank fuel capacity: 100 Full pit stop duration: 60
Resultado satisfactorio	Sí
Prueba	 <p>Figura 61. Prueba crear carrera</p>

Tabla 95. PCN-07: Prueba caja negra creación de carrera

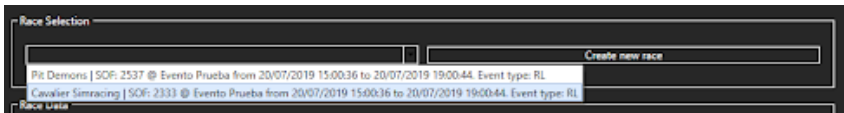
PCN-08 Eliminar carrera	
Descripción	Eliminar una carrera
Precondiciones	Debe existir por lo menos una carrera en el sistema
Entradas	Selected race: Equipo Team SOF 0 @ New Event..... Delete selected race
Salidas	Deleted race: Equipo Team SOF 0 @ New Event.....
Resultado satisfactorio	Sí
Prueba	 <p>Figura 62. Prueba eliminar carrera</p>

Tabla 96. PCN-08: Prueba caja negra eliminar carrera

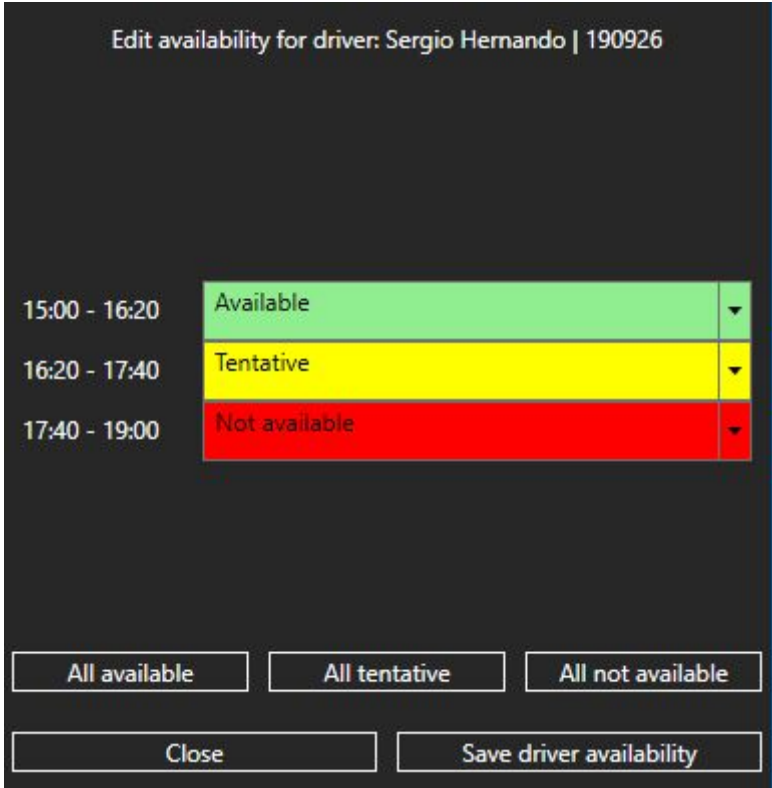
PCN-09 Editar disponibilidad	
Descripción	Editar una disponibilidad de un piloto en una carrera
Precondiciones	Debe existir por lo menos una carrera en el sistema
Entradas	Driver: Sergio Hernando 190926 Availability: <ul style="list-style-type: none"> - Available - Tentative - Not Available
Salidas	Availability: <ul style="list-style-type: none"> - Available - Tentative - Not Available
Resultado satisfactorio	Sí
Prueba	 <p>Figura 63. Prueba editar disponibilidad</p>

Tabla 97. PCN-09: Prueba caja negra editar disponibilidad

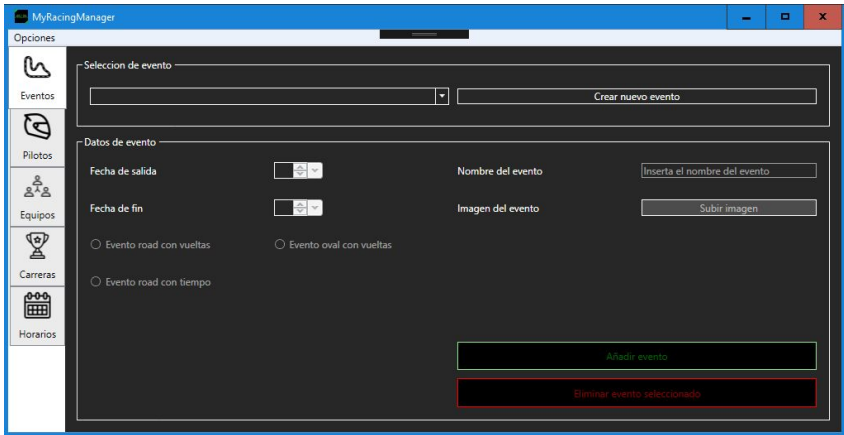
PCN-10 Cambiar idioma	
Descripción	Cambiar el idioma de la aplicación
Precondiciones	
Entradas	Change language
Salidas	Language changed
Resultado satisfactorio	Sí
Prueba	 <p>Figura 64. Prueba cambiar idioma</p>

Tabla 98. PCN-10: Prueba caja negra cambiar idioma

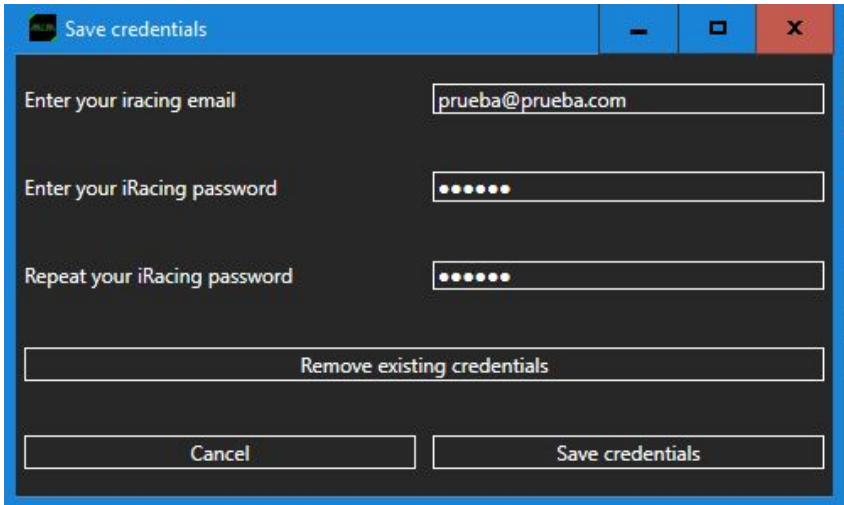
PCN-11 Guardar credenciales	
Descripción	Guardar credenciales
Precondiciones	
Entradas	Credenciales de prueba: <ul style="list-style-type: none"> - Email: prueba@prueba.com - Contraseña: prueba - Repetir contraseña: prueba
Salidas	Credenciales de prueba: <ul style="list-style-type: none"> - Email: prueba@prueba.com - Contraseña: cifrado y oculto - Repetir contraseña: cifrado y oculto
Resultado satisfactorio	Sí
Prueba	 <p>Figura 65. Prueba guardar credenciales</p>

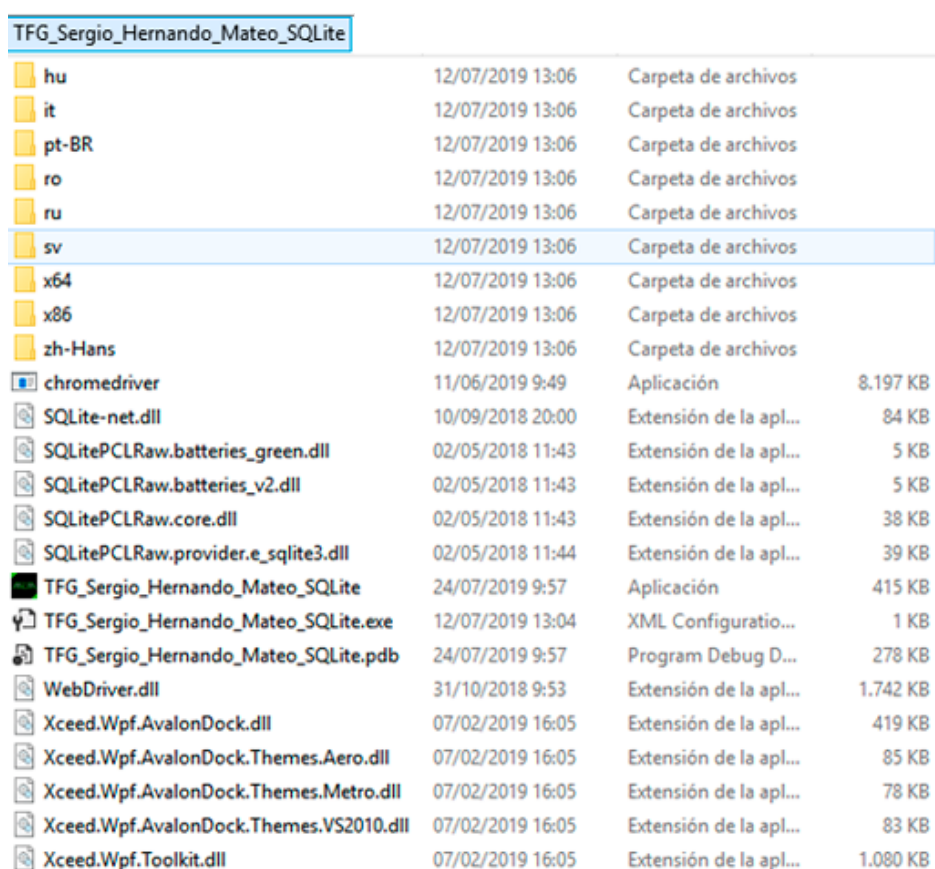
Tabla 99. PCN-11: Prueba caja negra guardar credenciales

7. Capítulo 7: Manuales de usuario (instalación)

Debido a que la aplicación desarrollada es exclusiva para Windows, ésta deberá ser ejecutada en un entorno Windows, preferiblemente con todas las actualizaciones del SO instaladas.

Tras asegurarnos de que se cumplen los requisitos, lo único que deberemos hacer será extraer la carpeta “TFG_Sergio_Hernando_Mateo”, en la cual encontraremos ejecutable llamado “TFG_Sergio_Hernando_Mateo_SQLite.exe” junto con otros archivos como el logo, los archivos .dll necesarios, etc.

Se ruega NO ELIMINAR ninguno de los archivos que se entregan, debido a que esto supondría un fallo completo de la aplicación.



Nombre	Fecha y hora	Tipo	Tamaño
hu	12/07/2019 13:06	Carpeta de archivos	
it	12/07/2019 13:06	Carpeta de archivos	
pt-BR	12/07/2019 13:06	Carpeta de archivos	
ro	12/07/2019 13:06	Carpeta de archivos	
ru	12/07/2019 13:06	Carpeta de archivos	
sv	12/07/2019 13:06	Carpeta de archivos	
x64	12/07/2019 13:06	Carpeta de archivos	
x86	12/07/2019 13:06	Carpeta de archivos	
zh-Hans	12/07/2019 13:06	Carpeta de archivos	
chromedriver	11/06/2019 9:49	Aplicación	8.197 KB
SQLite-net.dll	10/09/2018 20:00	Extensión de la apl...	84 KB
SQLitePCLRaw.batteries_green.dll	02/05/2018 11:43	Extensión de la apl...	5 KB
SQLitePCLRaw.batteries_v2.dll	02/05/2018 11:43	Extensión de la apl...	5 KB
SQLitePCLRaw.core.dll	02/05/2018 11:43	Extensión de la apl...	38 KB
SQLitePCLRaw.provider.e_sqlite3.dll	02/05/2018 11:44	Extensión de la apl...	39 KB
TFG_Sergio_Hernando_Mateo_SQLite	24/07/2019 9:57	Aplicación	415 KB
TFG_Sergio_Hernando_Mateo_SQLite.exe	12/07/2019 13:04	XML Configuratio...	1 KB
TFG_Sergio_Hernando_Mateo_SQLite.pdb	24/07/2019 9:57	Program Debug D...	278 KB
WebDriver.dll	31/10/2018 9:53	Extensión de la apl...	1.742 KB
Xceed.Wpf.AvalonDock.dll	07/02/2019 16:05	Extensión de la apl...	419 KB
Xceed.Wpf.AvalonDock.Themes.Aero.dll	07/02/2019 16:05	Extensión de la apl...	85 KB
Xceed.Wpf.AvalonDock.Themes.Metro.dll	07/02/2019 16:05	Extensión de la apl...	78 KB
Xceed.Wpf.AvalonDock.Themes.VS2010.dll	07/02/2019 16:05	Extensión de la apl...	83 KB
Xceed.Wpf.Toolkit.dll	07/02/2019 16:05	Extensión de la apl...	1.080 KB

Figura 66. Estructura de la carpeta “TFG_Sergio_Hernando_Mateo_SQLite”

Tras la primera ejecución del programa, se creará un archivo de Base de Datos SQLite en la carpeta Documentos que esté definida por defecto. Generalmente situada en “C:\Users\username\Documents”.

También se creará una carpeta llamada MRMImages en la misma ruta, en la cual se almacenarán las imágenes que seleccionemos para la representación de los eventos.

Para el funcionamiento de las funcionalidades de extracción de datos sobre pilotos se necesita una cuenta válida de iRacing, la cual se puede crear en la página web <https://www.iracing.com/membership/>.

8. Capítulo 8: Manuales de usuario

En este apartado se presenta un manual de usuario con una explicación clara del funcionamiento de la aplicación desde el punto de vista de un usuario.

En primer lugar, nos encontramos con la interfaz general a todo el programa, la cual es el menú superior, desde el cual podremos acceder al cambio de idioma, a la ventana de información sobre el TFG, y salir del programa. En el lateral izquierdo encontramos las pestañas que nos permiten desplazarnos por todos los apartados del programa.

En la pantalla de cambio de idioma encontraremos una lista con los idiomas en los que está disponible el programa, y un botón para salir y otro para aplicar el idioma seleccionado. Para cambiar el idioma simplemente seleccionar uno de la lista y presionar el botón de cambiar idioma.

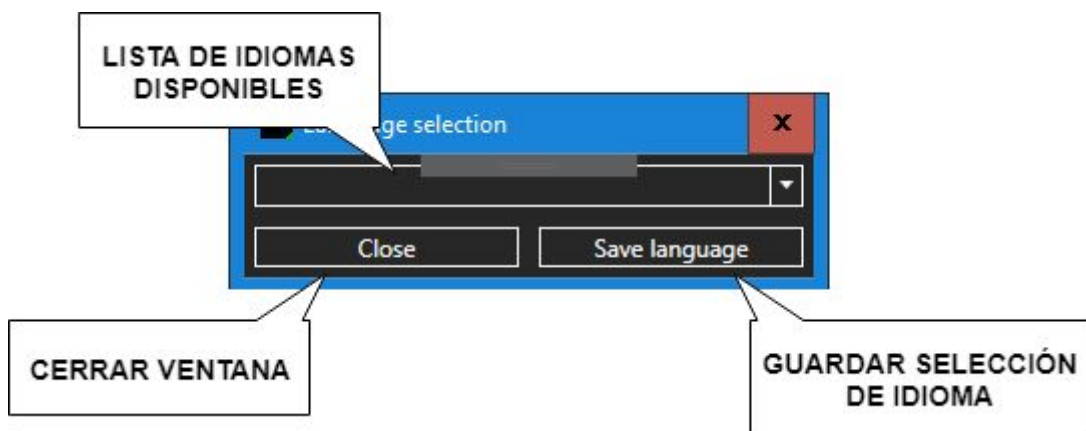


Figura 67. Manual de cambio de idioma.

En la ventana de información sobre el TFG nos encontraremos toda la información relacionada con el TFG, y el botón para cerrar la ventana. Si hacemos clic en el link de dicha pantalla, nos abre una pestaña del navegador predefinido en la página www.flaticon.com, de la cual se han extraído todos los iconos usados en la aplicación.

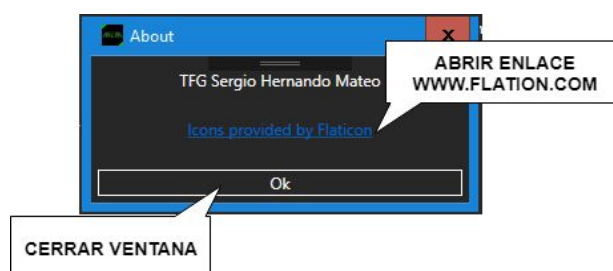


Figura 68. Manual de información sobre el TFG

Todas las pestañas del programa están divididas en dos secciones, la sección superior, en la que nos encontramos el selector de lista para los elementos, y diferentes opciones del programa, y la sección inferior, dirigida a la introducción/edición de datos y creación/borrado de datos.

En la pestaña eventos nos encontramos la lista de selección de eventos, y el botón para crear un evento desde 0.

- Para crear un evento, clicar el botón de crear nuevo evento, introducir los datos pedidos, y clicar el botón verde para crear un nuevo evento.
- Para editar un evento ya creado, seleccionar el evento en la lista superior, editar el dato que se desee, y clicar el botón verde.
- Para eliminar un evento ya creado, seleccionar el evento deseado en la lista superior, y clicar el botón rojo de eliminar evento. **ATENCIÓN:** Cuando se elimine un evento, se eliminarán todas las carreras creadas en ese evento.

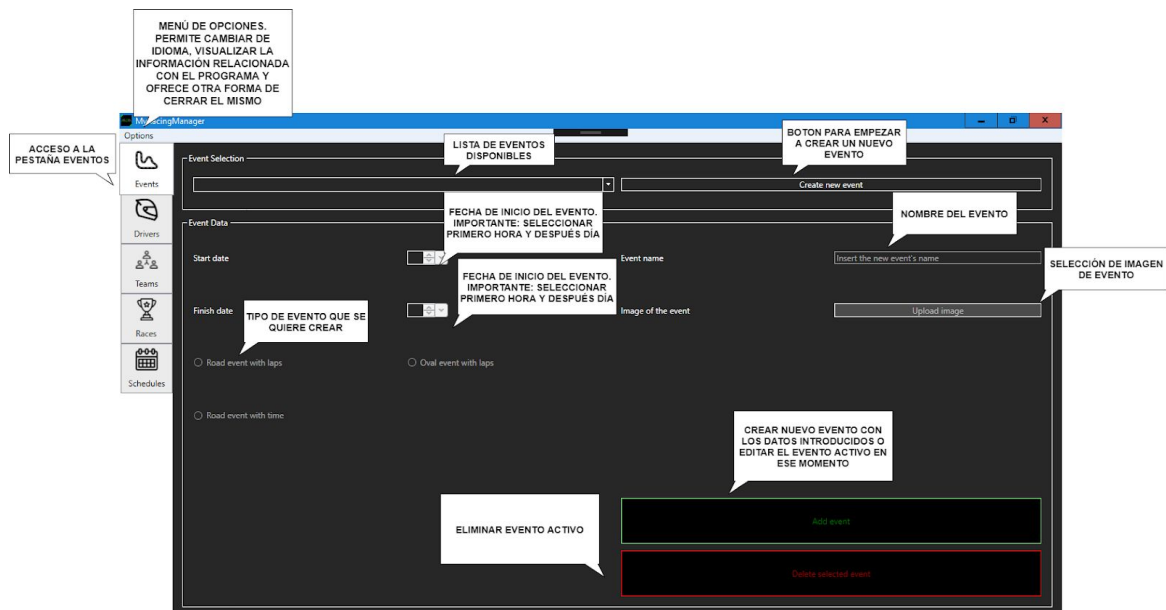


Figura 69. Manual de gestión de eventos

En la pestaña pilotos nos encontramos la lista de selección de pilotos, el botón para crear un piloto sin ID de iRacing, y el botón para crear un piloto con ID de iRacing.

- Para crear un piloto sin ID, clicar el botón de crear piloto sin ID, introducir el nombre del piloto, y clicar el botón verde de crear nuevo piloto.
- Para crear un piloto con ID de iRacing, clicar el botón de crear piloto con ID, seguir los pasos que se indican en este manual de usuario, y tras ello, se insertará automáticamente los datos recuperados del piloto. Clicar el botón verde de añadir piloto para añadir el piloto con ID al sistema.

- Para editar un piloto, tanto con ID como sin ID, seleccionar el piloto deseado de la lista superior, y editar su nombre en la sección inferior. Clicar el botón verde de edición de piloto cuando se quieran guardar los datos.
- Para eliminar un piloto, seleccionar el piloto a eliminar en la lista superior, y clicar el botón rojo de eliminar piloto. **ATENCIÓN:** Cuando se elimine un piloto, se eliminarán todos los horarios en los que participe ese piloto.

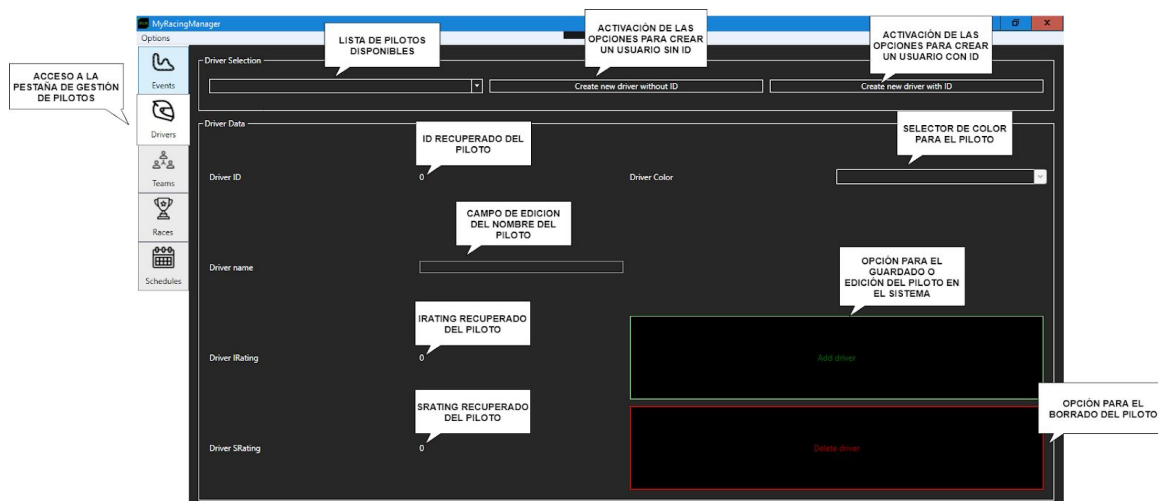


Figura 70. Manual de gestión de pilotos

En la ventana de creación de piloto con ID nos encontraremos tres secciones. la sección superior permitirá insertar las credenciales de iRacing del usuario, introducir el ID del piloto a añadir, y buscar el piloto con el ID introducido. La sección media, servirá para mostrar los datos extraídos sobre el piloto, y la sección inferior para aceptar los datos extraídos o cancelar el proceso.

- Para introducir las credenciales de iRacing clicar el botón de introducir credenciales y seguir las instrucciones especificadas en este manual.
- Para añadir un piloto por ID, introducir el ID deseado en el campo indicado, y clicar en el botón de buscar piloto, esperar un tiempo, y cuando los datos del piloto aparezcan en la sección correspondiente, clicar el botón de añadir piloto.

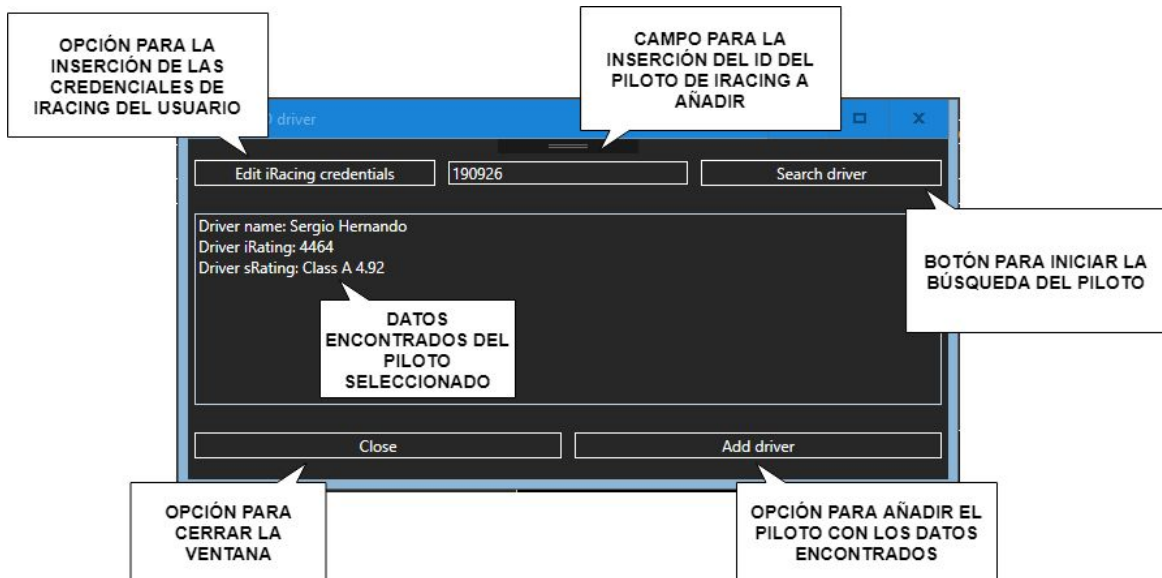


Figura 71. Manual de búsqueda de piloto con ID

En la ventana de inserción de credenciales de usuario encontraremos los campos necesarios para la inserción de las credenciales y tres botones que nos permitirán eliminar las credenciales guardadas en el sistema, cerrar la ventana, y guardar las credenciales introducidas.

- Para añadir credenciales al sistema, introducir los campos requeridos y clicar en el botón de guardar credenciales. **ATENCIÓN:** Las credenciales deberán ser válidas para funcionar, el sistema no comprueba la validez de las credenciales hasta que se usan para buscar un piloto. **ATENCIÓN:** El correo deberá tener el formato *****@****.***
- Para eliminar las credenciales clicar el botón de eliminar credenciales y cerrar la ventana.

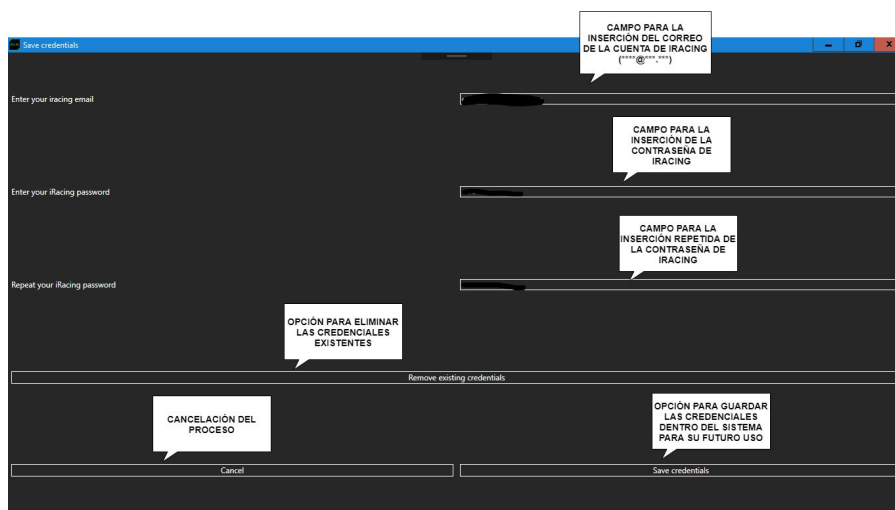


Figura 72. Manual de inserción de credenciales

En la pestaña de gestión de equipos encontraremos la lista de equipos disponibles y el botón para empezar a crear un equipo en la sección superior y todas las opciones para crear y editar un equipo en la sección inferior.

- Para crear un equipo clicar el botón de crear nuevo equipo, escribir el nombre del equipo y clicar el botón verde de crear nuevo equipo.
- Para añadir pilotos al equipo, tras crear un equipo, seleccionarlo en la lista y clicar el botón añadir piloto a equipo, tras ello seguir las instrucciones que se indican en este manual de uso. Tras ello, clicar el botón verde de editar equipo.
- Para eliminar pilotos de un equipo, seleccionar un equipo en la lista de pilotos del equipo y seleccionar la opción de eliminar piloto de equipo. Tras ello clicar el botón verde de editar equipo.
- Para editar un equipo, seleccionar el equipo en la lista superior, cambiar el nombre al que se desee, y clicar el botón verde de editar equipo.
- Para eliminar un equipo, seleccionar un equipo de la lista superior y clicar el botón rojo de eliminar equipo. **ATENCIÓN:** Al eliminar un equipo se eliminarán todas las carreras y horarios en las que el equipo participa.

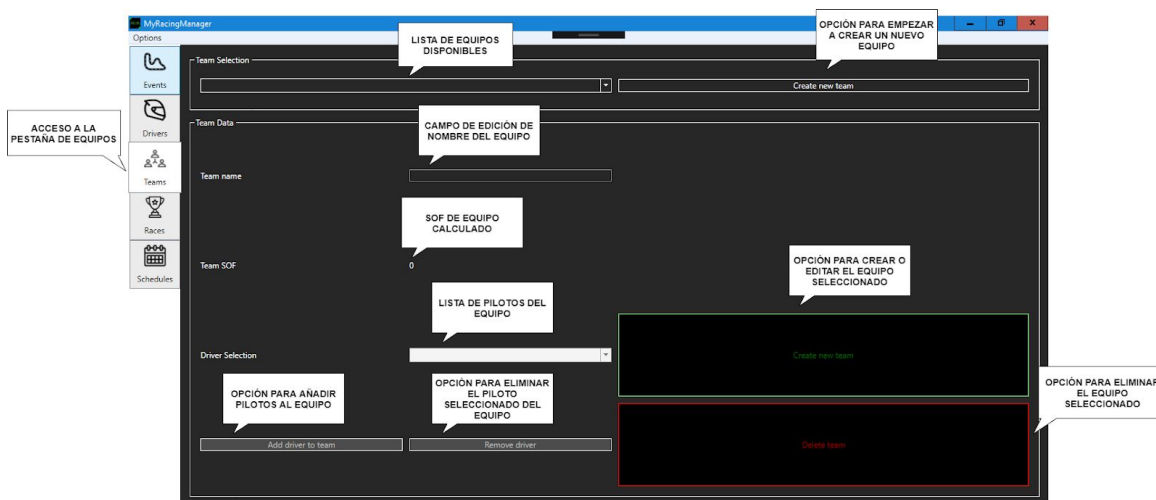


Figura 73. Manual de gestión de equipos

En la ventana de añadir piloto a equipo, encontramos una lista de los pilotos que no forman parte del equipo, y en la parte inferior la opción para salir y añadir el piloto seleccionado al equipo.

- Para añadir un piloto al equipo, seleccionar el piloto deseado y seleccionar el botón de añadir piloto a equipo. Cuando se haya acabado de añadir pilotos, cerrar la ventana.

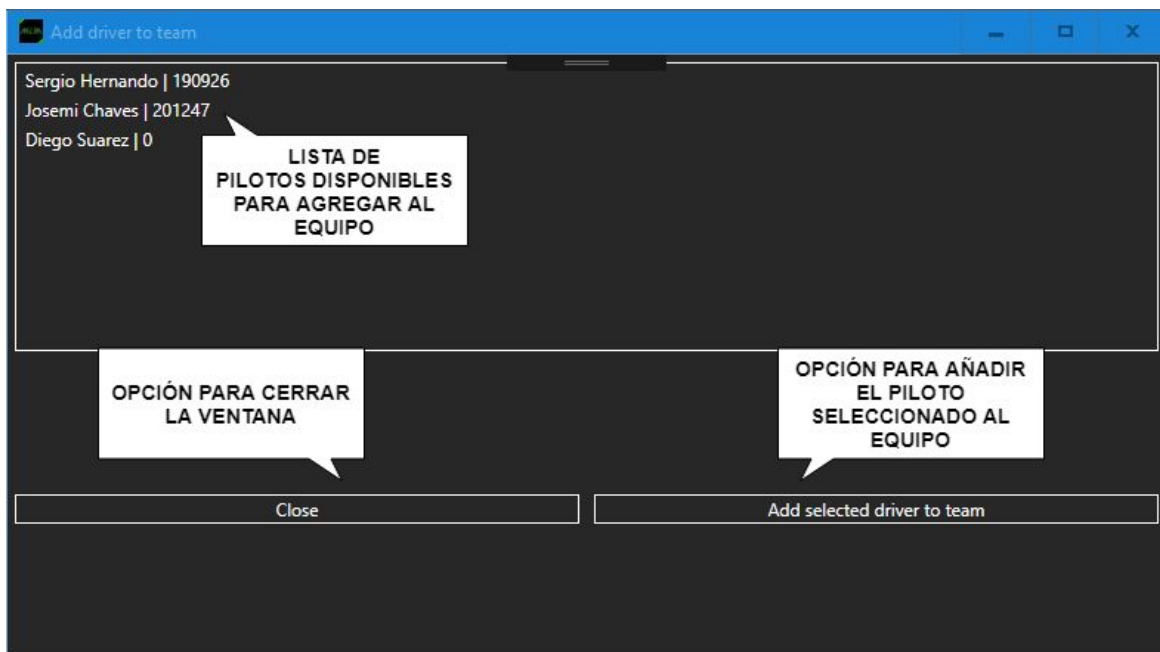


Figura 74. Manual de añadir piloto a equipo

En la pestaña carreras, encontramos la lista de carreras y el botón para empezar a crear una carrera en la parte superior, y los campos para todos los datos de una carrera en la sección inferior.

- Para crear una carrera, clicar el botón de crear nueva carrera, rellenar todos los campos necesarios para la carrera, y clicar el botón verde para añadir la carrera al sistema. **ATENCIÓN:** Una vez creada una carrera no se podrá editar.
- Para eliminar una carrera, seleccionar la carrera deseada en la lista superior y clicar el botón rojo de eliminar carrera para eliminarla. **ATENCIÓN:** Una vez eliminada la carrera se eliminarán también el horario relacionado con ella.

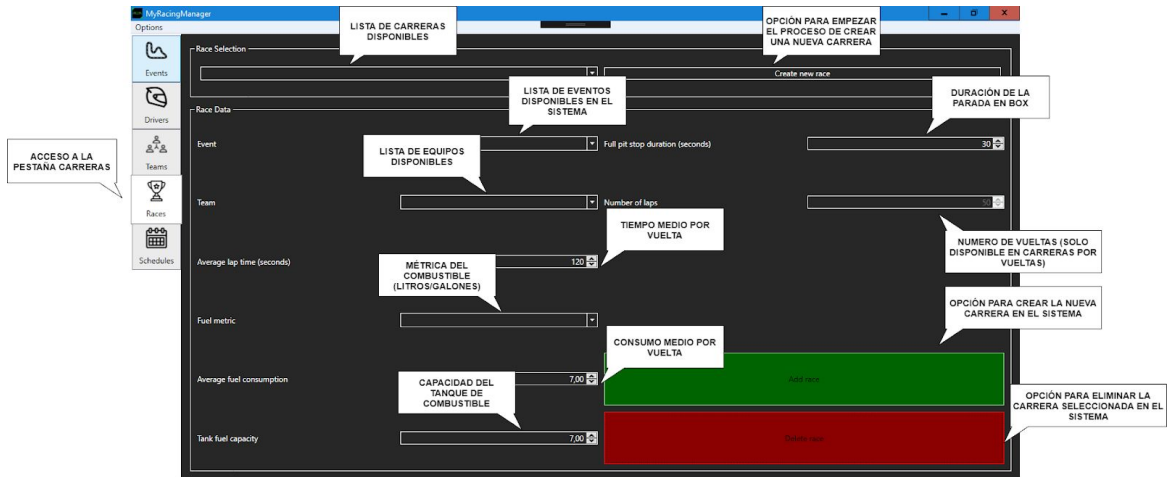


Figura 75. Manual de gestión de carreras

En la pestaña de gestión de horarios, encontramos la lista de horarios disponibles, y el botón para editar el horario seleccionado en la lista, en la sección inferior encontramos todas las herramientas necesarias para la creación y edición de horarios.

- Para editar un horario, se ha de seleccionar el horario a editar en la lista y clicar el botón de editar horario. Tras ello, y tras editar la disponibilidad de los pilotos, se podrán elegir en el horario de la derecha. Tras editar el horario, seleccionar la opción de guardar horario.
- Para generar un horario automáticamente, se ha de seleccionar el horario a editar en la lista y clicar el botón de editar horario. Tras ello, y tras editar la disponibilidad de los pilotos, elegir la opción de generar horario, y después, clicar el botón de guardar horario. **ATENCIÓN:** el horario sólo podrá ser creado si existen pilotos disponibles o en tentativa para TODOS los stints.

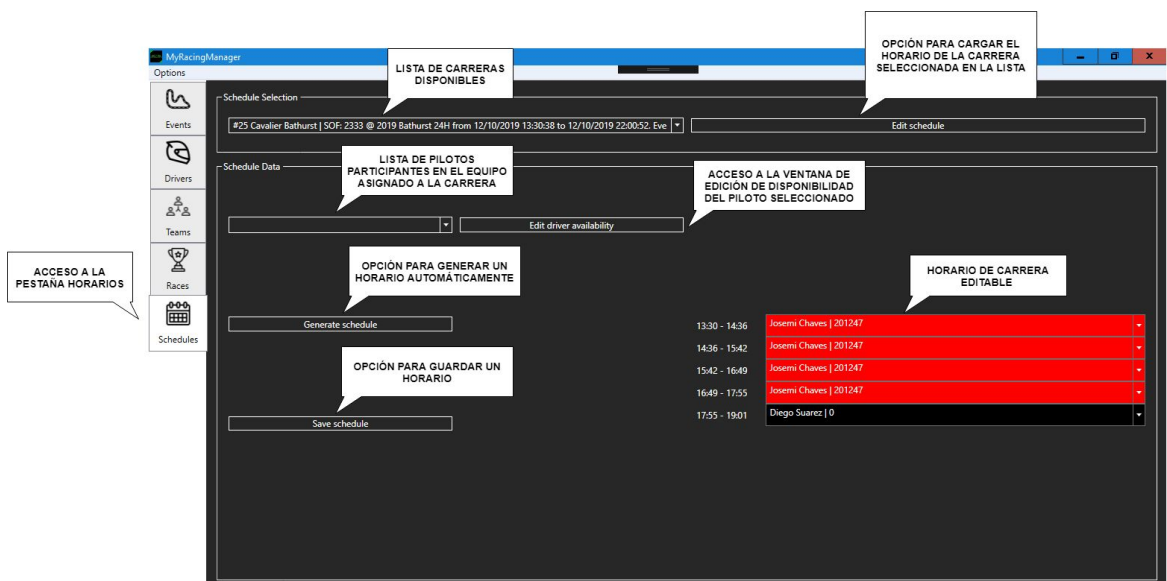


Figura 76. Manual de gestión de horarios

En la ventana de edición de disponibilidad del piloto, se distinguen dos secciones principales, la sección de elección de disponibilidad y las opciones para guardar la disponibilidad, poner todos los stints como no disponible / tentativa / disponible, y para cerrar la ventana.

- Para editar individualmente la disponibilidad de un piloto en cada uno de los stints, seleccionar cada stint por separado y elegir una disponibilidad para cada uno de ellos. Clicar el botón de guardar disponibilidad para guardar la disponibilidad del piloto.
- Para editar todos los stints como no disponible / tentativa / disponible, clicar el botón correspondiente y clicar el botón de guardar disponibilidad.



Figura 77. Manual de edición de disponibilidad

9. Capítulo 9: Conclusiones y futuras mejoras

9.1. Conclusiones

Tras el trabajo realizado se ha conseguido una aplicación que consigue los objetivos que se propusieron de una forma eficaz. La aplicación ha sido construida con una interfaz que simplifica los procesos lo máximo posible y hace el desarrollo de la actividad de uso una experiencia amigable y sencilla.

Desde un punto de vista personal, esta aplicación me ha ayudado a mejorar las capacidades y habilidades adquiridas durante toda la carrera. También me ha ayudado a relacionar todos los conceptos de la carrera entre sí, y ver cómo unos se complementan con otros.

Esta aplicación ha sido la primera que he desarrollado con un usuario final en mente y no con un profesor que la iba a corregir, y eso me ha hecho darme cuenta de lo importante que es la experiencia de usuario en una aplicación de este estilo, y me ha demostrado la capacidad de adaptación a los cambios que proporcionan las metodologías ágiles junto a una buena planificación inicial.

9.2. Mejoras a corto plazo

- Actualización de los datos de los pilotos con ID
- Seguimiento del estado de la carrera en el simulador de iRacing
- Especialización para otros simuladores

9.3. Mejoras a largo plazo

- Personalización de los parámetros de creación de horarios automáticamente (Nº máximo de stints seguidos, mº mínimo/máximo de stints por corredor).
- Uso de inteligencia artificial para la búsqueda del horario de pilotaje óptimo.
- Compartición y manejo online del programa por varias personas de un equipo

10. Bibliografía y Webgrafía

10.1. Bibliografía

- TURNER, R. “C#: The Ultimate Beginner's Guide to Learn C# Programming Step by Step”, 2019. Editorial: Independently published.
- B. WHITE, M. “Mastering C# (C Sharp Programming): A Step by Step Guide for the Beginner, Intermediate and Advanced User, Including Projects and Exercises”, 2019. Editorial: Independently published.
- MACVITTIE, L. “XAML in a nutshell”, 2009. Editorial: O’Reilly Media
- JAPIKSE, P. “Windows 10 development with XAML and C#”, 2017. Editorial: aPress

10.2. Webgrafía

- DUHAN, D. (20-Dic-2018) “Insight: Why it’s time to take the Sim Racing business seriously?” Disponible en:
<https://www.motorsport.com/esports/news/sim-racing-business-serious-money/4314653/> Visitada por última vez el: 2-Abr-2019
- TORIL, P. (2-Jul-2019) “SimRacing, ¿El último gran filón de los eSports?” Disponible en:
<https://www.masjerez.com/noticia/gajcd/simracing-el-ultimo-gran-filon-esports> Visitada por última vez el: 2-Abr-2019
- WIKIPEDIA “Glosary of motorsport terms” Disponible en:
https://en.wikipedia.org/wiki/Glossary_of_motorsport_terms Visitada por última vez el: 15-Jul-2019
- LOFTHOUSE, P. (3-Mar-2015) “What software do race engineers use in motorsport?” Disponible en:
<https://www.quora.com/What-software-do-race-engineers-use-in-motorsports> Visitada por última vez el: 15-Abr-2019
- MICROSOFT (25-Ene-2018) “Windows presentation foundation” Disponibe en:
<https://docs.microsoft.com/es-es/dotnet/framework/wpf/> Visitada por última vez el: 9-Jul-2019
- MICROSOFT (30-Ene-2018) “Guía de c#” Disponibe en:
<https://docs.microsoft.com/es-es/dotnet/csharp/> Visitada por última vez el: 9-Jul-2019
- SQLITE “SQLite” Disponible en: <https://www.sqlite.org/index.html> Visitada por última vez el: 9-Jul-2019
- GUDAPATI, S. “Selenium C# Webdriver Tutorial: NUnit Example” Disponible en:
<https://www.guru99.com/selenium-csharp-tutorial.html> Visitada por última vez el: 15-Jun-2019
- TWITTER (15-May-2015) (“Rubens Barrichello simracing tweet”) Disponible en:
<https://twitter.com/rubarrichello/status/731865583936000001> Visitada por última vez el: 16-Abr-2019

- SUTTILL, J. (21-Jul-2019) “F1 drivers Norris, Verstappen join to win virtual 24 hours of Spa” Disponible en: <https://www.autosport.com/esports/news/144910/norris-verstappen-win-virtual-24-hours-of-spa> Visitada por última vez el: 21-Jul-2019
- WPF TUTORIAL “WPF tutorial” Disponible en: <https://www.wpf-tutorial.com/> Visitada por última vez el: 9-Jul-2019
- MARTÍ, M. (8-Abr-2016) “¿Qué es el web scraping? Introducción y herramientas” Disponible en : <https://sitelabs.es/web-scraping-introduccion-y-herramientas/> Visitada por última vez el: 16-Abr-2019
- GASCÓN, U. (9-Mar-2018) “Scraping épico para gente sin APIs” Disponible en: <https://www.youtube.com/watch?v=-Gtv6BMvcgM> Visitada por última vez el: 16-Abr-2019

11. Anexos

11.1. Anexo I: Glosario de términos de carreras de coches

- **Stint:** periodo durante el cual un piloto está al volante de un coche en carreras donde están implicados más de un piloto. A menudo referido al tiempo entre pit stops.
- **Pit stop:** parada en boxes para realizar reparaciones en el vehículo, cambiar neumáticos, echar gasolina, cambiar de piloto, o cualquier combinación de las anteriores.
- **Pit lane:** carril adyacente a la pista en el cual se sitúan los garajes y se utiliza para realizar pit stops.
- **Driver Lineup:** Combinación de varios pilotos encargados de conducir un mismo coche en una carrera de resistencia.
- **SOF (Strength Of Field) (Único para iRacing):** Media de iRating de un equipo o una carrera.
- **iRating (Único para iRacing):** También conocido como IR. Número asignado a un piloto en iRacing que representa la habilidad de este. Como norma general, un piloto con más iRating tendrá una habilidad superior a un piloto con iRating inferior.
- **Safety Rating (Único para iRacing):** También conocido como sRating o SR. Indica la limpieza de un piloto dentro de la pista. Por regla general, un piloto será más seguro y tendrá una capacidad de mantenerse dentro de la pista y de no golpear a pilotos cuanto más Safety Rating tenga. No está relacionado de ninguna manera con SOF o iRating.

