

Universidad de las Ciencias Informáticas

Facultad 8



**Análisis, Diseño e Implementación del Módulo de
Anotación de Juegos del Proyecto Sistema Integral de
Béisbol**

**Trabajo de Diploma para optar por el Título de
Ingeniero en Ciencias Informáticas**

Autores: Daniel Eduardo Guerrero Ramírez

Ramón Ernesto Stevenson Borrell

Tutor: Ing. José Leandro Cabrera Pérez

Ciudad de La Habana, junio del 2009

“Año del 50 Aniversario del Triunfo de la Revolución”

DECLARACIÓN DE AUTORÍA

Declaramos ser autores de la presente tesis y reconocemos a la Universidad de las Ciencias Informáticas los derechos patrimoniales de la misma, con carácter exclusivo.

Para que así conste firmamos la presente a los _____ días del mes de _____ del año _____.

Daniel Eduardo Guerrero Ramírez

Ramón Ernesto Stevenson Borrell

Ing. José Leandro Cabrera Pérez

AGRADECIMIENTOS

De Daniel:

Quisiera agradecer en este momento tan importante de mi vida a todas las personas que de una forma u otra contribuyeron a mi formación como profesional y como persona:

A toda mi familia por el apoyo que siempre me han dado...

A mi mamá Milka por estar siempre presente...

A mis abuelos Edith y Eduardo por ser abuelos y padres a la vez...

A mi tía Magda...

A mi hermana Dayana...

A mi tía María Regla por su apoyo incondicional...

A mi novia Yalida y a su familia por haberme apoyado en todo momento durante estos cinco años...

A mi amigo Arturo Carrillo por su amistad...

A la Revolución Cubana y a su Comandante en Jefe Fidel por darme la oportunidad de estudiar en una universidad como la UCI...

A la UCI por formarnos como ingenieros, en especial a los profesores de la facultad 8...

A todos mis compañeros de grupo y amigos...

A mi tutor José Leandro y a Noel Miño por el apoyo y la ayuda que siempre me dieron...

A mi compañero de Tesis por el resultado obtenido...

Y en general a todos lo que me apoyaron y confiaron en mí.

A todos Muchas Gracias!!!

De Ramón:

Quisiera agradecer de todo corazón a las personas que de una forma u otra han influido positivamente en mi desarrollo como profesional y como ser humano, destacar de ellas a:

Mi familia, por estar siempre a mi lado, apoyarme y guiarme por el buen camino.

Mi madre Cecilia y mi padre Ramón por ser los más grandes maestros y amigos que tengo en la vida, y por hacer de mí el hombre que soy hoy en día.

Mi abuela Martha por tanto cariño y amor, y darme esos consejos que hicieron de mi una mejor persona.

Mi hermano Raúl por su cariño incondicional.

Mi novia Yordanka por apoyarme y ser incondicional.

Mi compañero de Tesis.

Mis amigos del grupo por estar siempre presentes cuando los necesité.

Mi tutor José Leandro y a Noel por brindarme su apoyo, amistad y ayuda incondicional en todo momento.

La Revolución Cubana por hacer de este sueño una realidad.

La UCI por brindarme la oportunidad de crecer profesional y moralmente.

El colectivo de profesores de la Facultad 8 por las enseñanzas y apoyo brindado hacia mi persona.

A todas estas personas mi más profunda muestra de agradecimiento y respeto.

DEDICATORIA

Dedico este trabajo a toda mi familia por haberme ayudado y apoyado siempre. Especialmente a mi mamá Milka, a mis abuelos Edith y Eduardo Manuel, a mi hermana Dayana y a mi tía Magda, por ser las personas más importantes de mi vida...

Daniel

Este trabajo es dedicado a mi familia, pues ha sido mi principal ejemplo de dedicación y devoción por una causa verdadera. En especial mi tesoro más preciado, mi mamá, papá, abuela y mi querido hermano.

Ramón

RESUMEN

En Cuba existe actualmente un sistema de anotación de juegos de béisbol, sin embargo el mismo no cumple con las necesidades y exigencias actuales de este deporte a nivel internacional. Con el objetivo de mejorar la calidad, eficiencia y precisión en la anotación de los juegos de béisbol desarrollados en Cuba, surge la idea de crear un sistema que cumpla con los requisitos necesarios para realizar esta tarea de una forma más eficiente y amena.

El nuevo sistema contará con nuevas funcionalidades y una interfaz de usuario amigable permitiendo una mejor interacción entre el operador y el sistema, además de cumplir con los requisitos y exigencias planteadas por los usuarios a los cuales va dirigido el mismo.

Este trabajo constituye todo el proceso de desarrollo de software realizado para la obtención de un producto final, y en él se justifica y argumenta la elección y utilización de cada una de las herramientas y técnicas de desarrollo utilizadas durante la concepción del mismo.

Palabras Claves

- ❖ Sistema
- ❖ Anotación
- ❖ Requisitos
- ❖ Operador
- ❖ Software

ÍNDICE

Introducción	1
Capítulo 1: Fundamentación Teórica	6
1.1 Introducción.....	6
1.2 Análisis de Otras Soluciones Existentes.....	6
1.3 Argumentación del Objeto de Estudio	9
1.3.1 Descripción General del Problema	9
1.3.2 Argumentación del Problema a Resolver	9
1.4 Conceptos Asociados al Problema	11
1.5 Metodologías	12
1.5.1 Proceso Unificado de Software (RUP), como metodología propuesta para el desarrollo del trabajo.....	13
1.6 Plataforma .NET	16
1.6.1 Plataforma .NET y Mono.....	16
1.6.2 XML.....	18
1.6.3 Servicio Web	19
1.7 Herramientas a utilizar	20
1.8 Lenguajes de Programación.....	23
1.9 Conclusiones	27
Capítulo 2: Características del Sistema	28
2.1 Introducción.....	28
2.2 Problema y situación problemática.....	28
2.3 Propuesta del sistema	29

2.4 Modelo de Negocio	30
2.4.1 Actores del Negocio.....	30
2.4.2 Trabajadores del Negocio	30
2.4.3 Diagrama de casos de Uso del Negocio.....	31
2.4.4 Descripción del Caso de Uso: Anotar Juego.....	31
2.4.5 Diagrama de Actividad: Anotar Juego	34
2.4.6 Diagrama de Objetos	35
2.5 Especificación de los Requisitos del Software	35
2.5.1 Requerimientos funcionales	36
2.5.2 Requerimientos No Funcionales	42
2.6 Casos de Uso del Sistema	43
2.6.1 Actores del Sistema	43
2.6.2 Listado de Casos de Uso del Sistema.....	44
2.6.3 Diagrama de Casos de Uso del Sistema.....	49
2.7 Conclusiones.....	51
Capítulo 3: Análisis y Diseño.....	52
3.1 Introducción.....	52
3.2 Análisis.....	52
3.2.1 Diagramas de Clases del Análisis.....	52
3.2.2 Diagramas de Colaboración	54
3.3 Diseño.....	56
3.3.1 Patrones de diseño.....	56
3.3.2 Diagrama de Clases de Diseño	57

3.3.3 Tratamiento de errores	59
3.3.4 Seguridad	59
3.3.5 Interfaz.....	59
3.3.6 Concepción de la ayuda	59
3.4 Arquitectura.....	60
3.5 Conclusiones.....	61
Capítulo 4: Implementación	62
4.1 Introducción.....	62
4.2 Diagrama de Despliegue.....	62
4.3 Diagrama de Componentes	63
4.4 Estándares de Codificación.....	64
4.5 Encriptación de Datos	66
4.6 Conclusiones.....	66
Conclusiones.....	67
Recomendaciones.....	68
Referencias Bibliográficas	69
Bibliografía Consultada	71
Anexos.....	73
Glosario de Términos.....	128

Índice de Figuras

Figura No. 1: Diagrama de Caso de Uso - Anotar Juego	31
Figura No. 2: Diagrama de Actividad - Anotar Juego	34
Figura No. 3: Diagrama de Objetos del Negocio	35
Figura No. 4: Diagrama de Casos de Usos del Sistema	50
Figura No. 5: Diagrama de Clases del Análisis - CU Autenticar Usuario	52
Figura No. 6: Diagrama de Clases del Análisis - CU Gestionar Alineaciones	53
Figura No. 7: Diagrama de Clases del Análisis - CU Gestionar Jugadas.....	54
Figura No. 8: Diagrama de Colaboración - CU Gestionar Anotador (Agregar Anotador)	55
Figura No. 9: Diagrama de Colaboración - CU Gestionar Anotador (Modificar Anotador).....	55
Figura No. 10: Diagrama de Colaboración - CU Gestionar Anotador (Eliminar Anotador).....	56
Figura No. 11: Diagrama de Clases de Diseño - CU Autenticar Usuario	57
Figura No. 12: Diagrama de Clases de Diseño - CU Gestionar Anotador	58
Figura No. 13: Diagrama de Despliegue.....	63
Figura No. 14: Diagrama de Componentes	64
Figura No. 15: Diagrama de Casos de Usos - Paquete Seguridad.....	73
Figura No. 16: Diagrama de Casos de Usos - Paquete Anotación	74
Figura No. 17: Diagrama de Caso de Uso del Sistema - Autenticar Usuario	74
Figura No. 18: Diagrama de Caso de Uso del Sistema - Gestionar Usuarios.....	76
Figura No. 19: Diagrama de Caso de Uso del Sistema - Gestionar Anotadores	78
Figura No. 20: Diagrama de Caso de Uso del Sistema - Gestionar Árbitros.....	82
Figura No. 21: Diagrama de Caso de Uso del Sistema - Gestionar Posición de los Árbitros	85
Figura No. 22: Diagrama de Caso de Uso del Sistema - Gestionar Alineaciones.....	88
Figura No. 23: Diagrama de Caso de Uso del Sistema - Gestionar Jugadas	92
Figura No. 24: Diagrama de Caso de Uso del Sistema - Gestionar Lanzamientos.....	95
Figura No. 25: Diagrama de Caso de Uso del Sistema - Gestionar Datos Adicionales.....	99
Figura No. 26: Diagrama de Clases del Análisis - CU Gestionar Anotadores.....	102
Figura No. 27: Diagrama de Clases del Análisis - CU Gestionar Datos Adicionales	103
Figura No. 28: Diagrama de Clases del Análisis - CU Gestionar Árbitros	103
Figura No. 29: Diagrama de Clases del Análisis - CU Gestionar Lanzamientos	104
Figura No. 30: Diagrama de Clases del Análisis - CU Gestionar Posición de los Árbitros	105
Figura No. 31: Diagrama de Clases del Análisis - CU Gestionar Usuarios	105
Figura No. 32: Diagrama de Colaboración - CU Autenticar Usuario.....	106
Figura No. 33: Diagrama de Colaboración - CU Gestionar Anotador (Mostrar Anotador)	106
Figura No. 34: Diagrama de Colaboración - CU Gestionar Alineación (Agregar Alineación)	107
Figura No. 35: Diagrama de Colaboración - CU Gestionar Alineación (Mostrar Alineación)	107

Figura No. 36: Diagrama de Colaboración - CU Gestionar Alineación (Modificar Alineación).....	108
Figura No. 37: Diagrama de Colaboración - CU Gestionar Alineación (Eliminar Alineación).....	108
Figura No. 38: Diagrama de Colaboración - CU Gestionar Datos Adicionales (Agregar Datos Adicionales).....	109
Figura No. 39: Diagrama de Colaboración - CU Gestionar Datos Adicionales (Modificar Datos Adicionales).....	109
Figura No. 40: Diagrama de Colaboración - CU Gestionar Datos Adicionales (Eliminar Datos Adicionales).....	110
Figura No. 41: Diagrama de Colaboración - CU Gestionar Datos Adicionales (Mostrar Datos Adicionales).....	110
Figura No. 42: Diagrama de Colaboración - CU Gestionar Árbitro (Agregar Árbitro).....	111
Figura No. 43: Diagrama de Colaboración - CU Gestionar Árbitro (Modificar Árbitro).....	111
Figura No. 44: Diagrama de Colaboración - CU Gestionar Árbitro (Eliminar Árbitro).....	112
Figura No. 45: Diagrama de Colaboración - CU Gestionar Árbitro (Mostrar Árbitro).....	112
Figura No. 46: Diagrama de Colaboración - CU Gestionar Jugadas (Agregar Jugada).....	113
Figura No. 47: Diagrama de Colaboración - CU Gestionar Jugadas (Actualizar Jugada).....	113
Figura No. 48: Diagrama de Colaboración - CU Gestionar Jugadas (Eliminar Jugada).....	114
Figura No. 49: Diagrama de Colaboración - CU Gestionar Jugadas (Mostrar Jugada).....	114
Figura No. 50: Diagrama de Colaboración - CU Gestionar Lanzamientos (Agregar Lanzamiento).....	115
Figura No. 51: Diagrama de Colaboración - CU Gestionar Lanzamientos (Modificar Lanzamiento).....	115
Figura No. 52: Diagrama de Colaboración - CU Gestionar Lanzamientos (Eliminar Lanzamiento).....	116
Figura No. 53: Diagrama de Colaboración - CU Gestionar Lanzamientos (Mostrar Lanzamiento).....	116
Figura No. 54: Diagrama de Colaboración - CU Gestionar Posición Árbitro (Agregar Posición Árbitro).....	117
Figura No. 55: Diagrama de Colaboración - CU Gestionar Posición Árbitro (Actualizar Posición Árbitro).....	117
Figura No. 56: Diagrama de Colaboración - CU Gestionar Posición Árbitro (Eliminar Posición Árbitro).....	118
Figura No. 57: Diagrama de Colaboración - CU Gestionar Usuarios (Agregar Usuario).....	118
Figura No. 58: Diagrama de Colaboración - CU Gestionar Usuarios (Modificar Usuario).....	119
Figura No. 59: Diagrama de Colaboración - CU Gestionar Usuarios (Eliminar Usuario).....	119
Figura No. 60: Diagrama de Clases de Diseño - CU Gestionar Datos Adicionales.....	120
Figura No. 61: Diagrama de Clases de Diseño - CU Gestionar Árbitros.....	121
Figura No. 62: Diagrama de Clases de Diseño - CU Gestionar Jugadas.....	122
Figura No. 63: Diagrama de Clases de Diseño - CU Gestionar Lanzamientos.....	123
Figura No. 64: Diagrama de Clases de Diseño - CU Gestionar Posición Árbitros.....	124
Figura No. 65: Diagrama de Clases de Diseño - CU Gestionar Usuarios.....	125
Figura No. 66: Diagrama de Clases de Diseño - CU Gestionar Alineaciones.....	126
Figura No. 67: Diagrama de Clases de Diseño.....	127

Índice de Tablas

Tabla No. 1: Justificación de los Actores del Negocio 30

Tabla No. 2: Justificación de los Trabajadores del Negocio 30

Tabla No. 3: Descripción del caso de Uso - Anotar Juego 33

Tabla No. 4: Justificación de los actores del sistema..... 43

Tabla No. 5: CU-1 - Autenticar Usuario 44

Tabla No. 6: CU-2 - Gestionar Usuario 45

Tabla No. 7: CU-3 - Gestionar Anotadores 45

Tabla No. 8: CU-4 - Gestionar Árbitros 45

Tabla No. 9: CU-5 - Gestionar Posición Árbitros..... 46

Tabla No. 10: CU-6 - Gestionar Alineaciones 47

Tabla No. 11: CU-7 - Gestionar Jugadas..... 47

Tabla No. 12: CU-8 - Gestionar Lanzamientos..... 48

Tabla No. 13: CU-9 - Gestionar Datos Adicionales..... 48

Tabla No. 14: Descripción del Caso de Uso - Autenticar Usuario..... 75

Tabla No. 15: Descripción del Caso de Uso - Gestionar Usuarios..... 78

Tabla No. 16: Descripción del Caso de Uso - Gestionar Anotadores 81

Tabla No. 17: Descripción del Caso de Uso - Gestionar Árbitros..... 85

Tabla No. 18: Descripción del Caso de Uso - Gestionar Posición de los Jueces 88

Tabla No. 19: Descripción del Caso de Uso - Gestionar Alineaciones..... 92

Tabla No. 20: Descripción del Caso de Uso - Gestionar Jugadas..... 95

Tabla No. 21: Descripción del Caso de Uso - Gestionar Lanzamientos..... 98

Tabla No. 22: Descripción del Caso de Uso - Gestionar Datos Adicionales..... 102

Introducción

En Cuba el béisbol es pasión, pues es el deporte nacional. A lo largo de los años se han obtenido muchísimos logros en este deporte, lo que ha llevado a reconocer a Cuba como una de las principales potencias del béisbol amateur en el mundo, a pesar de ser un país con poco desarrollo económico. El Instituto Nacional de Educación Física y Recreación (INDER) y la Federación Cubana de Béisbol Amateur (FCBA) han sido protagonistas de los logros alcanzados por este deporte. La FCBA es la encargada de realizar numerosos procesos y tareas que se cumplen durante un juego de béisbol. La anotación de los juegos es uno de estos procesos.

Desde el año 1977 la FCBA puso en práctica un sistema de anotación de juegos de béisbol denominado "Béisbol", el cual resulta de gran importancia, pues en él se recopilan una serie de datos y estadísticas necesarias para el desarrollo de este deporte. Sin embargo, con el desarrollo de las nuevas tecnologías surge también la necesidad de crear nuevos sistemas que sean capaces de recopilar una mayor cantidad de datos e información, que puedan ser almacenados para su posterior utilización, deviniendo en un uso óptimo de la información generada ya que permite brindar un grupo de estadísticas del juego en tiempo real y disponibles en cualquier momento.

De esta manera surge la idea de crear un sistema, que aparte de cumplir con todas las funcionalidades requeridas por la FCBA sea multiplataforma, debido a la tendencia actual que tiene el país hacia la migración al Software Libre, puesto que no está sujeto a las leyes del bloqueo económico impuesto por los Estados Unidos.

Situación Problemática

El sistema de anotación que existe actualmente tiene una serie de dificultades, las cuales son los aspectos fundamentales de la situación problemática planteada. Estos aspectos son los siguientes:

- ✓ El sistema actual está desarrollado en MS-DOS y no cuenta con una interfaz de usuario amigable que cumpla con las funcionalidades requeridas para un sistema de este tipo.
- ✓ Los juegos no se anotan lanzamiento por lanzamiento sino jugada por jugada, por lo que no se recopilan una serie de datos importantes del juego.

- ✓ La necesidad de un sistema que permita al anotador hacer un trabajo más rápido y eficiente; y que pueda aprender a manipularlo con gran facilidad.

Estos aspectos hacen necesaria una actualización de este sistema para contribuir al desarrollo del béisbol en Cuba.

El **problema a resolver** no es más que la no existencia de un sistema de anotación de juegos en el béisbol que se ajuste a las necesidades actuales de Cuba.

El **objeto de estudio** es el proceso de recopilación de datos estadísticos por parte de la FCBA en los juegos de béisbol.

El **objetivo general** del trabajo es realizar la implementación de un sistema de anotación de juegos para el béisbol en Cuba. Se cuenta además con varios **objetivos específicos** los que permitirán darle cumplimiento al objetivo general, estos son los siguientes:

- ✓ Investigar otros sistemas de anotación existentes en Cuba y en otros países del mundo.
- ✓ Identificar las características positivas de cada una de las aplicaciones estudiadas para tenerlas en cuenta a la hora de desarrollar el sistema.
- ✓ Realizar un estudio sobre las posibles herramientas a utilizar.
- ✓ Caracterizar las herramientas de desarrollo a utilizar, teniendo en cuenta las ventajas y desventajas de las mismas, así como su influencia en el desarrollo de la aplicación.
- ✓ Realizar análisis y diseño del sistema.

El **campo de acción** se encuentra enmarcado en la anotación de juegos de béisbol en la FCBA.

La **idea a defender** en esta investigación está expuesta a continuación:

Si se logra desarrollar un sistema que cuente con una interfaz visual agradable para el usuario, que cumpla con las funcionalidades necesarias para llevar a cabo una anotación con eficiencia, que se pueda aprender a manipular fácilmente y que se pueda anotar lanzamiento por lanzamiento, se podrá obtener

entonces una mayor cantidad de datos estadísticos e información de los juegos de béisbol y el trabajo de los anotadores se hará más fácil y eficiente.

Las **tareas** a desarrollar para cumplir los objetivos planteados son las siguientes:

- ✦ Realizar un estudio y comprensión del negocio del sistema actual de anotación de juegos de béisbol en Cuba.
- ✦ Estudiar y seleccionar la metodología y las herramientas a utilizar.
- ✦ Realizar entrevistas con los especialistas y personas a las cuales está destinado el software.
- ✦ Estudiar y enunciar las ventajas y desventajas del lenguaje de programación a utilizar.
- ✦ Investigar sobre el surgimiento y desarrollo de otros sistemas de anotación en el mundo.
- ✦ Realizar análisis, diseño e implementación del módulo en cuestión.

En el desarrollo de todo el proceso de investigación se emplearon una serie de **métodos científicos**. Estos **métodos de investigación** se clasifican en **teóricos** y **empíricos**.

Los **métodos teóricos** son los que permiten estudiar las características del objeto de investigación que no son observables directamente, facilitan la construcción de modelos e hipótesis de investigación y crean las condiciones para ir más allá de las características fenomenológicas y superficiales de la realidad.

Los métodos teóricos que se utilizaron fueron los siguientes:

- El método **histórico-lógico** para realizar un estudio sobre las diferentes aplicaciones de este tipo que existen en nuestro país y en el mundo.
- El método **hipotético-deductivo**, permite a través de la idea a defender, llegar a nuevos conocimientos, siguiendo una serie de pasos lógicos, que nos llevan al cumplimiento de los objetivos.

Los **métodos empíricos** son los que describen y explican las características fenomenológicas del objeto, representan un nivel de la investigación cuyo contenido procede de la experiencia y es sometido a cierta elaboración racional.

Los métodos empíricos que se utilizaron fueron los siguientes:

- La **observación** del entorno del problema en el que se desarrolla la investigación para obtener elementos importantes que sirvan de aporte a lo que se investiga.
- La **entrevista** con los diferentes miembros de la FCBA a los cuales va dirigida la aplicación para conocer sus criterios y opiniones y comprender la situación exacta del problema existente.

La realización de este trabajo es de gran importancia para el desarrollo del béisbol en Cuba. Contribuye a mejorar el sistema actual de anotación de juegos y con ello a obtener mayor cantidad de información que posteriormente pueda ser utilizada y almacenada para el trabajo estadístico, que es uno de los más importantes en este deporte.

Como se conoce, Cuba es un país que no posee un amplio desarrollo económico, y para estar a la altura de los países que si poseen una gran economía y dedican millones de dólares al desarrollo de este deporte, es necesario realizar diferentes inversiones en la automatización de las aplicaciones que se utilizan con estos fines. La solución propuesta reduce al país las inversiones en softwares de este tipo y en su lugar se podría invertir en otras necesidades del deporte.

Para llevar a cabo el desarrollo de este trabajo y realizar un mejor estudio del mismo, se ha dividido en varios capítulos.

Capítulo 1: Fundamentación teórica.

Se incluye un estudio del estado del arte del tema tratado, los objetivos y conceptos relacionados con el dominio del problema en cuestión. Aquí se encuentran relacionadas una serie de herramientas y metodologías estudiadas para la realización del trabajo.

Capítulo 2: Características del sistema.

Se explica la solución que proponen los desarrolladores del trabajo, de acuerdo a la situación problemática existente. Se realiza el Modelo de Negocio, y se lleva a cabo un levantamiento de requisitos con el objetivo de cumplir las exigencias planteadas por el cliente.

Capítulo 3: Análisis y Diseño.

Como su nombre lo indica, en este capítulo se realizan todas las actividades correspondientes al flujo de trabajo análisis y diseño, se generan los artefactos pertinentes y se define la arquitectura y los patrones de diseño que se van a utilizar, con el objetivo de fortalecer las bases para la implementación del sistema propuesto.

Capítulo 4: Implementación.

Aquí se muestran los artefactos generados en el flujo de trabajo implementación, los estándares de codificación que se utilizan en la programación del sistema y el algoritmo de encriptación escogido por los desarrolladores, permitiendo continuar la construcción del sistema propuesto obteniéndose una primera versión de la aplicación.

Capítulo 1: Fundamentación Teórica

1.1 Introducción

Para el desarrollo del béisbol es importante que se lleven a cabo una serie de procesos, y que estos procesos se puedan efectuar con un alto nivel de calidad. Dentro de estos procesos se encuentra la anotación de los juegos. De la eficacia del proceso de anotación depende la veracidad de las estadísticas de este deporte. Para ello es importante automatizar la mayoría de estos procesos de forma tal que los mismos se ejecuten con rapidez y eficiencia.

En este capítulo se exponen una serie de conceptos fundamentales para una mejor comprensión del problema y el desarrollo del trabajo. Se explica en detalles la situación problemática y se realiza un estudio de las soluciones que existen hasta el momento sobre este tema. Se realizará además un estudio de varias metodologías de desarrollo y una comparación entre las diferentes herramientas que pueden servir para la solución del problema a resolver con el objetivo de escoger la más adecuada para la implementación del sistema que se propone.

1.2 Análisis de Otras Soluciones Existentes

Realizando un estudio sobre diferentes sistemas de anotación en otros países se encontró una serie de características y funcionalidades que pueden contribuir al desarrollo de una aplicación que realice el proceso de anotación de una manera más óptima.

Sistema BallScore

Un sistema denominado “BallScore” se dedica a la anotación de juegos de béisbol, desarrollado por la compañía James Habel Software de Estados Unidos. Este sistema fue implementado en el lenguaje de programación Delphi y ofrece varias funcionalidades que ayudan al anotador del partido a desempeñar su labor de una forma eficiente y amena. “BallScore” engloba una serie de características positivas que se deben tener en cuenta en la concepción del sistema que será desarrollado por los autores de este trabajo, dentro de estas características se encuentran:

- Interfaz de Usuario Amigable.

- Manejo eficiente de las estadísticas, pues son mostradas en cada momento al anotador.
- Se facilita el proceso de codificación de los juegos.
- Se toman en cuenta los lanzamientos de los pitchers y se lleva la cantidad de lanzamientos que realiza.

Como se puede apreciar este sistema contiene una serie de elementos que agilizan y facilitan el trabajo de los anotadores en los partidos, sin embargo presenta algunas deficiencias en el proceso de codificación, ya que no son tomadas en cuenta una serie de jugadas sobre determinada acción del bateador y puesto que cada jugada en el béisbol tiene su codificación generaría un error en la anotación. También genera una serie de reportes sobre las estadísticas emanadas del juego, entre ellas la anotación completa del partido, listado de los equipos con todos sus integrantes y sus estadísticas durante el encuentro, entre otras. (1)

Sistema para el Manejo de un Tablero Electrónico de Béisbol

En México se emplea un sistema denominado: Sistema para el Manejo de un Tablero Electrónico de Béisbol, el cual se encarga de anotar los juegos de béisbol en tiempo real, este sistema además como su nombre lo indica permite manejar la pizarra electrónica de forma automática. También cuenta con un servicio web que permite a las personas seguir el partido de béisbol desde cualquier lugar en tiempo real.

Este sistema corre sobre la plataforma Windows y cuenta con una serie de formularios que tienen una interfaz amigable y de fácil uso para el anotador. El primer formulario permite cargar desde una base de datos a los equipos y jugadores que participarán en el encuentro y se insertan los datos del partido, tales como: fecha y hora. En otro formulario se selecciona la alineación de cada equipo y el equipo que será home club o visitante.

Ya en el partido se anota el juego, donde se van reflejando las estadísticas de cada jugador y estos datos pueden ser vistos en una página web. Por último en otro formulario se pueden recoger el resultado del partido y de forma más específica las estadísticas de cada jugador. (2)

Sistema de anotación cubano “Béisbol”

En Cuba se utiliza actualmente un sistema que lleva por nombre Béisbol, el cual permite optimizar un poco el proceso de anotación, que hasta el momento de su creación se desarrollaba sobre un papel y era un poco más engorroso. Ahora se cuenta con toda la información del partido jugada por jugada de manera digital, la cual se puede obtener en tres formatos diferentes: en .htm, para página web, en .doc, para imprimir sobre papel y en .list, para el trabajo con ficheros .bei, los cuales son compilados posteriormente. Este sistema fue desarrollado por integrantes del Instituto de Educación Física y Recreación (INDER), en el año 1977 sobre MS-DOS ya que en Cuba la tecnología con que se contaba era muy pobre.

Sin embargo este sistema no tiene una interfaz de usuario amigable y resulta difícil el manejo del mismo para el anotador. Con el desarrollo tecnológico y la aparición de Windows con un sistema de ventanas muy fácil de usar que brinda una serie de facilidades para el manejo de un sistema como este, es necesaria la realización de una versión actualizada de la aplicación la cual hasta el momento no se ha realizado. Se hizo una actualización en 1994 pero fue para corregir algunos problemas y agregar algunas funcionalidades sobre la misma plataforma de MS-DOS.

Existe una copia de la aplicación en cada uno de los estadios del país. Durante el desarrollo de un juego se llevan todas las jugadas del encuentro a la misma y se envían mediante ficheros .bei al estadio Latinoamericano donde se procesa toda la información y se genera una página .htm que se muestra en un sitio web para que los aficionados puedan seguir el juego mientras se efectúa. Este proceso es lento y muchas veces no se tiene información del partido hasta que no se termina el inning en desarrollo o durante varios inning. Cuando se termina el juego se compilan todos los ficheros que se obtuvieron en el estadio Latinoamericano y se envían de nuevo a cada uno de los estadios para actualizar la aplicación de forma independiente.

Muchos datos estadísticos se pierden, pues la información de cada lanzamiento no es registrada, debido a que la inserción de información estadística como se ha expresado en varias ocasiones solo ocurre cuando se completa una jugada.

1.3 Argumentación del Objeto de Estudio

1.3.1 Descripción General del Problema

Para llevar a cabo el proceso de anotación en los partidos de béisbol, en Cuba se utiliza un sistema de anotación y gestión de juegos de béisbol denominado “Béisbol”, concebido en 1977 y desarrollado en MS-DOS. Está provisto de una interfaz gráfica por consola dificultando así la interacción del anotador con el sistema, pero en el momento de su creación constituyó un proyecto innovador de gran ayuda para las tareas de gestión y anotación de los juegos de beisbol, que se realiza en papel y dificulta el trabajo de los anotadores y la gestión de las estadísticas que se generan a partir de la información recogida en cada partido. La anotación de los juegos como se explica anteriormente es en papel, el anotador debe tener un amplio conocimiento del Sistema Codificado de Anotación Cubano para desempeñar su tarea y paralelamente insertar en el sistema “Beisbol” cada jugada durante el encuentro, por lo que se realiza el mismo trabajo en dos ocasiones. Este sistema introdujo la posibilidad de guardar estadísticas en formato digital, sin embargo se pierden algunas estadísticas del juego pues los lanzamientos no se registran en el mismo.

1.3.2 Argumentación del Problema a Resolver

El sistema actual de anotación de juegos de béisbol en Cuba corre sobre MS-DOS, lo que trae consigo que se deriven una serie de dificultades con el uso del mismo por la interfaz que posee, además es necesario que el anotador conozca el sistema de codificación cubano, pues las jugadas se insertan en la aplicación mediante un código. Todos los datos del encuentro que se vayan a recoger están codificados, el nombre de los jugadores, los árbitros, anotadores oficiales, comisarios técnicos y las jugadas que pueden ocurrir en un partido.

Una vez comenzado el juego, cuando comparece al bate cada jugador, aparece una pantalla que recrea una serie de acciones posibles a la ofensiva y defensiva que describen una jugada. A continuación se muestra el proceso por el que transcurre la anotación. Se muestra una pantalla que contiene 7 casillas:

En la casilla número 1 se pone el número del jugador que está consumiendo su turno al bate.

En la casilla número 2 la acción ofensiva del bateador.

En la casilla número 3 posiciones defensivas que intervienen en la acción ofensiva del bateador.

En la casilla número 4 las acciones defensivas ocurridas sobre la acción ofensiva del bateador.

En la casilla número 5 el avance del corredor a primera base.

En la casilla número 6 el avance del corredor a segunda base.

En la casilla número 7 el avance del corredor a tercera base.

Todas las acciones descritas anteriormente son insertadas en el sistema mediante un código equivalente a cada una de ellas, por lo que las combinaciones de números a tener en cuenta por el anotador para llevar a cabo la anotación de cada jugada en el partido pueden ser relativamente complejas y debe consultar el Sistema Codificado de Anotación o conocer a la perfección el mismo. Los lanzamientos que lleva a cabo cada pitcher en el encuentro no son tomados en cuenta por el sistema, lo que imposibilita almacenar una serie de estadísticas sobre los pitcher que intervienen en el partido.

La aplicación cuenta con grandes limitaciones tecnológicas y debido al desarrollo progresivo de los sistemas informáticos no se hace un buen uso de todas las potencialidades que brindaría un sistema de anotación de este tipo, pues de él se derivaría la posibilidad de contar con otra serie de servicios, como por ejemplo un sistema para mostrar las incidencias del juego en tiempo real, del cual el sistema de anotación sería la base para su funcionamiento.

Actualmente se llevan los juegos en vivo a través de un sitio web, pero por la complejidad del sistema y del funcionamiento del mismo, en ocasiones no se conoce lo que está sucediendo en tiempo real, pues los usuarios deben esperar a que transcurra un inning completo, para saber lo que sucede, y a veces hasta varios inning pues el sistema no envía los datos en tiempo real.

1.4 Conceptos Asociados al Problema

Béisbol: Nombre del sistema actual cubano que se encarga de la gestión y anotación de los juegos de béisbol.

Anotación de juegos: Es un registro detallado de todos los lanzamientos y jugadas que ocurren durante un juego de béisbol.

Sistema Codificado de Anotación: Es un sistema que se emplea para la anotación de los juegos de béisbol en el cual cada jugada tiene una equivalencia de un código.

Árbitros: Son las personas responsables de conducir el juego de béisbol de acuerdo con las reglas oficiales y de mantener la disciplina y el orden en el terreno durante el juego.

Anotador Oficial: Es el encargado de observar el juego de béisbol para tomar las decisiones oficiales concernientes a la aplicación de las reglas de anotación, tales como si el avance de un bateador a primera base es consecuencia de un hit o un error.

Comisarios técnicos: Son los máximos responsables del juego de béisbol. Sobre ellos recae la responsabilidad de decidir en caso de que haya una reclamación por parte de uno de los dos equipos que se enfrentan en ese momento.

Código: Secuencia de números que se utiliza para identificar una jugada en el béisbol.

Estadísticas: Es un registro acumulativo de todos los récords de bateo, fildeo, corrido de bases y récords de lanzadores, para cada jugador que aparezca en un juego de campeonato de la liga o de post temporada.

1.5 Metodologías

Existen varias metodologías de desarrollo que se aplican a la hora de desarrollar algunos sistemas y aplicaciones. A continuación se enunciarán algunas de ellas.

Programación Extrema (eXtreme Programming)

La programación extrema o eXtreme Programming (XP) es un enfoque de la ingeniería de software formulado por Kent Beck, autor del primer libro sobre la materia, *Extreme Programming Explained: Embrace Change* (1999). Es el más destacado de los procesos ágiles de desarrollo de software. Al igual que estos, la programación extrema se diferencia de las metodologías tradicionales principalmente en que pone más énfasis en la adaptabilidad que en la previsibilidad. Los defensores de XP consideran que los cambios de requisitos sobre la marcha son un aspecto natural, inevitable e incluso deseable del desarrollo de proyectos. Creen que ser capaces de adaptarse a los cambios de requisitos en cualquier punto de la vida del proyecto, es una aproximación mejor y más realista que intentar definir todos los requisitos al comienzo del proyecto e invertir esfuerzos después en controlar los cambios en los requisitos. Los principios básicos de la programación extrema son: Simplicidad, Comunicación, Retroalimentación (feedback) y Coraje. (3)

Las características fundamentales del método son:

- Desarrollo iterativo e incremental.
- Pruebas unitarias continuas.
- Programación en parejas.
- Integración del equipo de programación con el cliente.
- Corrección de todos los errores.
- Refactorización del código.
- Propiedad del código compartida.
- Simplicidad.

Scrum

Scrum es un proceso de desarrollo de software iterativo e incremental utilizado comúnmente en entornos basado en la metodología ágil de desarrollo de software.

Aunque Scrum estaba enfocado a la gestión de procesos de desarrollo de software, puede ser utilizado en equipos de mantenimiento de software, o en una aproximación de gestión de programas: Scrum de Scrums.

Scrum es un modelo de referencia que define un conjunto de prácticas y roles, y que puede tomarse como punto de partida para definir el proceso de desarrollo que se ejecutará durante un proyecto. Los roles principales en Scrum son el ScrumMaster, que mantiene los procesos y trabaja de forma similar al director de proyecto, el ProductOwner, que representa a los stakeholders (clientes externos o internos), y el Team que incluye a los desarrolladores.

Existen varias implementaciones de sistemas para gestionar el proceso de Scrum, que van desde notas amarillas "post-it" y pizarras, hasta paquetes de software. Una de las mayores ventajas de Scrum es que es muy fácil de aprender, y requiere muy poco esfuerzo para comenzarse a utilizar. (4)

1.5.1 Proceso Unificado de Software (RUP), como metodología propuesta para el desarrollo del trabajo.

RUP

El Proceso Unificado de Desarrollo de Software (RUP) constituye un proceso de desarrollo de software que junto al Lenguaje Unificado de Modelado (UML) conforman la Metodología más utilizada en la actualidad para el análisis, implementación y documentación de Sistemas Orientados a Objetos. Esta metodología puede ser adaptable al contexto de cada problema en particular, por lo tanto constituye una herramienta flexible, configurable y aplicable para la solución de numerosos sistemas.

Como RUP constituye un proceso, en su modelación define como sus principales elementos a los Trabajadores, Actividades, Artefactos y El Flujo de Actividades.

Trabajadores: especifica el comportamiento y responsabilidades de cada individuo, grupo de individuos, sistema automatizado o máquina que trabajan en conjunto.

Actividades: es una tarea que tiene un propósito definido, ejecutada por un trabajador y manipula elementos.

Artefactos: son el resultado de la culminación de las actividades, también pueden ser modificados y usados por estas. Pueden ser modelos, elementos de modelado, ejecutables y código fuente.

Flujo de Actividades: son una secuencia de actividades llevadas a cabo por los trabajadores y que produce un resultado de valor.

RUP esta basado en 5 **principios de desarrollo**, estos son:

Adaptar el proceso: el proceso debe adaptarse a las características particulares del proyecto, el tamaño del mismo, así como las regulaciones que condicionan su funcionamiento ya que estas influirán en su diseño específico.

Balancear propiedades: puesto que los requerimientos de los participantes en el proyecto pueden ser diferentes o contradictorios, se debe encontrar un balance que satisfaga de alguna manera los deseos de todos los integrantes.

Demostrar valor iterativamente: la entrega del proyecto, aunque sea de un modo interno, se realiza en etapas iteradas. En cada una de las iteraciones se analizan las opiniones de inversores, la estabilidad y calidad del producto para poder refinar la dirección del proyecto e identificar riesgos.

Elevar el nivel de abstracción: este principio dominante motiva el uso de conceptos reutilizables tales como el patrón de software. Esto evita que los ingenieros de software vayan directo de los requisitos a la codificación, sin saber con certeza qué codificar, para satisfacer de la mejor manera los requerimientos pensando en la reutilización de código.

Enfocarse en la calidad: el control de la calidad en el proyecto no debe realizarse al final de cada iteración, sino en todos los aspectos de la producción. El aseguramiento de la calidad forma parte del proceso de desarrollo y no de un grupo independiente.

El ciclo de vida de RUP

La vida de un sistema transcurre a través de ciclos de desarrollo, desde su nacimiento hasta su muerte, en cada ciclo se repite el proceso unificado de desarrollo, Cada ciclo consta de cuatro fases: Inicio, elaboración, construcción y transición. Cada ciclo concluye con una versión del producto.

Dirigido por casos de uso: Los casos de uso reflejan lo que los usuarios futuros necesitan y desean, lo cual se capta cuando se modela el negocio y se representa a través de los requerimientos. A partir de aquí los casos de uso guían el proceso de desarrollo ya que los modelos que se obtienen, como resultado de los diferentes flujos de trabajo, representan la realización de los casos de uso (cómo se llevan a cabo).

Centrado en la arquitectura: La arquitectura muestra la visión común del sistema completo en la que el equipo de proyecto y los usuarios deben estar de acuerdo, por lo que describe los elementos del modelo que son más importantes para su construcción, los cimientos del sistema que son necesarios como base para comprenderlo, desarrollarlo y producirlo económicamente.

Iterativo e Incremental: RUP propone que cada fase se desarrolle en iteraciones. Una iteración involucra actividades de todos los flujos de trabajo, aunque desarrolla fundamentalmente algunos más que otros. Por ejemplo, una iteración de elaboración centra su atención en el análisis y diseño, aunque refina los requerimientos y obtiene un producto con un determinado nivel, pero que irá creciendo incrementalmente en cada iteración. Es práctico dividir el trabajo en partes más pequeñas o mini proyectos. Cada mini proyecto es una iteración que resulta en un incremento. Las iteraciones hacen referencia a pasos en los flujos de trabajo, y los incrementos, al crecimiento del producto. Cada iteración se realiza de forma planificada es por eso que se dice que son mini proyectos.

Con la implantación de RUP como metodología para el desarrollo del trabajo en cuestión se obtiene una mayor organización en la concepción de elementos pues provee un marco de trabajo que facilita la aplicación de esta metodología a muchos de los problemas que se presentan en la vida real, ya que puede ser adaptable a las condiciones de cada problema en particular. (5)

1.6 Plataforma .NET

.NET es un proyecto de Microsoft para crear una nueva plataforma de desarrollo de software con énfasis en transparencia de redes, con independencia de plataforma de hardware y que permita un rápido desarrollo de aplicaciones. Su objetivo es simplificar el desarrollo de aplicaciones Web. Provee las herramientas y tecnologías para transformar a Internet en una plataforma de computación distribuida en gran escala. Esta plataforma además soporta los estándares sobre los cuales se basan los servicios Web y utiliza tecnologías existentes, productos modificados para su uso dentro de la plataforma y elementos nuevos.

1.6.1 Plataforma .NET y Mono

.NET puede considerarse en ciertos aspectos como la respuesta de Microsoft a Java, aunque tiene bastantes diferencias. Reúne en una misma plataforma un conjunto interesante de características, como independencia de plataforma, independencia de lenguaje, soporte de bases de datos, soporte para XML, servicios Web y aplicaciones Web, entre otras. De entre todas estas, quizás una de las más importantes sean las dos primeras, referentes a la independencia tanto del lenguaje como de la plataforma.

Independencia del lenguaje

Se pueden desarrollar aplicaciones en múltiples lenguajes dentro de la plataforma .NET, pero lo más interesante, es que una aplicación puede tener diferentes partes desarrolladas en diferentes lenguajes, y todas estas pueden comunicarse entre sí, transparentemente, sin tener que utilizar ningún tipo de capa intermedia que posibilite esta comunicación. Esto permite a su vez una gran reutilización de código, ya que las clases desarrolladas para un proyecto en un lenguaje concreto, podrán ser reutilizadas en un nuevo proyecto, independientemente del lenguaje en el que se desarrolle éste.

Independencia de plataforma

Al igual que en Java, el código .NET no se compila a código máquina, sino a un código en un formato intermedio, independiente de la plataforma. Esto permite llevar los binarios producidos de una plataforma a otra, tal como sucede en Java. En la plataforma .NET existe el Common Language Runtime (CLR), que se encarga de ejecutar el código intermedio o Common Intermediate Language (CIL).

Proyecto Mono

Mono es un proyecto de implementación del Framework .NET de Microsoft utilizando código libre (open source), gestionado por Ximian y basado en las especificaciones definidas en ECMA.

Mono es una implementación del Framework .NET, pero de nada más relacionado con .NET, como pasaporte o software como servicios.

Los objetivos iniciales del proyecto Mono eran implementar en un entorno de software libre para el mundo Unix la especificaciones ECMA, para lo cual se incluye un compilador para C#, un entorno de ejecución CLR y un conjunto de librerías de clase que incluyen las FCL (Framework Class Library), así como otras añadidas.

El compilador de Mono es el Mono Runtime, que sería el equivalente al Common Language Runtime o entorno virtual de ejecución en la plataforma .NET. Implementa un compilador en tiempo de ejecución Just In Time (JIT) para el CIL de la máquina virtual, un compilador antes del tiempo de ejecución Ahead-of-Time (AOT), un cargador de clases, un recolector de basura, el sistema de threadings y las librerías de acceso a los metadatos.

El entorno de ejecución se puede utilizar también embebido dentro de otra aplicación, de forma que se pueden extender aplicaciones C y C++ mediante C#.

En comparación con la solución ofrecida por Microsoft, Mono incluye un intérprete y la opción de utilizar el runtime de forma embebida en una aplicación, además funciona sobre diversos sistemas operativos: Linux, MacOS X, BSD, SUN SOLARIS y Microsoft Windows. Aparte el intérprete puede ejecutarse sobre otras plataformas como HP-UX.

Mono constituye una opción para la portabilidad de aplicaciones desarrolladas en C #, puesto que brinda la posibilidad de hacer funcionar estas aplicaciones en una amplia variedad de sistemas operativos incrementando el alcance de las mismas.

Mono contiene una serie de elementos útiles para la construcción de un software:

- Un lenguaje común de infraestructura (CLI) máquina virtual que contiene un cargador de clases, justo a tiempo de compilación, recolección de basura y un tiempo de ejecución.
- Una biblioteca de clases que puede trabajar con cualquier lenguaje que trabaja en el CLR. Mono y .NET compatibles con las bibliotecas de clases.
- Un compilador para el lenguaje C #.

El CLR y el Sistema de Tipo Común (CTS) permiten que las aplicaciones y bibliotecas que se escriben en una colección de diferentes lenguajes se centren en el código de bytes. Esto significa por ejemplo que si se define una clase para hacer la manipulación algebraica en C #, esa clase puede ser reutilizada por cualquier otro lenguaje que apoye el CLI. Usted podría crear una clase en C #, subclase en C + + y crear una instancia en un programa Eiffel. Un solo sistema objeto, sistema de hilos, bibliotecas de clases, y sistema de recolección de basura, pueden ser compartidos a través de todos estos lenguajes. (6)

1.6.2 XML

XML (eXtensible Markup Language) es un lenguaje orientado a identificar estructuras de datos en un documento. La especificación XML define la manera estándar de cómo hay que realizar el marcado de expresiones en un documento no estructurado, para que con dicho marcado se defina una determinada estructura de datos.

La especificación XML no define el contenido de las estructuras de datos, son los expertos de cada dominio y las entidades reguladoras, los agentes que pueden utilizar el estándar XML para consensuar un lenguaje común que permita transformar los documentos no estructurados en estructuras procesables por un sistema.

Cuando hablamos de un documento nos referimos no solo al concepto tradicional de documento en papel o soporte electrónico sino a todos los tipos de documentos actuales: páginas Web, correo electrónico, gráficos vectoriales, transacciones de comercio electrónico, etc. Un documento XML es un documento que puede ser leído y entendido por una persona "human readable system" y a la vez puede ser procesado por un sistema para extraer información "machine readable system". (7)

Ventajas

- Es extensible: Después de diseñado y puesto en producción, es posible extender XML con la adición de nuevas etiquetas, de modo que se pueda continuar utilizando sin complicación alguna.
- El analizador es un componente estándar, no es necesario crear un analizador específico para cada versión de lenguaje XML. Esto posibilita el empleo de cualquiera de los analizadores disponibles. De esta manera se evitan bugs y se acelera el desarrollo de aplicaciones.
- Si un tercero decide usar un documento creado en XML, es sencillo entender su estructura y procesarla. Mejora la compatibilidad entre aplicaciones.

1.6.3 Servicio Web

Los Servicios Web son la revolución informática de la nueva generación de aplicaciones que trabajan colaborativamente, en las cuales el software está distribuido en diferentes servidores.

Un Servicio Web (en inglés Web Service) es un conjunto de protocolos y estándares que sirven para intercambiar datos entre aplicaciones. Distintas aplicaciones de software desarrolladas en lenguajes de programación diferentes, y ejecutadas sobre cualquier plataforma, pueden utilizar los Servicios Web para intercambiar datos en redes de ordenadores como Internet. La interoperabilidad se consigue mediante la adopción de estándares abiertos. Las organizaciones OASIS y W3C son los comités responsables de la arquitectura y reglamentación de los Servicios Web. Para mejorar la interoperabilidad entre distintas implementaciones de Servicios Web se ha creado el organismo WS-I, encargado de desarrollar diversos perfiles para definir de manera más exhaustiva estos estándares. (8)

Ventajas de los Servicios Web

- Aportan interoperabilidad entre aplicaciones de software independientemente de sus propiedades o de las plataformas sobre las que se instalen.
- Los servicios Web fomentan los estándares y protocolos basados en texto, que hacen más fácil acceder a su contenido y entender su funcionamiento.

- Al apoyarse en HTTP, los servicios Web pueden aprovecharse de los sistemas de seguridad firewall sin necesidad de cambiar las reglas de filtrado.
- Permiten que servicios y software de diferentes compañías ubicadas en diferentes lugares geográficos puedan ser combinados fácilmente para proveer servicios integrados.
- Permiten la interoperabilidad entre plataformas de distintos fabricantes por medio de protocolos estándar y abiertos. Las especificaciones son gestionadas por una organización abierta, la W3C, por tanto no hay secretismos por intereses particulares de fabricantes concretos y se garantiza la plena interoperabilidad entre aplicaciones.

1.7 Herramientas a utilizar

Para el desarrollo de un software es necesario utilizar algunas herramientas que permiten la creación de este y que suelen hacer el trabajo más fácil. Se utilizan herramientas para modelar, implementar y para el trabajo con Bases de Datos.

Visual Studio 2005

Microsoft Visual Studio es un entorno de desarrollo integrado (IDE) que provee un conjunto de herramientas de desarrollo diseñadas para ayudar a los desarrolladores de software (tanto si son principiantes como profesionales con experiencia) a enfrentarse a los desafíos complejos y crear soluciones innovadoras. Cada día, los desarrolladores de software se abren camino a través de duros problemas para crear software que suponga una diferencia en la vida de los demás. La función de Visual Studio es mejorar el proceso de desarrollo para hacer que el trabajo de lograr esos adelantos sea más fácil y satisfactorio.

Soporta varios lenguajes de programación tales como Visual C++, Visual C#, Visual J#, ASP.NET y Visual Basic .NET, aunque actualmente se han desarrollado las extensiones necesarias para muchos otros.

Visual Studio permite a los desarrolladores crear aplicaciones, sitios y aplicaciones web, así como servicios web en cualquier entorno que soporte la plataforma .NET. Así se pueden crear aplicaciones que se intercomunican entre estaciones de trabajo, páginas web y dispositivos móviles. (9)

Los desarrolladores de este trabajo escogieron esta herramienta porque:

- Aumenta la productividad y la calidad gracias a herramientas integradas y conocidas.
- Implementa y protege la infraestructura y las aplicaciones web críticas, así como ofrece compatibilidad entre ellas.
- Reduce los costos mediante una mejor visibilidad del proceso de desarrollo.
- Ofrece mejor capacidad de predicción y planeación mediante la compatibilidad integrada con procesos y metodologías.
- Tienen un conocimiento previo de la misma y tienen experiencia en el manejo de esta.

Visual Paradigm

Visual Paradigm es una herramienta que utiliza UML profesional y soporta el ciclo de vida completo de desarrollo de software: análisis y diseño orientados a objetos, construcción, pruebas y despliegue. El software de modelado UML ayuda a una rápida construcción de aplicaciones de calidad, mejores y a un menor coste. Visual Paradigm es muy potente y de fácil uso, permite crear todo tipo de diagramas UML, generar código desde diagramas y generar documentación. (10)

Entre sus características se encuentran:

- Soporte de UML versión 2.1.
- Diagramas de Procesos de Negocio: Proceso, Decisión, Actor de negocio, Documento.
- Ingeniería de ida y vuelta.

- Ingeniería inversa: Código a modelo, código a diagrama.
- Generación de código: Modelo a código, diagrama a código.
- Editor de Detalles de Casos de Uso: Entorno todo-en-uno para la especificación de los detalles de los casos de uso, incluyendo la especificación del modelo general y de las descripciones de los casos de uso.
- Diagramas de flujo de datos.
- Generación de bases de datos: Transformación de diagramas de Entidad-Relación en tablas de base de datos.
- Generador de informes para generación de documentación.

Es de fácil uso y permite generar código y documentación, lo que agiliza el trabajo de los desarrolladores que lo utilizan. Por todas las razones expuestas anteriormente se escoge como herramienta a utilizar por los implicados en el desarrollo de este trabajo.

PostgreSQL

Para la realización de esta aplicación no se trabajará con base de datos (BD), pero como la misma pertenece a un módulo de un proyecto en el cual la herramienta para el trabajo con BD es esta, se ha dedicado una parte para enunciar algunas de sus características.

PostgreSQL es un Sistema Gestor de Base de Datos relacional orientada a objetos de software libre, publicado bajo la licencia BSD. Desarrollado por el Grupo Global de Desarrollo de PostgreSQL (PGDG) que cuenta con cientos de programadores basándose en 21 años de desarrollo continuo e incremental, ofreciendo nuevas funcionalidades que mejoran sus prestaciones. Brinda a los usuarios la opción de instalar una base de datos no privativa, permitiéndoles ahorrar dinero, mejorar el rendimiento e incrementar la productividad. (11)

Entre sus principales características se encuentran:

- **Alta concurrencia:** mediante un sistema denominado Acceso Concurrente Multiversión (MVCC) PostgreSQL permite que mientras un proceso escribe en una tabla, otros accedan a la misma tabla sin necesidad de bloqueos. Cada usuario obtiene una visión consistente de lo último a lo que se actualizó. Esta estrategia es superior al uso de bloqueos por tabla o por filas común en otras bases, eliminando la necesidad del uso de bloqueos explícitos.
- **PostgreSQL provee nativamente soporte para:** Direcciones IP (IPv4 e IPv6) y para direcciones MAC.
- **Disparadores (triggers):** Un disparador se define en una acción específica basada en algo ocuriente dentro de la base de datos. Esto significa la ejecución de un procedimiento almacenado basado en una determinada acción sobre una tabla específica.

Por todas estas características, que a la vez son ventajas que posee este gestor de base de datos, es escogido por los desarrolladores para su utilización, además por las facilidades de uso que posee permitiendo crear una amplia funcionalidad a través de su sistema de activación de disparadores.

1.8 Lenguajes de Programación

C Sharp (C#)

Microsoft Visual C# 2005 es un lenguaje de programación diseñado para crear una amplia gama de aplicaciones que se ejecutan en .NET Framework. C# es simple, eficaz, con seguridad de tipos y orientado a objetos. Con sus diversas innovaciones, C# permite desarrollar aplicaciones rápidamente y mantiene la expresividad y elegancia de los lenguajes de tipo C. (12)

Visual Studio admite Visual C# con un editor de código completo, plantillas de proyecto, diseñadores, asistentes para código, un depurador eficaz y fácil de usar, además de otras herramientas. La biblioteca de clases .NET Framework ofrece acceso a una amplia gama de servicios de sistema operativo y a otras clases útiles y adecuadamente diseñadas que aceleran el ciclo de desarrollo de manera significativa.

En la actualidad existen los siguientes compiladores para el lenguaje C#:

- Microsoft.NET framework SDK incluye un compilador de C#, pero no un IDE.
- Microsoft Visual Studio .NET, IDE por excelencia de este lenguaje, versión 2002, 2003, 2005 y 2008.
- #develop, es un IDE libre para C# bajo licencia LGPL, muy similar a Microsoft Visual C#.
- Mono, es una implementación GPL de todo el entorno .NET desarrollado por Novell. Como parte de esta implementación se incluye un compilador de C#.
- Delphi 2006, de Borland Software Corporation.
- dotGNU Portable.NET, de la Free Software Foundation

Se escoge como lenguaje de programación por las **ventajas** que ofrece, las cuales se enuncian a continuación:

- Declaraciones en el espacio de nombres: al empezar a programar algo, se puede definir una o más clases dentro de un mismo espacio de nombres.
- Tipos de datos: en C# existe un rango más amplio y definido de tipos de datos que los que se encuentran en C, C++ o Java.
- Atributos: cada miembro de una clase tiene un atributo de acceso, que puede ser de tipo público, protegido, interno, interno protegido y privado.
- Pase de parámetros: aquí se puede declarar a los métodos para que acepten un número variable de parámetros. De forma predeterminada, el pase de parámetros es por valor, a menos que se use la palabra reservada ref, la cual indica que el pase es por referencia.
- Métodos virtuales y redefiniciones: antes de que un método pueda ser redefinido en una clase base, debe declararse como virtual. El método redefinido en la subclase debe ser declarado con la palabra override.

- **Propiedades:** un objeto tiene intrínsecamente propiedades, y debido a que las clases en C# pueden ser utilizadas como objetos, C# permite la declaración de propiedades dentro de cualquier clase.
- **Inicializador:** un inicializador es como una propiedad, con la diferencia de que en lugar de un nombre de propiedad, un valor de índice entre corchetes se utiliza en forma anónima para hacer referencia al miembro de una clase.
- **Control de versiones:** C# permite mantener múltiples versiones de clases en forma binaria, colocándolas en diferentes espacios de nombres. Esto permite que versiones nuevas y anteriores de software puedan ejecutarse en forma simultánea.

Aunque tiene muchas ventajas también tiene sus **desventajas**, entre ellas se encuentran:

- Se tiene que conseguir una versión reciente de Visual Studio .NET.
- Se debe contar con algunos requerimientos mínimos del sistema para poder trabajar adecuadamente tales como: contar con Windows NT 4 o superior, tener alrededor de 4 gigas de espacio libre para la pura instalación, etc.
- Además para quien no está familiarizado con ningún lenguaje de programación, le costará más trabajo iniciarse en su uso, y si se quiere consultar algún tutorial más explícito sobre la programación en C# se tendría que contar además con una conexión a Internet. (13)

SQL (Structured Query Language)

El Lenguaje de consulta estructurado SQL, es un lenguaje declarativo de acceso a bases de datos relacionales que permite especificar diversos tipos de operaciones sobre las mismas. Una de sus características es el manejo del álgebra y el cálculo relacional permitiendo lanzar consultas con el fin de recuperar -de una forma sencilla- información de interés de una base de datos, así como también hacer cambios sobre la misma.

Características Generales

El SQL es un lenguaje de acceso a bases de datos que explota la flexibilidad y potencia de los sistemas relacionales permitiendo gran variedad de operaciones sobre los mismos.

Es un lenguaje declarativo de "alto nivel" o "de no procedimiento", que gracias a su fuerte base teórica y su orientación al manejo de conjuntos de registros, y no a registros individuales, permite una alta productividad en codificación y la orientación a objetos. De esta forma una sola sentencia puede equivaler a uno o más programas que utilizas en un lenguaje de bajo nivel orientado a registro.

Principales Ventajas

- Permite la creación de Procedimientos Almacenados, Usuarios, Vistas y Triggers.
- Tiene un solo archivo de base de datos y uno de transacciones.
- Posee una interfaz gráfica para la administración de la Base de Datos.
- Permite la automatización de tareas de mantenimiento de la Base de Datos
- Posee una mayor velocidad en búsquedas.

La principal desventaja que tiene es que si no tienes una buena estructura de Base de datos y esta se corrompe, adiós a todo el trabajo hasta el último respaldo que tengas. (13)

1.9 Conclusiones

En este capítulo se realizó un análisis de las soluciones existentes a nivel nacional e internacional, así como una amplia descripción del problema a resolver. También se realizó un resumen de los principales conceptos asociados al problema, lo cual permite un mayor entendimiento del mismo. Se argumentó el objeto de estudio y se realizó un estudio profundo de algunas de las metodologías y herramientas de desarrollo existentes, y que podían servir para la realización del trabajo que se propone.

Se explica además cuales fueron las herramientas que se seleccionaron y el por qué de la selección, teniendo en cuenta sus características, sus ventajas y el conocimiento que poseían los desarrolladores de este trabajo de las mismas. Con este estudio se logró dar cumplimiento a varias de las tareas de investigación planteadas con anterioridad.

Se llega a la conclusión que con todos estos conocimientos incorporados están creadas las condiciones para comenzar a desarrollar la aplicación propuesta. Esta aplicación eliminará los problemas que tiene la actual, y se le agregarán funcionalidades como las que se mencionan a continuación:

1. Se anotará lanzamiento por lanzamiento, permitiendo contar los lanzamientos del pitcher y evitando una actuación excesiva del mismo que pueda provocarle una lesión.
2. El trabajo del anotador será más fácil pues ya no será necesario conocer el Sistema Codificado de Anotación al detalle.
3. La interfaz visual será más usable y amigable, lo que permitirá un mejor manejo de la aplicación para el anotador.

Con todos estos elementos incorporados la nueva aplicación superará con creces a la actual y permitirá que el proceso de anotación de juegos de béisbol en Cuba se agilice, ayudando al desarrollo de este deporte que tantos logros ha conseguido a lo largo de los años.

Capítulo 2: Características del Sistema

2.1 Introducción

En este capítulo se describe la anotación de juegos de béisbol, identificando las desventajas del sistema que se utiliza actualmente para llevar a cabo este proceso y las consecuencias que tienen en su funcionamiento.

Se hace referencia a los procesos que serán objeto de automatización en la investigación, así como la solución que proponen los autores del trabajo de diploma a la situación problemática que se presenta. También son enunciadas las ventajas que se derivan con la implementación del nuevo sistema de anotación. Es realizado un levantamiento de los requisitos del sistema, tanto funcionales como no funcionales, pues estos constituyen la base fundamental para la concepción de la aplicación. Además se definen los casos de uso ya que estos guían el proceso de desarrollo de software.

2.2 Problema y situación problemática

La Federación Cubana de Béisbol Amateur lleva a cabo una serie de procesos vinculados al béisbol. La anotación de estos juegos es uno de los procesos más importantes que se desarrollan, pues es la base de la recopilación de estadísticas que son un aspecto fundamental para este deporte. La aplicación actual que se utiliza para realizar la anotación no cumple con las necesidades actuales del béisbol en Cuba, lo que forma parte de la situación problemática del trabajo.

El sistema actual que se encarga de la anotación de juegos de béisbol está desarrollado en MS-DOS, porque en el momento en que se realizó, en Cuba no existía un desarrollo tecnológico como el que existe en la actualidad. Es necesario mejorar la interfaz que posee esta aplicación para permitir que los usuarios del mismo se sientan atraídos y le sea más fácil cumplir con su trabajo.

Es importante también permitir que los juegos se anoten lanzamiento por lanzamiento, lo que lograría recopilar mayor cantidad de información estadística acerca de los lanzadores del encuentro, algo que actualmente no se hace, pues solamente se insertan datos en la aplicación cuando se completa una jugada. Además es necesario que el anotador pueda hacer su trabajo de una forma más rápida y eficiente, esto se lograría eliminando la forma en que los anotadores insertan una jugada en el sistema.

Actualmente lo hacen mediante un código, lo que provoca que su trabajo sea más engorroso y los obliga a conocer profundamente el Sistema Codificado de Anotación Cubano.

2.3 Propuesta del sistema

Para dar solución a la problemática planteada será implementado un sistema de anotación que contará con un conjunto de características que optimizarán las funcionalidades de la aplicación existente. Dentro de las nuevas mejoras se encuentra la implementación de una interfaz de usuario amigable, facilitando la interacción del usuario con la aplicación, ya que el sistema anterior estaba desarrollado en MS-DOS y restringía de forma considerable el uso de la misma. También se incorpora la funcionalidad de anotar lanzamiento por lanzamiento y no jugada por jugada como se anotaba anteriormente; esta nueva funcionalidad permite obtener un conjunto de estadísticas, como: conocer la cantidad de lanzamientos que efectúa un pitcher, los tipos de lanzamientos que realiza, la velocidad y la zona donde cayó cada lanzamiento realizado durante el encuentro. La anotación de las jugadas ocurridas durante el encuentro será mostrada al anotador de forma descriptiva, el anotador escoge la jugada correcta y el sistema se encarga de codificar y guardar la información, por lo que se logra una mayor eficiencia y veracidad en la recopilación de los datos estadísticos.

El sistema debe ser desplegado en cada estadio del país con el propósito de anotar todos los partidos que se lleven a cabo en cada jornada de juegos. Debe contar con una conexión de red para poder acceder al servidor central mediante un Servicio Web, incorpora así la funcionalidad de salvar todos los datos estadísticos recopilados durante el juego en un fichero .XML, que posteriormente será enviado al servidor. El proceso de envío del fichero es realizado automáticamente por la aplicación mientras se desarrolle el partido, actualizando la información que se encuentra en el servidor. En caso de no haber conexión en un momento dado, los datos se seguirán guardando en el fichero .XML y se continuará haciendo peticiones de conexión al servidor, y en cuanto se restablezca la conexión se envía el fichero y se actualiza la información en el servidor. Esta nueva funcionalidad abre paso a la posibilidad de crear una aplicación Web de seguimiento de juego en tiempo real mejorando así las prestaciones del nuevo sistema de anotación.

2.4 Modelo de Negocio

La Federación Cubana de Béisbol Amateur es la encargada de organizar cualquier evento de béisbol que se desarrolle en Cuba. En todos estos eventos es necesario llevar a cabo el proceso de anotación de todos los juegos que se realicen. Para ello se utiliza una aplicación que como se ha dicho en varias ocasiones no cumple con la exigencias actuales de este deporte, pero se encuentra funcionando correctamente y permite recopilar un grupo de estadísticas importantes.

Con el objetivo de realizar una aplicación más funcional, usable y que permita un trabajo más fácil y eficiente, incorporándole algunas funcionalidades que no posee la actual, se desarrolló el Modelo de Negocio que se presenta a continuación, el cual permitió comprender los procesos del negocio de la organización, la forma en que se desarrollan las acciones y las respuestas que brindan las mismas.

2.4.1 Actores del Negocio

Actores	Justificación
Comisario Técnico	Es el encargado de la inicialización de la anotación de juego, una vez comprobado el estado de las alineaciones iniciales ordena el comienzo de la anotación.

Tabla No. 1: Justificación de los Actores del Negocio

2.4.2 Trabajadores del Negocio

Trabajadores	Justificación
Anotador Oficial	Es el encargado de insertar todos los datos necesarios para el comienzo de un juego de béisbol, así como de tomar las decisiones correspondientes según las reglas de anotación para anotar todas las jugadas ocurridas en el mismo.

Tabla No. 2: Justificación de los Trabajadores del Negocio

2.4.3 Diagrama de casos de Uso del Negocio

Caso de Uso Anotar Juego

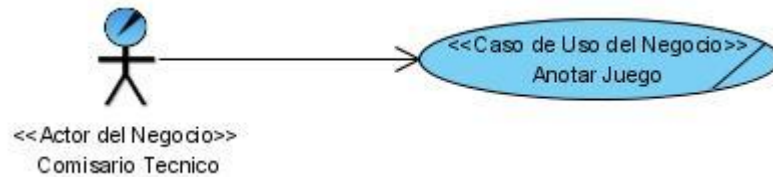


Figura No. 1: Diagrama de Caso de Uso - Anotar Juego

2.4.4 Descripción del Caso de Uso: Anotar Juego

Caso de Uso:	Anotar Juego
Actores:	Comisario Técnico
Resumen:	El caso de uso se inicia cuando el comisario técnico del encuentro le ordena al anotador que comience el proceso de anotación del juego. El anotador registra el nombre del estadio, las alineaciones de los equipos participantes, árbitros que intervendrán en el encuentro, comisario técnico del juego, así como la hora de comienzo del juego en la hoja de anotación. Una vez comenzado el juego el anotador lleva a cabo el proceso de anotación de todas las jugadas que ocurren en el encuentro de béisbol, cuando finaliza el partido envía la hoja de anotación a la Comisión Nacional de Béisbol.
Precondiciones:	Que haya llegado la fecha y hora programada en el calendario para el comienzo del partido.
Propósito	Llevar a cabo el proceso de anotación de un juego de béisbol.
Referencias	Ninguna
Prioridad	Crítico

Flujo Normal de Eventos	
Acción del Actor	Respuesta del Negocio
<p>1. El comisario técnico decide si se le da comienzo al juego o no.</p>	<p>2. El anotador recibe la orden del comienzo del juego emitida por el comisario.</p> <p>3. El anotador oficial del encuentro registra todos los datos del partido, como: estadio en que se desarrollará, árbitros y comisario que intervienen, alineaciones y hora de comienzo en la hoja de anotación.</p> <p>4. El anotador realiza el proceso de anotación de todas las jugadas acontecidas durante el encuentro.</p> <p>5. Al concluir el partido el anotador envía la hoja de anotación a la Comisión Nacional de Béisbol.</p>
Flujos Alternos	
Acción del Actor	Respuesta del Negocio
<p>1.a) El comisario técnico decide suspender el juego.</p>	<p>1. a 1) El anotador registra el motivo por el cual se decidió suspender el partido.</p> <p>1. a 2) El anotador envía la hoja de anotación a la Comisión Nacional de Béisbol.</p>
Flujos Alternos	
Acción del Actor	Respuesta del Negocio
<p>1.b) El comisario técnico decide sellar el juego.</p>	<p>1. b 1) El anotador registra el motivo por el cual</p>

Capítulo 2: Características del Sistema

	<p>se decidió sellar el parido.</p> <p>1. b 2) El anotador envía la hoja de anotación a la Comisión Nacional de Béisbol.</p>
--	--

Tabla No. 3: Descripción del caso de Uso - Anotar Juego

2.4.5 Diagrama de Actividad: Anotar Juego

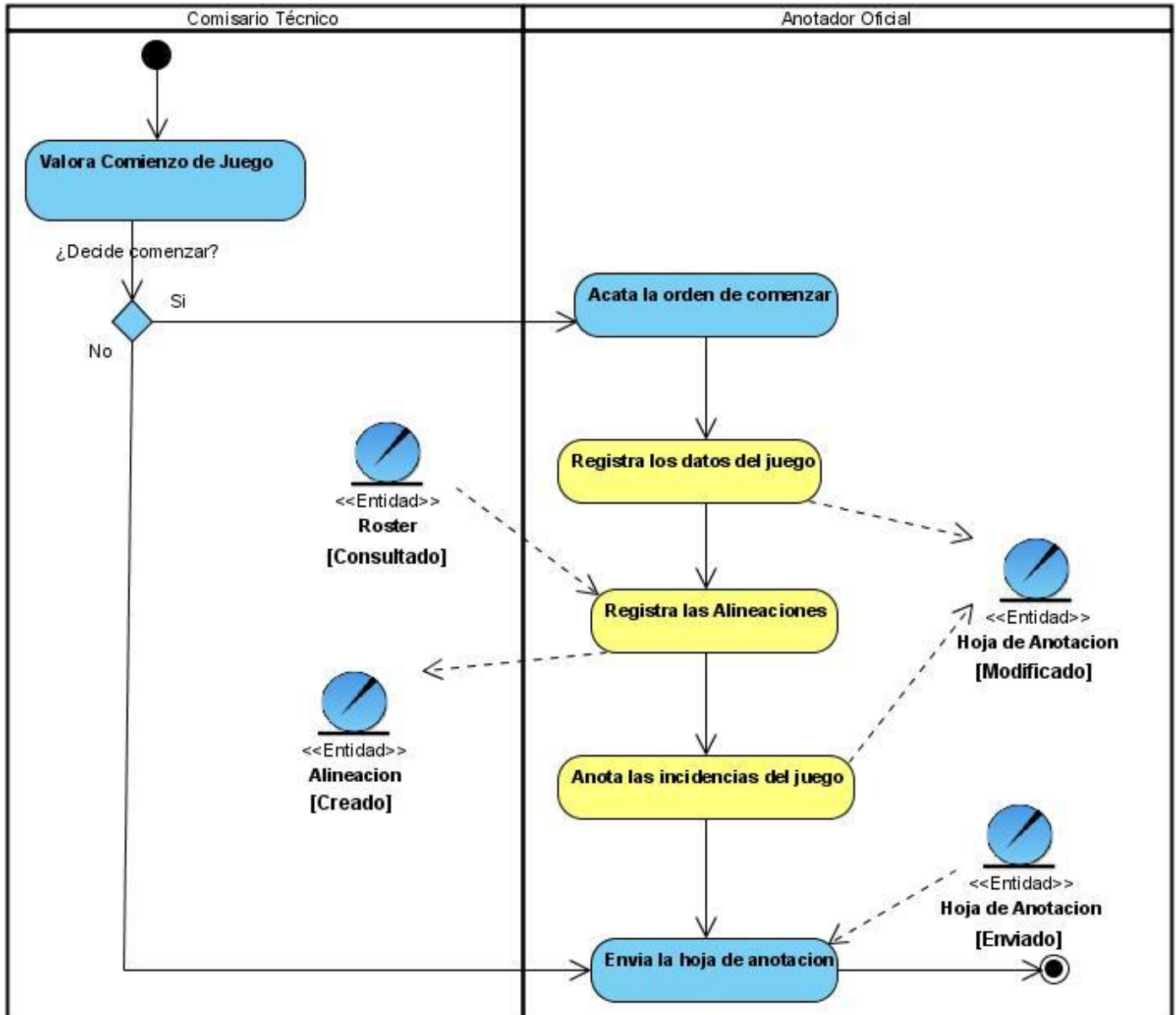


Figura No. 2: Diagrama de Actividad - Anotar Juego

2.4.6 Diagrama de Objetos



Figura No. 3: Diagrama de Objetos del Negocio

2.5 Especificación de los Requisitos del Software

La Especificación de Requisitos de Software (ERS) es una descripción completa del comportamiento del sistema que se va a desarrollar. Incluye un conjunto de casos de uso que describe todas las interacciones que tendrán los usuarios con el software. Los casos de uso también son conocidos como requisitos funcionales. Además de los casos de uso, la ERS también contiene requisitos no funcionales (o complementarios). Los requisitos no funcionales son requisitos que imponen restricciones en el diseño o la implementación (Como por ejemplo restricciones en el diseño o estándares de calidad). (14)

A continuación se muestra un listado de los requerimientos funcionales y no funcionales, obtenidos en el levantamiento de requisitos correspondiente al sistema propuesto.

2.5.1 Requerimientos funcionales

R1 - Gestionar Conexión

1.1 - El sistema debe permitir configurar la conexión con el servicio web.

1.2 - El sistema debe permitir guardar la conexión en fichero local.

R2 - Autenticar Usuario

- ✓ Usuario
- ✓ Contraseña

R3 - Gestionar Usuarios

3.1 - Agregar Usuario

- ✓ Usuario
- ✓ Contraseña

3.2 - Modificar Usuario

- ✓ Usuario
- ✓ Contraseña Actual
- ✓ Nueva Contraseña

3.3 - Eliminar Usuario

- ✓ Usuario

R4 - Gestionar Anotadores del Juego

4.1 - Agregar Anotador del Juego

- ✓ Juez
- ✓ Categoría (Anotador Oficial o Anotador Auxiliar)

4.2 - Actualizar Anotador del Juego

- ✓ Juez
- ✓ Categoría (Anotador Oficial o Anotador Auxiliar)

4.3 - Eliminar Anotador del Juego

- ✓ Anotador Juego

4.4 - Mostrar Anotadores del Juego

- ✓ Juez
- ✓ Categoría (Anotador Oficial o Anotador Auxiliar)

R5 - Gestionar Árbitros del Juego

5.1 - Agregar Árbitros del Juego

- ✓ Juez
- ✓ Posición

5.2 - Actualizar Árbitros del Juego

- ✓ Juez
- ✓ Posición

5.3 - Eliminar Árbitros del Juego

- ✓ Árbitro Juego

5.4 - Mostrar Árbitros del Juego

- ✓ Juez
- ✓ Posición

R6 - Gestionar Posición de los Árbitros

6.1- Agregar Posición de los Árbitros

- ✓ Posición
- ✓ Sigla

6.2 - Actualizar Posición de los Árbitros

- ✓ Posición
- ✓ Sigla

6.3 - Eliminar Posición Árbitros

- ✓ Posición

R7 - Gestionar Alineación del Juego

7.1 - Agregar Alineación del Juego

- ✓ Jugador Equipo
- ✓ Posición
- ✓ Turno

7.2 - Modificar Alineación del Juego

- ✓ Jugador Equipo
- ✓ Posición

- ✓ Turno

7.3 - Eliminar Alineación del Juego

- ✓ Alineación Juego

7.4 - Mostrar Alineación del Juego

- ✓ Jugador Equipo
- ✓ Posición
- ✓ Turno

R8 - Gestionar Jugadas del Juego

8.1 - Agregar Jugadas del Juego

- ✓ Equipo bateador
- ✓ Equipo lanzador
- ✓ Bateador
- ✓ Lanzador
- ✓ Casilla1 (Camiseta del Jugador)
- ✓ Casilla2 (Acción Ofensiva)
- ✓ Casilla3 (Acción Defensiva1)
- ✓ Casilla4 (Acción Defensiva2)
- ✓ Casilla5 (Avance del corredor a 1ra. base)
- ✓ Casilla6 (Avance del corredor a 2da. base)
- ✓ Casilla7 (Avance del corredor a 3ra. base)
- ✓ Casilla8 (Foul o buena (0 (buena),1(foul)))

8.2 - Actualizar Jugadas del juego

- ✓ Equipo bateador
- ✓ Equipo lanzador

- ✓ Bateador
- ✓ Lanzador
- ✓ Casilla1 (Camiseta del Jugador)
- ✓ Casilla2 (Acción Ofensiva)
- ✓ Casilla3 (Acción Defensiva1)
- ✓ Casilla4 (Acción Defensiva2)
- ✓ Casilla5 (Avance del corredor a 1ra. base)
- ✓ Casilla6 (Avance del corredor a 2da. base)
- ✓ Casilla7 (Avance del corredor a 3ra. base)
- ✓ Casilla8 (Foul o buena (0 buena, 2foul))

8.3 - Eliminar Jugadas del Juego

- ✓ Jugada

8.4 - Mostrar Jugadas

- ✓ Acción ofensiva
- ✓ Acción Defensiva
- ✓ Avance de los Corredores
- ✓ Nombre de los Corredores y bateadores

R9 - Gestionar Lanzamientos

9.1 - Agregar Lanzamiento

- ✓ Posición Batear
- ✓ Camiseta del Jugador
- ✓ Ocurrencia (bola por regla, bola, strike tirándole, strike cantado, foul, en jugada)
- ✓ Velocidad
- ✓ Tipo de lanzamiento

9.2 - Modificar Lanzamiento

- ✓ Posición Batear
- ✓ Camiseta del Jugador
- ✓ Ocurrencia (bola por regla, bola, strike tirándole, strike cantado, foul, en jugada)
- ✓ Velocidad
- ✓ Tipo de lanzamiento

9.3 - Eliminar Lanzamiento

- ✓ Lanzamiento

9.4 - Mostrar Lanzamientos

- ✓ Posición Batear
- ✓ Camiseta del Jugador
- ✓ Ocurrencia (bola por regla, bola, strike tirándole, strike cantado, foul, en jugada)
- ✓ Velocidad
- ✓ Tipo de lanzamiento

R10 - Gestionar Otros Datos del Juego

- ✓ Comisarios Técnicos
- ✓ Duración del juego
- ✓ Cantidad de personas participantes en el estadio
- ✓ Estado del juego (Iniciado, Terminado, Oficializado, Sellado, Suspendido)
- ✓ Forfeit (gana automáticamente el equipo visitante)
- ✓ Fecha
- ✓ Hora de comienzo

2.5.2 Requerimientos No Funcionales

Usabilidad

- ✓ Preparar a los usuarios de la aplicación para el trabajo con la misma, aunque esta sea fácil de usar.

Rendimiento

- ✓ El sistema debe responder en un tiempo relativamente rápido a las peticiones del usuario, para ello debe de cumplir con los siguientes parámetros: velocidad de respuesta, eficiencia, disponibilidad, tiempo de respuesta y tiempo de recuperación.

Implementación

- ✓ El sistema debe estar implementado en una aplicación de escritorio que permita cargar ficheros XML.

Interfaz

- ✓ El sistema debe contar con una interfaz amigable, que permita una amplia facilidad de uso, con un nivel de ventanas que haga el trabajo más fácil para los usuarios.

Portabilidad

- ✓ La aplicación debe estar implementada para correr sobre la plataforma Windows y LINUX.

Seguridad

- ✓ Protección contra acciones no autorizadas o que puedan afectar la integridad de los datos estadísticos.
- ✓ La información manejada por el sistema deberá estar protegida de acceso no autorizado, por lo cual se implementa un mecanismo de autenticación del sistema.
- ✓ La información de autenticación deberá ser encriptada.

2.6 Casos de Uso del Sistema

2.6.1 Actores del Sistema

Actor	Descripción
Anotador Oficial	Es el encargado de llevar a cabo el proceso de anotación durante el desarrollo del encuentro. Inserta todos los datos necesarios para el comienzo de un juego de béisbol, debe tomar las decisiones correspondientes según las reglas de anotación para anotar todas las jugadas ocurridas en el mismo.
Administrador del Sistema	Le proporciona los permisos necesarios al anotador para la realización de su trabajo. Es el que se encarga de gestionar los usuarios del sistema.

Tabla No. 4: Justificación de los actores del sistema.

2.6.2 Listado de Casos de Uso del Sistema

Autenticar Usuario

CU - 1	Autenticar Usuario
Actor	Administrador del Sistema
Descripción	El administrador selecciona la opción de autenticarse en el sistema, introduce los datos necesarios y se le brinda los permisos que necesita para trabajar con la aplicación.
Referencia	R1

Tabla No. 5: CU-1 - Autenticar Usuario

Gestionar Usuario

CU - 2	Gestionar Usuarios
Actor	Administrador del Sistema
Descripción	El administrador selecciona la opción Gestionar Usuarios, escoge una de las opciones que se pueden realizar, Adicionar Usuarios, Modificar Usuarios y Eliminar Usuarios.

Referencia	R2
-------------------	----

Tabla No. 6: CU-2 - Gestionar Usuario

Gestionar Anotadores

CU - 3	Gestionar Anotadores
Actor	Anotador Oficial
Descripción	El anotador oficial selecciona la opción Gestionar Anotadores, escoge una de las opciones que se pueden realizar, Mostrar Anotador, Adicionar Anotador, Actualizar Anotador y Eliminar Anotador. El anotador oficial realiza la operación escogida.
Referencia	R3

Tabla No. 7: CU-3 - Gestionar Anotadores

Gestionar Árbitros

CU - 4	Gestionar Árbitros
Actor	Anotador Oficial
Descripción	El anotador oficial selecciona la opción Gestionar Árbitros, elige una de las opciones que se pueden efectuar; Mostrar Árbitros, Adicionar Árbitros, Actualizar Árbitros y Eliminar Árbitros. El anotador realiza la operación escogida, el sistema verifica la validez de los datos, en caso de ser correctos los actualiza.
Referencia	R4

Tabla No. 8: CU-4 - Gestionar Árbitros

Gestionar Posición Árbitros

CU - 5	Gestionar Posición Árbitros
Actor	Anotador Oficial
Descripción	El anotador oficial selecciona la opción Gestionar Posición de los Árbitros, escoge una de las opciones que se pueden realizar; Agregar Posición, Actualizar Posición y Eliminar Posición. El anotador realiza la operación seleccionada, el sistema verifica la validez de los datos.
Referencia	R5

Tabla No. 9: CU-5 - Gestionar Posición Árbitros

Gestionar Alineaciones

CU - 6	Gestionar Alineaciones
Actor	Anotador Oficial
Descripción	El anotador oficial selecciona la opción Gestionar Alineaciones, escoge una de las opciones que se pueden realizar; Agregar Alineación, Actualizar Alineación, Eliminar Alineación y Mostrar Alineación. El anotador realiza la operación seleccionada, el sistema verifica la validez de los

	datos.
Referencia	R6

Tabla No. 10: CU-6 - Gestionar Alineaciones

Gestionar Jugadas

CU - 7	Gestionar Jugadas
Actor	Anotador Oficial
Descripción	El anotador oficial selecciona la opción Gestionar Jugadas, escoge una de las opciones que se pueden realizar; Agregar Jugada, Actualizar Jugada, Eliminar Jugada y Mostrar Jugada. El anotador realiza la operación seleccionada, el sistema verifica la validez de los datos.
Referencia	R7

Tabla No. 11: CU-7 - Gestionar Jugadas

Gestionar Lanzamientos

CU - 8	Gestionar Lanzamientos
Actor	Anotador Oficial
Descripción	El anotador oficial selecciona la opción Gestionar Lanzamientos, escoge una de las opciones que se pueden realizar; Agregar Lanzamientos, Actualizar Lanzamientos, Eliminar Lanzamientos y Mostrar Lanzamientos. El anotador realiza la operación

	seleccionada, el sistema verifica la validez de los datos
Referencia	R8

Tabla No. 12: CU-8 - Gestionar Lanzamientos

Gestionar Datos Adicionales

CU - 9	Gestionar Datos Adicionales
Actor	Anotador Oficial
Descripción	El anotador oficial selecciona la opción Gestionar Datos adicionales, escoge una de las opciones que se pueden realizar; Agregar Datos Adicionales, Actualizar Datos Adicionales, Eliminar Datos Adicionales y Mostrar Datos Adicionales. El anotador realiza la operación seleccionada, el sistema verifica la validez de los datos.
Referencia	R9

Tabla No. 13: CU-9 - Gestionar Datos Adicionales

2.6.3 Diagrama de Casos de Uso del Sistema

Los casos de uso son artefactos narrativos que describen, bajo la forma de acciones y reacciones, el comportamiento del sistema desde el punto de vista del usuario. Por lo tanto, establece un acuerdo entre clientes y desarrolladores sobre las condiciones y posibilidades (requisitos) que debe cumplir el sistema.

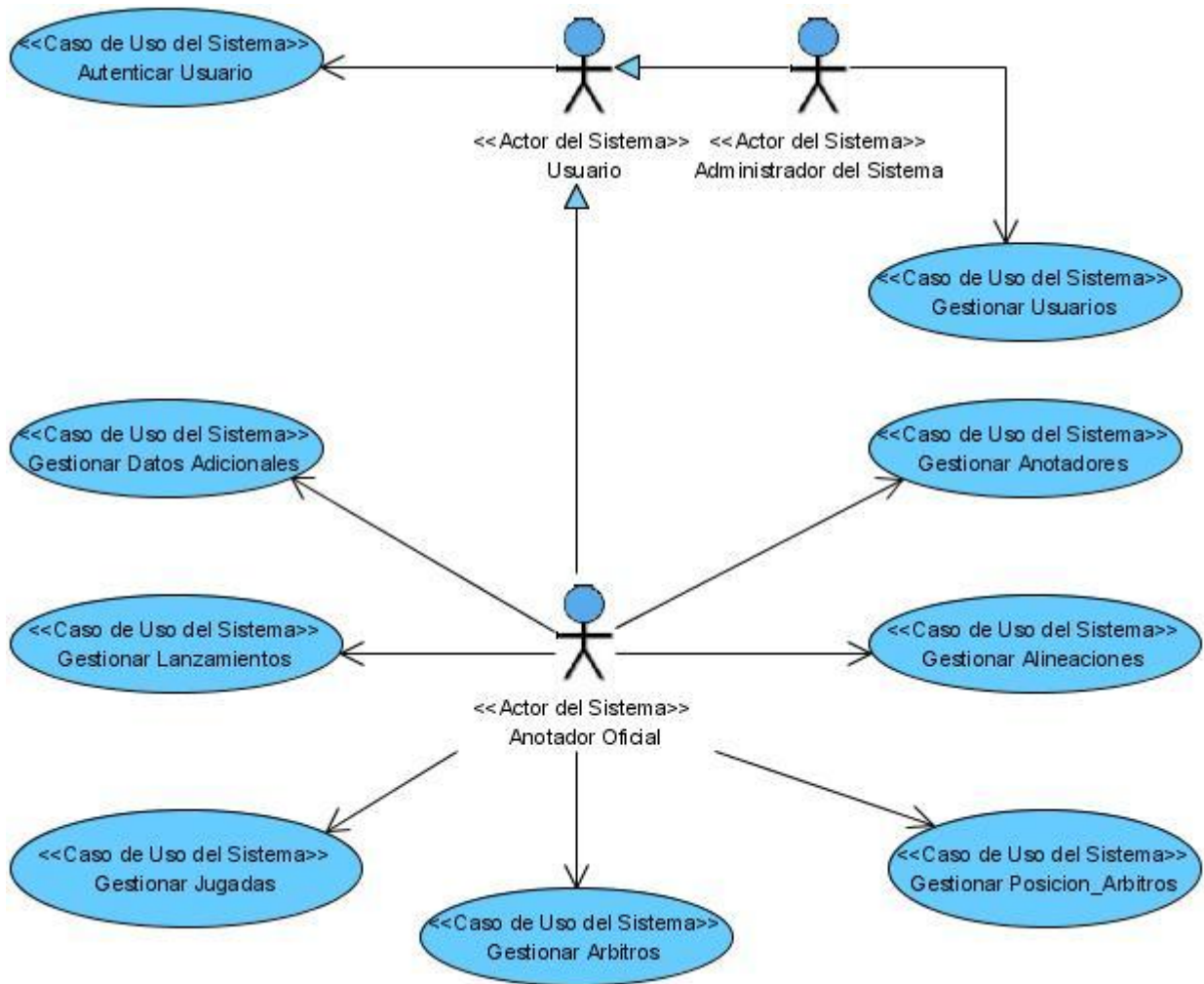


Figura No. 4: Diagrama de Casos de Usos del Sistema

Para profundizar sobre los diagramas de casos de uso del sistema y la descripción textual de los mismos, es necesario remitirse a los anexos 1 y 2.

2.7 Conclusiones

En este capítulo se ha desarrollado un análisis profundo de la situación problemática, lo que permitió a los desarrolladores del trabajo entender el proceso del negocio en el cual se enmarca el sistema a desarrollar. Para darle solución al problema planteado y comenzar con el desarrollo de la aplicación propuesta se construyó el Modelo de Negocio donde se identificaron los actores y trabajadores que intervienen en el mismo, así como las actividades que se realizan durante el transcurso de la anotación.

Se realizó un levantamiento de requisitos para cumplir con las características; tanto funcionales como no funcionales exigidas por el cliente. Esto permitió la modelación del sistema mediante un conjunto de diagramas que son de gran utilidad para la construcción de la aplicación que se propone. Estos diagramas describen el proceso que será objetivo de automatización y sirven como base para la implementación del sistema.

Capítulo 3: Análisis y Diseño

3.1 Introducción

Para el desarrollo de cualquier aplicación la fase de análisis y diseño es de gran importancia. El objetivo principal de esta disciplina es transformar los requerimientos a una especificación que describa cómo implementar el sistema. En este capítulo se dan una serie de elementos que son importantes para el diseño de la aplicación propuesta. Se muestran los diagramas de clases y las relaciones entre las entidades, se describen las clases fundamentales, se muestran los diagramas de interacción y se describen una serie de factores que son importantes para el comienzo de la implementación del sistema.

3.2 Análisis

El análisis fundamentalmente consiste en obtener una visión que se preocupa de ver que hace el sistema de software a desarrollar, por tal motivo este se interesa en los requerimientos funcionales. En este flujo se refinan y estructuran los requisitos obtenidos con anterioridad, profundizando el equipo del proyecto en el dominio de la aplicación lo que les permitirá una mayor comprensión del problema para modelar la solución. (15)

3.2.1 Diagramas de Clases del Análisis

Un Diagrama de clases del análisis es un artefacto en el que se representan los conceptos en un dominio del problema. Representa las cosas del mundo real, no de la implementación automatizada de estas. A continuación se muestran algunos diagramas de clases del análisis correspondiente a los casos de uso del sistema, los otros diagramas se pueden ver en el anexo 3.

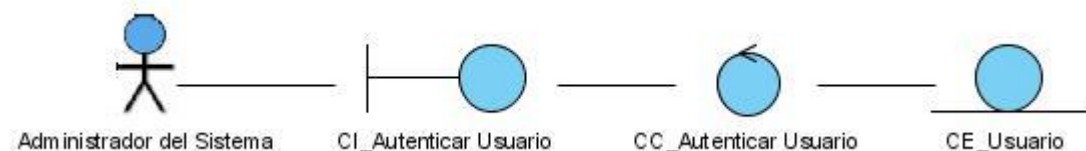


Figura No. 5: Diagrama de Clases del Análisis - CU Autenticar Usuario

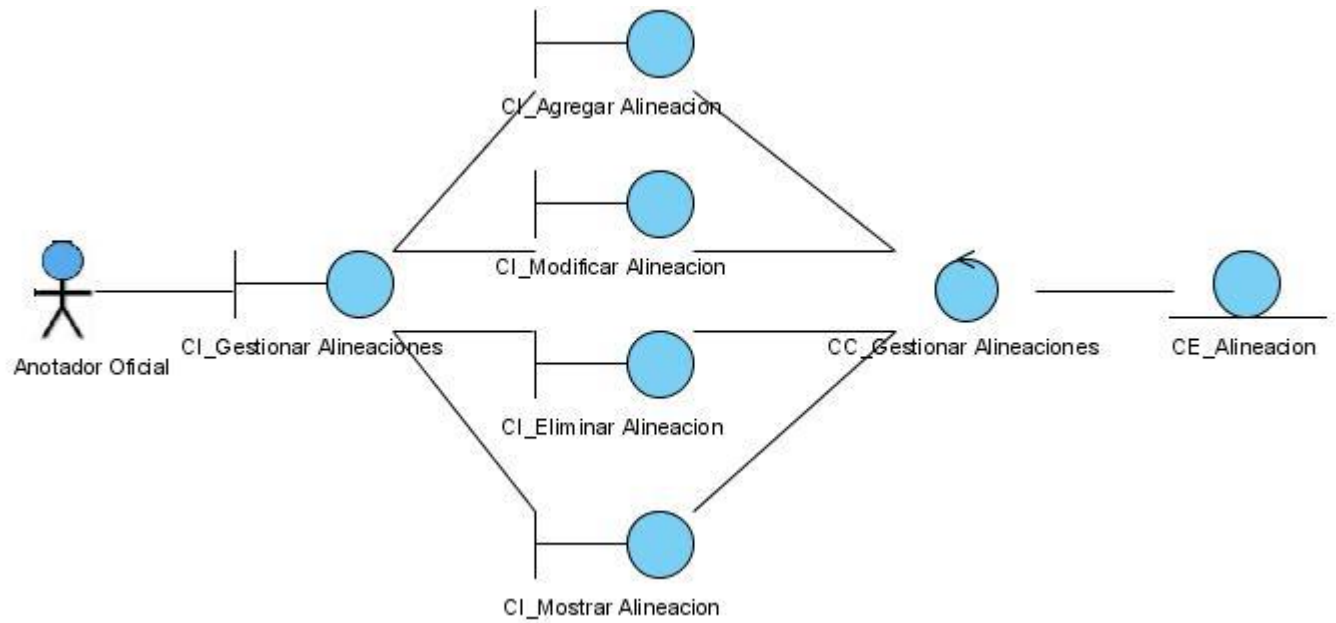


Figura No. 6: Diagrama de Clases del Análisis - CU Gestionar Alineaciones

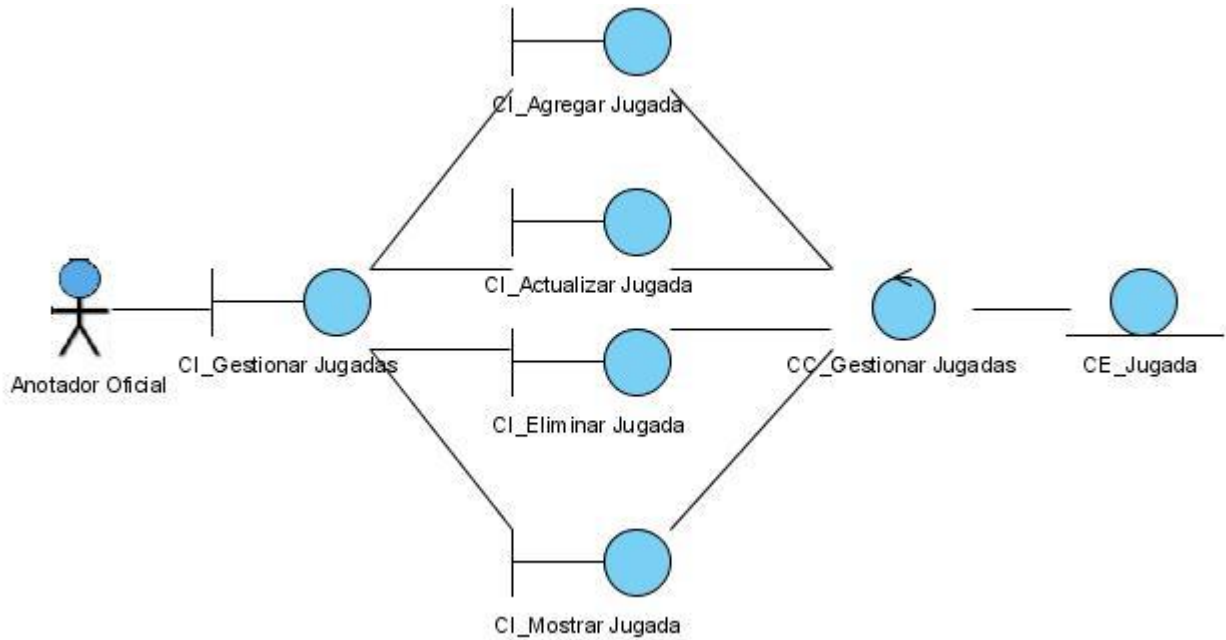


Figura No. 7: Diagrama de Clases del Análisis - CU Gestionar Jugadas

3.2.2 Diagramas de Colaboración

Un diagrama de colaboración destaca la organización de los objetos que participan en una interacción, así como el flujo de mensajes que ocurren durante el desarrollo del evento. A continuación se muestran algunos diagramas de colaboración del análisis, los otros diagramas se pueden ver en el anexo 4.

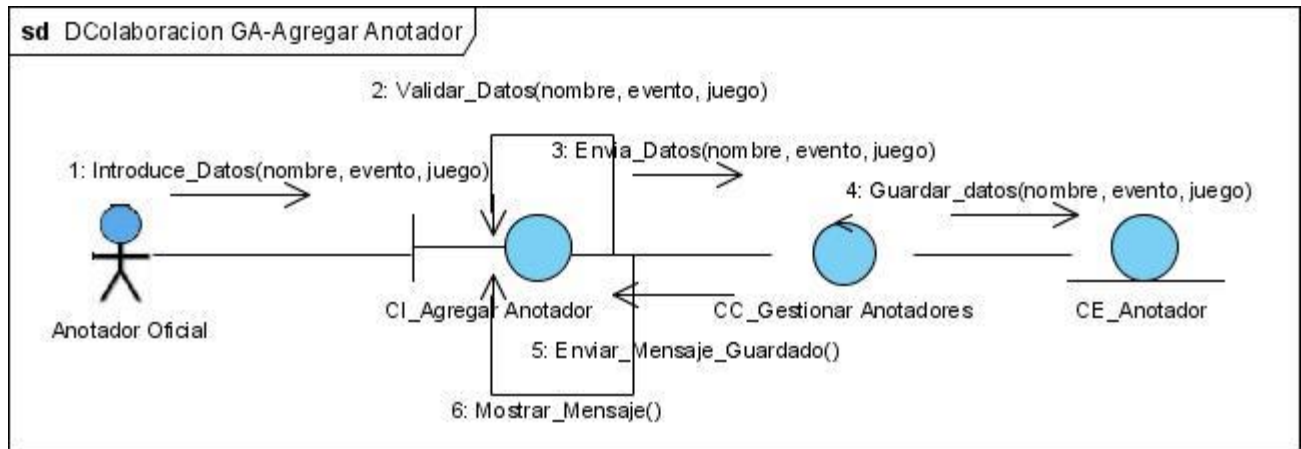


Figura No. 8: Diagrama de Colaboración - CU Gestionar Anotador (Agregar Anotador)

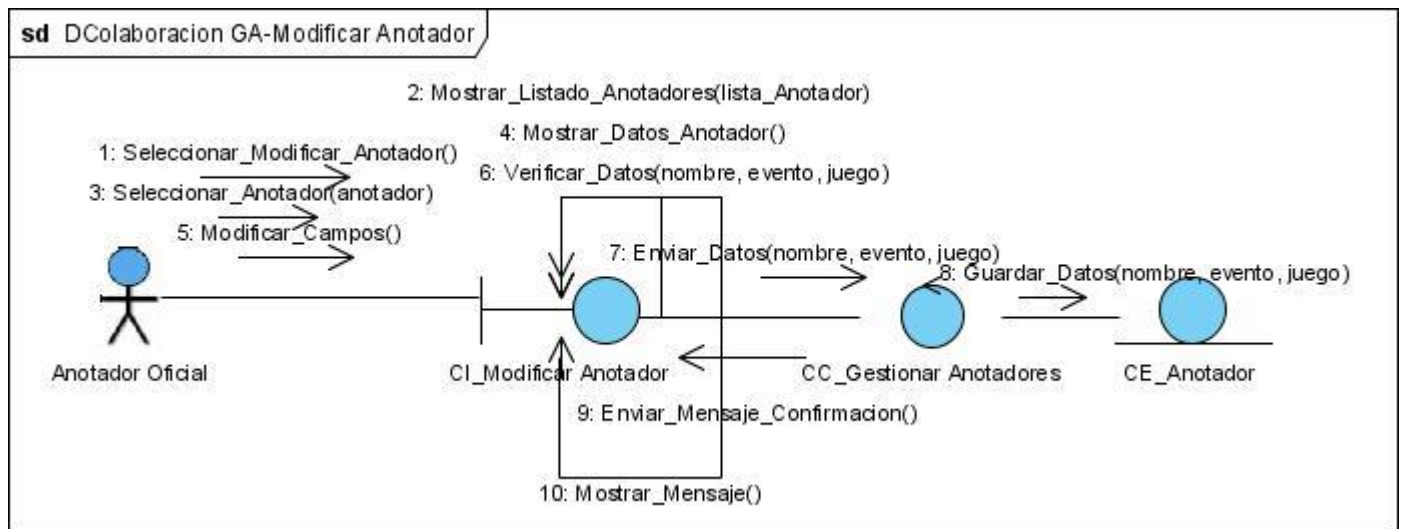


Figura No. 9: Diagrama de Colaboración - CU Gestionar Anotador (Modificar Anotador)

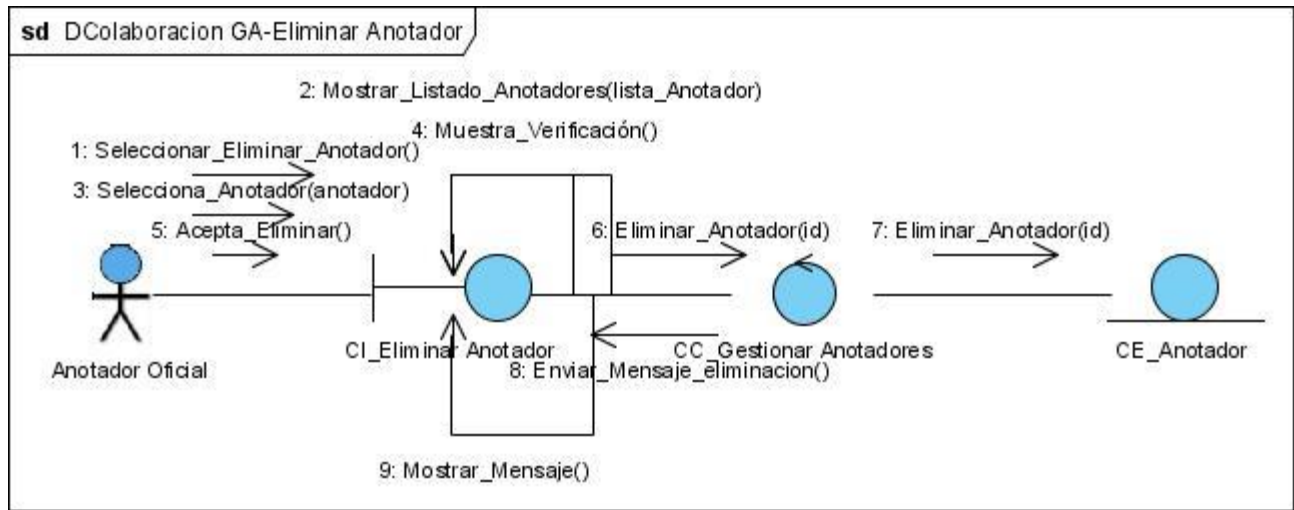


Figura No. 10: Diagrama de Colaboración - CU Gestionar Anotador (Eliminar Anotador)

3.3 Diseño

El diseño es un refinamiento que toma en cuenta los requerimientos no funcionales, por lo cual se centra en como el sistema cumple sus objetivos. En este flujo de trabajo se pretende crear un plano del modelo de implementación, por lo que el grueso de sus actividades se ejecuta fundamentalmente en las últimas iteraciones de la fase de elaboración y las primeras fases de construcción. (15)

3.3.1 Patrones de diseño

Los patrones de diseño constituyen una base fundamental para los desarrolladores de aplicaciones, pues le ayudan a construir sistemas que funcionen de manera óptima, tomando como referencia experiencias anteriores de su trabajo y de los expertos en desarrollo de software.

El uso de patrones es de gran importancia, ya que estos contribuyen a reutilizar diseño, identificando aspectos claves de la estructura de un diseño que puede ser aplicado en una gran cantidad de situaciones. La importancia de la reutilización del diseño no es despreciable, ya que ésta provee de numerosas ventajas: reduce los esfuerzos de desarrollo y mantenimiento, mejora la seguridad, eficiencia y consistencia de los diseños, y proporciona un considerable ahorro en la inversión. Mejoran (aumentan, elevan) la flexibilidad, modularidad y extensibilidad, factores internos e íntimamente relacionados con la

calidad percibida por el usuario. Incrementan el vocabulario de diseño, ayudando a diseñar desde un mayor nivel de abstracción. (16)

Los patrones GRASP, cuyo acrónimo significa General Responsibility Assignment Software Patterns (Patrones Generales de Software para Asignar Responsabilidades), son de gran importancia para el diseño orientado a objetos. Se considera que más que patrones propiamente dichos, son una serie de "buenas prácticas" recomendables para aumentar la eficiencia en el desarrollo de software. Es por ello que los desarrolladores de este trabajo escogieron este patrón para el diseño de la aplicación propuesta. Se utilizaron los patrones: Experto, Creador, Controlador, Alta Cohesión y Bajo Acoplamiento.

3.3.2 Diagrama de Clases de Diseño

Un diagrama de clases de diseño es una representación más concreta que el diagrama de clases del análisis, representa la parte estática del sistema, así como las clases y sus relaciones. A continuación se muestran algunos diagramas de clases de diseño, los otros diagramas se pueden ver en el anexo 5.

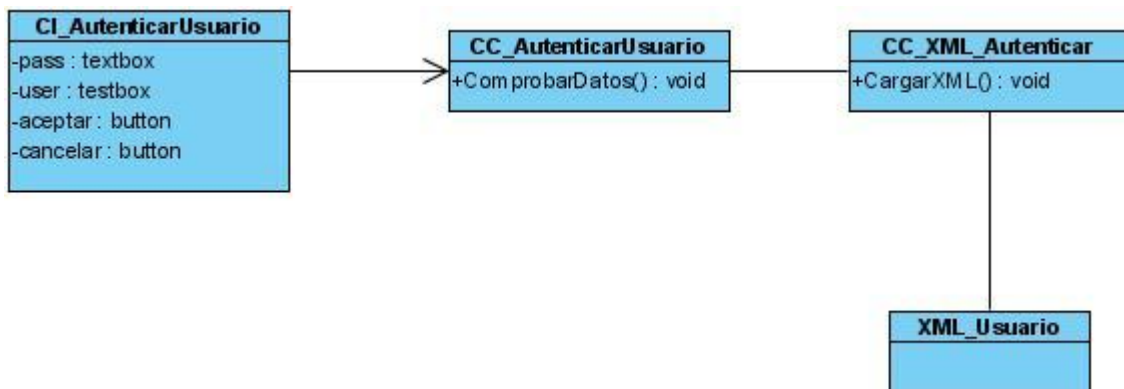


Figura No. 11: Diagrama de Clases de Diseño - CU Autenticar Usuario

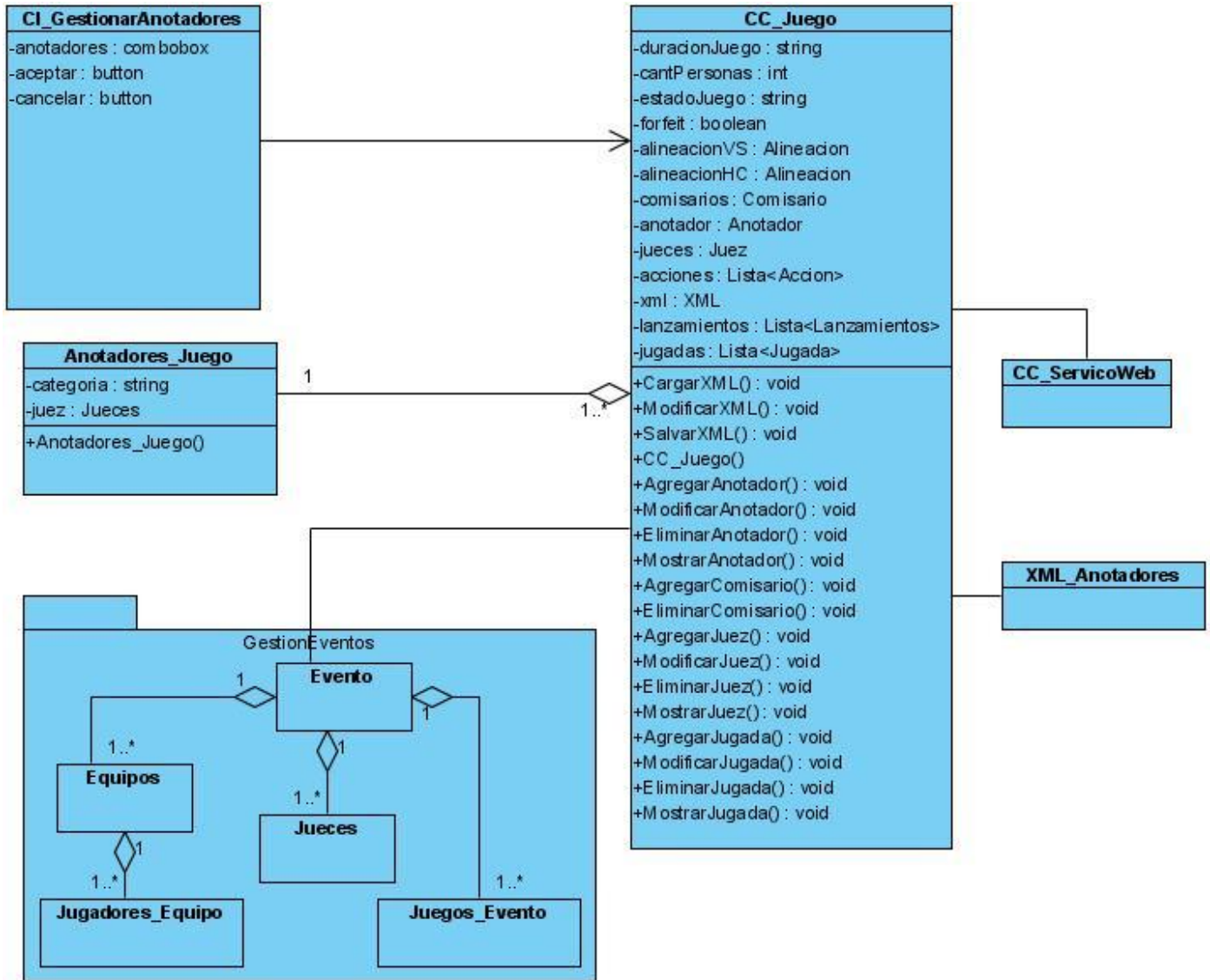


Figura No. 12: Diagrama de Clases de Diseño - CU Gestionar Anotador

3.3.3 Tratamiento de errores

Es de gran importancia para toda aplicación tener un buen tratamiento de errores. Esto posibilita un mejor funcionamiento de la misma, permitiendo que el usuario trabaje con cierta confianza y sin temor a equivocarse, pues el sistema en caso de que se equivoque le mostraría un mensaje de error indicándolo. Es importante también la validación de los datos que se insertan, que donde vaya un número no se pueda escribir una letra, esto evita que se introduzca información no válida en la aplicación.

Existe una amplia comunicación entre el sistema y el usuario, siempre que exista un error, el sistema mostrará un mensaje indicándolo lo más específico posible, con el objetivo de que el usuario pueda corregirlo y continuar trabajando.

3.3.4 Seguridad

La seguridad se lleva a cabo mediante la autenticación por parte de los usuarios que trabajan con la aplicación. Una persona que no sea usuario del sistema no tiene acceso a los datos estadísticos que se manejan, solo los autorizados a trabajar con la aplicación pueden acceder a esta. La contraseña de los usuarios será encriptada para evitar que pueda ser copiada fácilmente.

3.3.5 Interfaz

La interfaz de la aplicación cumple con una serie de facilidades que permiten al usuario realizar su trabajo de una forma más cómoda. Sigue las características que poseen las aplicaciones de escritorio realizadas para Windows. Es funcional y amigable a la vista del usuario que está utilizando la aplicación. La navegabilidad en el sistema se realiza a partir de una serie de menús de tareas que indican las operaciones correspondientes, organizando el trabajo del anotador.

3.3.6 Concepción de la ayuda

Para toda aplicación es de suma importancia dedicar un espacio para la concepción de la ayuda de usuario. Constituye un apoyo esencial para los operadores del sistema, pues en ella pueden consultar dudas acerca del funcionamiento del software que le ayuden a desempeñar su trabajo.

Con el propósito de brindar conocimientos acerca de la operación del sistema, se brinda una ayuda que ilustra los pasos requeridos para la realización de las distintas tareas que se llevan a cabo en el sistema.

Esto debido a que no todos los usuarios que interactúan con el sistema poseen amplios conocimientos informáticos.

3.4 Arquitectura

La arquitectura es el esqueleto o base de una aplicación, en esta se analiza la aplicación desde varios puntos de vista que ayudan a caracterizar estructuralmente el sistema según su naturaleza y complejidad.

La arquitectura en capas define cómo organizar el modelo de diseño a través de capas, que pueden estar físicamente distribuidas, lo cual quiere decir que los componentes de una capa sólo pueden hacer referencia a componentes en capas inmediatamente inferiores. Este patrón es importante porque simplifica la comprensión y la organización del desarrollo de sistemas complejos, reduciendo las dependencias de forma que las capas más bajas no son conscientes de ningún detalle o interfaz de las superiores. Además, ayuda a identificar qué puede ser reutilizado, y proporciona una estructura que ayuda a tomar decisiones sobre qué partes comprar y qué partes construir.

Existen tres capas distribuidas de la siguiente forma:

Capa de Aplicación

Esta capa contiene todas las clases de interfaz de usuarios que representan las pantallas de la aplicación que el usuario ve. Esta capa depende de la capa de Lógica de Negocio y de la capa de Acceso a Datos.

Capa de Lógica de Negocio

La capa de Lógica de Negocio tiene todas las clases controladoras que representan el comportamiento de la aplicación, según los casos de uso. Esta capa representa la frontera del cliente con la capa de Acceso a Datos. La capa de Lógica de Negocio depende de la capa de Acceso a Datos.

Capa de Acceso a Datos

La capa de Acceso a Datos apoya el acceso a la Base de Datos que será realizado mediante la implementación de un Servicio Web.

3.5 Conclusiones

En el presente capítulo se realizaron un conjunto de actividades relacionadas con el flujo de trabajo análisis y diseño. Se construyó el modelo de análisis y el modelo de diseño, en ellos se modelaron las clases y las relaciones entre estas, obteniendo los diagramas de clases de diseño y análisis, diagramas de interacción, en este caso el de colaboración. Se escogieron los patrones de diseño que se utilizarán en la implementación del sistema y se definió la arquitectura a utilizar.

Todo lo anteriormente expuesto permitió a los desarrolladores del trabajo obtener una serie de artefactos que proporcionan una base bien definida para la realización de la implementación del sistema que se propone.

Capítulo 4: Implementación

4.1 Introducción

En la fase de implementación se comienza con el resultado del diseño y se implementan las clases obtenidas en el diseño de la aplicación, todo esto con el objetivo de realizar la construcción del sistema. En este capítulo se realiza el modelo de implementación el cual está compuesto por el diagrama de componentes y el diagrama de despliegue. Se realiza una breve explicación de los estándares de codificación utilizados y el algoritmo de encriptación de datos, este último con el objetivo de mantener la seguridad en el sistema.

4.2 Diagrama de Despliegue

El Diagrama de Despliegue se utiliza para modelar el hardware utilizado en las implementaciones de sistemas y las relaciones entre sus componentes. Los elementos usados por este tipo de diagrama son nodos (representados como un prisma), componentes (representados como una caja rectangular con dos protuberancias del lado izquierdo) y asociaciones que son las relaciones entre los nodos y componentes.

(17)

A continuación se muestra el diagrama de despliegue de la aplicación.

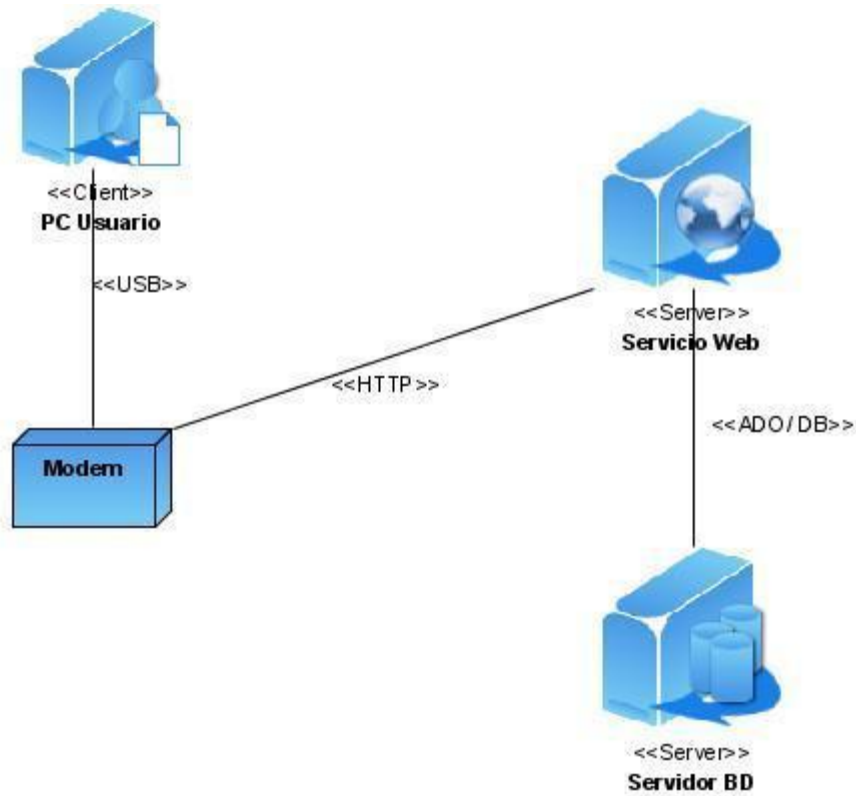


Figura No. 13: Diagrama de Despliegue

4.3 Diagrama de Componentes

Los Diagramas de Componentes ilustran las piezas del software que conformarán un sistema. Un diagrama de Componentes tiene un nivel más alto de abstracción que un diagrama de clase, usualmente un componente se implementa por una o más clases (u objetos) en tiempo de ejecución. Estos son bloques de construcción, un componente eventualmente puede comprender una gran porción de un sistema. (17)

A continuación se muestra el diagrama de componentes de la aplicación.

Métodos: comienzan con letra mayúscula, si son compuestos deben estar separados por `_`. **Ejemplos:** simple: `void Adicionar`, compuesto: `void Mostrar_Alineacion`

Clases: deben comenzar con letra mayúscula.

Comentarios: para comentar una línea se utiliza `//`, si es un bloque de texto se comienza con `/*` y se termina con `*/`.

Instrucciones: las llaves que se utilizan para abrir y cerrar un bloque de instrucciones deben estar al mismo nivel.

Ejemplo:

```
if(x=5)
```

```
{
```

```
...
```

```
}
```

```
else
```

```
{
```

```
...
```

```
}
```

Variables de control de ciclos: las variables utilizadas para los ciclos serán `i`, `j`, `k`.

Componentes Visuales: comienzan identificando que tipo de componente son y a continuación se sigue el estándar especificado para la declaración de las variables. Ejemplo: un `groupbox` que contiene los lanzamientos. `gbLanzamientos`.

4.5 Encriptación de Datos

Es de gran importancia mantener seguridad en los datos para evitar que intrusos puedan corromper la información que se maneja. Para ello se utiliza la encriptación, que no es más que el arte o ciencia de ocultar un mensaje o datos mediante técnicas de cifrado y descifrado utilizando una clave para el manejo de la información encriptada.

Existen varios algoritmos de encriptación, entre ellos se encuentran: AES, ARC4, CuaimaCrypt, DES, TripleDES, DSA, ECDSA, Enigma, IDEA, RSA y TEA / XTEA. Cada uno tiene su forma de codificación y son utilizados según el volumen de información que se maneje y la seguridad que esta requiera. En el caso del presente trabajo se escogió el algoritmo TripleDES, que aunque no es el más óptimo por su lentitud a la hora de cifrar y descifrar, esto se debe a que la longitud de la clave que utiliza es de 192 bits. Para los desarrolladores no constituye un impedimento la utilización de este algoritmo ya que el volumen de información a encriptar no es significativo, solo se reduce a algunos datos de importancia como usuario y contraseña, con el fin de impedir el acceso no autorizado por parte de personas ajenas al uso de la aplicación.

Este algoritmo fue desarrollado por la IBM (International Business Machines) en el año 1978, es una versión que realiza triple cifrado del DES. En la actualidad está siendo sustituido por el AES, sin embargo todavía se utiliza en la mayoría de los medios de pago electrónico. (18)

4.6 Conclusiones

En el capítulo en cuestión se realizaron una serie de tareas correspondientes a la fase de implementación del sistema propuesto, deviniendo en un conjunto de artefactos que tributaron a la obtención de una primera versión funcional de la aplicación. Se realizó el modelo de implementación que contiene el diagrama de componentes que representa los elementos de software que conforman el sistema y el diagrama de despliegue que modela la distribución del hardware que se va a utilizar.

Fue definido el estándar de codificación a utilizar, con el objetivo de evitar errores y lograr una mejor comprensión del código generado, así como el algoritmo de encriptación para cifrar y descifrar la información necesaria.

Conclusiones

Con el desarrollo del trabajo en cuestión se han cumplido los objetivos planteados en el mismo. Como resultado, se ha obtenido el Sistema de Anotación de Juegos de Béisbol que ayudará de manera ostensible la labor de anotación que lleva a cabo la FCBA en todos los juegos desarrollados en la Serie Nacional y otros eventos que tienen lugar en Cuba. Para la concepción de la aplicación fue necesario realizar todo un Proceso de Desarrollo de Software que condujo a la obtención de un producto que cumple las expectativas de la FCBA.

El uso del sistema en cuestión permite a los anotadores desarrollar su trabajo de una forma más amena y eficiente, sin la necesidad de conocer el código de anotación profundamente. Provee la nueva posibilidad de anotar lanzamiento por lanzamiento y las jugadas se insertan mediante una descripción lógica de los sucesos ocurridos durante el desarrollo de las mismas. La codificación es realizada automáticamente por la propia aplicación, minimizando la posibilidad de la existencia de errores a la hora de anotar las jugadas.

El sistema cuenta con las siguientes facilidades:

- ✓ Muestra las estadísticas de los jugadores que participan o participaron durante el juego.
- ✓ Permite anotar lanzamiento por lanzamiento.
- ✓ El usuario debe autenticarse antes de usar el sistema por lo que impide que personal ajeno a la FCBA pueda acceder a la misma.
- ✓ Lleva el conteo de los lanzamientos realizados por cada pitcher que participa en el juego.
- ✓ Muestra un marcador en tiempo real que visualiza el resultado parcial del encuentro.

Todas las ventajas anteriormente descritas hacen del sistema implementado una solución factible que favorece al proceso de anotación de juegos de béisbol, constituyendo una herramienta de trabajo para los anotadores, la cual facilita su desempeño profesional y es el reflejo de una nueva generación de aplicaciones que fomentan el desarrollo del béisbol en Cuba.

Recomendaciones

Teniendo en cuenta todo el proceso de desarrollo llevado a cabo durante la implementación del sistema de anotación, y con el objetivo de contribuir al desarrollo del béisbol en Cuba, se recomienda:

- ✓ Incorporarle al sistema realizado nuevas funcionalidades siempre que la anotación de los juegos de béisbol lo requiera.
- ✓ Desarrollar la funcionalidad que permita la conexión a la base de datos mediante un servicio web.
- ✓ Desarrollar un sistema que permita a todas las personas seguir los juegos en vivo lanzamiento por lanzamiento, teniendo como base el sistema de anotación realizado.
- ✓ Implementar un sistema que permita actualizar el marcador de las pizarras en los estadios, a partir de la información que maneja el sistema de anotación.
- ✓ Continuar desarrollando aplicaciones que permitan la realización más rápida y eficiente de los procesos que ocurren durante un juego de béisbol.

Referencias Bibliográficas

1. **James.** BallStat BallScore. *BallStat*. [En línea] 2008. [Citado el: 4 de Diciembre de 2008.] <http://www.ballstat.com/>.
2. **Pérez, Agustín.** Scielo Chile. *Scielo*. [En línea] 2006. [Citado el: 10 de Diciembre de 2008.] <http://www.scielo.cl/>.
3. **Wells, Don.** Extreme Programming: A gentle introduction. *Extreme Programming*. [En línea] 2006. [Citado el: 15 de Enero de 2009.] <http://www.extremeprogramming.org>.
4. **Albaladejo, Xavier.** proyectos agiles.org. *proyectos agiles.org*. [En línea] 2008. [Citado el: 15 de Enero de 2009.] <http://www.proyectosagiles.org>.
5. **Innova, Grupo Soluciones.** *GSInnova*. [En línea] 2007. [Citado el: 15 de Enero de 2009.] <http://www.rational.com.ar>.
6. **Echarte, Francisco.** eslomas.com. *EsLoMas*. [En línea] 2008. [Citado el: 20 de Enero de 2009.] <http://www.eslomas.com>.
7. **vico open modeling, s.l.** Vico.org: Open Modeling. *Vico.org*. [En línea] 2008. [Citado el: 20 de Enero de 2009.] <http://www.vico.org>.
8. **Moral, Juan Antonio Breña.** Desarrolloweb.com. *Desarrolloweb.com*. [En línea] 2004. [Citado el: 21 de Enero de 2009.] <http://www.desarrolloweb.com>.
9. **Corporation, Microsoft.** msdn. *msdn*. [En línea] 2008. [Citado el: 22 de Enero de 2009.] <http://msdn.microsoft.com>.
10. **International, Visual Paradigm.** Visual Paradigm. *Visual Paradigm*. [En línea] 2008. [Citado el: 22 de Enero de 2009.] <http://www.visual-paradigm.com>.
11. **PostgreSQL.** *PostgreSQL*. [En línea] 2009. [Citado el: 23 de Enero de 2009.] <http://www.postgresql.org>.

12. **Corporation, Microsoft.** msdn. *msdn*. [En línea] 2009. [Citado el: 3 de Febrero de 2009.] <http://msdn.microsoft.com>.
13. **LWP.** *LWP*. [En línea] 2009. [Citado el: 3 de Febrero de 2009.] <http://www.lawebdelprogramador.com>.
14. **Turner, Raymond.** Computer Science & Electronics Engineering. *Computer Science*. [En línea] 2005. [Citado el: 10 de Marzo de 2009.] http://cswww.essex.ac.uk/staff/turnr/cswww.essex.ac.uk_files/Mypapers/foundations-specification.pdf.
15. **Presman, Roger.** *Ingeniería del Software. Un enfoque práctico*. s.l. : McGraw-Hill/Interamericana de España, 2002.
16. **Larman.** *UML Y Patrones. Introducción al análisis y diseño orientado a objetos*. 2000.
17. **Ltd, Sparx Systems Pty.** Sparx Systems. *Sparx Systems*. [En línea] 2007. [Citado el: 10 de Abril de 2009.] <http://www.sparxsystems.com.ar>.
18. TextosCientificos.com. *TextosCientificos.com*. [En línea] 4 de Mayo de 2005. <http://www.textoscientificos.com/>.

Bibliografía Consultada

Pérez Julien, José y Mendez Rodríguez, Efrain. *Reglas Oficiales de Béisbol 2008*. Ciudad de la Habana : s.n., 2008.

González Valnuerdi, Eduardo. Sistema Codificado de Anotación Béisbol. *Manual*. Ciudad de la Habana : CINID-INDER, 1992.

Merelo, J. J. Geneura. [En línea] <http://geneura.ugr.es/~jmerelo/tutoriales/csharp/>.

Echarte, Francisco. eslomas.com. *EsLoMas*. [En línea] 2008. <http://www.eslomas.com>.

W3C. *World Wide Web Consortium*. [En línea] <http://www.w3c.es/divulgacion/guiasbreves/tecnologiasXML>.

LWP. *La Web del Programador*. [En línea] <http://www.lawebdelprogramador.com/>.

djk. *Devjoker*. [En línea] [http://www.devjoker.com/contenidos/Articulos/280/Desencriptar-Texto-y-variables-con-NET-VB-y-C-\(segunda-parte\).aspx](http://www.devjoker.com/contenidos/Articulos/280/Desencriptar-Texto-y-variables-con-NET-VB-y-C-(segunda-parte).aspx).

Larman. *UML Y Patrones. Introducción al análisis y diseño orientado a objetos*. 2000.

BOOCH, Grady, RUMBAUGH, James, JACOBSON, Ivar; "El lenguaje unificado de modelado". 2000. Addison Wesley.

vico.org. *Open Modeling*. [En línea] <http://www.vico.org/>.

Presman, Roger. *Ingeniería del Software. Un enfoque práctico*. s.l. : McGraw-Hill/Interamericana de España, 2002.

mono. *Mono*. [En línea] http://www.mono-project.com/Main_Page.

Visual Paradigm. [En línea] <http://www.visual-paradigm.com/>.

Corporation, Microsoft. msdn. *msdn*. [En línea] 2009. <http://msdn.microsoft.com>.

Innova, Grupo Soluciones. *GSInnova*. [En línea] 2007. <http://www.rational.com.ar>.

Seco, José Antonio González. El lenguaje de programación C#. *Programación en Catellano*. [En línea] <http://programacion.com/tutorial.php?id=csharp>.

Anexos

Anexo 1: Diagramas de Casos de Uso por Paquetes

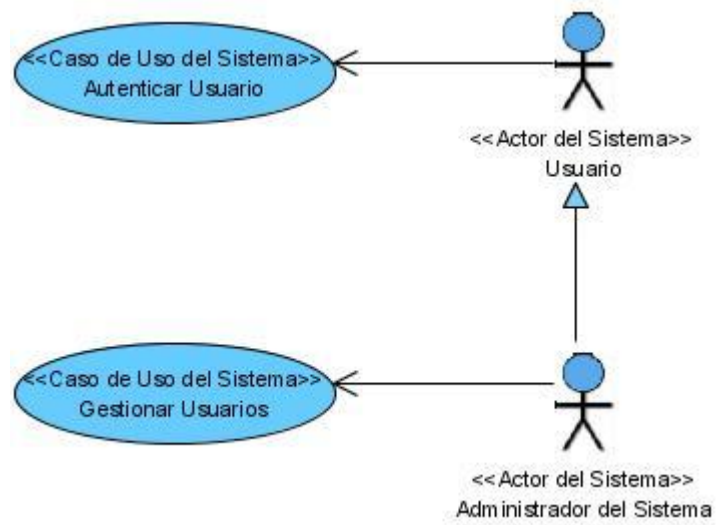


Figura No. 15: Diagrama de Casos de Usos - Paquete Seguridad

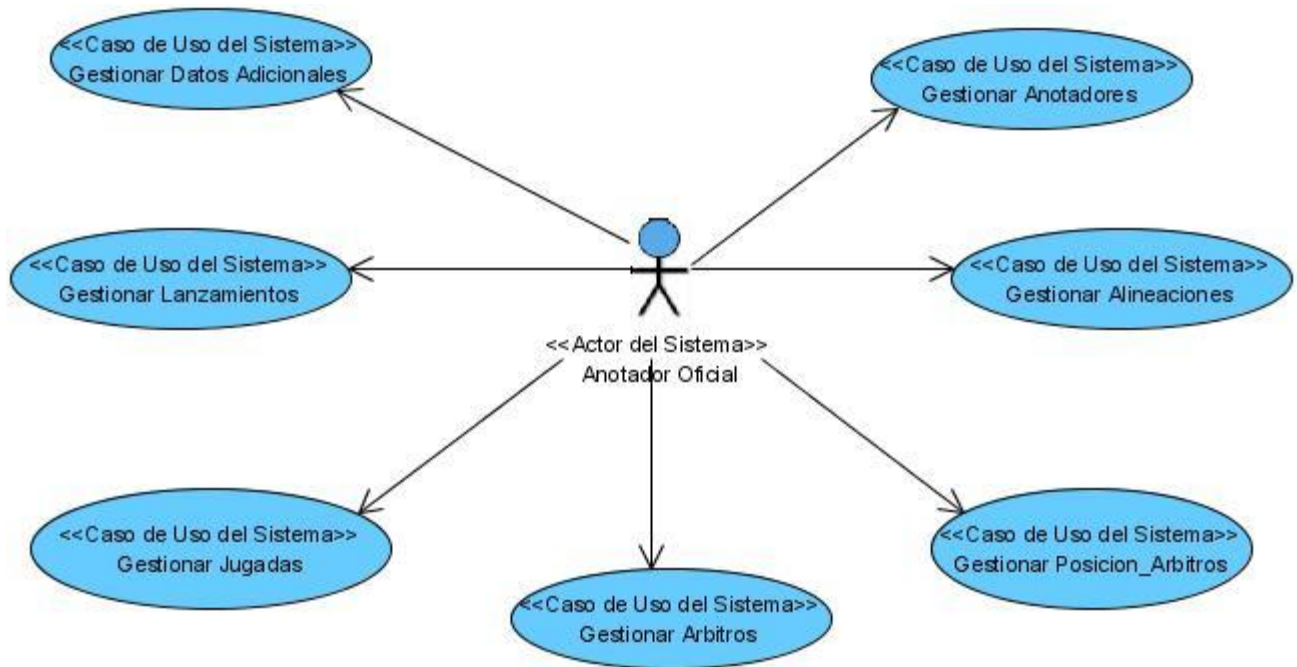


Figura No. 16: Diagrama de Casos de Usos - Paquete Anotación

Anexo 2: Casos de Uso del sistema y su descripción

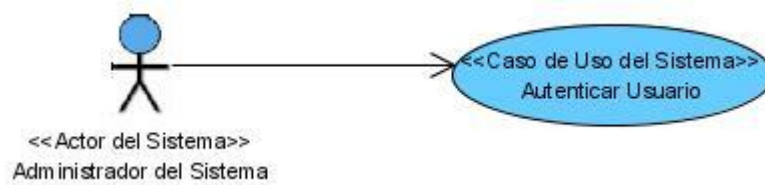


Figura No. 17: Diagrama de Caso de Uso del Sistema - Autenticar Usuario

Caso de Uso:	Autenticar Usuario	
Actores:	Administrador del Sistema	
Resumen:	El caso de uso comienza cuando el administrador selecciona la opción de autenticarse en el sistema, el administrador introduce los datos necesarios y se le brinda los permisos que necesita para trabajar con la aplicación.	
Precondiciones:		
Referencias	R1	
Prioridad	Crítico	
Flujo Normal de Eventos		
Acción del Actor	Respuesta del Sistema	
<p>1. El administrador selecciona la opción de autenticarse.</p> <p>3. El administrador introduce los datos solicitados por el sistema.</p>	<p>2. El sistema solicita nombre de usuario y contraseña.</p> <p>4. El sistema verifica si los datos introducidos por el administrador son correctos.</p> <p>5. Si los datos son correctos el sistema da los permisos necesarios al administrador para acceder a la aplicación. Termina el caso de uso.</p>	
Flujos Alternos		
Acción del Actor	Respuesta del Sistema	
	<p>5.1 Si los datos son incorrectos el sistema muestra un mensaje de error al administrador y retorna a la acción 2.</p>	

Tabla No. 14: Descripción del Caso de Uso - Autenticar Usuario

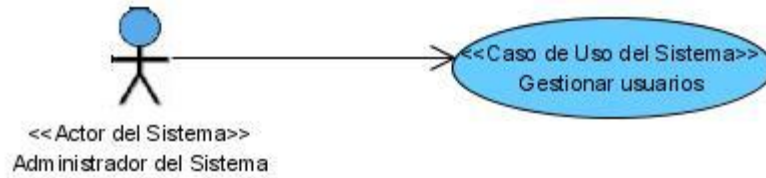


Figura No. 18: Diagrama de Caso de Uso del Sistema - Gestionar Usuarios

Caso de Uso:	Gestionar Usuarios
Actores:	Administrador del Sistema
Resumen:	El caso de uso inicia cuando el administrador selecciona la opción Gestionar Usuarios, escoge una de las opciones que se pueden realizar, Adicionar Usuarios, Modificar Usuarios y Eliminar Usuarios. El administrador realiza la operación deseada, el sistema actualiza los datos y termina el caso de uso.
Precondiciones:	El administrador tiene que estar registrado en el sistema.
Referencias	R2
Prioridad	Crítico
Flujo Normal de Eventos	
Acción del Actor	Respuesta del Sistema
1. El administrador del sistema selecciona la opción Gestionar Usuarios.	2. El sistema muestra las opciones a escoger: - Agregar Usuarios - Modificar Usuarios - Eliminar Usuarios

Sección 1: “Agregar Usuarios”

<p>1. El administrador del sistema selecciona la opción Agregar Usuarios.</p> <p>3. El administrador del sistema llena los campos requeridos.</p>	<p>2. El sistema muestra los campos que deben ser llenados por el administrador del sistema, usuario y contraseña.</p> <p>4. El sistema verifica que los datos insertados son correctos.</p> <p>5. El sistema guarda los datos y termina el caso de uso.</p>
---	--

Flujos Alternos

Acción del Actor	Respuesta del Sistema
	<p>4.1 Si los datos introducidos por el administrador del sistema son incorrectos, el sistema envía un mensaje de error indicándolo y permite retornar al paso 3.</p>

Sección 2: “Modificar Usuarios”

<p>1. El administrador del sistema selecciona la opción Modificar Usuarios.</p> <p>3. El administrador del sistema escoge el usuario que desea modificar.</p> <p>5. El administrador del sistema modifica los campos deseados.</p>	<p>2. El sistema muestra todos los usuarios que existen en la aplicación.</p> <p>4. El sistema muestra todos los campos que pueden ser modificados, usuario y contraseña.</p> <p>6. El sistema verifica que los datos insertados son correctos.</p> <p>7. El sistema guarda los datos y termina el caso de uso.</p>
--	---

Flujos Alternos

Acción del Actor	Respuesta del Sistema

	6.1 Si los datos introducidos por el administrador del sistema son incorrectos, el sistema envía un mensaje de error indicándolo y permite retornar al paso 5.
Sección 3: “Eliminar Usuarios”	
1. El administrador del sistema selecciona la opción Eliminar Usuarios.	2. El sistema muestra todos los usuarios que existen en la aplicación.
3. El administrador del sistema escoge el usuario que desea eliminar.	4. El sistema muestra un mensaje para que el administrador del sistema confirme si desea o no eliminar el usuario.
5. El administrador del sistema decide si desea o no eliminar el usuario seleccionado.	6. Si el administrador del sistema confirma la decisión de eliminar el usuario, el sistema elimina el usuario escogido y termina el caso de uso.
Flujos Alternos	
Acción del Actor	Respuesta del Sistema
5.1 El administrador del sistema decide no eliminar el usuario, cancela la acción y termina el caso de uso.	

Tabla No. 15: Descripción del Caso de Uso - Gestionar Usuarios



Figura No. 19: Diagrama de Caso de Uso del Sistema - Gestionar Anotadores

Caso de Uso:	Gestionar Anotadores
Actores:	Anotador Oficial
Resumen:	El caso de uso inicia cuando el anotador oficial selecciona la opción Gestionar Anotadores, escoge una de las opciones que se pueden realizar, Mostrar Anotador, Adicionar Anotador, Actualizar Anotador y Eliminar Anotador. El anotador oficial realiza la operación escogida, el sistema verifica la validez de los datos, en caso de ser correctos los actualiza y termina el caso de uso.
Precondiciones:	El anotador oficial tiene que estar registrado en el sistema.
Referencias	R3
Prioridad	Crítico

Flujo Normal de Eventos

Acción del Actor	Respuesta del Sistema
1. El anotador oficial selecciona la opción Gestionar Anotadores.	2. El sistema muestra las opciones a escoger: <ul style="list-style-type: none"> - Agregar Anotador - Actualizar Anotador - Eliminar Anotador - Mostrar Anotador

Sección 1: “Agregar Anotador”

1. El anotador oficial selecciona la opción Agregar Anotador.	2. El sistema muestra un formulario con los campos requeridos para adicionar un nuevo Anotador, estos son: anotador y categoría. 4. El sistema verifica que los datos insertados por el
---	--

<p>3. El anotador inserta los datos del anotador en el sistema.</p>	<p>anotador sean válidos. 5. En caso de ser válidos los datos insertados, la aplicación almacena los datos y termina el caso de uso.</p>
Flujos Alternos	
Acción del Actor	Respuesta del Sistema
	<p>4.1 Si los datos son incorrectos el sistema muestra un mensaje de error al anotador oficial y retorna a la acción 2.</p>
Sección 2: “Actualizar Anotador”	
<p>1. El anotador oficial selecciona la opción Actualizar Anotador. 3. El anotador hace los cambios pertinentes para la actualización del anotador en cuestión.</p>	<p>2. El sistema muestra un formulario con los campos requeridos para actualizar un anotador, estos son: anotador y categoría. 4. El sistema verifica que los datos actualizados por el anotador sean válidos. 5. En caso de ser válidos los datos actualizados, la aplicación almacena los datos y termina el caso de uso.</p>
Flujos Alternos	
Acción del Actor	Respuesta del Sistema
	<p>4.1 Si los datos son incorrectos el sistema muestra</p>

	un mensaje de error al anotador oficial y retorna a la acción 2.
Sección 3: “Eliminar Anotador”	
<p>1. El anotador oficial selecciona la opción Eliminar Anotador.</p> <p>3. El anotador oficial escoge el anotador a eliminar.</p> <p>5. El anotador oficial decide si desea o no eliminar el anotador seleccionado.</p>	<p>2. El sistema muestra un listado de los anotadores existentes en la aplicación.</p> <p>4. El sistema muestra un mensaje para que el anotador oficial confirme si desea o no eliminar el anotador.</p> <p>6. Si el anotador oficial confirma la decisión de eliminar el anotador, el sistema elimina el anotador escogido y termina el caso de uso.</p>
Flujos Alternos	
Acción del Actor	Respuesta del Sistema
5.1 El anotador oficial decide no eliminar el anotador, cancela la acción y termina el caso de uso.	
Sección 4: “Mostrar Anotador”	
<p>1. El anotador oficial selecciona la opción Mostrar Anotador.</p> <p>3. El anotador oficial escoge el anotador a mostrar.</p>	<p>2. El sistema muestra un listado de los anotadores.</p> <p>4. El sistema procede a mostrar los datos del anotador y termina el caso de uso.</p>

Tabla No. 16: Descripción del Caso de Uso - Gestionar Anotadores

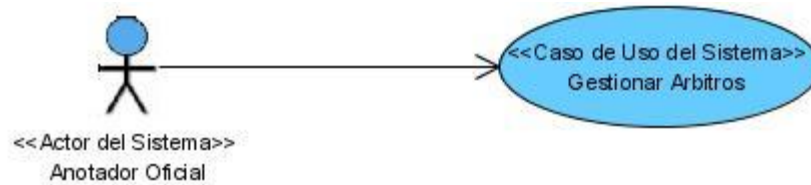


Figura No. 20: Diagrama de Caso de Uso del Sistema - Gestionar Árbitros

Caso de Uso:	Gestionar Árbitros
Actores:	Anotador Oficial
Resumen:	El caso de uso inicia cuando el anotador oficial selecciona la opción Gestionar Árbitros, elige una de las opciones que se pueden efectuar; Mostrar Árbitros, Adicionar Árbitros, Actualizar Árbitros y Eliminar Árbitros. El anotador realiza la operación escogida, el sistema verifica la validez de los datos, en caso de ser correctos los actualiza y termina el caso de uso.
Precondiciones:	El anotador oficial tiene que estar registrado en el sistema. El anotador oficial debe seleccionar el juego en el que se gestionarán los árbitros.
Referencias	R4
Prioridad	Crítico

Flujo Normal de Eventos

Acción del Actor	Respuesta del Sistema
1. El anotador oficial selecciona la opción Gestionar Árbitros.	2. El sistema muestra las opciones a escoger: - Agregar Árbitros - Actualizar Árbitros - Eliminar Árbitros

	- Mostrar Árbitros
Sección 1: “Agregar Árbitros”	
<p>1. El anotador oficial selecciona la opción Agregar Árbitros.</p> <p>3. El anotador inserta el juez en el sistema.</p>	<p>2. El sistema muestra un formulario con los campos necesarios para adicionar un nuevo Árbitro, estos son: juez y posición.</p> <p>4. El sistema verifica que los datos insertados por el anotador sean válidos.</p> <p>5. En caso de ser válidos los datos insertados, la aplicación almacena los datos y termina el caso de uso.</p>
Flujos Alternos	
Acción del Actor	Respuesta del Sistema
	<p>4.1 Si los datos son incorrectos el sistema muestra un mensaje de error al usuario y retorna a la acción 2.</p>
Sección 2: “Actualizar Árbitros”	
<p>1. El anotador oficial selecciona la opción Actualizar Árbitros.</p> <p>3. El anotador hace los cambios pertinentes para la actualización del árbitro.</p>	<p>2. El sistema muestra un formulario con los campos necesarios para actualizar un árbitro, estos son: juez y posición.</p> <p>4. El sistema verifica que los datos actualizados por el anotador sean válidos.</p> <p>5. En caso de ser válidos los datos actualizados, la aplicación almacena los datos y termina el caso de</p>

	USO.
Flujos Alternos	
Acción del Actor	Respuesta del Sistema
	4.1 Si los datos son incorrectos el sistema muestra un mensaje de error al usuario y retorna a la acción 2.
Sección 3: “Eliminar Árbitros”	
1. El anotador oficial selecciona la opción Eliminar Árbitros. 3. El anotador escoge el árbitro a eliminar. 5. El anotador oficial decide si desea o no eliminar el árbitro seleccionado.	2. El sistema muestra un listado de los árbitros existentes en el juego. 4. El sistema muestra un mensaje para que el anotador oficial confirme si desea o no eliminar el árbitro seleccionado. 6. Si el anotador oficial confirma la decisión de eliminar el árbitro, el sistema elimina el árbitro escogido y termina el caso de uso.
Flujos Alternos	
Acción del Actor	Respuesta del Sistema
5.1 El anotador oficial decide no eliminar el árbitro, cancela la acción y termina el caso de uso.	

Sección 4: “Mostrar Árbitros”	
1. El anotador oficial selecciona la opción Mostrar Árbitros.	2. El sistema muestra un listado de los árbitros que intervienen en el partido y termina el caso de uso.
Flujos Alternos	
Acción del Actor	Respuesta del Sistema
Poscondiciones	

Tabla No. 17: Descripción del Caso de Uso - Gestionar Árbitros



Figura No. 21: Diagrama de Caso de Uso del Sistema - Gestionar Posición de los Árbitros

Caso de Uso:	Gestionar Posición_ Árbitros
Actores:	Anotador Oficial
Resumen:	El caso de uso inicia cuando el anotador oficial selecciona la opción Gestionar Posición de los Árbitros, escoge una de las opciones que se pueden realizar; Agregar Posición, Actualizar Posición y Eliminar Posición. El anotador realiza la operación seleccionada, el sistema verifica la validez de los datos, en caso de ser correctos los actualiza y termina el

	caso de uso.
Precondiciones:	<p>El anotador oficial tiene que estar registrado en el sistema.</p> <p>El anotador oficial debe seleccionar el juego en el que se gestionarán las posiciones de los árbitros participantes en el mismo.</p>
Referencias	R5
Prioridad	Crítico

Flujo Normal de Eventos

Acción del Actor	Respuesta del Sistema
<p>1. El anotador oficial selecciona la opción Gestionar Posición de los Árbitros.</p> <p>3. El anotador escoge el árbitro al que desea gestionar su posición.</p>	<p>2. El sistema muestra los árbitros existentes en el juego.</p> <p>4. El sistema muestra las opciones a escoger:</p> <ul style="list-style-type: none"> - Agregar Posición - Actualizar Posición - Eliminar Posición

Sección 1: "Agregar Posición"

<p>1. El anotador oficial selecciona la opción Agregar Posición.</p> <p>3. El anotador inserta los datos de la posición en el sistema.</p>	<p>2. El sistema muestra un formulario con los campos necesarios para adicionar una nueva posición, estos son: Posición y Sigla.</p> <p>4. El sistema verifica que los datos insertados por el anotador sean válidos.</p> <p>5. En caso de ser válidos los datos insertados, la aplicación almacena los datos y termina el caso de</p>
--	--

	uso.
Flujos Alternos	
Acción del Actor	Respuesta del Sistema
	4.1 Si los datos son incorrectos el sistema muestra un mensaje de error al usuario y retorna a la acción 2.
Sección 2: "Actualizar Posición"	
1. El anotador oficial selecciona la opción Actualizar Posición. 3. El anotador oficial hace los cambios pertinentes para la actualización la posición deseada.	2. El sistema muestra un formulario con los campos requeridos para actualizar una posición, estos son: Posición y Sigla. 4. El sistema verifica que los datos actualizados por el anotador sean válidos. 5. En caso de ser válidos los datos actualizados, la aplicación almacena los datos y termina el caso de uso.
Flujos Alternos	
Acción del Actor	Respuesta del Sistema
	4.1. Si los datos son incorrectos el sistema muestra un mensaje de error al usuario y retorna a la acción 2.

Sección 3: “Eliminar Posición de los Árbitros”	
<p>1. El anotador oficial selecciona la opción Eliminar Posición.</p> <p>3. El anotador escoge la posición que desea eliminar.</p> <p>5. El anotador oficial decide si desea o no eliminar la posición del árbitro seleccionado.</p>	<p>2. El sistema muestra un listado de las posiciones que contiene el árbitro seleccionado.</p> <p>4. El sistema muestra un mensaje para que el anotador oficial confirme si desea o no eliminar la posición seleccionada.</p> <p>6. Si el anotador oficial confirma la decisión de eliminar la posición, el sistema elimina la posición escogida y termina el caso de uso.</p>
Flujos Alternos	
Acción del Actor	Respuesta del Sistema
<p>5.1 El anotador oficial decide no eliminar la posición del árbitro seleccionado, cancela la acción y termina el caso de uso.</p>	
Poscondiciones	

Tabla No. 18: Descripción del Caso de Uso - Gestionar Posición de los Jueces



Figura No. 22: Diagrama de Caso de Uso del Sistema - Gestionar Alineaciones

Caso de Uso:	Gestionar Alineaciones
Actores:	Anotador Oficial
Resumen:	El caso de uso inicia cuando el anotador oficial selecciona la opción Gestionar Alineaciones, escoge una de las opciones que se pueden realizar; Agregar Alineación, Actualizar Alineación, Eliminar Alineación y Mostrar Alineación. El anotador realiza la operación seleccionada, el sistema verifica la validez de los datos, en caso de ser correctos los actualiza y termina el caso de uso.
Precondiciones:	El anotador oficial tiene que estar registrado en el sistema. El anotador oficial debe seleccionar el juego en el que se gestionarán las alineaciones de los equipos participantes en el mismo.
Referencias	R6
Prioridad	Crítico

Flujo Normal de Eventos

Acción del Actor	Respuesta del Sistema
1. El anotador oficial selecciona la opción Gestionar Alineación.	2. El sistema muestra las opciones a escoger: - Agregar Alineación - Actualizar Alineación - Eliminar Alineación - Mostrar Alineación

Sección 1: “Agregar Alineación”

1. El anotador oficial selecciona la opción Agregar Alineación.	2. El sistema muestra un formulario con los campos necesarios para adicionar una alineación, Jugador y Posición.
---	--

<p>3. El anotador oficial inserta los datos de la alineación.</p>	<p>4. El sistema verifica que los datos insertados por el anotador sean válidos.</p> <p>5. En caso de ser válidos los datos insertados, la aplicación almacena los datos y termina el caso de uso.</p>
Flujos Alternos	
Acción del Actor	Respuesta del Sistema
	<p>4.1 Si los datos son incorrectos el sistema muestra un mensaje de error al anotador oficial y retorna a la acción 2.</p>
Sección 2: “Actualizar Alineación”	
<p>1. El anotador oficial selecciona la opción Actualizar Alineación</p> <p>3. El anotador oficial selecciona la alineación que desea cambiar.</p> <p>5. El anotador oficial modifica los datos deseados.</p>	<p>2. El sistema muestra las alineaciones pertenecientes al juego.</p> <p>4. El sistema muestra una forma con los campos necesarios para modificar la alineación seleccionada, Jugador y Posición.</p> <p>6. El sistema verifica que los datos modificados por el anotador sean válidos.</p> <p>7. En caso de ser válidos los datos actualizados, la aplicación almacena los datos y termina el caso de uso.</p>

Flujos Alternos	
Acción del Actor	Respuesta del Sistema
	6.1. Si los datos son incorrectos el sistema muestra un mensaje de error al usuario y retorna a la acción 4.
Sección 3: “Eliminar Alineación”	
1. El anotador oficial selecciona la opción Eliminar Alineación. 3. El anotador oficial selecciona la alineación que desea eliminar. 5. El anotador oficial decide si desea o no eliminar la alineación seleccionada.	2. El sistema muestra las alineaciones pertenecientes al juego. 4. El sistema muestra un mensaje para que el anotador oficial confirme si desea o no eliminar la alineación seleccionada. 6. Si el anotador oficial confirma la decisión de eliminar la alineación, el sistema elimina la alineación escogida y termina el caso de uso.
Flujos Alternos	
Acción del Actor	Respuesta del Sistema
5.1 El anotador oficial decide no eliminar la alineación seleccionada, cancela la acción y termina el caso de uso.	
Sección 4: “Mostrar Alineación”	
1. El anotador oficial selecciona la opción Mostrar Alineación.	2. El sistema muestra las alineaciones pertenecientes al juego y termina el caso de uso.

Flujos Alternos	
Acción del Actor	Respuesta del Sistema

Tabla No. 19: Descripción del Caso de Uso - Gestionar Alineaciones



Figura No. 23: Diagrama de Caso de Uso del Sistema - Gestionar Jugadas

Caso de Uso:	Gestionar Jugadas
Actores:	Anotador Oficial
Resumen:	El caso de uso inicia cuando el anotador oficial selecciona la opción Gestionar Jugadas, escoge una de las opciones que se pueden realizar; Agregar Jugadas, Actualizar Jugadas, Eliminar Jugadas y Mostrar Jugadas. El anotador realiza la operación seleccionada, el sistema verifica la validez de los datos, en caso de ser correctos los actualiza y termina el caso de uso.
Precondiciones:	El anotador oficial tiene que estar registrado en el sistema. El anotador oficial debe seleccionar el juego en el que se gestionarán las jugadas de los equipos participantes en el mismo.
Referencias	R7

Prioridad	Crítico	
Flujo Normal de Eventos		
Acción del Actor		Respuesta del Sistema
1. El anotador oficial selecciona la opción Gestionar Jugadas.		2. El sistema muestra las opciones a escoger: <ul style="list-style-type: none"> - Agregar Jugadas - Actualizar Jugadas - Eliminar Jugadas - Mostrar Jugadas
Sección 1: "Agregar Jugadas"		
1. El anotador oficial selecciona la opción Agregar Jugadas 3. El anotador oficial inserta los datos según la acción ocurrida en el terreno de juego.		2. El sistema muestra un formulario con los campos necesarios para adicionar una jugada. 4. El sistema verifica que los datos insertados por el anotador oficial sean válidos. 5. En caso de ser válidos los datos insertados, la aplicación almacena los datos y termina el caso de uso.
Flujos Alternos		
Acción del Actor		Respuesta del Sistema
		4.1 Si los datos son incorrectos el sistema muestra un mensaje de error al anotador oficial y retorna a la acción 2.

Sección 2: “Actualizar Jugadas”

<p>1. El anotador oficial selecciona la opción Actualizar Jugadas.</p> <p>3. El anotador oficial selecciona la jugada que desea modificar.</p> <p>5. El anotador oficial modifica los datos deseados en la jugada seleccionada.</p>	<p>2. El sistema muestra las jugadas insertadas en el juego.</p> <p>4. El sistema muestra una forma con los campos necesarios para modificar la jugada seleccionada.</p> <p>6. El sistema verifica que los datos modificados por el anotador sean válidos.</p> <p>7. En caso de ser válidos los datos actualizados, la aplicación almacena los datos y termina el caso de uso.</p>
---	--

Flujos Alternos

Acción del Actor	Respuesta del Sistema
	<p>6.1. Si los datos son incorrectos el sistema muestra un mensaje de error al usuario y retorna a la acción 4.</p>

Sección 3: “Eliminar Jugadas”

<p>1. El anotador oficial selecciona la opción Eliminar Jugadas.</p> <p>3. El anotador oficial selecciona la jugada que desea eliminar.</p> <p>5. El anotador oficial decide si desea o no eliminar la jugada seleccionada.</p>	<p>2. El sistema muestra las jugadas insertadas en el juego.</p> <p>4. El sistema muestra un mensaje para que el anotador oficial confirme si desea o no eliminar la jugada seleccionada.</p> <p>6. Si el anotador oficial confirma la decisión de eliminar la jugada, el sistema elimina la alineación</p>
---	---

	escogida y termina el caso de uso.
Flujos Alternos	
Acción del Actor	Respuesta del Sistema
5.1 El anotador oficial decide no eliminar la jugada seleccionada, cancela la acción y termina el caso de uso.	
Sección 4: “Mostrar Jugadas”	
1. El anotador oficial selecciona la opción Mostrar Jugadas.	2. El sistema muestra las jugadas pertenecientes al juego y termina el caso de uso.

Tabla No. 20: Descripción del Caso de Uso - Gestionar Jugadas



Figura No. 24: Diagrama de Caso de Uso del Sistema - Gestionar Lanzamientos

Caso de Uso:	Gestionar Lanzamientos
Actores:	Anotador Oficial
Resumen:	El caso de uso inicia cuando el anotador oficial selecciona la opción Gestionar Lanzamientos, escoge una de las opciones que se pueden realizar; Agregar Lanzamientos, Actualizar Lanzamientos, Eliminar Lanzamientos y Mostrar Lanzamientos. El anotador realiza la operación seleccionada, el sistema verifica la validez de los datos,

	en caso de ser correctos los actualiza y termina el caso de uso.
Precondiciones:	El anotador oficial tiene que estar registrado en el sistema. El anotador oficial debe seleccionar el juego en el que se gestionarán los lanzamientos que realiza el pitcher durante el juego.
Referencias	R8
Prioridad	Crítico

Flujo Normal de Eventos

Acción del Actor	Respuesta del Sistema
1. El anotador oficial selecciona la opción Gestionar Lanzamientos.	2. El sistema muestra las opciones a escoger: - Agregar Lanzamientos - Actualizar Lanzamientos - Eliminar Lanzamientos - Mostrar Lanzamientos

Sección 1: “Agregar Lanzamientos”

1. El anotador oficial selecciona la opción Agregar Lanzamientos. 3. El anotador oficial escoge el tipo de lanzamiento realizado por el pitcher. 5. El anotador oficial escoge los datos correspondientes al	2. El sistema muestra una forma con los tipos de lanzamientos posibles realizados por el pitcher. 4. El sistema muestra una serie de datos asociados al lanzamiento, que describe la ocurrencia, la velocidad, la zona y el ángulo de salida. 6. El sistema verifica que los datos insertados por el anotador sean válidos. 7. En caso de ser válidos los datos insertados, la aplicación almacena los datos y termina el caso de
--	--

lanzamiento realizado.	uso.
Flujos Alternos	
Acción del Actor	Respuesta del Sistema
	6.1 Si los datos son incorrectos el sistema muestra un mensaje de error al anotador oficial y retorna a la acción 4.
Sección 2: “Actualizar Lanzamientos”	
1. El anotador oficial selecciona la opción Actualizar Lanzamientos. 3. El anotador oficial selecciona el lanzamiento que desea modificar. 5. El anotador oficial modifica los datos deseados del lanzamiento seleccionado.	2. El sistema muestra los lanzamientos insertados en el juego. 4. El sistema muestra una forma con los campos necesarios para modificar el lanzamiento seleccionado. 6. El sistema verifica que los datos modificados por el anotador sean válidos. 7. En caso de ser válidos los datos actualizados, la aplicación almacena los datos y termina el caso de uso.
Flujos Alternos	
Acción del Actor	Respuesta del Sistema
	6.1. Si los datos son incorrectos el sistema muestra un mensaje de error al usuario y retorna a la acción

	4.
Sección 3: “Eliminar Lanzamientos”	
1. El anotador oficial selecciona la opción Eliminar Lanzamiento.	2. El sistema muestra los lanzamientos insertados en el juego.
3. El anotador oficial selecciona el lanzamiento que desea eliminar.	4. El sistema muestra un mensaje para que el anotador oficial confirme si desea o no eliminar el lanzamiento seleccionado.
5. El anotador oficial decide si desea o no eliminar el lanzamiento seleccionado.	6. Si el anotador oficial confirma la decisión de eliminar el lanzamiento, el sistema lo elimina y termina el caso de uso.
Flujos Alternos	
Acción del Actor	Respuesta del Sistema
5.1 El anotador oficial decide no eliminar el lanzamiento seleccionado, cancela la acción y termina el caso de uso.	
Sección 4: “Mostrar Lanzamientos”	
1. El anotador oficial selecciona la opción Mostrar Lanzamientos.	2. El sistema muestra los pitcher que trabajaron durante el juego.
3. El anotador oficial selecciona el pitcher del cual desea mostrar sus lanzamientos.	4. El sistema muestra los lanzamientos realizados por el pitcher seleccionado y termina el caso de uso.

Tabla No. 21: Descripción del Caso de Uso - Gestionar Lanzamientos

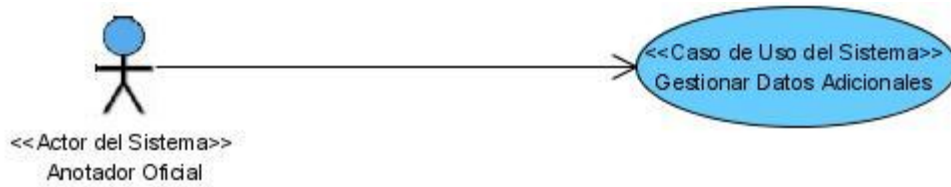


Figura No. 25: Diagrama de Caso de Uso del Sistema - Gestionar Datos Adicionales

Caso de Uso:	Gestionar Datos Adicionales
Actores:	Anotador Oficial
Resumen:	El caso de uso inicia cuando el anotador oficial selecciona la opción Gestionar Datos adicionales, escoge una de las opciones que se pueden realizar; Agregar Datos Adicionales, Actualizar Datos Adicionales, Eliminar Datos Adicionales y Mostrar Datos Adicionales. El anotador realiza la operación seleccionada, el sistema verifica la validez de los datos, en caso de ser correctos los actualiza y termina el caso de uso.
Precondiciones:	El anotador oficial tiene que estar registrado en el sistema. El anotador oficial debe seleccionar el juego en el que se gestionarán los datos.
Referencias	R9
Prioridad	Crítico

Flujo Normal de Eventos

Acción del Actor	Respuesta del Sistema
1. El anotador oficial selecciona la opción Gestionar Datos Adicionales.	2. El sistema muestra las opciones a escoger: - Agregar Datos Adicionales - Actualizar Datos Adicionales - Eliminar Datos Adicionales

	- Mostrar Datos Adicionales
Sección 1: “Agregar Datos Adicionales”	
<p>1. El anotador oficial selecciona la opción Agregar Datos Adicionales.</p> <p>3. El anotador oficial inserta los datos requeridos por el sistema.</p>	<p>2. El sistema muestra un formulario con los datos a insertar.</p> <p>4. El sistema verifica que los datos insertados por el anotador sean válidos.</p> <p>5. En caso de ser válidos los datos insertados, la aplicación almacena los datos y termina el caso de uso.</p>
Flujos Alternos	
Acción del Actor	Respuesta del Sistema
	4.1 Si los datos son incorrectos el sistema muestra un mensaje de error al anotador oficial y retorna a la acción 2.
Sección 2: “Actualizar Datos Adicionales”	
<p>1. El anotador oficial selecciona la opción Actualizar Datos Adicionales.</p> <p>3. El anotador oficial selecciona los datos que desea modificar.</p> <p>5. El anotador oficial modifica los datos deseados.</p>	<p>2. El sistema muestra los datos insertados en el juego.</p> <p>4. El sistema muestra una forma con los campos necesarios para modificar los datos deseados.</p> <p>6. El sistema verifica que los datos modificados por el anotador sean válidos.</p> <p>7. En caso de ser válidos los datos actualizados, la</p>

	aplicación almacena los datos y termina el caso de uso.
Flujos Alternos	
Acción del Actor	Respuesta del Sistema
	6.1. Si los datos son incorrectos el sistema muestra un mensaje de error al usuario y retorna a la acción 4.
Sección 3: “Eliminar Datos Adicionales”	
1. El anotador oficial selecciona la opción Eliminar Datos Adicionales. 3. El anotador oficial selecciona los datos que desea eliminar. 5. El anotador oficial decide si desea o no eliminar el dato seleccionado.	2. El sistema muestra los datos insertados en el juego. 4. El sistema muestra un mensaje para que el anotador oficial confirme si desea o no eliminar el dato seleccionado. 6. Si el anotador oficial confirma la decisión de eliminar el dato, el sistema lo elimina y termina el caso de uso.
Flujos Alternos	
Acción del Actor	Respuesta del Sistema
5.1 El anotador oficial decide no eliminar el dato seleccionado, cancela la acción y termina el caso de uso.	

Sección 4: “Mostrar Datos Adicionales”

- | | |
|--|--|
| 1. El anotador oficial selecciona la opción Mostrar Datos Adicionales. | 2. El sistema muestra los datos insertados en el juego y termina el caso de uso. |
|--|--|

Tabla No. 22: Descripción del Caso de Uso - Gestionar Datos Adicionales

Anexo 3: Diagramas de clases del análisis

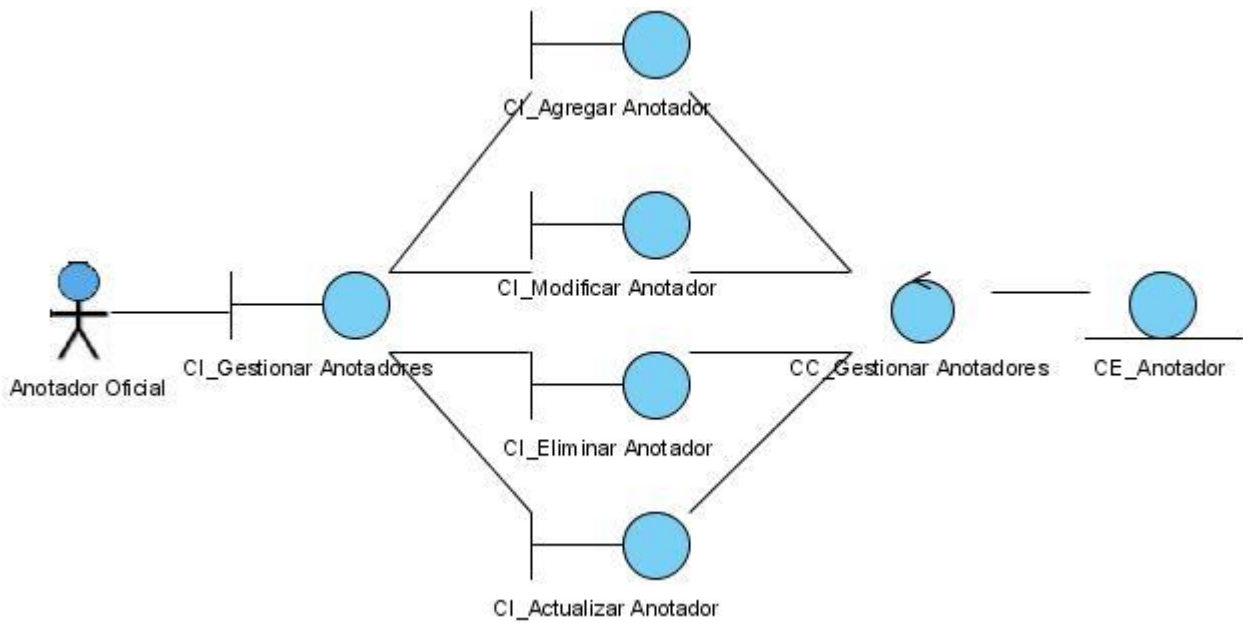


Figura No. 26: Diagrama de Clases del Análisis - CU Gestionar Anotadores

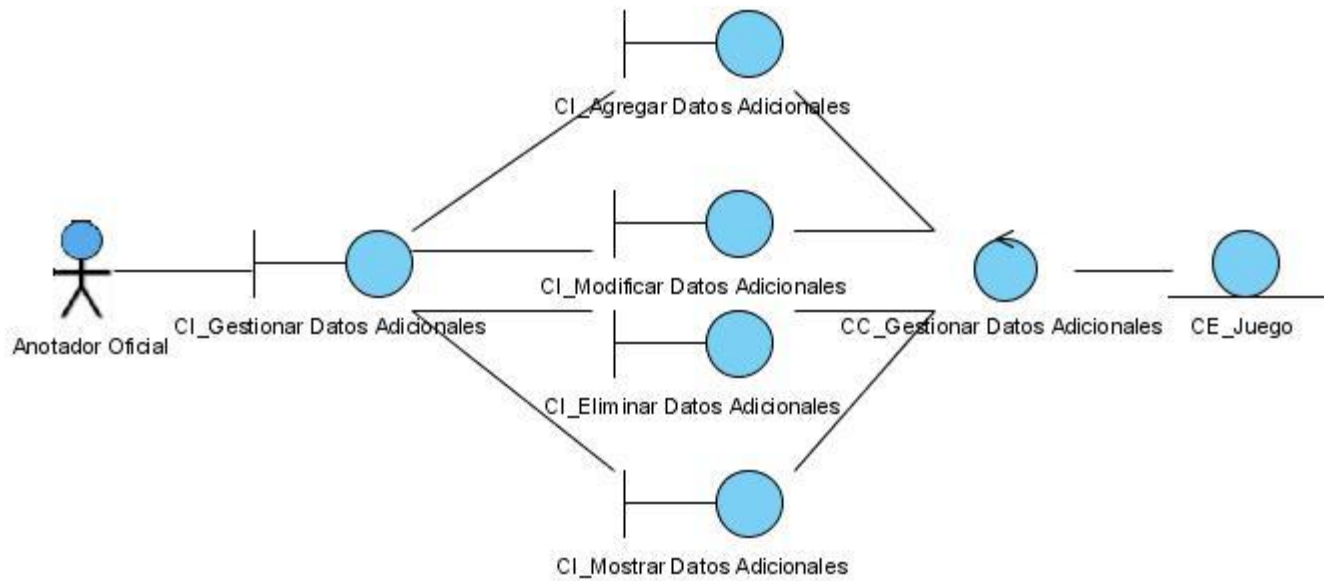


Figura No. 27: Diagrama de Clases del Análisis - CU Gestionar Datos Adicionales

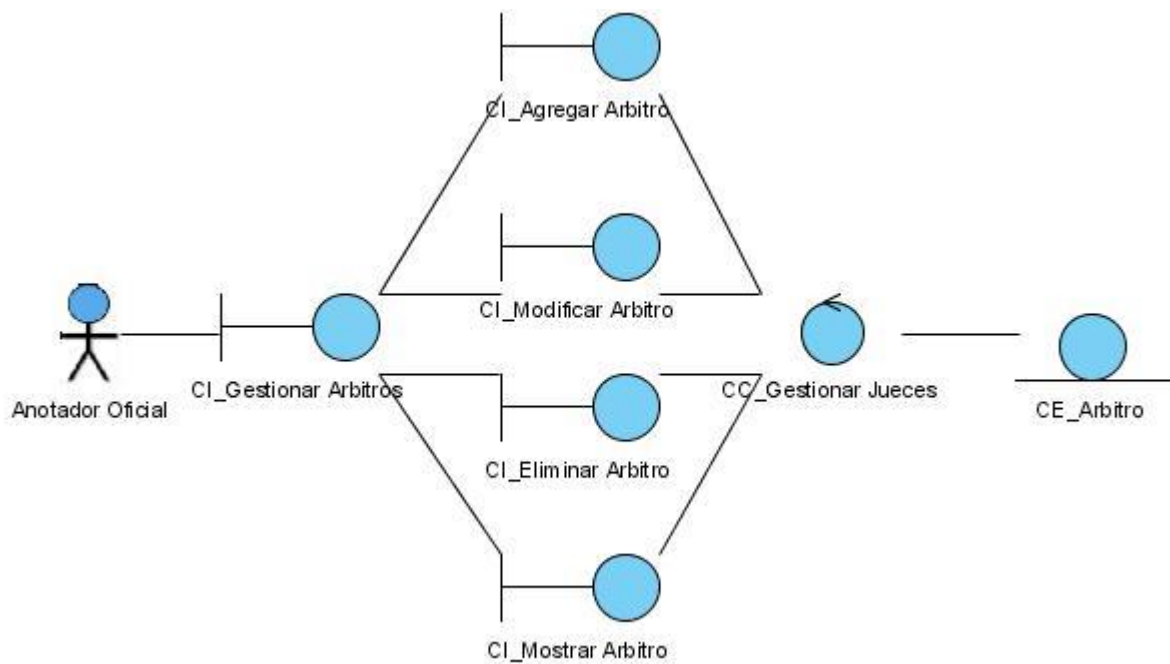


Figura No. 28: Diagrama de Clases del Análisis - CU Gestionar Árbitros

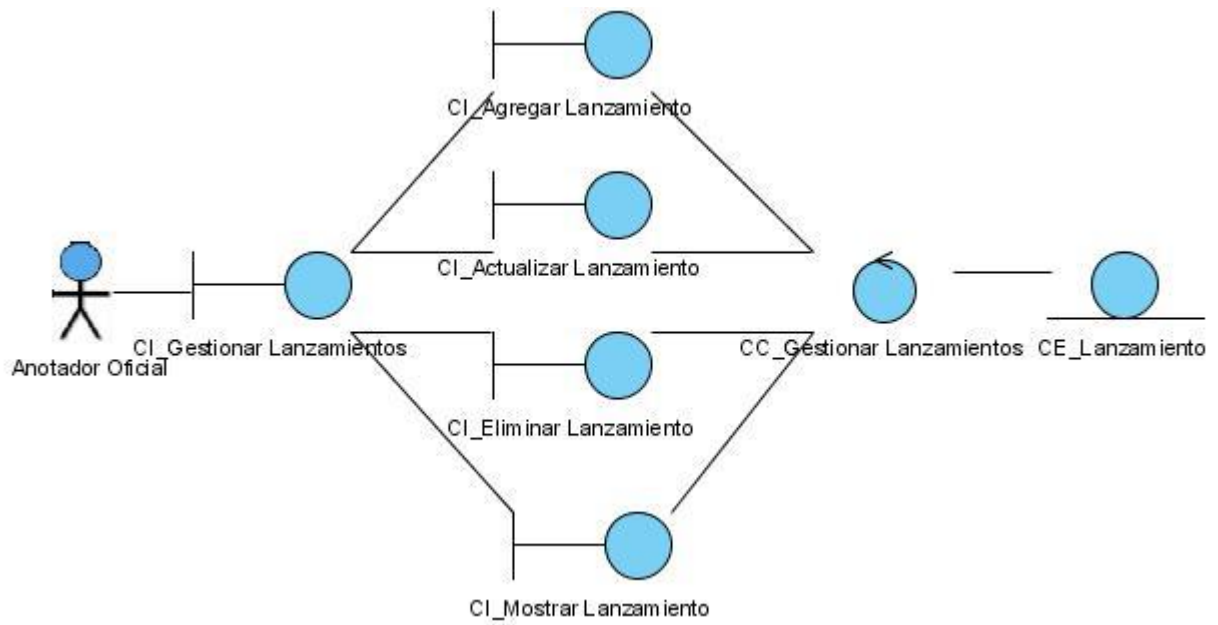


Figura No. 29: Diagrama de Clases del Análisis - CU Gestionar Lanzamientos

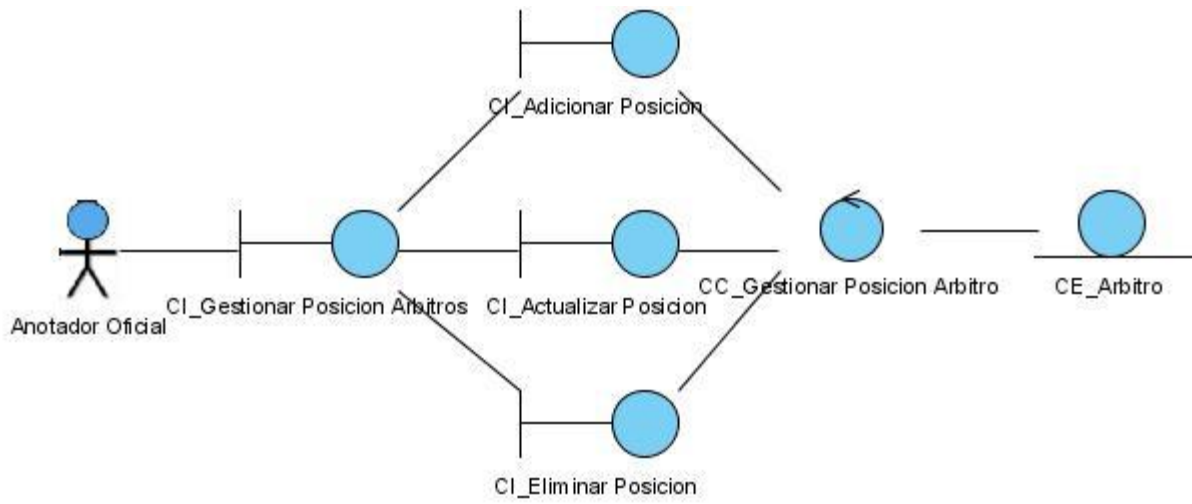


Figura No. 30: Diagrama de Clases del Análisis - CU Gestionar Posición de los Árbitros

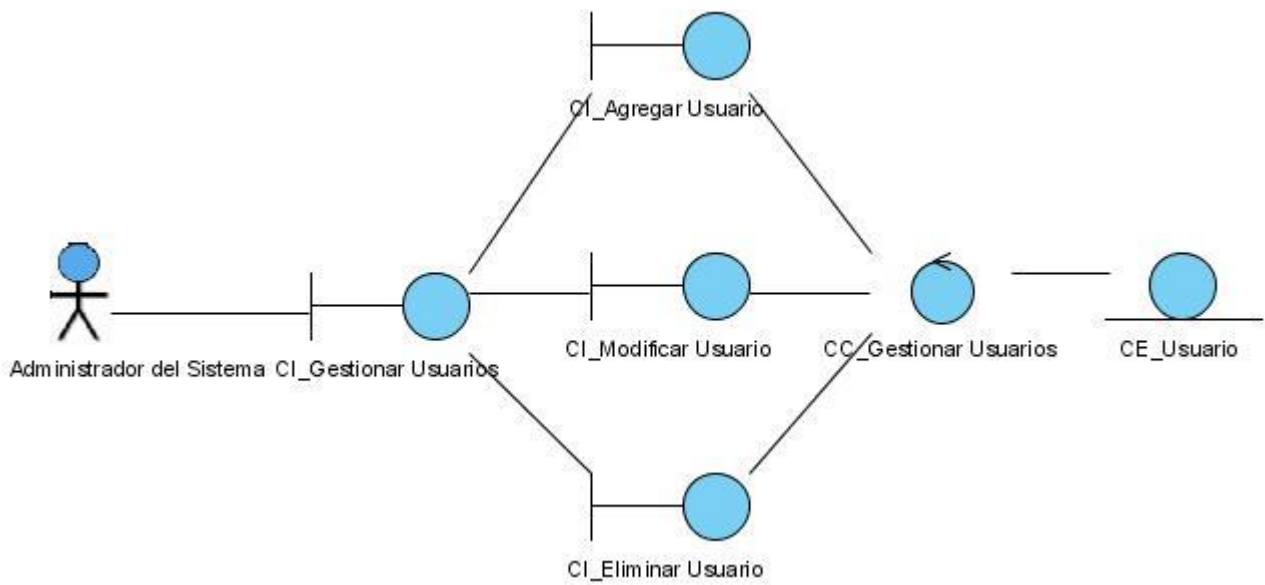


Figura No. 31: Diagrama de Clases del Análisis - CU Gestionar Usuarios

Anexo 4: Diagramas de Colaboración del Análisis

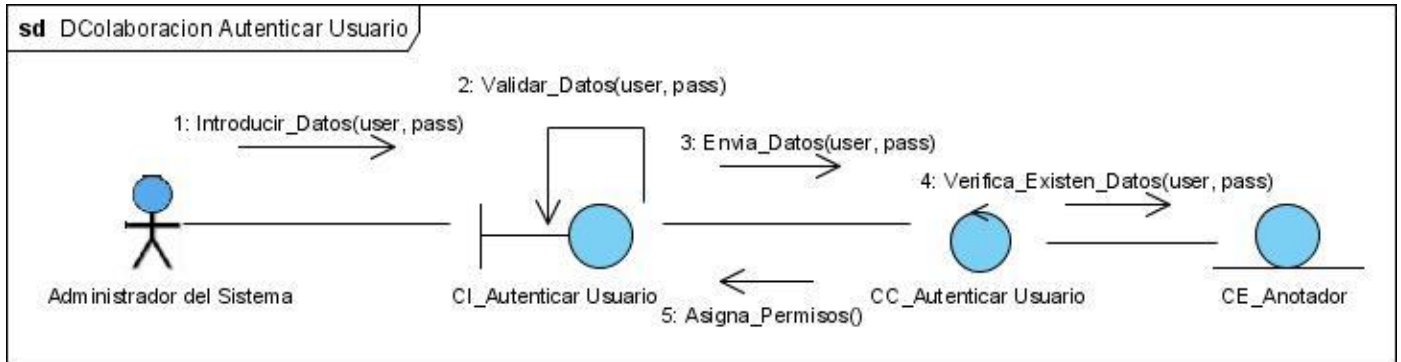


Figura No. 32: Diagrama de Colaboración - CU Autenticar Usuario

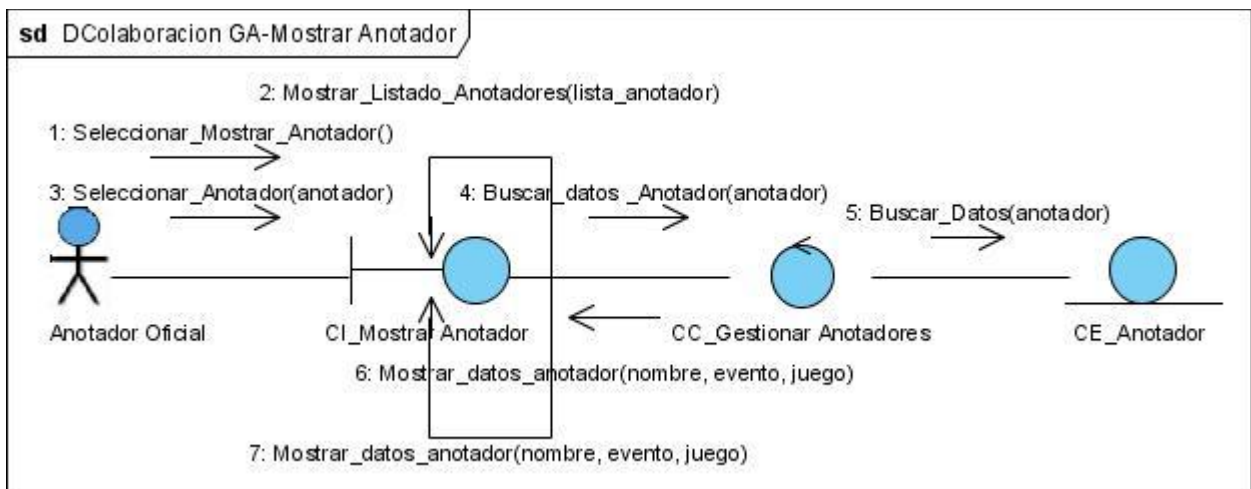


Figura No. 33: Diagrama de Colaboración - CU Gestionar Anotador (Mostrar Anotador)

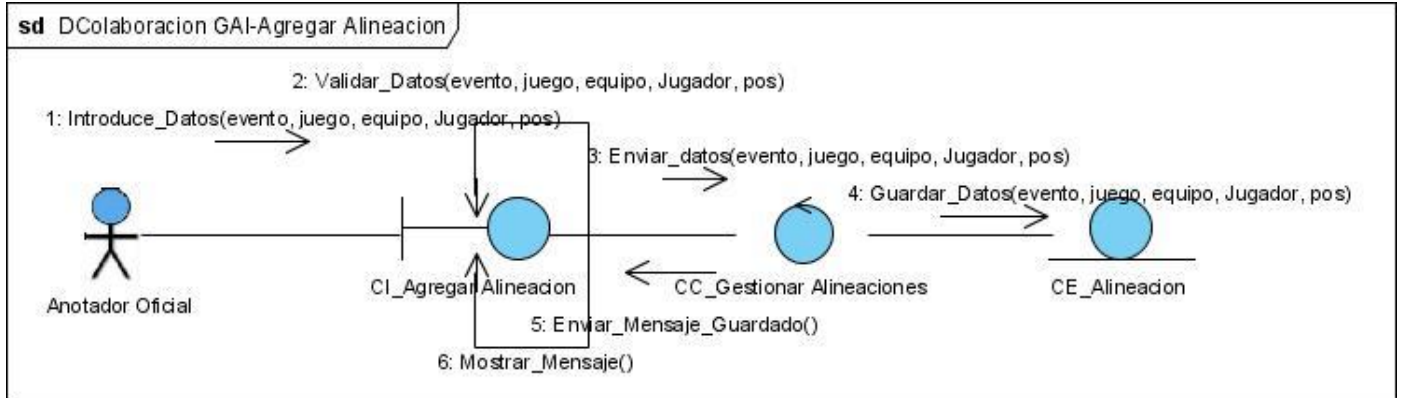


Figura No. 34: Diagrama de Colaboración - CU Gestionar Alineación (Agregar Alineación)

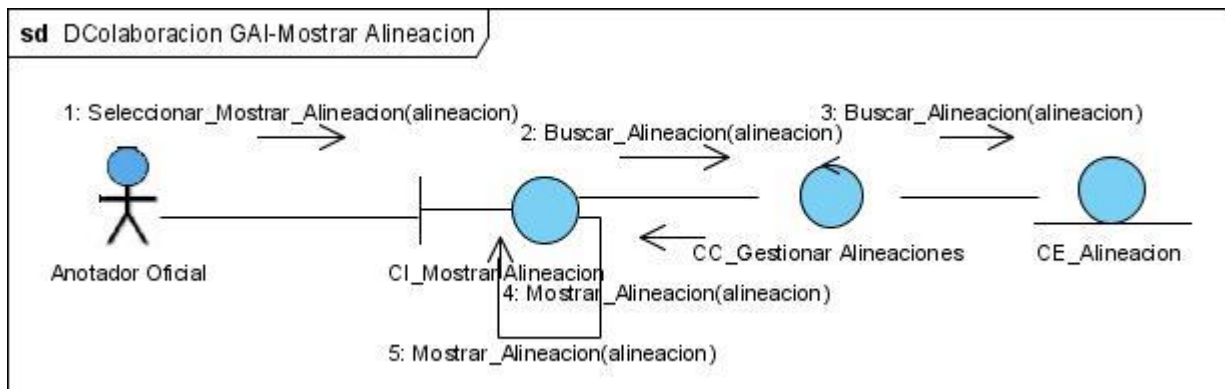


Figura No. 35: Diagrama de Colaboración - CU Gestionar Alineación (Mostrar Alineación)

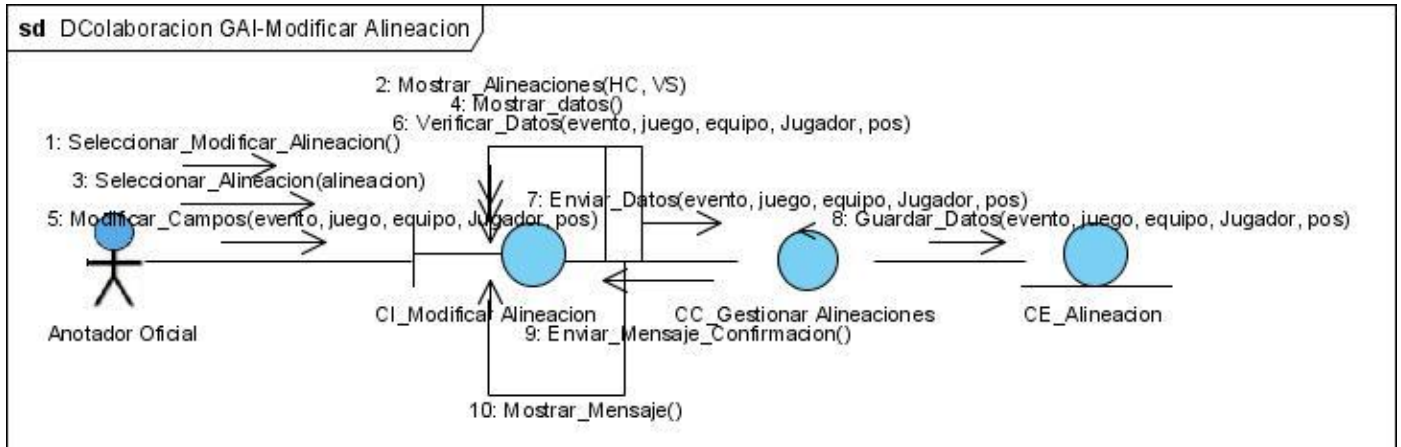


Figura No. 36: Diagrama de Colaboración - CU Gestionar Alineación (Modificar Alineación)

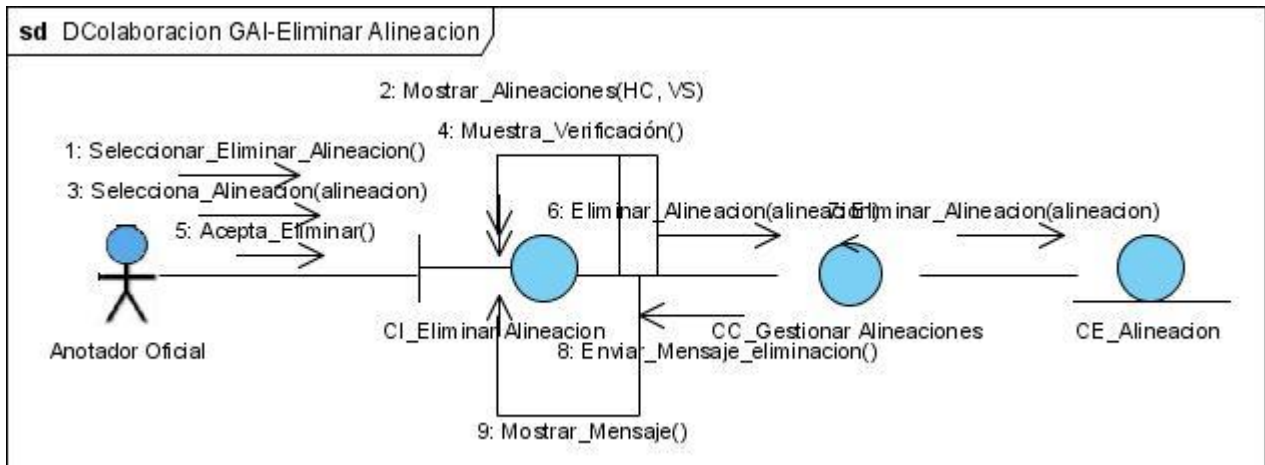


Figura No. 37: Diagrama de Colaboración - CU Gestionar Alineación (Eliminar Alineación)

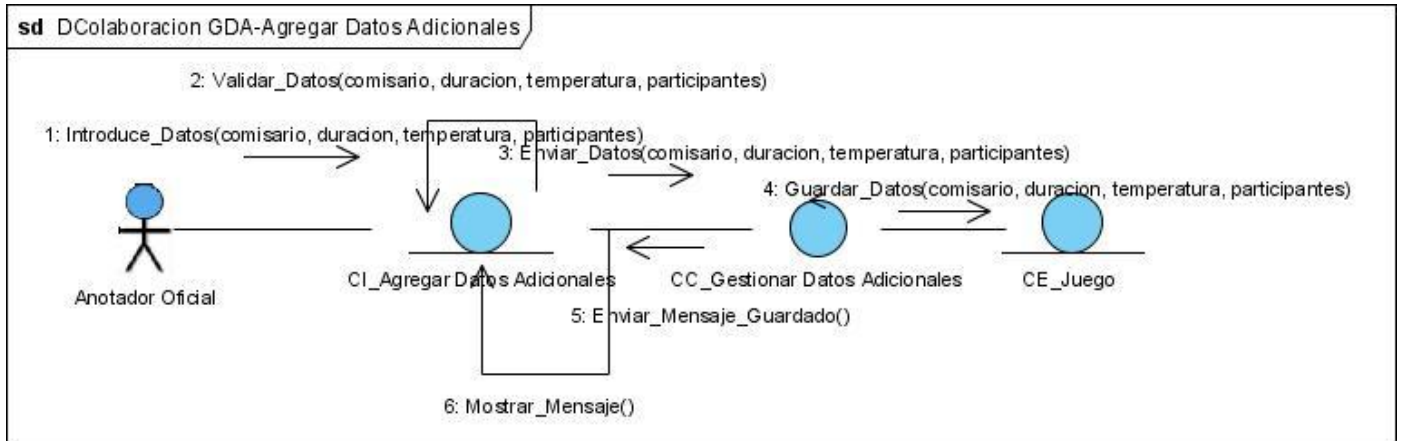


Figura No. 38: Diagrama de Colaboración - CU Gestionar Datos Adicionales (Agregar Datos Adicionales)

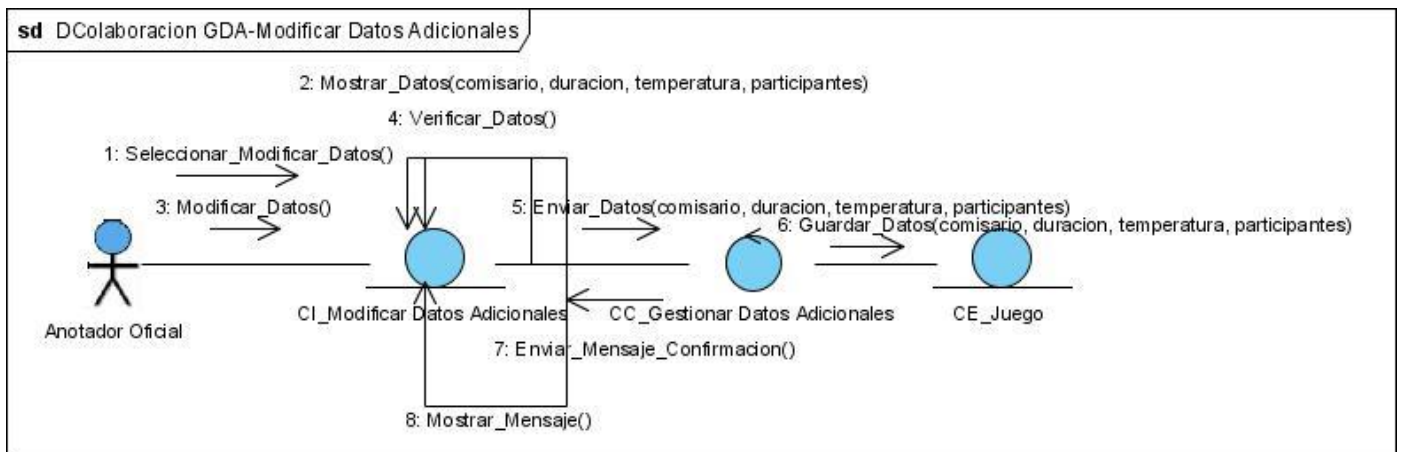


Figura No. 39: Diagrama de Colaboración - CU Gestionar Datos Adicionales (Modificar Datos Adicionales)

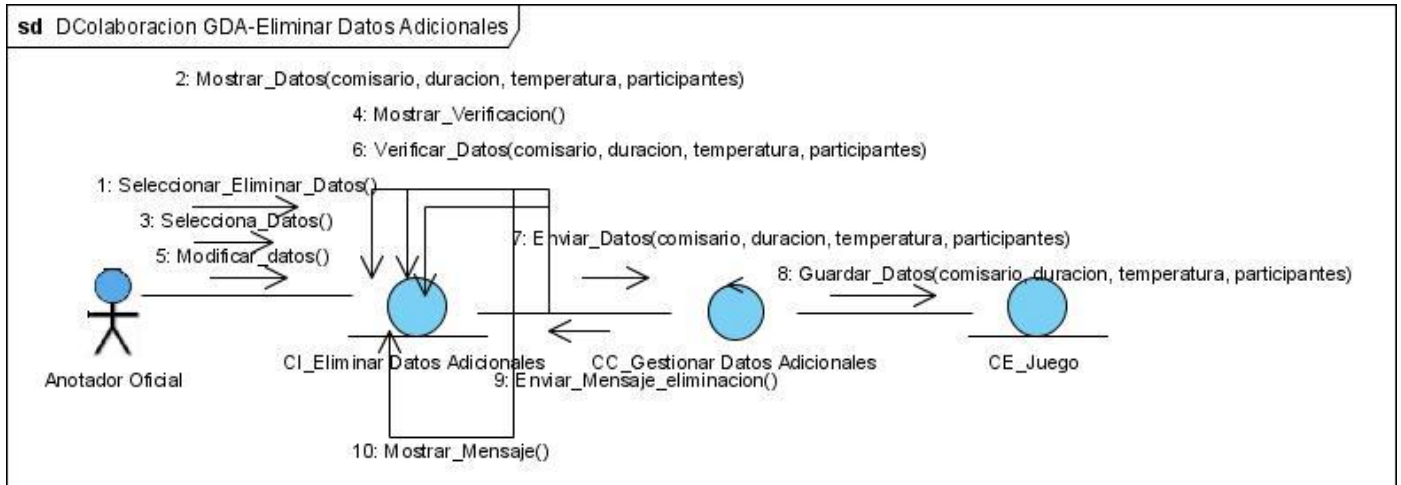


Figura No. 40: Diagrama de Colaboración - CU Gestionar Datos Adicionales (Eliminar Datos Adicionales)

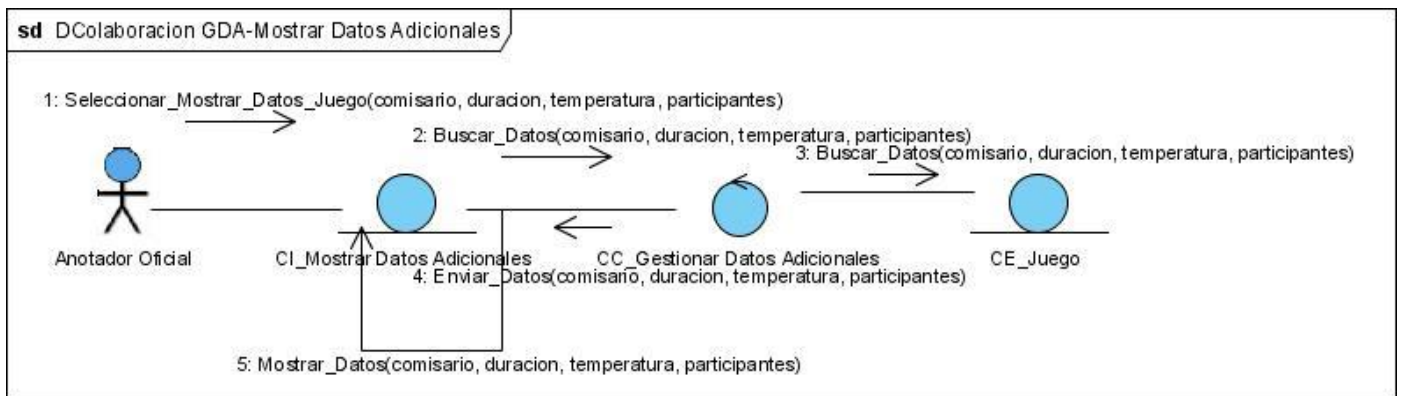


Figura No. 41: Diagrama de Colaboración - CU Gestionar Datos Adicionales (Mostrar Datos Adicionales)

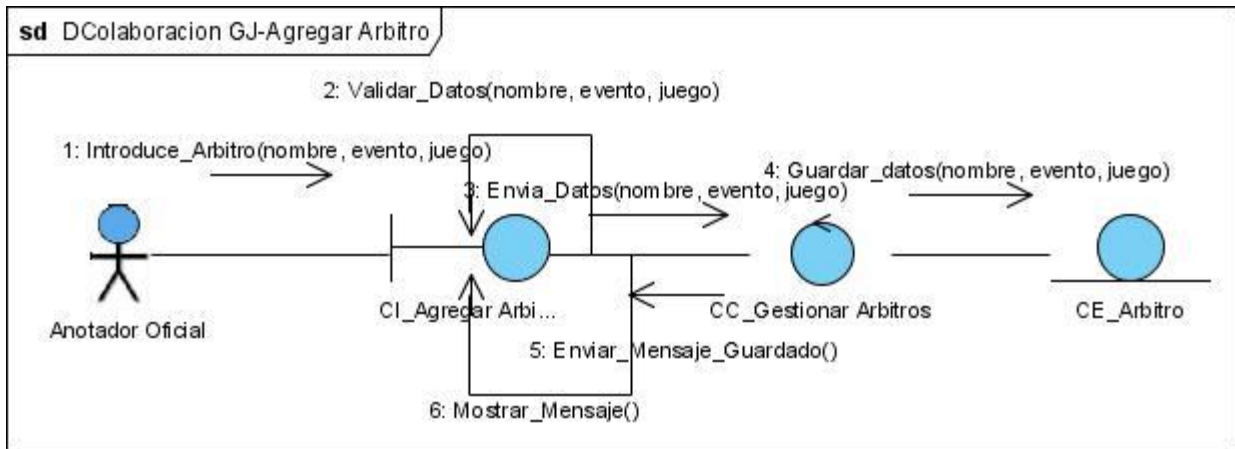


Figura No. 42: Diagrama de Colaboración - CU Gestionar Árbitro (Agregar Árbitro)

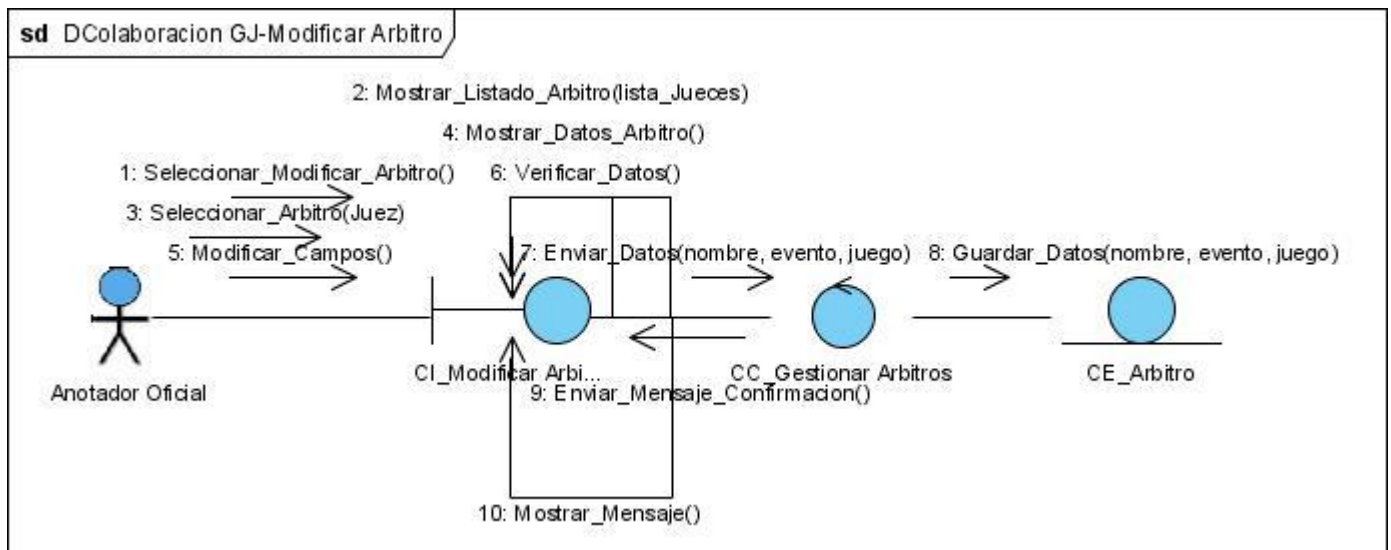


Figura No. 43: Diagrama de Colaboración - CU Gestionar Árbitro (Modificar Árbitro)

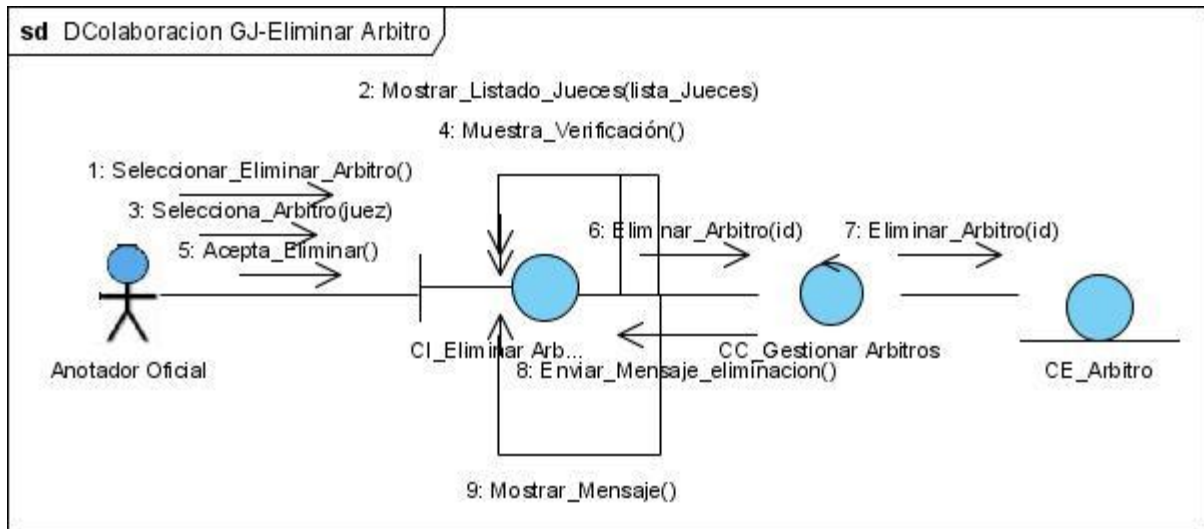


Figura No. 44: Diagrama de Colaboración - CU Gestionar Arbitro (Eliminar Arbitro)

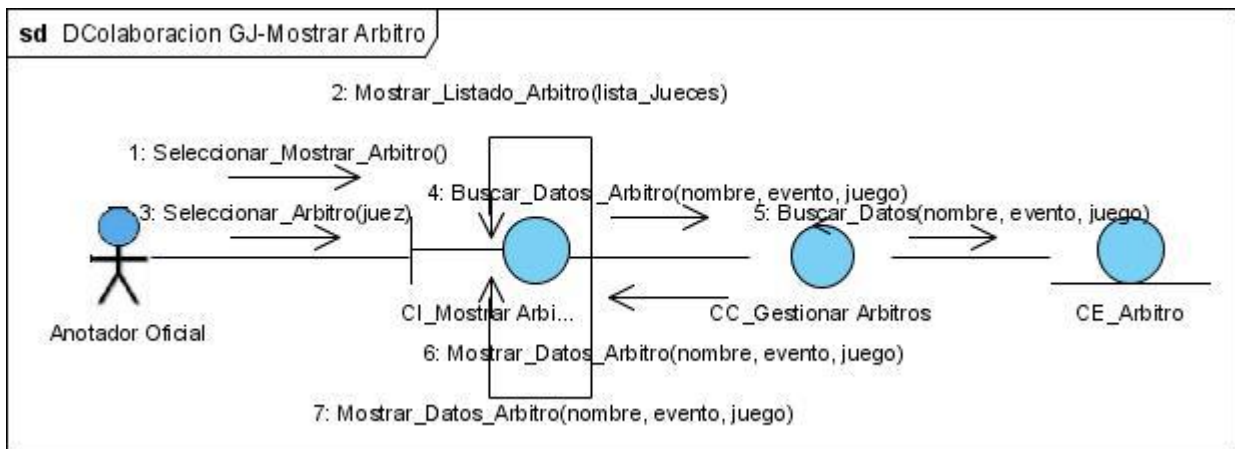


Figura No. 45: Diagrama de Colaboración - CU Gestionar Arbitro (Mostrar Arbitro)

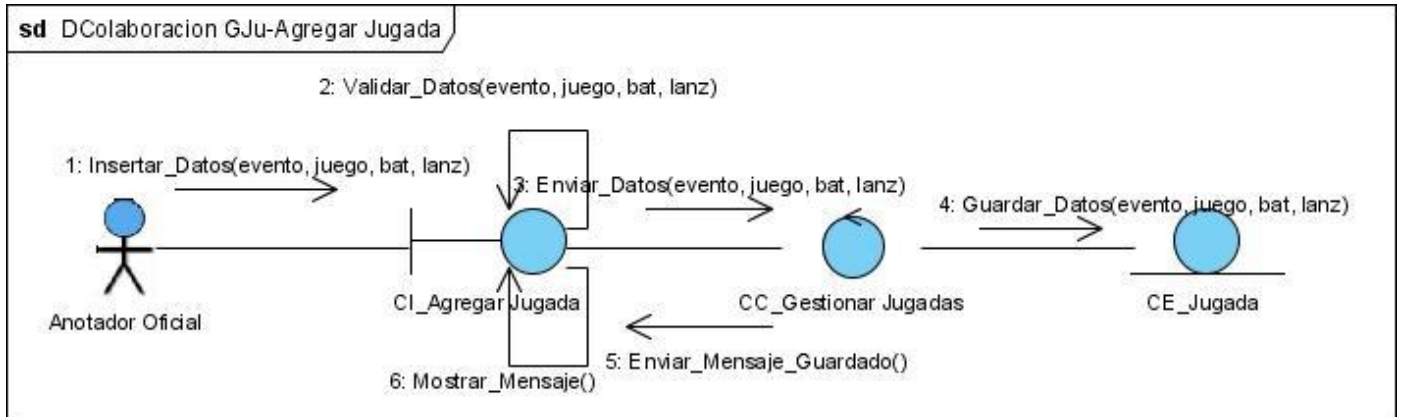


Figura No. 46: Diagrama de Colaboración - CU Gestionar Jugadas (Agregar Jugada)

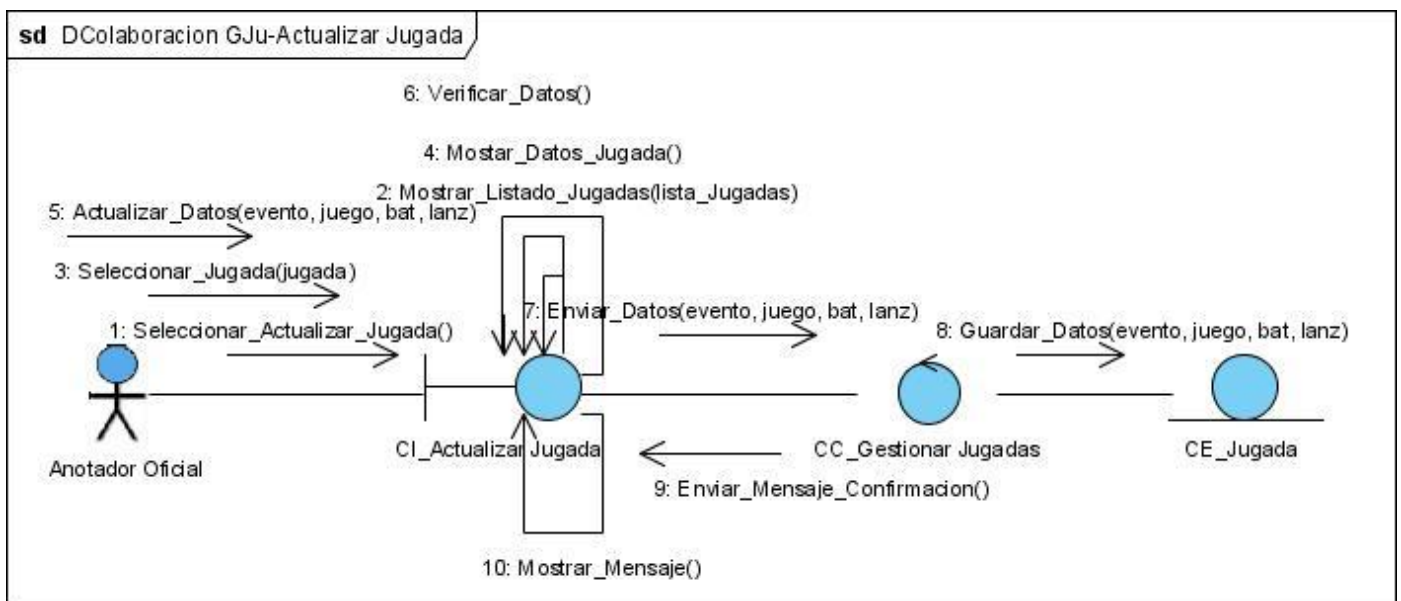


Figura No. 47: Diagrama de Colaboración - CU Gestionar Jugadas (Actualizar Jugada)

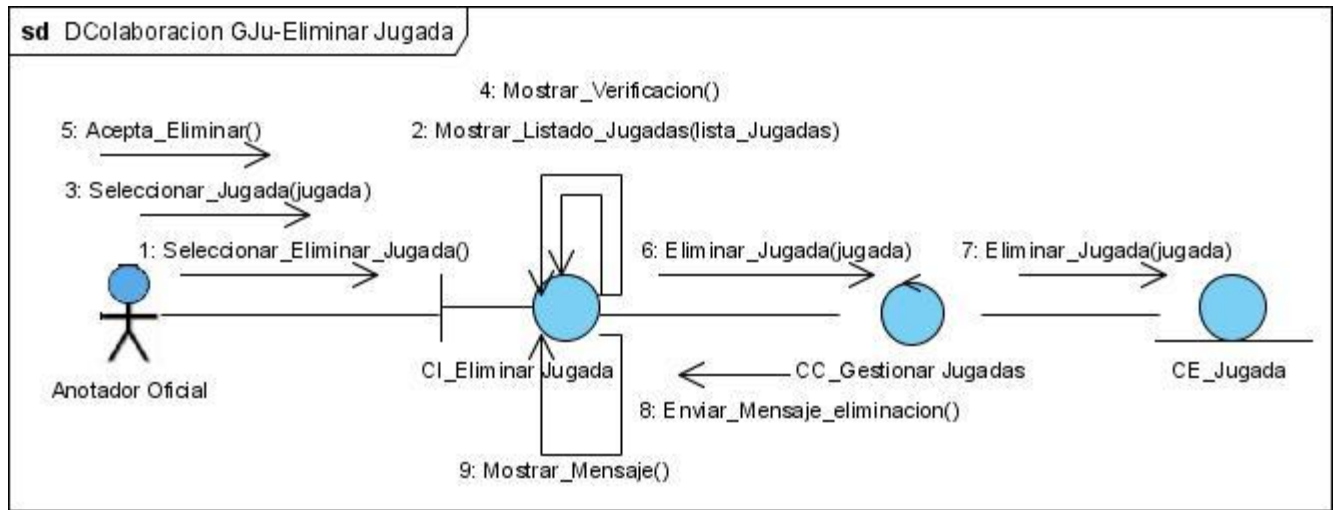


Figura No. 48: Diagrama de Colaboración - CU Gestionar Jugadas (Eliminar Jugada)

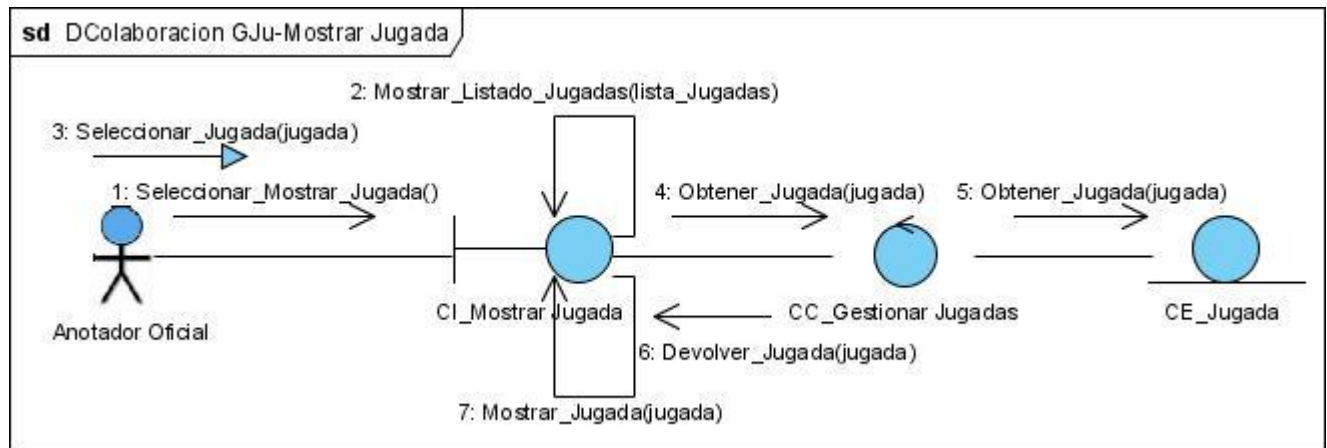


Figura No. 49: Diagrama de Colaboración - CU Gestionar Jugadas (Mostrar Jugada)

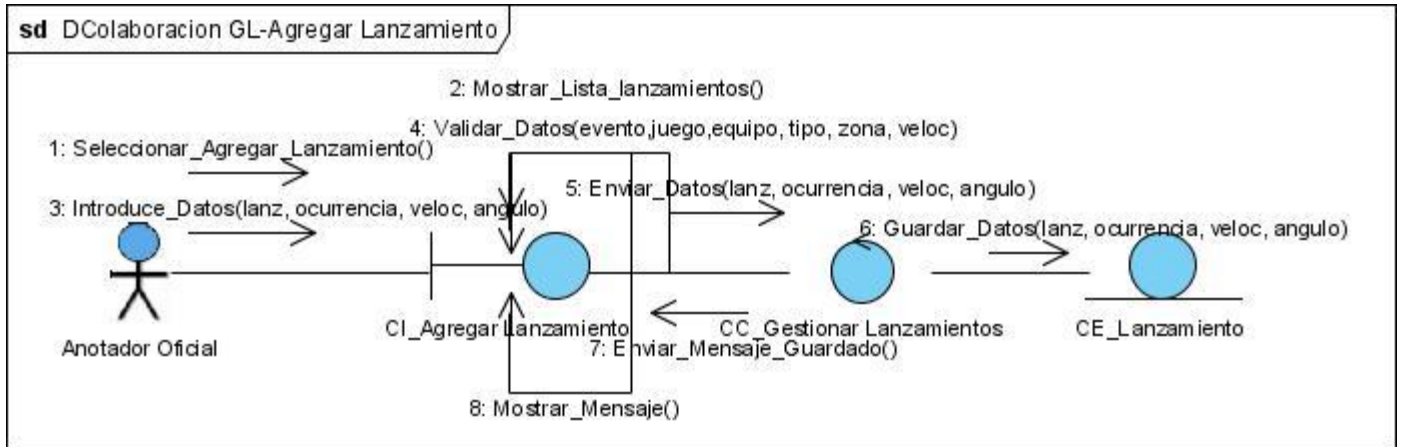


Figura No. 50: Diagrama de Colaboración - CU Gestionar Lanzamientos (Agregar Lanzamiento)

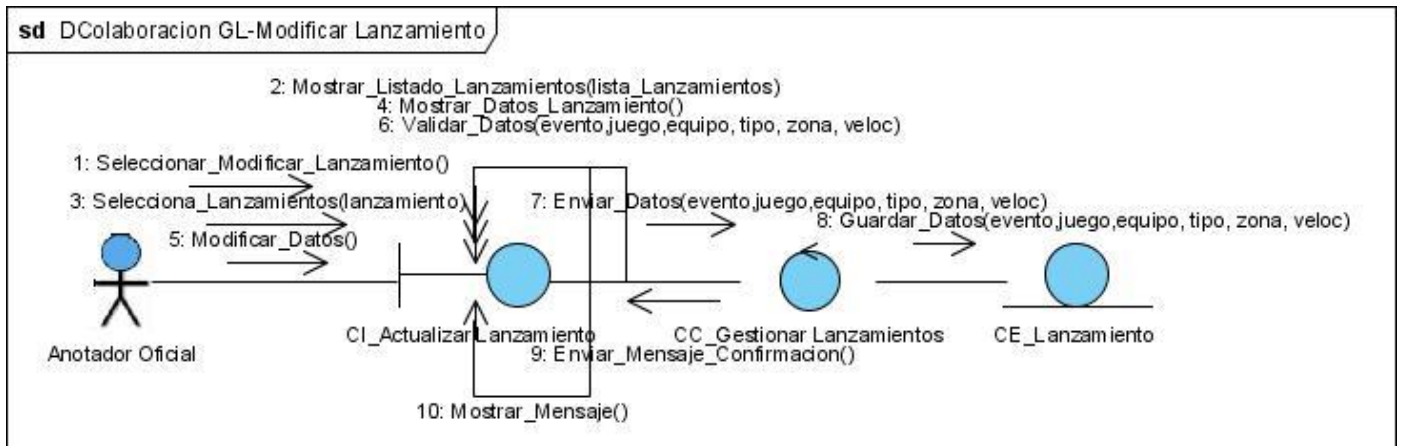


Figura No. 51: Diagrama de Colaboración - CU Gestionar Lanzamientos (Modificar Lanzamiento)

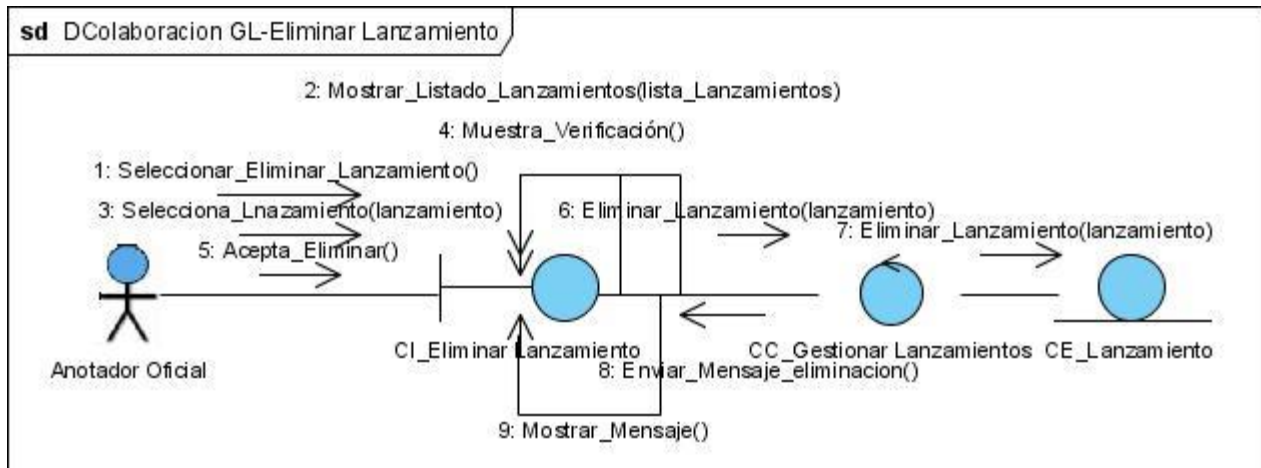


Figura No. 52: Diagrama de Colaboración - CU Gestionar Lanzamientos (Eliminar Lanzamiento)

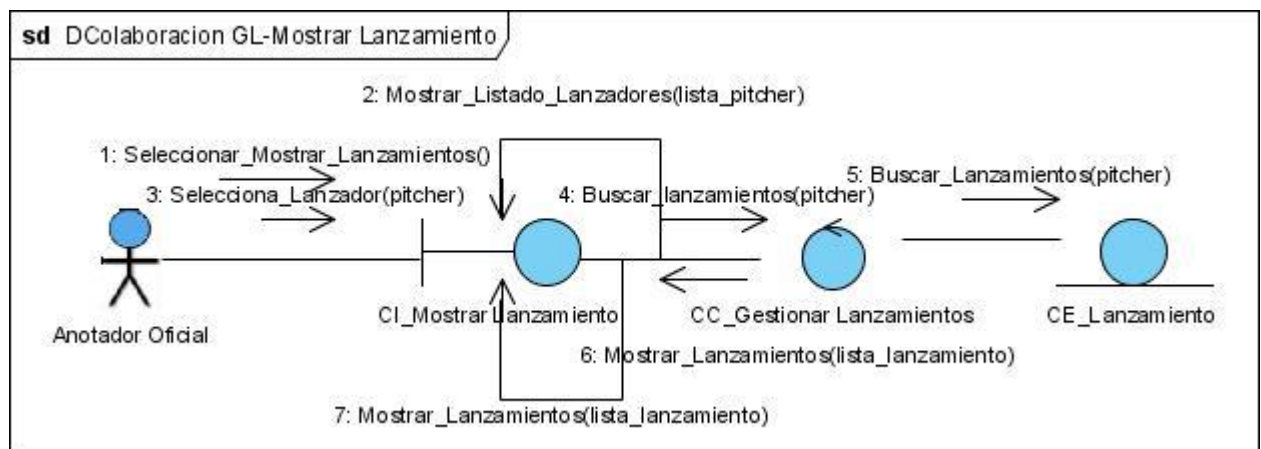


Figura No. 53: Diagrama de Colaboración - CU Gestionar Lanzamientos (Mostrar Lanzamiento)

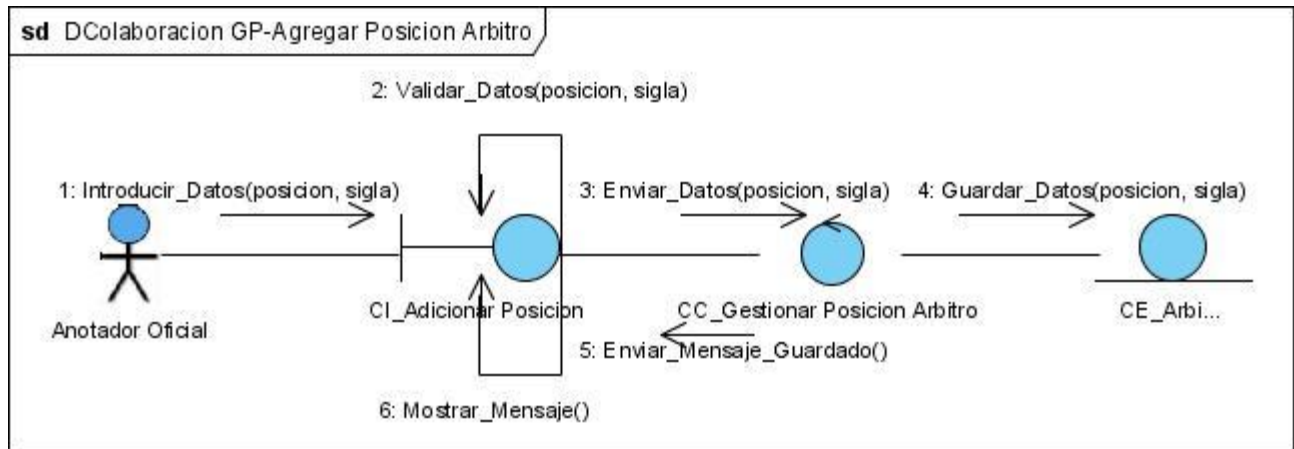


Figura No. 54: Diagrama de Colaboración - CU Gestionar Posición Árbitro (Agregar Posición Árbitro)

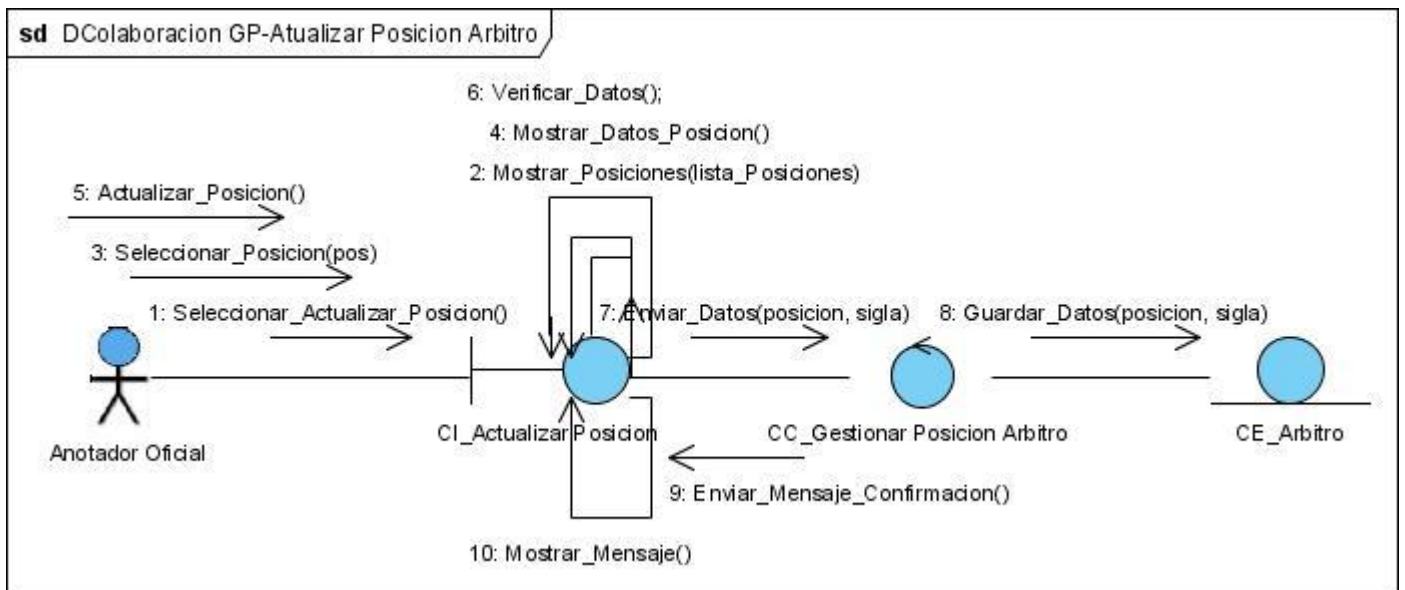


Figura No. 55: Diagrama de Colaboración - CU Gestionar Posición Árbitro (Actualizar Posición Árbitro)

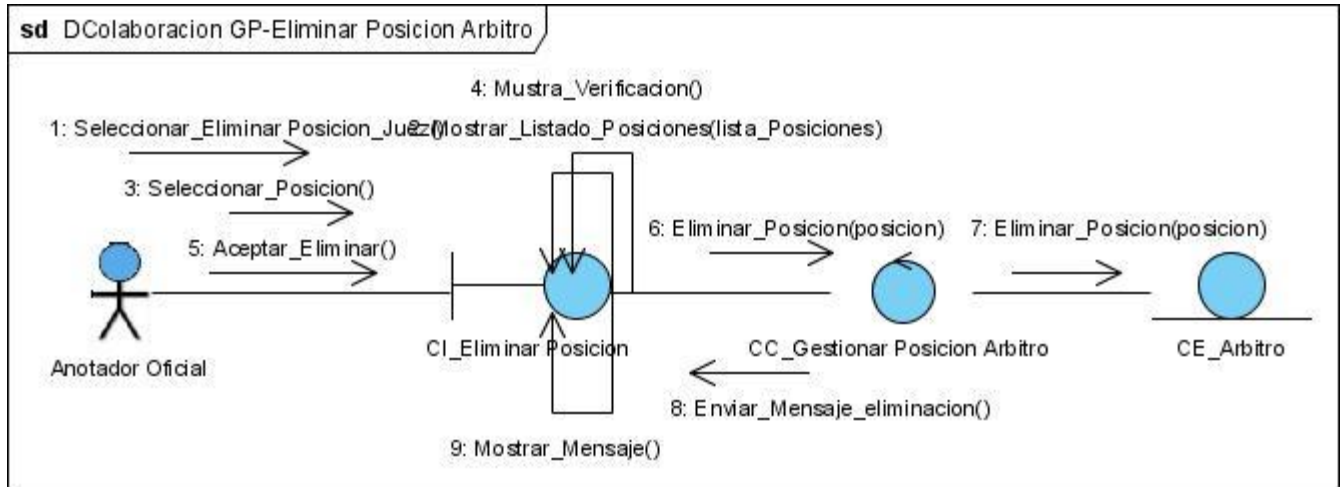


Figura No. 56: Diagrama de Colaboración - CU Gestionar Posición Árbitro (Eliminar Posición Árbitro)

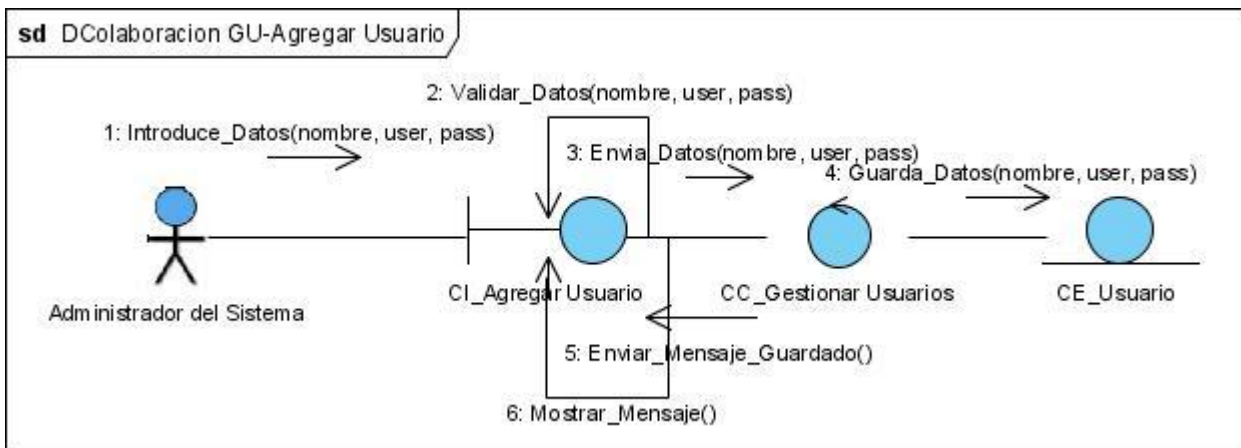


Figura No. 57: Diagrama de Colaboración - CU Gestionar Usuarios (Agregar Usuario)

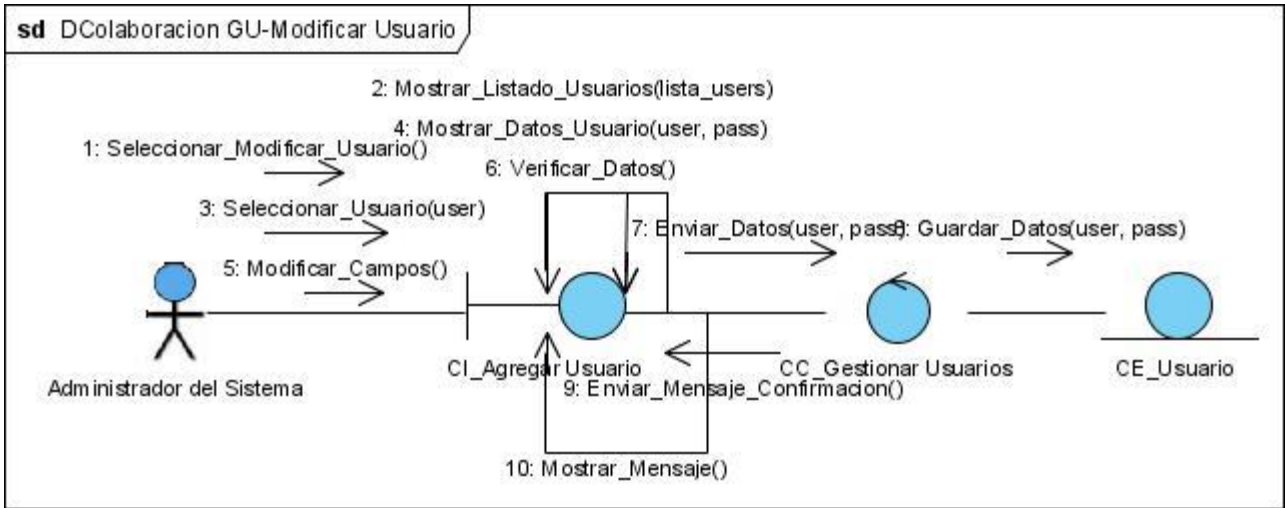


Figura No. 58: Diagrama de Colaboración - CU Gestionar Usuarios (Modificar Usuario)

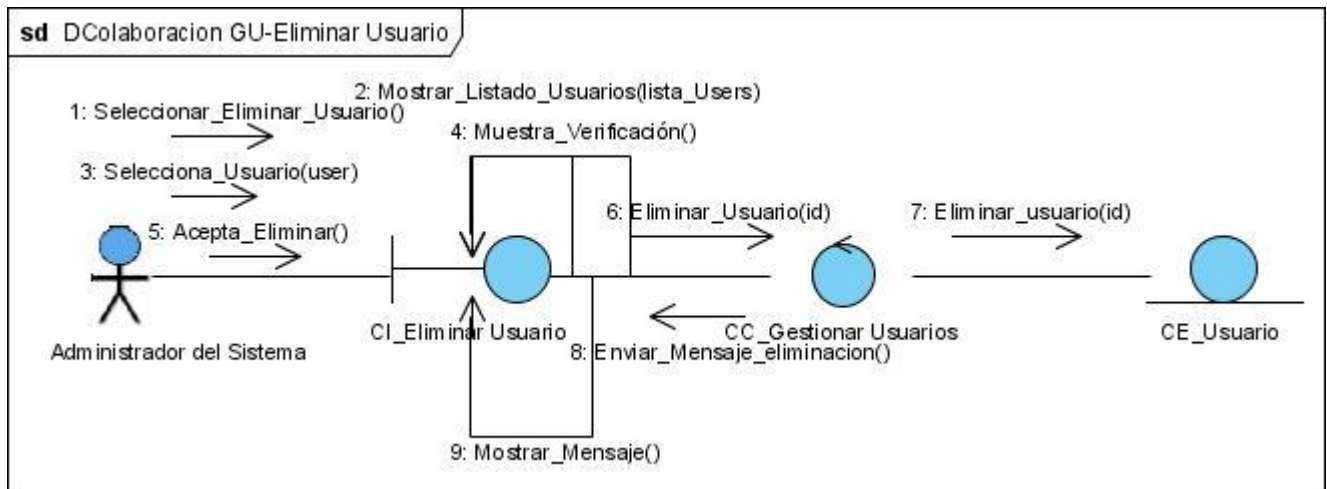


Figura No. 59: Diagrama de Colaboración - CU Gestionar Usuarios (Eliminar Usuario)

Anexo 5: Diagrama de Clases de Diseño

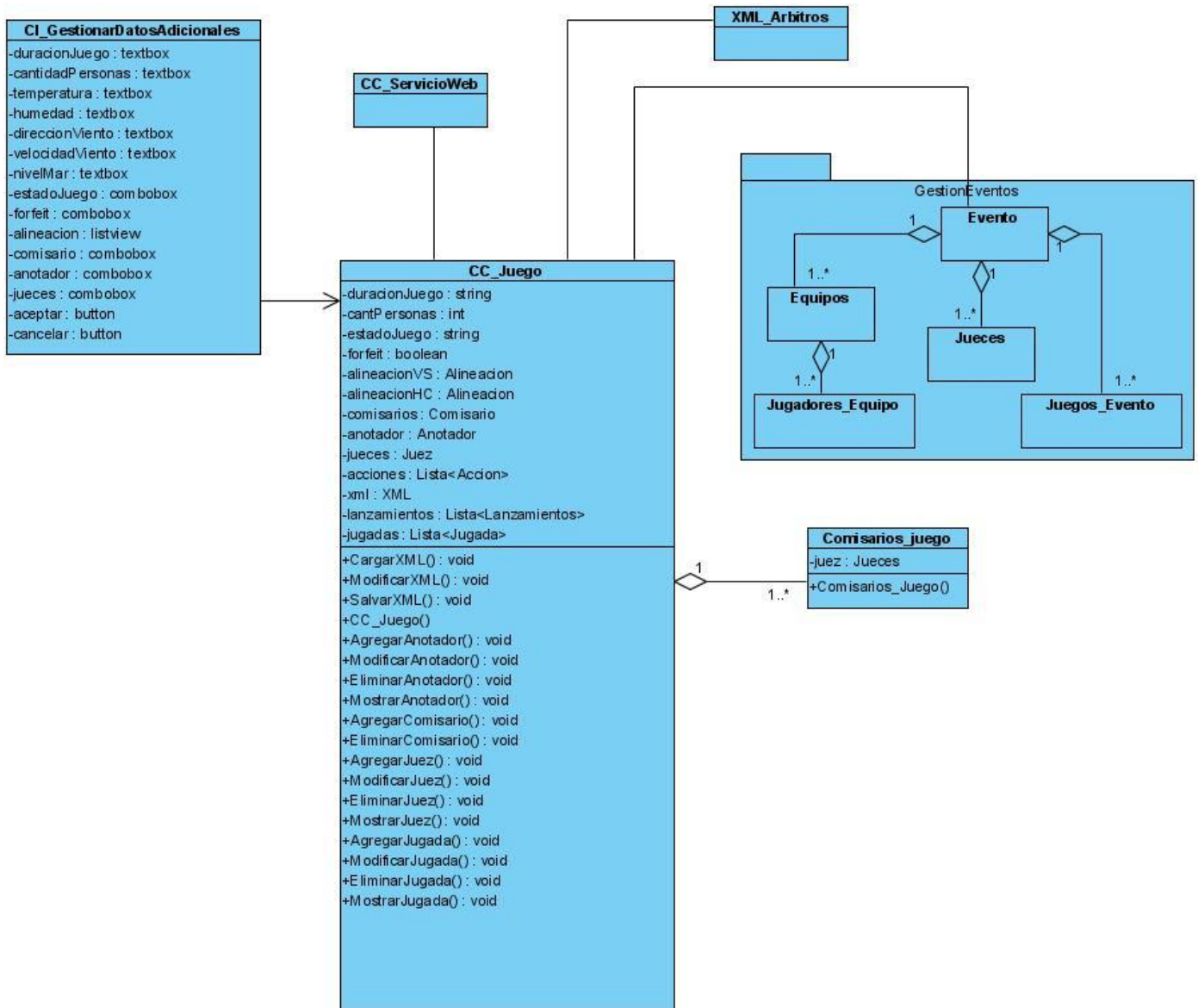


Figura No. 60: Diagrama de Clases de Diseño - CU Gestionar Datos Adicionales

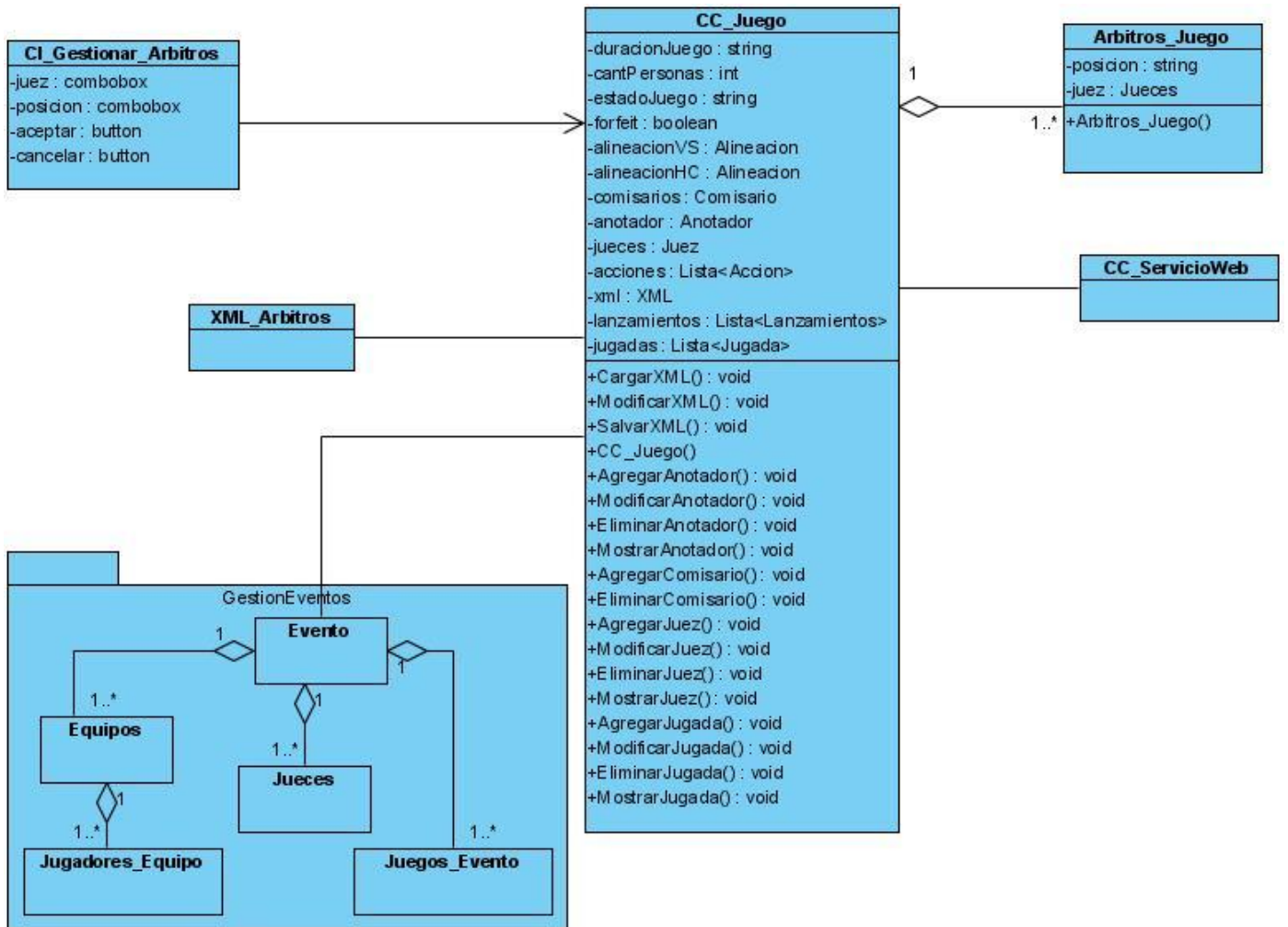


Figura No. 61: Diagrama de Clases de Diseño - CU Gestionar Árbitros

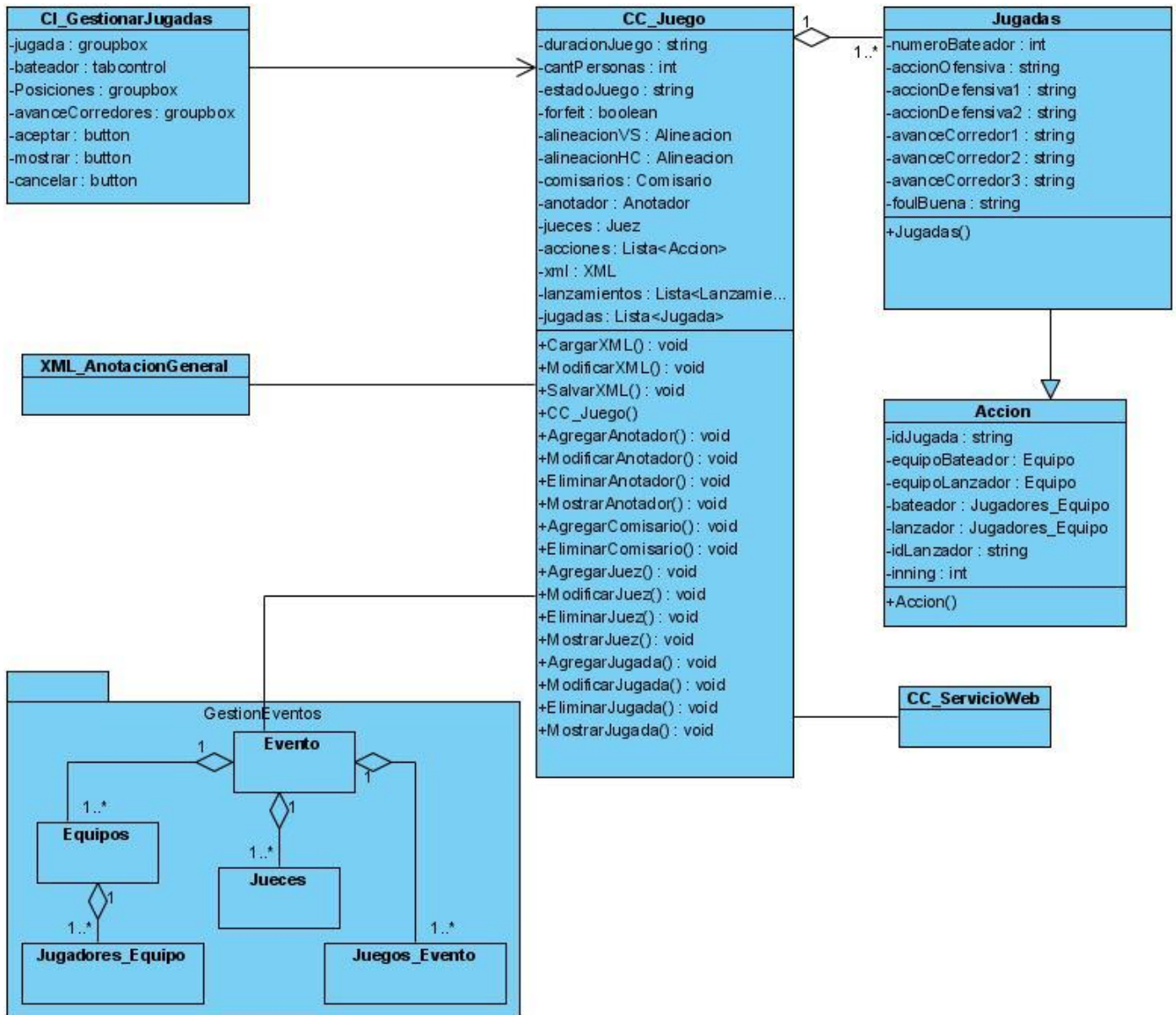


Figura No. 62: Diagrama de Clases de Diseño - CU Gestionar Jugadas

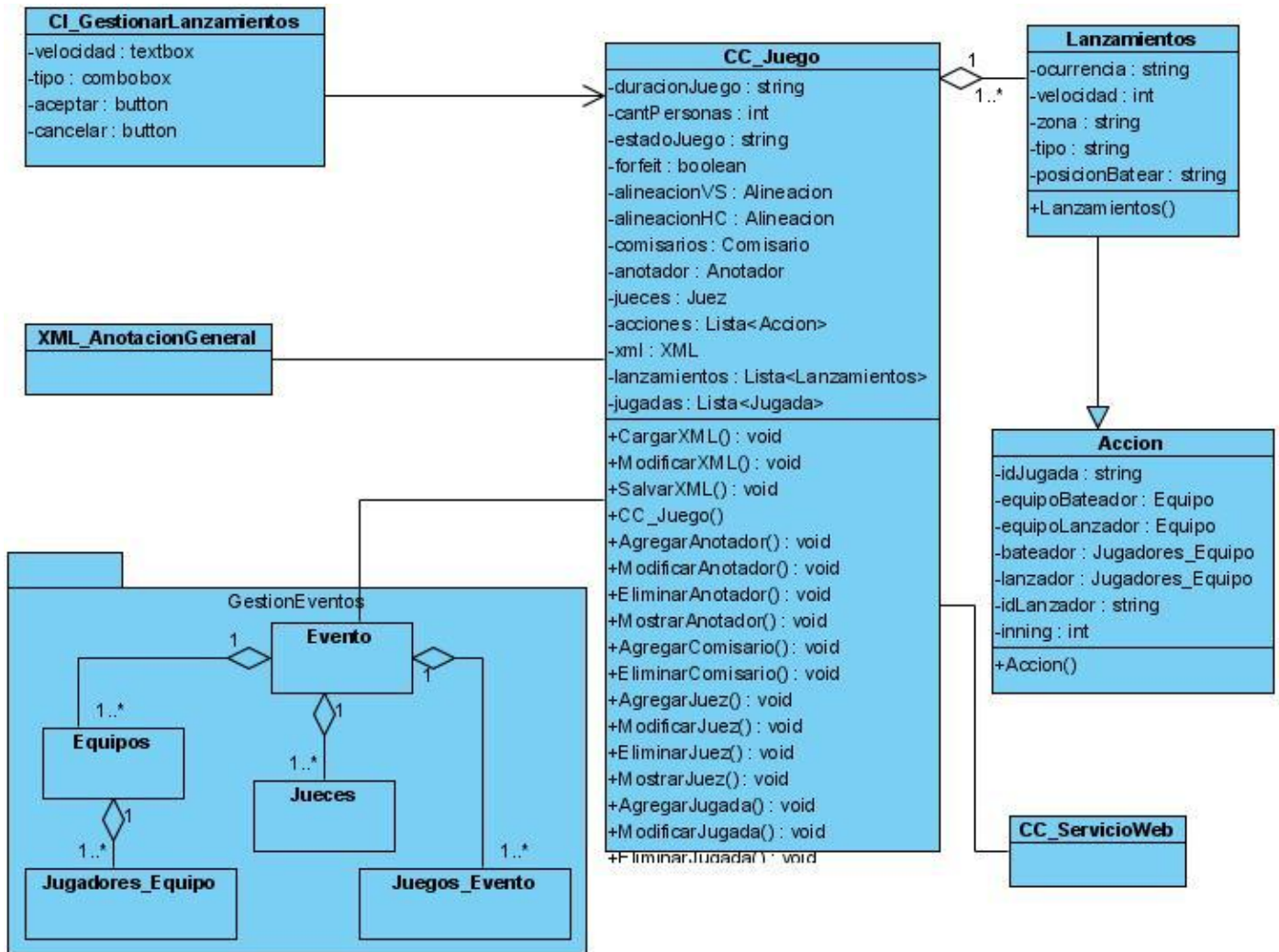


Figura No. 63: Diagrama de Clases de Diseño - CU Gestionar Lanzamientos

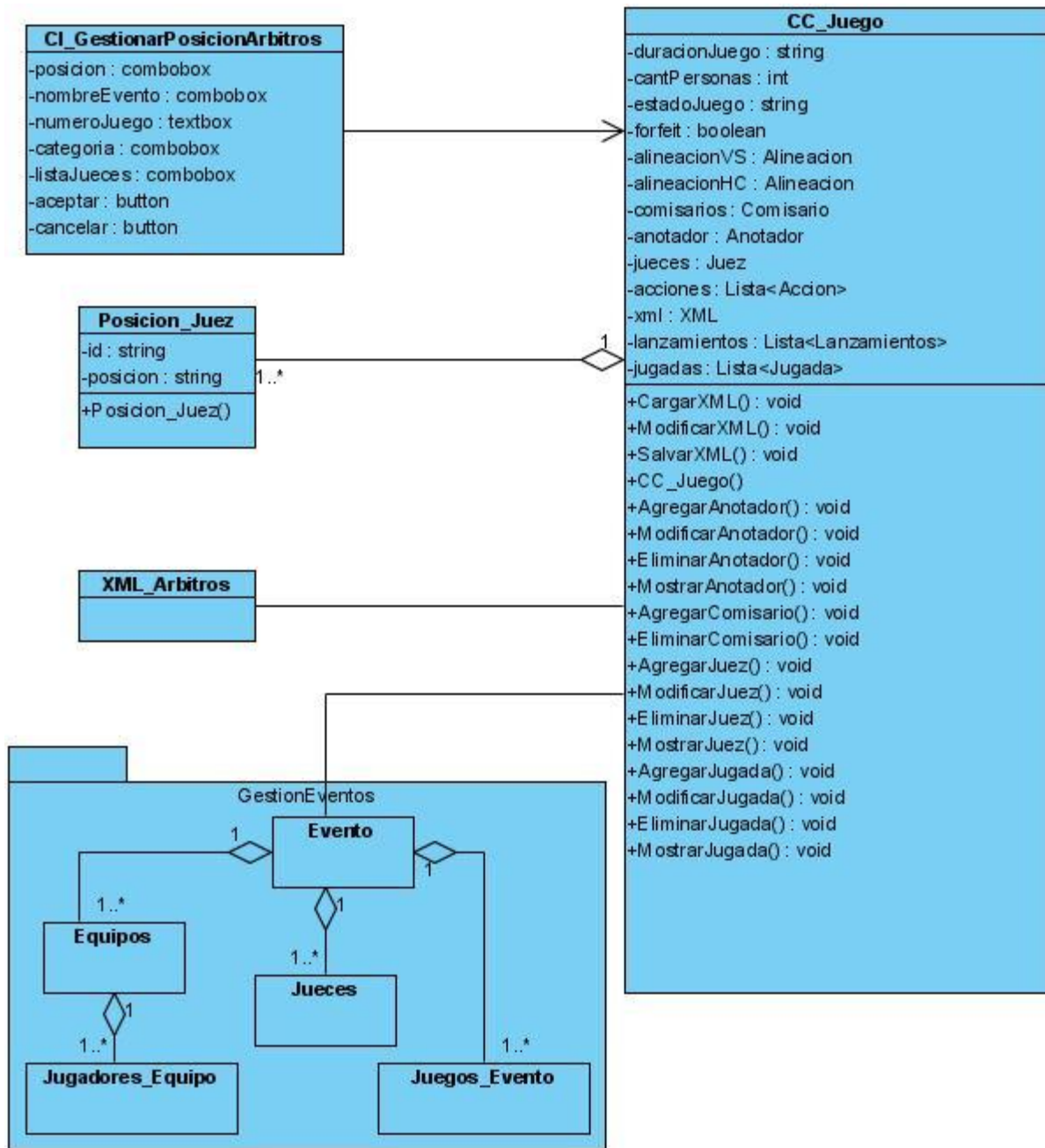


Figura No. 64: Diagrama de Clases de Diseño - CU Gestionar Posición Árbitros

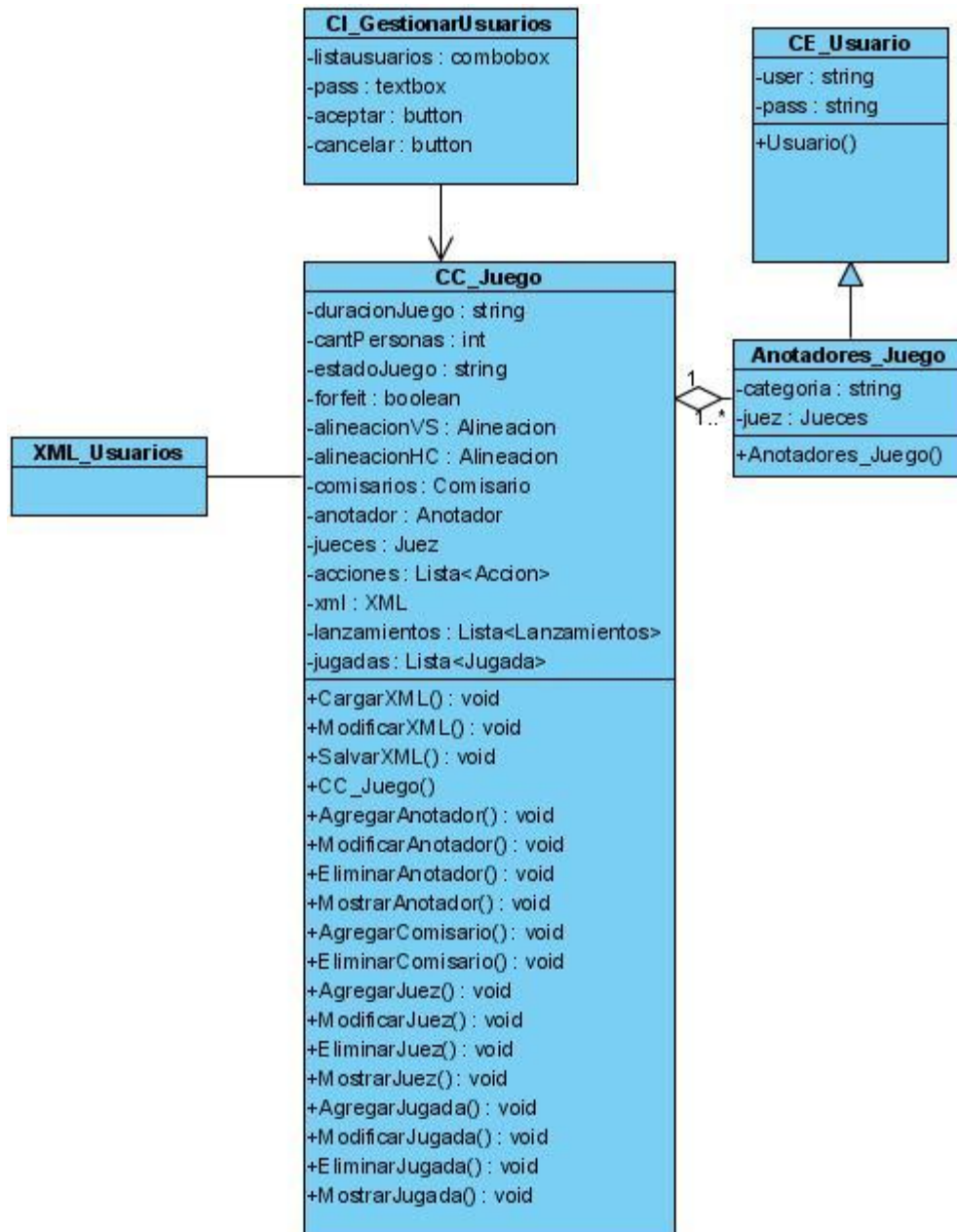


Figura No. 65: Diagrama de Clases de Diseño - CU Gestionar Usuarios

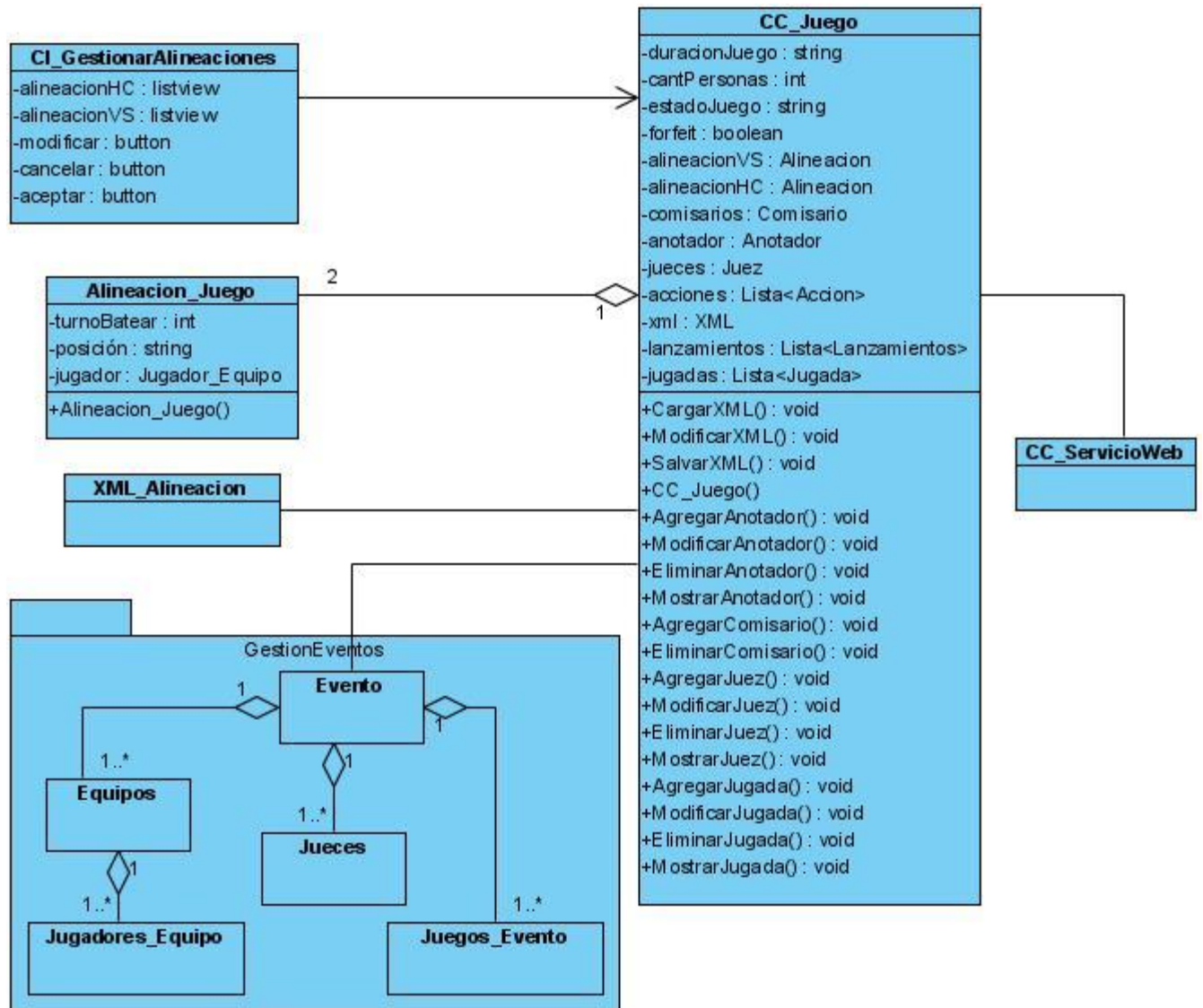


Figura No. 66: Diagrama de Clases de Diseño - CU Gestionar Alineaciones

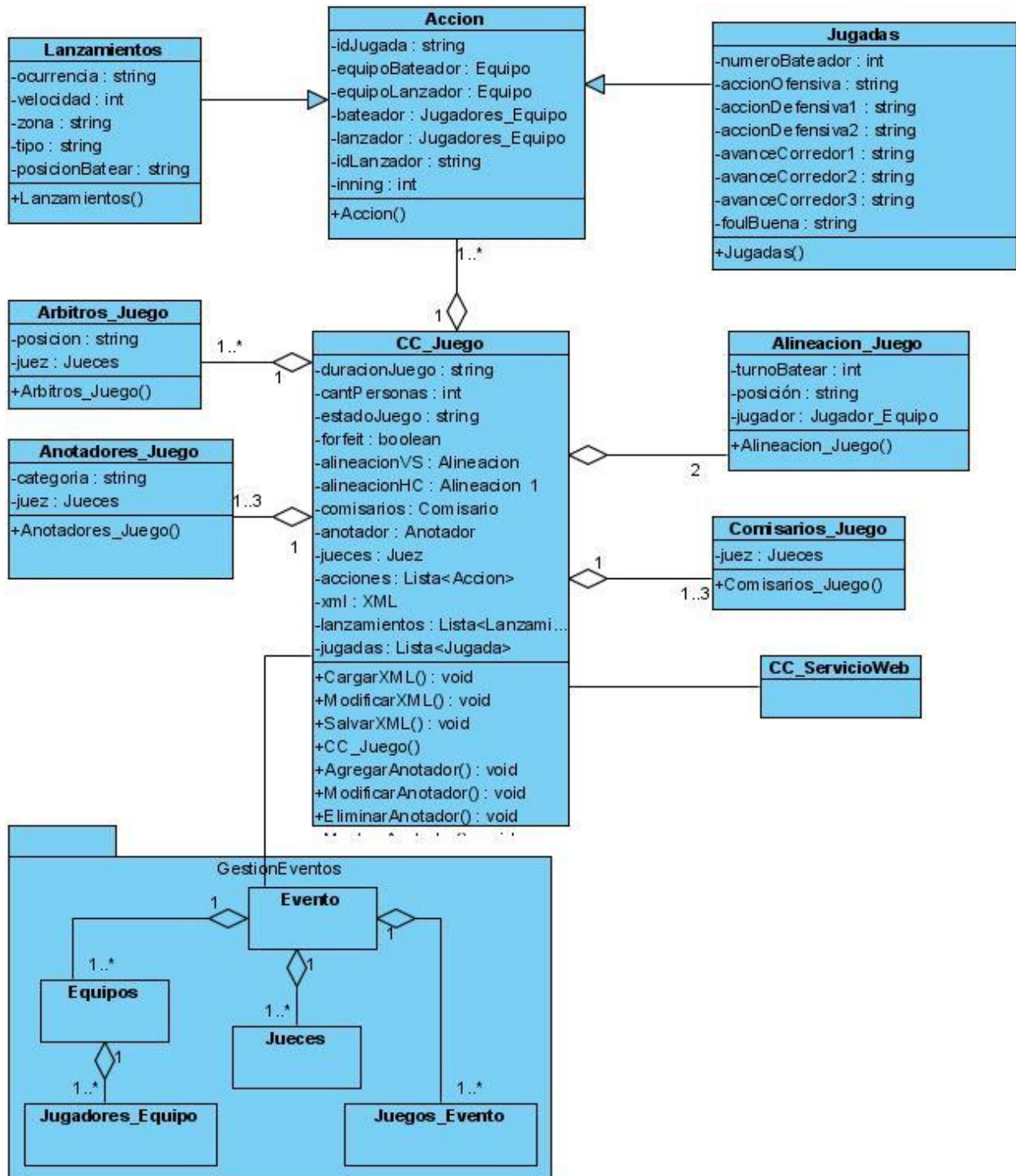


Figura No. 67: Diagrama de Clases de Diseño

Glosario de Términos

Aplicación: Una aplicación es un programa que permite a un determinado usuario realizar uno o varios tipos de trabajos o procesos.

Anotación: Es el proceso que permite llevar un registro detallado de todos los lanzamientos, jugadas y cambios a la ofensiva y defensiva que ocurren durante un juego de béisbol.

Código: Secuencia de números que se utiliza para identificar una jugada en el béisbol.

Codificación: Es el proceso mediante el cual la información es convertida en símbolos numéricos.

Estadísticas: Es un registro acumulativo de todos los récords de bateo, fildeo, corrido de bases y récords de lanzadores, para cada jugador que aparezca en un juego de campeonato de la liga o de post temporada.

ECMA: es una organización internacional basada en membresías de estándares para la comunicación y la información. Fue fundada para estandarizar los sistemas computarizados en Europa. Entre sus especificaciones se encuentra las asociadas al lenguaje de programación C-Sharp (C#).

Ficheros: Son los archivos que guarda y carga un sistema en un tipo de formato o extensión especificada.

FCBA: La Federación Cubana de Béisbol Amateur es la organización encargada de dirigir, controlar y divulgar todos los eventos relacionados con el béisbol que ocurren en Cuba.

Formularios: es el conjunto de campos solicitados por la aplicación que serán almacenados para su posterior uso.

Interfaz de usuario: La interfaz de usuario es lo que permite la comunicación entre el hombre y la computadora.

INDER: El Instituto de Educación Física y Recreación es el órgano rector que se encarga de dirigir y controlar todos los eventos relacionados con el deporte, la educación física y la recreación que ocurren en Cuba.

MS-DOS: Es un sistema operativo comercializado por la compañía Microsoft.

Multiplataforma: Es un término que se emplea para decir que un software puede ejecutarse en diferentes plataformas o sistemas operativos.

Operador: Es el usuario que trabaja con un sistema o software.

Plataforma: Es la base del sistema operativo, lo cual permite que un software se ejecute.

Software: Permite la realización de diferentes tareas a través de la computadora.

Software Libre: Es un software que brinda la posibilidad al usuario que lo adquiere de usarlo, modificarlo y redistribuirlo libremente.

Servicio Web: Permite la comunicación entre máquinas a través de la red.

Tecnología: Es el conjunto de software y hardware que se utilizan para almacenar, procesar y transmitir información de forma rápida y eficiente.