

Universidad de las Ciencias Informáticas

UCI

Facultad 5 Realidad Virtual



Gestión de los perfiles de usuario en el videojuego de conducción de automóviles “Rápido y Curioso”.

Trabajo de Diploma para optar por el título de
Ingeniero en Ciencias Informáticas

Autor(es): Juan Leandro Asin Vera.

Orlando Fernández Cremé.

Tutor(es): Ing. Marvyn Márquez Rodríguez.

Ing. Andy Trujillo Rivero.

OPINIÓN DEL TUTOR DEL TRABAJO DE DIPLOMA.

Título:

Autores:

El tutor del presente Trabajo de Diploma considera que durante su ejecución el estudiante mostró las cualidades que a continuación se detallan.

Por todo lo anteriormente expresado considero que el estudiante está apto para ejercer como Ingeniero en Ciencias Informática; y propongo que se le otorgue al Trabajo de Diploma la calificación de:

Ing. Marvyn Márquez Rodríguez

Firma _____

Ing. Andy Trujillo Rivero

Firma _____

Fecha

AGRADECIMIENTOS

No resulta fácil expresar los agradecimientos de un trabajo que aunque modesto en cuanto a su realización constituye la culminación del sacrificio de muchos años de estudio. Quisiera comenzar agradeciendo a mi madre sin la que indudablemente no habría podido alcanzar este sueño de convertirme en un profesional y de quién además siento un gran amor, cariño, respeto y una profunda admiración por haberme criado con tanto sacrificio y siempre estar para mí.

Además quiero agradecer:

A mis abuelos por brindarme apoyo en todo momento y ser sabios consejeros.

A Freddy por ser guía de mis estudios y ejemplo a seguir en cuanto inteligencia y determinación en la vida.

A mis hermanos por ser las personas que me inspiran a superarme para brindar un buen ejemplo que puedan asumir luego en su joven vida.

A mi padre y a mi tía Magda por aconsejarme y apoyarme cuando tienen la oportunidad de estar cerca de mí.

A mi primos Raulito y Robertico por ser compañeros de batalla.

A mi compañero de tesis Chepa por su amistad, su aporte de conocimientos y por luchar junto a mí para lograr nuestro sueño de ser ingenieros.

A mis tutores Andy y Marvyn que se han portado como familia, han aportando sus consejos y experiencias para la realización de este trabajo sin considerarlo molestia alguna.

A toda mi familia por alentarme, ayudarme y brindarme la mano protectora en todos estos años de estudio.

A todos mis amigos que a lo largo de estos años hemos compartido alegrías y momentos difíciles, pero siempre nos hemos mantenido juntos y como dice el buen cubano tirando pa'lante.

Orlando Fernández Cremé.

Especialmente a mis padres, que siempre han sido una guía y un ejemplo a seguir en toda mi vida como estudiante y persona, a ellos le doy gracias por ser el hombre que soy, los quiero con el corazón y por ustedes voy hasta el fin del mundo, un beso y un abrazo bien grande

A mi queridísima hermana que voy a estar a su lado por siempre y la quiero muchísimo.

A mi novia que me ha acompañado en momentos difíciles y me hace muy feliz

A mis amigos de siempre, sin orden de privilegios, Raciél, Miralles, Raudel, Eurlis y Ernesto que me brindan su amistad incondicionalmente.

A mi compañero de tesis (Orlando) por aguantarme en estos largos meses de trabajo

A los socios y amistades de la UCI.

A los tutores (Andy y Marvyn) por siempre encontrar un tiempo para atendernos y evacuar nuestras dudas.

A familia enterita, para que no se quede nadie sin mencionar y no se pongan celosos.

En fin a todos aquellos que han hecho posible este preciado sueño, les agradezco de corazón.

Juan Leandro Asin Vera.

DEDICATORIA

A mi madre la batalladora incansable, madre incondicional y amiga.

A mi hermanito el chico super super.

A Freddy y a mis abuelos, verdaderos familiares.

A todos mis familiares que me han ayudado en la vida.

Orlando Fernández Cremé.

A mis padres con todo el amor de mi corazón por toda su confianza y dedicación!!!!

A mi hermana por ser tan especial.

A mi novia (Lanny) que la quiero muchísimo

A mi familia

A todos mis amigos por su apoyo incondicional

Juan Leandro Asin Vera.

RESUMEN

La gestión de los perfiles de usuario es una importante funcionalidad que está presente en muchos videojuegos de prestigio y calidad actualmente, enfocada a la personalización del ambiente y a la comodidad de los usuarios. La presente investigación propone un método para gestionar dichos perfiles en el videojuego de conducción de automóviles “Rápido y Curioso”, que se desarrolla en la Facultad de Entornos Virtuales de la Universidad de las Ciencias Informáticas. Se describen diferentes estrategias empleadas para almacenar la información relativa a los jugadores o al juego en general, haciendo énfasis en la seleccionada para la realización de este trabajo, que es la creación de ficheros binarios. En la investigación se describe además la estructura del modelo de persistencia propuesto para el videojuego en cuestión. Se realizó un diseño de clases que modela computacionalmente el problema. Dicho modelo fue implementado, obteniéndose un módulo capaz de administrar el perfil de cada jugador.

Palabras clave:

Videojuegos, videojuego de conducción de automóviles, perfiles de usuario, fichero binario.

CONTENIDO

AGRADECIMIENTOS	I
DEDICATORIA	III
RESUMEN	IV
INTRODUCCIÓN	1
CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA	6
1.1 PROCESO DE DESARROLLO DE VIDEOJUEGOS	7
1.2 DIFERENTES MÓDULOS QUE CONFORMAN UN VIDEOJUEGO DE CONDUCCIÓN DE AUTOMÓVILES	8
1.2.1 MÓDULO GRÁFICO.....	8
1.2.2 MÓDULO SONIDO	8
1.2.3 MÓDULO RED	9
1.2.4 MÓDULO INTELIGENCIA ARTIFICIAL	9
1.2.5 MÓDULO FÍSICO-MATEMÁTICO.....	10
1.2.6 MÓDULO GESTIÓN DE PERFILES DE USUARIO.....	10
1.3 VENTAJAS DE GESTIONAR LOS PERFILES DE USUARIO EN UN VIDEOJUEGO	11
1.4 PERSISTENCIA DE LA INFORMACIÓN DIGITAL	12
1.4.1 GESTORES DE BASE DE DATOS	13
1.4.2 FICHEROS XML.....	13
1.4.3 FICHEROS DE TEXTO	14
1.4.4 FICHEROS BINARIOS	14
1.5 GESTIÓN DE LOS PERFILES DE USUARIOS EN DIFERENTES GÉNEROS DE VIDEOJUEGOS	15
1.5.1 VIDEOJUEGOS TIRADOR.....	15
1.5.2 VIDEOJUEGOS DE ROL	17
1.5.3 VIDEOJUEGOS DEPORTIVOS	18
1.5.4 VIDEOJUEGOS DE CARRERAS	19
1.6 EXPERIENCIAS SIMILARES EN PROYECTOS DEL POLO DE REALIDAD VIRTUAL DE LA FACULTAD 5	20
1.7 METODOLOGÍA Y HERRAMIENTAS DE DESARROLLO	21

CONCLUSIONES	23
CAPÍTULO 2: CARACTERÍSTICAS DEL SISTEMA	24
2.1 NECESIDAD DE LA CREACIÓN DEL MÓDULO PARA LA GESTIÓN DE LOS PERFILES DE USUARIOS EN EL VIDEOJUEGO “RÁPIDO Y CURIOSO”	25
2.2 SOLUCIÓN TÉCNICA	25
2.3 DESCRIPCIÓN DE LAS FUNCIONALIDADES POR INTERFAZ	26
2.4 DATOS PERSISTENTES	28
2.4.1 SELECCIÓN DE LOS DATOS QUE SE SALVAN DEL JUGADOR.....	28
2.4.2 ESPECIFICACIÓN DE LA ESTRUCTURA INTERNA DEL FICHERO.....	29
2.5 APARIENCIA DEL FICHERO	33
2.6 ANÁLISIS DEL DOMINIO	34
2.6.1 MODELO DEL DOMINIO.....	35
2.6.2 CONCEPTOS DEL MODELO DE DOMINIO.....	36
2.7 REQUISITOS DEL SOFTWARE	36
2.7.1 REQUISITOS FUNCIONALES.....	37
2.7.2 REQUISITOS NO FUNCIONALES.....	37
2.8 MODELO DE CASOS DE USOS DEL SISTEMA	38
2.8.1 ACTOR DEL SISTEMA.....	38
2.8.2 CASOS DE USO DEL SISTEMA.....	38
2.8.3 DIAGRAMA DE CASOS DE USO DEL SISTEMA.....	40
2.8.4 EXPANSIÓN DE LOS CASOS DE USO DEL SISTEMA.....	41
2.9 ANÁLISIS	45
2.9.1 DIAGRAMAS DE CLASE DE ANÁLISIS.....	46
CONCLUSIONES	49
CAPÍTULO 3: DISEÑO E IMPLEMENTACIÓN DEL SISTEMA	50
3.1 PATRONES	51
3.2 DISEÑO DE LOS PAQUETES	52
3.2.1 PAQUETE MÓDULO GESTIÓN DE LOS PERFILES DE USUARIO.....	53

3.2.2 PAQUETE MENÚ	55
3.3 DESCRIPCIÓN DE LAS CLASES	55
3.4 DIAGRAMAS DE SECUENCIA	59
3.5 DIAGRAMA DE COMPONENTES	59
CONCLUSIONES.....	60
CAPÍTULO 4: PRUEBAS	61
4.1 ESPECIFICACIONES DE HARDWARE Y SOFTWARE.....	62
4.2 DISEÑO DE CASOS DE PRUEBA POR CADA CASO DE USO	62
4.2.1 CASO DE USO CREAR PERFIL	62
4.2.2 CASO DE USO ELIMINAR PERFIL.....	66
4.2.3 CASO DE USO CARGAR PERFIL	68
4.2.4 CASO DE USO EDITAR PERFIL	70
CONCLUSIONES.....	81
CONCLUSIONES.....	82
RECOMENDACIONES	83
BIBLIOGRAFÍA.....	84
GLOSARIO	88
ANEXOS.....	91
ANEXO I DIAGRAMA DE COMPONENTES.....	91
ANEXO II DIAGRAMA DE SECUENCIA CU CREAR PERFIL	92
ANEXO III DIAGRAMAS DE SECUENCIAS CU EDITAR PERFIL	93
ANEXO IV DIAGRAMA DE SECUENCIA CU CARGAR PERFIL.....	98
ANEXO V DIAGRAMA DE SECUENCIA CU ELIMINAR PERFIL.....	99

INTRODUCCIÓN

El desarrollo tecnológico ha acelerado la velocidad y capacidad de procesamiento de las computadoras, permitiendo que el desarrollo de videojuegos, se convirtiera en unos de los negocios más rentables del planeta, con ingresos similares o superiores a los de otros medios de entretenimiento como la música y el cine.

Actualmente los videojuegos han evolucionado en todos los aspectos, desde los gráficos, el sonido y la simulación físico-matemática que nos sumergen en un mundo virtual cada vez más semejante a la realidad, hasta los increíbles comportamientos desarrollados por agentes artificiales controlados por la computadora.

Con el ánimo de ser competitivas en un mercado tan complicado, las empresas desarrolladoras de videojuegos están en un constante proceso de investigación, creación y desarrollo tecnológico que les permita obtener productos más ingeniosos y atractivos, y que a la vez se ajuste a las necesidades de los usuarios. Precisamente en este sentido tratan de hacer juegos configurables, personalizables y con los que los jugadores se sientan identificados. Una forma de alcanzar este objetivo es gestionando los diferentes perfiles de usuario, donde cada cual podrá establecer sus preferencias en relación a varios aspectos del juego.

La gestión de los perfiles de usuario en un videojuego brinda las facilidades de configuración a los jugadores, les permite adaptar el entorno del videojuego a sus necesidades, incluso influye en el desarrollo de la lógica del juego, además de permitir que todas estas configuraciones sean guardadas para partidas futuras sin tener que hacer las mismas modificaciones cada vez que se inicie la partida, por esas razones casi todos los videojuegos brindan la posibilidad de gestionar los perfiles de usuario, lo cual es una característica importante y necesaria para el agrado y aceptación de los clientes.

Cuba ha centrado esfuerzos en el desarrollo de la industria del software, como una de las principales tareas de la Batalla de Ideas para contribuir al desarrollo socio-económico de la nación. Con el objetivo de insertar a nuestro país en el mercado del software a nivel mundial y fomentar la informatización, se crea en el 2002 la Universidad de las Ciencias Informáticas (UCI). Desde sus inicios la universidad ha tratado de revolucionar la industria del software nacional, además de que ha alcanzado notables resultados en el ámbito internacional.

La UCI, en aras de convertir la informática en una herramienta fundamental en el desarrollo de la producción de la nación cubana y en una de las esferas más productivas, se ha transformado en un centro científico donde se vincula la docencia con el proceso productivo en el desarrollo de varios perfiles de la informática. Uno de ellos es el de Entornos Virtuales, dentro del cual se encuentra el proyecto Juegos de Consola, que desarrolla videojuegos en tres dimensiones (3D) para Computadoras Personales (del inglés Personal Computer) y videoconsolas. El primer compromiso de este proyecto productivo fue desarrollar el videojuego “Rápido y Curioso”, dirigido a lograr una sana recreación en niños que competirán en autos clásicos por algunos lugares del casco histórico de Ciudad de la Habana, en carreras contrarreloj o contra otros usuarios.

Situación Problémica:

La inexistencia de un módulo para la gestión de los perfiles de usuario en el videojuego “Rápido y Curioso” trae como inconveniente para los jugadores no poder crearse un perfil que los identifique, no poder modificar las opciones relacionadas con los controles de teclado, resolución, cantidad de jugadores y algunos parámetros de red para el modo multijugador (del inglés multiplayer), además de no poder salvar las configuraciones que realiza, tampoco puede eliminar y cargar los perfiles ya existentes. Todas estas importantes características están presentes en la mayoría de los videojuegos de prestigio y calidad en la actualidad.

Problema Científico:

¿Cómo gestionar los perfiles de usuario en el videojuego de conducción de automóviles “Rápido y Curioso”?

Objeto de Investigación:

Desarrollo de videojuegos de conducción de automóviles.

Campo de Acción:

Gestión de perfiles de usuario.

Objetivo General:

Desarrollar un módulo para la gestión de los perfiles de usuario en el videojuego de conducción de automóviles “Rápido y Curioso”.

Tareas de Investigación:

- Investigar sobre el proceso de desarrollo de los videojuegos, sus características y los diferentes módulos que los conforman.
- Caracterizar algunas variantes empleadas para gestionar los perfiles de usuario en videojuegos.
- Identificar los elementos a tener en cuenta para gestionar los perfiles de usuario en el videojuego de conducción de automóviles “Rápido y Curioso”.
- Seleccionar y diseñar el modelo de persistencia que se utilizará para exportar e importar los perfiles de usuario en el videojuego de conducción de automóviles “Rápido y Curioso”.
- Diseñar e implementar el módulo para la gestión de los perfiles de usuario en el videojuego de conducción de automóviles “Rápido y Curioso”.
- Probar las funcionalidades del módulo para gestionar los perfiles de usuario en el videojuego de conducción de automóviles “Rápido y Curioso”.

Idea a Defender:

La incorporación del módulo de gestión de los perfiles de usuario al videojuego “Rápido y Curioso” incrementará la aceptación de los clientes.

Métodos de investigación empleados:

Para desarrollar esta investigación se utilizaron diferentes métodos científicos, entre los que se destacan el histórico-lógico, el analítico-sintético y la observación.

Métodos Teóricos:

Histórico-Lógico: Permite estudiar sobre la historia y desarrollo de los videojuegos, los diferentes módulos que lo conforman y además sobre las técnicas y métodos utilizados en los videojuegos para gestionar los perfiles de usuario.

Analítico – Sintético: Permite concretar y resumir el conocimiento obtenido de los materiales consultados sobre el tema de gestión de perfiles de usuario y del propio conocimiento como usuarios de videojuegos con el fin de utilizarlo en el desarrollo de esta investigación.

Métodos Empíricos:

Observación: Se empleó para identificar en diferentes juegos la forma de gestionar los perfiles de usuario. Además, para observar las interfaces gráficas creadas y corregir posibles errores de diseño y/o funcionamiento.

Este documento está conformado por 4 capítulos:

Capítulo 1: En este capítulo se destacan algunos aspectos y características del proceso de desarrollo de un videojuego, y las formas de gestionar los perfiles de usuario en sus principales géneros. Se hace referencia además a las variantes empleadas para guardar datos persistentemente en el disco duro de una computadora; así como los diferentes módulos que conforman un videojuego de conducción de automóviles.

Capítulo 2: En este capítulo se describe la solución propuesta para resolver el problema de la presente investigación. Se identifican los elementos que conformarán el perfil de usuario del videojuego en cuestión, se toma partido por uno de los modelos de persistencia estudiados y se diseña cómo quedará internamente. Además se incluyen algunos artefactos de la etapa de inicio del proceso de desarrollo de software: análisis de dominio y captura de requisitos (funcionales y no funcionales).

Capítulo 3: En este capítulo se expone el diseño del módulo implementado, de esta manera se explica cómo se pretende dar solución a las funcionalidades derivadas de los requisitos funcionales. Se detallan los patrones usados en la solución y se explican las principales clases. También quedan plasmados los

diagramas de colaboración confeccionados a partir de los casos de uso y los diagramas de despliegue y componentes.

Capítulo 4: En este capítulo se realizan las pruebas correspondientes al sistema, para verificar si el mismo funciona correctamente. Además se muestran imágenes de la aplicación desarrollada en ejecución.

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

Introducción:

En este capítulo se destacan algunos aspectos y características del proceso de desarrollo de un videojuego, y las formas de gestionar los perfiles de usuario en sus principales géneros. Se hace referencia además a las variantes empleadas para guardar datos persistentemente en el disco duro de una computadora; así como los diferentes módulos que conforman un videojuego de conducción de automóviles. Todas estas consideraciones son obtenidas a través de un estudio de la bibliografía existente sobre cada uno de los temas, experiencias propias obtenidas en pruebas realizadas a diferentes videojuegos para observar y conocer la forma de gestionar los perfiles de usuario.

Se mencionan además distintas formas usadas para salvar información cuya existencia se necesita que exceda el tiempo de ejecución de una aplicación. Estos aspectos contribuirán a proporcionar un mayor conocimiento del estado del arte en el tema de los perfiles de usuario en videojuegos, y posibilitarán elegir el camino más adecuado para resolver el problema de la presente investigación.

1.1 Proceso de desarrollo de videojuegos

Para tener una idea más clara sobre los videojuegos se debe conocer el significado de los mismos: un videojuego se define como un programa informático que sirve para entretener a sus usuarios [1], se basa en la interacción entre una o varias personas y un dispositivo electrónico. El dispositivo que ejecuta el juego, puede ser una computadora, una videoconsola o un celular.

No todos los videojuegos sirven para entretener; también los hay que pueden crear toda una variedad de emociones, como, por ejemplo: miedo. [1]

Aproximadamente en 1947 tiene sus inicios la historia de los videojuegos, comenzó con una idea de realización de un videojuego de misiles creada y patentada por Thomas T. Goldsmith Jr. y Estle Ray Mann. Algunos de los primeros videojuegos fueron el OXO (1952), el cual es un juego gráfico computarizado, derivado de la tesis doctoral de Alexander Sandy Douglas [1], y el Tennis For Two creado en 1958. Desde aquellos primeros intentos, los videojuegos han evolucionado considerablemente; si bien antes eran bastante sencillos y ordinarios en la actualidad esto ha sufrido grandes cambios debido al desarrollo de la tecnología en materia de hardware, y por supuesto también a las técnicas y algoritmos implementados por los programadores.

Cada uno de estos juegos tiene sus propias características, pero coinciden en algunos puntos como la necesidad de entretener al usuario. Para lograrlo es necesario el uso de elementos de video, audio, retroalimentación táctil, que cuenten con una historia interesante, y que le planteen un desafío aceptable para el jugador.

Desde el punto de vista del desarrollo, una tarea compleja como esta necesita ser repartida entre los miembros de un equipo. Para que dicho trabajo se lleve a cabo con éxito, se dividen las tareas en módulos independientes y cada cual trabaja en uno. A continuación se mencionan algunos de los módulos en los que se puede dividir un videojuego, específicamente uno de conducción de automóviles, ya que es el objeto de esta investigación.

1.2 Diferentes módulos que conforman un videojuego de conducción de automóviles

Un módulo es una parte de un programa, o más general, un componente de un sistema que tiene una interfaz bien definida para interactuar con otros módulos. Generalmente los módulos pueden ser reutilizados en varias partes de un programa o en otros programas. El desarrollo de un videojuego, puede dividirse en módulos para facilitar su realización, para independizar las responsabilidades y lograr de esta manera que los cambios que se realicen no tengan por qué afectar toda la aplicación.

1.2.1 Módulo Gráfico

Este módulo es muy importante, pues la retroalimentación visual le llega al usuario gracias a los gráficos. Mientras más atractivos resulten, mayor sensación de inmersión se conseguirá, permitiéndole al jugador adentrarse cada vez más en el juego. [13]

Los videojuegos pueden ofrecer más emoción gracias a las mejoras en animaciones y expresiones faciales, el reflejo de los personajes en el espejo, el diseño atractivo de los autos en un videojuego de carreras, los efectos de iluminación, sombras, choques que provocan chispas, marcas de derrape en las calles, etc.

Los avances en las animaciones de un videojuego son posibles gracias al nivel tecnológico existente hoy en día, a las mejoras en los procesadores, las nuevas tarjetas gráficas y el hardware en general. Realizar videojuegos con buenos gráficos es tarea primordial de los desarrolladores, ya que este es uno de los requisitos que más valoran los usuarios, y es determinante en muchas ocasiones en las ventas y aceptación del producto.

1.2.2 Módulo Sonido

El sonido es otra importante forma de interacción en un videojuego. [2] Muchas de las grandes empresas desarrolladoras de videojuegos invierten considerables sumas de dinero en la banda sonora y los efectos de sonido, contratando incluso a reconocidos compositores y orquestas para que interpreten sus canciones. Esto no lo hacen por gusto, sino porque se ha comprobado que aumenta el suspenso, la emoción y por consiguiente la aceptación del videojuego por parte del público.

En el caso de los videojuegos de carros, para darle realismo a la trama se usan sonidos grabados de la vida real. Por ejemplo, el sonido del motor de cada uno de los modelos que se muestran en el juego, el producido por las gomas al acelerar o frenar, la exclamación del público, las voces de los jueces de arrancada, la del segundo conductor indicando la peligrosidad del próximo segmento del circuito, etc.

Todo ello es controlado en un juego por el módulo de sonido, que generalmente hace uso de bibliotecas de clase especializadas como *OpenAL* o *DirectSound* para mezclar y producir efectos de sonido tridimensionales, capturar sonidos de un micrófono o controlar propiedades de audio como el volumen y la frecuencia, además de la posición del emisor y el receptor de la señal sonora.

1.2.3 Módulo Red

La inclusión de tecnologías de red en los juegos se ha convertido en un mecanismo clave para aumentar las ventas y la calidad del mismo. Al añadir una opción multijugador incrementa su tiempo de vida (tiempo que el jugador puede utilizar el juego sin aburrirse) ya que tiene la posibilidad de enfrentarse a oponentes de su mismo nivel, aumentando el valor que el usuario percibe del juego.

El módulo de red en un videojuego se encarga de establecer una conexión rápida y estable entre el cliente y el servidor, garantizando la satisfacción de los usuarios que usan el modo multijugador, mediante el que pueden enfrentarse a otros competidores de cualquier lugar, con el único requisito de que estén conectados mediante una red con un ancho de banda adecuado.

La arquitectura de red (o de comunicación) se refiere a la forma en que se implementa, a nivel lógico y de aplicación, el software que gestiona la comunicación entre las computadoras conectadas. Es muy importante, pues define el ancho de banda a consumir, así como la complejidad en el desarrollo del juego para lograr una coherencia entre los nodos terminales conectados y el tiempo de reacción, factores importantes para lograr resultados satisfactorios. [25]

1.2.4 Módulo Inteligencia Artificial

La inteligencia artificial es un módulo realmente importante a la hora de diseñar cualquier videojuego, ya que es la encargada de dar capacidad de decisión y acción a los jugadores no controlados por el usuario.

Mientras más “inteligentes” sean los oponentes artificiales, más difícil y entretenido será el juego, ya que los jugadores tendrán que emplearse a fondo para poder avanzar. Estos elementos se garantizan en el módulo de inteligencia artificial.

Especialmente en un videojuego de conducción de automóviles los jugadores tienen que enfrentar situaciones complejas, como por ejemplo cuando se intente sobrepasar a otro vehículo el mismo tratará de bloquearlo. Los autos inteligentes cometen pocas imprecisiones, por lo que hay que aprovechar cada uno de sus fallos para poder ganar la carrera. [20]

1.2.5 Módulo Físico-Matemático

Este módulo constituye un pilar fundamental en la confección de un videojuego, es el encargado de manejar toda la matemática y la física, como por ejemplo la detección de colisiones y las reacciones de los objetos del videojuego ante la acción de fuerzas.

Centrando la atención en los videojuegos de conducción de automóviles dicho módulo se pone de manifiesto en el correcto funcionamiento de los autos, si al coche se le modifican los parámetros de velocidad máxima, aceleración, fuerza, estabilidad en la dirección, adherencia a la pista, etc. esto influirá en su maniobrabilidad y comportamiento en la carrera [19]; si el auto choca contra cualquier obstáculo en la carrera, debe cambiar su orientación y velocidad, incluso en muchos juegos se ven las deformaciones en la carrocería, incrementando notablemente el realismo y la aceptación del juego.

1.2.6 Módulo Gestión de perfiles de usuario

A medida que se ponen a prueba sus habilidades, se perfeccionan y descubren nuevos niveles, los usuarios se convierten en consumidores activos del producto que puso en sus manos un grupo de creadores pensando en la satisfacción del cliente en primer lugar.

Es útil y muy apreciada por los jugadores la posibilidad de personalizar el juego a su antojo, y que además se le brinde la posibilidad de guardar los avances logrados al adentrarse en la trama del mismo.

En los juegos de carros como el “*Need for Speed Underground*” por ejemplo, se emplea un sistema en el que el jugador debe competir en una serie de eventos para ganar dinero. A medida que avanza, podrá

comprar nuevos autos o personalizar el que ya tiene para mejorar su rendimiento, así estará en condiciones de competir en circuitos de mayor dificultad.

“*Gran Turismo*” constituye otro impresionante juego de carreras, contiene once circuitos, múltiples eventos contrarreloj y la modalidad de carreras de resistencia. El jugador tiene la posibilidad de usar varios modelos de autos. Comienza con un crédito con el que puede comprar coches nuevos o usados de las tiendas especializadas de fabricantes. A medida que avanza y gana eventos, el jugador obtiene más créditos, los que pueden ser usados en modificaciones para su auto actual o para comprar uno nuevo.

El objetivo del juego “*Burnout Paradise Ultimate Boxy*” es obtener puntos mediante choques espectaculares y saltos imposibles a toda velocidad, para poder seguir avanzando por la ciudad y adquirir nuevos coches por el camino. Dichos logros son guardados en registros correspondientes a cada perfil de usuario. Es una dinámica de juego sencilla pero muy divertida, por lo que centra la atención de muchos jugadores. [21]

Precisamente la utilidad del módulo encargado de gestionar los perfiles de usuario en un videojuego reside en brindarle al usuario la posibilidad de configurar el entorno a su gusto, editar las opciones que estime conveniente y de guardar los progresos que sea capaz de lograr dentro del juego. Para ello puede crear su propio perfil, además de cargar o eliminar uno de los existentes. Estas características pueden contribuir a incrementar la aceptación y comercialización de un videojuego. Mientras más opciones de configuración se le brinden al usuario, más complacido estará.

1.3 Ventajas de gestionar los perfiles de usuario en un videojuego

Las principales ventajas que brinda la gestión de los perfiles de usuarios en un videojuego son las siguientes:

- Cuando se interrumpe el fluido eléctrico y la PC no cuenta con un backup, entonces se apaga el CPU (del inglés Central Processing Unit) y lógicamente se pierden todos los avances logrados en el juego hasta ese momento. Si el juego cuenta con una función que permita guardar automáticamente los avances, entonces no es necesario preocuparse por tener que comenzar la partida desde el principio si ocurre esta indeseable situación, que se puede dar en cualquier momento por motivos ajenos a la voluntad del usuario.

- A la hora de comenzar a jugar sería incómodo cada vez que se inicie el juego tener que configurar las opciones del juego en cuanto a controles, volumen, resolución de la pantalla, nivel de dificultad del juego u otras características particulares que los usuarios deseen tener configuradas. Una vez que el usuario del juego cuente con la posibilidad de salvar su perfil, solo tendrá que hacer esta configuración la primera vez que juegue, ahorrando tiempo y brindándole mayor comodidad.
- La estrategia de contar con un fichero de salva por cada usuario (que se guarda en una dirección que tenga establecida el juego) es una idea brillante, la cual permite a los jugadores transportar los ficheros correspondientes a su perfil hacia cualquier PC en la que desee jugar. Allí tan solo tendría que copiar los ficheros en la misma dirección asociada al juego y contaría con la misma configuración, avances en el juego y detalles que tenía en la PC original.
- La seguridad, confiabilidad y conformidad que brinda a los usuarios la salva automática de su perfil en un juego, constituye una satisfacción enorme que todo desarrollador debería brindar. El usuario, al sentirse seguro de que sus partidas están garantizadas mediante la opción de autoguardado, no tiene que preocuparse por hacerlo manualmente, y solo centra sus preocupaciones en avanzar en el juego para ganarlo. Además, al sentir seguridad y confianza puede recomendar el juego a otros amigos y el círculo de usuarios seguirá creciendo.

1.4 Persistencia de la información digital

Persistencia es el término utilizado en computación para guardar y recuperar datos u objetos. La mayoría de las aplicaciones siguen una estructura sencilla de funcionamiento: recuperación de datos de una anterior ejecución, procesado, y salvaguarda de los nuevos datos. [23]

En muchas ocasiones se necesita que los datos manejados por una aplicación tengan un tiempo de vida que exceda el tiempo que dicha aplicación se mantiene en activo. O sea, que cuando se cierre y se vuelva a abrir, se puedan tener nuevamente los mismos datos que antes de cerrarla. Esto se conoce como persistencia de datos. Existen diversas formas de garantizar la persistencia en una aplicación informática que se analizan a continuación.

1.4.1 Gestores de base de datos

Un gestor de base de datos o sistema de gestión de base de datos es un software que permite introducir, organizar y recuperar la información de las bases de datos; en fin, administrarlas. [5] Permiten a los usuarios crear bases de datos y especificar su esquema por medio de algún lenguaje de definición, además de consultar los datos y modificarlos usando para ello un lenguaje de consulta y manipulación.

Una base de datos es un conjunto de datos almacenados entre los que existen relaciones lógicas y que ha sido diseñada para satisfacer los requerimientos de información. Son utilizadas en sistemas que requieren una interacción fluida con la aplicación que las usa. Además de los datos, también se almacena su descripción. [24]

Existen videojuegos que usan las bases de datos con el fin de almacenar información para su funcionamiento y gestión, por ejemplo el videojuego “*World of Warcraft*” basa toda su estructura en las consultas que realiza sobre la base de datos que contiene información referente al entorno del videojuego y la información de cada jugador.

También el simulador web “*Second Life*” usa las facilidades de las bases de datos para almacenar todo el volumen de información que puede generar la creación de miles de personajes distintos, cada uno con diferentes características y modos de vida. [16]

La ventaja del uso de base de datos en un videojuego tiene que ver directamente con el volumen de información a manejar, es recomendable usarlas cuando son muchos datos. Se emplean generalmente en videojuegos que fueron diseñados específicamente para aceptar muchos jugadores online a cualquier hora del día y en cualquier lugar del planeta, por lo que los volúmenes de datos generados son impresionantes.

1.4.2 Ficheros XML

XML (del inglés Extensible Markup Language) desarrollado por del W3C (del inglés World Wide Web Consortium), es una versión simplificada del SGML (del inglés Standard Generalized Markup Language) es considerado un método para introducir datos estructurados en un fichero de texto. [6]

XML consiste en varias reglas y convenciones definidas para conformar formatos texto para almacenar datos, con el fin de crear ficheros que sean fáciles de generar y leer por una computadora. De esta

manera se evitan problemas muy comunes como la falta de extensibilidad, soporte para la internacionalización y dependencia a una determinada plataforma.

Tiene un papel muy importante en la actualidad ya que permite la compatibilidad entre sistemas para compartir la información de una manera segura, fiable y fácil. [12]

Este tipo de fichero puede ser utilizado en los juegos web para la transferencia de información, ya que un cliente puede escribir en un documento los datos en texto plano que estaba manejando el formato XML sin necesidad de tener alguna aplicación en específico para este fin y trabajar con estos datos.

En el caso de los juegos mencionados en el epígrafe anterior (*1.4.1 Gestores de base de datos*) como alternativa para la migración de datos puede usar XML ya que este formato es muy sencillo para mover datos de una base de datos a otra.

1.4.3 Ficheros de texto

Los ficheros de texto son la manera que se utiliza en la programación informática para salvar los datos procesados por los programas. En los ficheros de texto la información se guarda como caracteres ASCII u otro estándar de codificación, lo que los hace eficaces a la hora de guardar textos y números pequeños. Estos archivos están compuestos por bytes que representan caracteres ordinarios como letras, números y puntuación, también incluye algunos caracteres de control como tabulaciones, separaciones de líneas y fin de línea. Esta simplicidad permite que una gran variedad de programas pueda leer y editar ese contenido. [14]

Es una alternativa que puede ser usada por cualquier videojuego para salvar información de la aplicación o bien de los jugadores que la usan. Solamente tendrían que plantearse algunas interrogantes acerca de la seguridad de los archivos, ya que los mismos exponen texto plano que es configurable con facilidad, además de que es entendible; razón que puede afectar el uso de ficheros de texto en caso de querer guardar información que sea necesario se mantenga oculta.

1.4.4 Ficheros binarios

Es un archivo que contiene información de cualquier tipo, codificada en forma binaria para el propósito de almacenamiento y procesamiento en computadoras. En el formato binario los números se guardan tal y

cómo están representados en la memoria: los números enteros representan dos bytes, los reales cuatro bytes, etc. [14] El uso de ficheros binarios aumenta los niveles de confidencialidad de la información del archivo, dificultándose algún tipo de modificación efectiva fuera del entorno de modelación. [26]

Las últimas dos características explicadas son aprovechadas por algunos videojuegos para salvar información que se necesita leer en el menor tiempo posible, que sea confidencial y difícil de cambiar (al menos manualmente no se conseguirán resultados coherentes) Por ejemplo, en el videojuego “MVP 08” el fichero de configuración de los perfiles de usuario está en formato binario, aumentando la confidencialidad de los datos de cada jugador y a su vez que el sistema pueda realizar una lectura rápida sin tener que hacer conversión de datos.

1.5 Gestión de los perfiles de usuarios en diferentes géneros de videojuegos

Como se analizó anteriormente, la gestión de los perfiles de usuario es una característica que muchos videojuegos poseen. Para ello emplean diversas técnicas y formas de almacenar físicamente la información, seleccionado entre las descritas en el epígrafe anterior (1.4 *Persistencia de la información digital*) la que más se ajuste a sus necesidades.

A continuación se hará una descripción de cómo se manejan estas funcionalidades en varios de los géneros de videojuegos existentes. La manera de hacer la salva de los perfiles es bastante similar en muchos de ellos, pero vale la pena tratarlos por separado, ya que las pequeñas diferencias que existen constituyen ventajas provechosas a tener en cuenta a la hora de decidir una variante para resolver el problema que se aborda en esta investigación.

1.5.1 Videojuegos Tirador

Son juegos con una temática muy sencilla, donde el jugador básicamente sólo puede moverse y disparar para cumplir su objetivo. Estos juegos también se llaman “shooters”, que es la traducción al inglés de pistoleros. [1]

Dentro de este tipo de videojuegos se encuentra el “*Medall of Honor*” el cual es muy popular, cuenta básicamente con dos modos de juegos: multijugador, que permite la conexión en red de varios usuarios

para formar equipos contrarios; y jugador individual, donde se juega contra otro usuario o en modo de misión, en el cual el jugador debe cumplir una serie de peligrosas tareas para llegar a obtener la victoria. En este último se realizan salvas cada vez que se avanza en el juego, se pasa una emboscada, se ejecuta una explosión o se realiza una maniobra complicada, el usuario puede salvar esos logros haciendo uso de la tecla designada para este fin (F5). [18]

También se realizan salvas de información cuando el jugador configura los controles de teclado, nivel de dificultad del juego, nivel de detalle de los gráficos, el volumen de sonido o la sensibilidad del dispositivo de retroalimentación de fuerza. Además cuando se cumplen tareas claves de la misión, el mecanismo de salva automático del juego se activa y realiza esta operación. Todas las salvas de las misiones son grabadas en ficheros dentro de una carpeta del directorio del juego reservada para este fin, y dentro de ella una para cada perfil existente.



Figura 1.8 Vista de la salva en el videojuego "Medall of Honor".

1.5.2 Videojuegos de rol

En este tipo de juego el jugador encarna uno o varios personajes con características propias a los que debe desarrollar una personalidad, combinando una serie de características como son la fuerza, inteligencia, magia y ciertas habilidades. [1]

El videojuego “*World of WarCraft*” de la compañía estadounidense “*Blizzard Entertainment*” forma parte de este género. Usa un servidor de base de datos (MySQL Admin) en el cual se encuentran todos los elementos que lo conforman, desde el total de vida de un simple personaje controlado por la computadora hasta la más complicada información de administración de permisos.

A medida que un jugador va adentrándose en las misiones, obteniendo elementos y subiendo de nivel, se le brinda la oportunidad de guardar sus progresos, de forma tal que pueda pausar el juego y posteriormente reiniciar donde se había quedado. Por otra parte, el juego presenta tolerancia a fallos, o sea, que en caso de existir un error en el servidor, o se interrumpa la electricidad y se cierre el juego bruscamente, el usuario al iniciar nuevamente la aplicación tiene salvado todos sus avances y logros. [17]

La estrategia seguida por el administrador del servidor es programar el autoguardado del servidor cada cierto tiempo (no mayor de 3 minutos). Además dentro del juego se puede crear una macro que efectúa una salva cada vez que el usuario estime conveniente: a petición suya se envía un comando al servidor para que guarde todos los cambios hasta ese momento. La configuración de los controles de teclado, nivel de detalle de los gráficos, volumen del ambiente del juego implica una salva de información en el servidor para garantizar estos cambios en próximos inicios de partida.



Figura 1.3 Vista de la salva en el videojuego “World of WarCraft”.

1.5.3 Videojuegos Deportivos

Este tipo de juegos se caracteriza por tener como núcleo argumental el tema deportivo. Se basa en deportes reales o ficticios, y se subdividen en simuladores o de menor realismo que un simulador. [1]

Para explicar la gestión de los perfiles de usuarios en este género se toman de ejemplo los videojuegos de béisbol “MVP 08” y de fútbol “FIFA 08”. En ellos la estrategia de salva es similar. Al iniciar, el jugador escoge el nombre que tendrá su perfil, así como la liga y el equipo con que jugará. Luego puede hacer todo tipo de configuraciones en cuanto a la alineación del equipo, los controles, nivel de dificultad del juego, así como otros elementos relacionados con el nivel de detalle de los gráficos, el volumen del juego o la sensibilidad del dispositivo de retroalimentación de fuerza. A medida que el jugador obtiene resultados favorables, acumula puntos que le permiten desbloquear terrenos, trajes, trofeos o jugadores legendarios, a los cuales no tiene acceso inicialmente. El usuario tiene la posibilidad de salvar estos logros mediante uno de los botones del menú. [15]

Casi todas las generaciones de juegos deportivos usan la estrategia de salvar los perfiles de usuario en ficheros que se generan en algún directorio de fácil acceso, en esta localización se crea una carpeta con

el nombre del juego, que dentro contiene otra por cada perfil con sus respectivos ficheros de configuración. Si se desea jugar en otra computadora y mantener los mismos avances, tan solo debe copiarse esta carpeta en el mismo directorio de la otra computadora.

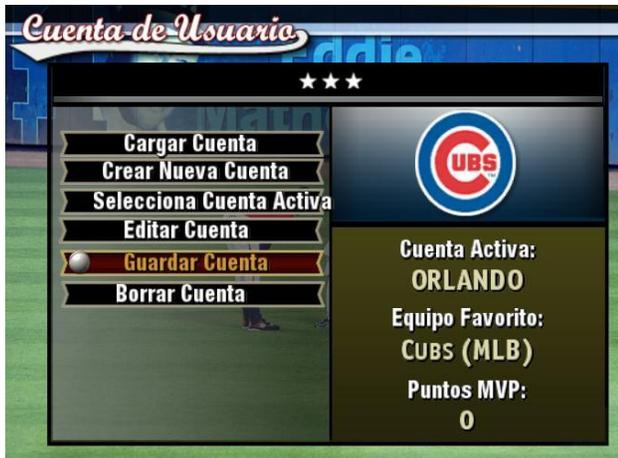


Figura 1.4 Vista de la salva en el videojuego "MVP08".



Figura 1.5 Vista de la salva en el videojuego "FIFA08".

1.5.4 Videojuegos de Carreras

El objetivo de este tipo de videojuegos es recrear una sensación de velocidad al jugador. Se pueden conducir todo tipo de vehículos, desde coches hasta aviones, pasando por lanchas motoras o naves espaciales. El objetivo del jugador es llegar el primero en el menor tiempo posible a la meta. [1]

En el videojuego *"Need for Speed Underground"* que forma parte de este género, para realizar la salva de los perfiles de usuario se usa la misma idea que los videojuegos deportivos explicados anteriormente. Cuando el usuario comienza a jugar debe elegir el nombre del perfil y el auto. En las diferentes modalidades de carreras que existen, el jugador va obteniendo dinero, con el cual puede desbloquear nuevas pistas, carros y piezas para mejorarlos. [16]

También puede configurar los controles de teclado, nivel de dificultad del juego, nivel de detalle de los gráficos, el volumen del ambiente del juego o la sensibilidad del dispositivo de retroalimentación de fuerza.

Todos estos cambios son salvados en un fichero que se genera en la dirección que tiene predefinida el juego; un directorio de fácil acceso para el usuario.

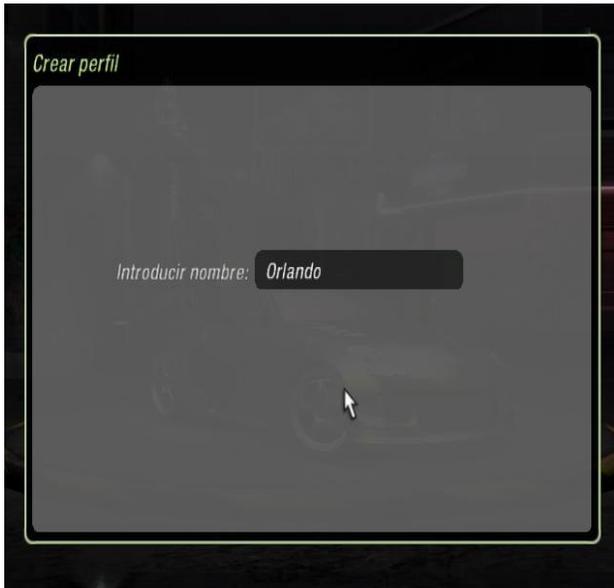


Figura 1.6 Vista 1 de la salva en el videojuego “NFSU”.

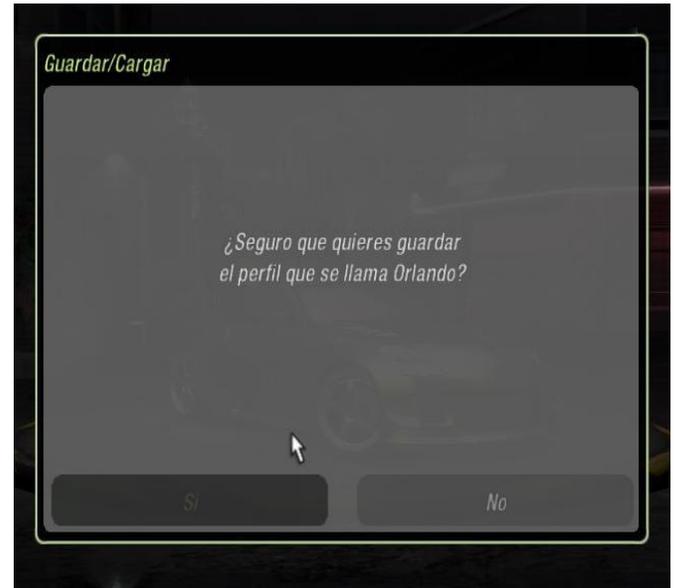


Figura 1.7 Vista 2 de la salva en el videojuego “NFSU”.

1.6 Experiencias similares en proyectos del polo de Realidad Virtual de la Facultad 5

Para concretar una investigación y hacer nuevos aportes hay que tener en cuenta los trabajos realizados con anterioridad relacionados con la gestión de los perfiles de usuario. Al investigar sobre este tema se pudo conocer de la existencia de una tesis realizada en la Universidad de las Ciencias Informáticas, que aunque no está exactamente relacionada con los videojuegos, acumula algunas experiencias que pueden tomarse de punto de partida.

Se trata del “Módulo Registro de Evaluación para el Simulador Quirúrgico” realizada en el 2008, en la cual se realizó la investigación e implementación de un módulo para el registro de la evaluación de cada médico. Las pruebas realizadas por los cirujanos en el simulador se guardan en una base de datos con toda la información correspondiente al ejercicio para mantenerla organizada . El simulador brinda además reportes de los resultados obtenidos por los médicos examinados.

Para el desarrollo de ese trabajo se empleó como medio de almacenamiento de la información el gestor de Base de Datos Firebird, encargado de todo el trabajo con el volumen de información registrado por los médicos que pasaron el examen. El módulo contiene datos personales de los médicos, brinda la posibilidad de llevar un control de los valores de todas las variables de cada prueba, permitiendo la gestión de la información de los resultados alcanzados por cada usuario.

Para lograr la correcta automatización de todo el proceso de almacenamiento de datos fue necesario procesar los siguientes datos personales de los médicos cirujanos que realizan el ejercicio: nombre, apellidos, usuario, contraseña, dirección particular, teléfono, centro de trabajo, años de experiencia.

Estos parámetros permiten a los profesores o especialistas encargados de la evaluación acceder a los reportes de toda la actividad relacionada con el ejercicio, facilitando el trabajo y la rapidez en la evaluación. [23]

1.7 Metodología y herramientas de desarrollo

Para la realización del módulo para la gestión de los perfiles de usuario existe un gran número de herramientas de desarrollo. Para seleccionarlas se tuvieron en cuenta los requisitos no funcionales del proyecto al cual tributa el trabajo.

Las metodologías son el conjunto de procedimientos, técnicas, herramientas y un soporte documental que ayuda a los desarrolladores a realizar nuevo software. No existe una metodología de software universal. Las características de cada proyecto o equipo de desarrollo exigen que el proceso sea configurable.

- *Proceso Unificado de Modelado (RUP)*

La metodología usada es RUP. Es un proceso de desarrollo de software y junto con el Lenguaje Unificado de Modelado (UML), constituye la metodología estándar más utilizada para el análisis, implementación y documentación de sistemas orientados a objetos. RUP no es un sistema con pasos firmemente establecidos, sino un conjunto de metodologías adaptables al contexto y necesidades de cada organización. Esta metodología es más apropiada para proyectos grandes, dado que requiere un equipo de trabajo capaz de administrar un proceso complejo en varias etapas, guiando a cada trabajador durante el ciclo de vida completo del desarrollo de software: análisis y diseño orientados a objetos, construcción, pruebas y despliegue.

- *Visual Studio 2005*

En el presente trabajo se usa el lenguaje de programación C++ y la herramienta de desarrollo Visual Studio 2005 ya que son requisitos no funcionales establecidos por el proyecto, además se puede decir que el lenguaje es versátil, portable y multi-paradigma.

El Visual Studio 2005, es una herramienta poderosa, fuerte y voluminosa la cual permite la integración de varios lenguajes, entre ellos el C++, C#, y Asp.Net. El Visual Studio 2005 también ofrece la posibilidad de implementar aplicaciones para soluciones integrales que aprovechen de manera óptima la ventaja de cada lenguaje, tiene una interfaz amigable y proporciona una gran cantidad de herramientas que ofrecen ventajas a los desarrolladores individuales y equipos de desarrollo de software.

Visual C++ ha sido durante bastante tiempo un conjunto básico de herramientas para los programadores de Windows. Otros lenguajes permiten crear prototipos y desarrollar aplicaciones de forma más rápida, pero nada puede sustituir a Visual C++ a la hora de realizar tareas de desarrollo a bajo nivel. [8]

- *Motor gráfico G3D*

G3D (del inglés Graphics Three Dimensional) es un motor gráfico clasificado como código abierto y libre, fue desarrollado con el lenguaje C++. Es usado en la actualidad en la realización de videojuegos, simuladores, entre otros. Cuenta con un conjunto de fuertes y definidas funcionalidades para el trabajo con aplicaciones con implicaciones gráficas.

En el caso del presente trabajo se tuvo en cuenta además la utilidad de G3D de poseer facilidades de propósito general como las clases para la creación y trabajo con las interfaces, así como las clases para realizar la salva de la información en formato binario y además es el motor gráfico usado actualmente en el proyecto.

- *Rational Rose*

Es una buena solución de modelado visual y una buena herramienta para traducir requisitos de alto nivel a una arquitectura flexible basada en componentes. Rational se encuentra a la cabeza del desarrollo del Lenguaje Unificado de Modelado (UML), que se ha convertido en la notación estandarizada empleada en Rational Rose para especificar, visualizar y construir desarrollos de software y sistemas. Rational Rose domina el mercado de herramientas para el análisis, diseño, modelado y construcción orientada a objetos; tiene todas las características que los desarrolladores, analistas, y arquitectos exigen: soporte UML, incomparable desarrollo basado en componentes con soporte para arquitecturas líderes en la industria y

modelos de componentes, facilidad de uso e integración optimizada. Rose es una herramienta con plataforma independiente que ayuda a la comunicación entre los miembros del equipo, a monitorear el tiempo de desarrollo y a entender el entorno de los sistemas. Una de las grandes ventajas de Rose es que utiliza la notación estándar en la arquitectura de software (UML), la cual permite a los arquitectos de software y desarrolladores visualizar el sistema completo utilizando un lenguaje común. Además, los diseñadores pueden modelar sus componentes e interfaces en forma individual y luego unirlos con otros componentes del proyecto. [26]

Conclusiones

En este capítulo se le dio cumplimiento a las dos primeras tareas trazadas en el presente trabajo, las cuales sirven de soporte teórico para la investigación. Se abordó en el estado del arte las principales formas de salvar información en ficheros así como los diferentes videojuegos que realizan la salva de los perfiles de usuario. Además fueron seleccionadas las herramientas y metodología de desarrollo a seguir.

CAPÍTULO 2: CARACTERÍSTICAS DEL SISTEMA

En este capítulo, considerando la necesidad de crear un módulo que gestione los perfiles de usuario en el videojuego “Rápido y Curioso” se define la forma en que se le da solución al problema, para ello se hizo un estudio para seleccionar los datos a salvar de cada jugador, además de seleccionar y diseñar la estructura del fichero que se utilizará para exportar e importar los perfiles de usuario. Se realiza el modelo del dominio, la captura de los requisitos funcionales y no funcionales, sin dejar de explicar y describir todos los casos de uso que en su definición reflejan exactamente cómo responderá el sistema ante la acción de los usuarios y además las funcionalidades del mismo. Además se explican las funcionalidades de las clases BinaryInput, BinaryOutput, TextInput y TextOutput que brinda la biblioteca G3D, las cuales son usadas para el trabajo con los ficheros.

2.1 Necesidad de la creación del módulo para la gestión de los perfiles de usuarios en el videojuego “Rápido y Curioso”.

- Antes de la creación del módulo, los ficheros de configuración del juego “*GameSettings.txt*” y “*NetSettings.txt*” debían ser configurados de forma manual por el administrador del sistema, lo que era bastante incomodo de realizar. Ahora cuentan con la facilidad de hacerlo mediante una interfaz diseñada para ello.
- Con la creación de la interfaz para la configuración de los controles, el jugador tendrá la oportunidad de establecer los controles a su gusto y no tiene que usar necesariamente los predefinidos por el juego.
- Si en algún momento el cliente del producto desea cambiar el tiempo de duración de la carrera o la cantidad de vueltas para dar cumplimiento a una carrera, está creada e implementada la interfaz que permite realizar dichos cambios.
- Con la creación del módulo se le da la posibilidad a los usuarios de tener un perfil que los identifique, con sus propias opciones de juego, que podrán cargar cada vez que inicie el juego, sin tener que realizarlas nuevamente.

2.2 Solución técnica

La idea principal a desarrollar es la implementación de un módulo que se encarga de la gestión de los perfiles de usuario, para la cual se usan las clases que brinda el motor gráfico G3D para el trabajo con ficheros binarios y de texto, además de algunas clases auxiliares que se implementan para dar solución al problema.

Se hace uso de las formas más populares usadas en los distintos tipos de videojuegos descritos en el Capítulo 1 con el fin de crear, para la gestión de los perfiles, interfaces amigables, fáciles de entender y de una correcta estética y claridad a la hora de usarlas.

Con la inserción en el juego de este módulo cambiará de alguna forma el diseño actual del mismo en la aplicación que le da inicio, ya que se le agregan dos botones nuevos, el botón Configuración y el botón Perfiles. En el primero se podrán configurar todas las opciones relacionadas con la resolución, volumen,

controles de teclado, opciones de configuración de red y opciones de juego. En el segundo se podrán crear, cargar y eliminar los perfiles de usuario.

Entre las funcionalidades que se brindan se encuentran la de permitir al jugador crearse un perfil que lo identifique, posibilitando que los cambios realizados en cuanto a las configuraciones de las opciones de juego puedan ser diferenciados por jugador. Además podrá cargar los perfiles existentes y eliminarlos si lo desea. En cuanto a la gestión de estos perfiles se brindan una gran variedad de opciones que el jugador puede configurar, algunas relacionadas con el ambiente del juego y otras con características particulares de preferencias de los usuarios.

Todos los ficheros que se vayan generando después de crear un nuevo perfil o realizar configuraciones, se ubican en la carpeta raíz del juego, dentro de otra carpeta que recibe el nombre de “data”, la cual contendrá otra denominada “Jugadores”; ahí estarán todos los ficheros correspondientes a los jugadores que usen el videojuego.

2.3 Descripción de las funcionalidades por interfaz

Debido a la cantidad de opciones que se le brindan al jugador para configurar se hace necesario crear varias interfaces para este trabajo, ya que sería de poca estética ofrecer solo una pantalla para modificarlas todas juntas. A continuación se describen cada una de las interfaces creadas.

Crear Perfil

En esta interfaz el jugador puede crearse un perfil, cuenta con un área donde se escribe el nombre que recibirá el perfil, después de escribir el nombre el jugador debe presionar el botón *Aceptar* para completar la acción de creación del perfil.

Cargar Perfil

Interfaz que cuenta con una lista de los perfiles creados en el juego, brindándole la posibilidad al jugador de seleccionar el perfil que desea sea el activo para jugar. Después de seleccionar el perfil que desea activar el jugador debe presionar el botón *Aceptar* para completar la activación del perfil.

Eliminar Perfil

Muestra una lista con todos los perfiles existente en el juego. El jugador selecciona el que desea eliminar, se muestra una alerta para confirmar la eliminación y si realmente lo desea el jugador presiona el botón *SI* para eliminar completamente el perfil, en caso contrario presiona el botón *NO*.

Opciones Visuales

En esta interfaz el jugador puede cambiar el modo de pantalla, cuenta con dos opciones, la cual selecciona, puede ser *Full Screen* y *Modo Normal*, después de seleccionar la opción que desea debe presionar el botón *Aceptar* para hacer efectivo el cambio.

Opciones de Red

En esta interfaz el jugador puede cambiar algunos parámetros de red entre los que se encuentran la dirección del servidor, el puerto y el tiempo de espera del servidor, estos cambios son posibles mediante la edición de tres campos de textos con que cuenta la interfaz, para hacer efectivos los cambios debe presionar el botón *Aceptar*. En caso de que el jugador desee las opciones por defecto debe presionar el botón *Defecto* y luego el botón *Aceptar*.

Controles de teclado

Los controles que están establecidos en el juego son modificables, en esta interfaz el jugador puede cambiarlos, cuenta con trece campos de texto, modificar un control de teclado implica borrar la opción que estaba anteriormente y escribir la nueva que desea, después de hacer toda la edición en los controles debe presionar el botón *Aceptar* para hacer validos los cambios. En caso de haberse equivocado en algún control y desee tener nuevamente los controles por defecto, debe presionar el botón *Defecto* y luego el botón *Aceptar*.

Opciones de juego

En esta interfaz se muestra tres opciones de juego, pero el usuario solo podrá cambiar una de estas que es la de cantidad de jugadores, las otras dos están solo de manera informativa y no son configurables ya que son peticiones directas del cliente. Para modificar la opción de cantidad de jugadores debe borrar la anterior usando la tecla *Delete* y escribiendo nuevamente la cantidad que desea, para terminar y hacer efectivo el cambio debe presionar el botón *Aceptar*, en caso de desear la opción por defecto debe presionar el botón *Defecto* y luego el botón *Aceptar*.

2.4 Datos persistentes

Después que el usuario gestiona su perfil, se hace necesario almacenar todos estos datos en un fichero, con el objetivo de que pueda ser reutilizado para que cuando inicie nuevamente la aplicación tenga todas las configuraciones tal y como las dejó en la partida anterior. Los ficheros correspondientes a cada jugador son creados con las clases para el tratamiento de ficheros binarios que provee G3D como biblioteca, estas clases son *BinaryInput* y *BinaryOutput*, las mismas brindan un conjunto eficiente de funcionalidades que permiten hacer las más diversas operaciones.

En el caso del videojuego *Rápido y Curioso*, se decidió que el tipo de formato utilizado para salvar la información de cada usuario fuese de tipo binario, esto influye en la rapidez de lectura del sistema ya que no tiene que hacer conversión de datos, además de aumentar la confidencialidad de los mismos. Además, el uso de formato binario es una de las especificaciones del problema que fue planteado en la tesis por el proyecto “*Rápido y Curioso*”.

2.4.1 Selección de los datos que se salvan del jugador

Después de realizar una investigación detallada de todos los datos que se necesitaban salvar, se definieron veinticuatro tipos de elementos a salvar. Así se llegó a la conclusión de que se necesitan cinco entidades contenedoras de datos, las cuales se diferencian en cuanto a la opción de juego que desee cambiar el jugador, las cuales se diferencian según el tipo de información que contengan.

Un perfil completo de un jugador contiene los veinticuatro elementos y se usa la clase Settings para agruparlos. Después que se crea un perfil, el usuario tiene la facilidad de configurar las opciones del juego, y esto no lo hace a la misma vez; cuando desea modificar las opciones de los controles de teclado la clase que contiene los datos es ControllerSettings, para configurar las opciones de red la clase usada es NetworkSettings, para las opciones de modo de pantalla la clase es VisualSettings, para modificar las opciones de audio se usa la clase AudioSettings y finalmente para otras opciones del juego específicas se usa la clase GameplayOptions.

Vale resaltar que para las opciones de juego número de vueltas y tiempo de carrera se ha creado la posibilidad que si en algún momento el cliente decide que estos valores varíen, pueden ser modificados. De momento estas opciones están bloqueadas y no reciben cambios, solo están de manera informativa.

2.4.2 Especificación de la estructura interna del fichero

La estructura de un fichero que contiene la información de un usuario será la siguiente: antes de cada valor guardado estará la descripción del mismo, luego aparece el nombre que recibe el parámetro y seguidamente el valor correspondiente para que el programador tenga claro qué va a leer.

Los tipos de datos encontrados en el fichero son datos de tipo entero (int) y datos de tipo cadena de caracteres (string). A continuación se mencionan todos los parámetros que son salvados en la siguiente tabla:

Elementos del fichero

Tipo de dato	Descripción	Nombre en el fichero
string	Nombre que identifica el perfil del jugador.	profile_name
int	Modelo de auto preferido por el jugador.	car_model
int	Color de auto preferido por el jugador.	car_color

int	Cantidad de jugadores para el modo multijugador.	players_count
int	Número de vueltas en una carrera.	laps_count
int	Tiempo máximo de duración de la carrera.	time_limit
int	Modo de pantalla predeterminada por el usuario.	screen_mode
int	Volumen del ambiente del juego.	environment_volume
string	Tecla escogida por el usuario para acelerar.	accelerate
string	Tecla escogida por el usuario para frenar.	brake
string	Tecla escogida por el usuario para girar a la izquierda.	turn_left
string	Tecla escogida por el usuario para moverse a la derecha.	turn_rigth
string	Tecla escogida por el usuario para poner el auto en marcha adelante.	change_gear_forward
string	Tecla escogida por el usuario para poner el auto en marcha atrás.	change gear_backward
string	Tecla escogida por el usuario	claxon

	para sonar el claxon del auto.	
string	Tecla escogida por el usuario para reproducir la música.	play_music
string	Tecla escogida por el usuario para sacar el espejo retrovisor.	actívate_mirrors
string	Tecla escogida por el usuario para cambiar la cámara.	camera
string	Tecla escogida por el usuario para posicionar el auto correctamente después de chocar.	restore_car
string	Tecla escogida por el usuario para mostrar el mapa de la carrera.	actívate_GPS
string	Tecla escogida por el usuario para salir de la carrera.	leave_game
string	Dirección IP del servidor para el modo multijugador.	server_IP
int	Puerto usado para la conexión entre el cliente y el servidor.	connection_port
int	Tiempo que espera el servidor para la conexión de todos los jugadores en el modo multijugador. Después del cual no acepta más	server_waiting_time

	conexiones.	
--	-------------	--

Es importante señalar que los parámetros *server_IP*, *players_count* y *server_waiting_time*, además de guardarse en el fichero destinado para el usuario en cuestión, modifica parámetros de igual nombre que ya están guardados en el fichero *GameSettings.txt* y el parámetro *connection_port*, se guarda en el fichero del jugador y además en el fichero *NetSettings.txt*.

Para el trabajo con los ficheros en formato binario la biblioteca G3D ofrece las clases *BinaryInput* y *BinaryOutput* mediante las cuales se puede hacer un buen trabajo con este tipo de formato. A continuación se describirán dichas clases explicando sus ventajas.

Formato binario de las clases *BinaryInput* y *BinaryOutput*

La clase *BinaryOutput* es la encargada de crear los perfiles, cuenta con métodos que hacen fácil el trabajo con todos los tipos de datos, por ejemplo: el método *writeString (valor)* es usado para escribir un dato de tipo cadena de caracteres (string) en el fichero, el método *writeUInt16 (valor)* es usado para escribir un dato de tipo entero (int) en el fichero, entre otros tan importantes como el método *commit ()* que es el encargado de crear físicamente el fichero.

La clase *BinaryInput* es usada para cargar ficheros ya existentes, cuenta con métodos muy sencillos de usar para leer todos los parámetros almacenados en un fichero, por ejemplo: el método *readString ()* se usa para leer un dato de tipo cadena de caracteres (string) del fichero, el método *readUInt16 ()* se usa para leer un dato de tipo entero (int) del fichero. Debido a la sencillez en el uso de estas clases, además de ser eficientes para resolver el problema planteado en el presente trabajo, se decidió que fueran las candidatas perfectas a usar.

Formato de texto de las clases *TextInput* y *TextOutput*

Los ficheros "*GameSettings.txt*" y "*NetSettings.txt*" contienen información relacionada con elementos de configuración del juego y de la red. Muchos de estos elementos serán configurados por los usuarios gracias a las nuevas funcionalidades que ofrece el módulo de gestión de sus perfiles. Por lo tanto es

necesario actualizarlos cuando el usuario hace una modificación en sus valores ya que el juego los usa directamente.

Los datos (*server_IP*, *players_count*, *server_waiting_time* y *connection_port*) se salvan en el fichero de cada jugador y además en los dos ficheros mencionados anteriormente porque estos datos cuando los jugadores lo configuran, son particulares para cada jugador, de ahí la necesidad de tenerlo en ambos ficheros y garantizar que cuando se cambie de jugador activo se use la información correcta.

Para realizar la operación de actualización de los ficheros *GameSettings.txt* y *NetSettings.txt* se usa la clase *TextInput* encargada de leer los valores que contienen, mediante los métodos *readSymbol ()* y *readNumber ()*, después de leerlos se actualizan haciendo uso de la clase *TextOutput* la cual tiene los métodos *writeString (valor)*, *writeNumber (valor)* y *writeSymbol (valor)*, los cuales están implementados para escribir en el fichero el tipo de dato correspondiente a cada uno, luego de escribir todos los datos necesarios se usa el método *commit ()* para crear físicamente el fichero.

2.5 Apariencia del fichero

Esta vista del fichero es para tener una idea de cómo queda conformado ya que después de ser guardado en formato binario no es entendible. (Figura 2.1)

```

/*Nombre con el cual se identificara el jugador*/
profile_name "nombre"
/*Modelo preferido de auto*/
car_model 1
/*Color preferido de auto*/
car_color 2
/*Cantidad de jugadores que se agregaran al juego*/
Players_count 4
/*Cantidad de vueltas a la pista*/
laps_count 3
/*Tiempo de duracion de la carrera*/
time_limit 8
/*Resolucion de la pantalla*/
screen_mode 1
/*Volumen del juego*/
Environment_volume 9
/*Tecla para acelerar*/
accelerate "w"
/*Tecla para frenar*/
brake "s"
/*Tecla para moverse a la izquierda*/
turn_left "a"
/*Tecla para moverse a la derecha*/
turn_rigth "d"
/*Tecla para marcha adelante*/
change_gear_forward "q"
/*Tecla para marcha atras*/
Change_gear_backward "z"
/*Tecla para claxon*/
claxon "t"
/*Tecla para reproducir la música*/
play_music "m"
/*Tecla para espejo*/
activate_mirror "e"
/*Tecla para camara*/
camera "c"
/*Tecla para reposicionar*/
restore_car "r"
/*Tecla para GPS*/
activate_gps "g"
/*Tecla para abandonar partida*/
leave_game "esc"
/*Servidor que usara el juego para la partida en red*/
Server_IP "10.5.2.228"
/*Puerto usado por el juego para la comunicacion cliente
servidor*/
connection port 1425

```

Figura 21 Apariencia del fichero.

2.6 Análisis del Dominio

El objetivo del modelo de dominio es detallar y describir el problema de manera entendible a los desarrolladores, además constituye la base para diseñar y crear entidades del problema a resolver.

2.6.1 Modelo del Dominio

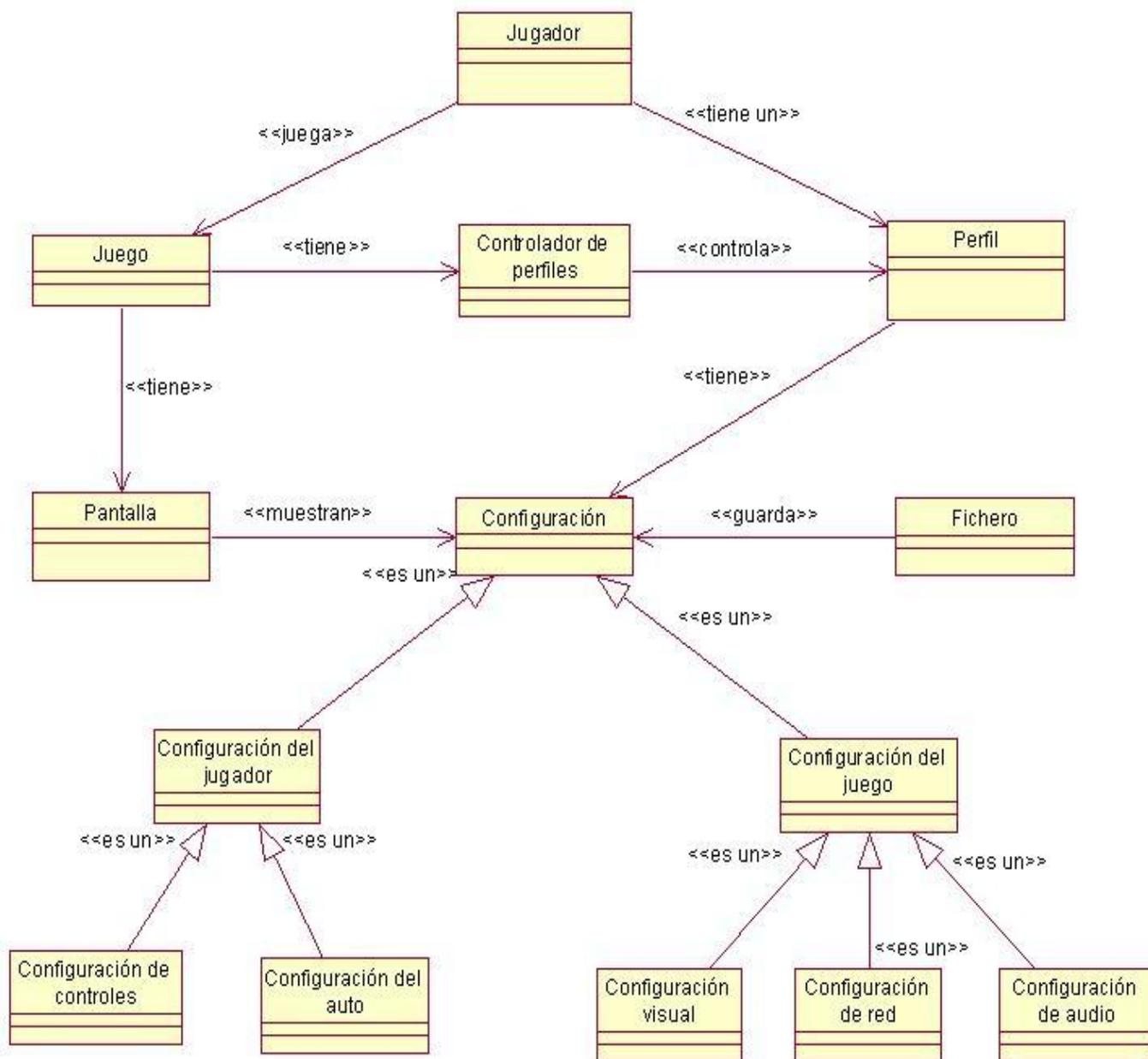


Figura 2.2 Modelo de dominio.

2.6.2 Conceptos del modelo de dominio

Jugador: Es quien juega el juego.

Juego: Videojuego "Rápido y Curioso".

Perfil: Identifica a cada jugador.

Controlador de perfiles: Es el encargado de gestionar todos los perfiles.

Pantalla: Muestra la configuración de cada jugador.

Fichero: Encargado de guardar todas las configuraciones.

Configuración: Cambios que puede realizar el jugador con respecto a su perfil.

Configuración del juego: Configuraciones que contienen las opciones de juego.

Configuración del jugador: Configuraciones que contienen las opciones del jugador.

Configuración de controles: Configuraciones que contienen las opciones de los controles de teclado.

Configuración del auto: Configuraciones que contienen las opciones del auto.

Configuración visual: Configuraciones que contienen las opciones visuales.

Configuración de red: Configuraciones que contienen las opciones de red.

Configuración de audio: Configuraciones que contienen las opciones de audio.

2.7 Requisitos del software

Se puede decir que un requisito traduce las necesidades del modelo de negocio a requisitos de sistemas automatizables y con carácter más técnico (casos de uso con UML), persiguen obtener un entendimiento más profundo del modelo de negocio por parte de los integrantes del equipo de desarrollo.

Es una condición o capacidad que tiene que ser alcanzada o poseída por un sistema o componente de un sistema para satisfacer un contrato, estándar, u otro documento impuesto formalmente. [11]

2.7.1 Requisitos funcionales

RF1. Crear perfil de usuario.

RF2. Editar perfil de usuario.

RF2.1. Modificar el nombre del jugador.

RF2.2. Modificar cantidad de jugadores.

RF2.3. Modificar resolución de la pantalla.

RF2.4. Especificar el IP del servidor.

RF2.5. Modificar tiempo de espera del servidor.

RF2.6. Modificar el puerto de conexión.

RF2.7. Especificar el modelo del auto preferido por el usuario.

RF2.8. Especificar el color del modelo de auto preferido por el usuario.

RF2.9. Modificar los controles de teclado y de joystick.

RF3. Cargar perfil de usuario.

RF4. Eliminar perfil de usuario.

2.7.2 Requisitos no funcionales

RNF1. Requerimientos de Software:

Sistema Operativo: Windows XP, Windows 2003, o versiones superiores de Windows.

RNF2. Requerimientos Hardware:

Recomendado:

Procesador Intel P4+ 2.0Ghz, 256Mb RAM

Video Card NVIDIA 128MB o Video Card ATI 256MB

Herramientas de desarrollo:

IDE: Visual Studio .NET 2005

CASE: Rational Rose 2003 Enterprise Edition

Management: Visual Source Safe 6.0

Otras: Microsoft Office 2003, Adobe Photoshop CS3

Lenguajes de programación: C++

RNF3. Tiempo de respuesta:

El tiempo que demora el juego para gestionar las peticiones del usuario debe ser lo más rápido posible, debe estar en el orden de los milisegundos.

2.8 Modelo de casos de usos del sistema

A continuación se define el actor del sistema y se crean los casos de uso del sistema a partir de los requisitos funcionales. Además, se hace una descripción textual de los casos de uso del sistema.

2.8.1 Actor del sistema

Actores	Justificación
Jugador	Este es quien utiliza el juego propuesto para divertirse e instruirse

Tabla # 1 Justificación del actor

2.8.2 Casos de uso del Sistema

CU-1	Crear perfil de usuario
Actor	Jugador
Descripción	El caso de uso se inicia cuando el jugador crea su perfil y finaliza cuando este se encuentra creado.
Referencia	RF1

Tabla # 2 CU Crear perfil de usuario

CU-2	Editar perfil
Actor	Jugador
Descripción	El caso de uso se inicia cuando el jugador desea realizar cualquier cambio en el perfil y finaliza cuando esta operación ha sido llevada a cabo.

Referencia	RF2,RF2.1,RF2.2,RF2.3,RF2.4,RF2.5,RF2.6,RF2.7,RF2.8,RF2.9
-------------------	---

Tabla # 3 CU Editar perfil

CU-3	Cargar perfil
Actor	Jugador
Descripción	El caso de uso se inicia cuando el jugador desea cargar los datos de su perfil y finaliza cuando esta acción ha sido llevada a cabo.
Referencia	RF3

Tabla # 4 CU Salvar perfil

CU-4	Eliminar perfil
Actor	Jugador
Descripción	El caso de uso es iniciado cuando el jugador desea eliminar su perfil, es decir, borrar todos sus datos, y se termina cuando dicha tarea fue realizada.
Referencia	RF4

Tabla # 5 CU Eliminar perfil

2.8.3 Diagrama de Casos de Uso del sistema

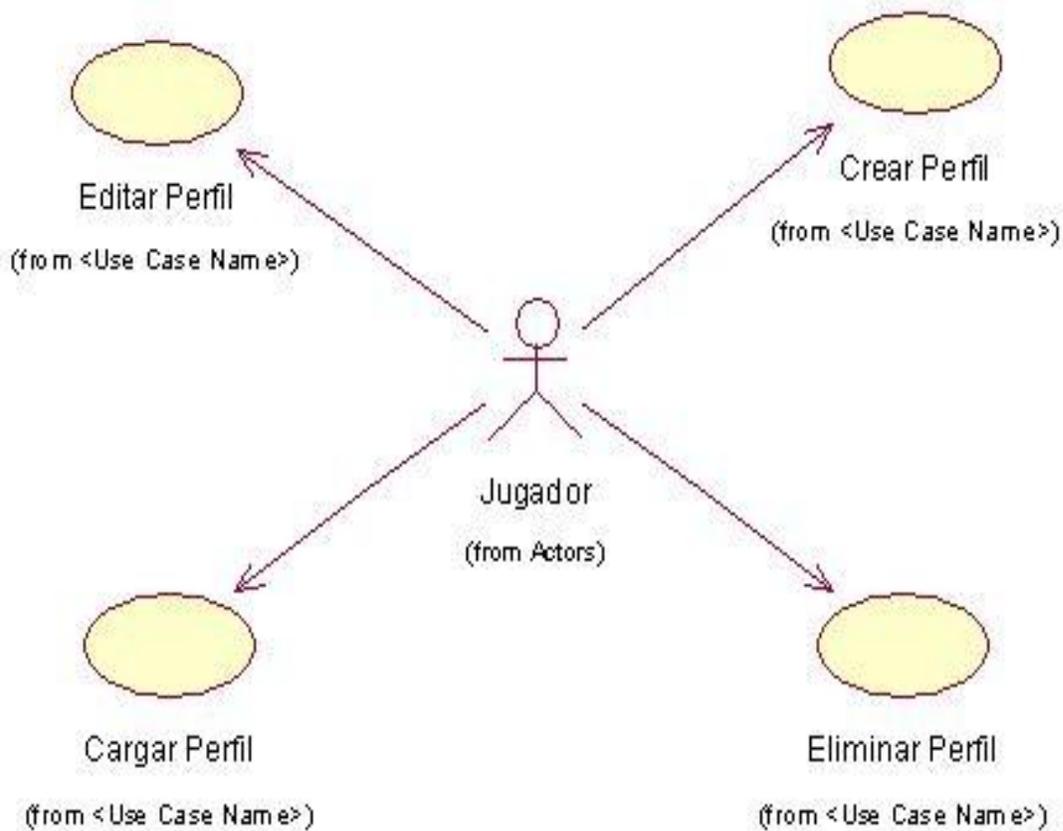


Figura 2.3 Diagrama de Casos de Uso del sistema.

2.8.4 Expansión de los casos de uso del sistema

Caso de Uso Crear Perfil

Nombre del CU	Crear Perfil.	
Actor	Jugador (Inicia).	
Propósito	Permitir al jugador crear su perfil para comenzar a jugar.	
Resumen	El caso de uso se inicia cuando el jugador crea su perfil y finaliza cuando este se encuentra creado.	
Referencias	RF1	
Curso Normal de Eventos		
Acciones del Actor	Respuesta del Sistema	
<ol style="list-style-type: none"> 1. El jugador inicia el juego. 2. El jugador escoge la opción “Jugadores”. 3. El jugador escoge la opción “Crear”. 4. El jugador escribe el nombre que desea y presiona el botón crear. 	<ol style="list-style-type: none"> 1.1 El sistema muestra una interfaz con las opciones iniciales que brinda el juego. 2.1 El sistema muestra la interfaz “Opciones Jugador”. 3.1 El sistema muestra la interfaz “Crear Jugador”. 4.1 El sistema crea un fichero donde se salvan los datos del perfil creado y lo ubica como usuario activo, con el cual se inicia el juego. 	
Flujo Alternativo		
<ol style="list-style-type: none"> 3. El jugador escribe el nombre que desea y presiona el botón crear. 3. a El jugador escoge la opción atrás. 	<ol style="list-style-type: none"> 3.1.1 El sistema muestra un cartel diciendo que el usuario existe y brinda la posibilidad de crear un nuevo usuario. 3.1.2 El sistema no crea ningún perfil y regresa a la interfaz principal. 	
Poscondiciones	Se efectuó la creación del perfil de usuario con todos los datos requeridos	

Tabla # 6 Expansión del CU Crear perfil

Caso de Uso Editar Perfil

Nombre del CU	Editar Perfil.
Actor	Jugador (Inicia).
Propósito	Permitir al jugador modificar su perfil de la forma que estime conveniente.
Resumen	El caso de uso se inicia cuando el jugador desea modificar su perfil realizando los cambios correspondientes y finaliza cuando esta acción es realizada
Referencias	RF2, RF2.1, RF2.2, RF2.3, RF2.4, RF2.5, RF2.6, RF2.7, RF2.8, RF2.9
Curso Normal de Eventos	
Acciones del Actor	Respuesta del Sistema
<ol style="list-style-type: none"> 1. El jugador inicia el juego 2. El jugador escoge la opción "Configuración" 3. El jugador escoge la opción que desea configurar. 4. El jugador realiza los cambios y presiona el botón aceptar 	<ol style="list-style-type: none"> 1.1 El sistema muestra una interfaz con las opciones iniciales que brinda el juego. 2.1 El sistema muestra la interfaz "Panel de Configuración". 3.1 El sistema muestra la interfaz de la opción escogida por el usuario para que el mismo realice los cambios. 4.1 El sistema guarda los cambios realizados en el fichero.
Flujo Alternativo	

<p>2 El jugador escoge la opción “Configuración”.</p> <p>4. a. El jugador presiona botón atrás</p>	<p>2.1.1 El sistema muestra un cartel diciendo que está activo el perfil por defecto, y para poder configurar debe crear un nuevo perfil.</p> <p>4.1.1 El sistema descarta los cambios realizados, se mantienen las opciones iniciales y regresa a la interfaz “Panel de Configuración”.</p>
<p>Poscondiciones</p>	<p>Se realizaron las modificaciones deseadas por el jugador.</p>

Tabla # 7 Expansión del CU Editar perfil

Caso de Uso Cargar Perfil

<p>Nombre del CU</p>	<p>Cargar Perfil.</p>
<p>Actor</p>	<p>Jugador (Inicia).</p>
<p>Propósito</p>	<p>Permitir al jugador cargar el perfil con el que desea jugar.</p>
<p>Resumen</p>	<p>El caso de uso es iniciado cuando el jugador determine cargar su perfil con toda la información almacenada en el mismo y finaliza cuando esta acción es realizada.</p>
<p>Referencias</p>	<p>RF5</p>
<p style="text-align: center;">Curso Normal de Eventos</p>	
<p>Acciones del Actor</p>	<p>Respuesta del Sistema</p>
<p>1. El jugador inicia el juego.</p> <p>2. El jugador escoge la opción “Jugadores”.</p> <p>3. El jugador escoge la opción “Perfil Activo”.</p> <p>4. El jugador selecciona el perfil que desea activar.</p>	<p>1.1 El sistema muestra una interfaz con las opciones iniciales del juego.</p> <p>2.1 El sistema muestra la interfaz “Opciones Jugador”.</p> <p>3.1 El sistema muestra una interfaz indicando el perfil que está activo actualmente y carga sus valores.</p>

	4.1 El sistema cambia a activo el perfil seleccionado por el usuario y carga sus valores.
Flujo Alternativo	
4. a. El jugador presiona botón atrás	3.1.1 El sistema deja el perfil que estaba seleccionado como activo y regresa a la interfaz “Opciones Jugador”.
Poscondiciones	Se cargó el perfil de usuario.

Tabla # 8 Expansión del CU Cargar perfil

Caso de Uso Eliminar Perfil

Nombre del CU	Eliminar Perfil.
Actor	Jugador (Inicia).
Propósito	Permitir al usuario eliminar su perfil.
Resumen	El caso de uso es iniciado cuando el jugador determine eliminar su perfil perdiendo así toda la información almacenada y finaliza cuando esta acción es realizada.
Referencias	RF1, RF1.1, RF1.2
Curso Normal de Eventos	
Acciones del Actor	Respuesta del Sistema
1. El jugador inicia el juego	1.1 El sistema muestra una interfaz con las opciones iniciales del juego.
2. El jugador selecciona la opción “Jugadores”.	
3. El jugador selecciona opción “Eliminar”.	2.1 El sistema muestra la interfaz “Opciones Jugador”.
4. El jugador selecciona el perfil que desea eliminar y presiona el botón	3.1 El sistema muestra una interfaz con los perfiles

<p>aceptar. 5. El jugador presiona el botón "SI".</p>	<p>creados. 4.1 El sistema pide la confirmación para realizar esta acción. 5.1 El sistema borra el perfil seleccionado por el usuario eliminando el fichero con todo su contenido.</p>
<p>Flujo Alternativo</p>	
<p>4.a El jugador presiona botón atrás 5.a El jugador presiona el botón "NO".</p>	<p>4.1.1 El sistema no elimina ningún perfil y te envía a la interfaz "Opciones Jugador". 5.1.1 El sistema no elimina ningún perfil y te envía a la interfaz "Eliminar Jugador".</p>
<p>Poscondiciones</p>	<p>Se eliminó el perfil de usuario con todo su contenido.</p>

Tabla # 9 Expansión del CU Eliminar perfil

2.9 Análisis

El análisis consiste en obtener una visión del sistema que se preocupa de ver qué hace, de modo que sólo se interesa por los requisitos funcionales.

A pesar de que en el modelo del análisis hay un refinamiento de los requisitos, no se toman en cuenta el lenguaje de programación a usar en la construcción, la plataforma en la que se ejecutará la aplicación, los componentes prefabricados o reusables de otras aplicaciones, entre otras características que afectan al sistema, ya que el objetivo del análisis es comprender perfectamente los requisitos del software y no precisar cómo se implementará la solución. [9]

2.9.1 Diagramas de Clase de Análisis

El diagrama de clase del análisis permite representar cómo se soluciona el problema, siendo más entendible para el cliente, ya que no incluye aspectos de implementación.

Caso de Uso Crear Perfil

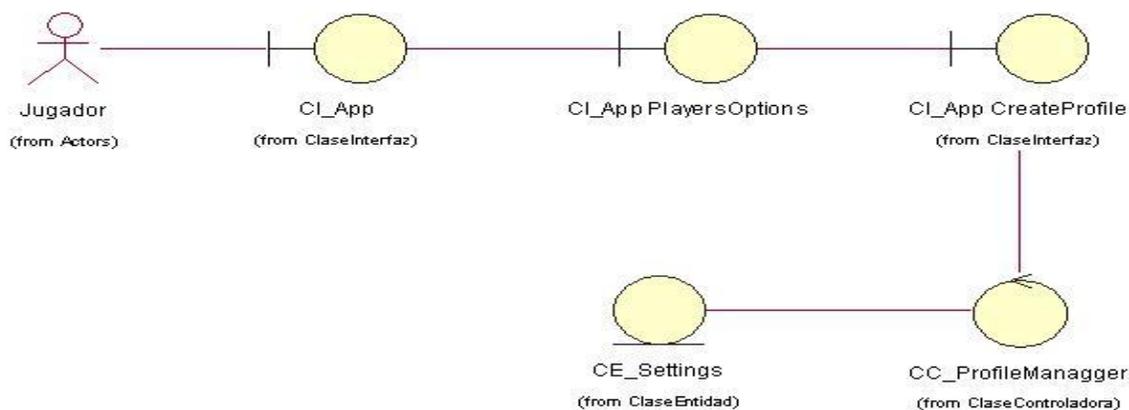


Figura 2.4 Diagrama de clases del análisis CU Crear Perfil.

Caso de Uso Editar Perfil

- Editar Red

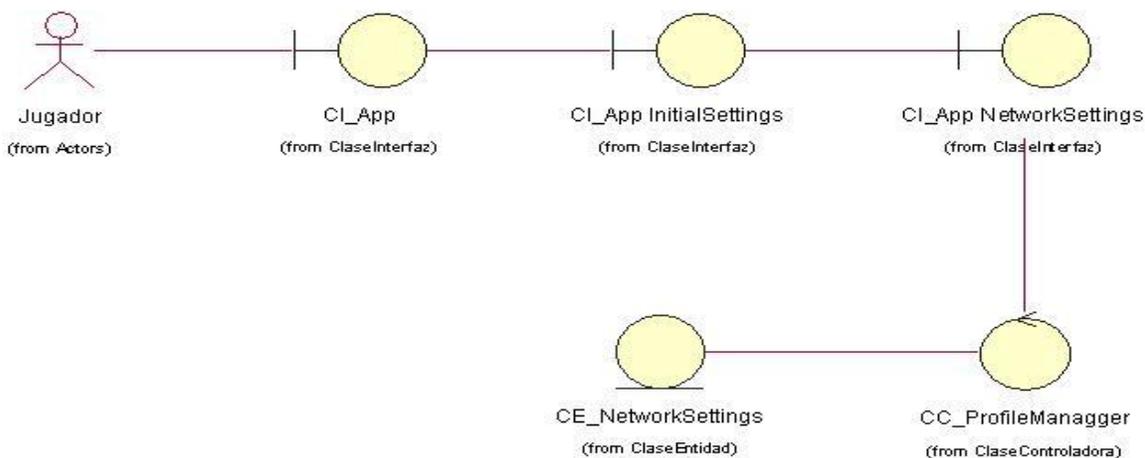


Figura 2.5 Diagrama de clases del análisis CU Editar Perfil (Editar Red).

• **Editar Controles**

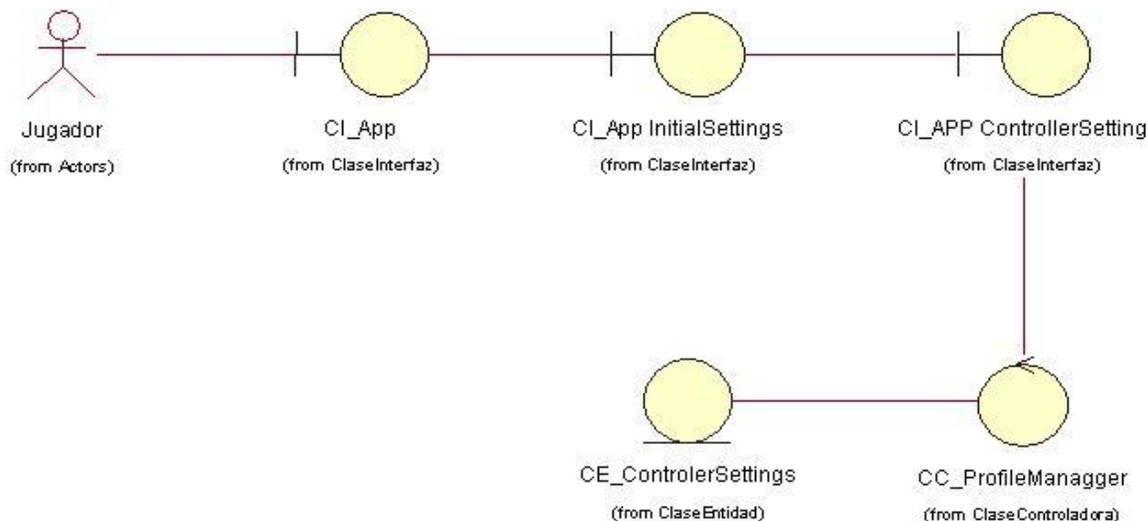


Figura 2.6 Diagrama de clases del análisis CU Editar Perfil (Editar Controles).

• **Editar Opciones Jugador**

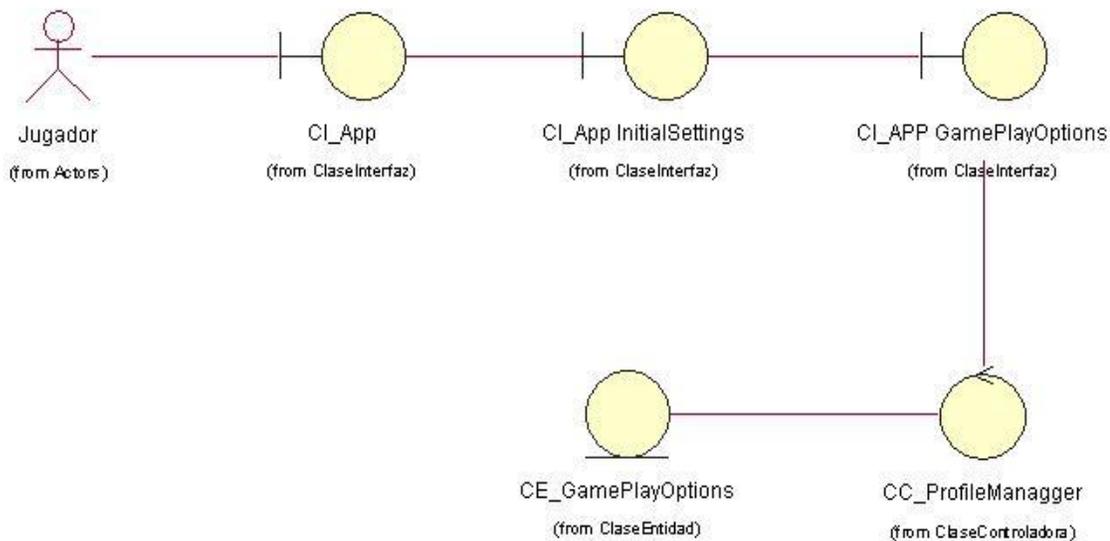


Figura 2.7 Diagrama de clases del análisis CU Editar Perfil (Editar Opciones Jugador).

- **Editar Visual**

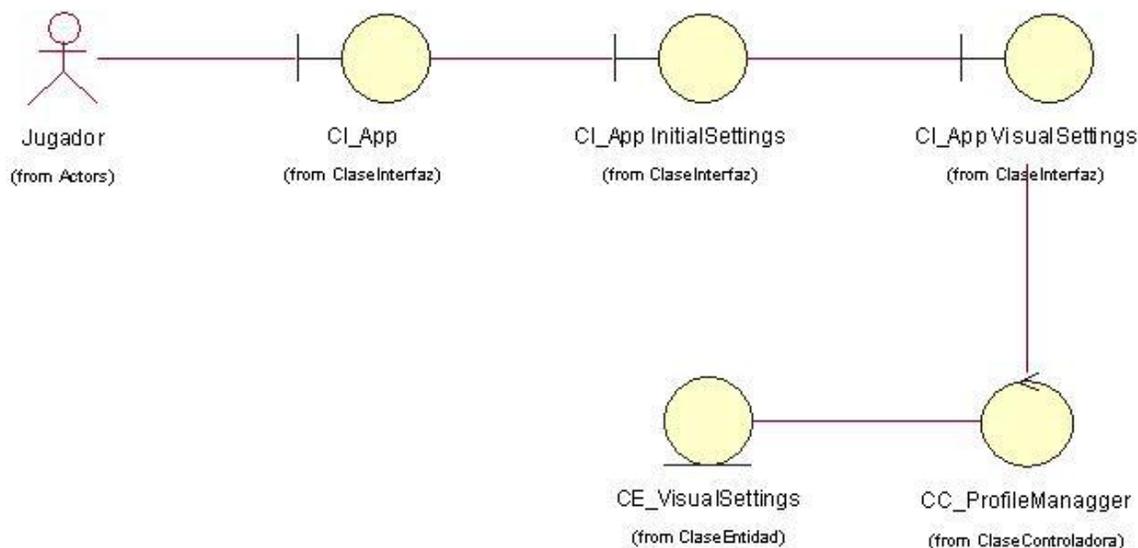


Figura 2.8 Diagrama de clases del análisis CU Editar Perfil (Editar Visual).

Caso de Uso Cargar Perfil

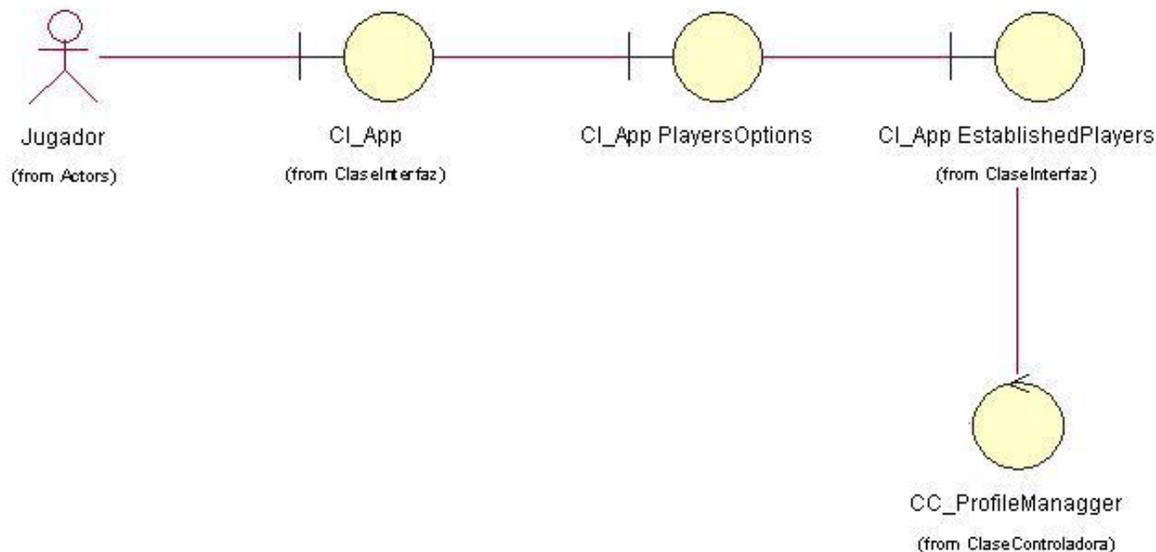


Figura 2.9 Diagrama de clases del análisis CU Cargar Perfil.

Caso de Uso Eliminar Perfil

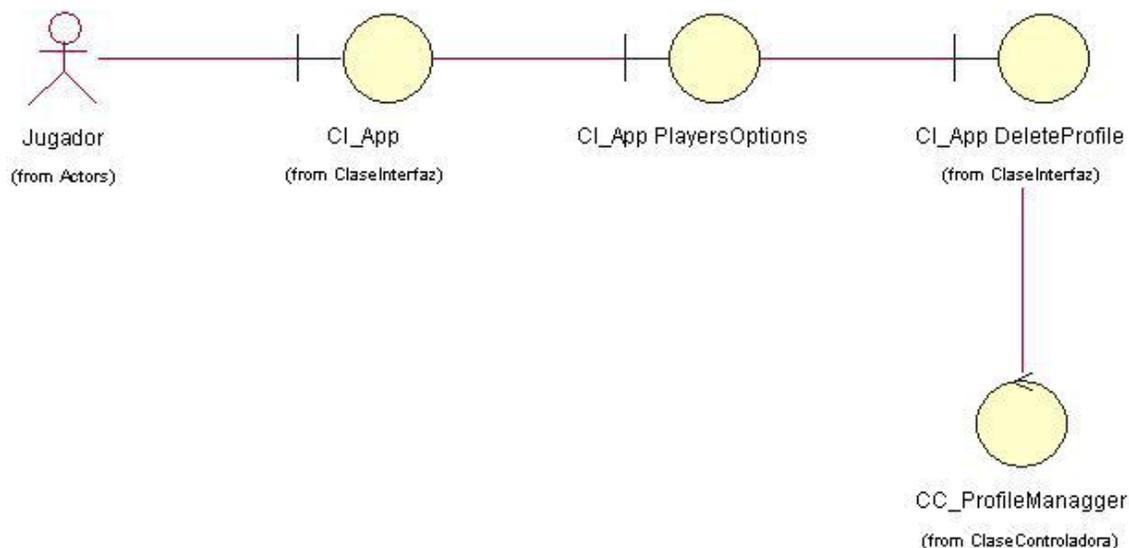


Figura 2.10 Diagrama de clases del análisis CU Eliminar Perfil.

Conclusiones

En este capítulo se le dio cumplimiento a las tareas planteadas para esta etapa. Se explicó la necesidad de gestionar los perfiles de usuario en el videojuego Rápido y Curioso además de seleccionar el modelo de persistencia adecuado para el proyecto según el estudio realizado. También se describió la composición del fichero en el cual se almacenará los datos de los usuarios. Se realizó el análisis del dominio y se hizo un levantamiento de los requisitos funcionales y no funcionales, de donde salieron los casos de uso a resolver en el problema, los que fueron descritos para un mejor entendimiento.

CAPÍTULO 3: DISEÑO E IMPLEMENTACIÓN DEL SISTEMA

En este capítulo se expone el diseño del módulo implementado, se explica cómo se pretende dar solución a las funcionalidades derivadas de los requisitos funcionales. Se detallan los patrones usados en la solución y se explican detalladamente las principales clases facilitando el entendimiento del trabajo. También quedan plasmados los diagramas de colaboración confeccionados a partir de los casos de uso y también los diagramas de despliegue y componentes.

3.1 Patrones

Un patrón es una descripción de un problema y su solución, que recibe un nombre y que puede emplearse en otros contextos; en teoría, indica la manera de utilizarlo en circunstancias diversas, es una solución estándar de un problema, expresa como solucionar un problema que ocurre repetidas veces en algún ámbito determinado de desarrollo de software y brindan una buena solución probada con anterioridad. Ahorra el tiempo dedicado al diseño, la mayoría de las veces se logran construir soluciones reutilizables y extensibles. [7]

En el presente trabajo se aplican patrones pertenecientes al grupo de los patrones GOF.

Command: Patrón de comportamiento de objeto, el cual encapsula una acción en un objeto. Un objeto que responda a la descripción de objeto Command puede ejecutar una acción al ocurrir un evento determinado. Este patrón es una alternativa orientada a objetos de las funciones callback (función que previamente es registrada y luego ejecutada), además permite al sistema abstraerse de las acciones específicas a ejecutar al ocurrir ciertos eventos, esto se debe a que el comando brinda una interfaz común para ejecutar la acción encapsulada por el mismo. El patrón permite además facilidad para incrementar la funcionalidad de la aplicación al agregar nuevos comandos sin la necesidad de cambiar las clases existentes. Dentro de todas las interfaces creadas en este trabajo se aplica dicho patrón, ya que existe variedad de objetos que tienen asignados un comando determinado, el cual ejecutarán en respuesta a una acción como presionar un botón, por ejemplo. [31]

Singleton: El patrón de diseño Singleton es uno de los más sencillos y utilizados del grupo de los GOF. Diseñado para restringir la creación de objetos pertenecientes a una clase o el valor de un tipo a un único objeto. Su intención consiste en garantizar que una clase sólo tenga una instancia y proporcionar un punto de acceso global a ella. Se implementa creando en nuestra clase un método que crea una instancia del objeto sólo si todavía no existe alguna. Para asegurar que la clase no puede ser instanciada nuevamente se regula el alcance del constructor (con atributos protegidos o privados). [31]

Además de los patrones de diseño existen los patrones de arquitectura como por ejemplo el MVC el cual se utiliza también en este trabajo.

MVC: Patrón Modelo / Vista / Controlador (MVC) se usa para construir interfaces de usuario. Está formado por tres tipos de objetos: el modelo es el objeto de la aplicación, la vista es la pantalla de presentación, o sea todas las interfaces que ve el usuario en su interacción con la aplicación y el controlador trata los eventos de la vista y modifica el estado del modelo, también informa a la vista de los posibles cambios ocurridos en el modelo. [3]

Con el uso de este patrón aumenta la flexibilidad y reutilización del código. MVC establece entre las vistas y los modelos suscripciones y notificaciones lo que podría llamarse como protocolo de comunicación entre ambas. Las vistas en todo momento reflejan el aspecto del estado del modelo. Cada vez que el modelo cambia sus datos, el modelo notifica a las vistas que dependen de él, en respuesta a esto cada vista tiene la oportunidad de actualizarse. [31]

Este enfoque permite conectar múltiples vistas a un modelo para proveer diferentes presentaciones. También se puede crear nuevas vistas para un modelo sin reescribirlo. Este patrón establece que los objetos modelo no deberían conocer directamente los objetos vista ni estar directamente acoplados a ellos. [7]

El funcionamiento de las interfaces creadas en el módulo para la gestión de los perfiles de usuarios con los ficheros de cada jugador y la clase controladora de los perfiles justifica el uso de este patrón de arquitectura.

3.2 Diseño de los Paquetes

Los paquetes ofrecen un mecanismo general para la organización de los subsistemas agrupando elementos de modelado. Es muy importante saber que un paquete puede contener otros paquetes, sin límite de anidamiento pero cada elemento pertenece a un solo paquete.

Un paquete se muestra gráficamente como una carpeta con etiquetas. Los paquetes subordinados se incluyen en su interior. El nombre del paquete se encuentra dentro de la etiqueta si el paquete describe sus elementos; en caso contrario estará en el centro de la misma carpeta. [7]

Haciendo un buen diseño se garantiza que el sistema quede modelado correctamente, además garantizamos una buena base para comenzar a implementar. A medida que se fue avanzando e implementando las clases que intervendrán en la solución del problema se identificaron las clases más

importantes y las clases auxiliares, todas estas fueron ubicadas en dos paquetes teniendo en cuenta sus funcionalidades.



Figura 3.1 Diagrama de clases por paquete.

3.2.1 Paquete Módulo Gestión de los Perfiles de Usuario

Este paquete está formado por todas las nuevas interfaces creadas, las cuales contienen una instancia de la clase GlobalResource la cual a su vez es la encargada de relacionarse con la clase controladora ProfileManager, que en resumen es la que realiza todas las funcionalidades que le dan cumplimiento a los requisitos funcionales. (Figura 3.2).

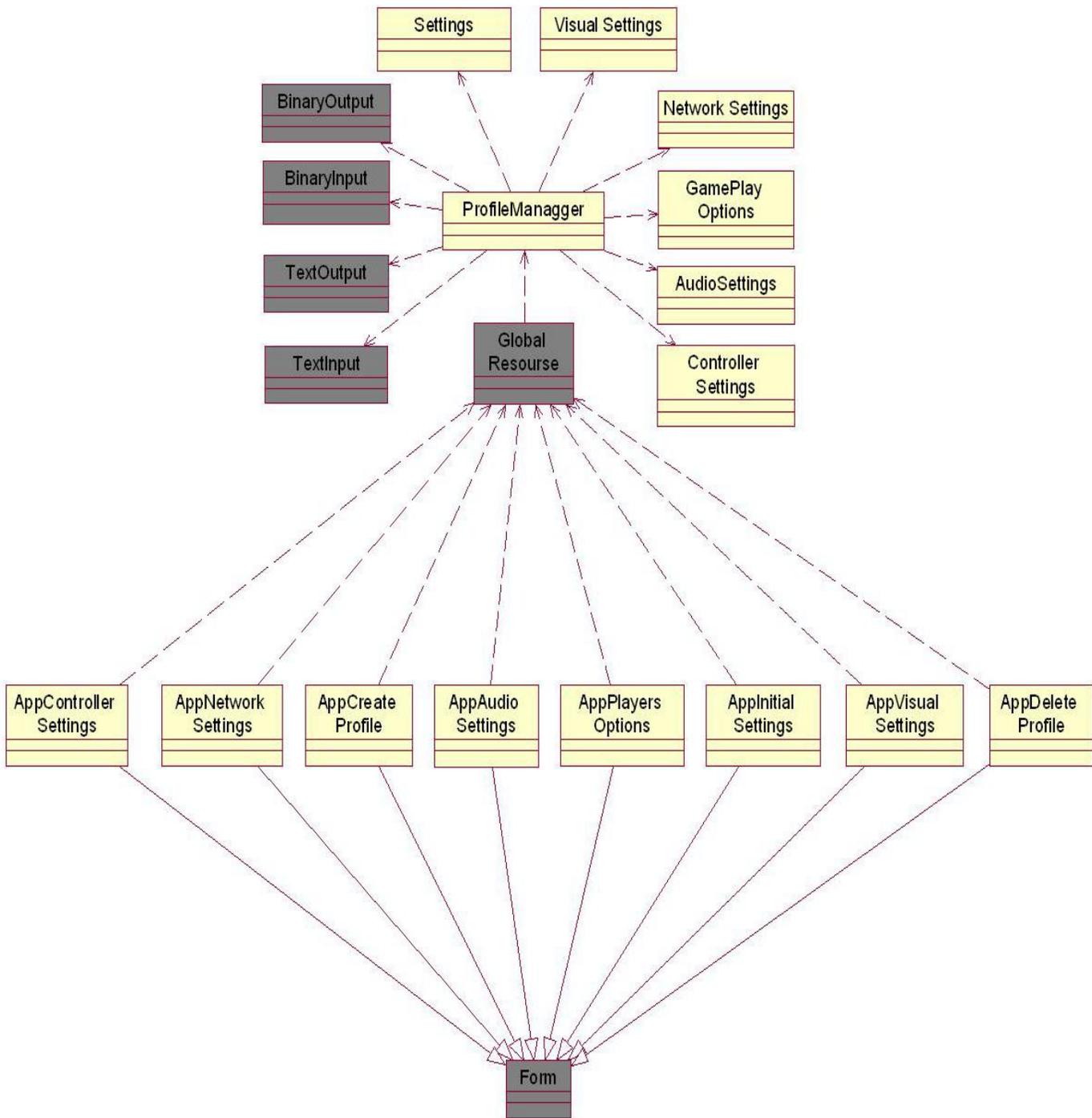


Figura 3.2 Paquete Módulo Gestión de los perfiles de usuario.

3.2.2 Paquete Menú

Este paquete contiene todas las interfaces del menú del videojuego “Rápido y Curioso”, por orientación en el proyecto, dentro de este paquete Menú debe estar el módulo para la gestión de los perfiles de usuario, por tal motivo es imprescindible tener en cuenta todas las clases que lo conforman, y además, de estas clases depende en gran medida el funcionamiento de todo el juego y brindan información visual al usuario de cómo van quedando configurados sus datos.

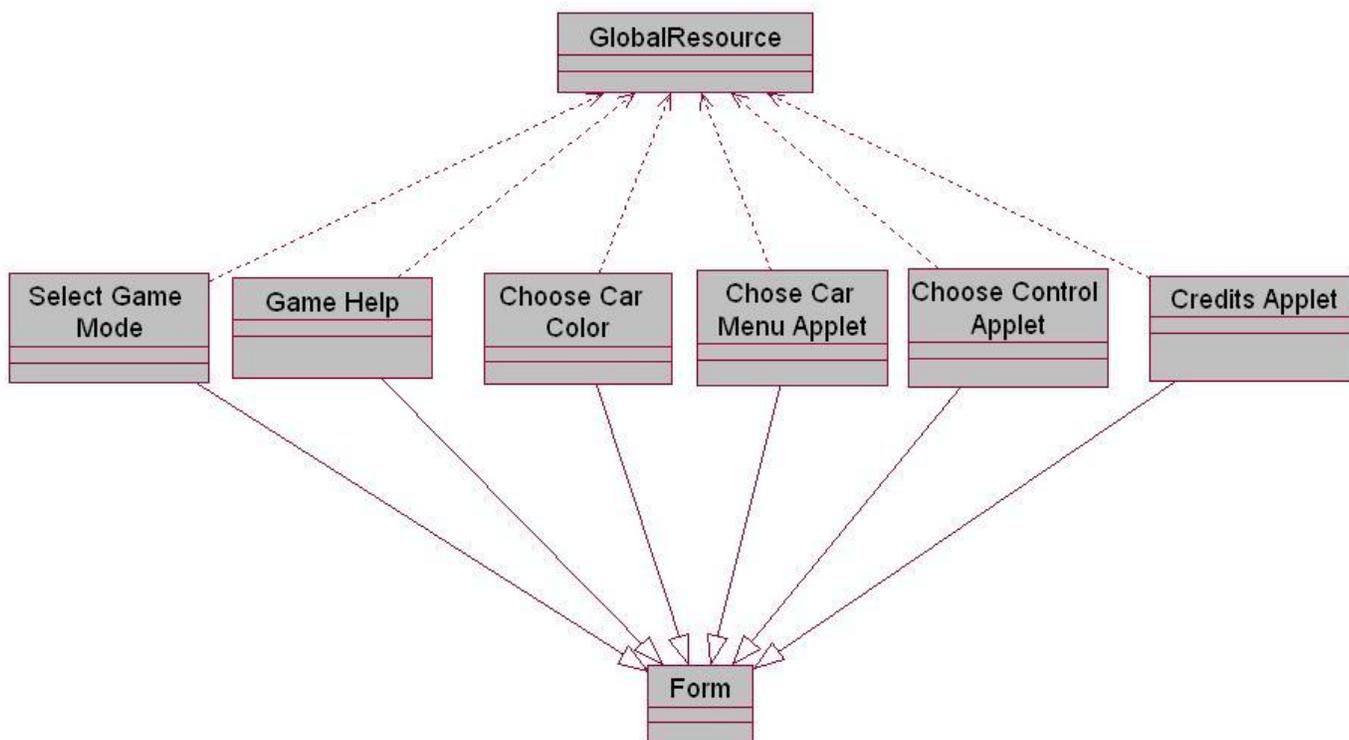


Figura 3.3 Paquete Menú.

3.3 Descripción de las clases

La descripción de las clases implica comentar detalladamente sobre los atributos y métodos que tiene una clase, proporcionando la información técnica necesaria para lograr la comprensión de la misma. Las principales clases implementadas en el presente trabajo son ProfileManager, Settings, ControllerSettings,

GamePlayOptions, NetworkSettings, VisualSettings las cuales responden directamente con el trabajo de creación y edición de los perfiles de usuario.

También enmarcadas como clases principales tenemos las clases interfaces creadas para la interacción con el usuario, estas clases son AppPlayersOptions, AppInitialSettings, AppCreateProfile, AppControllerSettings, AppDeleteProfile, AppEstablishedPlayers, AppNetworkSettings, AppVisualSetting, y dentro de las clases interfaces auxiliares creadas tenemos AppNewUserWarning, AppFullProfileWarning, AppRepeatedProfileWarning.

Las clases interfaz que tienen alguna entrada de datos, como por ejemplo la de edición de los controles, la de creación de los usuarios, la de edición de red, cuentan con métodos que realizan la misma funcionalidad lo que ajustados a cada interfaz, estos métodos son Draw, AssignPainting y Delete, estos métodos fueron implementados para reforzar el trabajo con la escritura de las letras en pantalla, resaltando que es la mejor vía encontrada para realizarlo teniendo en cuenta las particularidades del trabajo.

Nombre:	ProfileManager	
Tipo de clase:	Controladora	
Atributo	Tipo	
managger	ProfileManager	
GameSettingsDirection	string	
NetSettingsDirection	string	
PlayerDirection	string	
EstablishedPlayersFileDirection	string	
UsserActivePlayers	string	
Para cada función:		
Nombre:	UpdateProfile()	
Descripción:	Función que se encarga cada vez que inicia el juego de verificar si existe el usuario por defecto, en caso de no existir lo crea, además esta función garantiza que los perfiles existentes en la carpeta Jugadores formen parte del juego,	

	resolviendo así una de las funcionalidades que brinda el módulo relacionada con copiar un perfil de una computadora a otra y poder cargarlo.
Nombre:	GetActiveProfile()
Descripción:	Función que se encarga de retornar el valor tomado por el atributo nombreUsuarioActivo.
Nombre:	GetPlayers()
Descripción:	Función que se encarga de retornar un arreglo de string con los jugadores creados.
Nombre:	GetCountPlayerS()
Descripción:	Función que se encarga de retornar la cantidad de jugadores creados.
Nombre:	MoveActiveProfile(int pos)
Descripción:	Función que se encarga, pasándole una posición poner al jugador en esa posición como el jugador activo.
Nombre:	SetActiveProfile(string name)
Descripción:	Función que se encarga, pasándole el nombre de un jugador de ponerlo como el jugador activo.
Nombre:	ProfilePosition(int pos)
Descripción:	Función que se encarga, pasándole una posición retornar el nombre del jugador en esa posición.
Nombre:	CreateProfile(Settings * profile)
Descripción:	Función que se encarga, pasándole un objeto de tipo Settings de crear un nuevo jugador con todas las características del objeto profile.
Nombre:	CreateByEdition(Settings * profile)
Descripción:	Función que se encarga, pasándole un objeto de tipo Settings de crear un nuevo jugador después de una edición del perfil.
Nombre:	DeleteProfile(int pos)
Descripción:	Función que se encarga, pasándole una posición eliminar el perfil en esa posición.
Nombre:	EditGamePlayOptions(GamePlayOptions * g, string name)

Descripción:	Función que se encarga, pasándole un objeto de tipo <code>GamePlayOptions</code> y el nombre de un usuario, editar las opciones de juego del usuario con todas las características del objeto <code>g</code> .
Nombre:	<code>EditVisual(VisualSettings * v, string name)</code>
Descripción:	Función que se encarga, pasándole un objeto de tipo <code>VisualSettings</code> y el nombre de un usuario, editar las opciones visuales del usuario con todas las características del objeto <code>v</code> .
Nombre:	<code>EditController(ControllerSettings * c, string name)</code>
Descripción:	Función que se encarga, pasándole un objeto de tipo <code>ControllerSettings</code> y el nombre de un usuario, editar los controles del usuario con todas las características del objeto <code>c</code> .
Nombre:	<code>EditNetwork(NetworkSettings * n, string name)</code>
Descripción:	Función que se encarga, pasándole un objeto de tipo <code>NetworkSettings</code> y el nombre de un usuario, editar las opciones de red del usuario con todas las características del objeto <code>n</code> .
Nombre:	<code>EditNetSettings_GameSettings(server_IP, players_count ,server_waiting_time)</code>
Descripción:	Función que se encarga, pasándole el nombre del servidor, el número del puerto y el tiempo de espera modificar estos valores en los ficheros que contienen estos datos en el juego.
Nombre:	<code>LoadCarModel_CarColor_CarNPC_ScreenMode_Volumen(string name, string description)</code>
Descripción:	Función que se encarga de retornar un valor entero que puede ser el color del carro, modelo del carro, cantidad de jugadores, resolución o volumen, esta función recibe el nombre del jugador que se desea conocer la información y el parámetro descripción indica cual dato se retornará.
Nombre:	<code>LoadControl (string name)</code>
Descripción:	Función que se encarga de retornar un arreglo de string que contiene los controles almacenados en el fichero perteneciente al usuario pasado por parámetro.

Nombre:	NumberSDL(string character)
Descripción:	Función que se encarga de retornar un entero que representa el código ASCII del carácter pasado por parámetro.
Nombre:	LoadControlActiveProfile()
Descripción:	Función que se encarga de retornar un arreglo de enteros con los controles del usuario activo, estos valores enteros representan el código ASCII de cada control.
Nombre:	LoadServer_IP(string name)
Descripción:	Función que se encarga de retornar el servidor que tiene almacenado el fichero perteneciente al usuario pasado por parámetro.
Nombre:	LoadPort_ServerWaitingTime(string name, string description)
Descripción:	Función que se encarga de retornar un valor entero que puede ser el puerto o tiempo de espera, esta función recibe el nombre del jugador que se desea conocer la información y el parámetro descripción indica cual dato desea retornar.

Tabla # 10 Descripción de la clase ProfileManager.

En el diseño se describen todos los aspectos del sistema a construir mediante el diagrama de clases que ya fue detallado anteriormente y los diagramas de secuencia correspondientes a los casos de uso. Este paso constituye el refinamiento de las etapas anteriores.

3.4 Diagramas de Secuencia

Los diagramas de secuencia se muestran la interacción de los objetos que componen un sistema de forma temporal. [10] Todos los diagramas se encuentran en los Anexos.

3.5 Diagrama de componentes

El diagrama de componentes muestra las relaciones de dependencia entre las partes modulares de un sistema y encapsula la implementación. También en éste se pueden expresar las relaciones que existen

entre los diferentes ficheros del sistema así como las bibliotecas empleadas para el desarrollo del mismo. En el Anexo I se muestra el diagrama de componentes del módulo desarrollado en el presente trabajo.

Conclusiones

En este capítulo se le dio cumplimiento a la tarea de diseñar e implementar el módulo para la gestión de los perfiles de usuario en el videojuego de conducción de automóviles “Rápido y Curioso”, de manera detallada se explicó el diseño del módulo implementado, la solución a las funcionalidades derivadas de los requisitos funcionales. Además se habló de los patrones usados en la solución y se describen las principales clases facilitando el entendimiento del trabajo. También se muestran los diagramas de colaboración confeccionados a partir de los casos de uso y los diagramas de despliegue y de componentes.

CAPÍTULO 4: PRUEBAS

Al igual que la etapa de análisis, el diseño de las pruebas es una parte considerable de trabajo. Durante esta fase se diseñan e implementan los casos de prueba. Tomando como punto de partida el resultado del análisis de pruebas, se diseñan las pruebas en detalle, indicando cuáles son los pasos o el procedimiento a seguir, las técnicas, medidas o análisis que hay que aplicar así como los conjuntos de datos de prueba que se usarán, y los resultados esperados que se producirán, para conseguir demostrar que los objetivos de la prueba se cumplen y por lo tanto verifican y validan el requisito que cubren. El diseño de las pruebas persigue la prevención de defectos, antes de que por codificación puedan haberse llegado a producir. Un buen diseño de pruebas previene, en consecuencia, los defectos antes de que lleguen a etapas más tardías.

El nivel de prueba a realizar es “**Prueba de Sistema**” las cuales verifican el correcto funcionamiento del sistema completo, incluyendo casos de prueba que busquen los fallos del sistema. Son pruebas destructivas y persiguen demostrar la robustez del sistema aun en condiciones adversas. Verifican requisitos funcionales y no funcionales.

El tipo de prueba elegido para aplicar es la “**Prueba de caja negra**” las cuales son pruebas funcionales. Se parte de los requisitos funcionales, a muy alto nivel, para diseñar pruebas que se aplican sobre el sistema sin necesidad de conocer como está construido por dentro. Las pruebas se aplican sobre el sistema empleando un determinado conjunto de datos de entrada y observando las salidas que se producen para determinar si la función se está desempeñando correctamente por el sistema bajo prueba. Las herramientas básicas son observar la funcionalidad y contrastar con la especificación. [4]

Cada caso de prueba a parte de su descripción, se le adjunta una foto con el resultado que se espera del sistema y de esta forma quedan ilustradas también todas las nuevas interfaces creadas en el juego.

4.1 Especificaciones de Hardware y Software

Hardware

- Sistema Operativo: Windows 5.1 Service Pack 2
- Procesador: Intel, Pentium 4, 3.00 GHz
- Memoria RAM: 512 MB

Software

- IDE: Visual Studio .NET 2005
- Versión de G3D: 6.10

4.2 Diseño de Casos de Prueba por cada caso de uso

A continuación se elaboran los casos de prueba para los cuatro casos de uso del módulo donde se verifican detalladamente que el sistema cumple con las funcionalidades esperadas.

4.2.1 Caso de Uso Crear Perfil

Descripción General

Mediante este caso de uso el usuario podrá crear un nuevo perfil.

Pruebas planificadas:

- Crear usuario.

Caso de Prueba # 1 Crear Usuario

Breve Descripción

Este caso de prueba le permite al usuario crearse un nuevo perfil.

Flujo Principal

1. El usuario presiona el botón Jugadores.
2. El sistema muestra una nueva interfaz con tres nuevos botones: Crear, Eliminar y Jugador Activo.
3. El usuario presiona el botón Crear.
4. El sistema muestra una nueva interfaz que contiene una caja de texto donde el usuario debe escribir el nombre.
5. El usuario introduce los datos del nuevo perfil y presiona el botón Aceptar.
6. El sistema crea un nuevo fichero con el mismo nombre que fue escrito por el usuario e inmediatamente destruye esa interfaz y muestra una nueva interfaz donde están todos los usuarios creados hasta el momento, donde aparece en la primera posición el usuario creado recientemente.

Condiciones de ejecución

- Deben existir menos de cinco perfiles creados en el juego.

Iteraciones:

Clases Válidas	Clases Inválidas	Resultado Esperado	Resultado de la Prueba	Observaciones
<p>Presiona el botón Aceptar Adicionar un nuevo usuario el nombre obligatoriamente con al menos un carácter.</p> <p>Ejemplo: Nombre: Orlando</p>		<p>El sistema debe adicionar el nuevo usuario y mostrar la interfaz de los usuarios activos y el perfil activo debe ser el creado.</p>	<p>El sistema adiciona el usuario y muestra la interfaz actualizada correctamente.</p>	

	Adicionar un nuevo usuario dejando en blanco la caja de texto que recibe el nombre del usuario.	El sistema no permite la creación del usuario hasta que sea escrito al menos un carácter en la caja de texto y se mantiene en la misma interfaz.	El sistema no crea el usuario y se mantiene en la misma interfaz.	
	Adicionar un nuevo usuario con el nombre de uno ya existente.	El sistema debe mostrar una nueva interfaz con un mensaje que indica que el usuario ya existe.	El sistema una nueva interfaz con un mensaje que indica que el usuario ya existe.	

Tabla # 11 Caso de prueba Crear Usuario.



Figura 4.1 Interfaz Opciones del Jugador.



Figura 4.2 Interfaz Crear Jugador.



Figura 4.3 Interfaz Jugador Activo.

4.2.2 Caso de Uso Eliminar Perfil

Descripción General

Mediante este caso de uso el usuario podrá eliminar un nuevo perfil existente.

Pruebas planificadas:

- Eliminar usuario.

Caso de Prueba # 2 Eliminar Usuario

Breve Descripción

Este caso de prueba le permite al usuario eliminar un perfil existente.

Flujo Principal

1. El usuario presiona el botón Jugadores.
2. El sistema muestra una nueva interfaz con tres nuevos botones: Crear, Eliminar y Jugador Activo.
3. El usuario presiona el botón Eliminar.
4. El sistema muestra una nueva interfaz que contiene una lista con todos los usuarios existentes que pueden ser eliminados.
5. El usuario hace clic sobre el perfil que desea eliminar.
6. El sistema elimina el fichero y muestra la misma interfaz actualizada con los usuarios que quedaron.

Condiciones de ejecución

- Debe existir al menos un perfil.

Iteraciones:

Clases Válidas	Clases Inválidas	Resultado Esperado	Resultado de la Prueba	Observaciones
Hace clic sobre el perfil que desea eliminar. Obligatoriamente debe existir el perfil en esa posición para poderlo eliminar.		El sistema debe eliminar el perfil en la posición señalado por el usuario al hacer clic en esa posición.	El sistema elimina el perfil en esa posición y muestra la interfaz actualizada sin la presencia del usuario eliminado.	

Tabla # 12 Caso de prueba Eliminar Usuario.

En este caso de prueba nunca existirá una clase inválida ya que la interfaz está programada para no reaccionar al clic del usuario en caso de que no existan perfiles, por tal motivo nunca se podrá dar el caso de querer eliminar un perfil que no existe.



Figura 4.4 Interfaz Eliminar Jugador.

4.2.3 Caso de Uso Cargar Perfil

Descripción General

Mediante este caso de uso el usuario podrá cargar el perfil que desee, entendiéndose por cargar la acción de señalar el perfil deseado como el perfil activo.

Pruebas planificadas:

- Cargar perfil.

Caso de Prueba # 3 Cargar Perfil

Breve Descripción

Este caso de prueba le permite al usuario cargar el perfil que desee.

Flujo Principal

1. El usuario presiona el botón Jugadores.
2. El sistema muestra una nueva interfaz con tres nuevos botones: Crear, Eliminar y Jugador Activo.

3. El usuario presiona el botón Jugador Activo.
4. El sistema muestra una nueva interfaz que contiene una lista de todos los usuarios creados hasta el momento, en la primera posición de esta lista está el perfil activo.
5. El usuario hace clic sobre el perfil que desea sea el activo.
6. El sistema actualiza la interfaz poniendo en la primera posición de la lista el perfil seleccionado e inmediatamente pasa a ser el perfil activo.

Condiciones de ejecución

- Deben existir al menos dos usuarios.

Iteraciones:

Clases Válidas	Clases Inválidas	Resultado Esperado	Resultado de la Prueba	Observaciones
Hace clic sobre el perfil que desea cargar. Obligatoriamente debe existir el usuario en esa posición para poderlo cargar.		El sistema actualiza la lista de la interfaz poniendo en la primera posición el nuevo usuario activo.	El sistema actualiza la lista de la interfaz poniendo en la primera posición el nuevo usuario activo.	

Tabla # 13 Caso de prueba Cargar Usuario.

En este caso de prueba nunca existirá una clase inválida ya que la interfaz está programada para no reaccionar al clic del usuario en caso de que no existan al menos dos perfiles.

4.2.4 Caso de Uso Editar Perfil

Descripción General

Mediante este caso de uso el usuario podrá editar su perfil como estime conveniente, entre las opciones de edición que puede hacer está configurar los controles, el volumen, la resolución, la red y opciones de juego.

Pruebas planificadas:

- Configurar controles.
- Configurar red.
- Configurar opciones de juego.

Caso de Prueba # 4 Configurar controles

Breve Descripción

Este caso de prueba le permite al usuario configurar los controles.

Flujo Principal

1. El usuario presiona el botón Configuración.
2. El sistema muestra una nueva interfaz con cinco nuevos botones: Controles, Red, Visual y Opciones de juego.
3. El usuario presiona el botón Controles.
4. El sistema muestra una nueva interfaz que contiene trece cajas de texto con su descripción, cada una de ellas muestra la tecla asignada para controlar el auto.
5. El usuario puede borrar y redefinir la tecla que desee, nunca estará repetida una misma tecla, cuando termina de configurar las teclas a su antojo presiona el botón Aceptar.
6. El sistema muestra la misma interfaz actualizada con los controles configurados.

Flujo Alternativo

1. El usuario presiona el botón Configuración.
2. El sistema muestra una nueva interfaz con cinco nuevos botones: Controles, Red, Visual y Opciones de juego.
3. El usuario presiona el botón Controles.
4. El sistema muestra una nueva interfaz que contiene trece cajas de texto con su descripción, cada una de ellas muestra la tecla asignada para controlar el auto.
5. El usuario puede borrar y redefinir la tecla que desee, nunca estará repetida una misma tecla, cuando termina de configurar las teclas a su antojo presiona el botón Aceptar, al salvar los controles como fueron configurados tiene la opción de presionar el botón Defecto con el cual garantiza usar los controles por defecto, después de presionar este botón debe accionar nuevamente el botón Aceptar para guardar los cambios.
6. El sistema muestra la misma interfaz actualizada con los controles configurados.

Condiciones de ejecución

- Deben existir al menos un perfil diferente del perfil por defecto para poder configurar, ya que el perfil por defecto que siempre está presente no es configurable.

Iteraciones para el flujo central:

Clases Válidas	Clases Inválidas	Resultado Esperado	Resultado de la Prueba	Observaciones
Presiona el botón Aceptar Salva las nuevas configuraciones hechas en los controles.		El sistema debe configurar los controles y actualizar la interfaz con los nuevos valores.	El sistema configura los controles y actualiza la interfaz con los nuevos valores.	
	Escribir una tecla que ya está presente en los controles; nunca se permitirá una tecla repetida.	El sistema no permite la edición de la caja de texto hasta tanto la tecla sea diferente a todas las teclas ya configuradas.	El sistema no permite la edición de la caja de texto hasta tanto la tecla sea diferente a todas las teclas ya configuradas.	

Tabla # 14 Caso de prueba Configurar Controles (Flujo central).

Iteraciones para el flujo alternativo:

Clases Válidas	Clases Inválidas	Resultado Esperado	Resultado de la Prueba	Observaciones
Presiona el botón Defecto Restablece los controles por defecto.		El sistema actualiza la interfaz con los valores por defecto.	El sistema actualiza la interfaz con los valores por defecto.	

Tabla # 15 Caso de prueba Configurar Controles (Flujo alternativo).



Figura 4.5 Interfaz Opciones de Configuración.



Figura 4.6 Interfaz Configurar Controles.

Caso de Prueba # 5 Configurar red

Breve Descripción

Este caso de prueba le permite al usuario configurar los parámetros de red.

Flujo Principal

1. El usuario presiona el botón Configuración.
2. El sistema muestra una nueva interfaz con cinco nuevos botones: Controles, Red, Visual y Opciones de juego.
3. El usuario presiona el botón Red.
4. El sistema muestra una nueva interfaz que contiene tres cajas de texto con su descripción, cada una de ellas muestra el valor que contienen los parámetros de red que el usuario desea configurar.

5. El usuario puede borrar y redefinir la tecla que desee, nunca estará repetida una misma tecla, cuando termina de configurar las teclas a su antojo presiona el botón Aceptar.
6. El sistema muestra la misma interfaz actualizada con los parámetros de red configurados.

Flujo Alternativo

1. El usuario presiona el botón Configuración.
2. El sistema muestra una nueva interfaz con cinco nuevos botones: Controles, Red, Visual y Opciones de juego.
3. El usuario presiona el botón Red.
4. El sistema muestra una nueva interfaz que contiene tres cajas de texto con su descripción, cada una de ellas muestra el valor que contienen los parámetros de red que el usuario desea configurar.
5. El usuario puede borrar y redefinir el parámetro que desee, luego de configurar presiona el botón Aceptar para guardar los cambios realizados, también puede restablecer los parámetros por defecto presionando el botón Defecto y si desea salvarlos nuevamente pulsar el botón Aceptar.
6. El sistema muestra la misma interfaz actualizada con los parámetros de red configurados.

Condiciones de ejecución

- Deben existir al menos un perfil diferente del perfil por defecto para poder configurar, ya que el perfil por defecto que siempre está presente no es configurable.

Iteraciones para el flujo central:

Clases Válidas	Clases Inválidas	Resultado Esperado	Resultado de la Prueba	Observaciones
Presiona el botón Aceptar Salva las nuevas configuraciones hechas en los parámetros de red.		El sistema debe configurar los parámetros de red y actualizar la interfaz con los nuevos valores.	El sistema configura los parámetros de red y actualiza la interfaz con los nuevos valores.	
	Dejar en blanco la caja de texto perteneciente al tiempo de espera.	Si se realiza la salva el sistema salvará este parámetro con el valor por defecto y actualiza la interfaz con ese valor.	Si se realiza la salva el sistema salvará este parámetro con el valor por defecto y actualiza la interfaz con ese valor.	
	Escribir en la primera caja de texto perteneciente al IP más de tres puntos.	El sistema no permite la edición de la caja de texto con el carácter punto hasta tanto no sea eliminado un punto.	El sistema no permite la edición de la caja de texto con el carácter punto hasta tanto no sea eliminado un punto.	

Tabla # 16 Caso de prueba Configurar Red (Flujo central).

Iteraciones para el flujo alternativo:

Clases Válidas	Clases Inválidas	Resultado Esperado	Resultado de la Prueba	Observaciones
Presiona el botón Defecto Restablece los parámetros de red por defecto.		El sistema actualiza la interfaz con los valores por defecto.	El sistema actualiza la interfaz con los valores por defecto.	

Tabla # 17 Caso de prueba Configurar Red (Flujo alternativo).

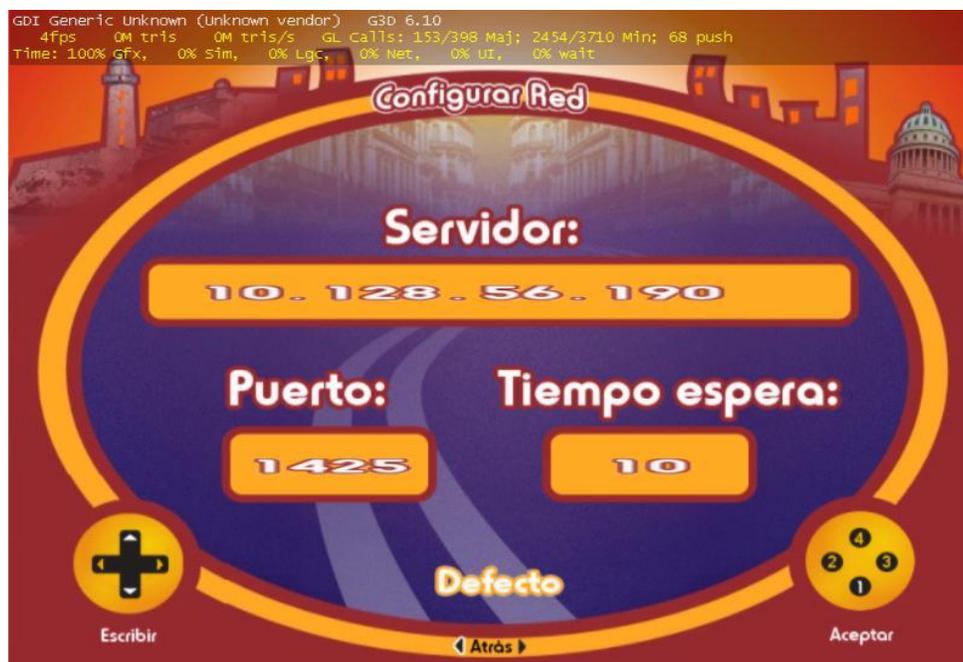


Figura 4.7 Interfaz Configurar Red.

Caso de Prueba # 6 Configurar opciones de juego

Breve Descripción

Este caso de prueba le permite al usuario configurar una opción de juego, ya que las otras dos están bloqueadas por peticiones del cliente.

Flujo Principal

1. El usuario presiona el botón Configuración.
2. El sistema muestra una nueva interfaz con cinco nuevos botones: Controles, Red, Visual y Opciones de juego.
3. El usuario presiona el botón Opciones de juego.
4. El sistema muestra una nueva interfaz que contiene tres cajas de texto con su descripción, cada una de ellas muestra el valor que contienen las opciones de juego que el usuario desea configurar, solo se podrán configurar dos de estas opciones.
5. El usuario puede borrar y redefinir la primera opción que corresponde con la cantidad de jugadores, al concluir la configuración debe presionar el botón Aceptar.
6. El sistema muestra la misma interfaz actualizada con las opciones de juego actualizadas.

Flujo Alternativo

1. El usuario presiona el botón Configuración.
2. El sistema muestra una nueva interfaz con cinco nuevos botones: Controles, Red, Visual y Opciones de juego.
3. El usuario presiona el botón Opciones de juego.
4. El sistema muestra una nueva interfaz que contiene tres cajas de texto con su descripción, cada una de ellas muestra el valor que contienen las opciones de juego que el usuario desea configurar, solo se podrán configurar dos de estas opciones.

5. El usuario puede borrar y redefinir la opción que desee, luego de configurar presiona el botón Aceptar para guardar los cambios realizados, también puede restablecer las opciones por defecto presionando el botón Defecto y si desea salvarlas nuevamente pulsar el botón Aceptar.
6. El sistema muestra la misma interfaz actualizada con las opciones de juego actualizadas.

Condiciones de ejecución

- Deben existir al menos un perfil diferente del perfil por defecto para poder configurar, ya que el perfil por defecto que siempre está presente no es configurable.

Iteraciones para el flujo central:

Clases Válidas	Clases Inválidas	Resultado Esperado	Resultado de la Prueba	Observaciones
Presiona el botón Aceptar Salva las nuevas configuraciones hechas en la opción de juego cantidad de jugadores.		El sistema debe configurar las opciones de juego y actualizar la interfaz con los nuevos valores.	El sistema configura las opciones de juego y actualiza la interfaz con los nuevos valores.	
	Dejar en blanco la caja de texto perteneciente a la cantidad de jugadores.	Si se realiza la salva el sistema salvará este parámetro con el valor por defecto y actualiza la interfaz con ese valor.	Si se realiza la salva el sistema salvará este parámetro con el valor por defecto y actualiza la interfaz con ese valor.	

Tabla # 18 Caso de prueba Configurar Opciones de Juego (Flujo central).

Iteraciones para el flujo alternativo:

Clases Válidas	Clases Inválidas	Resultado Esperado	Resultado de la Prueba	Observaciones
Presiona el botón Defecto Restablece los parámetros de red por defecto.		El sistema actualiza la interfaz con los valores por defecto.	El sistema actualiza la interfaz con los valores por defecto.	

Tabla # 19 Caso de prueba Configurar Opciones de Juego (Flujo alternativo).

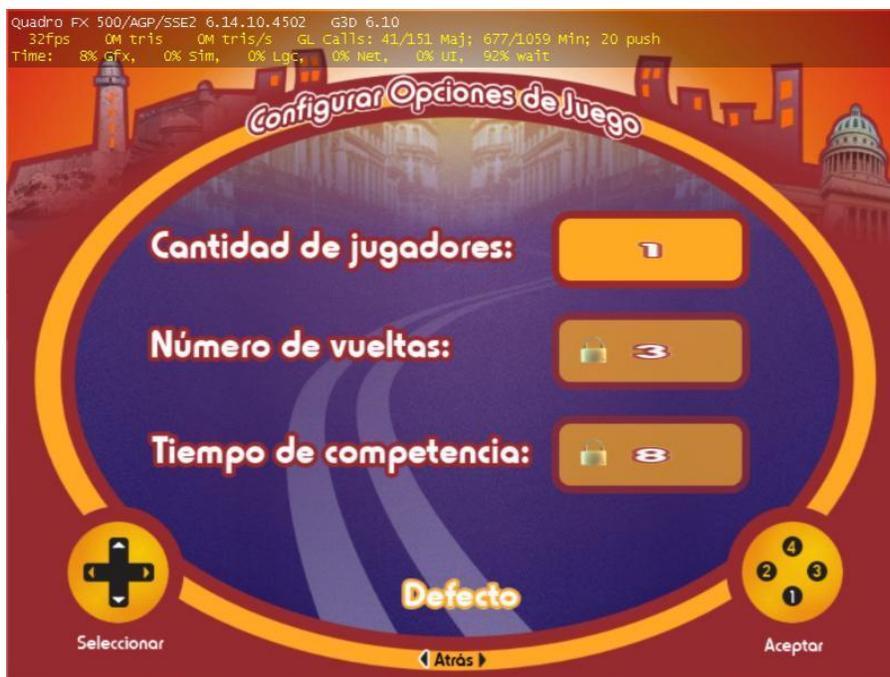


Figura 4.8 Interfaz Configurar Opciones de Juego.

Conclusiones

Después de realizar todos los casos de prueba los resultados arrojados fueron exitosos, el sistema en todos los casos se comportó satisfactoriamente, ante las clases validas y las inválidas, cumpliéndose los objetivos trazados para dar cumplimiento a los requisitos funcionales. Importante señalar que en la aplicación del segundo caso de prueba se detectó un error en la respuesta del sistema, el cual fue corregido inmediatamente, de tal forma se manifiesta la importancia de aplicar pruebas a una aplicación para darse cuenta de la robustez de la misma o bien de las debilidades que presenta.

CONCLUSIONES

Para dar cumplimiento al objetivo general de este trabajo en correspondencia con las exigencias del proyecto Rápido y Curioso, se hizo un estudio de diferentes géneros de videojuegos para ver las técnicas utilizadas para gestionar los perfiles de usuario.

Como la mayoría de los videojuegos pertenecen a empresas privadas, a los desarrolladores no les reporta ningún beneficio compartir información referente al tema de perfiles de usuario, por lo que la información existente es escasa, por tal motivo, este trabajo servirá de apoyo a todas aquellas personas interesadas en el tema ya que se obtuvo una bibliografía actualizada que será de gran ayuda.

Se identificaron los elementos del videojuego en cuestión de los que se necesitaba mantener un registro permanente o que fuera necesario configurar. A partir de ahí se seleccionó y diseñó el modelo de persistencia que se usó para exportar e importar los perfiles de cada jugador.

Se tuvieron en cuenta los requisitos funcionales y no funcionales, de donde surgieron los primeros casos de uso del sistema. Se transitó por las etapas de análisis, diseño e implementación donde surgieron y maduraron las clases, se implementaron los casos de uso y se creó el diagrama de componentes.

Finalmente se probaron las funcionalidades del módulo de gestión de los perfiles de usuario en el videojuego “Rápido y Curioso”, obteniéndose resultados satisfactorios.

Con la creación del módulo para la gestión de los perfiles de usuario en el videojuego Rápido y Curioso, se le dio cumplimiento al objetivo de esta investigación y se resolvió el problema científico planteado al inicio de la misma.

RECOMENDACIONES

Se recomienda continuar incorporando funcionalidades al módulo desarrollado, como por ejemplo:

- Salvar y mostrar los mejores resultados de cada competidor en cada una de las pistas, y compararlo con los mejores tiempos para esa misma pista entre todos los usuarios registrados.
- Salvar las repeticiones de las carreras que desee el jugador y que pueda verlas cuando estime conveniente.
- Salvar datos referentes a las habilidades de cada conductor a medida que estos adquieren destreza. Por ejemplo, las velocidades a las que ataca cada curva en dependencia de su amplitud, la distancia en la que comienza a desacelerar para tomar dichas curvas, así como las variaciones del timón en estas situaciones. Todos esos datos pudieran procesarse mediante técnicas de aprendizaje de la inteligencia artificial para mejorar el desempeño de los jugadores controlados por la computadora, incrementando la competitividad de los mismos.

BIBLIOGRAFÍA

- [1]. **Duch i Gavaldà, Jordi y Tejedor Navarro, Heliodoro.** “Introducción a los videojuegos”. Universidad Abierta de Cataluña, 2005.
- [2]. **Duch i Gavaldà, Jordi y Tejedor Navarro, Heliodoro.** “Sonido, interacción y redes”. Universidad Abierta de Cataluña, 2005.
- [3]. **Duch i Gavaldà, Jordi y Tejedor Navarro, Heliodoro.** “Lógica del videojuego”. Universidad Abierta de Cataluña, 2005.
- [4]. **Kynetia, Empresa de arquitectura y desarrollo de software.** “Diseño de Pruebas”, Madrid, 2007.
<http://www.kynetia.es/calidad/disenio-de-pruebas.htm>
- [5]. **Álvarez, Sara.** “Sistemas gestores de bases de datos”, 2007.
<http://www.desarrolloweb.com/articulos/sistemas-gestores-bases-datos.html>
- [6]. **Labra Gayo, José Emilio.** “Lenguaje XML”. Universidad de Oviedo, 2002.
- [7]. **Craig Larman.** “UML y Patrones”, 1999.
- [8]. **Mueller, John.** “Programación Visual C++ .Net”.
- [9]. **Teleformacion.uci.cu.** 2009
http://teleformacion.uci.cu/mod/resource/view.php?id=4598&subdir=/Materiales_Basicos_Conf_7
- [10] **Hernández Orallo, Enrique.** “El Lenguaje Unificado de Modelado (UML)”, 2002.
<http://www.disca.upv.es/enheror/pdf/ActaUML.PDF>
- [11]. **Teleformacion.uci.cu.** 2009
http://teleformacion.uci.cu/mod/resource/view.php?id=4598&subdir=/Materiales_Basicos_Conf_4
- [12]. **Thinkink.** “La Revolución del XML”, 2009.
http://thinkink.es/index.php?option=com_content&view=article&id=158%3Aa-revolucion-del-xml&catid=61%3Aautomatizacion&Itemid=97&lang=es

[13] **Pérez, Renzo**. “Curso Videojuegos”, 2006.

<http://www.mailxmail.com/curso-videojuegos>

[14]. **Universidad del País Vasco**. “Lectura y escrituras de ficheros”, 2006.

<http://usuarios.lycos.es/dhascome/practicass/ProgC/Cap9.pdf>

[15]. **Porteiro Rives, Juan**. “FIFA 08”, MeriStationXBOX 360, 2007.

http://www.meristation.com/v3/des_analisis.php?pic=360&id=cw46fa6a645eb7e&idj=cw4655607c0ad29&idp=&tipo=art&c=1&pos=0

[16]. **Freijomil Viña, Sebastián**. “Second Life”, MeriStationXBOX 360, 2006.

http://www.meristation.com/v3/des_analisis.php?pic=PC&id=cw44f70f76e7747&idj=6123&idp=&tipo=art&c=1&pos=0

[17]. **García Sánchez-Colomer, Miguel**. “Need For Speed Underground”, MeriStationXBOX 360, 2003.

http://www.meristation.com/v3/des_analisis.php?pic=PC&id=4588&idj=4470&idp=&tipo=art&c=1&pos=0

[18]. **Sitio Oficial WoW Europe**. “Guía del juego”, 2008.

<http://www.wow-europe.com/es/info/basics/>

[19]. **Saiz, Aurelio**. “Medall of Honor”, MeriStationXBOX 360, 2002.

http://www.meristation.com/v3/des_analisis.php?pic=PC&idj=109&pes=1&id=2366

[20]. **Pino Garcia, Francisco Gabriel**. “Need For Speed Pro Street”, MeriStation Wii, 2008.

http://www.meristation.com/v3/des_avances.php?id=cw469a024420312&idj=cw465f19764f3c8&pic=Wii

[21]. **MÉNDEZ GONZÁLEZ, RAMÓN**. “PARAISO EXPANDIDO”, MERISTATIONXBOX 360, 2009.

http://www.meristation.com/v3/des_analisis.php?id=cw498ee13056b24&idj=cw49515f45db1ae&pic=PC

[22] **Need for Speed Pro Street**. “La saga se reinventa”, 2008.

<http://elblogdemanu.com/need-for-speed-pro-street-la-saga-se-reinventa/>

[23]. **Garcia Perez-Schofield, Baltasar**. “Zero: una máquina virtual persistente, orientada a objetos pura y basada en prototipos”, 2005.

http://euitio178.ccu.uniovi.es/infoeuitio/IMG/pdf/transpas_zero-2.pdf

[24]. **Arango Bujardón, Aylin y Quintana Echevarria, Yohandri.** “Módulo Registro de Evaluación para el Simulador Quirúrgico”, Universidad de las Ciencias Informáticas, 2008.

[25]. **Rodríguez Noa, Carlos Arcenio y Gómez Finalé, Yordanis.** “Arquitectura de red para la confección de videojuegos multijugadores”, Universidad de las Ciencias Informáticas, 2008.

[26]. **Lombera Rodríguez, Hassán y Pacheco Allende, Alberto Eliseo.** “Edición de sistemas articulados de cuerpos rígidos para las animaciones en los videojuegos”, Universidad de las Ciencias Informáticas, 2008.

[27]. **McGuire, Morgan.** G3D Manual and Library Source Code, 2007.

[28]. **IEEE Standard Glossary of Software Engineering Terminology**, 1990.

<http://www.idi.ntnu.no/grupper/su/publ/ese/ieee-se-glossary-610.12-1990.pdf>

[29]. **Money, CNN.** “Video Game Sales Jump 8 Percent in 2004”, 2004.

<http://money.cnn.com/2005/01/18/technology/gamesales/>

[30]. **Boggs, Wendy y Boggs, Michael.** “UML with Rational Rose 2002”, 2002.

[31]. **Gamma, Erich y Helm, Richard.** “Design Patterns, Elements of Reusable Object-Oriented Software”, 1998.

[32]. **Ulrich, Laurie Ann.** “La biblia del Photoshop 7”.

[33]. **Montero Ayala, Ramón.** “Iniciación y Referencia XML”.

[34]. **Academia de Ciencias de Cuba.** “Metodología del conocimiento científico”, 1975.

[35]. **PubliDirecta.** “Marketing en los videojuegos”, 2002.

http://www.publidirecta.com/marketing/marketing_videojuegos.php

[36]. **ConsumoWeb.** “Heavenly Sword y el marketing de los videojuegos”, 2007.

<http://consumoweb.wordpress.com/2007/08/21/heavenly-sword-y-el-marketing-de-los-videojuegos>

[37]. **Marketing y Éxito**. “El marketing es una variable de éxito empresarial”, 2008.

<http://www.marketingyexito.com/2008/02/27/el-marketing-es-una-variable-de-exito-empresarial>

[38]. **Mailxmail**. “Curso de Videojuegos”, 2006.

<http://www.mailxmail.com/curso/informatica/cursodevideojuegos/>

[39]. **OjGames**. “La Importancia de la Música en los Videojuegos”, 2005.

<http://www.ojgames.com/article/articleview/956512/1/223>

[40]. **Escuela de Ingeniería de Sistemas y Computación**. “Patrones GoF (Gang of Four)”, 2008.

<http://eisc.univalle.edu.co/cursos/web/material/750092M/80/DS2-CLASE-DISENO-GoF.pdf>

GLOSARIO

Aplicación: Es un tipo de programa informático diseñado para facilitarle al usuario la realización de un determinado tipo de trabajo.

Botón: Es una metáfora común utilizada en interfaces gráficas con objetivo similar al de un botón corriente. Los botones suelen ser representados como rectángulos con una leyenda o icono dentro, generalmente con efecto de relieve. Consta de dos imágenes que cambian en dependencia del estado del botón (oprimido o no oprimido). El usuario puede pulsarlo para efectuar acciones indicadas por él.

Componente: Elementos genéricos con una funcionalidad muy concreta, cuya única finalidad es la reutilización.

CPU: Acrónimo utilizado para abreviar Central Processing Unit que significa Unidad de Procesamiento Central. Parte de un computador que controla todas las demás partes. Recoge instrucciones de la memoria y las decodifica. Esto le permite transferir datos hacia o desde la memoria o activar los periféricos para que realicen acciones. También conocido como procesador o microprocesador siendo este último manufacturado con circuitos integrados.

Engine 3D: Conocido también como Game Engine (Motor de Juego), es el software principal (core o núcleo) de un videojuego, o más general de una aplicación gráfica en tiempo real (del inglés real time).

Entorno Virtual: Referido a Mundo virtual. Simulación de mundos o entornos denominados virtuales en los que el hombre interactúa con la máquina en entornos artificiales semejantes a la vida real. Ejemplo de aplicaciones desarrolladas sobre mundos virtuales son los simuladores y los videojuegos.

G3D: Graphics Three Dimensional. Engine 3D disponible como código abierto (*del inglés* Open Source) y libre (*del inglés* free) con licencia BSD desarrollado en el lenguaje de programación C++. G3D Proporciona una base sólida y altamente optimizada para el desarrollo en general de aplicaciones gráficas y extiende las funcionalidades de OpenGL y sockets incluyendo formatos de modelos 3D y bibliotecas utilitarias.

Gestor de base de datos: un sistema gestor de base de datos se define como el conjunto de programas que administran y gestionan la información contenida en una base de datos.

Imagen: Un archivo codificado que, al abrirlo, muestra una representación visual de algo (ya sea fotografía, gráfica o dibujo).

IP: Internet Protocol (Protocolo de Internet), es un protocolo orientado a datos usado tanto por la fuente como por el destino para la comunicación de datos a través de una red de paquetes conmutados.

Interfaz de usuario: Forma en que los usuarios pueden comunicarse con una computadora y comprende todos los puntos de contacto entre el usuario y el equipo.

Juego: Referido a videojuego

Mensaje: En el sentido más general, es el objeto de la comunicación. Está definido como la información que el emisor envía al receptor a través de un canal determinado o medio de comunicación.

Módulo: Es una parte de un programa, o más general un componente de un sistema que tiene una interfaz bien definida para interactuar con otros módulos.

Menú: Es una imagen o conjunto de imágenes que representan diferentes opciones.

Plataforma: Combinación de hardware y software usada para ejecutar aplicaciones.

Sistema: Sistema informático, en general es un conjunto de hardware, software y de un soporte humano. Una aplicación ejecutándose sobre una computadora se considera un sistema.

UML: Unified Modeling Language. Es una notación estándar para modelar objetos del mundo real como primer paso en el desarrollo de programas orientados a objetos. Es un lenguaje para visualizar, especificar, construir y documentar los artefactos de un sistema de software.

Videojuego: Programa informático creado para el entretenimiento, basado en la interacción entre una o varias personas y un aparato electrónico llamado consola (o videoconsola) que ejecuta el videojuego. Estos recrean entornos y situaciones virtuales en los cuales el jugador puede controlar a uno o varios personajes (o cualquier otro elemento de dicho entorno), para conseguir uno o varios objetivos por medio de unas reglas determinadas.

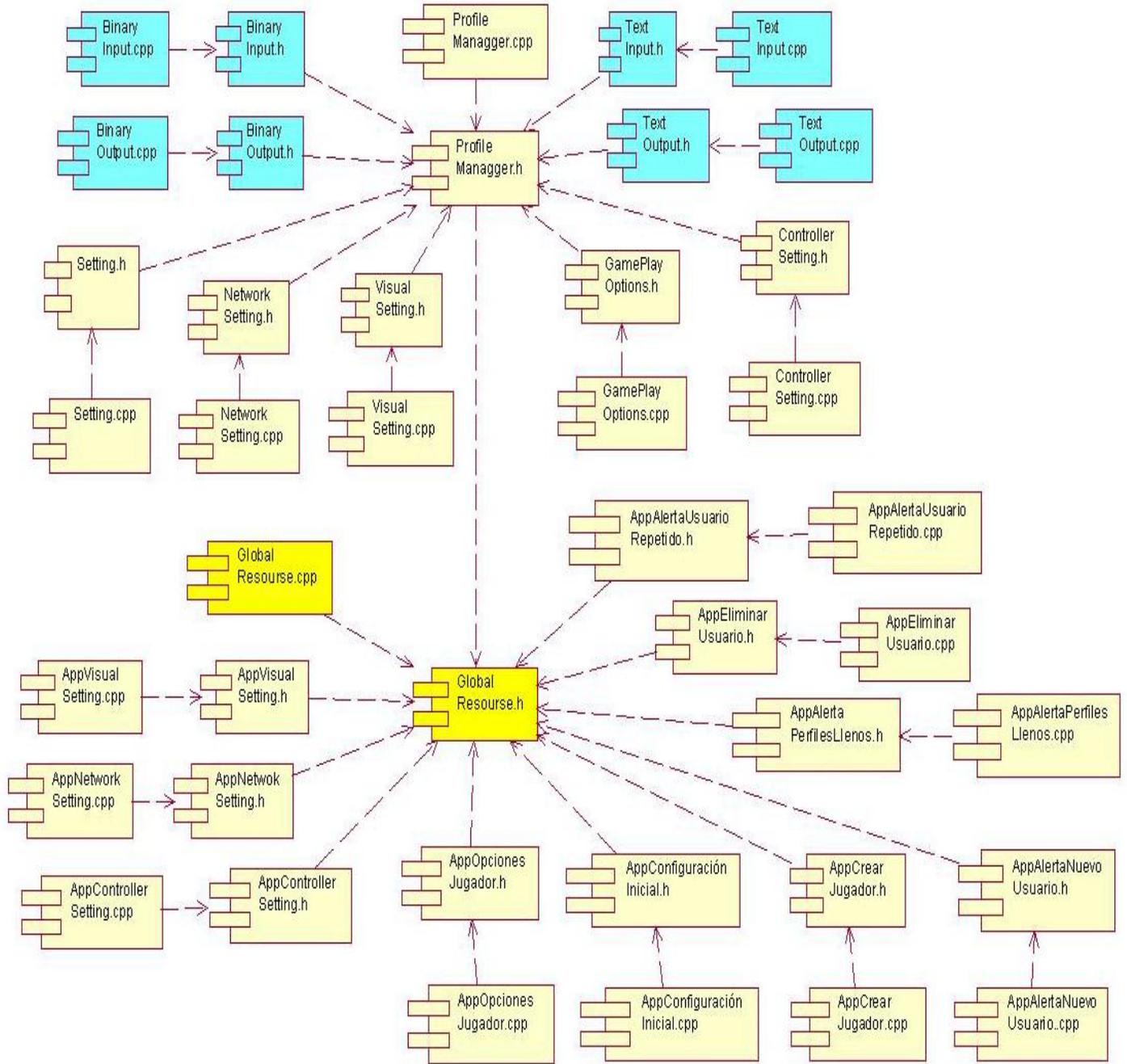
Videojuego Multijugador: Videojuego diseñado para que puedan participar varios jugadores. Más concretamente este término se suele utilizar para definir videojuegos que hacen uso de Internet u otro tipo de red.

Videoconsola: Sistema electrónico de entretenimiento para el hogar que ejecuta juegos electrónicos (videojuegos) que están contenidos en cartuchos o discos ópticos.

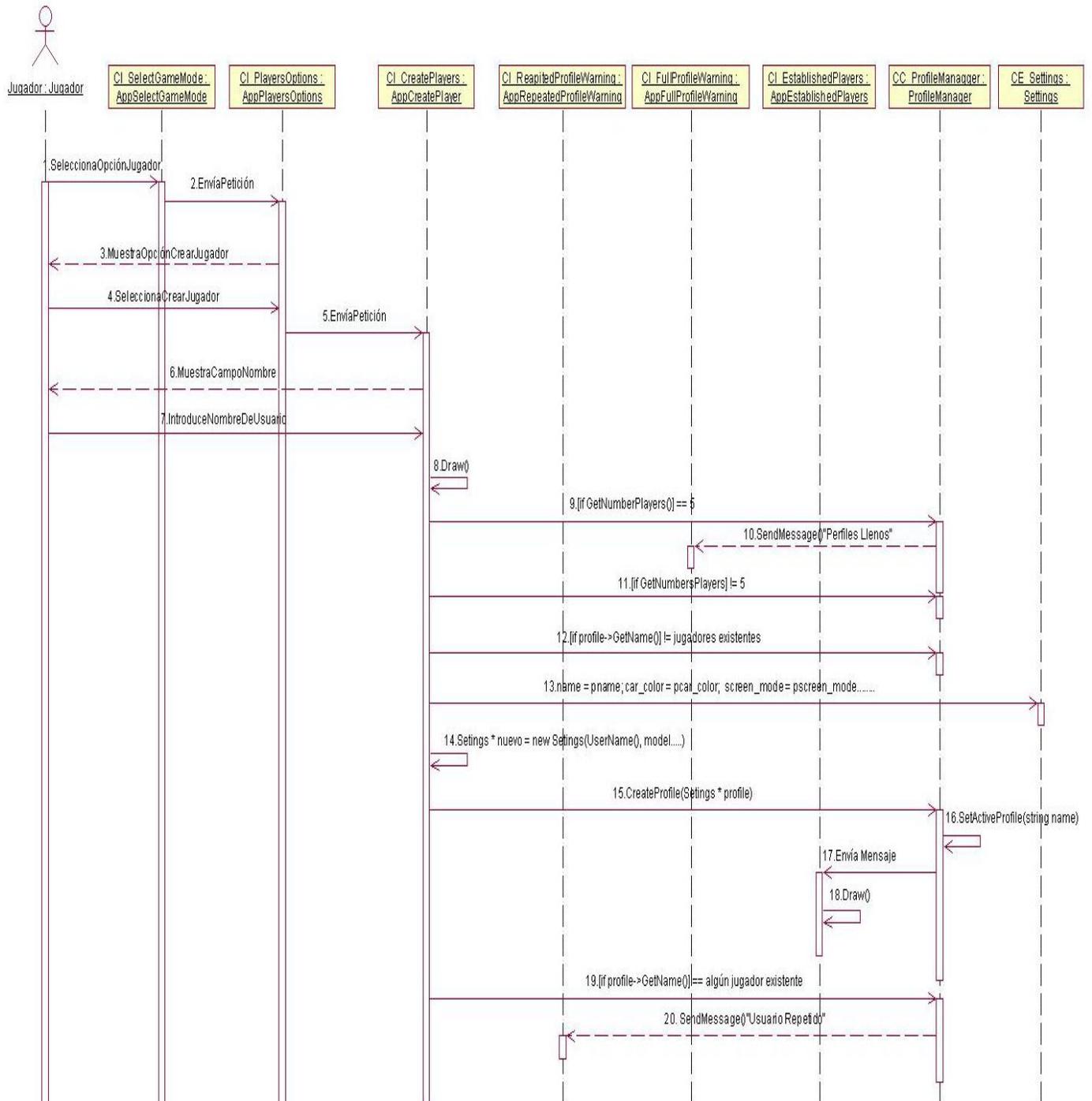
XML: XML (del inglés Extensible Markup Language) desarrollado por World Wide Web Consortium, es una versión simplificada del SGML (del inglés Standard Generalized Markup Language), es considerado un método para introducir datos estructurados en un fichero de texto.

ANEXOS

Anexo I Diagrama de componentes

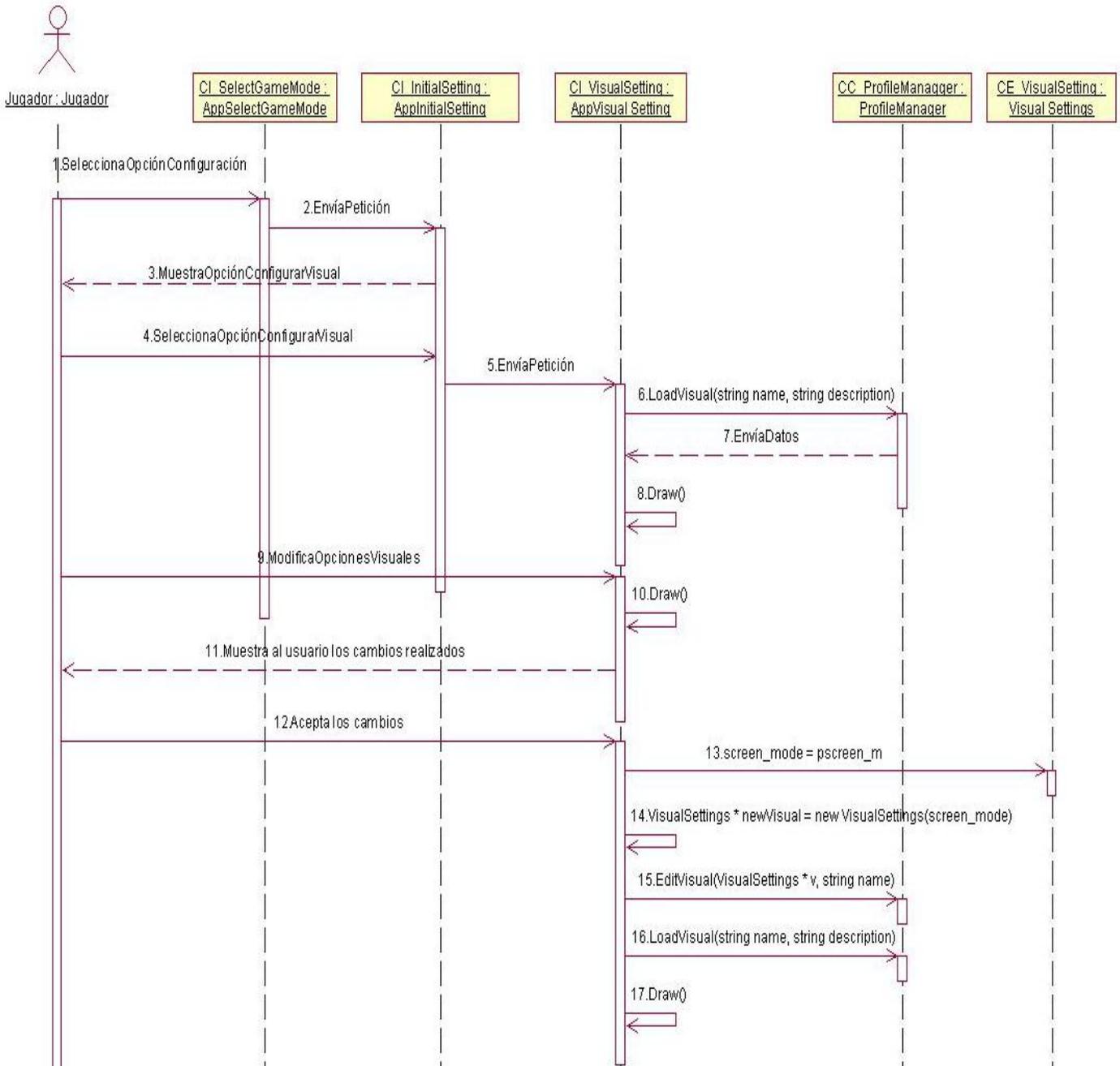


Anexo II Diagrama de Secuencia CU Crear Perfil

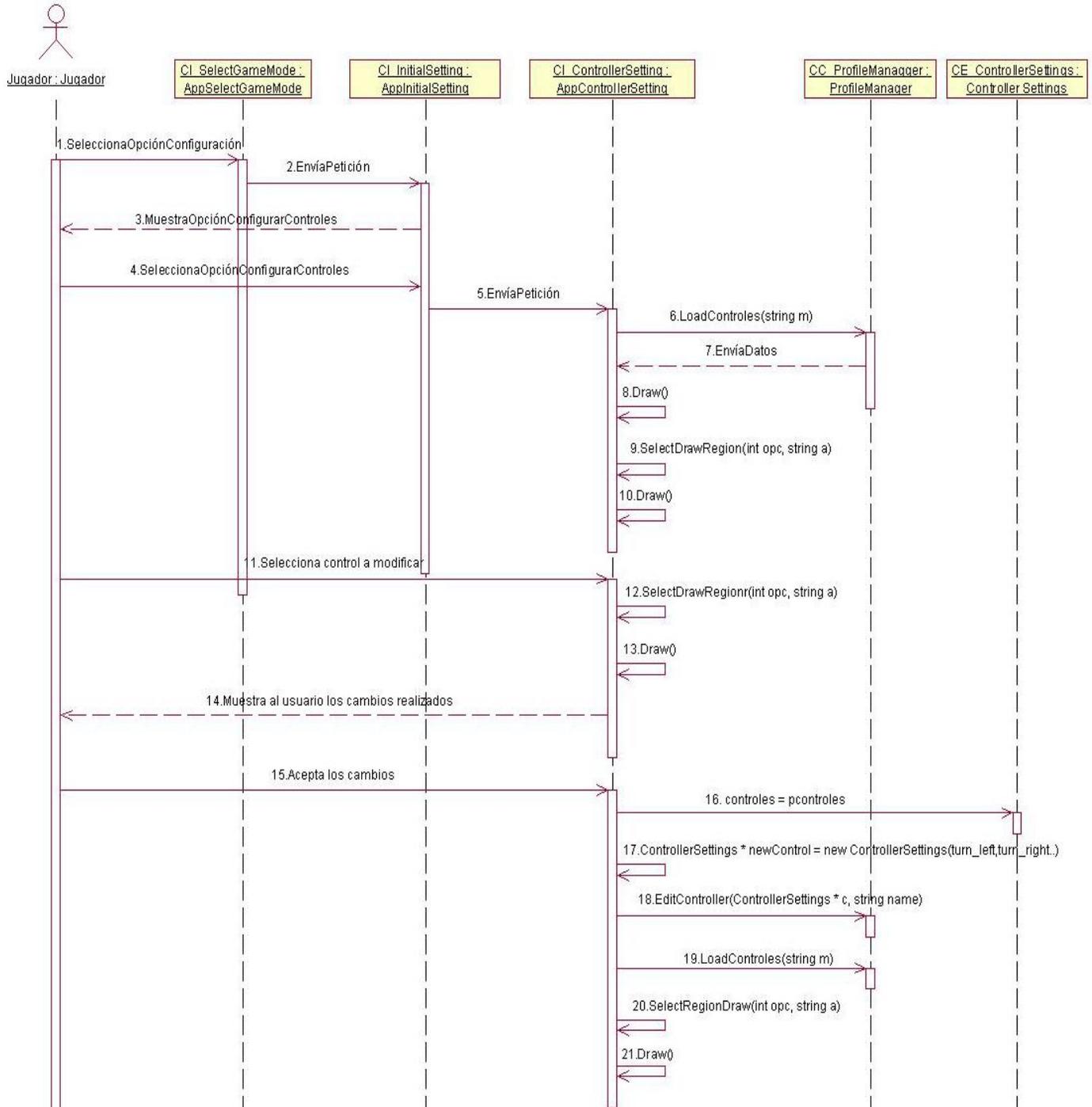


Anexo III Diagramas de Secuencias CU Editar Perfil

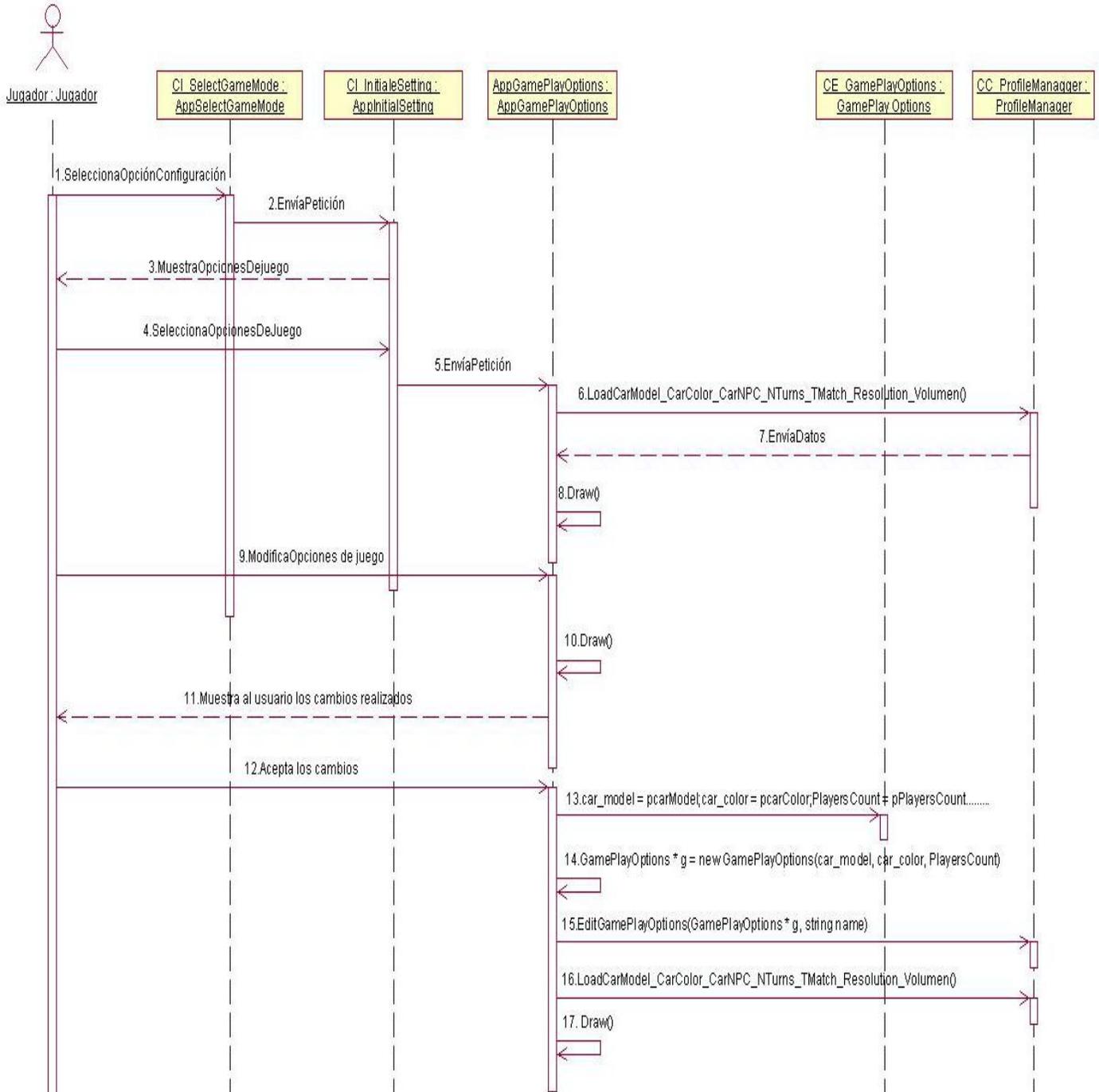
- Configurar Opciones Visuales**



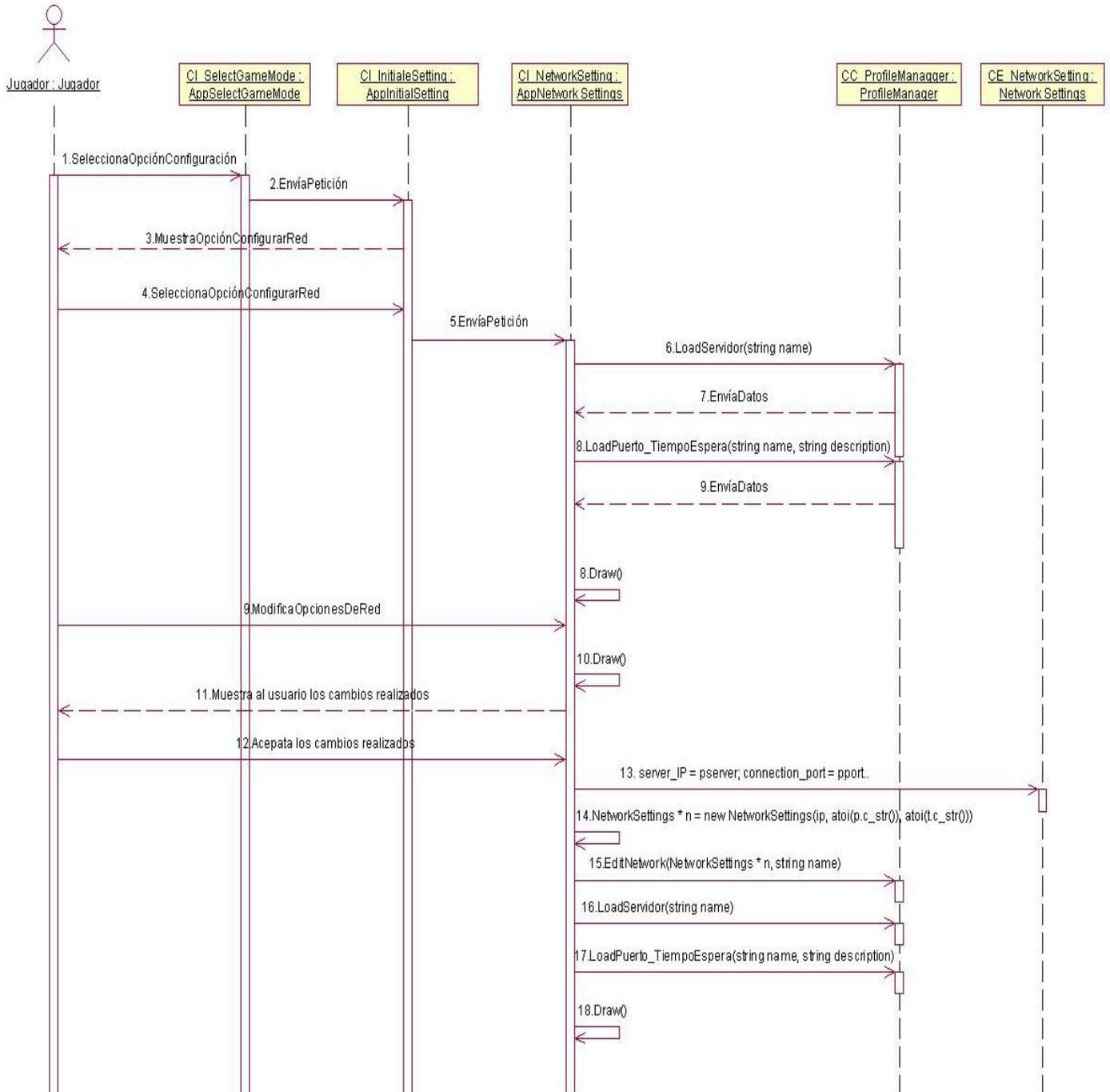
• Configuración de Controles



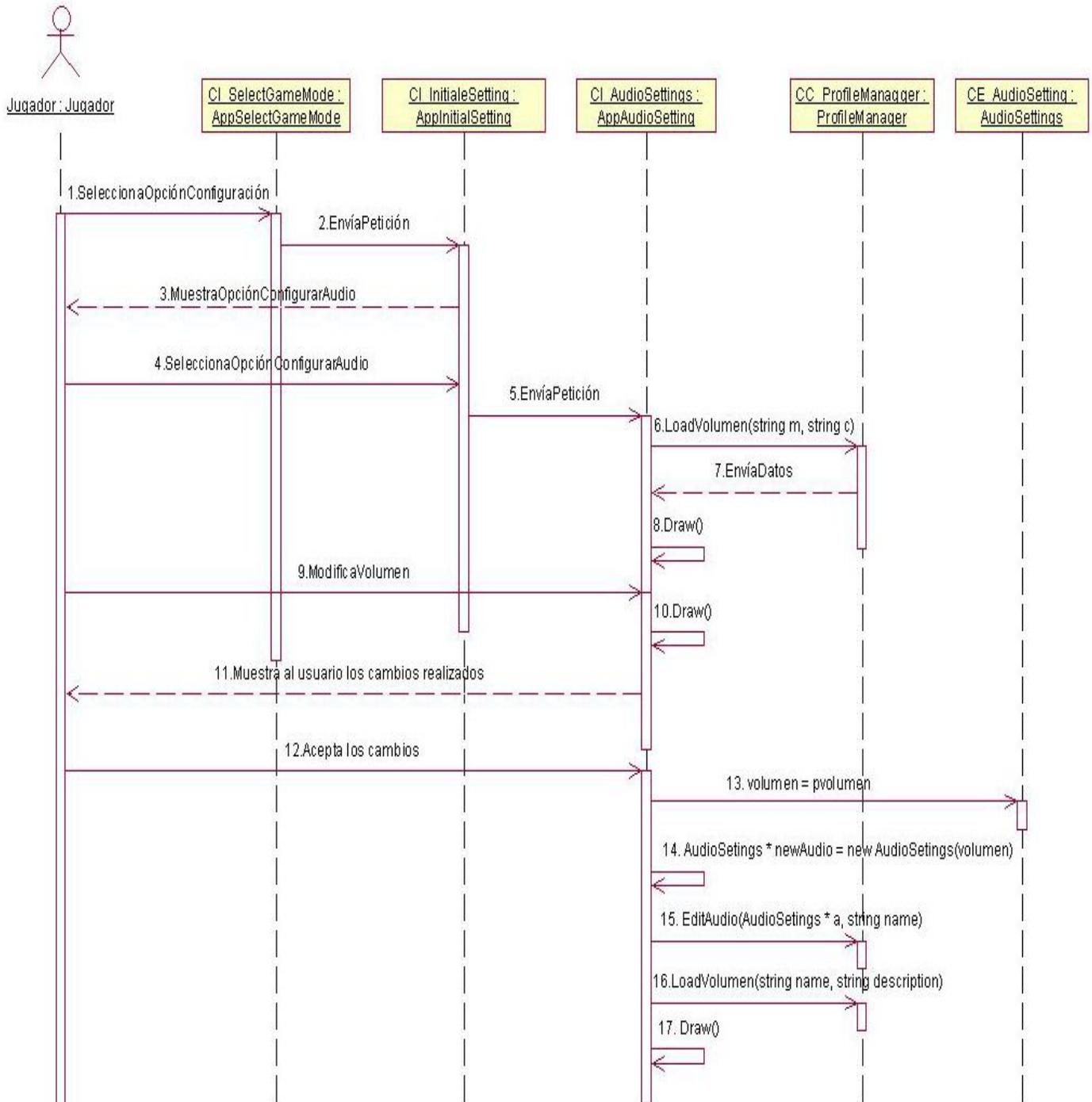
• **Configurar Opciones de Juego**



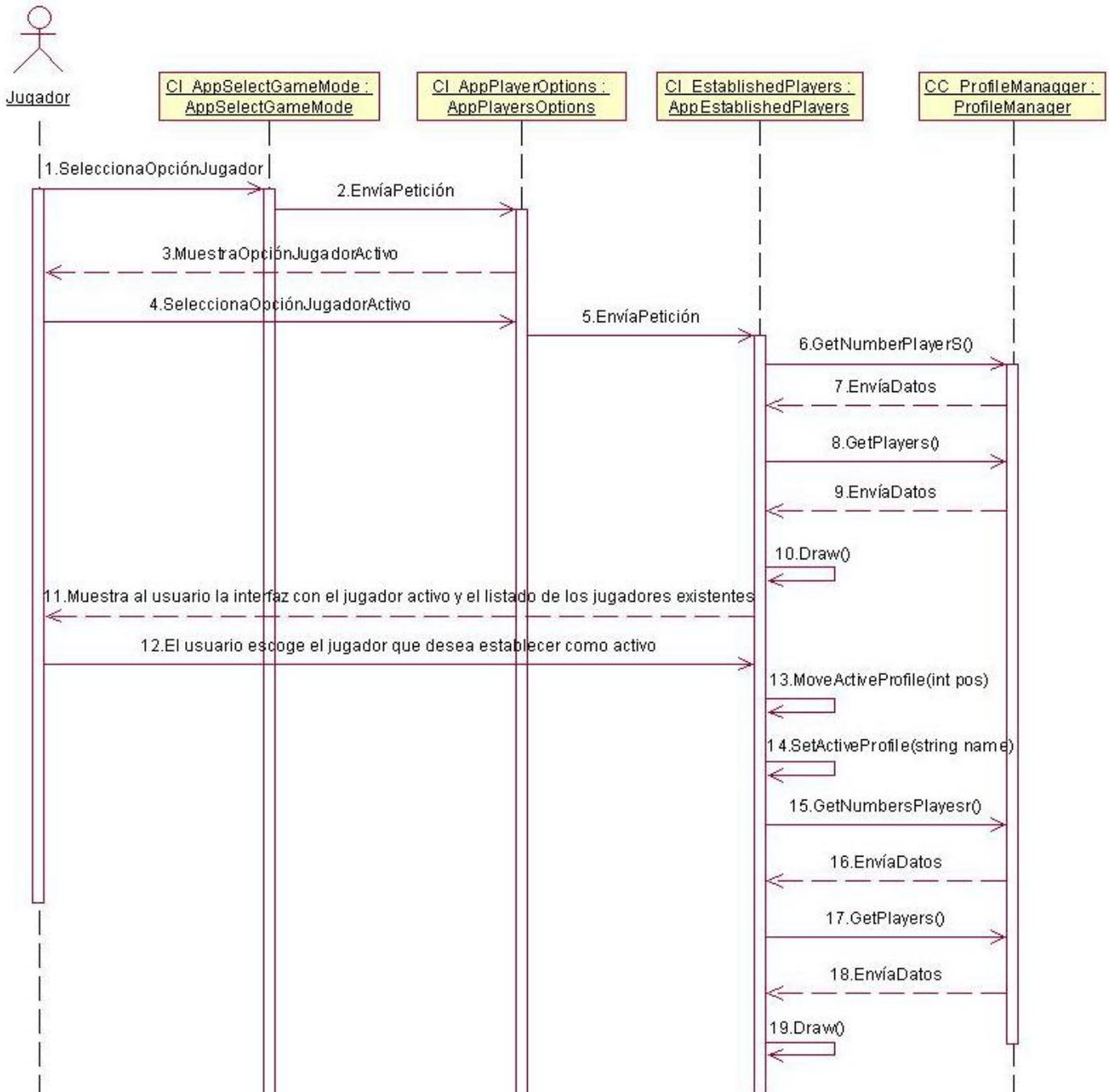
• **Configurar Opciones de Red**



• **Configurar Opciones de Audio**



Anexo IV Diagrama de Secuencia CU Cargar Perfil



Anexo V Diagrama de secuencia CU Eliminar Perfil

