

UNIVERSIDAD DE LAS CIENCIAS INFORMÁTICAS

FACULTAD 9



SISTEMA PARA LA GESTIÓN DE EVENTOS COMPETITIVOS Y ANOTACIÓN DE LOS JUEGOS EN EL BÉISBOL

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

Autores:

José Leandro Cabrera Pérez.

Yosvany Rodríguez Batista.

Tutor:

Ing. Arian Cabezas Regal

Co-Tutor:

Ing. Carlos del Pino Muñoz

Ciudad de La Habana, Junio de 2007

“Año 49 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 firmo la presente a los ____ días del mes de _____ del año _____.

José Leandro Cabrera Pérez

Yosvany Rodríguez Batista

Tutor: **Arian Cabezas Regal**

AGRADECIMIENTOS

A nuestro Comandante en Jefe Fidel, por permitirnos ser parte de este sueño convertido en realidad...

A la Revolución Cubana, por las posibilidades que siempre nos ha brindado...

A la UCI por formarnos como verdaderos Ingenieros...

A nuestros familiares por su amor y comprensión eternos...

A nuestros amigos por darnos el aliento y apoyo que necesitamos...

A nuestro tutor Arian Cabezas por su ayuda durante todo este largo período de trabajo...

A Carlos del Pino por su magnífica atención en cada una de nuestras entrevistas...

A los profesores de la Facultad 9 que nos brindaron su ayuda en todo momento...

A Noel, Anyer, Alain, Pedro y a todos nuestros compañeros de aula que siempre nos ayudaron...

Y en General a todos aquellos que de una manera u otra nos apoyaron y confiaron en nosotros...

Gracias a todos !!!!

DEDICATORIA

A mis abuelos Marialys y Joel, por ser abuelos y padres a la vez y merecer lo mejor de mí...

A mis padres porque los quiero...

A mi novia, por el amor que cada día me hace vivir...

A mis hermanos por ser mi vida...

Yosvany

A mi familia y en especial a mi Abuela y mi Madre.

Jose

RESUMEN

El Instituto Nacional de Deportes, Educación Física y Recreación (INDER) posee actualmente un sistema que lleva por nombre “Béisbol” encargado de gestionar y controlar los eventos competitivos y la anotación de los partidos que se desarrollan de dicho deporte. Los usuarios de esta aplicación han manifestado la existencia de múltiples desventajas que limitan su calidad y eficiencia laboral. Con el objetivo de dar solución a estas dificultades y en respuesta a la solicitud realizada por el INDER, la Federación Cubana de Béisbol Amateur y la Comisión Nacional de Béisbol, se decide implementar un nuevo sistema para la gestión de eventos competitivos y anotación de los juegos de béisbol.

El nuevo sistema tiene la responsabilidad de sustituir al existente, cumpliendo con las exigencias y sugerencias de los usuarios y además incorporando nuevas funcionalidades en una interfaz usable, funcional y accesible.

En este trabajo se presenta todo el proceso de desarrollo de software que se llevó a cabo para la obtención de la aplicación final; así como se explica y justifica el uso de cada herramienta y técnica de desarrollo.

PALABRAS CLAVES

- Sistema codificado
- Anotación
- Jugadas
- Modelado

ÍNDICE

INTRODUCCIÓN.....	1
Capítulo 1 Fundamentación teórica	6
1.1 Introducción.....	6
1.2 Conceptos asociados al dominio del problema.....	6
1.3 Argumentación del objeto de estudio	7
1.3.1 Descripción general.....	7
1.3.2 Descripción actual del dominio del problema	8
1.3.3 Argumentación de la Situación Problemática.....	9
1.4 Análisis de otras soluciones existentes	11
1.5 Conclusiones	12
Capítulo 2 Tendencias y Tecnologías actuales a desarrollar.....	13
2.1 Introducción.....	13
2.2 Las Tecnologías de la Información y las Comunicaciones (TIC)	13
2.3 El Lenguaje Unificado de Modelado (UML) como soporte de la modelación de la solución propuesta	14
2.4 Metodologías	15
2.4.1 El proceso Unificado de Desarrollo de Software (RUP) como base en el desarrollo de la solución propuesta	16
2.5 Plataforma .Net.....	18
2.5.1 XML.....	19
2.5.2 Servicio Web.....	20
2.6 Herramientas a utilizar	21
2.6.1 Herramienta de modelado	22
2.6.2 Herramientas de desarrollo	23

2.7	Lenguajes de programación	25
2.7.1	Lenguaje de programación C#	25
2.7.2	SQL (Structure Query Language).....	26
2.8	Conclusiones	27
Capítulo 3 Presentación de la solución propuesta.....		28
3.1	Introducción.....	28
3.2	Modelación del negocio.....	28
3.2.1	Determinación y justificación de los actores del negocio	29
3.2.2	Determinación y justificación de los trabajadores del negocio.....	30
3.2.3	Diagrama de casos de uso del negocio	31
3.2.4	Descripción de los casos de uso del negocio	31
3.2.5	Diagramas de actividades de los casos de uso del negocio	35
3.2.6	Diagrama del modelo de objetos del negocio	36
3.3	Especificación de requisitos.....	36
3.3.1	Requerimientos funcionales	37
3.3.2	Requerimientos no funcionales	54
3.4	Descripción del sistema propuesto.....	55
3.5	Modelo de casos de uso del sistema	57
3.5.1	Definición de los actores del sistema.....	57
3.5.3	Relación entre paquetes de casos de uso del sistema	58
3.5.4	Descripción textual de los casos de uso del sistema	58
3.6	Conclusiones	58
Capítulo 4 Construcción de la solución propuesta		59
4.1	Introducción.....	59

4.2 Diagramas de clases del diseño	59
4.2.1 Diagramas de clases para cada caso de uso	59
4.2.2 Diagrama de clases persistentes.....	59
4.3 Principios de diseño	59
4.3.1 Estándares de la interfaz	60
4.3.2 Concepción general de la ayuda.....	60
4.3.3 Tratamientos de Excepciones	61
4.4 Estilos y patrones utilizados.....	61
4.4.1 Diseño	61
4.4.2 Arquitectura.....	62
4.5 Diseño de la base de datos	63
4.6 Estándares de la codificación.....	63
4.7 Encriptación de datos	64
4.8 Modelo de despliegue	65
4.9 Modelo de Componentes	66
4.10 Conclusiones.....	66
Capítulo 5 Estudio de factibilidad	67
5.1 Introducción.....	67
5.2 Planificación basada en casos de uso. Análisis de puntos de casos de uso	67
5.3 Costo del Proyecto.....	72
5.4 Beneficios tangibles e intangibles	73
5.4.1 Beneficios tangibles.....	73
5.4.2 Beneficios intangibles.....	73
5.5 Análisis de costos y beneficios.....	74

5.6 Conclusiones	74
COCLUSIONES.....	75
RECOMENDACIONES.....	76
BIBLIOGRAFÍA CONSULTADA.....	78
GLOSARIO DE TÉRMINOS.....	79
ANEXOS.....	80

INTRODUCCIÓN

En Cuba el desarrollo del deporte se ha realizado de manera satisfactoria con el transcurso de los años. El Instituto Nacional de Deportes, Educación Física y Recreación (INDER¹), ha sido el protagonista de todo cuanto se ha logrado y se continúa realizando para beneficio del mismo. Un gran equipo de trabajadores ha dedicado su vida por completo al desarrollo de las diversas necesidades del deporte cubano, haciendo posible que se obtengan grandes logros, de manera que en la actualidad Cuba constituye una potencia deportiva mundial a pesar de su pequeña extensión territorial.

Un buen número de tareas o procesos son llevados a cabo por esta institución. La gestión de eventos competitivos y las anotaciones asociadas al juego de béisbol en Cuba son dos de los procesos a los cuales irán dirigidos los objetivos de este trabajo.

Desde ya hace varios años el INDER ha venido desarrollando sistemas propios para gestionar sus estadísticas, por lo que en el año 1977 implementa un sistema de Gestión y Anotación de los eventos en el béisbol el cual es de mucha importancia pues gracias a él se cuenta con un gran número de datos y estadísticas del deporte nacional. Con el avance de las tecnologías se pueden implementar nuevos sistemas capaces de recoger más información y así brindar una serie de estadísticas más amplias.

En la actualidad existen dificultades reales en el sistema de anotaciones asociadas al juego de béisbol en Cuba por lo que quedan enmarcados como **situación problemática** los siguientes aspectos:

- Se emplea un sistema obsoleto y con una interfaz de usuario poco funcional y poco usable por estar desarrollada en MS-DOS.
- La difícil manipulación de los datos por parte del sistema hace que todo el peso de las anotaciones recaiga en el anotador del encuentro, haciendo que éste requiera aprender un sistema codificado para poder desarrollar su labor.
- Las estadísticas se ven restringidas a nivel de jugada o sea que solo se pueden recoger estadísticas

¹ INDER (Instituto Nacional de Deportes, Educación Física y Recreación): Ver el epígrafe “Conceptos asociados al dominio del problema” del Capítulo 1.

generales por cada jugada que se realice.

Por tanto, el **problema a resolver** reside en la existencia de un sistema para la gestión de eventos competitivos y anotación de los juegos de béisbol limitado en información y la ausencia de una interfaz usable, funcional y accesible para el usuario.

El **objeto de estudio** lo constituye el proceso de gestión y control de la información por parte del INDER.

El **objetivo general** del trabajo consiste en desarrollar un prototipo funcional para la gestión de los eventos competitivos y anotación de los juegos de béisbol en Cuba.

El **campo de acción** queda enmarcado en la gestión de eventos competitivos y anotación del juego de béisbol en el INDER.

Tomando como base lo antes expuesto se puede plantear la siguiente **hipótesis**:

Si se desarrolla un sistema capaz de almacenar un mayor número de datos en el momento de anotar los juegos, se desarrolla una interfaz visual usable, funcional y accesible y se automatiza el proceso de gestión de los eventos competitivos, se podrá llevar un registro amplio y detallado de cada acción del partido, además de facilitar la labor de los informáticos-anotadores y de la Comisión Nacional en el instante de realizar las anotaciones y la gestión de los eventos.

Para poder desarrollar estas actividades se realizaron las siguientes **tareas**:

- Investigación sobre antecedentes del tema.
- Estudio de sistema de anotación que se utiliza actualmente en Cuba.
- Revisión Bibliográfica para constatar el estado del arte del Objeto de Estudio.
- Entrevistas con los especialistas en las materias a automatizar.
- Selección de la plataforma y herramienta que se utilizarán para el diseño de la aplicación.
- Capacitación en la tecnología a usar para su desarrollo e implementación.
- Modelación del prototipo funcional para la gestión de eventos competitivos y anotación de los juegos

de béisbol en Cuba.

- Diseño e Implementación del prototipo funcional para la gestión de eventos competitivos y anotación de los juegos de béisbol en Cuba.

Durante todo el proceso investigativo realizado se utilizaron un conjunto de **métodos científicos de investigación**. Estos métodos se clasifican en:

Teóricos: Posibilitan el conocimiento del estado del arte del fenómeno, su evolución en una etapa determinada, su relación con otros fenómenos, así como su aislamiento como objeto estudiado

Empíricos: Estos métodos permiten extraer de los fenómenos analizados las informaciones que se necesitan sobre ellos a través de observaciones, del uso de técnicas opináticas y la propia experimentación.

Dentro de los teóricos se emplearon los siguientes:

- Análisis histórico lógico para investigar sobre las aplicaciones informáticas de esta tipo implementadas en Cuba y el resto del mundo, y su uso en el ámbito nacional, así como las ventajas y desventajas que posee el que se utiliza actualmente.
- Analítico-Sintético para resumir, enunciar y describir los requerimientos enunciados por los profesionales.
- Modelación para realizar una reproducción simplificada de la realidad. Permite descubrir nuevas relaciones y cualidades del objeto de estudio.

Dentro de los empíricos se emplearon:

- Entrevistas individuales y colectivas con los profesionales del INDER y de la Federación Cubana de Beisbol Amateur para comprender la situación real del problema existente, así como las opiniones y sugerencias de los clientes.
- Observación para realizar un registro visual de lo que ocurre en el entorno del problema aportar nuevos elementos que puedan ser de interés científico.

La solución propuesta posee una elevada importancia para el béisbol y el país en general. Tecnológicamente se logra el aporte de un sistema que a pesar de no ser el primero en esta rama, mejora considerablemente las condiciones del actual, pues automatiza el proceso de anotación de los juegos de béisbol y la manipulación de los datos para la gestión de los eventos competitivos.

En el plano económico representa una importante ventaja pues reduciría ampliamente el costo de aplicaciones de esta índole y daría lugar a la implementación de todo un conjunto de módulos que favorecerán el desarrollo del béisbol en Cuba y además le solucionaría al país la necesidad de realizar inversiones en la compra de softwares con estos fines en el extranjero.

Como se menciona con anterioridad, este software no sería el primero de su clase en el país en el sentido de que existe una aplicación utilizada actualmente para dar solución a la anotación y control de los juegos y eventos del béisbol; pero si se tiene en consideración el alto nivel de automatización que se pretende alcanzar en la anotación de los juegos, la reestructuración de la base de datos que se desea realizar con el objetivo de lograr recoger mayor cantidad de información y una mejor manipulación de la misma, así como las ventajas que representaría para los usuarios la aparición de una interfaz Windows más funcional y accesible; entonces sí podría catalogarse la aplicación como única de su tipo en el país.

Para mayor facilidad y organización a la hora de realizar el estudio del documento, este trabajo se encuentra dividido en cinco capítulos y cada uno de ellos estructurado por epígrafes.

Capítulo 1: Se plantean todos los elementos teóricos que sustentan el problema científico y los objetivos del trabajo.

Capítulo 2: Se realiza un análisis comparativo de las herramientas y tecnologías existentes que resultan adecuadas para la construcción del sistema a desarrollar.

Capítulo 3: Se realiza la modelación del negocio, presentando los diagramas de casos de uso del negocio y los diagramas de actividades para una mejor visión del mismo, utilizándose el Rational Rose como herramienta de modelación. Se realiza además el levantamiento de requisitos del sistema propuesto.

Capítulo 4: Está dedicado al diseño del sistema. Se ofrecen los diagramas de casos de uso del sistema, diagrama de clases persistentes, modelo de datos, modelo de despliegue, modelo de componentes, etc. Se presentan además las generalidades de la implementación y de la interfaz del sistema.

Capítulo 5: Se realiza el estudio de factibilidad sobre el sistema obteniéndose los beneficios tangibles e intangibles del mismo y realizando el análisis de los costos de desarrollo de esta propuesta.

Capítulo 1

Fundamentación teórica

1.1 Introducción

En este capítulo se exponen una serie de argumentos y explicaciones que permiten comprender de manera clara y concisa la situación real existente en el entorno beisbolero.

Se ofrece un conjunto de conceptos asociados al dominio del problema. También se explica de forma detallada el objeto de estudio del trabajo a través de una descripción general del mismo; la descripción actual del dominio del problema y se argumenta detalladamente la situación problemática existente. Además se realizó un análisis de otras soluciones existentes hasta el momento.

1.2 Conceptos asociados al dominio del problema

Anotación de los Juegos: Registro detallado de las jugadas y lanzamientos que tienen lugar durante un juego de béisbol.

Comisión Provincial de Béisbol: Institución encargada de controlar, gestionar y asegurar las actividades relacionadas con este deporte en cada provincia. Son los máximos responsables de la calidad y desarrollo de los eventos provinciales a celebrar. Se encargan de seleccionar a los jugadores de mejor rendimiento en la provincia y conformar un equipo que los represente en los eventos nacionales.

Comisión Nacional de Béisbol: Institución encargada de controlar, gestionar y asegurar las actividades relacionadas con el béisbol en Cuba. Constituye la máxima autoridad de este deporte en el país. Son los responsables de aprobar o no los rosters de cada equipo que participa en los eventos nacionales a desarrollar, así como de seleccionar los jugadores de mayor rendimiento del país para conformar un equipo que represente a Cuba en eventos internacionales.

Calendario de los Juegos: Planificación de los juegos que se desarrollan durante un evento atendiendo a la fecha, hora y equipos involucrados en el encuentro.

Evento Competitivo: Competencia que se desarrolla en una fecha determinada, donde participan varios equipos por alcanzar un lugar.

INDER: El Gobierno Revolucionario crea el Instituto Nacional de Deportes, Educación Física y Recreación “INDER” el 23 de febrero de 1961, con la idea de que este instituto tenga la misión de ser el órgano superior encargado de planificar, dirigir, racionalizar, divulgar y ejecutar las actividades del deporte, la educación física y la recreación, en todos los niveles y lugares del país.(CORUJEDO 1965)

Jueces del juego: Conjunto de árbitros, comisionados y anotadores que participan en el partido.

Libro del Evento: Lugar donde se recogen todos los datos correspondientes a un evento en específico. En él queda asentado el calendario de juego del evento, los rosters aprobados de cada equipo y los jueces que participarán en dicho evento.

Sistema codificado de anotación: Sistema que se emplea para la anotación de las estadísticas del juego de béisbol. Como su nombre lo indica posee una codificación para cada jugada posible a suceder sobre el terreno de béisbol. Consta de seis casillas para el registro del código.

1.3 Argumentación del objeto de estudio

1.3.1 Descripción general

La gestión de eventos competitivos relacionados con el béisbol en Cuba tiene lugar a través del personal del INDER destinado para estos fines. Para ello se utiliza un sistema implementado sobre MS-DOS que se encarga de realizar la mayoría de las operaciones. En esta aplicación se recogen los datos necesarios para organizar el evento. Del evento en general se registran datos como la cantidad de equipos que participarán en el event, calendario de enfrentamientos del torneo, estadios donde se desarrollarán los diferentes encuentros, jueces del evento, etc. Del equipo se recogen los jugadores que lo conforman, el nombre del equipo y las siglas que lo representan, director del equipo, entre otros. De cada jugador se recogen datos como nombre y apellidos, número de la camiseta, posición que ocupará en el juego, mano

de batear, mano de lanzar, etc. Del estadio se recoge el municipio y provincia donde se encuentra, nombre del estadio, etc.

Con este trabajo se pretende automatizar la manipulación de los datos para la gestión de los eventos competitivos relacionados con el béisbol. El objetivo es realizar una aplicación capaz de gestionar internamente todo cuanto tenga relación con el evento, de manera que no sea necesario que el usuario, en este caso los coordinadores del torneo, tengan que realizar manualmente toda una serie de movimientos y operaciones para conformar los equipos, grupos, etc.

Otro de los procesos que se pretende automatizar es el de anotación del juego de béisbol. Actualmente este proceso se lleva a cabo de forma manual. En cada estadio se encuentran uno o varios anotadores oficiales del encuentro, los cuales son los responsables de llevar al papel jugada tras jugada lo que ocurre en el terreno. Para realizar esta tarea se emplea el sistema de anotaciones codificado cubano. El anotador del juego tiene que dominar a la perfección el manual de anotaciones para poder desempeñar su labor correctamente. Tras finalizar el encuentro estos datos son llevados a formato digital a través de un sistema desarrollado sobre MS-DOS que permite la inserción de las anotaciones.

1.3.2 Descripción actual del dominio del problema

En la actualidad existen dificultades reales en el sistema de anotaciones y gestión de eventos competitivos asociados al juego de béisbol en Cuba, pues se emplea un sistema obsoleto y con una interfaz de usuario poco funcional y poco usable desarrollada en MS-DOS. Además el peso de las anotaciones recae completamente en el anotador del partido pues requiere del aprendizaje del sistema codificado de anotación² para poder desarrollar su labor y la anotación del juego se lleva a cabo a nivel de jugada por lo que se dejan de recoger estadísticas muy importantes en cada juego que se desarrolla.

² Sistema codificado de anotación: Ver el epígrafe “Conceptos asociados al dominio del problema” del Capítulo 1.

1.3.3 Argumentación de la Situación Problemática

La aplicación que se emplea actualmente para la anotación de los juegos de béisbol está vigente en Cuba desde 1977. Los realizadores de dicha aplicación en aquel entonces se basaron en el sistema de anotaciones codificado para el béisbol cubano. En este sistema, como su nombre lo indica, cada jugada posee un código de anotación fijo. Básicamente consta de siete casillas como se muestra en la **figura 1.1**

1	2	3	4
	5	6	7

Figura 1.1 Casillero de anotación del Sistema Codificado Cubano.

Casilla 1: Numero del jugador que esta consumiendo su turno al bate

Casilla 2: Acción ofensiva del jugador.

Casilla 3: Acción defensiva realizada sobre la acción ofensiva del jugador.

Casilla 4: Complementación de la acción defensiva realizada sobre la acción ofensiva del jugador.

Casilla 5, 6 y 7: Avance de los corredores en base, primera, segunda y tercera respectivamente.

En la etapa en que se desarrolló esta aplicación las tecnologías existentes en Cuba en el mundo de la informática eran escasas. MS-DOS era el sistema operativo que se utilizaba en la mayoría de las computadoras del momento, motivo por el cual la aplicación no posee una interfaz gráfica usable y funcional para el usuario. Esta situación continúa siendo un problema actual pues la aplicación que se continúa empleando es ésta y tanto los usuarios como los creadores de la misma se han conformado con ella y se resisten o temen al cambio tecnológico existente, de manera que no se ha implementado ninguna versión que perfeccione la inicial.

La manipulación de los datos es otro de los problemas existentes en la aplicación actual. La inserción de una jugada está implementada de manera que el anotador tiene que insertar el código perteneciente a la jugada. Por esta razón los anotadores oficiales del béisbol tienen que aprender de memoria el manual de anotaciones completamente, de forma tal que al producirse cualquier jugada en el terreno, el anotador tiene que saber cual es el código que le pertenece a la misma para poder realizar la operación, situación que complica su labor y que sería mejorable si la codificación de la jugada se llevara a cabo internamente por la aplicación de forma transparente para el usuario.

De la forma de manipular los datos también se desprende otra dificultad real y que si es analizada bien a fondo se puede observar a simple vista las consecuencias que trae consigo. El problema radica en que actualmente las anotaciones del juego de béisbol se llevan a nivel de jugada; o sea que el anotador del encuentro no recoge ninguna acción o evento hasta tanto no se produzca una jugada en cuestión. Por ejemplo: si el bateador en turno batea n veces de foul, el anotador permanecerá sin anotar ninguna acción; tiene que esperar a que el bateador batee de hit o sea retirado por alguna de las vías posibles en el béisbol para poder llevar a la hoja de anotaciones la jugada codificada.

A corto plazo quizás esta situación no sea vista como algo serio, pero lo cierto es que cuando transcurren largos períodos de tiempo y se recurre a las anotaciones de los encuentros, se puede apreciar la ausencia de detalles de los partidos que en ocasiones pueden ser de mucha utilidad.

Otro de los ejemplos que ilustra este detalle lo es el caso del pitcheo. Las características de cada lanzamiento que realiza el pitcher durante su estancia en el montículo no se registran en la hoja de anotaciones. Cuando pasa un tiempo y se desea saber sobre cual lanzamiento el pitcher tiene mayor o menor efectividad en sus lances, no se puede acudir a las anotaciones porque estos datos se pierden y en casos como en el del *Clásico Mundial*³ donde se necesitaba conocer todas las características de los rivales, no se puede contar con esos detalles, de ahí la importancia de recoger el partido lanzamiento por lanzamiento y no a nivel de jugada.

³ Clásico Mundial: Evento deportivo. Ver el Glosario de términos.

1.4 Análisis de otras soluciones existentes

Como se manifestó anteriormente en Cuba existe un software que lleva por nombre Béisbol⁴ a través del cual se gestionan las estadísticas correspondientes al juego de béisbol y se realiza el control de los eventos competitivos asociados al mismo. Este software a lo largo de estos años ha cumplido con las expectativas del INDER para la realización de dichas anotaciones. Es necesario destacar que el mismo consta de una implementación muy bien realizada y completa. Brinda un gran número de estadísticas relacionadas con el deporte nacional cubano desde sus inicios.

Fue implementado los propios trabajadores del INDER en el año 1977 con el objetivo mejorar la situación de las anotaciones y control de los eventos competitivos nacionales e internacionales del béisbol, que hasta ese momento eran llevadas y archivadas en papeles. La realización de este software logró un importantísimo avance en dicho proceso, pues resulta mucho más cómodo y de fácil acceso el poder contar con toda la documentación existente en formato digital.

La tecnología existente en Cuba en esa etapa resultaba muy pobre si se compara con la que actualmente posee el país. Por este motivo el software fue implementado para MS-DOS, sistema operativo que no posee una interfaz gráfica usable por lo que resulta incómodo su uso después de la revolución gráfica implantada por la familia WINDOWS en el mundo informático. A lo largo de estos años el INDER no volvió a implementar ninguna versión que mejorara aspectos negativos del software, ni tampoco se ha encargado de realizar una implementación para WINDOWS que gane en funcionalidad y usabilidad. Por tanto, el software con el cual se labora actualmente continúa desplegándose en MS-DOS y desaprovechando las potencialidades gráficas de los actuales manejadores de ventanas, situación donde reside una de las dificultades mayores del mismo.

Básicamente consiste en una aplicación que gestiona los eventos competitivos y además permite compilar las anotaciones de los juegos de béisbol. Trabaja con ficheros .txt para el control y almacenamiento de los datos.

En cada estadio del país existe una copia de la aplicación para llevar a cabo la anotación del encuentro

⁴ Béisbol: Nombre del software que se utiliza actualmebte en Cuba para la gestión de eventos competitivos y anotación de los juegos.

beisbolero. Cada vez que tiene lugar una jugada en el terreno esta es llevada a la aplicación de manera codificada a través del sistema de anotaciones cubano para el béisbol. Después de insertarse la jugada se compila, se toma el .txt y se envía por FTP para el servidor principal en el INDER. Este proceso se realiza hasta durante el desarrollo del partido de béisbol.

Una vez que se hayan terminado todos los partidos de la jornada, en el *Estadio Latinoamericano*⁵ un compañero realiza la compilación de todos los ficheros y genera nuevas páginas de estadísticas. Luego tiene que enviar a todas las provincias cada uno de los .txt para que actualicen su aplicación en cada estadio de manera independiente.

Todo esto trae consigo que el proceso sea lento pues se invierte mucho tiempo anotando, compilando y subiendo el fichero para el servidor principal y en la mayoría de los casos este fichero solo se sube cuando finaliza el inning completo por lo que no se puede contar con los datos del partido en tiempo real.

Además la anotación que se lleva a cabo tiene lugar como se ha mencionado con anterioridad a nivel de jugada, por lo que los datos de cada lanzamiento que se realiza se pierden en su gran mayoría.

1.5 Conclusiones

En este capítulo se realizó un resumen de los principales conceptos asociados al dominio del problema con el objetivo de lograr un mayor entendimiento del mismo. Se explica de manera argumentada el objeto de estudio del trabajo.

También se realizó el análisis de aplicaciones existentes en nuestro país que sirven de guía en los objetivos que persigue este trabajo.

⁵ Estadio Latinoamericano: Ver Glosario de términos.

Capítulo **2**

Tendencias y Tecnologías actuales a desarrollar

2.1 Introducción

En la actualidad existen diversos recursos que pueden ser utilizados para el desarrollo e implementación de un software. Las tecnologías actuales hacen posible que el proceso de desarrollo de un sistema resulte cada vez más fácil y eficiente a la vez. El gran número de herramientas, lenguajes y metodologías existentes hasta el momento ofrece al personal de trabajo la ventaja de poder escoger técnicas y herramientas más poderosas y cómodas en dependencia del objetivo que se persiga en el proyecto.

En este capítulo se realiza un análisis comparativo de las tecnologías existentes que resultan adecuadas para la construcción del sistema a desarrollar con el objetivo de brindar una explicación concreta que haga entender fácilmente el por qué de la propuesta final.

2.2 Las Tecnologías de la Información y las Comunicaciones (TIC)

Las TIC constituyen un elemento esencial de la Sociedad de la Información, estas brindan la capacidad universal de acceder y contribuir a la información, las ideas y el conocimiento. Promover el intercambio y el fortalecimiento de los conocimientos mundiales en favor del desarrollo permitiendo un acceso equitativo a la información para actividades económicas, sociales, políticas, sanitarias, culturales, educativas y científicas dando acceso a la información que está en el dominio público solo son posibles gracias a las TIC. Son muchas las ventajas que generan el uso de las tecnologías de la información, tales como aumento de la cultura general integral, tener cada día un público mejor instruido, posibilidad de nuevos puestos laborales, innovación, oportunidades comerciales y el avance de las ciencias.

Desde le punto de vista de la educación las TIC elevan la calidad del proceso educativo derribando las barreras del espacio y del tiempo, permitiendo la interacción y colaboración entre las personas para la construcción colectiva del conocimiento y de fuentes de información de calidad.

El uso de las TIC en las empresas posibilita una mejora de los procesos productivos internos al proporcionar herramientas que facilitan la clasificación, organización, manejo y filtro de la información, y de cara al exterior mejoran los procesos de interacción con los agentes externos: clientes, proveedores y socios, permitiendo la independencia del tiempo y del espacio, posibilitando el acceso desde cualquier lugar y a cualquier hora, abriendo la posibilidad de nuevos negocios en la red. (PARDO 2003)

2.3 El Lenguaje Unificado de Modelado (UML) como soporte de la modelación de la solución propuesta

Un lenguaje para el modelado de objetos es un conjunto estandarizado de símbolos y de modos de disponerlos para modelar un diseño de software orientado a objetos. El uso de un lenguaje de modelado es más sencillo que la auténtica programación, pues existen menos medios para verificar efectivamente el funcionamiento adecuado del modelo.

UML (Unified Modeling Language, Lenguaje Unificado de Modelación) Es un lenguaje de modelado visual que se utiliza para especificar, visualizar, construir y documentar artefactos del sistema de un Software. Se emplea para entender, diseñar, configurar, mantener y controlar la información sobre los sistemas a construir. (IVAR JACOBSON 1999)

UML capta la información sobre la estructura estática y el comportamiento dinámico de un sistema. Un sistema se modela como una colección de objetos discretos que interactúan para realizar un trabajo que finalmente beneficia a un usuario externo. El lenguaje de modelado pretende unificar la experiencia pasada sobre técnicas de modelado e incorporar las mejores prácticas actuales en un acercamiento estándar. (GRADY BOOCH 1999)

Además tiene las siguientes características:

- Permite modelar sistemas utilizando técnicas orientadas a objetos (OO).
- Permite especificar todas las decisiones de análisis, diseño e implementación, construyéndose así modelos precisos, no ambiguos y completos.
- Puede conectarse con lenguajes de programación (Ingeniería directa e inversa).

- Permite documentar todos los artefactos de un proceso de desarrollo (requisitos, arquitectura, pruebas, versiones, etc.).
- Cubre las cuestiones relacionadas con el tamaño propio de los sistemas complejos y críticos.
- Es un lenguaje muy expresivo que cubre todas las vistas necesarias para desarrollar y luego desplegar los sistemas.
- Existe un equilibrio entre expresividad y simplicidad, pues no es difícil de aprender ni de utilizar.
- UML es independiente del proceso, aunque para utilizarlo óptimamente se debería usar en un proceso que fuese dirigido por los casos de uso, centrado en la arquitectura, iterativo e incremental.

2.4 Metodologías

La metodología en el desarrollo de un software es el plano de apoyo, la guía que conducirá a realizar un software de calidad, es el conjunto de métodos que se deben realizar para realizar un software con calidad. Con la selección y utilización de la metodología se tienen clientes más complacidos y desarrolladores satisfechos, además servirá para que el desarrollo y confección del software no parezca complicado, riguroso y difícil de controlar.

Algunas metodologías conocidas:

Extreme Programming (XP)

Es una de las metodologías de desarrollo de software más exitosas en la actualidad utilizadas para proyectos de corto plazo, corto equipo y cuyo plazo de entrega era ayer. La metodología consiste en una programación rápida o extrema, cuya particularidad es tener como parte del equipo, al usuario final, pues es uno de los requisitos para llegar al éxito del proyecto (SANCHEZ 2006).

Lo fundamental en este tipo de metodología es:

- La comunicación, entre los usuarios y los desarrolladores.
- La simplicidad, al desarrollar y codificar los módulos del sistema.
- La retroalimentación, concreta y frecuente del equipo de desarrollo, el cliente y los usuarios finales.

Microsoft Solution Framework (MSF)

Esta es una metodología flexible e interrelacionada con una serie de conceptos, modelos y prácticas de uso, que controlan la planificación, el desarrollo y la gestión de proyectos tecnológicos. MSF se centra en los modelos de proceso y de equipo dejando en un segundo plano las elecciones tecnológicas. (SANCHEZ 2006)

MSF tiene las siguientes características:

- Adaptable.
- Escalable.
- Flexible.

2.4.1 El proceso Unificado de Desarrollo de Software (RUP) como base en el desarrollo de la solución propuesta

La calidad en el desarrollo y mantenimiento del software cobra cada día mayor importancia entre los objetivos estratégicos de las organizaciones. Diversos estudios relacionados con el tema han sido publicados en los últimos años. Para guiar el proceso de desarrollo del software es necesario escoger la metodología a emplear.

Durante el proceso de desarrollo de software se utilizó la metodología RUP (Rational Unified Process, Proceso Unificado de Rational). RUP hace énfasis en la adopción de las mejores prácticas del desarrollo de software, como una manera de reducir los riesgos inherentes en el desarrollo de una nueva aplicación de software, de esta manera se logran resultados más predecibles unificando al equipo con procesos comunes que mejoran la comunicación y crean un entendimiento de todas las tareas y responsabilidades.

RUP posee las siguientes características:

- Iterativo e Incremental.
- Dirigido por los Casos de Uso.
- Centrado en la Arquitectura.

Iterativo e Incremental: Una iteración involucra actividades de todos los flujos de trabajo, aunque desarrolla algunos en mayor grado 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 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.

Los mini-proyectos incorporan incrementalmente nueva funcionalidad y su desarrollo es una iteración.

- Obtiene un sistema robusto
- Reduce el riesgo de tener un mal producto
- Reduce el riesgo de no obtener el producto en el tiempo previsto
- Permite atacar problemas con requisitos incompletos.

Dirigido por los Casos de Uso: Los casos de uso reflejan lo que los usuarios futuros necesitan y desean, lo que es representado por los requerimientos. A partir de aquí los casos de uso guían el proceso de desarrollo pues los modelos que se obtienen, como resultado de los diferentes flujos de trabajo, representan la realización de los casos de uso.

- Proporcionan los Requisitos Funcionales del Sistema.
- Describen toda la funcionalidad del sistema.
- Cambios en requisitos de un caso de uso fácil de detectar las clases y componentes que afectan.

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. RUP se desarrolla mediante iteraciones, comenzando por los CU relevantes desde el punto de vista de la arquitectura.

- Los casos de uso describen la funcionalidad del sistema.

- La arquitectura define la forma del sistema.
- Se describe mediante vistas que incorporan el 5-10% de los casos de uso más relevantes.

2.5 Plataforma .Net

La plataforma .NET permite la creación y consumo de aplicaciones, procesos y sitios WEB basados en XML que comparten y combinan información y funcionalidad entre cada uno de ellos por diseño, en cualquier plataforma o dispositivo inteligente, y provee soluciones para las necesidades específicas de organizaciones e individuos.

La plataforma .NET incluye una familia de productos, construidos sobre estándares de Internet, que provee funcionalidad para los aspectos de construcción de software (herramientas), administración (servidores) y uso (servicios base y clientes inteligentes) y experiencias de servicios Web⁶ XML. .NET se convertirá en parte de las aplicaciones, herramientas y servidores que ya se usan hoy en día, así como en productos que extienden los servicios Web XML a capacidades para todas las necesidades de negocios.(DESARROLLOWEB.COM 2006)

Para integración, los “Servicios Web XML” sobre .NET hacen posible que diferentes piezas de software trabajen en conjunto para:

- Unir aplicaciones. Convertir aplicaciones independientes a constelaciones de aplicaciones para usar datos reales.
- Intercambiar datos. Los datos de clientes residen en aplicaciones aisladas, impidiendo ofrecer nuevos servicios que exploten esos datos.

Existen más de 40 lenguajes adaptados a .Net, entre estos se encuentran: Visual Basic .NET, Visual C#.NET, Visual C++ .NET, Visual J#, entre otros. Además ofrece plena interoperabilidad entre ellos, por lo que es posible construir un componente en un lenguaje, introducirlo en una aplicación escrita en otro e

⁶ Servicio Web: Ver Glosario de términos.

incluso heredarlo y añadir nuevas características en un tercero, pero el lenguaje nativo que ha sido desarrollado con el objetivo de ser utilizado en .NET es C# (MORAN 2006)

2.5.1 XML

XML está transformado completamente a la creación y el uso de software. El Web revolucionó la comunicación entre usuarios y aplicaciones. XML está revolucionando la comunicación entre aplicaciones o, de forma más general, la comunicación entre equipos, pues ofrece un formato de datos universal que permite adaptar o transformar fácilmente la información.

Ya existen muchas herramientas de programación preparadas para trabajar con formatos basados en XML. Es una tendencia y se está consolidando. Es ya un formato universal, que lo están adoptando multitud de programadores y que puede abarcar cualquier caso de intercambio de información.

XML sirve para que muchos programas interpreten bien cualquier tipo de dato. No solo eso. XML sirve para que algunos programas hablen entre ellos sin intervención humana. La Computación Distribuida, Interoperabilidad, Monitorización, son situaciones en las que resulta imprescindible este tipo de comunicación. XML es también la solución en estos casos. Los Servicios Web son un caso particular de "Computación Distribuida" y XML es su lenguaje de base.(VILLATE)

Resumiendo los puntos más significativos del XML como lenguaje de desarrollo de aplicaciones:

- XML será probablemente la principal tecnología para ofrecer funcionalidad avanzada a las aplicaciones de Internet, Extranets e Intranets. En este sentido permite ofrecer información actualizada y segura, con independencia del medio de conexión, ordenadores de sobremesa, PDA, teléfono WAP, etc.
- Se está convirtiendo en el formato estándar de intercambio de datos, permitiendo la integración de datos entre diversas aplicaciones y bases de datos.
- Reemplazará a HTML en aplicaciones Web donde se requieran grados elevados de reutilización, intercambio de datos, automatización e interacción con otras aplicaciones externas e internas, así como en la relación con otras personas y entidades.

Los puntos anteriores son debidos a las características y ventajas que tiene sobre lenguajes alternativos, entre las que se destacan:

- XML muestra el significado y las relaciones de la información contenida (en documentos y bases de datos), lo que permite gestionar y manejar datos, tanto estructurados como no estructurados.
- Es un formato para intercambiar y compartir información en bases de datos, documentos y aplicaciones.
- Es extensible y por lo tanto adaptable tanto a necesidades futuras, en términos generales como a necesidades específicas de una empresa o de una relación comercial en concreto.
- Utilización independiente del mecanismo de acceso a datos, transacciones e interacciones.

Las aplicaciones confeccionadas con XML son aplicaciones Web nativas, fáciles de gestionar, mantener y actualizar. Sus componentes permiten un elevado grado de reutilización, por lo que son más baratas de desarrollar que las convencionales.

2.5.2 Servicio Web

Los servicios Web XML permiten que las aplicaciones compartan información y que además invoquen funciones de otras aplicaciones independientes de cómo se hayan creado las aplicaciones, cuál sea el sistema operativo o la plataforma en que se ejecutan y cuáles los dispositivos utilizados para obtener acceso a ellas. Aunque los servicios Web XML son independientes entre sí, pueden vincularse y formar grupos de colaboración para realizar tareas determinadas.

¿Para qué sirve un Servicio Web? La respuesta puede ser otra pregunta: ¿Para qué sirve en programación una rutina? Es conocido que una rutina es como una caja negra, que encierra cierto proceso o algoritmo, y que cumple una función clara. Muchas rutinas y un guión central componen un programa en lo que se llama "programación estructurada". Un Servicio Web viene a ser una rutina en Internet.

¿Por qué se llama "Servicio Web" y no "Rutina en Internet"? Los protocolos que soportan los Servicio Web se comunican normalmente por el puerto 80 y se basan en HTTP, métodos GET y PUT. Esto permite que se pueda acceder a ellos al igual que en una página Web. La diferencia entre una página Web y un Servicio Web, es que la página la visita cualquier individuo interesado, mientras que el servicio sólo lo visitan programas que lo requieren.

De modo, que el conjunto de Servicio Web en Internet es una World Wide Web paralela, de carácter no humano, sino cibernético.

Los Servicios Web se actualizan de forma transparente para el programador y para el encargado de mantenimiento de la aplicación. Además, mediante un Servicio Web se puede implementar a programas funciones imposibles de contemplar bajo el uso de rutinas de librerías, como por ejemplo, incorporar un buscador de páginas Web. Por otro lado, la carga de CPU que supone la ejecución de una rutina, desaparece al usar Servicios Web. La carga se reparte por Internet, sobre el servidor del Servicio Web. Esto es un comienzo de "Computación Distribuida".

Los Servicios Web no pretenden eliminar del mapa a las librerías o módulos de programación, que de toda la vida se han utilizado. No pueden hacerlo, ya que no son una versión mejorada de éstas, sino una herramienta con distintas aplicaciones en determinados casos.

Los Servicios Web permiten a los usuarios usar aplicaciones que comparten datos con otros programas modulares. Son aplicaciones independientes de la plataforma que pueden ser fácilmente publicadas, localizadas e invocadas mediante protocolos Web estándar, como XML. El objetivo final es la creación de un directorio online de Servicios Web, que pueda ser localizado de un modo sencillo y que tenga una alta fiabilidad. La integración de aplicaciones hará posible obtener información demandada en tiempo real, acelerando el proceso de toma de decisiones.

2.6 Herramientas a utilizar

Para el buen desarrollo de una aplicación se requiere del empleo de herramientas que posibiliten el modelado del diseño, otras para la implementación y un Sistema de Gestión de Bases de Datos.

2.6.1 Herramienta de modelado

Las herramientas de modelado de objetos, son fundamentales para el análisis del sistema. En el proceso de desarrollo de software de la solución propuesta se utilizó el Rational Rose Enterprise Edition.

Rational Rose es la herramienta Case desarrollada por los creadores de UML que cubren todo el ciclo de vida de un proyecto: concepción y formalización del modelo, construcción de los componentes y certificación de las distintas fases. Permite una trazabilidad real entre modelo (análisis y diseño) y el código ejecutable.

Rational Rose domina el mercado de herramientas para el análisis, modelado, diseño y construcción orientada a objetos, tiene todas las características que los desarrolladores, analistas, y arquitectos exigen, soporte UML incomparable, desarrollo basado en componentes con soporte para arquitecturas líderes en la industria y modelos de componentes, facilidad de uso e integración optimizada. (MENÉNDEZ 2003)

La corporación Rational Rose ofrece el Proceso Unificado de Rational (RUP), que unifica las mejores prácticas de muchas disciplinas en un consistente y completo proceso del ciclo de vida, que permite al equipo de desarrollo disminuir los tiempos de liberación, además de hacer más predecible el software que ellos producen. Este proceso está basado en el Lenguaje Unificado de Modelación (UML estándar de la industria) y únicamente integrado a herramientas líderes en el desarrollo de software de Rational, el Proceso Unificado de Rational apoya el equipo completo de desarrollo de software con guías detalladas e información crítica aplicable a la mayoría de las aplicaciones de la industria.

Características principales:

- Admite como notaciones: UML, COM, OMT y Booch.
- Realiza Chequeo semántico de los modelos.
- Ingeniería “de ida y vuelta”: Rose permite generar código a partir de modelos y viceversa.
- Desarrollo multiusuario.
- Integración con modelado de datos.
- Generación de documentación.
- Tiene un lenguaje de script para poder ampliar su funcionalidad.
- Disponible en múltiples plataformas.

2.6.2 Herramientas de desarrollo

Para el desarrollo de esta aplicación se utilizaron como herramientas desarrolladoras el Visual Studio 2005 para su implementación y el SQL Server 2000 como Sistema de Gestión de Bases de Datos.

¿Qué es un Sistema de Gestión de Bases de Datos (SGBD)?

Un Sistema de Gestión de Bases de Datos (SGBD) consiste en un conjunto de programas, procedimientos y lenguajes que proporcionan las herramientas necesarias para trabajar con una base de datos. Incorporar una serie de funciones que permita definir los registros, sus campos, sus relaciones, insertar, suprimir, modificar y consultar los datos. (DATE 2001)

Para comunicarse con un SGBD, tanto para definir datos y estructuras como para hacer consultas sobre los datos se puede utilizar SQL (*Structured Query Language*), que no es más que un lenguaje de consultas estructurado compuesto por comandos, cláusulas, operadores y funciones de agregado 13 . Estos elementos se combinan en grupos de instrucciones (consultas) para actualizar y manipular las bases de datos. Su forma más común de utilización es a través de un lenguaje de programación anfitrión como, por ejemplo, los lenguajes de programación en la Web del lado del Servidor.(MARTIN 2001)

Entre los SGBD comúnmente utilizados en el mundo están Oracle, MySQL, Microsoft SQL Server, PostgreSQL, InterBase, entre otros. Todos estos presentan un enfoque relacional con un buen basamento matemático centrado en el Álgebra Relacional.

SQL Server 2000

Una solución integral de base de datos y análisis, SQL Server 2000 ofrece el rendimiento, escalabilidad y confiabilidad que requieren el exigente Web y los entornos empresariales de línea de negocios. El nuevo soporte de XML y HTTP simplifica el acceso a datos y el intercambio, mientras que las poderosas capacidades de análisis mejoran el valor de los datos. Las características de disponibilidad mejoradas maximizan el tiempo de actividad, las funciones de la administración avanzadas automatizan las tareas rutinarias y las herramientas mejoradas de programación y los servicios aceleran el desarrollo.(J GROFF 1994)

SQL Server 2000 es el eje principal de la administración y análisis de datos de la siguiente generación de productos y servicios de Microsoft .NET.

- Simplifica la integración de sistemas del back-end y la transferencia de datos por medio de firewalls que usan la completa funcionalidad de SQL Server 2000 XML y el soporte de otros estándares de Internet, como XPath, XSL y XSLT. Los desarrolladores de Web pueden acceder a los datos usando XML sin la programación de base de datos relacional, mientras que los administradores de bases de datos pueden manipular los datos fácilmente en formato XML usando Transact-SQL⁷ (T-SQL) y los procedimientos almacenados.
- Conecta las bases de datos de SQL Server 2000 y cubos de OLAP usando el Web flexiblemente sin la programación adicional. Usa la segura conectividad de bases de datos en HTTP para asegurar que se puedan consultar los datos a través de un URL y las interfaces de usuario intuitivas.
- Habilita para el Web e integra los sistemas de negocios existentes a través del penetrante soporte de XML.
- Aprovecha por completo sus recursos de hardware ejecutando aplicaciones aisladas múltiples en una sola computadora que use el soporte de instancias múltiples de SQL Server 2000.
- Entrega rápidamente aplicaciones de bases de datos robustas y escalables que usan las herramientas mejoradas de desarrollo de SQL Server 2000. El Analizador de Consultas de SQL ahora incluye un depurador de procedimientos almacenados. Los Data Transformation Services (DTS) extienden la capacidad para mover y transformar los datos desde cualquier fuente.
- Mejora la productividad con las mejoras de T-SQL. Las nuevas funciones definidas por el usuario permiten la reutilización de código y simplifican el desarrollo. Mantiene datos consistentes con la integridad referencial para controlar la propagación en operaciones de eliminación y actualización.

Visual Studio 2005

Visual Studio 2005 proporciona una amplia gama de herramientas que ofrecen multitud de ventajas para desarrolladores individuales y equipos de desarrollo de software. Se selecciona Microsoft Visual Studio 2005 porque:

⁷ Transact-SQL: Ver Glosario de términos.

- Tiene mayor productividad y obtención más rápida de resultados.
- Crea soluciones dinámicas basadas en Windows, la Web, dispositivos móviles y Office.
- Tiene comunicación y colaboración más eficaz en sus equipos de software.
- Garantiza calidad rápida y continua en todo el proceso de desarrollo.
- Hay más ediciones diferenciadas por el precio y las características.
- Ayuda con refactorización.
- Añadido soporte de tests para todo tipo de aplicaciones.

2.7 Lenguajes de programación

En la implementación de la sistema propuesto se utiliza la tecnología orientada a objetos. Como lenguajes de programación se emplean C# y SQL teniendo en cuenta como aspecto fundamental la solicitud realizada por el cliente.

2.7.1 Lenguaje de programación C#

C# es el nuevo lenguaje de propósito general orientado a objetos creado por Microsoft para su nueva plataforma .NET. Combina la potencia de lenguajes como C++ con la sencillez de lenguajes como Visual Basic, y que además la migración a este lenguaje por los programadores de C/C++/Java sea lo más inmediata posible.

Aunque es posible escribir código para la plataforma .NET en muchos otros lenguajes, C# es el único que ha sido diseñado específicamente para ser utilizado en ella, por lo que programarla usando C# es mucho más sencillo e intuitivo que hacerlo con cualquiera de los otros lenguajes ya que C# carece de elementos heredados innecesarios en .NET. Por esta razón, se suele decir que C# es el lenguaje nativo de .Net

Características de C#.

Sencillez: C# elimina muchos elementos que otros lenguajes incluyen y que son innecesarios en .Net

Por ejemplo:

- El código escrito en C# es autocontenido, lo que significa que no necesita de ficheros adicionales al propio fuente tales como ficheros de cabecera.
- El tamaño de los tipos de datos básicos es fijo e independiente del compilador, sistema operativo o máquina para quienes se compile.
- No se incluyen elementos poco útiles de lenguajes como C++ tales como macros, herencia múltiple o la necesidad de un operador diferente del punto (.) acceder a miembros de espacios de nombres (::)

Modernidad: C# incorpora en el propio lenguaje elementos que a lo largo de los años ha ido demostrándose son muy útiles para el desarrollo de aplicaciones y que en otros lenguajes como Java o C++ hay que simular, como un tipo básico decimal que permita realizar operaciones de alta precisión con reales de 128 bits y la inclusión de una instrucción foreach que permita recorrer colecciones con facilidad y es ampliable a tipos definidos por el usuario, entre otras cualidades.

Orientación a objetos: Como todo lenguaje de programación de propósito general actual, C# es un lenguaje orientado a objetos. Una diferencia de este enfoque orientado a objetos respecto al de otros lenguajes como C++ es que el de C# es más puro en tanto que no admiten ni funciones ni variables globales sino que todo el código y datos han de definirse dentro de definiciones de tipos de datos, lo que reduce problemas por conflictos de nombres y facilita la legibilidad del código.

2.7.2 SQL (Structure Query Language)

Es mucho mejor manejar datos desde bases de datos, que escribir y leer datos para/desde archivos de texto. SQL nos brinda esa posibilidad.

El SQL (Structured Query Language), es un lenguaje de consulta estructurado, es un lenguaje surgido de un proyecto de investigación de IBM para el acceso a bases de datos relacionales. Actualmente se ha convertido en un estándar de lenguaje de bases de datos, y la mayoría de los sistemas de bases de datos lo soportan, desde sistemas para ordenadores personales, hasta grandes ordenadores.

SQL es un lenguaje para describir todas las acciones u operaciones que se realizan sobre los motores de las bases de datos relacionales. Se realizan para consultar y programar bases de datos, acceder a los datos y para consultar, actualizar y gestionar sistemas de bases de datos. Se divide en dos secciones:

1. Las DDL: (Data Description Language), lenguaje de definición de datos, incluye órdenes para definir, modificar o borrar las tablas en las que se almacenan los datos y de las relaciones entre estas. (Es el que más varía de un sistema a otro). Normalmente las instrucciones incluidas son CREATE, ALTER, DROP aplicadas sobre diferentes objetos.

2. Las DML: (Data Manipulation Language), lenguaje de manipulación de datos, nos permite recuperar los datos almacenados en la base de datos y también incluye órdenes para permitir al usuario actualizar la base de datos añadiendo nuevos datos, suprimiendo datos antiguos o modificando datos previamente almacenados. La manipulación se hace a nivel de tuplas (filas).

Algunas de las características del SQL son:

- Es una forma estándar de consulta de datos específicos.
- Es una forma de extraer y manipular datos de una base de datos.
- Usado para todas las funciones de bases de datos, incluyendo administración.
- Creación de esquemas y datos recuperables.
- Puede ser usado de forma implícita dentro de una aplicación.

2.8 Conclusiones

En este capítulo se realizó un análisis de las tecnologías que a utilizar para el desarrollo del sistema propuesto. Además se relacionan algunos conceptos asociados al tema que esclarecen las explicaciones y elecciones realizadas.

Se fundamenta la metodología de desarrollo de software a utilizar, el lenguaje de programación, el sistema gestor de bases de datos que fueron especificados por el cliente, así como otras tecnologías a emplear y las principales herramientas de trabajo de las cuales se hace uso, brindando para cada elección una explicación detallada de la misma.

Capítulo 3

Presentación de la solución propuesta

3.1 Introducción

En este Capítulo se realiza la descripción de la solución propuesta en el trabajo. Se realiza una breve explicación de algunos conceptos importantes relacionados con el Modelo del Negocio. Además se listan los requisitos funcionales y no funcionales del sistema propuesto de manera que permita una visión más completa del mismo. También se presenta un conjunto de diagramas que serán los encargados de ilustrar el modelo del negocio realizado, entre los cuales podemos mencionar los diagramas de casos de uso y los diagramas de actividades correspondientes a cada caso de uso del negocio.

3.2 Modelación del negocio

Las Comisiones Provinciales y Nacionales del béisbol en Cuba son las encargadas de organizar los eventos competitivos relacionados con este deporte. El INDER cuenta actualmente con una aplicación a través de la cual realiza todas las operaciones relacionadas con la gestión de los eventos competitivos y la anotación del juego de béisbol.

Aunque se encuentra funcionando correctamente, esta aplicación cuenta con algunos inconvenientes que ya han sido mencionados en este trabajo con anterioridad. Con el objetivo de implementar una aplicación menos rígida, más usable, funcional y accesible, se desarrolló el Modelo del Negocio que se presenta a continuación, a través del cual se pretende ilustrar detalladamente cómo tienen lugar las acciones en el negocio y las respuestas que se brindan a dichas acciones.

Antes de comenzar el evento competitivo las Comisiones Provinciales se encargan de realizar los *rosters*⁸ de los equipos de cada provincia y de seleccionar los estadios que poseen condiciones para que se

⁸ Rosters: Ver Glosario de términos.

desarrollen partidos en ellos. Luego envían esta información a la Comisión Nacional quien la revisa nuevamente y la aprueba en caso de estar de acuerdo o la devuelve a las Comisiones Provinciales para que realicen algunos cambios requeridos. Una vez recibidos todos los rosters y estadios la Comisión Nacional crea el libro del evento⁹ y envía copias de este a cada una de las Comisiones Provinciales.

Durante el período en que se desarrolla el evento, las Comisiones Provinciales tienen derecho a dar altas y bajas a los jugadores de sus equipos debido a determinadas situaciones que se presentan. Para ello deben informar a la Comisión Nacional el trámite que desean realizar y enviar los datos. La Comisión Nacional los revisa y si no hay inconvenientes los aprueba, actualiza el libro del evento y envía la actualización a cada Comisión Provincial; en caso de no estar de acuerdo con el trámite solicitado lo informa a la Comisión Provincial que lo solicitó.

Otra de las tareas consiste en la anotación del juego. Llegado el día y la hora planificados para el desarrollo del partido, el anotador oficial del encuentro busca las alineaciones¹⁰ de ambos equipos y las registra en la hoja de anotación, así como también registrará el nombre del estadio y los jueces participantes en el encuentro. Además se encargará de anotar en la hoja todas las jugadas que ocurran durante el partido y una vez terminado el mismo envía la hoja de anotación a la Comisión Nacional.

3.2.1 Determinación y justificación de los actores del negocio

Actores del Negocio	Justificación
Comisión Provincial	Es la organización encargada de hacer llegar el roster de su equipo y los estadios que se encuentran listos para celebrar encuentros a la Comisión Nacional para que esta cree el libro del evento. También es la encargada de emitir Altas y Bajas en el equipo durante el evento.
Reloj	Es el encargado de iniciar la anotación del juego al llegar la fecha y hora programada en el calendario de juego.

Tabla 3.1 Determinación y justificación de los actores del negocio.

⁹ Libro del evento: Ver Epígrafe “Conceptos asociados al dominio del problema” del Capítulo 1.

¹⁰ Alineaciones: Ver Glosario de términos.

3.2.2 Determinación y justificación de los trabajadores del negocio

Trabajadores del Negocio	Justificación
Comisión Nacional	Es la organización encargada de recibir los rosters y estadios enviados por las comisiones provinciales, si los aprueba crea el libro del evento, en caso de no aprobarlo lo informa a las comisiones provinciales. También es la encargada de aprobar las Altas o Bajas solicitada por las comisiones provinciales si la aprueba se encarga de asentarlos en el libro del evento, en caso contrario lo informa a las comisiones provinciales.
Anotador	Es la persona encargada cuando llega el día y la hora del juego de buscar las alineaciones de ambos equipos y llevarlas a la hoja de anotación; así como el nombre del estadio donde se va a jugar, jueces participantes, etc. Además asentará en la hoja de anotación todos los sucesos del encuentro y al finalizar el juego se encargará de enviarla a la Comisión Nacional.

Tabla 3.2 Determinación y justificación de los trabajadores del negocio.

3.2.3 Diagrama de casos de uso del negocio

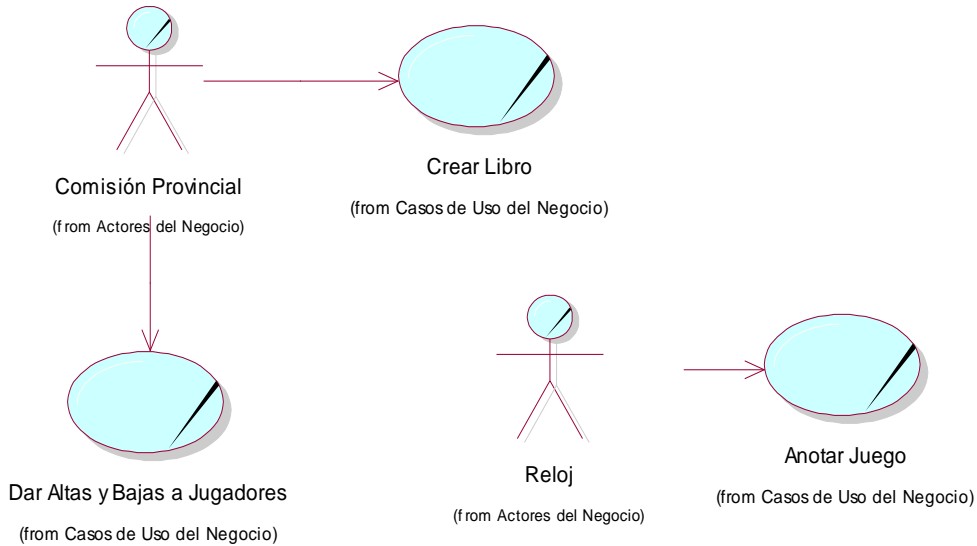


Figura 3.1 Diagrama de casos de uso del negocio.

3.2.4 Descripción de los casos de uso del negocio

Nombre del Caso de Uso		Crear libro
Actores		Comisión Provincial(inicia)
Propósito	Crear el libro del evento que se va a desarrollar.	
Resumen	El caso de uso comienza cuando la Comisión Provincial envía el roster del equipo y los estadios que se encuentran listos para celebrar encuentros. La Comisión Nacional analiza el roster, si no está de acuerdo lo informa a la Comisión Provincial para que esta lo modifique y reenvíe. Cuando la Comisión Nacional aprueba el roster crea el libro del evento, donde se registran todas las informaciones del mismo.	
Curso Normal de los eventos		
Acciones del Actor		Respuesta del proceso de negocio

1. La Comisión Provincial envía el roster de su equipo y los estadios con condiciones para efectuar encuentros.	1.1 La Comisión Nacional analiza la información.
	1.2 La Comisión Nacional no detecta problemas y aprueba la información.
	1.3 La Comisión Nacional registra la información en el libro del evento y añade los jueces participantes, el calendario de competencia y el reglamento del mismo.
	1.4 La Comisión Nacional envía una copia del libro a la Comisión Provincial terminando así el caso de uso.
Curso Alternativo de los eventos	
	1.2 La Comisión Nacional detecta problemas y no aprueba la información.
	1.3 La Comisión Nacional informa del problema a la Comisión Provincial.
2. La Comisión Provincial rectifica el roster y los estadios y envía nuevamente la información a la Comisión Nacional volviendo al curso 1.1	
Prioridad	Crítico

Tabla 3.3 Descripción del caso de uso del negocio Crear libro.

Nombre del Caso de Uso		Dar Altas y Bajas a Jugadores
Actores		Comisión Provincial(inicia)
Propósito	Dar Altas y Bajas a lo largo del evento en los diferentes equipos participantes.	
Resumen	El caso de uso comienza cuando la Comisión Provincial informa a la Comisión Nacional que desea dar Alta o Baja a algún jugador y entrega los datos del mismo. La Comisión Nacional analiza la solicitud y si no está de acuerdo le informa a la Comisión Provincial que el Alta o Baja fue rechazada. En caso de aprobarse la solicitud, la Comisión Nacional registra los cambios en el libro del evento y envía la actualización a la Comisión Provincial.	
Curso Normal de los eventos		
Acciones del Actor	Respuesta del proceso de negocio	
1. La Comisión Provincial informa a la Comisión Nacional que desea realizar un Alta o Baja en su equipo y ofrece los datos.	1.1 La Comisión Nacional analiza la solicitud.	
	1.2 La Comisión Nacional aprueba el Alta o Baja solicitada.	
	1.3 La Comisión Nacional registra la los cambios en el libro del evento.	
	1.4 La Comisión Nacional envía los cambios realizados a la Comisión Provincial terminando así el caso de uso.	
Curso Alternativo de los eventos		
	1.2 La Comisión Nacional no aprueba el Alta o Baja solicitada.	
	1.3 La Comisión Nacional informa a la Comisión	

		Provincial que su solicitud fue rechazada terminando así el caso de uso.
Prioridad	Crítico	

Tabla 3.4 Descripción del caso de uso del negocio Dar Altas y Bajas.

Nombre del Caso de Uso		Anotar juego
Actores		Reloj(inicia)
Propósito	Realizar la anotación del encuentro programado.	
Resumen	El caso de uso comienza cuando llega la fecha y hora señalados para el encuentro. El anotador oficial del partido busca las alineaciones de ambos equipo y las registra en la hoja de anotación e incluye además el nombre del estadio y los jueces participantes. Durante el encuentro el anotador lleva a la hoja de anotación todas las jugadas que ocurren en el mismo y una vez finalizado envía la hoja de anotación a la Comisión Nacional.	
Curso Normal de los eventos		
Acciones del Actor		Respuesta del proceso de negocio
1. Llega la fecha y hora señalados para el encuentro.		1.1 El anotador oficial del encuentro busca las alineaciones de ambos equipos.
		1.2 El anotador oficial del encuentro registra en la hoja de anotación las alineaciones, nombre del estadio y jueces que participarán en el encuentro.

	1.3 El anotador oficial del encuentro realiza la anotación de las jugadas ocurridas durante todo el partido.
	1.4 Una vez terminado el encuentro el anotador oficial del encuentro envía la hoja de anotación a la Comisión Nacional.
Prioridad	Crítico

Tabla 3.5 Descripción del caso de uso del negocio Anotar juego.

3.2.5 Diagramas de actividades de los casos de uso del negocio

En el presente subepígrafe se realizaron los diagramas de actividades correspondientes a los tres casos de uso del negocio. Para ver el contenido de los mismos remítase al Anexo 1. Diagramas de actividades de los casos de uso del negocio.

3.2.6 Diagrama del modelo de objetos del negocio

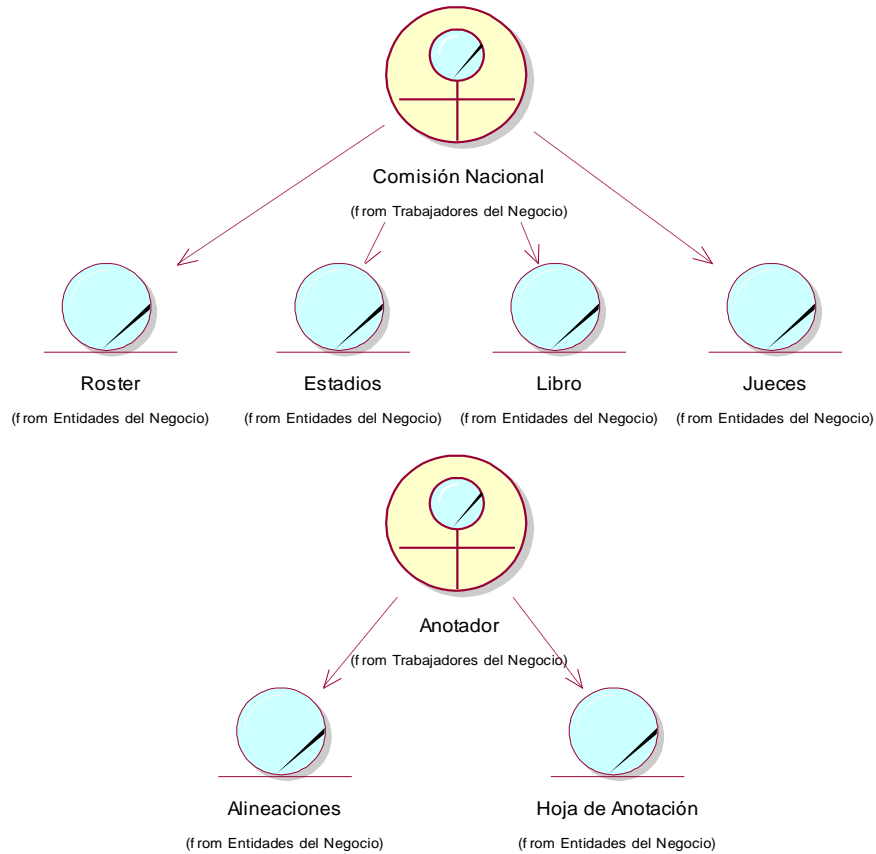


Figura 3.5 Diagrama del modelo de objetos del negocio.

3.3 Especificación de requisitos

La especificación de requisitos es una de las tareas más importantes en el proceso de desarrollo de un software.

Un requisito es una condición o capacidad que tiene que tener un sistema para satisfacer un contrato o documento formal. Estos se clasifican en funcionales y no funcionales. Los requisitos representan capacidades o funciones que debe poseer el sistema a implementar para cumplir con las expectativas del

cliente, mientras que los requisitos no funcionales constituyen propiedades o cualidades que el producto debe tener.(D GAUSE 1989)

A continuación se realiza el levantamiento de requisitos correspondiente al sistema propuesto.

3.3.1 Requerimientos funcionales

R1. Gestionar Conexión

1.1 El sistema debe permitir configurar la conexión con el Web Service

- Servidor Web

1.2 El sistema debe permitir guardar Conexión en fichero local

- Servidor Web

R2. Autenticarse

- Usuario
- Contraseña

R3. Gestionar Usuarios

3.1 Agregar Usuario

- Nombre
- Primer Apellido
- Segundo Apellido
- Usuario
- Contraseña
- Rol de Usuario

3.2 Modificar Usuario

- Nombre
- Primer Apellido
- Segundo Apellido
- Usuario

- Contraseña
- Rol de Usuario

3.3 Eliminar Usuario

- Usuario

3.4 Mostrar Usuario

- Nombre
- Primer Apellido
- Segundo Apellido
- Usuario
- Rol de Usuario

R4. Gestionar Jugadores

4.1 Insertar Jugador

- Nombre
- Primer Apellido
- Segundo Apellido
- Fecha Nacimiento
- Lugar Nacimiento
- Mano Batear
- Mano Lanzar
- Peso
- Altura
- Reseña Histórica

4.2 Actualizar Jugador

- Nombre
- Primer Apellido
- Segundo Apellido
- Fecha Nacimiento
- Lugar Nacimiento

- Mano Batear
- Mano Lanzar
- Peso
- Altura
- Reseña Histórica
- Activo

4.3 Eliminar Jugador

- Jugador

4.4 Mostrar Jugadores

- Nombre
- Primer Apellido
- Segundo Apellido
- Fecha Nacimiento
- Lugar Nacimiento
- Mano Batear
- Mano Lanzar
- Peso
- Altura
- Reseña Histórica
- Activo

R5. Gestionar Equipos

5.1 Agregar Equipo

- Nombre Equipo
- Siglas del Equipo (tres caracteres)

5.2 Actualizar Equipo

- Nombre Equipo
- Siglas del Equipo(tres caracteres)

5.3 Eliminar Equipo

- Equipo

5.4 Mostrar Equipo

- Nombre Equipo
- Siglas del Equipo(tres caracteres)

R6. Gestionar Estadios

6.1 Agregar Estadio

- Nombre
- Capacidad
- Municipio
- Descripción

6.2 Actualizar Estadio

- Nombre
- Capacidad
- Municipio
- Descripción

6.3 Eliminar Estadio

- Estadio

6.4 Mostrar Estadio

- Nombre
- Capacidad
- Municipio
- Descripción

R7. Gestionar Jueces (Anotadores, Árbitros y Comisionados)

7.1 Agregar Juez

- Nombre
- Primer Apellido
- Segundo Apellido

- Provincia
- Tipo(Anotadores, Árbitros y Comisionados)

7.2 Actualizar Juez

- Nombre
- Primer Apellido
- Segundo Apellido
- Provincia
- Tipo(Anotadores, Árbitros y Comisionados)

7.3 Eliminar Juez

- Juez

7.4 Mostrar Juez

- Nombre
- Primer Apellido
- Segundo Apellido
- Provincia
- Tipo(Anotadores, Árbitros y Comisionados)

R8. Gestionar Directivos

8.1 Agregar Directivo

- Nombre
- Primer Apellido
- Segundo Apellido
- Nivel Cultural

8.2 Actualizar Directivo

- Nombre
- Primer Apellido
- Segundo Apellido
- Nivel Cultural

8.3 Eliminar Directivo

- Directivo

8.4 Mostrar Directivos

- Nombre
- Primer Apellido
- Segundo Apellido
- Nivel Cultural

R9. Gestionar Eventos

9.1 Agregar Evento

- Nombre del Evento
- Notación Oficial
- Cantidad de Juegos
- Fecha Inicio
- Fecha Fin
- Cantidad de Inning de los Juegos
- Lanzador batea? (bool)

9.2 Actualizar Evento

- Nombre del Evento
- Notación Oficial
- Cantidad de Juegos
- Fecha Inicio
- Fecha Fin
- Cantidad de Inning de los Juegos
- Lanzador batea? (bool)

9.3 Eliminar Evento

- Evento

9.4 Mostrar Evento

- Nombre del Evento

- Notación Oficial
- Cantidad de Juegos
- Fecha Inicio
- Fecha Fin
- Cantidad de Inning de los Juegos
- Lanzador batea? (bool)

R10. Gestionar Equipo a Evento

10.1 Agregar Equipo a Evento

- Equipo
- Evento

10.2 Actualizar Equipo a Evento

- Equipo
- Evento

10.3 Eliminar Equipo a Evento

- Equipo
- Evento

10.4 Mostrar Equipos del Eventos

- Equipo
- Evento

R11. Gestionar Jugador a Equipo

11.1 Agregar Jugador a Equipo

- Evento
- Equipo
- Jugador
- Numero (número 1-99)
- Posición
- Novato

11.2 Actualizar Jugador a Equipo

- Evento
- Equipo
- Jugador
- Numero(número 1-99)
- Posición
- Novato

11.3 Eliminar Jugador a Equipo

- Jugador Equipo

11.4 Mostrar Jugadores de Equipos

- Evento
- Equipo
- Jugador
- Número(número 1-99)
- Posición
- Novato

R12. Gestionar Directivos a Equipo

12.1 Agregar Directivo a Equipo

- Evento
- Equipo
- Directivo
- Cargo Dirección
- Número (número 1-99)

12.2 Actualizar Directivo a Evento

- Evento
- Equipo
- Directivo
- Cargo Dirección

- Número (número 1-99)

12.3 Eliminar Directivo a Evento

- Directivo Evento

12.4 Mostrar Directivos de Equipo

- Evento
- Equipo
- Directivo
- Cargo Dirección
- Número (número 1-99)

R13. Gestionar Juegos

13.1 Agregar Juego

- Evento
- Numero del Juego
- Fecha
- Equipo Home Club
- Equipo Visitador
- Hora Comienzo
- Hora Terminado
- Estadio
- Descripción
- Etapa del Juego(Etapas de esa serie creada)

13.2 Actualizar Juego a Evento

- Evento
- Numero del Juego
- Fecha
- Equipo Home Club
- Equipo Visitador
- Hora Comienzo

- Hora Terminado
- Estadio
- Descripción
- Etapa del Juego(Etapas de esa serie creada)

13.3 Eliminar Juego a Evento

- Juego Evento

13.4 Mostrar Juegos de Eventos

- Evento
- Numero del Juego
- Fecha
- Equipo Home Club
- Equipo Visitador
- Hora Comienzo
- Hora Terminado
- Estadio
- Descripción
- Etapa del Juego(Etapas de esa serie creada)

R14. Gestionar Cargo Dirección

14.1 Agregar Cargo Dirección

- Cargo Dirección

14.2 Actualizar Cargo Dirección

- Cargo Dirección

14.3 Eliminar Cargo Dirección

- Cargo Dirección

14.4 Mostrar Cargo Dirección

- Cargo Dirección

R15. Gestionar Posición Jugador

15.1 Agregar Posición Jugador

- Posición

- Sigla

15.2 Actualizar Posición Jugador

- Posición

- Sigla

15.3 Eliminar Posición Jugador

- Posición

15.4 Mostrar Posición Jugador

- Posición

- Sigla

R16. Gestionar Posición Juego

16.1 Agregar Posición Juego

- Posición

- Sigla

16.2 Actualizar Posición Juego

- Posición

- Sigla

16.3 Eliminar Posición Juego

- Posición

16.4 Mostrar Posición Juego

- Posición

- Sigla

R17 Mostrar Juegos del Día

- Numero de Juego

- Equipo Visitador

- Equipo Home Club

- Fecha del Juego

- Hora del Juego
- Estadio

R18 Mostrar Juegos del Evento

- Numero de Juego
- Equipo Visitador
- Equipo Home Club
- Fecha del Juego
- Hora del Juego
- Estadio

R19. Gestionar Juez del Juego

19.1 Agregar Juez al Juego

- Evento
- Juego
- Juez
- Posición Juez

19.2 Actualizar Juez del Juego

- Evento
- Juego
- Juez
- Posición Juez

19.3 Eliminar Juez del Juego

- Juez del Juego

19.4 Mostrar Jueces del Juego

- Evento
- Juego
- Juez
- Posición Juez

R20. Gestionar Posición de los Jueces

20.1 Agregar Posición de los Jueces

- Posición
- Sigla

20.2 Actualizar Posición de los Jueces

- Posición
- Sigla

20.3 Eliminar Posición Jueces

- Posición Juez

R21. Gestionar Anotadores del juego

21.1 Agregar Anotador del Juego

- Evento
- Juego
- Anotador

21.2 Actualizar Anotador del Juego

- Evento
- Juego
- Anotador

21.3 Eliminar Anotador del Juego

- Anotador Juego

21.4 Mostrar Anotadores del Juego

- Evento
- Juego
- Anotador

R22. Gestionar Alineación del Juego

22.1 Agregar Alineación del Juego

- Evento
- Juego
- Equipo
- Jugador
- Posición

22.2 Modificar Alineación del Juego

- Evento
- Juego
- Equipo
- Jugador
- Posición

22.3 Eliminar Alineación del Juego

- Alineación Juego

22.4 Mostrar Alineación del Juego

- Evento
- Juego
- Equipo
- Jugador
- Posición

R23. Gestionar Jugadas del Juego

23.1 Agregar Jugadas del juego

- Evento
- Juego
- Equipo
- Camiseta del Jugador
- Casilla2
- Casilla3
- Casilla4

- Casilla5
- Casilla6

23.2 Actualizar Jugadas del juego

- Evento
- Juego
- Equipo
- Camiseta del Jugador
- Casilla2
- Casilla3
- Casilla4
- Casilla5
- Casilla6

23.3 Eliminar Jugadas del juego

- Jugada

23.4 Mostrar Jugadas

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

R24. Gestionar Lanzamientos

24.1 Agregar Lanzamiento

- Evento
- Juego
- Equipo
- Camiseta del Jugador
- Zona Lanzamiento
- Ocurrencia
- Velocidad

- Lanzamiento

24.2 Modificar Lanzamiento

- Evento
- Juego
- Equipo
- Camiseta del Jugador
- Zona Lanzamiento
- Ocurrencia
- Velocidad
- Lanzamiento

24.3 Eliminar Lanzamiento

- Lanzamiento

24.4 Mostrar Lanzamientos

- Evento
- Juego
- Equipo
- Camiseta del Jugador
- Zona Lanzamiento
- Ocurrencia
- Velocidad
- Lanzamiento

R25. Gestionar Países

25.1 Agregar País

- Nombre del país

25.2 Modificar País

- Nombre del país

25.3 Eliminar País

- País

25.4 Mostrar País

- Nombre del País

R26. Gestionar Provincia

26.1 Agregar Provincia

- País
- Nombre de la Provincia

26.2 Modificar Provincia

- Nombre de la Provincia

26.3 Eliminar Provincia

- Provincia

26.4 Mostrar Provincia

- Nombre de la provincia

R27. Gestionar Municipio

27.1 Agregar Municipio

- País
- Provincia
- Nombre del Municipio

27.2 Modificar Municipio

- Nombre del Municipio

27.3 Eliminar Municipio

- Municipio

27.4 Mostrar Municipio

- Nombre del Municipio

R28. Gestionar Etapas de un Evento

28.1 Agregar Etapa

- Evento

- Nombre de la Etapa

28.2 Modificar Etapa

- Nombre de la Etapa

28.3 Eliminar Etapa

- Etapa

28.4 Mostrar Etapa

R29 Gestionar Etapas

3.3.2 Requerimientos no funcionales

Interfaz externa

1. El sistema interactúa con el usuario mediante una interfaz Windows de escritorio usable y funcional.
2. Diseño sencillo, con pocas entradas, permitiendo que no sea necesario mucho entrenamiento para utilizar el sistema.

Usabilidad

3. El sistema está concebido para el uso de personal no experto o con pocos conocimientos de informática, por tanto deberá ser práctico y fácil de usar.

Hardware

4. Requiere estar instalada en una PC Pentium, 256 Mb de RAM y se requiere un Modem de 56 Kb/s.
5. Un servidor donde va a estar el SQL Server 2000 y el Servicio Web, esta PC debe ser Pentium 4 mas de 1 GB de RAM. Con una tarjeta de red de 100Mbps.

Software

6. Se debe disponer en el servidor con Windows 2000 Server o 2000 Advanced Server. Además en las PC clientes deberá estar instalado el framework 2.0.

Portabilidad

7. La aplicación está implementada para su uso sobre plataforma Windows.

Seguridad

Integridad

8. Protección contra acciones no autorizadas o que puedan afectar la integridad de los datos.
9. Protección contra estados inconsistentes y corrupción de los datos.
10. Realizar copias de seguridad de la Base de datos periódicamente, como respaldo de la información.

Confidencialidad

11. La información manejada por el sistema deberá estar protegida de acceso no autorizado. Se logra a través de la implementación del mecanismo de autenticación del sistema.
12. Garantizar que la información sea vista únicamente por quien tiene derecho a verla. Se logrará solicitando registrarse para hacer uso del sistema estableciendo los diferentes roles de usuarios.

Encriptación

13. La información de autenticación se encriptará en el cliente para que viaje a través de la red ya encriptada. Para ello se hará uso del algoritmo de encriptación Triple DES.

Firewall

14. El servidor de la Base de Datos deberá estar protegido con la implementación de un firewall.

Confiabilidad

15. Garantía de un tratamiento adecuado de las excepciones y validación de las entradas del usuario.

3.4 Descripción del sistema propuesto

Se desea implementar un software que mejore las deficiencias existentes en el que es utilizado por el INDER actualmente. Para lograr este objetivo el software estará concebido con las siguientes características:

Se implementará de manera que al anotar el encuentro de béisbol se haga lanzamiento por lanzamiento y no jugada por jugada como se realiza hoy. Este proceso permitirá recoger un número mayor de

estadísticas.

La aplicación contaría además con una interfaz WINDOWS usable, funcional y accesible para el usuario, características que mejorarán las condiciones anteriores del software y harán más cómoda la labor de los anotadores pues para su uso no tendrán necesariamente que conocer la codificación de las jugadas para llevarlas a la aplicación, sino que esta le pedirá de manera interactiva los datos requeridos para realizar la anotación y se encargará internamente de codificar cada jugada, logrando así transparencia en el proceso.

El software deberá estar instalado en cada estadio donde se realice el encuentro deportivo, con conexión para poder acceder a la base de datos central que estará en el servidor del INDER. De esta manera se realizará la transferencia de datos automáticamente cada vez que se inserte una anotación en la aplicación. En casos de fallos con la conectividad, el software realizará de manera automática la salva de las anotaciones en ficheros .XML y una vez restablecida la conexión enviará los datos a la base de datos del INDER de manera automática. Este modo de funcionamiento permite que el anotador no tenga que preocuparse por el estado de la conexión al no verse interrumpida su labor por este tipo de afectaciones y además los encuentros serán llevados a través de la red prácticamente en tiempo real.

La Comisión Nacional tendrá permisos para crear un evento competitivo en la aplicación, para esto deberá insertar los datos correspondientes a dicha acción (nombre de los equipos, rosters, jueces del evento, estadios donde se jugará, etc.) y creará el calendario de juego. La aplicación brindará la posibilidad de conformar eventos con diferentes características; o sea que podrá o no estar dividido en grupos, puede o no variar el número de etapas que tendrá el evento así como la forma en que se jugará y cantidad de juegos por etapas. Esta flexibilidad del sistema permite su utilización para cualquier evento relacionado con el béisbol que se desee desarrollar.

3.5 Modelo de casos de uso del sistema

3.5.1 Definición de los actores del sistema

Actores	Descripción
Comisión Nacional	Tiene todos los privilegios para operar en el sistema. Se encarga de gestionar las acciones relacionadas con la creación de los eventos. Puede agregar, modificar, eliminar y mostrar todos los componentes que maneja el sistema (jugadores, equipos, jueces, directivos, estadios, jugadas, lanzamientos, juegos, etc.) así como también puede hacer uso de los privilegios del anotador. Es el actor principal del sistema.
Anotador	Es el encargado de la anotación de los juegos, así como de la gestión de todos los componentes de cada juego que se efectúa (alineaciones, árbitros del partido, información del partido, etc.) limitándose sus privilegios en el sistema a estas funciones.

Tabla 3.6 Definición de los actores del sistema.

3.5.2 Jerarquía de los actores del sistema

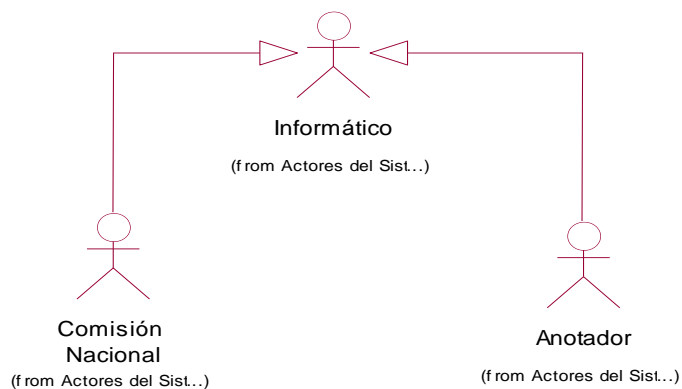


Figura 3.6 Jerarquía de los actores del sistema.

3.5.3 Relación entre paquetes de casos de uso del sistema

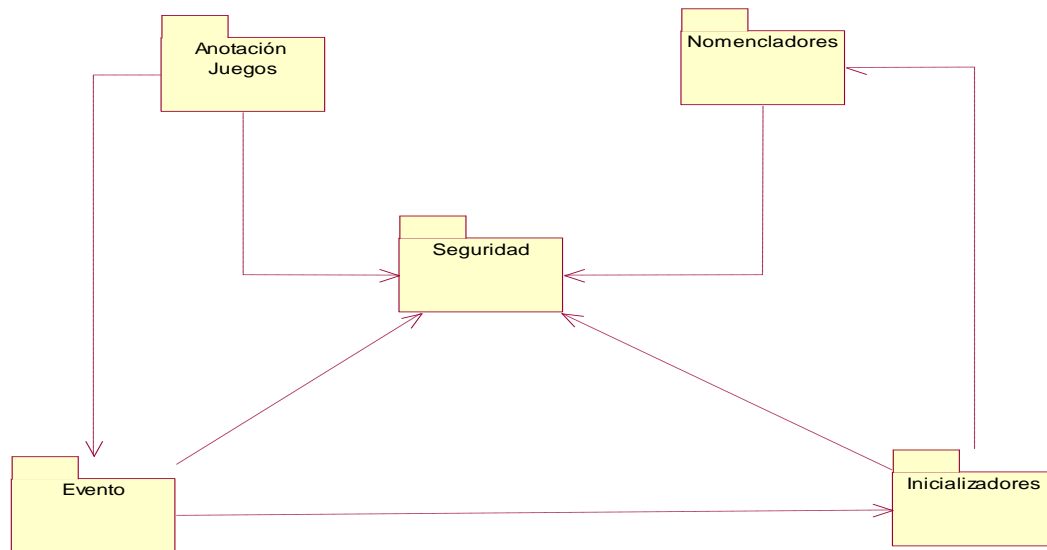


Figura 3.7 Relación entre paquetes de casos de uso del sistema.

Para el estudio del contenido de cada paquete, remítase al Anexo 2. Diagramas de casos de uso para cada paquete.

3.5.4 Descripción textual de los casos de uso del sistema

En este subepígrafe se realizó la descripción textual de los casos del sistema. Para profundizar en esta tarea remítase al Anexo 3. Descripción textual de los casos de uso del sistema.

3.6 Conclusiones

Este capítulo estuvo dirigido a profundizar en el desarrollo de la propuesta de solución. Para lograr los objetivos perseguidos se realizó la modelación del negocio, el levantamiento de requisitos y la modelación del sistema. En el capítulo se incluye un conjunto de diagramas y tablas que hacen visible la solución propuesta, dando paso de esta manera a la construcción del sistema.

Capítulo **4**

Construcción de la solución propuesta

4.1 Introducción

Este capítulo está dedicado al diseño del sistema, en él se exponen los diagramas de clases que muestran las relaciones entre las clases involucradas en estos CU y se describen las clases fundamentales del sistema, además se describe el diseño de la base de datos y las principales tablas que en él aparecen, así como el modelo de despliegue, los patrones usados y los Estándares de Codificación que se utilizó en la implementación del sistema.

4.2 Diagramas de clases del diseño

4.2.1 Diagramas de clases para cada caso de uso

En este subepígrafe se realizaron los diagramas de clases para cada caso de uso del sistema. Para hacer uso de los mismos, remítase al Anexo 4. Diagramas de clase del diseño para cada caso de uso.

4.2.2 Diagrama de clases persistentes

En este subepígrafe se realizó el diagrama de clases persistente correspondiente al proyecto. Para hacer uso del mismo remítase al Anexo 5. Diagrama de clases persistentes.

4.3 Principios de diseño

El diseño de una aplicación constituye uno de los puntos más importantes a tratar en el proceso de desarrollo de la misma. Factores como: usabilidad, navegabilidad, diseño gráfico y distribución del contenido serán los principales responsables de que la aplicación sea bien aceptado o no por los usuarios del sistema.

Para el diseño de la interfaz de usuario de este sistema se han seguido los siguientes principios:

- Permitir al usuario acceder solamente a las opciones a las que, dado su rol, puede ejecutar.
- Permitir su utilización desde el primer momento, por cualquier persona que tenga un mínimo dominio de la informática.
- Requerir de los usuarios un mínimo esfuerzo para alcanzar sus objetivos.

4.3.1 Estándares de la interfaz

Para la interfaz de la aplicación se sigue la línea clásica de las aplicaciones de escritorio para Windows. Se presentan al usuario dos interfaces, una para la anotación de los juegos y otra para la gestión de los eventos competitivos. Ambas presentan diseño y características similares, de forma que manifieste claramente la dependencia o conectividad existente entre ambas.

Se hace uso de los componentes que ofrece el Visual Studio 2005 para el desarrollo de aplicaciones de escritorio como por ejemplo: Textbox, Label, Button, Table, Menu, etc.

Se encuentra implementada de manera que sea mínimo el número de entradas requeridas para realizar las operaciones, de este modo se evita la existencia de múltiples errores. Además es accesible para todos los usuarios a pesar de que no posea conocimientos informáticos.

4.3.2 Concepción general de la ayuda

El sistema propuesto está concebido de manera que pueda ser utilizado tanto por personas con un amplio conocimiento de la informática como por aquellas que su nivel de conocimiento del tema sea bajo. Por tal motivo se decidió desarrollar una ayuda capaz de ilustrar detalladamente los pasos a seguir para las diferentes tareas que puede llevar a cabo el usuario que esté haciendo uso del software.

4.3.3 Tratamientos de Excepciones

El tratamiento de errores posibilita el buen funcionamiento de una aplicación dándole una mejor apariencia ante los clientes. Para prevenir errores por parte del usuario, sólo se le brindan las opciones mínimas necesarias a la hora de efectuar cualquier operación, por ejemplo: se deshabilitan los botones, si el usuario no se ha loggeado. Una vez determinado su rol, se le habilitan las opciones a las cuales tiene acceso.

Otro tipo de error que puede ocurrir son los que no pueden ser detectados en la parte del cliente, pues ocurren internamente en la aplicación. En este caso se muestra un mensaje de error donde se especifica el tipo de error claramente.

4.4 Estilos y patrones utilizados

4.4.1 Diseño

Los patrones de diseño proponen una forma reutilizar la experiencia de los desarrolladores, para ello clasifica y describe formas de solucionar problemas que ocurren de forma frecuente en el desarrollo. Por tanto está basado en la recopilación del conocimiento de los expertos en desarrollo de software.

No debe verse los Patrones de Diseño como una teoría o una corriente. No trata de tomar partido por una u otra alternativa. Es una experiencia real, probada y que funciona. Es Historia y ayuda a no cometer los mismos errores.(LARMAN 2004)

Cada patrón describe un problema que ocurre una y otra vez en el entorno, para describir después el núcleo de la solución a ese problema, de tal manera que esa solución pueda ser usada más de un millón de veces sin hacerlo ni siquiera dos veces de la misma forma.

Para el diseño de la aplicación se hizo uso de los Patrones Generales de Software para Asignar Responsabilidades (GRASP). Los cinco patrones de GRASP que se utilizaron fueron: Experto, Creador, Alta Cohesión, Bajo Acoplamiento y Controlador.

4.4.2 Arquitectura

La arquitectura en capas se encuentra centrada básicamente en dividir un problema en pequeñas partes que puedan ser manejadas e implementadas de forma independiente, dichas partes poseerán responsabilidades específicas que no dependan del funcionamiento de las otras o al menos que su dependencia sea mínima. Este aspecto constituye una ventaja considerable pues proporciona una amplia reutilización de las clases implementadas al hacer abstracciones de las distintas funcionalidades o responsabilidades del sistema agrupándolas en capas.

La arquitectura en capas brinda además beneficios como:

- Aislamiento de la lógica de la aplicación en componentes separados reutilizables en otras aplicaciones.
- Distribución de capas en diferentes máquinas o procesos, lo que puede mejorar el rendimiento, aumentar la coordinación y el compartimiento de información entre cliente y servidor.
- Dedicación de recursos a cada una de las capas y posibilidad de desarrollarlas en paralelo.

Se definen tres capas: Presentación, Lógica del Negocio y Acceso a Datos.

1. Presentación: En esta capa se diseña todo lo que constituye la interfaz gráfica y la interacción del usuario con el sistema.
2. Lógica del Negocio: Contiene todas las subrutinas creadas con el propósito de regular alguna acción del usuario.
3. Acceso a Datos: En esta capa se programa todo lo que tiene que ver con el acceso a la base de datos. Esta capa queda encargada de tomar la información de la base de datos dada una petición de la capa de Lógica del Negocio, que a su vez es generada por la capa de presentación.

En la capa de Presentación se encuentran relacionadas las clases de interfaz de usuario, sobre las cuales solo se realiza la programación básica de la interfaz y la validación de los componentes de las mismas. Ejemplo: CI:FAutenticarse, CI:FPrincipal, CI:FDirectivos etc.

En la capa de Lógica del Negocio se encuentran relacionadas las clases controladoras y entidades del sistema. En estas clases se encuentran implementados los métodos que permiten regular las acciones de los usuarios. En ellas se crean e interactúa con los objetos que forman parte del negocio. Ejemplo:

CC:ControlaDirectivos, CC:ControlaEquipo, CC:ControlaJugador, CE:Equipos, CE:Eventos, CE:CargoDireccion, etc.

En el caso de la capa de Acceso a Datos se utilizó un webservice como abstracción de la clase de accesos a datos. En este webservice se encuentran todos los métodos que permiten la interacción con la base de datos del sistema. Ejemplo: InsertarJugador(), CargarJugador(), EliminarJugador(), InsertarEvento(), etc.

Es necesario mencionar que no se hace uso del webservice con el objetivo de brindar servicios a terceras personas, sino que su uso se limita única y exclusivamente al aprovechamiento de las ventajas que esta tecnología ofrece al sistema.

4.5 Diseño de la base de datos

En el diseño de la base de datos, se nombran las tablas con un sustantivo que la identifica con la primera letra en mayúscula. Ejemplo Evento.

Los nombres de los atributos de las tablas comienzan con mayúscula, y en caso de ser un nombre compuesto, las dos palabras empiezan con mayúsculas, sin espacios intermedios.

Para ver el diagrama correspondiente remítase al Anexo 6. Modelo de Datos.

4.6 Estándares de la codificación

Resulta muy ventajoso utilizar un estándar para escribir código, pues se reducen considerablemente los errores, y los códigos resultan más compresibles y fáciles de leer. Con vistas a garantizar la homogeneidad de dicho código, se establece el estilo descrito a continuación:

Comentarios: Los comentarios se definen comenzando con los caracteres */** y terminando con **/* para los comentarios de varias líneas, y comenzando con los caracteres *//* para los de una sola línea.

Declaraciones:

- Las constantes se declaran en mayúsculas.
- Las variables deben ser explícitas.
- Para declarar una clase se utiliza un sustantivo comenzando con mayúscula
- Los Atributos de las clases siempre empiezan con el _.

Para nombrar variables, y demás elementos, se precede cada nombre con un prefijo para su fácil identificación,

Ejemplo:

- Campos de edición: **TXT**Nombre.
- Label: **LB**Nombre.
- Botones de acción: **BTN**Nombre.
- Variables de control de ciclos: **i, j, k**
- Listview: **LV**Nombre
- Datasets: **DS**
- RadioButtonList: **RB**Nombre

Espacios en blanco: Colocar espacios en blanco entre operadores lógicos-aritméticos y sus operandos.

Miscelánea: Indentar al mismo nivel del bloque al que pertenecen las llaves para abrir y cerrar un método o un bloque de control de flujo. Ejemplo:

```
If (a != 1)
{
...
}
```

4.7 Encriptación de datos

Existen muchos algoritmos de encriptación reconocidos mundialmente. Algunos de ellos son: AES, MD5, DES y Triple DES. En el caso del sistema propuesto se llegó a la conclusión de que existe información que requiere cierto nivel de seguridad y confidencialidad como es el caso de los usuarios y contraseñas del sistema. Garantizar este aspecto se utilizó el algoritmo Triple DES.

Este algoritmo constituye una versión mejorada del DES. Se implementó con el objetivo de reforzar la clave de 64 bits del DES con una mucho más potente de 192 bits. El procedimiento para la encriptación que sigue este algoritmo es exactamente igual al DES pero repetido tres veces, de ahí su nombre Triple DES. Los datos son encriptados con una primera clave, luego descriptado con una segunda clave y finalmente encriptado nuevamente con una tercera.

El Triple DES está desapareciendo lentamente, siendo reemplazado por el algoritmo AES. Sin embargo, la mayoría de las tarjeta de crédito y otros medios de pago electrónico tienen como estándar el algoritmo Triple DES (anteriormente usaban el DES).

El algoritmo DES tiene la desventaja de ser lento en el proceso de encriptación y por tanto el Triple DES es prácticamente tres veces más lento que el anterior por lo que su uso se ve cuestionado en algunas ocasiones a pesar de la alta seguridad que ofrece. En el caso del sistema propuesto, no constituye un factor importante el hecho de su lentitud pues no se utiliza para encriptar grandes volúmenes de datos solo usuario, contraseña, etc. Sin embargo si constituye una ventaja muy grande para la aplicación la seguridad que este le proporciona.

4.8 Modelo de despliegue

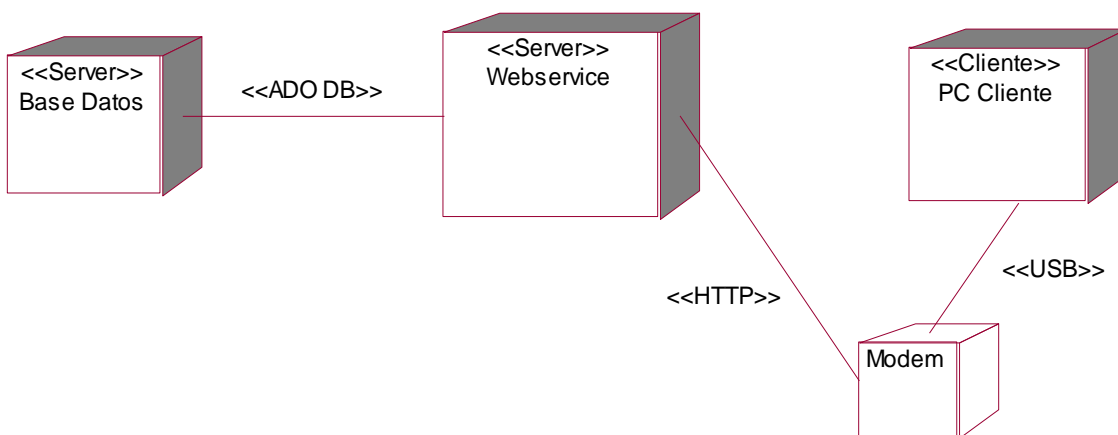


Figura 4.12 Diagrama del Modelo de Despliegue.

4.9 Modelo de Componentes

En este epígrafe se realizó el modelo de componentes. Para hacer uso del mismo remitirse al Anexo 7. Modelo de Componentes.

4.10 Conclusiones

Este capítulo estuvo dirigido a presentar el diseño del sistema propuesto. En él se muestran los diagramas de clases del diseño, el modelo de datos, el modelo de componentes del sistema y el diagrama de despliegue. Se hace referencia a las generalidades que posee la implementación del sistema, así como los principios de diseño empleados y las diferentes pruebas realizadas al sistema.

Capítulo 5

Estudio de factibilidad

5.1 Introducción

El estudio de factibilidad es un paso importante que no se debe obviar en la realización de un proyecto, pues como resultado de este análisis se obtienen las estimaciones de: esfuerzo, tiempo de desarrollo en meses, costo del producto, la cantidad de personas que se necesitan para desarrollar el proyecto, entre otras; sirviendo de elemento esencial de planificación para el equipo de trabajo y posibilitando fijar con los clientes una fecha de terminación del producto.

En este capítulo se describe la estimación de costos del sistema propuesto y sus beneficios, basado en las técnicas de Análisis de Puntos de Casos de Uso.

5.2 Planificación basada en casos de uso. Análisis de puntos de casos de uso

Paso 1. Cálculo de los Puntos de casos de uso Desajustados.

$$UUCP=UAW+UUCW$$

Donde:

- UUCP: Puntos de casos de uso sin ajustar.
- UAW: Factor de peso de los actores sin ajustar.
- UUCW: Factor de peso de los casos de uso sin ajustar.

Tipo de actor	Descripción	Factor de peso	Actores	Total
Simple	Sistema con sistema a través de interfaz de programación.	1	0	0
Medio	Sistema con sistema mediante protocolo de interfaz basada en texto.	2	0	0
Complejo	Persona que interactúa con el sistema mediante interfaz gráfica.	3	2	6

Tabla 5.1 Factor de peso de los actores sin ajustar.

$$UAW = \sum cant\ actores * peso$$

$$UAW=6$$

Tipo de CU	Descripción	Peso	Cantidad de CU	Total
Simple	El caso de uso tiene de 1 a 3 transacciones.	5	4	20
Medio	El caso de uso tiene de 4 a 7 transacciones.	10	25	250
Complejo	El caso de uso tiene más de 8 transacciones.	15	0	0

Tabla 5.2 Factor de peso de los casos de uso sin ajustar.

$$UUCW = \sum cant\ CU * Peso$$

$$UUCW=270$$

$$UUCP=6+270$$

$$UUCP=276$$

Paso 2. Cálculo de los Puntos de casos de uso ajustados.

$$UCP=UUCP*TCF*EF$$

Donde:

- UCP: Puntos de casos de uso ajustados.
- UUCP: Puntos de casos de uso sin ajustar.
- TCP: Factor de complejidad técnica.
- EF: Factor de ambiente.

El factor de complejidad técnica (TCF) se calcula mediante la cuantificación de un conjunto de factores que determinan la complejidad técnica del sistema. Cada factor se cuantifica en un valor desde 0 (aporte irrelevante) hasta 5 (aporte muy relevante).(PRESSMAN)

Factor	Descripción	Peso	Valor asignado	Total
T1	Sistema distribuido	2	4	8
T2	Tiempo de respuesta	1	4	4
T3	Eficiencia del usuario final	1	4	4
T4	Funcionamiento Interno complejo	1	5	5
T5	El código debe ser reutilizable	1	4	4
T6	Facilidad de instalación	0.5	5	2.5
T7	Facilidad de uso	0.5	4	2
T8	Portabilidad	2	4	8
T9	Facilidad de cambio	1	4	4
T10	Concurrencia	1	5	5

T11	Incluye objetivos especiales de seguridad	1	4	4
T12	Provee acceso directo a terceras partes	1	0	0
T13	Se requieren facilidades especiales de entrenamiento de usuarios	1	2	2

Tabla 5.3 Factor de complejidad técnica.

$$TCF = 0.6 + 0.01 * \sum (\text{peso} * \text{valor asignado})$$

$$TCF = 0.6 + 0.01 * 52.5$$

$$TCF = 0.6 + 0.525$$

$$TCF=1.125$$

El factor de ambiente (EF) está relacionado con las habilidades y entrenamiento del grupo de desarrollo que realiza el sistema. Cada factor se cuantifica con un valor desde 0 (aporte irrelevante) hasta 5 (aporte muy relevante).(PRESSMAN)

Factor	Descripción	Peso	Valor asignado	Total
E1	Familiaridad con el modelo de proyecto utilizado	1.5	4	6
E2	Experiencia en la aplicación	0.5	5	2.5
E3	Experiencia en la orientación a objetos.	1	4	4
E4	Capacidad del analista líder.	0.5	4	2
E5	Motivación.	1	5	5
E6	Estabilidad de requerimientos	2	5	10
E7	Personal Part–Time	-1	3	-3
E8	Dificultad del lenguaje de programación	-1	3	-3

Tabla 5.4 Factor de ambiente.

$$EF = 1.4 - 0.03 * \sum (\text{peso} * \text{valor asignado})$$

$$EF = 1.4 - 0.03 * 23.5$$

$$EF = 1.4 - 0.705$$

$$EF = 0.695$$

$$\mathbf{UCP = UUCP * TCF * EF}$$

$$UCP = 276 * 1.125 * 0.695$$

$$UCP = 215.7975$$

Paso 3. Estimación de esfuerzo a través de los puntos de casos de uso.

$$\mathbf{E = UCP * CF}$$

Donde:

- E: Esfuerzo estimado en horas hombres.
- UCP: Punto de casos de usos ajustados.
- CF: Factor de conversión.

Para obtener el factor de conversión (CF) se cuentan cuantos valores de los que afectan el factor ambiente (E1...E6) están por debajo de la media (3), y los que están por arriba de la media para los restantes (E7, E8). Si el total es 2 o menos se utiliza el factor de conversión 20 Horas-Hombre / Punto de Casos de uso. Si el total es 3 o 4 se utiliza el factor de conversión 28 Horas-Hombre / Punto de Casos de uso. Si el total es mayor o igual que 5 se recomienda efectuar cambios en el proyecto ya que se considera que el riesgo de fracaso del mismo es demasiado alto. (PRESSMAN)

En este caso se puede decir que:

$$CF = 20 \text{ Horas-Hombre / Punto de Casos de uso.}$$

$$E = 215.7975 * 20$$

$$E = 4315.95 \text{ Horas-Hombre}$$

Paso 4. Calcular esfuerzo de todo el proyecto.

Actividad	Porcentaje %	Horas-Hombres
Análisis	10	1078.9875
Diseño	20	2157.975
Implementación	40	4315.95
Pruebas	15	1618.48125
Sobrecarga (otras actividades)	15	1618.48125
Total	100	10789.875

Tabla 5.5 Cálculo del esfuerzo.

Si $E_T = 10789.875$ horas-hombre y cada mes los desarrolladores trabajan como promedio 200 horas, eso daría un:

$$E_T = 53.949375 \text{ mes-hombre.}$$

Esto quiere decir que 2 personas pueden realizar el problema analizado en 27 meses aproximadamente.

5.3 Costo del Proyecto

Se asume como salario promedio mensual \$100.00

$$CHM = 2 * \text{Salario Promedio}$$

$$CHM = 200.00 \text{ \$/mes}$$

$$\text{Costo} = \text{Salario Promedio} * E_T$$

$$\text{Costo} = 100.00 * 53.949375$$

Costo = \$5394.9375

5.4 Beneficios tangibles e intangibles

5.4.1 Beneficios tangibles

El proyecto realizado aporta al INDER y a la Comisión Nacional y Comisiones Provinciales de Béisbol los siguientes beneficios tangibles:

- Existencia de una aplicación para la gestión de eventos competitivos que automatiza gran parte del proceso, mejorando las condiciones del software “Béisbol”.
- Existencia de una aplicación para la anotación de los juegos de béisbol que automatiza el proceso de codificación de las jugadas haciendo más fácil la labor de los anotadores.
- Reducción de los gastos en softwares para el béisbol gracias a la posibilidad que brinda el proyecto de construir importantes módulos asociados al mismo.

La comercialización del producto final obtenido queda en manos única y exclusivamente de los directivos de la Universidad de las Ciencias Informáticas, ya sea nacional o internacionalmente. La comercialización del proyecto constituye un beneficio tangible adicional aportado.

5.4.2 Beneficios intangibles

Entre los beneficios intangibles que brinda podemos mencionar:

- Brinda una vía más efectiva para la anotación de los juegos y la coordinación de eventos competitivos.
- Ofrece la posibilidad de recoger mayor cantidad de datos en cada partido, característica que aumentará el haber estadístico existente en el país.
- Dará paso a un conjunto de módulos asociados a este sistema como: módulo para la televisión, sitio oficial del béisbol cubano, etc.
- Mejorará en gran medida la aplicación que actualmente es utilizada para estos fines, logrando mayor accesibilidad, funcionalidad y usabilidad del sistema por parte de los usuarios.

5.5 Análisis de costos y beneficios

El desarrollo de un producto informático es justificable en dependencia del costo de su desarrollo y de los beneficios que reportaría su implantación y uso.

El sistema para la gestión de eventos competitivos y anotación de los juegos de béisbol brinda un conjunto de beneficios que se relacionan a continuación:

- Permite al usuario reutilizar los datos existentes en la Base de Datos de manera que al registrarse un nuevo evento competitivo no es necesario adicionar los jugadores, jueces y directivos que ya han participado en eventos anteriores.
- Mejora la estructuración de los datos en la Base de Datos brindando la posibilidad de recoger un número mayor de estadísticas que las que se registran actualmente.
- Ofrece una interfaz con mayor usabilidad, funcionalidad y accesibilidad que la aplicación actual. Permite al anotador realizar la anotación de un juego de béisbol sin necesidad de aprender el sistema codificado para la anotación. Además de ofrecer un entorno gráfico Windows que mejora la interfaz en MS-DOS del software actual.

Luego de haberse realizado el estudio del proyecto y el cálculo del costo del mismo se determinó que estos beneficios que brinda la implementación del sistema, se pueden alcanzar con solo un costo de \$ 5394.9375 lo cual constituye un precio razonable si se tiene en cuenta la complejidad e importancia del sistema. Por todo lo antes expuesto, se considera factible el desarrollo de este sistema.

5.6 Conclusiones

En este capítulo se describió el estudio de factibilidad realizado correspondiente al sistema propuesto, teniendo en cuenta el costo estimado y los beneficios que reportará al ser implantado.

El sistema propuesto reportará beneficios significativos para el INDER y las Comisiones Provinciales y Nacionales de Béisbol; así como también reportará beneficios para el país en general al no tener que realizar grandes inversiones financieras la compra de software para desarrollar estas tareas. Por estas razones y por lo antes expuesto en este capítulo, se considera factible el desarrollo e implementación del mismo.

COCLUSIONES

Con el desarrollo del Sistema de Gestión de Eventos Competitivos y Anotación de los Juegos de Béisbol, se da cumplimiento a los objetivos de este trabajo, pues da camino a la obtención de un sistema en el que se aplican los resultados de todo el proceso investigativo realizado a lo largo de las etapas del proyecto, lográndose:

- Un mecanismo de autenticación de usuarios para identificarlos y establecer el control de acceso a la información, así como a las diferentes operaciones que les es accesible en dependencia de su rol de usuario.
- La implementación de un sistema de gestión de eventos competitivos que brinda los servicios de la aplicación existente y e incorpora nuevas funcionalidades al sistema.
- La reutilización de los datos existentes en la base de datos a la hora de crear nuevos eventos deportivos, logrando que no se requiera de la inserción repetitiva de deportistas, directivos o jueces a participar en dichos eventos.
- La modelación del sistema para la anotación de los juegos de béisbol lográndose la codificación automatizada de las jugadas y lanzamientos del partido.
- La mejor estructuración de los datos por parte de la aplicación, brindando la posibilidad de recoger mayor información de cada juego de béisbol desarrollado.
- La posibilidad de implementar nuevos módulos asociados a este proyecto utilizándolo como base de su implementación.
- La creación de un sistema usable, funcional y accesible para los usuarios que mejora las condiciones del software existente.

RECOMENDACIONES

De acuerdo a los resultados de todo el proceso de investigación realizado y basados en la experiencia acumulada se proponen las siguientes recomendaciones:

- Ampliar las funcionalidades del Sistema de Gestión de Eventos Competitivos y Anotación de los Juegos de Béisbol de manera que se pueda realizar la codificación de las jugadas que hoy por hoy no pueden ser recogidas en el béisbol cubano.
- Desarrollar la implementación de un módulo para la televisión que visualice la mayor cantidad de información posible al usuario, utilizando como base para su implementación el Sistema de Gestión de Eventos Competitivos y Anotación de los Juegos de Béisbol.
- Diseñar un nuevo Portal del Béisbol Cubano adicionándole nuevos servicios e informaciones basados en los datos novedosos recogidos por el Sistema de Gestión de Eventos Competitivos y Anotación de los Juegos de Béisbol.
- Asumir como proyecto productivo de la UCI conjuntamente con el personal del INDER, la Comisión Nacional de Béisbol y la Federación Cubana de Béisbol, la implementación absoluta del Sistema de Gestión de Eventos Competitivos y Anotación de los Juegos de Béisbol y los módulos que se derivan de este, en beneficio del deporte cubano y del país en general.

REFERENCIAS BIBLIOGRÁFICAS

- CORUJEDO, J. F. *Panorama Histórico*. Ediciones INDER, 1965. p.
- D GAUSE, G. W. *Exploring Requirements*. Dorset House, 1989. p. 0932633137
- DATE, C. J. *Introducción a los Sistemas de Bases de Datos.*, Addison Wesley, 2001. 960 p. 9684444 192
- DESARROLLOWEB.COM. *Ventajas de .Net*, [Web]. 2006. [Disponible en: <http://www.desarrolloweb.com/articulos/1329.php>]
- GRADY BOOCH, J. R., IVAR JACOBSON. *The Unified Modeling Language User Guide*. Addison-Wesley 1999. 476 p. 0-201-57168-4
- IVAR JACOBSON, G. B., JAMES RUMBAUGH. *The Unified Software Development Process*. Addison-Wesley, 1999. 460 p. 0-201-57169-2
- J GROFF, P. W. *LAN Times: Guide to SQL*. Mc Graw-Hill 1994. p. 007882026X
- LARMAN, C. *UML y Patrones. Introducción al análisis y diseño orientado a objetos*. Addison-Wesley, 2004. 499 p.
- MARTIN, R. *Designing SQL Server 2000 Databases for .net Enterprises Servers*. Edición digital. Syngress., 2001. p.
- MENÉNDEZ, R. *Rational Rose*, 2003. [Disponible en: <http://www.rational.com.ar/herramientas/rose.html>]
- MORAN, B. *Análisis comparativo entre Microsoft® .NET y Sun® J2EE*, [web]. 2006. [Disponible en: <http://www.ciberteca.net/articulos/programacion/net>].
- PARDO, V. G. V. *Nuevas tecnologías de la información y la comunicación*. Villa Clara, Editorial Feijoo, 2003. p.
- PRESSMAN, R. S. *Ingeniería de Software. Un enfoque práctico*. Madrid, Mcgraw-Hill, 836 p. 0077096770
- SANCHEZ, M. M. *Metodologías De Desarrollo De Software*, [web]. 2006. [Disponible en: http://www.informatizate.net/articulos/metodologias_de_desarrollo_de_software07062004.html]
- VILLATE, J. E. *Introducción al XML*. Universidad de Oporto, p.

BIBLIOGRAFÍA CONSULTADA

- GONZÁLEZ SECO, JOSÉ ANTONIO. *El lenguaje de programación C#*
- ¿QUÉ ES SQL? <http://www.desarrolloweb.com/articulos/262.php>, 05/05/2004.
- Microsoft Developer Network. MSDN Library para Visual Studio .NET 2005
- Conferencias de Ingeniería de software (Curso 2005-2006) UCI. La Habana, 2007.
- Diagramas de Casos de Uso. <http://www.vico.org/MuestrarioDiagCU.pdf>.
- Curso de SQL <http://www.cybercursos.net/sql/sql.html> 08/07/2004.
- CORUJEDO, J. F. *Panorama Histórico. Ediciones INDER, 1965. p.*
- D GAUSE, G. W. *Exploring Requirements. Dorset House, 1989. p. 0932633137*
- DATE, C. J. *Introducción a los Sistemas de Bases de Datos., Addison Wesley, 2001. 960 p. 9684444192*
- DESARROLLOWEB.COM. *Ventajas de .Net, [Web]. 2006. [Disponible en: <http://www.desarrolloweb.com/articulos/1329.php>*
- GUERRERO, L. A. *Rational Unified Process. Universidad de Chile. Dpto.Ciencias de la Computación, 2001. p.*
- J GROFF, P. W. *LAN Times: Guide to SQL. Mc Graw-Hill 1994. p. 007882026X*
- LARMAN, C. *UML y Patrones. Introducción al análisis y diseño orientado a objetos. Addison-Wesley, 2004. 499 p.*
- MARTIN, R. *Designing SQL Server 2000 Databases for .net Enterprises Servers. Edición digital. Syngress., 2001. p.*
- MENÉNDEZ, R. *Rational Rose, 2003. [Disponible en: <http://www.rational.com.ar/herramientas/rose.html>*
- PARDO, V. G. V. *Nuevas tecnologías de la información y la comunicación. Villa Clara, Editorial Feijoo, 2003. p.*
- PRESSMAN, R. S. *Ingeniería de Software. Un enfoque práctico. Madrid, Mcgraw-Hill, 836 p. 0077096770*
- VILLATE, J. E. *Introducción al XML. Universidad de Oporto, p.*

GLOSARIO DE TÉRMINOS

Alineación: Orden al bate y posición de los jugadores durante un juego de béisbol.

Box Score: Resumen detallado del Juego, donde se muestran algunas de las estadísticas de los jugadores involucrados en el mismo y el marcador final.

Clásico Mundial: Evento mundial de béisbol celebrado en el año 2006 donde participaron equipos representativos de los países con béisbol profesional y al cual asistió una selección de peloteros cubanos como único equipo de béisbol amateur invitado al evento y obtuvo el valioso segundo lugar del torneo, únicamente superado por el equipo de Japón.

Estadio Latinoamericano: Estadio de Béisbol ubicado en la capital cubana. En esta institución se encuentran los encargados de la Comisión Nacional de Béisbol.

Rosters: En términos del béisbol la palabra roster está dirigida al conjunto de nombres de todos los jugadores que conforman un equipo, relacionados en un listado donde además del nombre se relacionan datos específicos del mismo.

Transact-SQL: (T-SQL) es el lenguaje de programación del SQL Server.

WebService: Sistema de software diseñado para soportar interacción máquina-a-máquina sobre una red. Posee una interfaz descrita en un formato procesable por máquina.

ANEXOS

Anexo 1. Diagramas de actividades de los casos de uso del negocio.

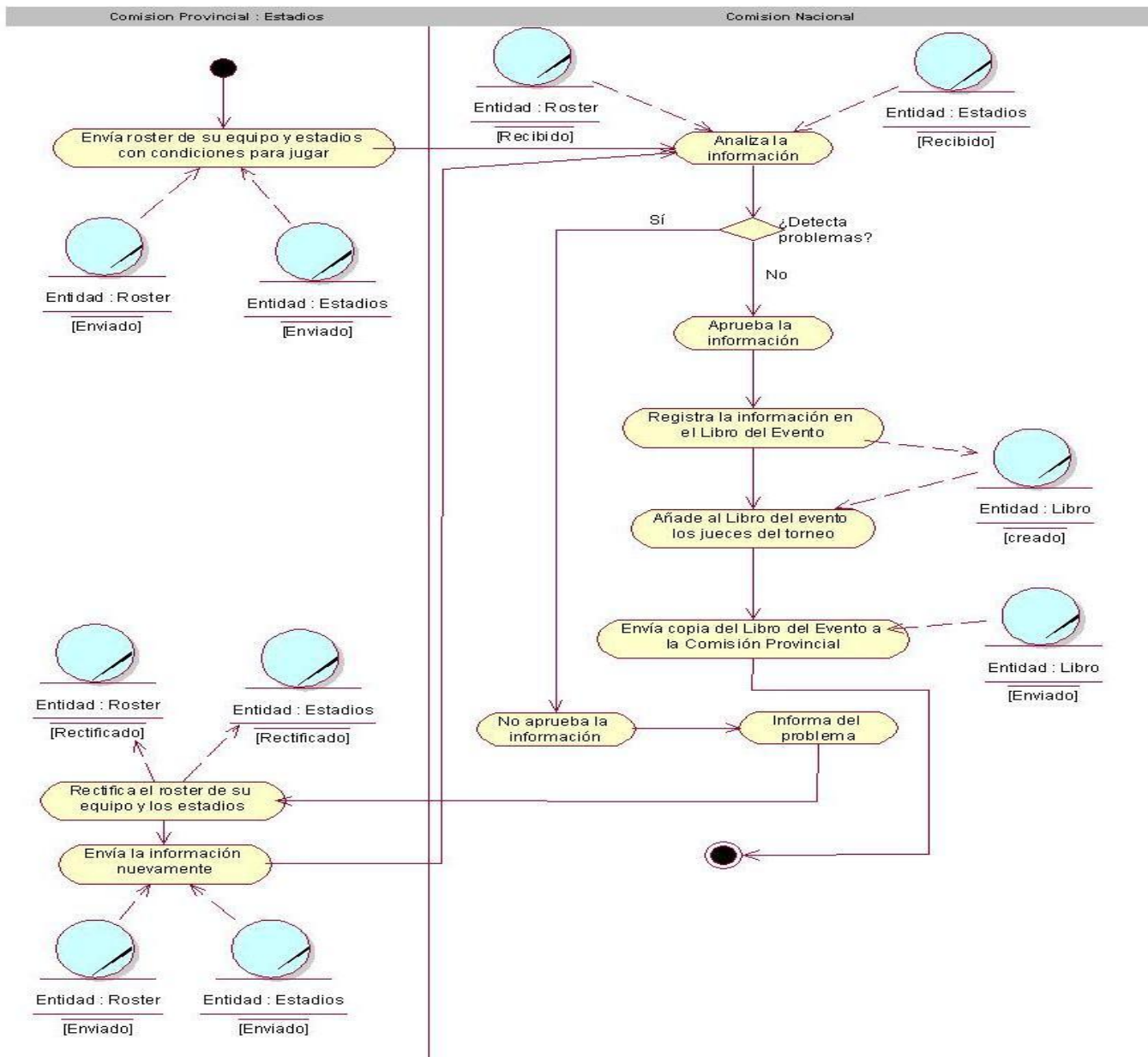


Figura 3.2 Diagrama de actividades. Caso de uso Crear Libro.

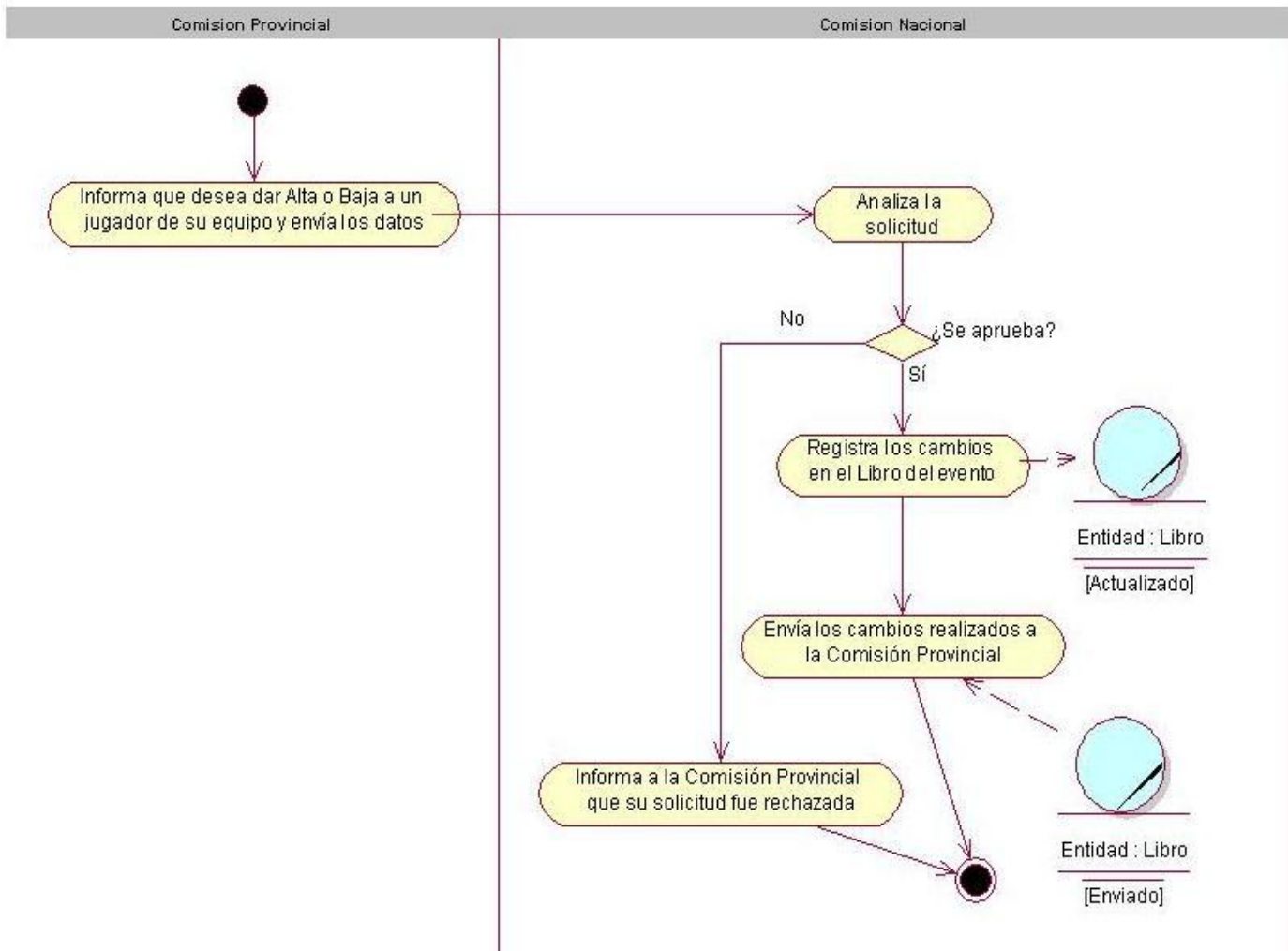


Figura 3.3 Diagrama de actividades. Caso de uso Dar Altas y Bajas.

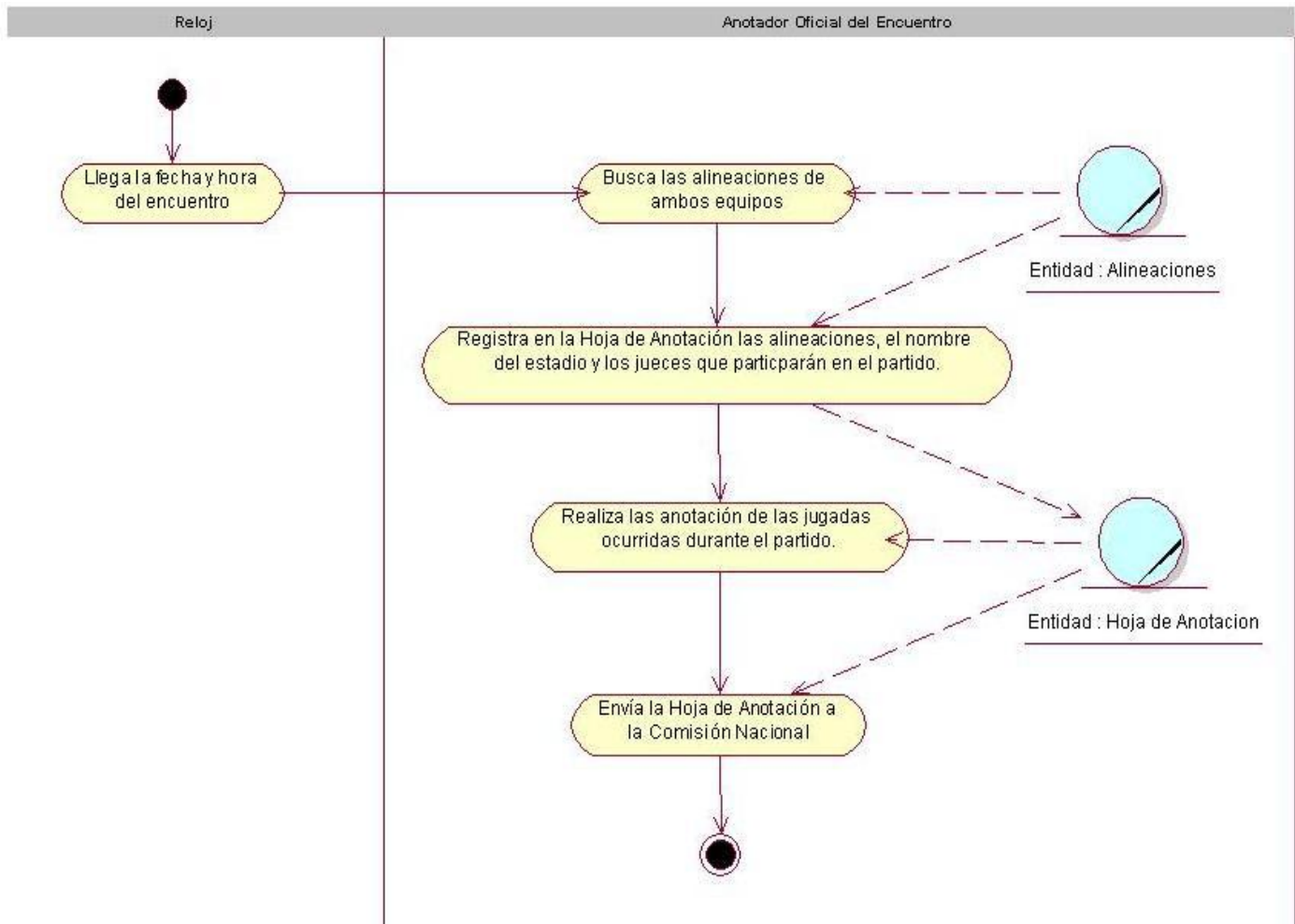


Figura 3.4 Diagrama de actividades. Caso de uso Anotar juego.

Anexo 2. Diagramas de casos de uso para cada paquete.

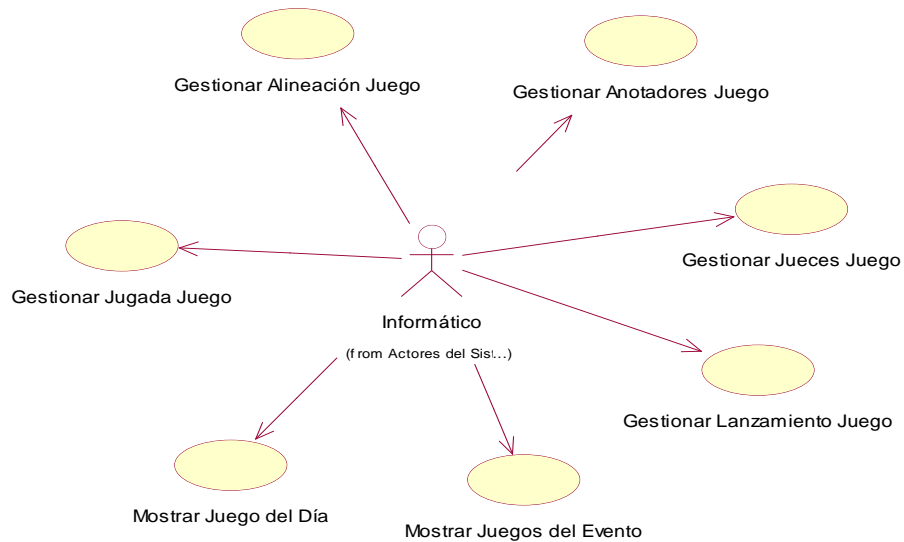


Figura 3.8 Diagrama de casos de uso del sistema para el paquete: Anotación Juegos.

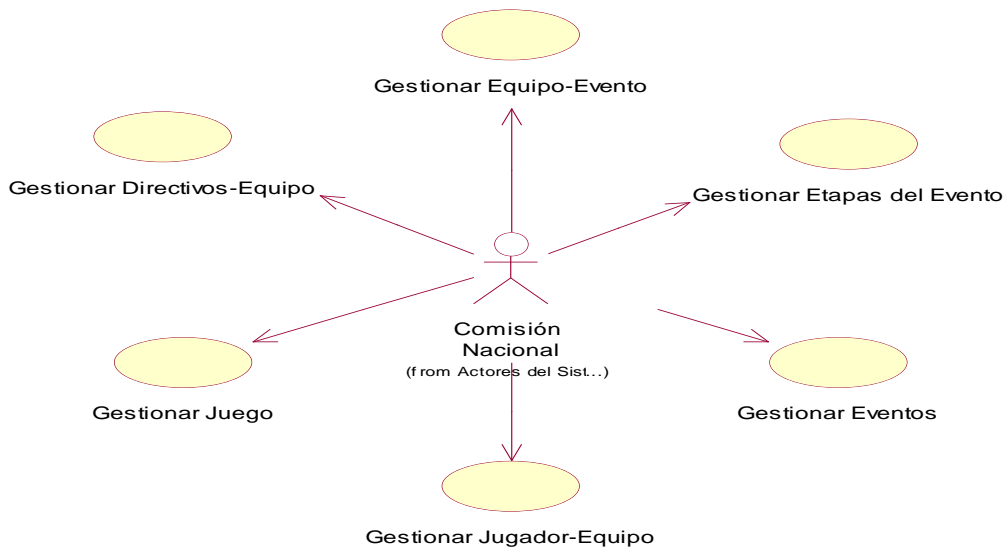


Figura 3.9 Diagrama de casos de uso del sistema para el paquete: Evento.

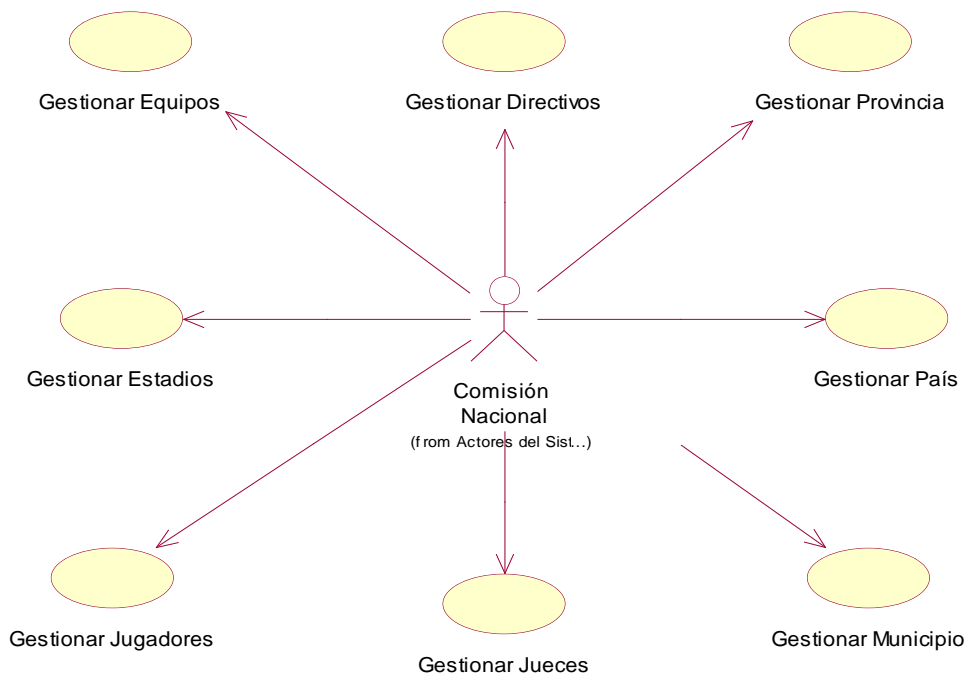


Figura 3.10 Diagrama de casos de uso del sistema para el paquete: Inicializadores.

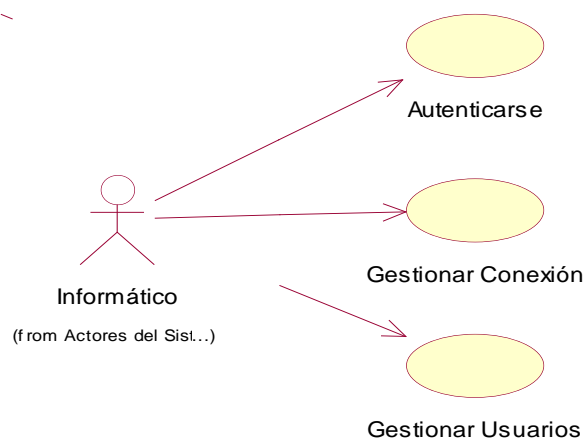


Figura 3.11 Diagrama de casos de uso del sistema para el paquete: Seguridad.

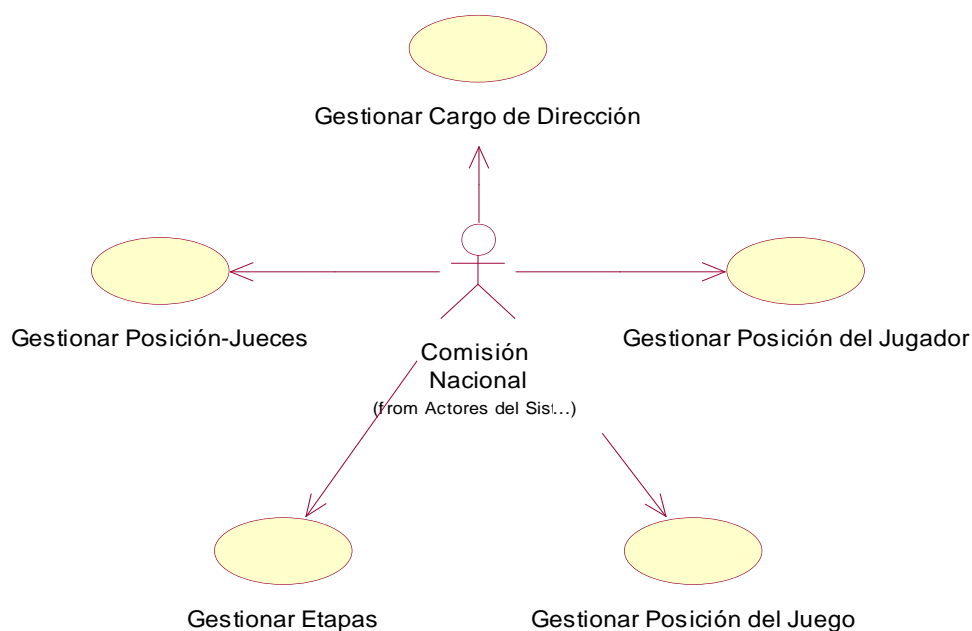


Figura 3.12 Diagrama de casos de uso del sistema para el paquete: Nomencladores.

Anexos 3. Descripción textual de los casos de uso del sistema.

Caso de Uso:	Autenticarse
Actores:	Informático(Inicia)
Propósito	Reconocer al usuario que utilizará el sistema para establecer el nivel de acceso requerido por el mismo.
Resumen:	El CUS se inicia cuando el Informático selecciona la opción de Autenticarse, el sistema le brinda la posibilidad de insertar sus datos y una vez reconocido el usuario le ofrece los permisos que posee dentro del sistema y termina el CUS.
Referencia:	R2
Precondiciones:	
Poscondiciones:	El Informático es reconocido por el sistema y se establece el nivel de acceso

	que posee según el usuario con que se registró.
Flujo Normal de Eventos	
Acción del Actor	Respuesta del Sistema
1 – El Informático selecciona la opción de <u>Autenticarse</u>	1.1 – El sistema solicita nombre de usuario y contraseña.
2 – El Informático introduce los datos solicitados por el sistema.	2.1 – El sistema verifica los datos introducidos por el Informático. 2.2 – Si los datos son correctos el sistema establece el nivel de acceso que posee el usuario registrado y da acceso a la aplicación. Termina así el CUS.
Flujos Alternos	
Curso Alternativo de Eventos	2.2 – Si los datos introducidos por el Informático son incorrectos el sistema le muestra un mensaje de error e indica al Informático retornar a la acción 2.
Prioridad:	Crítico

Tabla 3.7 Descripción textual. Caso de uso Autenticarse.

Caso de Uso:	Gestionar Jugadores
Actores:	Comisión Nacional(Inicia)
Propósito	Permite al especialista de la Comisión Nacional Agregar, Modificar, Eliminar y Mostrar los datos de los jugadores.
Resumen:	El CUS se inicia cuando el especialista de la Comisión Nacional selecciona la opción de Gestionar Jugadores, luego selecciona el tipo de gestión, introduce los datos necesarios, el sistema realiza la acción seleccionada y termina el CUS.
Referencia:	R4

Precondiciones:	1. El especialista de la Comisión Nacional tiene que autenticarse.
Poscondiciones:	1. Información del jugador adicionada a la base de datos. 2. Información del jugador actualizada en la base de datos. 3. Información del jugador eliminada de la base de datos. 4. Información del jugador mostrada.
Flujo Normal de Eventos	
Acción del Actor	Respuesta del Sistema
1 – El especialista de la Comisión Nacional selecciona la opción de <u>Gestionar Jugadores</u>	1.1 – El sistema muestra las opciones: - <u>Adicionar Jugador</u> - <u>Actualizar Jugador</u> - <u>Eliminar Jugador</u> - <u>Mostrar Jugador</u>
Sección 1 : Adicionar Jugador	
Acción del Actor	Respuesta del Sistema
1 – El especialista selecciona la opción de <u>Adicionar Jugador</u> .	1.1 – El sistema le muestra un formulario a completar para la adición de un nuevo jugador.
2 – El especialista introduce los datos solicitados por el sistema.	2.1 – El sistema verifica los datos introducidos por el especialista. 2.2 – Si los datos introducidos son correctos el sistema procede a adicionar al jugador en la base de datos y termina el CUS.
Flujos Alternos	
Curso Alternativo de Eventos	2.2 – Si los datos introducidos por el especialista son incorrectos el sistema le muestra un mensaje de error indicándole donde está el dato erróneo e indica al especialista retornar a la acción 2.
Sección 2: Actualizar Jugador	
Acción del Actor	Respuesta del Sistema
1 – El especialista selecciona la opción de	1.1 – El sistema le muestra un listado con todos los

<u>Actualizar Jugador.</u>		jugadores existentes en la base de datos.
2 – El especialista selecciona el jugador a actualizar.		2.1 – El sistema localiza los datos del jugador y los muestra, listos para actualizar.
3 – El especialista realiza los cambios necesarios en los datos.		3.1 – El sistema verifica los datos modificados por el especialista. 3.2 – Si los datos son correctos el sistema actualiza los datos del jugador en la base de datos y termina el CUS.
Flujos Alternos		
Curso alternativo de eventos	3.2 – Si los datos introducidos por el especialista son incorrectos el sistema muestra un mensaje de error indicando donde está el dato erróneo e indica al especialista retornar a la acción 3.	
Sección 3: Eliminar Jugador		
Acción del Actor		Respuesta del Sistema
1 – El especialista selecciona la opción de <u>Eliminar Jugador.</u>		1.1 – El sistema le muestra un listado con los jugadores existentes en la base de datos.
2 – El especialista selecciona el jugador a eliminar.		2.1 –El sistema localiza los datos del jugador seleccionado y los muestra, listo para eliminar.
3 – El especialista selecciona la opción de eliminar jugador.		3.1 – El sistema le muestra un mensaje de advertencia para la acción a realizar.
4 – El especialista confirma si quiere o no eliminar el jugador.		4.1 – Si el especialista acepta, el sistema procede a la eliminación de los datos del jugador seleccionado y termina el CUS.
Flujos Alternos		
Curso alternativo de eventos	4.1 – Si el especialista cancela la acción se culmina el CUS sin ejecutar ninguna acción.	
Sección 4: Mostrar Jugador		
Acción del Actor		Respuesta del Sistema
1 – El especialista selecciona la opción de <u>Mostrar Jugador.</u>		1.1 – El sistema muestra un listado con todos los jugadores de la base de datos.

2 – El especialista selecciona el jugador deseado.	2.1 – El sistema localiza los datos del jugador y muestra la información del mismo terminando el CUS.
Flujos Alternos	
Curso alternativo de eventos	
Prioridad:	Crítico

Tabla 3.8 Descripción textual. Caso de uso Gestionar Jugadores.

Caso de Uso:	Gestionar Equipos
Actores:	Comisión Nacional(Inicia)
Propósito	Permite al especialista de la Comisión Nacional Agregar, Modificar, Eliminar y Mostrar los datos de los equipos.
Resumen:	El CUS se inicia cuando el especialista de la Comisión Nacional selecciona la opción de Gestionar Equipos, luego selecciona el tipo de gestión, introduce los datos necesarios, el sistema realiza la acción seleccionada y termina el CUS.
Referencia:	R5
Precondiciones:	1. El especialista de la Comisión Nacional tiene que autenticarse.
Poscondiciones:	1. Información del equipo adicionada a la base de datos. 2. Información del equipo actualizada en la base de datos. 3. Información del equipo eliminada de la base de datos. 4. Información del equipo mostrada.
Flujo Normal de Eventos	
Acción del Actor	Respuesta del Sistema
1 – El especialista de la Comisión Nacional selecciona la opción de <u>Gestionar Equipos</u>	1.1 – El sistema muestra las opciones: - <u>Adicionar Equipo</u> - <u>Actualizar Equipo</u>

	<ul style="list-style-type: none"> - <u>Eliminar Equipo</u> - <u>Mostrar Equipo</u>
Sección 1 : Adicionar Equipo	
Acción del Actor	Respuesta del Sistema
1 – El especialista selecciona la opción de <u>Adicionar Equipo</u> .	1.1 – El sistema le muestra un formulario a completar para la adición de un nuevo equipo.
2 – El especialista introduce los datos solicitados por el sistema.	2.1 – El sistema verifica los datos introducidos por el especialista. 2.2 – Si los datos introducidos son correctos el sistema procede a adicionar el equipo en la base de datos y termina el CUS.
Flujos Alternos	
Curso Alternativo de Eventos	2.2 – Si los datos introducidos por el especialista son incorrectos el sistema le muestra un mensaje de error indicándole donde está el dato erróneo e indica al especialista retornar a la acción 2.
Sección 2: Actualizar Equipo	
Acción del Actor	Respuesta del Sistema
1 – El especialista selecciona la opción de <u>Actualizar Equipo</u> .	1.1 – El sistema le muestra un listado con todos los equipos existentes en la base de datos.
2 – El especialista selecciona el equipo a actualizar.	2.1 – El sistema localiza los datos del equipo y los muestra, listos para actualizar.
3 – El especialista realiza los cambios necesarios en los datos.	3.1 – El sistema verifica los datos modificados por el especialista. 3.2 – Si los datos son correctos el sistema actualiza los datos del equipo en la base de datos y termina el CUS.
Flujos Alternos	
Curso alternativo de eventos	3.2 – Si los datos introducidos por el especialista son incorrectos el sistema muestra un mensaje de error indicando donde está el dato erróneo e indica al especialista retornar a la acción 3.

Sección 3: Eliminar Equipo	
Acción del Actor	Respuesta del Sistema
1 – El especialista selecciona la opción de <u>Eliminar Equipo</u> .	1.1 – El sistema le muestra un listado con los equipos existentes en la base de datos.
2 – El especialista selecciona el equipo a eliminar.	2.1 –El sistema localiza los datos del equipo seleccionado y los muestra, listo para eliminar.
3 – El especialista selecciona la opción de eliminar equipo.	3.1 – El sistema le muestra un mensaje de advertencia para la acción a realizar.
4 – El especialista confirma si quiere o no eliminar el equipo.	4.1 – Si el especialista acepta, el sistema procede a la eliminación de los datos del equipo seleccionado y termina el CUS.
Flujos Alternos	
Curso alternativo de eventos	4.1 – Si el especialista cancela la acción se culmina el CUS sin ejecutar ninguna acción.
Sección 4: Mostrar Equipo	
Acción del Actor	Respuesta del Sistema
1 – El especialista selecciona la opción de <u>Mostrar Equipo</u> .	1.1 – El sistema muestra un listado con todos los equipos de la base de datos.
2 – El especialista selecciona el equipo deseado.	2.1 – El sistema localiza los datos del equipo y muestra la información del mismo terminando el CUS.
Flujos Alternos	
Curso alternativo de eventos	
Prioridad:	Crítico

Tabla 3.9 Descripción textual. Caso de uso Gestionar Equipos.

Caso de Uso:	Gestionar Eventos
Actores:	Comisión Nacional(Inicia)
Propósito	Permite al especialista de la Comisión Nacional Agregar, Modificar, Eliminar y Mostrar los datos de los eventos.
Resumen:	El CUS se inicia cuando el especialista de la Comisión Nacional selecciona la opción de Gestionar Eventos, luego selecciona el tipo de gestión, introduce los datos necesarios, el sistema realiza la acción seleccionada y termina el CUS.
Referencia:	R9
Precondiciones:	1. El especialista de la Comisión Nacional tiene que autenticarse.
Poscondiciones:	1. Información del evento adicionada a la base de datos. 2. Información del evento actualizada en la base de datos. 3. Información del evento eliminada de la base de datos. 4. Información del evento mostrada.

Flujo Normal de Eventos

Acción del Actor	Respuesta del Sistema
1 – El especialista de la Comisión Nacional selecciona la opción de <u>Gestionar Eventos</u>	1.1 – El sistema muestra las opciones: - <u>Adicionar Evento</u> - <u>Actualizar Evento</u> - <u>Eliminar Evento</u> - <u>Mostrar Evento</u>

Sección 1 : Adicionar Evento

Acción del Actor	Respuesta del Sistema
1 – El especialista selecciona la opción de <u>Adicionar Evento</u> .	1.1 – El sistema le muestra un formulario a completar para la adición de un nuevo evento.
2 – El especialista introduce los datos solicitados por el sistema.	2.1 – El sistema verifica los datos introducidos por el especialista. 2.2 – Si los datos introducidos son correctos el sistema procede a adicionar el evento en la base de datos y

	termina el CUS.
Flujos Alternos	
Curso Alternativo de Eventos	2.2 – Si los datos introducidos por el especialista son incorrectos el sistema le muestra un mensaje de error indicándole donde está el dato erróneo e indica al especialista retornar a la acción 2.
Sección 2: Actualizar Evento	
Acción del Actor	Respuesta del Sistema
1 – El especialista selecciona la opción de <u>Actualizar Evento</u> .	1.1 – El sistema le muestra un listado con todos los eventos existentes en la base de datos.
2 – El especialista selecciona el evento a actualizar.	2.1 – El sistema localiza los datos del evento y los muestra, listos para actualizar.
3 – El especialista realiza los cambios necesarios en los datos.	3.1 – El sistema verifica los datos modificados por el especialista. 3.2 – Si los datos son correctos el sistema actualiza los datos del evento en la base de datos y termina el CUS.
Flujos Alternos	
Curso alternativo de eventos	3.2 – Si los datos introducidos por el especialista son incorrectos el sistema muestra un mensaje de error indicando donde está el dato erróneo e indica al especialista retornar a la acción 3.
Sección 3: Eliminar Evento	
Acción del Actor	Respuesta del Sistema
1 – El especialista selecciona la opción de <u>Eliminar Evento</u> .	1.1 – El sistema le muestra un listado con los eventos existentes en la base de datos.
2 – El especialista selecciona el evento a eliminar.	2.1 – El sistema localiza los datos del evento seleccionado y los muestra, listo para eliminar.
3 – El especialista selecciona la opción de eliminar evento.	3.1 – El sistema le muestra un mensaje de advertencia para la acción a realizar.
4 – El especialista confirma si quiere o no eliminar el evento.	4.1 – Si el especialista acepta, el sistema procede a la eliminación de los datos del evento seleccionado y

	termina el CUS.
Flujos Alternos	
Curso alternativo de eventos	4.1 – Si el especialista cancela la acción se culmina el CUS sin ejecutar ninguna acción.
Sección 4: Mostrar Evento	
Acción del Actor	Respuesta del Sistema
1 – El especialista selecciona la opción de <u>Mostrar Evento</u> .	1.1 – El sistema muestra un listado con todos los eventos de la base de datos.
2 – El especialista selecciona el evento deseado.	2.1 – El sistema localiza los datos del evento y muestra la información del mismo terminando el CUS.
Flujos Alternos	
Curso alternativo de eventos	
Prioridad:	Crítico

Tabla 3.10 Descripción textual. Caso de uso Gestionar Eventos.

Caso de Uso:	Gestionar Equipo-Evento
Actores:	Comisión Nacional(Inicia)
Propósito	Permite al especialista de la Comisión Nacional Agregar, Modificar y Eliminar equipos a un evento, así como mostrar la información de los equipos ya registrados en el evento.
Resumen:	El CUS se inicia cuando el especialista de la Comisión Nacional selecciona la opción de Gestionar Equipo-Evento, luego selecciona el tipo de gestión, introduce los datos necesarios, el sistema realiza la acción seleccionada y termina el CUS.
Referencia:	R10
Precondiciones:	<ol style="list-style-type: none"> 1. El especialista de la Comisión Nacional tiene que autenticarse. 2. Debe existir como mínimo un evento creado.

Poscondiciones:	<ol style="list-style-type: none"> 1. Equipo adicionado al evento. 2. Información del equipo a participar en el evento actualizada. 3. Equipo eliminado del evento. 4. Información de los equipos que participan en el evento mostrada.
Flujo Normal de Eventos	
Acción del Actor	Respuesta del Sistema
1 – El especialista de la Comisión Nacional selecciona la opción de <u>Gestionar Equipo-Evento</u>	1.1 – El sistema muestra las opciones: <ul style="list-style-type: none"> - <u>Adicionar Equipo-Evento</u> - <u>Actualizar Equipo-Evento</u> - <u>Eliminar Equipo-Evento</u> - <u>Mostrar Equipo-Evento</u>
Sección 1 : Adicionar Equipo-Evento	
Acción del Actor	Respuesta del Sistema
1 – El especialista selecciona la opción de <u>Adicionar Equipo-Evento</u> .	1.1 – El sistema muestra un listado de los eventos existentes en la base de datos.
2 – El especialista selecciona el evento al cual desea agregar el equipo.	2.1 – El sistema muestra un listado con los equipos existentes en la base de datos.
3 – El especialista selecciona el equipo que desea agregar al evento y da a la opción “Agregar”.	3.1 – El sistema agrega el equipo seleccionado al evento correspondiente y termina el CUS.
Flujos Alternos	
Curso Alternativo de Eventos	
Sección 2: Actualizar Equipo-Evento	
Acción del Actor	Respuesta del Sistema
1 – El especialista selecciona la opción de	1.1 – El sistema le muestra un listado con todos los

<u>Actualizar Equipo-Evento.</u>		eventos existentes en la base de datos.
2 – El especialista selecciona el evento al que pertenece el equipo a actualizar.		2.1 – El sistema muestra un listado con todos los equipos que participan en el evento seleccionado.
3 – El especialista selecciona el equipo a actualizar.		3.1 – El sistema muestra los datos del equipo listos para modificar.
4 – El especialista modifica los datos del equipo.		4.1 – El sistema verifica los datos modificados por el especialista. 4.2 – Si los datos son correctos el sistema procede a actualizar la información del equipo modificado en la base de datos y termina el CUS.
Flujos Alternos		
Curso alternativo de eventos	4.2 – Si los datos introducidos por el especialista son incorrectos el sistema muestra un mensaje de error indicando donde está el dato erróneo e indica al especialista retornar a la acción 4.	
Sección 3: Eliminar Equipo-Evento		
Acción del Actor		Respuesta del Sistema
1 – El especialista selecciona la opción de <u>Eliminar Equipo-Evento.</u>		1.1 – El sistema le muestra un listado con los eventos existentes en la base de datos.
2 – El especialista selecciona el evento a que pertenece el equipo que desea eliminar.		2.1 –El sistema muestra un listado de los equipos pertenecientes al evento seleccionado.
3 – El especialista selecciona el equipo que desea eliminar.		3.1 – El sistema muestra los datos del equipo listo para eliminar.
4 – El especialista selecciona la opción “Eliminar equipo”.		4.1 – El sistema muestra un mensaje para que el especialista confirme su solicitud.
5 – El especialista confirma si desea o no eliminar el equipo del evento.		5.1 – Si el especialista confirma que desea eliminar el equipo del evento, el sistema procede a eliminar los datos de la base de datos y termina el CUS.
Flujos Alternos		

Curso alternativo de eventos	5.1 – Si el especialista cancela la acción se culmina el CUS sin ejecutar ninguna acción.
Sección 4: Mostrar Equipo-Evento	
Acción del Actor	Respuesta del Sistema
1 – El especialista selecciona la opción de <u>Mostrar Equipo-Evento</u> .	1.1 – El sistema muestra un listado con todos los eventos de la base de datos.
2 – El especialista selecciona el evento deseado.	2.1 – El sistema localiza los datos del evento y muestra un listado con los equipos que participan en el evento seleccionado terminando el CUS.
Flujos Alternos	
Curso alternativo de eventos	
Prioridad:	Crítico

Tabla 3.11 Descripción textual. Caso de uso Gestionar Equipo-Evento.

Caso de Uso:	Gestionar Jugador-Equipo
Actores:	Comisión Nacional(Inicia)
Propósito	Permite al especialista de la Comisión Nacional Agregar, Modificar y Eliminar jugadores a un equipo que participa en un evento, así como mostrar la información de los jugadores ya agregados a los diferentes equipos del evento.
Resumen:	El CUS se inicia cuando el especialista de la Comisión Nacional selecciona la opción de Gestionar Jugador-Equipo, luego selecciona el tipo de gestión, introduce los datos necesarios, el sistema realiza la acción seleccionada y termina el CUS.
Referencia:	R11
Precondiciones:	<ol style="list-style-type: none"> 1. El especialista de la Comisión Nacional tiene que autenticarse. 2. Debe existir como mínimo un evento creado.

	3. Debe haber como mínimo un equipo registrado en el evento.
Poscondiciones:	<ol style="list-style-type: none"> 1. Jugador adicionado al equipo. 2. Información del jugador seleccionado actualizada. 3. Jugador eliminado del equipo. 4. Información de los jugadores del equipo seleccionado mostrada.
Flujo Normal de Eventos	
Acción del Actor	Respuesta del Sistema
1 – El especialista de la Comisión Nacional selecciona la opción de <u>Gestionar Jugador-Equipo</u>	1.1 – El sistema muestra las opciones: <ul style="list-style-type: none"> - <u>Adicionar Jugador-Equipo</u> - <u>Actualizar Jugador-Equipo</u> - <u>Eliminar Jugador-Equipo</u> - <u>Mostrar Jugador-Equipo</u>
Sección 1 : Adicionar Jugador-Equipo	
Acción del Actor	Respuesta del Sistema
1 – El especialista selecciona la opción de <u>Adicionar Jugador-Equipo</u> .	1.1 – El sistema muestra un listado de los eventos existentes en la base de datos.
2 – El especialista selecciona el evento donde se encuentra el equipo al cual desea agregar un jugador.	2.1 – El sistema muestra un listado con todos los equipos existentes en el evento seleccionado.
3 – El especialista selecciona el equipo donde desea agregar al jugador.	3.1 – El sistema muestra el formulario correspondiente para la inserción del jugador, así como un listado con los jugadores existentes en la base de datos.
4 – El especialista introduce los datos requeridos para la inserción del jugador y selecciona el jugador que desea insertar.	4.1 – El sistema verifica los datos insertados por el especialista. 4.2 – Si los datos son correctos el sistema procede a insertar al jugador en el equipo seleccionado y termina el

		CUS.
Flujos Alternos		
Curso Alternativo de Eventos	4.2 – Si los datos introducidos por el especialista son incorrectos el sistema le muestra un mensaje de error indicándole donde está el dato erróneo e indica al especialista retornar a la acción 4.	
Sección 2: Actualizar Jugador-Equipo		
Acción del Actor		Respuesta del Sistema
1 – El especialista selecciona la opción de <u>Actualizar Jugador-Equipo</u> .		1.1 – El sistema le muestra un listado con todos los eventos existentes en la base de datos.
2 – El especialista selecciona el evento al que pertenece el equipo donde se encuentra el jugador a actualizar.		2.1 – El sistema muestra un listado con todos los equipos que participan en el evento seleccionado.
3 – El especialista selecciona el equipo donde se encuentra el jugador a actualizar.		3.1 – El sistema muestra un listado con todos los jugadores que pertenecen al equipo seleccionado.
4 – El especialista selecciona el jugador que desea modificar.		4.1 – El sistema muestra los datos del jugador listo para modificar.
5 – El especialista modifica los datos del jugador.		5.1 – El sistema verifica los datos modificados. 5.2 – Si los datos son correctos el sistema procede a actualizar la información del jugador modificado en la base de datos y termina el CUS.
Flujos Alternos		
Curso alternativo de eventos	5.2 – Si los datos introducidos por el especialista son incorrectos el sistema muestra un mensaje de error indicando donde está el dato erróneo e indica al especialista retornar a la acción 5.	
Sección 3: Eliminar Jugador-Equipo		
Acción del Actor		Respuesta del Sistema
1 – El especialista selecciona la opción de		1.1 – El sistema le muestra un listado con los eventos

<u>Eliminar Jugador-Equipo.</u>		existentes en la base de datos.
2 – El especialista selecciona el evento a que pertenece el equipo donde se encuentra el jugador que desea eliminar.		2.1 –El sistema muestra un listado de los equipos pertenecientes al evento seleccionado.
3 – El especialista selecciona el equipo donde se encuentra el jugador a eliminar.		3.1 – El sistema muestra un listado con los jugadores que pertenecen al equipo seleccionado.
4 – El especialista selecciona el jugador que desea eliminar.		4.1 – El sistema muestra los datos del jugador listos para eliminar.
5 – El especialista selecciona la opción “Eliminar”.		5.1 – El sistema muestra un mensaje para que el especialista confirme su solicitud.
6 – El especialista confirma si desea o no eliminar el jugador del equipo.		6.1 – Si el especialista confirma que desea eliminar el jugador del equipo el sistema procede a eliminar los datos de la base de datos y termina el CUS.
Flujos Alternos		
Curso alternativo de eventos	6.1 – Si el especialista cancela la acción se culmina el CUS sin ejecutar ninguna acción.	
Sección 4: Mostrar Jugador-Equipo		
Acción del Actor		Respuesta del Sistema
1 – El especialista selecciona la opción de <u>Mostrar Jugador-Equipo.</u>		1.1 – El sistema muestra un listado con todos los eventos de la base de datos.
2 – El especialista selecciona el evento donde se encuentra el equipo que desea mostrar sus jugadores.		2.1 – El sistema muestra un listado con todos los equipos registrados en el evento seleccionado.
3 – El especialista selecciona el equipo deseado.		3.1 – El sistema localiza los datos del equipo y muestra un listado con los jugadores que pertenecen al equipo terminando el CUS.
Flujos Alternos		
Curso alternativo de eventos		

Prioridad:	Crítico

Tabla 3.12 Descripción textual. Caso de uso Gestionar Jugador-Equipo.

Caso de Uso:	Gestionar Alineación- Juego
Actores:	Informático(Inicia)
Propósito	Permite al informático Agregar, Modificar, Eliminar y Mostrar las alineaciones de un juego determinado.
Resumen:	El CUS se inicia cuando el informático selecciona la opción de Gestionar Alineación-Juego, luego selecciona el tipo de gestión, introduce los datos necesarios, el sistema realiza la acción seleccionada y termina el CUS.
Referencia:	R22
Precondiciones:	<ol style="list-style-type: none"> 1. El informático tiene que autenticarse. 2. Tiene que existir como mínimo un juego en la base de datos.
Poscondiciones:	<ol style="list-style-type: none"> 1. Alineación adicionada al juego. 2. Alineación del juego modificada. 3. Alineación eliminada del juego. 4. Información de las alineaciones del juego seleccionado mostrada.

Flujo Normal de Eventos

Acción del Actor	Respuesta del Sistema
1 – El informático selecciona la opción de <u>Gestionar Alineación-Juego.</u>	1.1 – El sistema muestra las opciones: - <u>Adicionar Alineación -Juego</u> - <u>Actualizar Alineación -Juego</u> - <u>Eliminar Alineación -Juego</u> - <u>Mostrar Alineación -Juego</u>

Sección 1 : Adicionar Alineación-Juego

Acción del Actor	Respuesta del Sistema
------------------	-----------------------

1 – El informático selecciona la opción de <u>Adicionar Alineación -Juego.</u>	1.1 – El sistema muestra el formulario correspondiente para la inserción de la alineación.
2 – El informático inserta los datos en el formulario.	2.1 – El sistema verifica los datos insertados por el informático. 2.2 – Si los datos son correctos el sistema procede a insertar la alineación en el juego seleccionado y termina el CUS.
Flujos Alternos	
Curso Alternativo de Eventos	2.2 – Si los datos introducidos por el informático son incorrectos el sistema le muestra un mensaje de error indicándole donde está el dato erróneo e indica al informático retornar a la acción 2.
Sección 2: Actualizar Alineación-Juego	
Acción del Actor	Respuesta del Sistema
1 – El informático selecciona la opción de <u>Actualizar Alineación-Juego.</u>	1.1 – El sistema muestra las alineaciones pertenecientes al juego.
2 – El informático selecciona la alineación que desea modificar.	2.1 – El sistema muestra los datos de la alineación seleccionada listos para modificar.
3 – El informático modifica los datos de la alineación.	3.1 – El sistema verifica los datos modificados. 3.2 – Si los datos son correctos el sistema procede a actualizar la información de la alineación modificada y termina el CUS.
Flujos Alternos	
Curso alternativo de eventos	3.2 – Si los datos introducidos por el informático son incorrectos el sistema muestra un mensaje de error indicando donde está el dato erróneo e indica al informático retornar a la acción 3.
Sección 3: Eliminar Alineación-Juego	
Acción del Actor	Respuesta del Sistema
1 – El informático selecciona la opción de	1.1 – El sistema muestra las alineaciones

<u>Eliminar Alineación-Juego.</u>		correspondientes al juego.
2 – El informático selecciona la alineación que desea eliminar.		2.1 – El sistema muestra los datos de la alineación listos para ser eliminados.
3 – El informático selecciona la opción “Eliminar”.		3.1 – El sistema muestra un mensaje para que el informático confirme su solicitud.
4 – El informático confirma si desea o no eliminar la alineación del juego.		4.1 – Si el informático confirma que desea eliminar la alineación del juego, el sistema procede a eliminar los datos de la base de datos y termina el CUS.
Flujos Alternos		
Curso alternativo de eventos	4.1 – Si el informático cancela la acción se culmina el CUS sin ejecutar ninguna acción.	
Sección 4: Mostrar Alineación-Juego		
Acción del Actor		Respuesta del Sistema
1 – El informático selecciona la opción de <u>Mostrar Alineación-Juego.</u>		1.1 – El sistema muestra un listado con las alineaciones que pertenecen al juego terminando el CUS.
Flujos Alternos		
Curso alternativo de eventos		
Prioridad:	Crítico	

Tabla 3.13 Descripción textual. Caso de uso Gestionar Alineación-Juego.

Caso de Uso:	Gestionar Etapas del Evento
Actores:	Comisión Nacional(Inicia)
Propósito	Permite al especialista de la Comisión Nacional Agregar, Modificar, Eliminar y Mostrar los datos de una etapa de un evento
Resumen:	El CUS se inicia cuando el especialista de la Comisión Nacional selecciona la opción de Gestionar Etapas del Evento, luego selecciona el tipo de gestión,

	introduce los datos necesarios, el sistema realiza la acción seleccionada y termina el CUS.
Referencia:	R28
Precondiciones:	<ol style="list-style-type: none"> 1. El especialista de la Comisión Nacional tiene que autenticarse. 2. Tiene que existir como mínimo una etapa creada en la base de datos.
Poscondiciones:	<ol style="list-style-type: none"> 1. Etapa adicionada al evento. 2. Etapa actualizada en el evento. 3. Etapa eliminada del evento. 4. Información de etapa del evento seleccionado mostrada.
Flujo Normal de Eventos	
Acción del Actor	Respuesta del Sistema
1 – El especialista de la Comisión Nacional selecciona la opción de <u>Gestionar Etapas del Evento</u> .	1.1 – El sistema muestra las opciones: <ul style="list-style-type: none"> - <u>Adicionar Etapas del Evento</u> - <u>Actualizar Etapas del Evento</u> - <u>Eliminar Etapas del Evento</u> - <u>Mostrar Etapas del Evento</u>
Sección 1 : Adicionar Etapas del Evento	
Acción del Actor	Respuesta del Sistema
1 – El especialista selecciona la opción de <u>Adicionar Etapas del Evento</u> .	1.1 – El sistema muestra un listado con todos los eventos existentes en la base de datos.
2 – El especialista selecciona la etapa a la cual desea adicionar la nueva etapa.	2.1 – El sistema muestra el formulario correspondiente para la inserción de una nueva etapa.
3 – El especialista inserta los datos en el formulario.	3.1 – El sistema verifica los datos insertados por el especialista. 3.2 – Si los datos son correctos el sistema procede a insertar la etapa en el evento seleccionado y termina el

		CUS.
Flujos Alternos		
Curso Alternativo de Eventos	3.2 – Si los datos introducidos por el especialista son incorrectos el sistema le muestra un mensaje de error indicándole donde está el dato erróneo e indica al especialista retornar a la acción 3.	
Sección 2: Actualizar Etapas del Evento		
Acción del Actor		Respuesta del Sistema
1 – El especialista selecciona la opción de <u>Actualizar Etapas del Evento.</u>		1.1 – El sistema le muestra un listado con todos los eventos existentes en la base de datos.
2 – El especialista selecciona el evento que desea actualizar.		2.1 – El sistema muestra las etapas pertenecientes al evento seleccionado.
3 – El especialista selecciona la etapa que desea modificar.		3.1 – El sistema muestra los datos de la etapa seleccionada listos para modificar.
4 – El especialista modifica los datos de la etapa.		4.1 – El sistema verifica los datos modificados. 4.2 – Si los datos son correctos el sistema procede a actualizar la información de la etapa modificada y termina el CUS.
Flujos Alternos		
Curso alternativo de eventos	4.2 – Si los datos introducidos por el especialista son incorrectos el sistema muestra un mensaje de error indicando donde está el dato erróneo e indica al especialista retornar a la acción 4.	
Sección 3: Eliminar Etapas del Evento		
Acción del Actor		Respuesta del Sistema
1 – El especialista selecciona la opción de <u>Eliminar Etapas del Evento.</u>		1.1 – El sistema le muestra un listado con los eventos existentes en la base de datos.
2 – El especialista selecciona el evento del cual desea eliminar la etapa.		2.1 – El sistema muestra las etapas correspondientes al evento seleccionado.
3 – El especialista selecciona la etapa que desea eliminar.		3.1 – El sistema muestra los datos de la etapa listos para ser eliminados.

4 – El especialista selecciona la opción “Eliminar”.	4.1 – El sistema muestra un mensaje para que el especialista confirme su solicitud.
5 – El especialista confirma si desea o no eliminar la etapa del evento.	5.1 – Si el especialista confirma que desea eliminar la etapa del evento, el sistema procede a eliminar los datos de la base de datos y termina el CUS.
Flujos Alternos	
Curso alternativo de eventos	5.1 – Si el especialista cancela la acción se culmina el CUS sin ejecutar ninguna acción.
Sección 4: Mostrar Etapas del Evento	
Acción del Actor	Respuesta del Sistema
1 – El especialista selecciona la opción de <u>Mostrar Etapas del Evento</u> .	1.1 – El sistema muestra un listado con todos los eventos de la base de datos.
2 – El especialista selecciona el evento deseado.	2.1 – El sistema localiza los datos del evento y muestra un listado con las etapas que pertenecen a dicho evento terminando el CUS.
Flujos Alternos	
Curso alternativo de eventos	
Prioridad:	Crítico
Flujos Alternos	

Tabla 3.14 Descripción textual. Caso de uso Gestionar Etapas del Evento.

Anexo 4. Diagramas de clases del diseño para cada caso de uso.

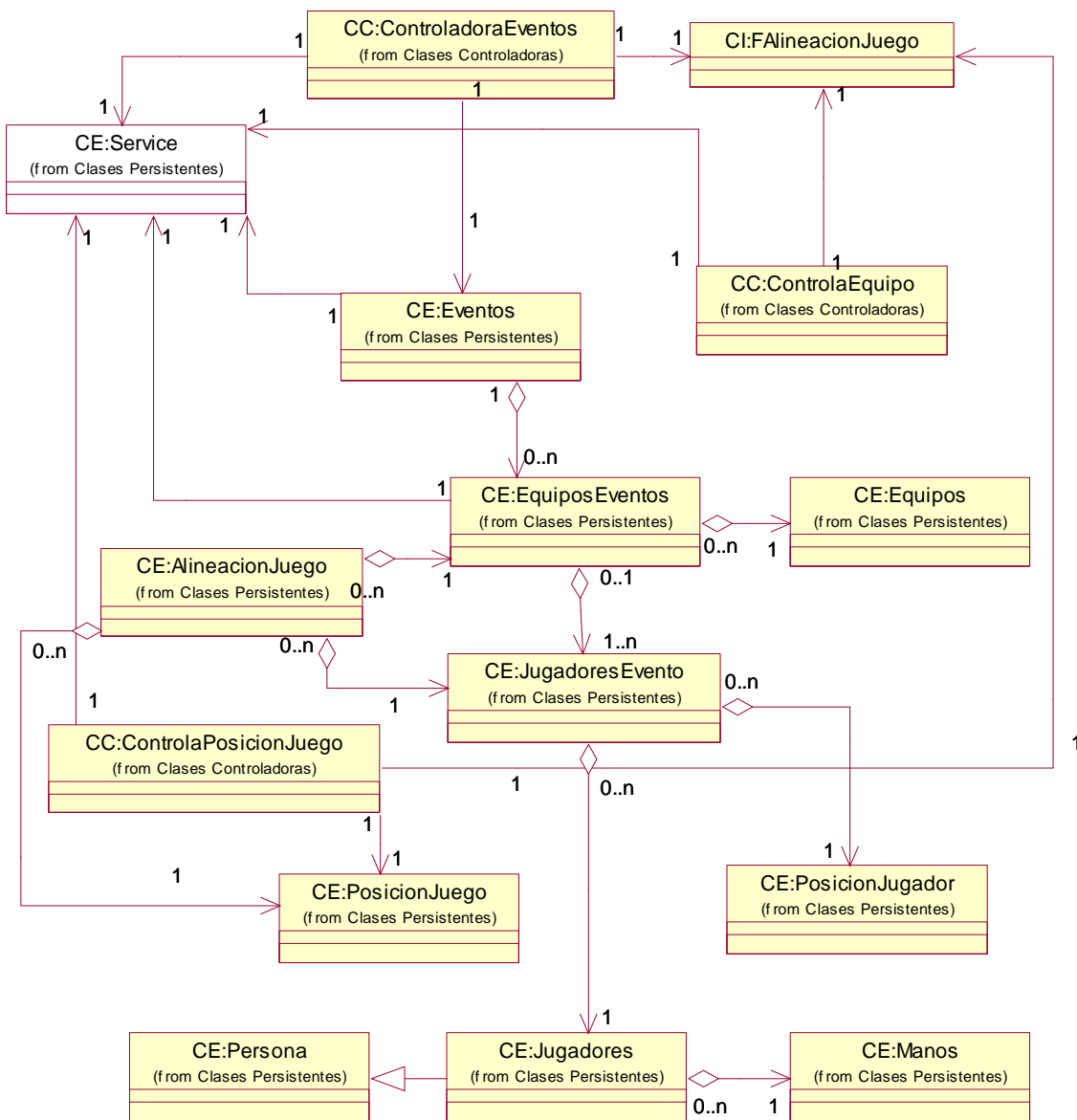


Figura 4.1 Diagrama de clases del diseño. Caso de uso Gestionar Alineación-Juego.

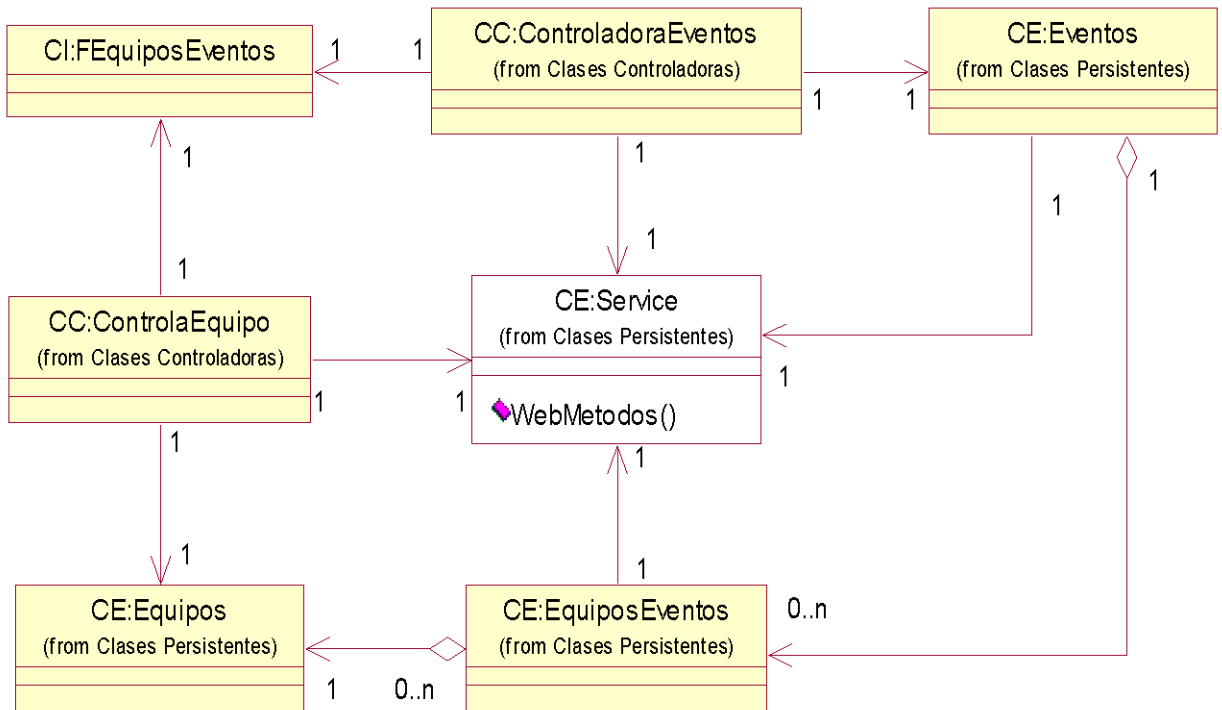


Figura 4.2 Diagrama de clases del diseño. Caso de uso Gestionar Equipo-Evento.

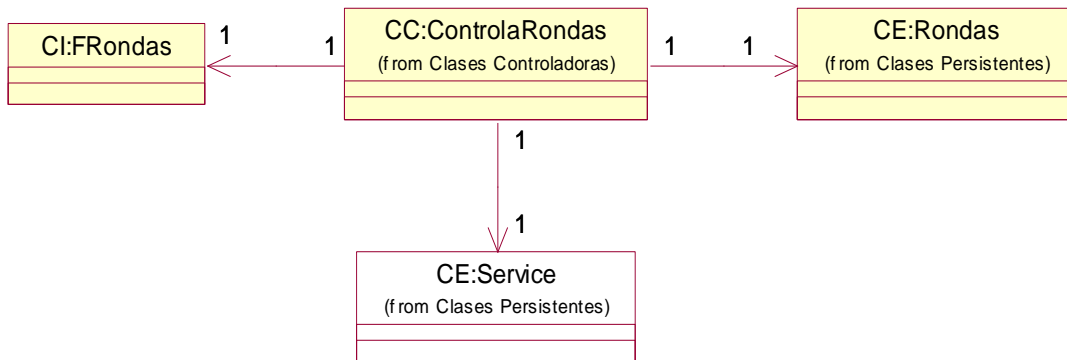


Figura 4.3 Diagrama de clases del diseño. Caso de uso Gestionar Etapas.

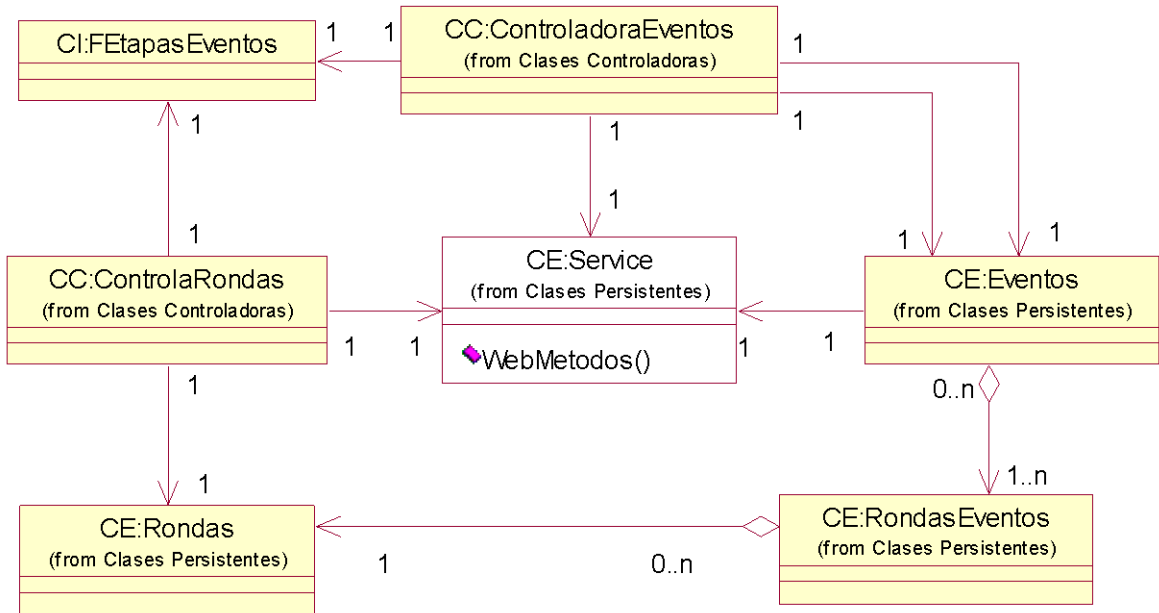


Figura 4.4 Diagrama de clases del diseño. Caso de uso Gestionar Etapas-Evento.

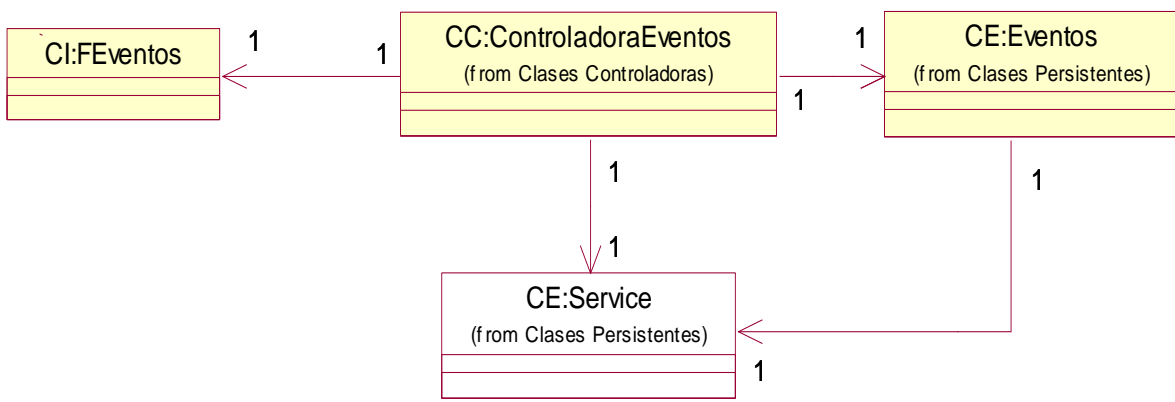


Figura 4.5 Diagrama de clases del diseño. Caso de uso Gestionar Evento.

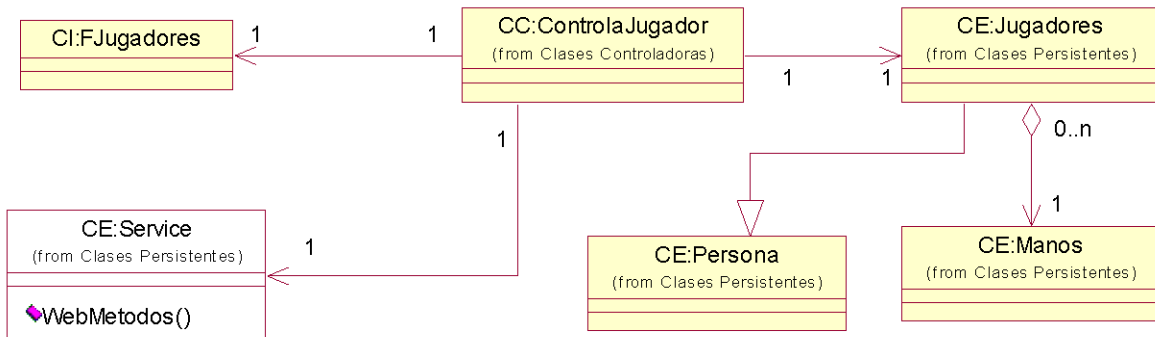


Figura 4.6 Diagrama de clases del diseño. Caso de uso Gestionar Jugador.

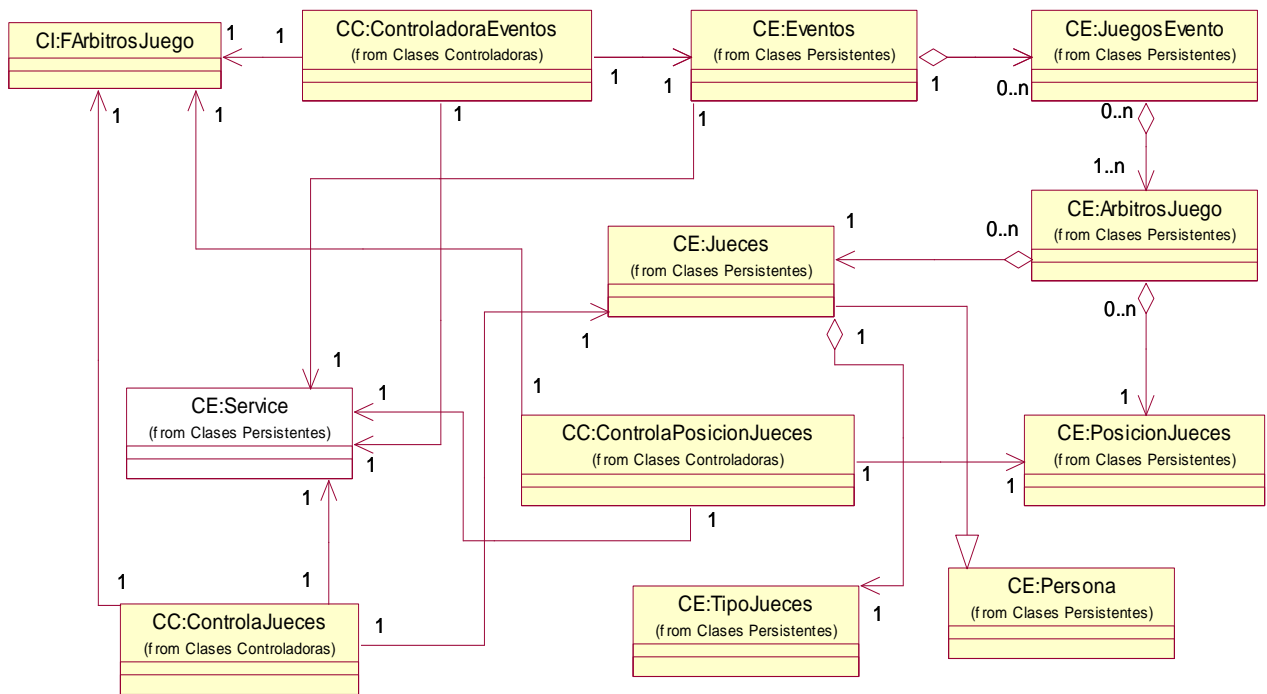


Figura 4.7 Diagrama de clases del diseño. Caso de uso Gestionar Jueces-Juego.

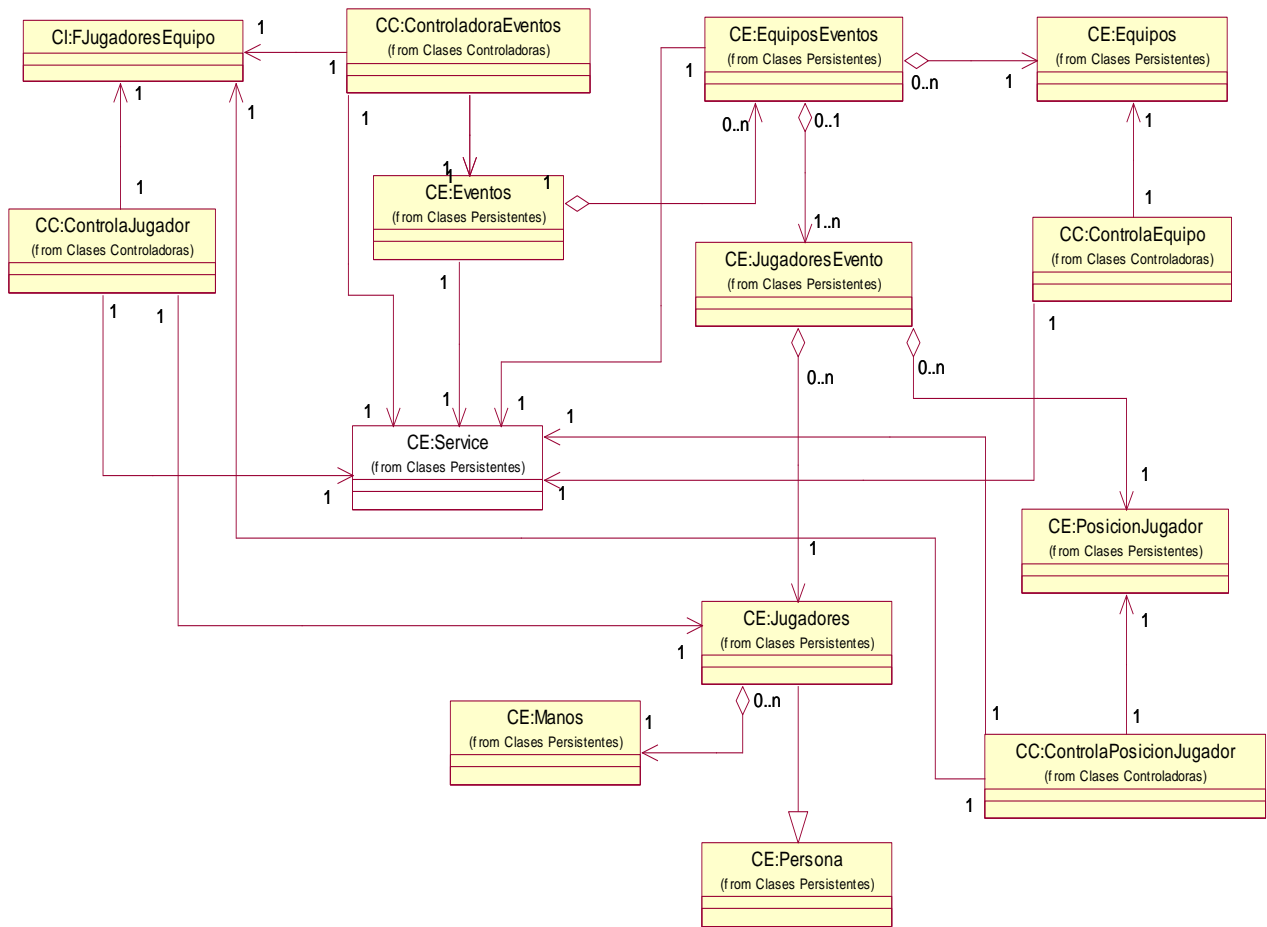


Figura 4.8 Diagrama de clases del diseño. Caso de uso Gestionar Jugador-Equipo.

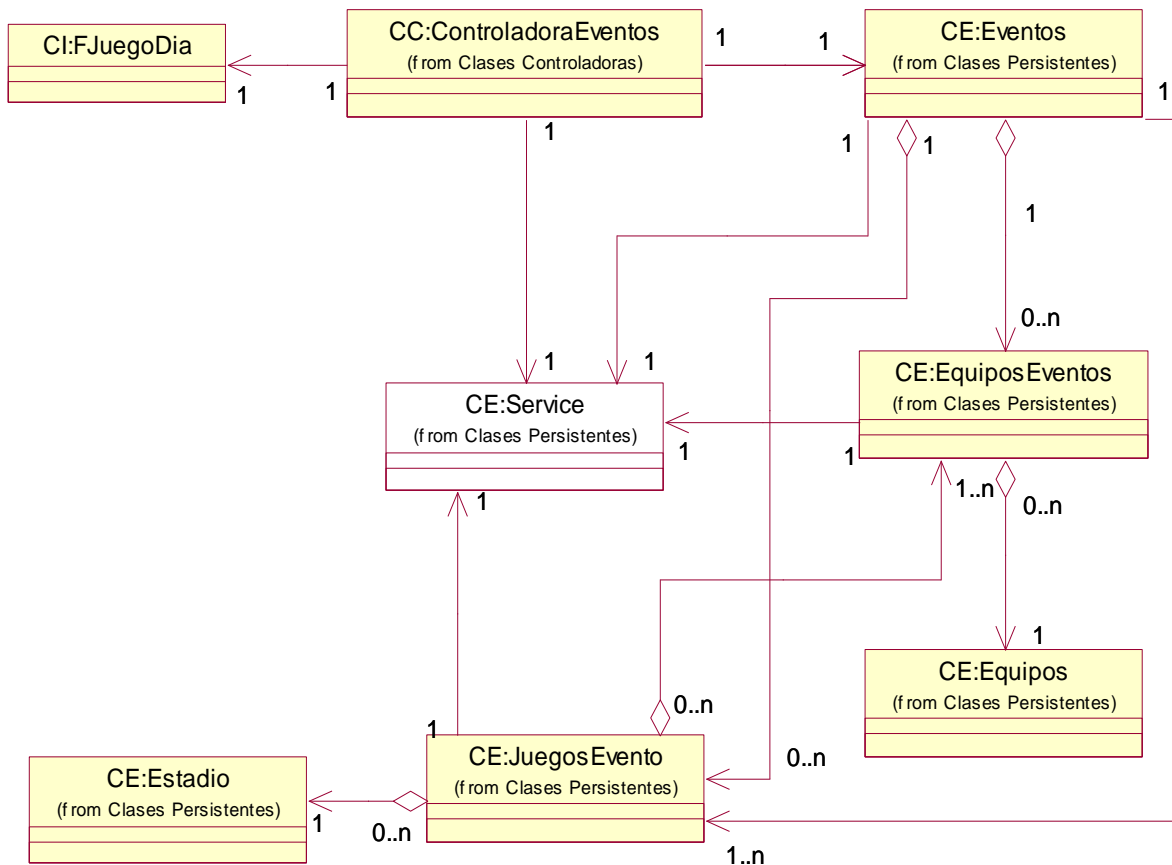


Figura 4.9 Diagrama de clases del diseño. Caso de uso Mostrar Juego del Día.

Anexo 5. Diagrama de clases persistentes.

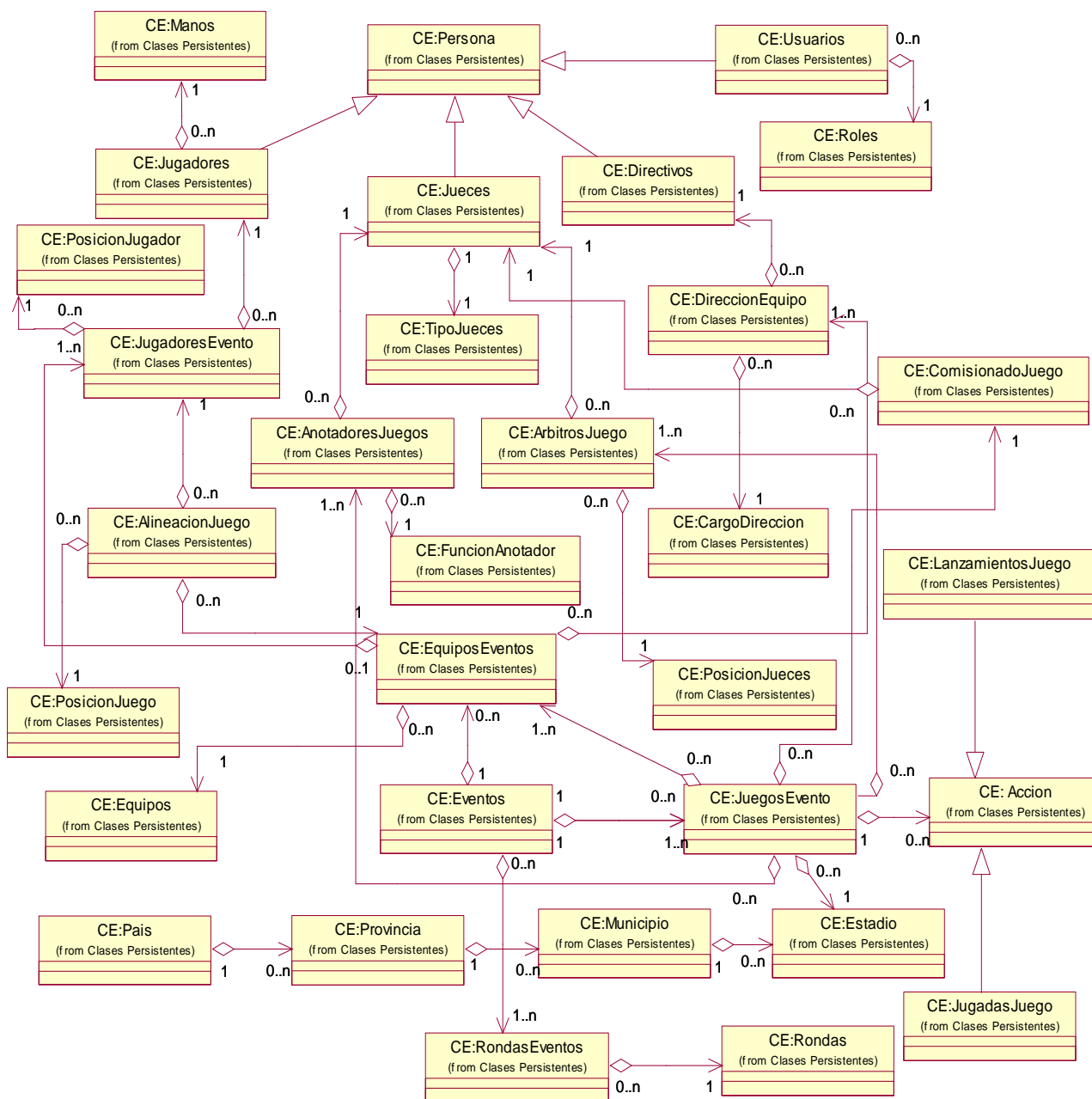


Figura 4.10 Diagrama de clases persistentes.

Anexo 6. Modelo de Datos.

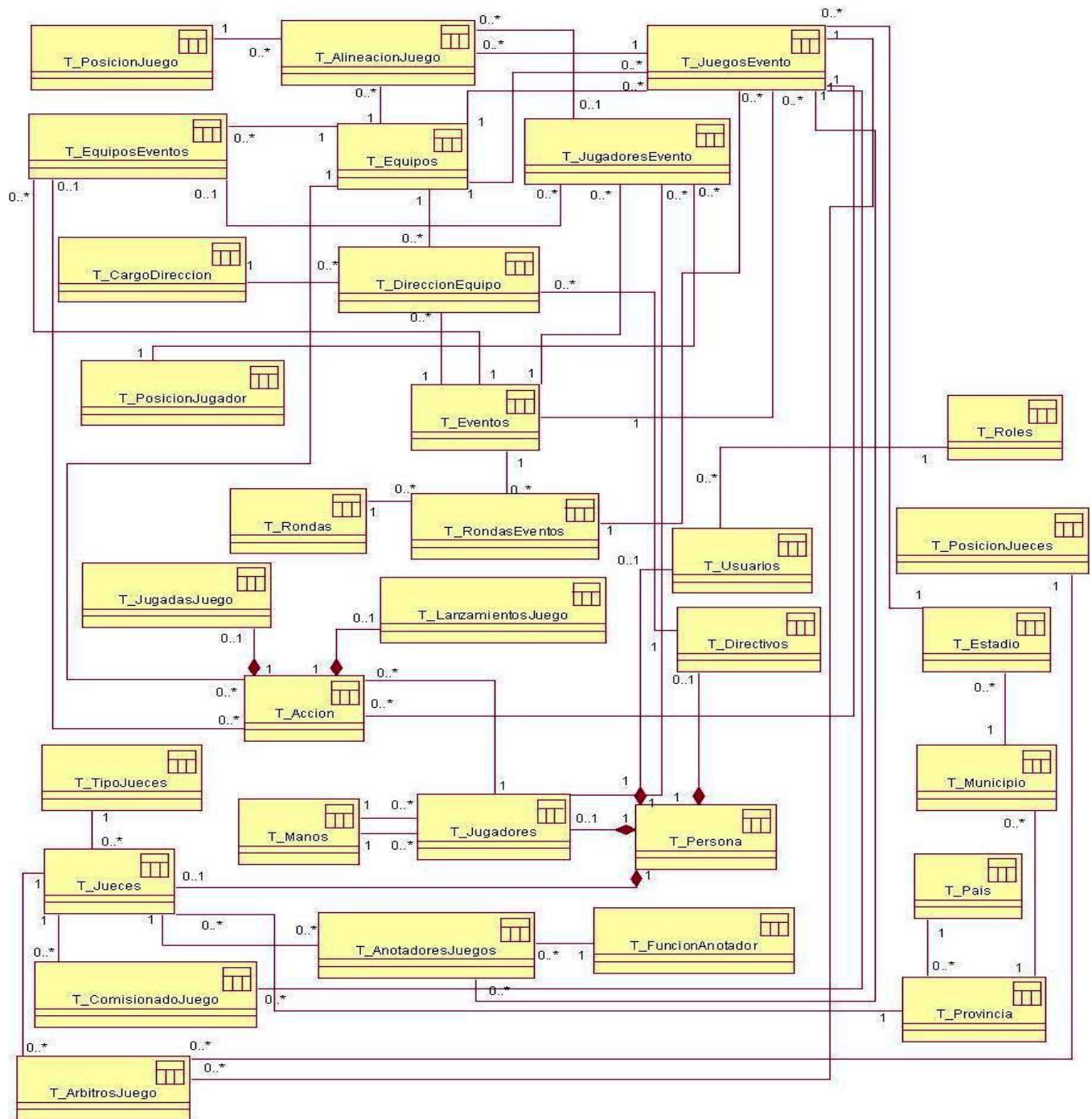


Figura 4.11 Modelo de Datos.

Anexo 7. Modelo de Componentes.

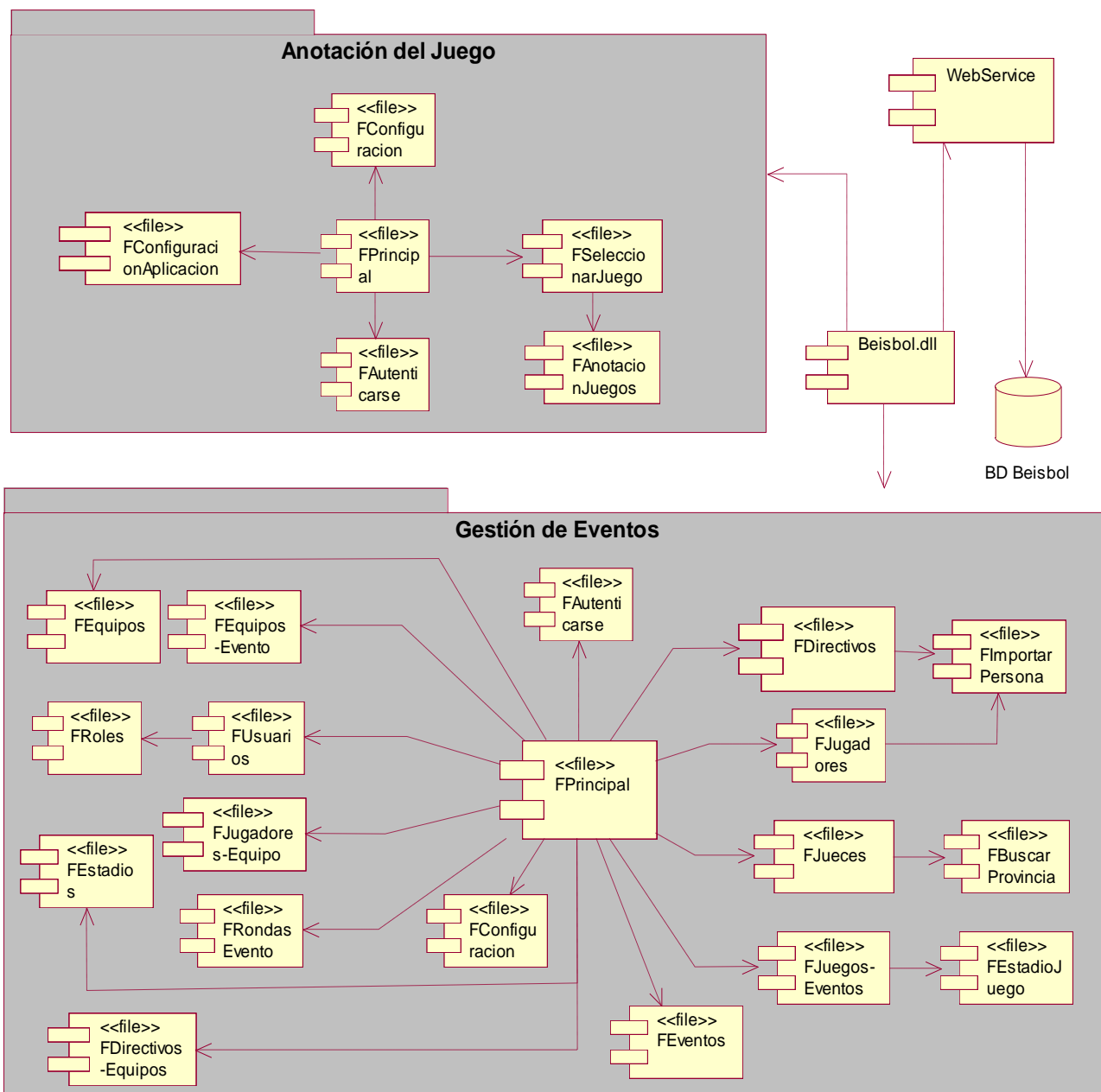


Figura 4.13 Modelo de Componentes.